Earth Science Informatics (2025) 18:135
https://doi.org/10.1007/s12145-024-01679-8

RESEARCH

PAR-YOLO: a precise and real-time YOLO water surface garbage detection model

Ning Li[1,2] · Mingliang Wang[1] · He Huang[1] · Bo Li[1,3] · Baohua Yuan[1] · Shoukun Xu[1]

Received: 13 September 2024 / Accepted: 25 December 2024 / Published online: 2 January 2025

Abstract

In the scenario of water surface garbage detection, the model must accurately detect different types of objects and be able to respond continuously within a short time frame, enabling timely retrieval by the surface cleaning robot. Therefore, this paper proposes a surface garbage detection model named Precise and Real-time YOLO (PAR-YOLO), with a focus on real-time performance and detection accuracy. Firstly, to reduce model computation and improve detection efficiency, the Ghost Bottleneck module is designed and utilized in the backbone section as a replacement for the traditional Bottleneck module. Secondly, in order to effectively reduce the interference of factors such as water ripples, lighting variations, or reflections on object feature recognition, we have designed a Noise Suppression Module (NSM) and integrated it into the neck section. Lastly, to enhance the model's attention to challenging samples and improve detection accuracy, the Varifocal Loss function is employed in the head section. Experimental results demonstrate that the PAR-YOLO model achieves a Frames Per Second (FPS) of 238, with a mean average precision (mAP) of 85.53%, 47.3%, and 28.5% on our self-made water surface garbage dataset, the Flow public water surface garbage dataset and the Pascal VOC2007 dataset, respectively. Compared to other comparative models, our model achieves the best results.

Keywords Water surface garbage detection · Lightweight · PAR-YOLO · Noise suppression module

Introduction

The issue of water surface garbage pollution poses serious hazards to aquatic life, water environments, and the entire ecosystem. However, manual retrieval of water surface garbage faces many challenges. Therefore, utilizing object detection technology in conjunction with robotic boats for automated retrieval has become an important approach to addressing this problem. This technology can significantly improve retrieval efficiency, reduce the consumption of human resources, and maximize the cleanup of garbage in water bodies, thereby minimizing damage to the ecosystem.

In recent years, with the rapid development of deep learning, significant progress has been made in object detection algorithms. Based on the different processing approaches, convolutional neural network-based object detection algorithms can be categorized into two types: one-stage and two-stage. Common examples of one-stage algorithms include YOLOv3 (Redmon and Farhadi 2018), YOLOv4 (Bochkovskiy et al. 2020), YOLOX (Ge et al. 2021), SSD (Liu et al. 2016), etc., while popular two-stage algorithms include Faster R-CNN (Ren 2015), Cascade R-CNN (Cai and Vasconcelos 2018), Dynamic R-CNN (Zhang et al. 2020), etc. Additionally, with the rise of Transformer (Vaswani et al. 2017) models, detection algorithms based on Transformers have emerged, such as DETR (Carion et al. 2020), Deformable DETR (Zhu et al. 2020), DINO (Zhang et al. 2022), etc. Despite the remarkable advancements in object detection algorithms, challenges and issues still exist. One of the challenges is the lack of compatibility when dealing with specific application scenarios. For instance, in the context of water surface garbage detection, objects are often affected by

Communicated by: Hassan Babaie

✉ Shoukun Xu
xsk@cczu.edu.cn

[1] School of Computer Science and Artificial Intelligence, Aliyun School of Big Data, School of Software, Changzhou University, Changzhou 213164, China

[2] School of Computer Science and Software Engineering, HoHai University, Nanjing 210098, China

[3] Jiangsu Petrochemical Process Key Equipment Digital Twin Technology Engineering Research Center, Changzhou University, Changzhou 213164, China

background interference and have small scales. This can lead to lower detection accuracy in one-stage algorithms. Additionally, two-stage algorithms typically involve a two-step processing pipeline, resulting in relatively higher computational costs. Moreover, although the Transformer architecture excels at capturing long-range dependencies in images, its significant computational expense remains an issue that cannot be overlooked. Therefore, for specific application scenarios, it is crucial to select appropriate detection algorithms and make improvements to create a high-performance detection algorithm tailored for the specific task.

