ITC 3/54
Information Technology and Control
Vol. 54 / No. 3/ 2025
pp. 900-917
DOI 10.5755/j01.itc.54.3.38966

Optimized YOLOv8 for Lightweight Floating Object Detection on Unmanned Surface Vehicles

Received 2024/09/29 | Accepted after revision 2025/02/11

HOW TO CITE: Dong, H., Li, D., Zhang, S., Li, B., Chen, X., Wu, X. (2025). Optimized YOLOv8 for Lightweight Floating Object Detection on Unmanned Surface Vehicles. *Information Technology and Control*, 54(3), 900-917. https://doi.org/10.5755/j01.itc.54.3.38966

Optimized YOLOv8 for Lightweight Floating Object Detection on Unmanned Surface Vehicles

Honghong Dong , Dan Li, Shuailong Zhang
College of The Academy of Digital China, Fuzhou University, Fuzhou 350003, China

Binjie Li, Xin Chen
The College of Computer and Data Science, Fuzhou University, Fuzhou 350003, China

Xiaozhu Wu*
College of The Academy of Digital China, Fuzhou University, Fuzhou 350003, China
The College of Computer and Data Science, Fuzhou University, Fuzhou 350003, China

Corresponding author: wxz@fzu.edu.cn

Unmanned surface vehicles (USVs) are increasingly being applied in water environment protection and management. A primary function is recognizing and detecting floating objects in aquatic environments. However, water surface floating object detection from USVs faces challenges such as high scene complexity, including sunlight reflection and shoreline reflections, in addition to identifying small objects. To tackle these issues, this study presents an improved YOLOv8s method for water surface floating object detection, named PSP-YOLOv8s. Firstly, we integrated the original C2f module with the Polarized Self-Attention (PSA) mechanism to design the C2f-PSA structure, thereby improving the model's ability to extract features in intricate environments. Secondly, we add a detection head specialized for small objects by fusing deep and shallow features, which effectively reduces the miss rate for small objects. Meanwhile, the Partial Convolution (PConv) technique is used to reconstruct the detection head, making the model lightweight. Finally, the Wise-IoUv3(WIoUv3) loss function is introduced to mitigate the impact of low-quality anchor frames in complex environments. Experimental results demonstrate that PSP-YOLOv8s achieves improvements of 4.3% in AP, 3.8% in AP_{50}, and a significant 12.9% in AP_S on the self-constructed USV-WSFO dataset. The model's parameters, computational overhead, and size were reduced by 8.1%, 4.2%, and 2.8%, respectively. The proposed model's generalization capability is further validated through experiments on the Orca dataset and field trials. This work extends the application of vision technology in USVs, providing

significant support for water resource and ecosystem protection. Code is available at https://github.com/hongh07/PSP-YOLOv8s.

KEYWORDS: USV application, Object detection, Small object, YOLOv8s, Floating object detection.

1. Introduction

Floating objects are commonly found in ponds, lakes, rivers, and oceans, contributing significantly to water pollution [23]. Common examples of floating debris, such as bottles and plastic bags, not only disrupt the ecological balance of water bodies but also endanger aquatic ecosystems and human well-being [12]. As water pollution continues to worsen, the need for effective methods to detect and remove floating objects from water surfaces has become more urgent. Traditional approaches to managing surface floating debris primarily rely on manual labor, which is both inefficient and costly [27]. However, due to the rapid progress in artificial intelligence technologies and convolutional neural networks, unmanned surface vehicles (USVs) have demonstrated great potential for water quality monitoring and surface objects removal [1, 9]. Consequently, water surface object detection is becoming increasingly crucial for USVs and their aquatic vision applications.

Detecting floating objects in water remains a challenge for autonomous waste collection systems like USVs. Elements such as water surface fluctuations, sunlight reflections, and shoreline reflections contribute to significant background clutter. Additionally, the small size of floating objects and the fact that distant objects appear smaller in RGB images due to the wide viewing angle of the USV camera further complicate detection [5]. These factors can result in undetected objects.

Previous research has made significant progress in detecting waterborne objects [13, 26], with many studies focusing primarily on improving detection accuracy. However, these efforts often neglect the computational efficiency and lightweight design of detection algorithms in practical applications. In particular, on resource-constrained USVs, balancing computational efficiency with detection accuracy remains a critical issue to resolve.

This study provides a lightweight approach based on YOLOv8 for detecting floating objects on water surfaces, aiming to balance accuracy and speed. The algorithm effectively locates and identifies floating objects, greatly aiding in the conservation of water resources and environmental protection.

In summary, the primary contributions of this article are outlined as follows:

1. We constructed an inland water surface floating object dataset from the viewpoint of a USV, named the USV-WSFO dataset, containing 1200 accurately labeled images of eight different types of water floaters.
2. We propose an algorithm called PSP-YOLOv8s for water float detection in USVs. In YOLOv8s, we developed the C2f-PSA module to improve the backbone network's feature extraction performance in challenging environments. Meanwhile, the original detection layer was improved by incorporating a layer that specifically focuses on small objects, enhancing its ability to collect feature information from them. The detection head was reconstructed by introducing a lightweight partial convolution. Finally, the WIoUv3 loss function is employed to prioritize the general mass anchor frame.
3. We evaluated the PSP-YOLOv8s algorithm against eight other deep learning models on both the homemade USV-WSFO dataset and the publicly available FloW-Img dataset. Significant improvements in performance, cost-efficiency, and model complexity are demonstrated by the experimental findings of the PSP-YOLOv8s approach.

2. Related Work

2.1. Object Detection

The objective of object detection, one of computer vision's numerous crucial tasks, is to recognize specific object categories in images [39]. Two primary advancements in object detection have occurred over the past two decades. Before 2014, object detection depended on traditional methods that employed manually constructed features and sliding windows. Convolutional Neural Networks (CNNs) are now widely regarded as the benchmark in deep learning for object detection [43]. Traditional methods primarily relied on the Vi-

ola-Jones detector [34], the Histogram of Oriented Gradients (HOG) detector [7], and the Deformable Part-based Model (DPM) [8]. These methods often suffered from poor accuracy and efficiency, as feature extraction and candidate region scaling were manually designed [40]. Deep convolutional networks, leveraging their strong discriminative ability and capacity to learn from data, have significantly advanced object detection, improving both accuracy and efficiency.

There are mainly two categories of object detectors that utilize deep learning [17]. Two-stage detectors, including R-CNN [11], Fast R-CNN [10], Faster R-CNN[25] and Mask R-CNN [14], attain high-precision detection via region proposals and CNN-based classification. In contrast, one-stage detectors provide benefits such as expedited detection speed and reduced model size [24], exemplified by algorithms like OverFeat [29], the YOLO series [31, 35, 37], SSD [22] and CornerNet [18].

Thanks to its optimized network architecture and efficient computational modules, YOLOv8 achieves high accuracy with a small model size and reduced computational resource requirements, making it stand out among one-stage detectors [15]. These features make it particularly well-suited for object detection on USVs. Based on this, this study proposes an improved YOLOv8s algorithm for water surface floating object detection.

2.2. Object Detection for USVs

Vision-based methods are commonly used to detect surface objects for USVs in marine environments. Wang et al. [38] introduced the Marine Vessel Detection Dataset (MVDD13), which serves as essential support for object detection in USVs. The dataset comprises 35,474 images spanning 13 distinct vessel categories. Zhang et al. [41] introduced an improved version of YOLOv5, specifically designed for object detection on USVs in complex marine environments. By integrating the Ghost and Transformer modules, this approach optimizes feature extraction and minimizes model complexity. Consequently, the accuracy increases by 1.3%, and the model size is compressed to 12.24 MB. Cai et al. [2] employed the Multi-modal Marine Obstacle Detection Dataset2 (MODD2) to assess the performance of their proposed lightweight water-obstacle detection network, LWDNet. Their results demonstrate that LWDNet significantly enhances image inference speed while maintaining water-obstacle detection accuracy at a comparable level. Additionally, Wang et al. [36] proposed the Shuffle-High-Resolution-Net (SHR det). This model incorporates an enhanced Shuffle Block, a small feature fusion module, and the Focal Efficient Intersection over Union loss. It is optimized for incremental few-shot detection and small object identification on the sea surface. USVs in inland waters are garnering increasing attention due to their potential applications. As an example, the first dataset (FLoW) for floating waste detection was created by Cheng et al. [6]. It is collected from the perspective of USVs operating in real-world inland waters under various conditions. Sravanthi et al. [30] validated their proposed low-cost method for detecting weeds in complex inland water environments using their own datasets, which included floating plants, small cluster plants, and birds, as well as OSF datasets. Currently, there is a paucity of studies regarding inland water object detection with USV platforms.