Water surface garbage detection has not yet formed a mainstream system, and only a few researchers are dedicated to studying in this field. Wen et al. (2021) proposed a method that utilizes deep convolutional neural networks for multi-frame detection of small objects on the water surface, demonstrating better detection performance compared to traditional multi-frame detection algorithms. However, this method has limited adaptability to object shapes and is highly sensitive to lighting and weather conditions. Zhang et al. (2021) proposed a real-time detection method for floating objects on the water surface based on an improved RefineDet (Zhang et al. 2018) model. They improved the anchor refinement module by adding convolutional layers to obtain higher-level semantics and fused high-level features with low-level features to enhance detection accuracy. However, this method has a slow detection speed. Therefore, we are conducting further research on water surface garbage detection, addressing the existing issues and building upon the work of previous researchers.

In traditional object detection algorithms, YOLOX adopts a lightweight network architecture that provides significant inference speed advantages while maintaining high detection accuracy. Therefore, we have chosen YOLOX as the baseline model for water surface garbage detection and made improvements to it. Firstly, to make the model more lightweight and ensure real-time performance for the water surface cleaning robot, we designed and utilized the Ghost Bottleneck module in the backbone section of YOLOX. This reduces the computational burden of the model while achieving better performance. Moreover, water ripples, lighting variations, and reflections are common sources of noise interference in water surface scenes. During the detection process, these factors often lead to the generation of false objects, resulting in model misidentification and decreased detection accuracy. To address this issue, we have designed a lightweight Noise Suppression Module (NSM) in the neck section of YOLOX. This allows the model to focus on the object regions in the image and suppress irrelevant noise features, effectively achieving noise suppression at a lower computational cost and improving detection accuracy. Lastly, in the head section of YOLOX, we incorporated Varifocal Loss (Zhang et al. 2021) function. Varifocal Loss function dynamically adjusts the weights of each sample based on their predicted confidences, allowing the model to pay more attention to difficult-to-detect high-quality positive samples. This enhances the model's learning capability for these samples and improves its detection performance. Ultimately, we have named the improved model Precise and Real-time YOLO (PAR-YOLO).

The contributions of this work can be summarized in the following aspects:

(1) In order to ensure real-time detection and make the model more lightweight, we designed the Ghost Bottleneck module. This module reduces the computational cost of the model while achieving better performance.
(2) To suppress the interference from factors such as water ripples, lighting variations, or reflections on object recognition, we have designed a lightweight Noise Suppression Module (NSM) in the Neck section of the network. This module effectively suppresses noise on the feature map and enhances the object features.
(3) To prioritize high-quality positive sample training, we have introduced the Varifocal Loss function in the head section of the network. Experimental results have demonstrated significant improvements in the model's performance and robustness.
(4) Compared to other comparative models, PAR-YOLO achieves the fastest and highest precision results on our self-made water surface garbage dataset, the Flow public water surface garbage dataset and the Pascal VOC2007 dataset.

Related work

Water surface garbage detection

Water surface garbage detection was not a mainstream research direction, with only a few people in the industry focusing on research in this field. However, it provided us with valuable references. Yang et al. (2022) proposed a new water surface garbage detection model called YOLOv5_CBS. This model improved the feature extraction capability for water surface objects and accelerated the convergence speed. Zhang et al. (2021) introduced a water surface garbage detection method based on the RefineDet model, which fused high-level and low-level features to enhance detection accuracy. Yi and Luo (2023) addressed the real-time and accuracy issues in water surface garbage detection scenes by proposing the GFL_HAM algorithm based on Generalized Focal Loss, which is better suited for water surface garbage detection applications. Pang and Qin (2021) presented a one-step water surface garbage detection model called GDT-Net, overcoming the limitations of low detection accuracy, slow speed, and susceptibility to interference in traditional water surface garbage detection algorithms. Cheng et al. (2021) created the first inland water surface garbage detection dataset, capturing images from the perspective of unmanned boats and manually annotating water surface garbage. Ma et al. (2023) proposed a water surface garbage detection algorithm that combines image semantic segmentation networks and object detection networks. This algorithm first uses the UNet (Ronneberger et al. 2015) network to segment floating garbage in complex backgrounds, then applies the dark channel prior algorithm for noise removal to reduce the interference from lighting and water waves, and finally detects it.

Lightweight network

Lightweight networks maintained relatively good performance with lower parameter count and computational complexity, providing effective solutions for resource-constrained environments. Howard et al. (2017) first introduced the MobileNet network, which introduced the concept of depthwise separable convolution, decomposing standard convolution into depthwise and pointwise convolutions to reduce computational cost while maintaining model performance. Sandler et al. (2018) proposed MobileNetV2, which introduced inverted residual with linear bottleneck, making the model even more lightweight while preserving performance. Howard et al. (2019) presented MobileNetV3 through hardware-aware neural architecture search combined with the NetAdapt algorithm. Compared to the MobileNetV2 model with the same latency, MobileNetV3 achieved greater improvement in accuracy. Zhang et al. (2018) proposed ShuffleNet, which utilized two new operations, pointwise group convolution and channel shuffle, to accelerate computation by employing group convolutions as much as possible while reducing computational cost without sacrificing accuracy. Ma et al. (2018) introduced ShuffleNetV2, which proposed four lightweight network design principles and a novel convolutional block architecture based on actual inference speed. Tan and Le (2019) conducted in-depth research on model scaling and introduced EfficientNet, which achieved better accuracy and efficiency by balancing network depth, width, and resolution. Han et al. (2020) proposed the GhostNet model, which replaced traditional convolution operations with linear transformations using fewer parameters to generate multiple feature maps revealing intrinsic feature information, thereby reducing overall parameter count and computational complexity.

Attention mechanism

The attention mechanism could adaptively make the model focus on important input information, reducing interference from irrelevant information, and found wide application in various domains and tasks. Hu et al. (2018) proposed a channel attention mechanism that explicitly modeled the interdependencies between channels, adaptively recalibrating the channel feature responses to enhance the network's expressive ability. Wang et al. (2020) introduced an efficient module for channel attention mechanism, addressing the issue of increased model complexity, reducing parameters, and improving performance. Woo et al. (2018) presented the Convolutional Block Attention Module (CBAM), inferring attention maps in both channel and spatial dimensions and multiplying them with the input feature map to achieve adaptive feature refinement. CBAM is a lightweight general module with negligible computational overhead. Hou et al. (2021) pointed out that the channel attention mechanism overlooks spatial position information and proposed the Coordinate Attention, which embeds spatial position information into channel attention. It can be flexibly applied to other networks with low computational cost and without additional overhead. Misra et al. (2021) introduced Triplet Attention, which captures cross-dimensional interactions using a three-branch structure to calculate attention weights.

Method

The network architecture of PAR-YOLO is shown in Fig. 1. Firstly, in the backbone section of the network, the traditional bottleneck structure is replaced with the Ghost Bottleneck to make the network more lightweight. Secondly, the NSM is introduced in the neck section of the network. This NSM suppresses noise interference and improves the detection accuracy of the network. Lastly, the Varifocal Loss function is introduced in the head section of the network. This loss function balances the loss of positive and negative samples and focuses on difficult-to-detect high-quality positive samples to enhance detection accuracy of the network.

Ghost bottleneck module

In water surface garbage detection scenarios, the position of objects is often influenced by factors such as water currents or wind speed, resulting in their movement. Therefore, real-time or near real-time detection is necessary for timely retrieval by water surface cleaning robots. To ensure real-time performance, the model needs to be lightweight. Consequently, in designing algorithms for water surface garbage detection, the lightweight design of the model becomes particularly crucial.

The Ghost Module is a method for model compression, which consists of two main parts in its design concept. The first part generates intrinsic feature maps with fewer channels using traditional convolution. The second part utilizes the intrinsic feature maps from the first part to generate