2.3. Object Detection of Water Surface Floating Objects

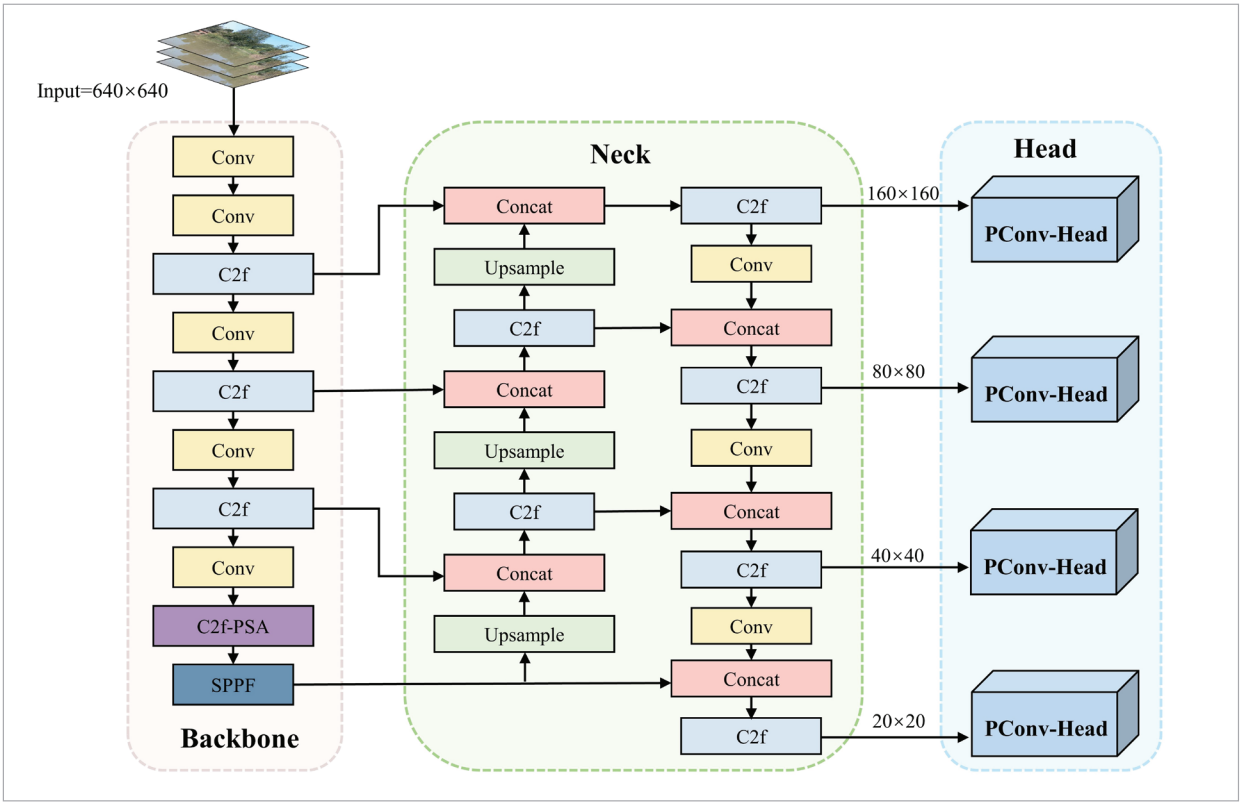
More researchers are using deep learning for floating object detection. For instance, Van et al. [33] utilized a dataset derived from footage captured by cameras installed on bridges across five distinct river locations in Jakarta, Indonesia. They employed Faster R-CNN to identify regions potentially containing plastic floating objects. Chen et al. [4] established 26 fixed CCD cameras along river sections in Deqing City, Zhejiang Province, China, to capture floating objects on the water, primarily including fishing boats, water hyacinths, floating weeds, and plastic bottles. They proposed the SSD-FT algorithm to tackle challenges such as complex detection scenarios and varying sizes of floating objects. Li et al. [19] curated a dataset encompassing floating waste commonly found in natural water bodies, primarily including plastic bottles, cans, tetra packs, and plastic bags. To enhance the efficiency and accuracy of detecting and managing objects in water, they later proposed an improved deep learning method based on Faster R-CNN to facilitate effective identification and categorization of floating litter. Jang et al. [16] utilized Unmanned Aerial Vehicles (UAVs) imagery coupled with an optimized YOLOv5n model to detect floating objects. The enhanced model was evaluated on Jetson Nano, reaching a mean average precision (mAP) of 87.2%.

3. Methodology

3.1. Overview of the Proposed Model

In 2023, the Ultralytics team released YOLOv8, which offers significant improvements in both performance and efficiency over previous versions. This algorithm employs a single-stage detector architecture that integrates a lightweight backbone network with efficient detection heads, enabling fast target detection and localization. The primary difference between YOLOv8 and YOLOv5 is that YOLOv8's backbone network is more lightweight and supports a richer gradient flow, as a result of substituting the C3 module with the C2f module. The pooling layer uses SPPF instead of the traditional SPP, effectively reducing computational costs and improving efficiency. The Neck component of the network incorporates PANet, which integrates feature information across multiple levels. The detection head follows YOLOX's decoupled head design. Additionally, YOLOv8 features an Anchor-Free structure that adapts to different object sizes, enhancing detection accuracy. It maintains a stable network architecture across all versions, with model sizes (N, S, M, L, and X) adjustable through specific scaling factors. These characteristics contribute to YOLOv8's efficiency and flexibility, making it well-suited for detecting floating objects on USV platforms, even in challenging environments with complex backgrounds and numerous small objects. Therefore, this paper proposes the PSP-YOLOv8s algorithm, an enhancement of YOLOv8. The main improvement strategies include (1) incorporating the PSA mechanism into the C2f module before the SPPF in the YOLOv8s backbone to improve convolutional feature extraction; (2) establishing a small object detection head to enhance YOLOv8s's capability in capturing small object features; (3) employing PConv to reconstruct the detection head, reducing computational overhead and storage costs; and(4) utilizing WIoUv3 to prioritize general mass object frames, thereby enhancing model convergence speed and accuracy. Figure 1 illustrates the upgraded network architecture in detail.

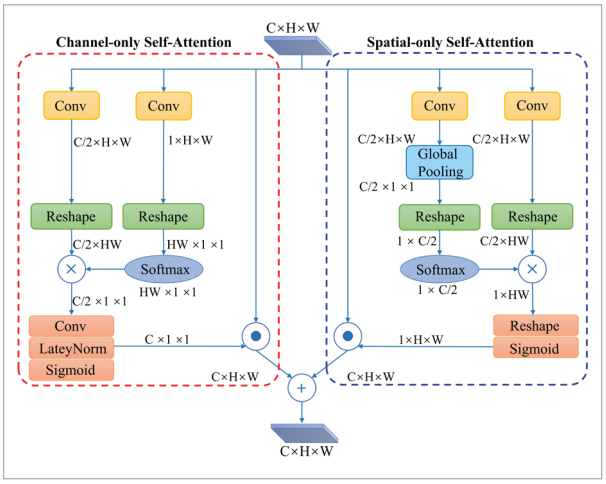
Figure 1
Diagram of PSP-YOLOv8s network structure.



3.2. Enhanced Feature Extraction via C2f-PSA Module

To solve the difficulties given by complex backgrounds, such as sunlight reflection and shoreline reflections, and to overcome the limitations of the original YOLOv8 algorithm's feature extraction in these environments, we introduce the Polarized Self-Attention (PSA) mechanism [21]. This mechanism effectively captures the contextual dependencies between adjacent regions in the feature maps, as depicted in Figure 2.

Figure 2
The architecture diagram of the PSA module.



The PSA module generates attention maps in both channel and spatial dimensions based on a given feature map $x \in R^{C\times H\times W}$, which consists of two subsets: channel-only self-attention and spatial-only self-attention. To enhance adaptive feature representation, the attention maps are then multiplied by the input feature map.

Based on the definition in Equation (2), the channel attention mechanism module collects all of the detected object's content and extracts its contour features.

$$F_{SM}(X)=\sum_{j=1}^{N_p}\frac{e^{x}j}{\sum_{m=1}^{N_p}e^{x}m} \quad (1)$$

$$A^{ch}(X)=F_{SG}\left[W_{z|\theta_1}\left(\sigma_1(W_v(X))\times F_{SM}\left(\sigma_2\left(W_q(X)\right)\right)\right)\right], \quad (2)$$

where W_z, W_v and W_q denote different 1×1 convolution operations; σ_1 and σ_2 denote different reshape operations; "×" denotes the dot product operation of matrix; $F_{SM}(\cdot)$ denotes the softmax operation; the internal channel count in W_z, W_v and W_q is C/2, and the output is $Z^{ch}=A^{ch}(X)\odot^{ch}X$, where $A^{ch}(X)\in R^{C\times 1\times 1}$, $Z^{ch}\in R^{C\times H\times W}$, $\odot^{ch}$ denotes the channel-wise multiplication operation.