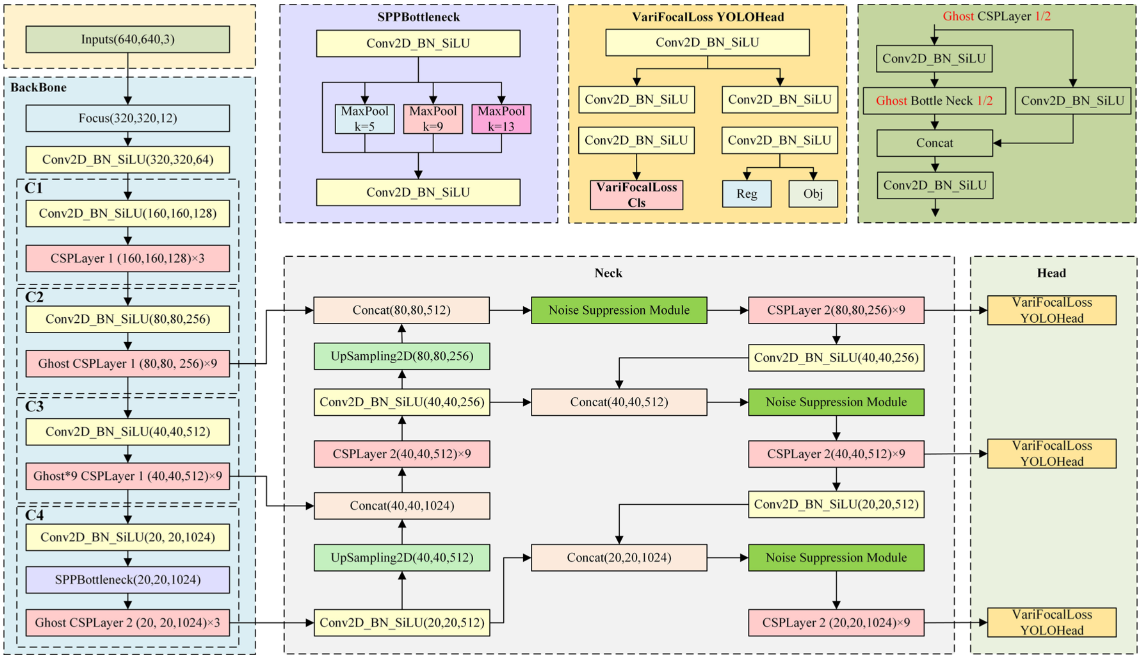


Fig. 1 The network architecture diagram of PAR-YOLO

Ghost feature maps through grouped convolution. Finally, the intrinsic feature maps obtained in the first step and the Ghost feature maps obtained in the second step are concatenated to form the final output result.The specific process is shown in Fig. 2.

The Floating Point Operations (FLOPs) of traditional convolution are shown in Eq. 1.

$$\text{FLOPs}_1 = n \times H \times W \times C \times k \times k \tag{1}$$

where n represents the number of output channels, H and W represent the spatial dimensions of the input feature map, C represents the number of input channels, and k represents the size of the convolutional kernel.

The FLOPs of Ghost Module are shown in Eq. 2.

$$\begin{aligned}\text{FLOPs}_2 =& \frac{n}{t} \times H \times W \times C \times k \times k \\ &+ (t-1) \times H \times W \times \frac{n}{t} \times d \times d\end{aligned} \tag{2}$$

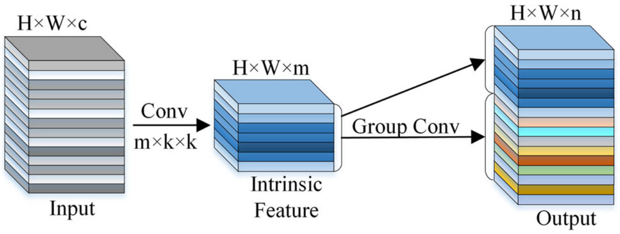


Fig. 2 The Ghost Module implementation process

where t is a hyperparameter of the Ghost Module, referred to as compression rate, where a value of $t = 1$ corresponds to the FLOPs of traditional convolution and d represents the size of the grouped convolutional kernel.

The ratio r of $FLOPs_1$ to $FLOPs_2$ is shown in Eq. 3.

$$\begin{aligned} r =& \frac{FLOPs_1}{FLOPs_2} \\ =& \frac{n \times C \times k \times k}{\frac{n}{t} \times C \times k \times k + (t-1) \times \frac{n}{t} \times d \times d} \\ =& \frac{C \times k \times k}{\frac{1}{t} \times C \times k \times k + \frac{(t-1)}{t} \times d \times d} \\ \approx& \frac{C \times t}{C + t - 1} = \frac{C}{1 + \frac{(t-1)}{C}} \approx t \end{aligned} \tag{3}$$

Therefore, compared to traditional convolution, using the Ghost Module theoretically saves t times the FLOPs.

In the YOLOX model, CSPLayer(Cross Stage Partial Layer), as a key component, has a significant impact on the model's performance and efficiency due to its internal structural design. To optimize computational complexity and achieve model lightweighting, we considered embedding the Ghost Module within CSPLayer. CSPLayer internally iterates the use of bottleneck structures multiple times, which come in two configurations as shown in Fig. 3. One configuration includes residual connections, while the other does not. We introduced the Ghost Module for these two types of bottleneck structures, leading to the derivation of two Ghost Bottleneck structures, as depicted in Fig. 4. Considering that

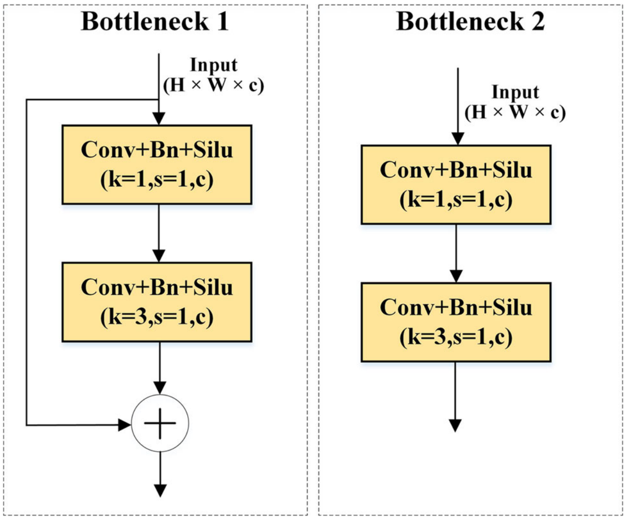


Fig. 3 The bottleneck structure in CSPLayer

the neck part of the model is primarily responsible for feature fusion between feature maps after dimensionality reduction, and its computational burden is relatively light, retaining the original bottleneck structure aligns better with its design intent. Therefore, we chose not to optimize the computation for the neck part and instead focused on optimizing the backbone network. Through ablation experiments, we determined the optimal insertion points for Ghost Bottleneck in the backbone network, with experimental results detailed in Table 5. Ultimately, we decided to replace the bottleneck structures in the C2, C3, and C4 parts of the backbone network to maximize computational efficiency while maintaining model accuracy.

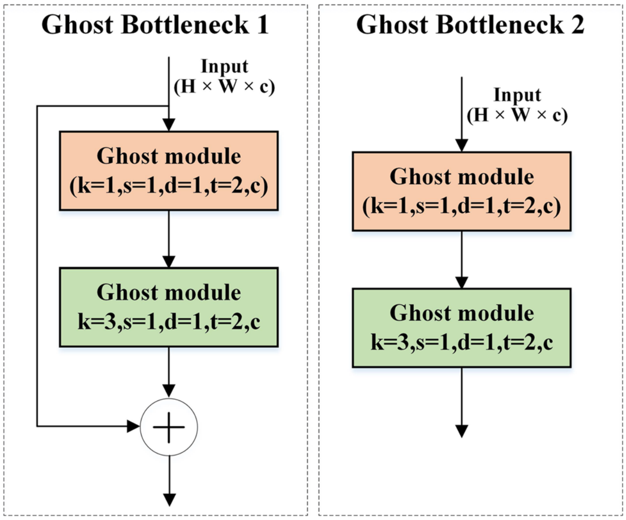


Fig. 4 The Ghost Bottleneck Module

Noise suppression module

Due to the complexity and variability of the water surface environment, garbage on the water surface is often hindered by complex environmental noise. This makes it difficult for the detection head to capture the key feature information of the objects, resulting in a decrease in the model's detection performance. Therefore, reducing noise interference on the feature map and enhancing the features of the objects are crucial for improving model performance. To address this, we have designed a lightweight Noise Suppression Module that effectively suppresses noise information on the feature map, as shown in Fig. 5.

The noise suppression module consists of a channel attention branch and a spatial attention branch. The implementation process can be divided into two steps: firstly, calculating the channel attention weights and spatial attention weights, and secondly, multiplying these weights with the original feature map. Through these two attention calculations, the noise suppression module enhances important channel and spatial dimension information while suppressing noise. The specific implementation process of noise suppression will be described in detail below

The first step involves calculating the channel attention weights. Firstly, the input feature map F undergoes global average pooling. Subsequently, the resulting feature map is fed into a two-layer fully connected network. The number of neurons in the first fully connected layer is C/r, where r is the reduction ratio, and the number of neurons in the second fully connected layer is C. The output of the first fully connected layer is passed through a ReLU activation function. Finally, the channel attention weights are obtained by applying a Sigmoid activation function. These channel attention weight coefficients are then multiplied element-wise with the original input feature map F along the spatial dimensions. The formula for calculating the channel attention weights $M_C(F)$ is as follows:

$$\begin{aligned} M_C(F) &= \sigma(\text{MLP}(\text{Avg}_s(m))) \\ &= \sigma\left(W_1\left(W_0\left(m_{\text{avg}}^C\right)\right)\right) \end{aligned} \tag{4}$$

where $\sigma(\cdot)$ represents the Sigmoid function, W_0 represents the weights of the first fully connected layer, W_1 represents the weights of the second fully connected layer, $\text{Avg}_s(\cdot)$ represents average pooling of spatial dimensions, and C represents the number of channels in the feature map.