Equation (4) demonstrates that the spatial attention mechanism can enhance object detection accuracy and precisely localize detected targets.

$$F_{GP}(X)=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}X(:,i,j) \quad (3)$$

$$A^{sp}(X)=F_{SG}\left[\sigma_3\left(F_{SM}\left(\sigma_1\left(F_{GP}\left(W_q(X)\right)\right)\right)\times\sigma_2(W_v(X))\right)\right], \quad (4)$$

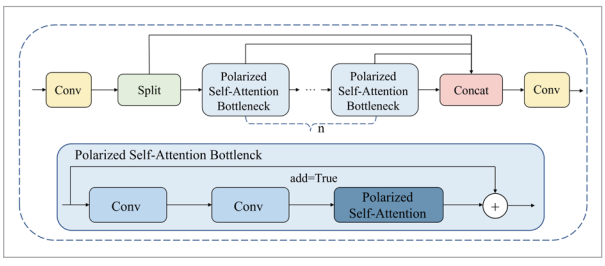
where W_q and W_v denote different 1×1 convolution operations; σ_1, σ_2, σ_3 denote different Reshape operations; "×" denotes the dot-product operation of matrix; $F_{SM}(\cdot)$ denotes softmax operation; $F_{GP}(\cdot)$ denotes global pooling. The output is $Z^{sp}=A^{sp}(X)\odot^{sp}X$, where $A^{sp}(X)\in R^{1\times H\times W}$, $Z^{sp}\in R^{C\times H\times W}$, and $\odot^{ch}$ denotes spatial-wise multiplication operation.

We utilized PSA modules in parallel form, which are calculated as follows:

$$PSA_P(X)=Z^{ch}+Z^{sp}=A^{ch}(X)\odot^{ch}X+A^{sp}(X)\odot^{sp}X \quad (5)$$

As shown in Figure 3, this study designs the feature extraction module C2f-PSA by integrating the PSA and C2f modules. In the bottleneck of C2f, the PSA module is inserted following the second 3×3 convolution. The C2f-PSA architecture is employed to augment the model's comprehension of the image's context and intricate spatial interactions. As a result, the model is more proficient in detecting objects floating on the water surface in intricate natural environments.

Figure 3
Diagram of C2f-PSA structure.



3.3. Additional Detection Layer for Small Objects

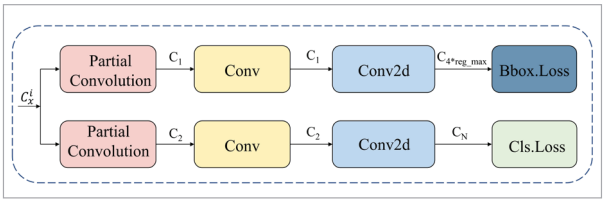
Considering the characteristics of floating objects observed in this study, we noted that small objects with variable morphology account for more than half of the USV-WSFO and FloW-Img datasets. We also found that the primary feature of small objects is predominantly distributed in the shallow feature map. However, the original YOLOv8s model utilizes a large downsampling factor, which compromises the retention of feature information for small objects after downsampling three times. The baseline model's P3, P4, and P5 detection layers provide feature maps of different sizes (80 × 80, 40 × 40, and 20 × 20, respectively) to identify targets of varying sizes. However, this design uses a minimum detection head size of 80 × 80 for each grid in the image, resulting in a receptive field of only 8 × 8. With images sized at 640 × 640, several little objects measuring less than 8 × 8 may result in erroneous or overlooked detections.

As shown in Figure 4, this issue was resolved by integrating a small object detection layer into YOLOv8s. We augmented the network by concatenating the 80×80 scaled feature map from the second layer of the backbone network with the 80×80 up-sampled feature map from the neck layer, resulting in a 160×160 feature map with improved resolution. The 160×160 feature map was subsequently combined with the corresponding feature map produced by the initial C2f module in the backbone architecture. Finally, the integrated feature maps improved feature information extraction from small objects, and they were fed into the brain structure for detection and classification tasks.

3.4. Reconstructed Detection Head

Due to the constraints of computing and storage resources on USVs, floating object detection methods must reconcile model complexity with real-time performance. The incorporation of a small object detection head significantly raises the model's computational cost. To resolve this matter, we introduce a lightweight Partial Convolution (PConv)[3] technique to design the PConv-head, which reduces unnecessary computational and storage demands while accelerating inference speed, without sacrificing detection accuracy. The PConv-Head structure is illustrated in Figure 5.

Figure 5
Diagram of PConv-Head structure.



The diagram illustrating conventional convolution and partial convolution is presented in Figure 6. A quarter of the input channels undergo convolution in PConv, while the other channels are left unchanged.

Figure 4
Small object detection head.

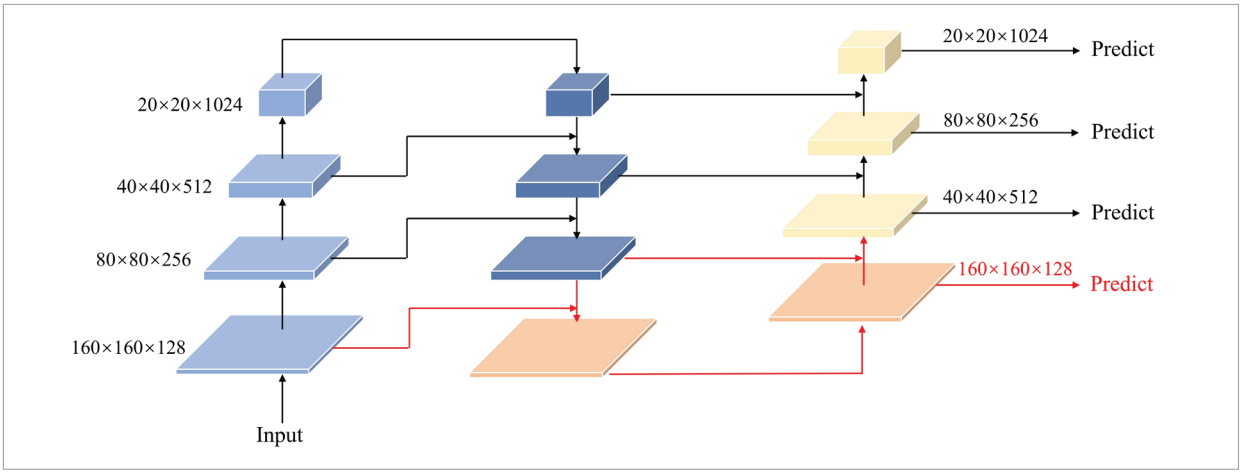
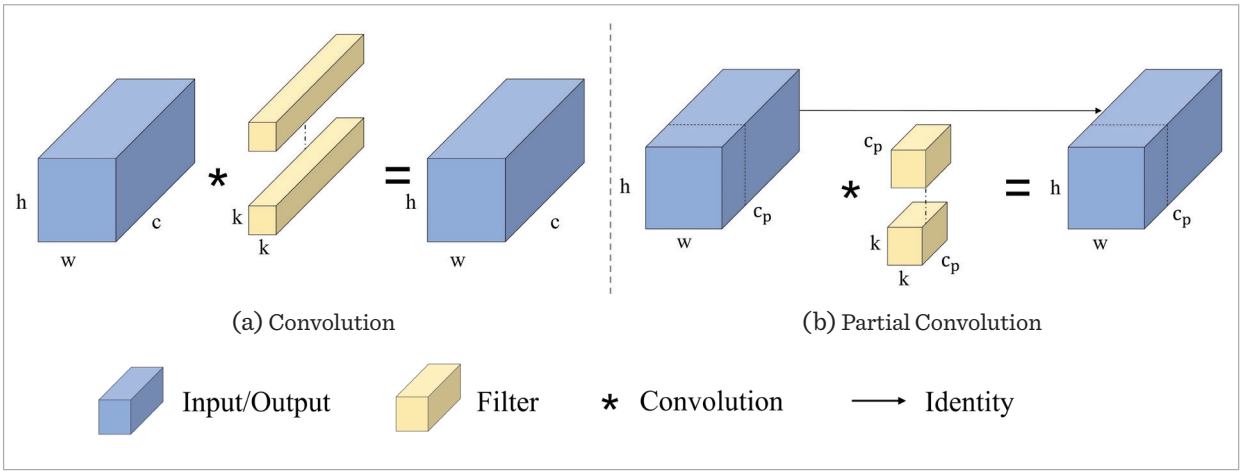


Figure 6
(a) Convolution; (b) Partial Convolution



Subsequently, the output is obtained by concatenating the processed 1/4 channels with the unchanged channels. This design allows for better utilization of correlations and redundancies between features.

The floating point operations per second (FLOPS) of conventional convolution is:

$$FLOPS_{Conv}=h\times w\times k^2\times c^2. \quad (6)$$

The FLOPS of PConv is:

$$FLOPS_{PConv}=h\times w\times k^2\times c_p{}^2. \quad (7)$$

When the partial ratio r=cp/c=1/4 , the FLOPS ratio of PConv to conventional convolution is:

$$\frac{FLOPS_{PConv}}{FLOPS_{Conv}}=\frac{c_p{}^2}{c^2}=\frac{1}{16}, \quad (8)$$

where h and w signify the height and width of the feature map, respectively; c indicates the number of channels in the standard convolution feature map, while c_p represents the number of channels in the partial convolution operation. As shown in Equation (8), PConv's computational cost is just 1/16 of regular convolution, greatly reducing the computational burden. Therefore, the incorporation of the PConv structure leads to a substantial reduction in both computational and parametric quantities in the improved model.

YOLOv8s adopts a decoupling head design to partition the feature channels into bounding box coordinate regression and object classification. In the head, a 3×3 convolution is employed to convert the feature layer's channel count C_x^i from the neck into C1 for localization and C2 for classification. Subsequently, the localization task's output channels are set to 4×reg_max, where each feature point represents the bounding box centroid (x, y) along with its width (w) and height (h) for parameter adjustment. For the classification task, the output channel count is denoted by C_N, representing the categories of floating objects detected. By replacing the first 3×3 convolutional layer of the original detection head with a PConv structure, the PConv-Head effectively reduces computational load and parameter count, while preserving image details. This modification enhances both the model's functionality and performance. We applied the PConv-Head to reconstruct all four detection heads.

3.5. Improvement of the Loss Function

The bounding box regression loss in YOLOv8 is calculated by combining DFL and CIoU losses. The DFL loss measures the probabilistic difference between predicted and ground-truth boxes using cross-entropy. On the other hand, CIoU loss quantifies the discrepancy between the true and predicted frames to improve the overall frame prediction. The formula for CIoU [42] is shown below:

$$L_{CIoU}=1-IoU+\frac{\rho^2(b+b^{gt})}{c^2}+\alpha v \quad (9)$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w}{h}\right) \quad (10)$$

$$\alpha=\frac{v}{(1-IoU)+v}, \quad (11)$$

where IoU quantifies the intersection between the true and predicted bounding boxes; b and b^{gt} are their centroids; ρ represents the Euclidean distance between them; c denotes the diagonal distance of the smallest enclosing region containing both boxes; w^{gt} and h^{gt} are the ground truth box's width and height, while *w* and *h* are those of the predicted box; *α* and *v* act as trade-off parameters, assessing aspect ratio consistency.

Complete Intersection over Union (CIoU) incorporates dimensional factors of predicted and ground truth bounding boxes, including overlap area, center-of-mass distance, and aspect ratio. However, training data often contain low-quality samples, and geometric variables like aspect ratio and center-of-mass distance can further amplify their negative effects. Consequently, CIoU inadequately balances complicated and simple samples, hence diminishing the model's detection capability. Rather than using CIoU, this study adopts Wise-IoUv3(WIoUv3) [32] to mitigate this issue. WIoUv3, built on a dynamic non-monotonic focusing mechanism, uses outliers *β* instead of IoU and dynamically allocates gradient gains. This can improve floating object detection by making high-quality anchor frames more competitive and reducing the negative effects of gradients induced by low-quality samples. The WIoUv3 expression is as follows:

$$\beta=\frac{L_{IoU}^{*}}{\overline{L_{IoU}}}\in[0,\ +\infty),\ L_{IoU}=1-IoU \quad (12)$$

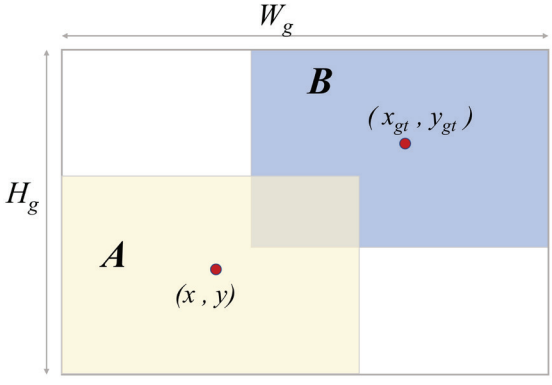
$$L_{WIoUv3}=rR_{WIoU}L_{IoU},\ r=\frac{\beta}{\sigma\alpha^{\beta-\sigma}} \quad (13)$$

$$R_{WIoU}=\exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^{*}}\right), \quad (14)$$

where * indicates the separation of W_g and H_g from the computational map; $\overline{L_{IoU}}$ is the average value of L_{IoU}; β quantifies outliers, where smaller values indicate higher anchor frame quality, leading to greater gradient gain; r is the non-monotonic focusing coefficient, σ and α are hyperparameters tailored to different models and datasets.

Figure 7 shows a schematic of WIoUv3 parameters.

Figure 7
Sketch map of WIoUv3.



4. Experiments and Results

4.1. Datasets

4.1.1. Constructing the USV-WSFO Dataset

As far as we know, publicly available datasets of floating objects in complex natural environments are limited. For this purpose, we built a small unmanned surface vehicle using a Raspberry Pi as the main control unit, equipped with a 4G module, a GPS module, a motor module, and an Intel RealSense series D435 camera to photograph floating objects and construct the dataset, as shown in Figure 8, which

Figure 8
(a) Exterior view of the unmanned surface vehicle;
(b) Internal details of the unmanned surface vehicle

(a) (b)

depicts the vehicle. The data collection occurred at various times and under diverse lighting conditions to mitigate the impact of variables such as illumination on detection accuracy. We collected 1200 images, all labeled using labelImg, after data cleaning. The annotations were then stored in YOLO format.

The dataset collected from the river section of Fuzhou City, Fujian Province, China, is designated as the Unmanned Surface Vehicle Water Surface Floating Objects Dataset (USV-WSFO Dataset). It includes eight categories of water surface floating objects: Bottle, Plastic bag, Foam board, Leaf, Branch, Boat, Others (unclassifiable monolithic garbage), and Mixed garbage. A sample dataset is shown in Figure 9.

Data augmentation improves the detection model's ability to generalize by increasing the amount and variety of training data. In our study, the USV-WSFO dataset underwent random contrast adjustment, image rotation, horizontal flipping, random gamma transformation, and Gaussian noise addition. The resolution of the augmented images was consistently established at 1280×720. Figure 10 displays examples of images prior to and subsequent to augmentation. Following the augmentation process, the dataset expanded to 10,800 images. The images in the dataset were categorized into three sections: training, validation, and testing, with the breakdown shown in Table 1. The training set contained 6,480 images, the validation set 2,160 images, and the testing set 2,160 images. The images were randomly assigned to each group to ensure a diverse selection in each subset.

Figure10
Samples of the data augmentations.

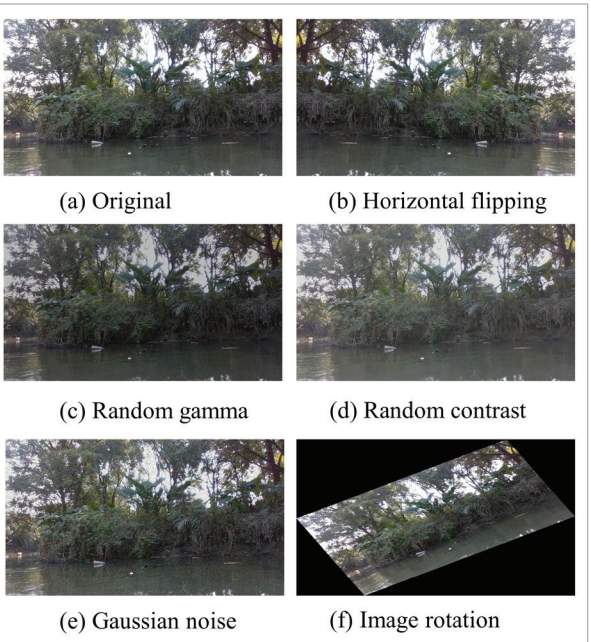
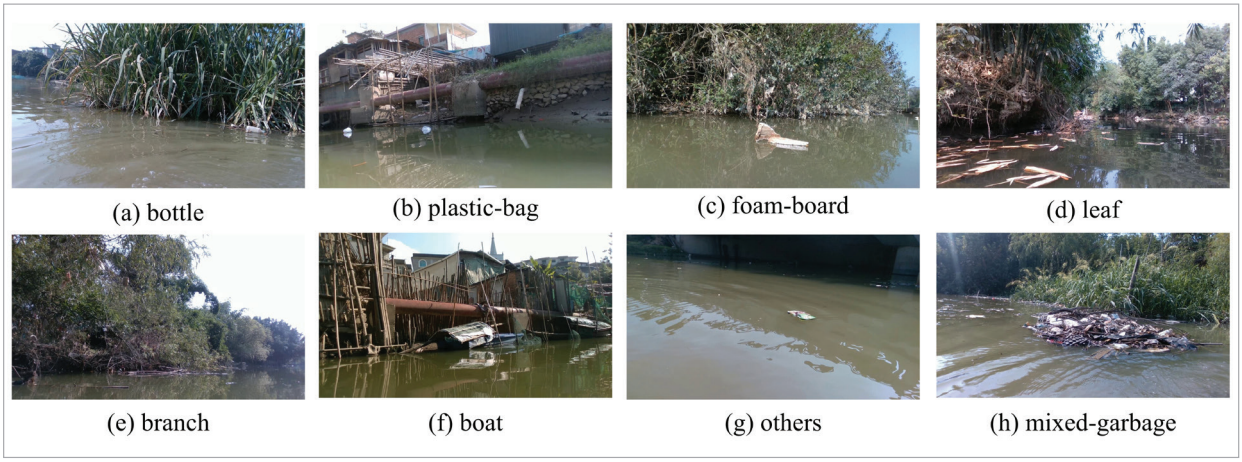


Table 1
Details of the USV-WSFO dataset.

Class	Number of Images	Number of Instances
Train	6480	28146
Val	2160	9705
Test	2160	9372

Figure 9
Sample of USV-WSFO Dataset.



4.1.2. FloW-Img Dataset

We validated our model using the FloW-Img dataset to enhance its generalizability. This dataset, a subset of the FloW dataset developed by ORCA, consists of 2000 images exclusively featuring the 'bottle' category. The FloW dataset itself is the first open-source collection for detecting floating objects from USVs in real-world inland waterway conditions. The dataset was subdivided into three sections for this study: a training set with 1200 images, a validation set with 200 images, and a test set with 400 images.

4.1.3. Statistical Analysis of USV- WSFO and FloW-Img

According to the COCO evaluation criteria[20], objects are categorized into small objects (area < 32^2), medium objects (32^2 < area < 96^2), and large objects (area > 96^2). We counted the number of targets of different sizes in the USV-WSFO dataset and the FloW-Img dataset, as shown in Figure 11. The findings demonstrate that, in both datasets, more than fifty percent of the targets are small objects. Additionally, the distribution pattern of target sizes is similar across the training, validation, and test sets. These findings contribute to understanding the characteristics of water surface objects and aid in the development of more effective detection algorithms.

4.2. Experimental Details

4.2.1. Experimental Settings

The experiment used an NVIDIA GeForce RTX 3070 GPU, along with Python 3.8 and PyTorch 1.7.1 frameworks. The YOLOv8s model was trained using pretrained weights from the VOC dataset. A detailed experimental setup is provided in Table 2. Stochastic gradient descent (SGD) with momentum was used as the optimization algorithm for both the YOLOv8s model and its variants during training. The initial learning rate was set to 0.01, with momentum and weight decay coefficients of 0.937 and 0.0005, respectively. The number of iterations was set to 200. Due to the 8GB memory limitation of a single GPU, a batch size of 16 was chosen to ensure consistency with comparative studies.

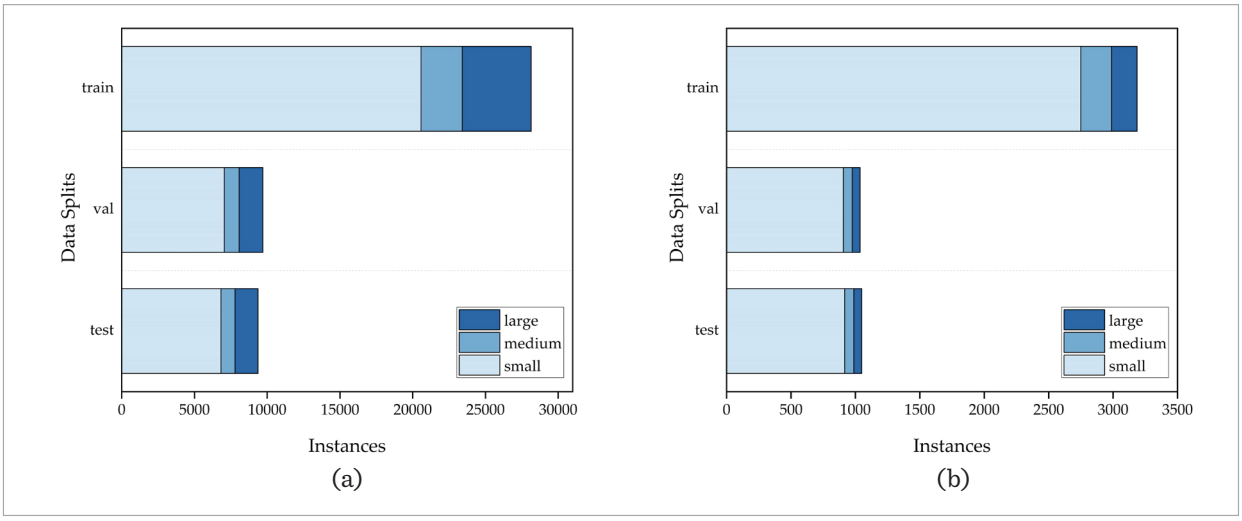
Table 2

Experimental environment configuration.

Item	Name
Operating system	Windows10
GPU	NVIDIA GeForce RTX 3070
RAM	8GB
Deep learning framework	PyTorch (1.7.1)
Interpreter	Python (3.8.18)
CUDA version	CUDA (11.0.1)
CUDNN version	CUDNN (8.0.5)

Figure 11

(a) Object box area statistics in the USV-WSFO dataset; (b) Object box area statistics in the FloW-Img dataset. 'Instances' represents the number of detected floating objects, while 'small', 'medium', and 'large' indicate different object size categories based on bounding box area.



4.2.2. Evaluation Indicators

We evaluated various object detection methods using a range of metrics. The primary statistic is AP (Average Precision), which is the average of AP values calculated at 10 IoU thresholds, ranging from 0.50 to 0.95 with a 0.05 interval. Additional metrics comprise AP_{50} (AP at IoU = 0.50), AP_{75} (AP at IoU = 0.75), average precision AP_S, AP_M, AP_L for targets of different scales, and FPS (Frames Per Second). The formulas for precision and recall are provided below.

$$P=\frac{TP}{TP+FP} \tag{15}$$

$$R=\frac{TP}{TP+FN} \tag{16}$$

In the formula, TP refers to the correctly detected positive samples, FP represents false positives, and FN denotes false negatives. The Precision-Recall (P-R) curve is plotted using precision (P) and recall (R) values, and its integration gives the Average Precision (AP), reflecting the detection accuracy of a category in the dataset. The formula is as follows:

$$AP=\int_0^1 PRdR \tag{17}$$

The AP of each category is summed and averaged to obtain the mAP for multi-category object detection.

$$mAP=\frac{\sum_{k=1}^{n} AP_k}{n} \tag{18}$$

The FPS (Frames Per Second) of YOLOv8 can be determined using the subsequent formula.

$$FPS=\frac{1}{\text{inference time per frame}} \tag{19}$$

FPS is the inverse of the inference time per frame, indicating how many frames can be processed per second.

4.3. Ablation Study

Ablation experiments were performed on the USV-WSFO dataset to validate the effectiveness of each module. The results, using YOLOv8s as the baseline, are summarized in Table 3.

According to the analysis results in Table 3, firstly, on the USV-WSFO dataset, we observed that the original YOLOv8s achieved an AP of 65.8%, an AP_{50} of 91.7%, and an AP_S of 43.7%. Introducing the C2f-PSA module to improve YOLOv8s to YOLOv8s_C2f-PSA improved its AP by 1.6%, AP_{50} by 0.7%, and AP_S by 1.3%. Subsequently, the network structure was modified by incorporating a small object detection head into YOLOv8s, resulting in YOLOv8s_small, which exhibited an increase in AP by 4%, AP_{50} by 3.4%, and AP_S by 11.9%. But now, compared to the original, the model's computational overhead has increased by 32.6%. To address this issue, the PConv-Head was devised, which successfully mitigated the model's complexity. The resulting model, YOLOv8s_small_PCHead, effectively balances speed and accuracy, maintaining an FPS of 64.3 and achieving 3.3% im-

Table 3
Effects of each module on the USV-WSFO dataset.

Methods	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)	FPS (f/s)	Params (M)	Model Size (MB)	FLOPS (G)
YOLOv8s	65.8	91.7	72.3	43.7	74.7	88.7	64.7	11.1	21.4	28.5
YOLOv8s_C2f-PSA	67.4	92.4	74.4	45.0	76.2	88.9	47.8	11.3	22.8	28.6
YOLOv8s_small	69.8	95.1	79.5	55.6	76.0	85.0	57.9	10.9	21.0	37.8
YOLOv8s_PCHead	62.9	89.8	70.9	40.8	72.1	85.0	68.1	9.6	18.5	21.5
YOLOv8s_WIoU	66.7	91.9	74.5	44.3	75.9	88.9	66.3	11.1	21.4	28.5
YOLOv8s_small_PCHead	68.9	95.0	78.4	53.8	74.8	83.1	64.3	10.1	19.5	27.2
YOLOv8s_small_PCHead_WIoU	68.7	95.3	79.5	56.0	74.3	84.1	64.1	10.1	19.5	27.2
PSP-YOLOv8s (Ours)	**70.1**	**95.5**	**80.4**	**56.6**	75.6	84.2	61.5	10.2	20.8	27.3

provement in AP_{50} by 3.3% over the original model. Finally, replacing CIoU with WIoUv3 led to the development of YOLOv8s_WIoU, which increased AP, AP_{50}, and AP_S by 0.9%, 0.2%, and 0.6%, respectively.

In our experiments, the PSP-YOLOv8s model outperformed all other models, achieving the highest scores in AP, AP_{50}, AP_{75}, and AP_S metrics, with results of 70.1%, 95.5%, 80.4%, and 56.6%, respectively. Compared to YOLOv8s, the improvements are notable: AP increased by 4.3%, AP_{50} by 3.8%, AP_{75} by 8.1%, and AP_S by 12.9%. Furthermore, the parameters, model size, and computing overhead diminished by 8.1%, 2.8%, and 4.2%, respectively, compared to the original model. Achieving an optimal balance between precision and real-time efficiency, the improved YOLOv8s model effectively tackles the challenges of complex backgrounds and small objects within USV imagery.

4.4. Comparisons with State-of-the-Arts

4.4.1. Results on USV-WSFO

To assess the suitability of the PSP-YOLOv8 model for detecting floating objects, we quantitatively compared its performance against that of the original YOLOv8s model and several popular algorithms, including Faster R-CNN, RetinaNet, CenterNet, SSD-ResNet50, SSD-MobileNetV2, YOLOv7, and YOLOX, using the USV-WSFO dataset.

The results, presented in Table 4, show that the enhanced YOLOv8s model significantly improves in AP, AP_{50}, AP_{75}, AP_S, and AP_M over the baseline YOLOv8s model and other competing algorithms such as Faster R-CNN and RetinaNet. The improved YOLOv8s algorithm model surpasses other algorithms with an AP of 70.1% and an AP_{50} of 95.5%. Particularly noteworthy is its remarkable improvement in small object detection, with AP_S values increasing by 40.7%, 34.6%, 32.7%, 49.5%, 55.9%, 22.1%, 30.6%, and 12.9%, compared to Faster R-CNN, RetinaNet, CenterNet, SSD-ResNet50, SSD-MobileNetV2, YOLOv7, YOLOX, and YOLOv8s, respectively. Although the model exhibits slightly reduced speed compared to the original YOLOv8s, which may be attributed to alterations in the network structure, its overall performance surpasses that of the compared algorithms.

Figure 12 displays visualization results of the USV-WSFO dataset obtained using both the YOLOv8s and PSP-YOLOv8s models. Our approach improves object detection accuracy and resilience by successfully reducing instances of misdetection and omission.

Table 5 presents a comparison of the PSP-YOLOv8s model, the original YOLOv8s, and mainstream algorithms, including Faster R-CNN, RetinaNet, CenterNet, SSD-ResNet50, SSD-MobileNetV2, YOLOv7, and YOLOX, on the USV-WSFO dataset in terms of AP_{50} for different object categories. Our method consistently outperforms other comparable approaches across all categories. Notably, the most significant accuracy improvements are observed for the 'bottle,' 'plastic bag,' and 'leaf' categories, which are characterized by an abundance of small objects. This further validates the applicability of our model for detecting small objects.

Table 4

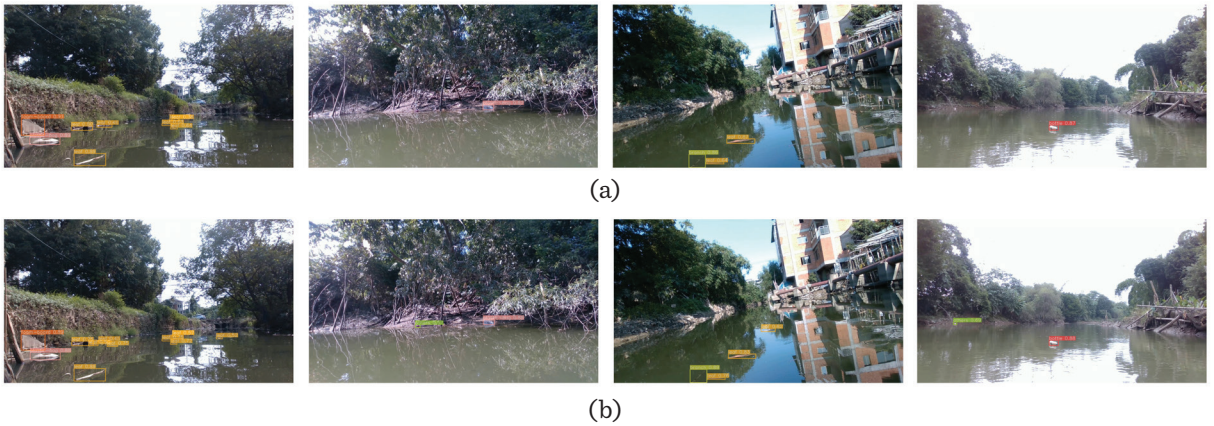
Comparisons with other methods on the USV-WSFO dataset.

Methods	Backbone	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)	FPS (f/s)
Faster R-CNN	Vgg-16	31.4	64.0	25.5	15.9	38.5	50.9	14.3
RetinaNet	ResNet-50	39.6	67.2	39.5	22.0	49.4	58.6	15.0
CenterNet	ResNet-50	45.6	84.9	42.0	23.9	51.2	70.4	40.5
SSD	ResNet-50	17.9	36.9	14.9	7.1	21.7	42.9	61.0
SSD	MobileNet-V2	7.6	21.3	4.0	0.7	5.8	31.2	79.2
YOLOv7	CSPDarknet-53	54.4	89.2	57.8	34.5	58.2	77.0	57.5
YOLOX	CSPDarknet-53	44.2	85.9	39.7	26.0	49.5	68.0	48.7
YOLOv8s	CSPDarknet-53	65.8	91.7	72.3	43.7	74.7	88.7	64.7
PSP-YOLOv8s(Ours)	CSPDarknet-53	**70.1**	**95.5**	**80.4**	**56.6**	**75.6**	84.2	61.5

Table 5
Comparison of different categories in the USV-WSFO Dataset with other methods.

Methods	AP_{50}(%)								All (%)
	bottle	plastic-bag	foam-board	leaf	branch	boat	others	mixed-garbage	
Faster R-CNN	56.4	62.6	74.9	67.1	55.7	76.4	55.9	62.6	64.0
RetinaNet	52.5	70.8	76.2	69.8	59.0	83.4	62.3	63.6	67.2
CenterNet	80.9	76.1	87.6	85.5	83.3	91.4	81.6	92.7	84.9
SSD-ResNet50	14.9	40.1	46.4	42.0	28.6	68.4	26.4	28.8	36.9
SSD-MobileNetV2	4.7	15.5	23.4	23.6	14.3	49.7	6.6	32.4	21.3
YOLOv7	87.3	85.8	93.1	89.5	84.6	93.0	84.7	95.3	89.2
YOLOX	78.5	81.3	91.4	87.2	80.9	92.6	80.7	94.2	85.9
YOLOv8s	87.9	84.2	95.7	92.6	92.2	97.3	86.3	97.3	91.7
PSP-YOLOv8s (Ours)	**94.3**	**93.2**	**96.5**	**96.6**	**94.5**	**97.4**	**94.2**	**97.5**	**95.5**

Figure 12
Comparison of detection outcomes between YOLOv8s and PSP-YOLOv8s. The detection outcomes of YOLOv8s are presented at the top, while the results of our method are displayed at the bottom.



(a)

(b)

4.4.2. Results on FloW-Img

We compared the enhanced YOLOv8s method with other popular approaches, including Faster R-CNN, RetinaNet, CenterNet, SSD-ResNet50, SSD-MobileNetV2, YOLOv7, YOLOX, and YOLOv8s, using the FloW-Img dataset. This comparison further validated the robustness of our method, with the numerical results of the metrics presented in Table 6. Despite the FloW dataset containing only one class of objects, namely, the 'bottle' class, our model outperforms others in the AP, AP_{50}, AP_{75}, and AP_S metrics, with AP_{50} reaching 88.8%.

Figure 13 displays some visualization results for the FloW dataset. Our method effectively distinguishes each bottle and generates high-quality bounding boxes even for small objects located at a distance, as compared to YOLOv8s.

4.5. Experimental Validation of Floating Object Detection

The floating object detection experiment was conducted in a pond on the Fuzhou University campus, where bottles were placed randomly as test targets. Utilizing the bidirectional communication capabilities of Web-

Table 6
Comparisons with other methods on the FloW-Img dataset.

Methods	Backbone	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)	FPS (f/s)
Faster R-CNN	Vgg-16	40.4	82.9	33.5	27.5	53.3	62.7	14.9
RetinaNet	ResNet-50	45.3	87.0	40.8	30.7	57.8	75.0	15.5
CenterNet	ResNet-50	37.7	80.7	30.5	28.7	54.6	90.0	40.9
SSD	ResNet-50	8.4	23.1	3.9	3.8	14.4	20.0	50.5
SSD	MobileNet-V2	15.2	43.9	7.9	5.0	29.9	44.7	81.7
YOLOv7	CSPDarknet-53	46.7	86.4	45.3	34.3	57.1	76.4	79.4
YOLOX	CSPDarknet-53	36.6	77.8	28.7	26.9	55.1	90.0	50.0
YOLOv8s	CSPDarknet-53	47.6	86.9	46.6	34.0	59.8	75.1	74.2
PSP-YOLOv8s (Ours)	CSPDarknet-53	**48.0**	**88.8**	**46.9**	**34.8**	59.4	74.5	72.6

Figure 13
Comparison of detection outcomes between YOLOv8s and PSP-YOLOv8s. The detection outcomes of YOLOv8s are presented at the top, while the results of our method are displayed at the bottom.

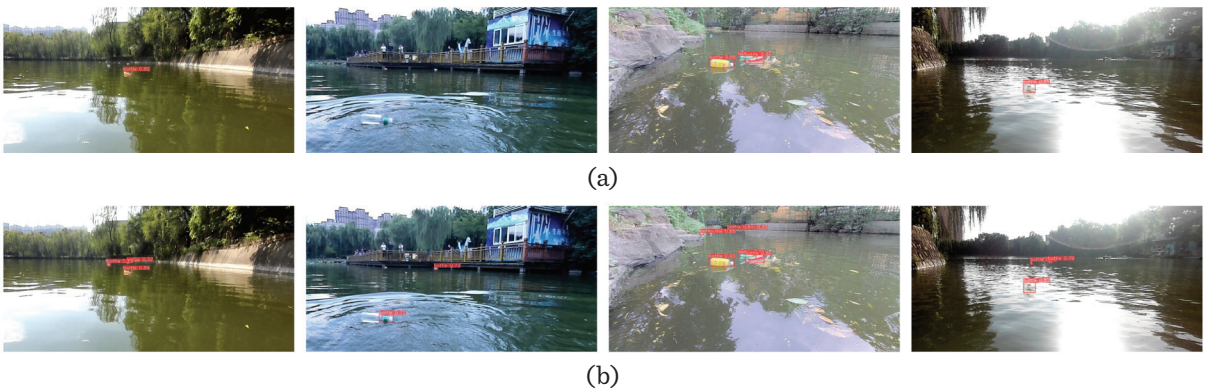


Figure 14
Field measurement results of floating object detection.

Socket, we enabled data exchange between the USV, cloud server, and user side. The PSP-YOLOv8s model was encapsulated in ONNX format and deployed to the user side, where images are received in real time via the Flask framework, and the model is invoked for recognition. Experimental results show that our model achieves 87.06% accuracy in identifying floating objects, as illustrated in Figure 14.

5. Discussion

We illustrate and examine the outcomes of the method we propose for detecting floating objects utilizing Gradient-weighted Class Activation Mapping(Grad-CAM) [28] , a technique designed to enhance the interpretability of deep learning models. It uses the gradient information of a class to flow to the last convolutional layer, producing a category-specific map that reveals the model's region of interest on the image.

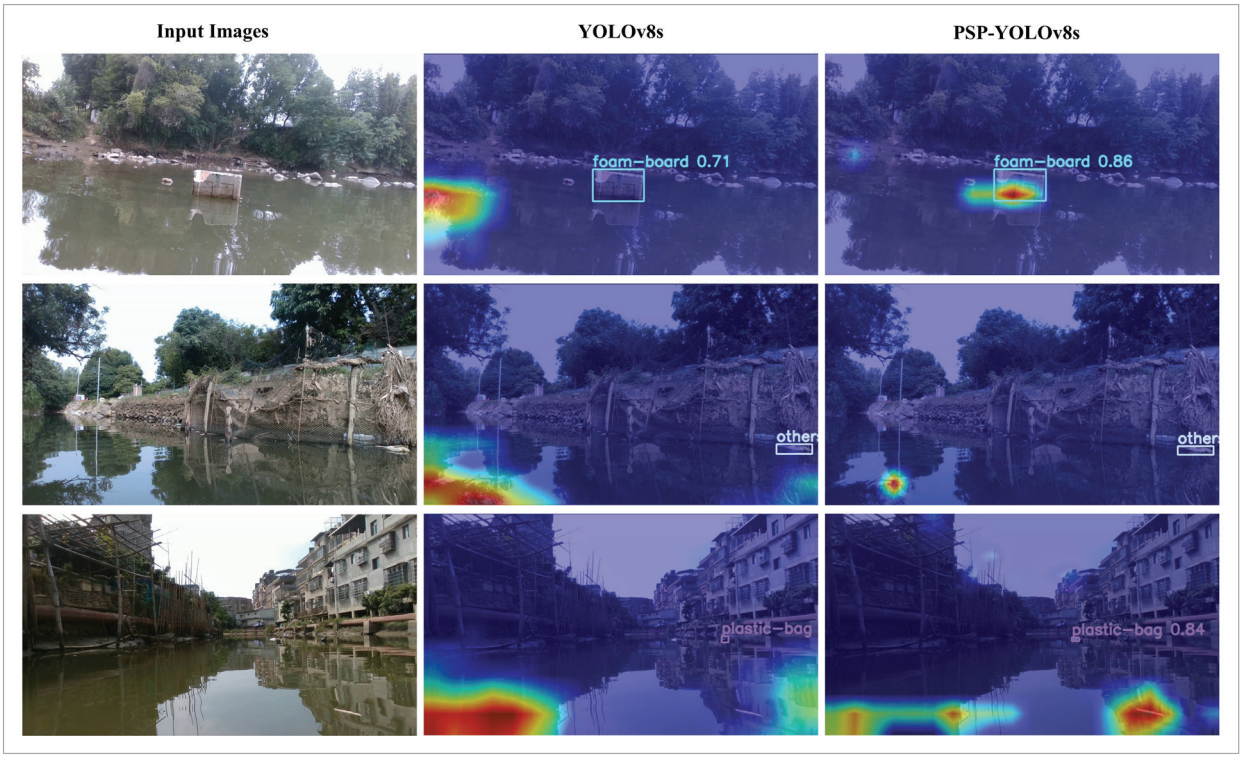
Figure 15 presents several examples of Grad-CAM visualizations comparing the PSP-YOLOv8s and YOLOv8s models. It is evident that our model accurately identifies floating objects on the water surface and generates precise bounding boxes around them. Nonetheless, the Grad-CAM visualizations from the baseline model exhibit diminished accuracy, frequently extending to regions beyond the water surface objects and being vulnerable to solar reflections and water waves. Conversely, the Grad-CAM visualization of our method is more focused and accurately highlights the key features of floating objects.

However, there are still some shortcomings in this work. For example, the impact of seasonal variations on surface floating object detection and the applicability of unmanned surface vehicle operations at night have not been considered. Future work could focus on achieving more accurate localization of water surface floating objects through multi-sensor fusion, incorporating technologies such as millimeter-wave radar and infrared cameras, which could further enhance detection performance.

6. Conclusions

In this paper, we design a lightweight model for detecting floating objects on USVs, PSP-YOLOv8s, which successfully solves the challenges of background complexity and small target identification in surface floating object detection. The method improves YOLOv8 by integrating the PSA attention mechanism into the C2f module preceding the SPPF layer within the backbone, forming the C2f-

Figure 15
Grad-CAM comparison of PSP-YOLOv8s and YOLOv8s.



PSA module. This enhancement boosts feature extraction efficiency in challenging environments. We also added a new small object detection head, which significantly reduces the small object leakage rate. Meanwhile, the four detection heads are redesigned using the lightweight PConv-Head, which effectively reduces the computational overhead and storage cost of the model. Finally, we adopt WIoUv3 instead of CIoU to focus on the anchor frame with general quality. The ablation experiments show that all four improvement strategies are significantly effective, with AP and AP_{50} improved by 4.3% and 3.8%, respectively, and the improvement in small target detection is particularly significant, with AP_S improved by 12.9%. Furthermore, the parameters, model size, and computing overhead were diminished by 8.1%, 2.8%, and 4.2%, respectively. Comparative experiments show that PSP-YOLOv8s outperforms eight mainstream algorithms on both the USV-WSFO and FLoW-Img datasets, achieving an AP of 70.1% and AP_{50} of 95.5% on the USV-WSFO dataset, and an AP of 48% and AP_{50} of 88.8% on the FLoW-Img dataset. Our method demonstrates its effectiveness in detecting floating objects from USVs, with potential for further optimization and application in more complex scenarios.

Acknowledgement

This research received no external funding.

Availability of data and materials

The data that support the findings of this study are openly available at https://github.com/hongh07/USV-WSFO-Dataset and https://orca-tech.cn/en/datasets/FloW/FloW-Img.

References

1. Akib, A., Tasnim, F., Biswas, D., Hashem, M. B., Rahman, K., Bhattacharjee, A., Fattah, S. A. Unmanned Floating Waste Collecting Robot. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON), 2019, 2645-2650. https://doi.org/10.1109/TENCON.2019.8929537
2. Cai, Q., Wang, Q., Zhang, Y., He, Z., Zhang, Y. LWD-Net-A Lightweight Water-Obstacles Detection Network for Unmanned Surface Vehicles. Robotics and Autonomous Systems, 2023, 166,104453. https://doi.org/10.1016/j.robot.2023.104453
3. Chen, J., Kao, S.-h., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H. G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 12021-12031. https://doi.org/10.1109/CVPR52729.2023.01157
4. Chen, R., Wu, J., Peng, Y., Li, Z., Shang, H. Solving Floating Pollution with Deep Learning: A Novel SSD for Floating Objects Based on Continual Unsupervised Domain Adaptation. Engineering Applications of Artificial Intelligence, 2023, 120,105857. https://doi.org/10.1016/j.engappai.2023.105857
5. Cheng, Y., Xu, H., Liu, Y. Robust Small Object Detection on the Water Surface through Fusion of Camera and Millimeter Wave Radar. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 17, 15263-15272. https://doi.org/10.1109/ICCV48922.2021.01498
6. Cheng, Y., Zhu, J., Jiang, M., Fu, J., Pang, C., Wang, P., Sankaran, K., Onabola, O., Liu, Y., Liu, D. FloW: A Dataset and Benchmark for Floating Waste Detection in Inland Waters. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021, 10953-10962. https://doi.org/10.1109/ICCV48922.2021.01077
7. Dalal, N., Triggs, B. Histograms of Oriented Gradients for Human Detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 1, 886-893. https://doi.org/10.1109/CVPR.2005.177
8. Felzenszwalb, P., McAllester, D., Ramanan, D. A Discriminatively Trained, Multiscale, Deformable Part Model. In 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, 1-8. https://doi.org/10.1109/CVPR.2008.4587597
9. Gao, K., Gao, M., Zhou, M., Ma, Z. Artificial Intelligence Algorithms in Unmanned Surface Vessel Task Assignment and Path Planning: A Survey. Swarm and Evolutionary Computation, 2024, 86,101505. https://doi.org/10.1016/j.swevo.2024.101505
10. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision. 2015,1440-1448. https://doi.org/10.1109/ICCV.2015.169
11. Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich

Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

12. Hahladakis, J. N. A Meta-Research Analysis on the Biological Impact of Plastic Litter in the Marine Biota. Science of The Total Environment, 2024, 928, 172504. https://doi.org/10.1016/j.scitotenv.2024.172504
13. He, J., Cheng, Y., Wang, W., Gu, Y., Wang, Y., Zhang, W., Shankar, A., Shitharth, S., Kumar, S. A. EC-YOLOX: A Deep Learning Algorithm for Floating Objects Detection in Ground Images of Complex Water Environments. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024, 7359-7370. https://doi.org/10.1109/JSTARS.2024.3367713
14. He, K., Gkioxari, G., Dollár, P.and Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, 2017, 2961-2969. https://doi.org/10.1109/ICCV.2017.322
15. Huang, M., Mi, W., Wang, Y. EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model. Drones, 2024, 8(7), 337. https://doi.org/10.3390/drones8070337
16. Jiang, X., Yang, Z., Huang, J., Jin, G., Yu, G., Zhang, X., Qin, Z. YOLOv5n++: An Edge-Based Improved YOLOv5n Model to Detect River Floating Debris. Journal of Intelligent & Fuzzy Systems,46(1),2507-2520. https://doi.org/10.3233/JIFS-234222
17. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R. A Survey of Deep Learning-Based Object Detection. IEEE Access, 2019, 7, 128837-128868. https://doi.org/10.1109/ACCESS.2019.2939201
18. Law, H., Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision (ECCV). 2018, 734-750. https://doi.org/10.1007/978-3-030-01264-9_45
19. Li, Q., Wang, Z., Li, G., Zhou, C., Chen, P., Yang, C. An Accurate and Adaptable Deep Learning-Based Solution to Floating Litter Cleaning Up and Its Effectiveness on Environmental Recovery. Journal of Cleaner Production, 2023, 388,135816. https://doi.org/10.1016/j.jclepro.2022.135816
20. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L. Microsoft COCO: Common Objects in Context. Springer International Publishing, 2014, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48
21. Liu, H., Liu, F., Fan, X., Huang, D. Polarized Self-Attention: Towards High-Quality Pixel-Wise Regression. arXiv preprint arXiv:2107.00782, 2021. https://doi.org/https://arxiv.org/abs/2107.00782
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C. SSD: Single Shot Multibox Detector. In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
23. Nava, V., Chandra, S., Aherne, J., Alfonso, M. B., Antão-Geraldes, A. M., Attermeyer, K., Bao, R., Bartrons, M., Berger, S. A., Biernaczyk, M. Plastic Debris in Lakes and Reservoirs. Nature, 2023, 619(7969), 317-322. https://doi.org/10.1038/s41586-023-06168-4
24. Qiu, S., Cai, B., Wang, W., Wang, J., Zaheer, Q., Liu, X., Hu, W., Peng, J. Automated Detection of Railway Defective Fasteners Based on YOLOv8-FAM and Synthetic Data Using Style Transfer. Automation in Construction, 2024, 162, 105363. https://doi.org/10.1016/j.autcon.2024.105363
25. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031
26. Renfei, C., Jian, W., Yong, P., Zhongwen, L., Hua, S. Detection and Tracking of Floating Objects Based on Spatial-Temporal Information Fusion. Expert Systems with Applications, 2023, 225, 120185. https://doi.org/10.1016/j.eswa.2023.120185
27. Ruangpayoongsak, N., Sumroengrit, J., Leanglum, M. A Floating Waste Scooper Robot on Water Surface. IEEE, 2017, 1543-1548. https://doi.org/10.23919/ICCAS.2017.8204234
28. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, 2017, 618-626. https://doi.org/10.1109/ICCV.2017.74
29. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. arXiv preprint arXiv:1312.6229, 2013. https://arxiv.org/abs/1312.6229
30. Sravanthi, R., Sarma, A. S. V. Efficient Image-Based Object Detection for Floating Weed Collection with Low-Cost Unmanned Floating Vehicles. Soft Computing, 2021, 25(20), 13093-13101. https://doi.org/10.1007/s00500-021-06171-9

31. Terven, J., Cordova-Esparza, D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 2023, 5(4), 1680-1716. https://doi.org/10.3390/make5040083

32. Tong, Z., Chen, Y., Xu, Z.and Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv preprint arXiv:2301.10051, 2023. https://arxiv.org/abs/2301.10051

33. Van Lieshout, C., Van Oeveren, K., Van Emmerik, T., Postma, E. Automated River Plastic Monitoring Using Deep Learning and Cameras. Earth and Space Science, 2020, 7(8), e2019EA000960. https://doi.org/10.1029/2019EA000960

34. Viola, P., Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01). IEEE, 2001, I-I. https://doi.org/ 10.1109/CVPR.2001.990517

35. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G. Yolov10: Real-time End-to-end Object Detection. arXiv preprint arXiv:2405.14458, 2024. https://doi.org/10.48550/arXiv.2405.14458

36. Wang, B., Jiang, P., Liu, Z., Li, Y., Cao, J., Li, Y. An Adaptive Lightweight Small Object Detection Method for Incremental Few-Shot Scenarios of Unmanned Surface Vehicles. Engineering Applications of Artificial Intelligence, 2024, 133, 107989. https://doi.org/10.1016/j.engappai.2024.107989

37. Wang, C.-Y., Yeh, I. H., Liao, H.-Y. M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv preprint arXiv:2402.13616. https://doi.org/arXiv:2402.13616

38. Wang, N., Wang, Y., Wei, Y., Han, B., Feng, Y. Marine Vessel Detection Dataset and Benchmark for Unmanned Surface Vehicles. Applied Ocean Research, 2024, 142,103835. https://doi.org/10.1016/j.apor.2023.103835

39. Wei, W., Cheng, Y., He, J., Zhu, X. A Review of Small Object Detection Based on Deep Learning. Neural Computing and Applications, 2024, 36(12), 6283-6303. https://doi.org/10.1007/s00521-024-09422-6

40. Wu, X., Sahoo, D., Hoi, S. C. Recent Advances in Deep Learning for Object Detection. Neurocomputing, 2020, 396, 39-64. https://doi.org/10.1016/j.neucom.2020.01.085

41. Zhang, J., Jin, J., Ma, Y., Ren, P. Lightweight Object Detection Algorithm Based on YOLOv5 for Unmanned Surface Vehicles. Frontiers in Marine Science, 2023, 9, 1058401. https://doi.org/10.3389/fmars.2022.1058401

42. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R,. Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07), 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

43. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J. Object Detection in 20 Years: A Survey. Proceedings of the IEEE, 2023,111(3),257-276. https://doi.org/10.1109/JPROC.2023.3238524