The second step is to calculate the spatial attention weights. Firstly, the input feature map F is convolved in parallel with kernel sizes of 1, 3, and 5 to extract contextual information of each pixel on the feature map F. The resulting

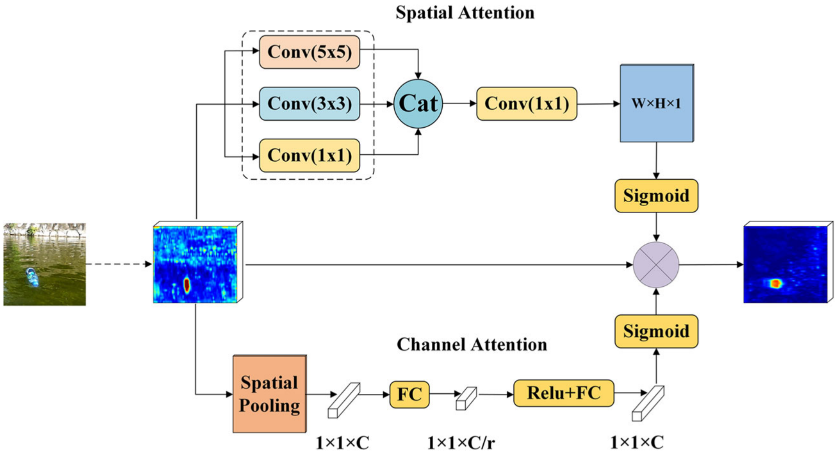


Fig. 5 The overall structure diagram of NSM

feature maps of size H×W×C are then concatenated along the channel dimension. Subsequently, a 1 × 1 convolution is applied to the concatenated feature map. After the convolution, a Sigmoid activation function is used to obtain the spatial attention weight $M_S(F)$. The formula for calculating the spatial attention weights $M_S(F)$ is as follows:

$$\begin{aligned} M_S(F) &= \sigma\left\{f^1\left[\psi\left(f^1(F), f^3(F), f^5(F)\right)\right]\right\} \\ &= \sigma\left\{f^1\left[F'\right]\right\} \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ represents the Sigmoid function, $f^k(\cdot)$ represents $k \times k$ convolutional operation, $\psi(\cdot)$ represents the concatenation operation of feature maps along the channel dimension, and F' represents the feature map obtained after the concatenation operation along the channel dimension.

The formula for the NSM can be expressed as follows:

$$NSM(F) = M_C(F) \times F \times M_S(F) \tag{6}$$

Compared to using the channel attention mechanism or the spatial attention mechanism separately, NSM simultaneously considers both the channel information and the spatial information of detecting objects. Applying this module to the model can suppress noise interference and improve the model's detection performance on objects. Moreover, NSM is a lightweight attention module that is plug-and-play, providing a significant performance boost to the model with minimal computational overhead.

Varifocal loss function

In the YOLOX network, the network prediction section consists of three decoupled detection heads at different levels. Each detection head structure can be further divided into three prediction branches: Cls, Reg, and Obj. The Cls branch is responsible for predicting the classification scores, the Reg branch predicts the offset parameters for prior boxes' positions, and the Obj branch predicts the object confidence scores. In the Cls branch, binary cross-entropy loss is employed as the loss function for classification score prediction. However, binary cross-entropy loss is sensitive to class imbalance situations. During network training, the number of background prediction boxes significantly outweighs the number of object prediction boxes. This causes the model to tend to predict the boxes as the background class, which has a higher quantity, rather than treating both classes equally. As a result, the predictive performance of the model on object classes decreases. Additionally, the binary cross-entropy loss has limited capability in handling difficult samples. For those samples that are hard to classify, the loss function cannot provide more attention and correspondingly increase their loss weight, leading to poorer performance of the model in classifying these challenging samples.

To address the issues mentioned above, we have adopted Varifocal Loss in the classification loss, which is an existing method that draws inspiration from Focal Loss (Lin et al. 2017). Focal Loss reduces the weight of background class samples by introducing an adjustable modulation factor, alleviating the contribution of numerous easily classified background samples to the loss function. Specifically, it shown in Eq. 7.

$$\mathrm{FL(p, y)} = \begin{cases} -\alpha(1-\mathrm{p})^{\gamma}\log(\mathrm{p}) & \text{if } \mathrm{y}=1 \\ -(1-\alpha)\mathrm{p}^{\gamma}\log(1-\mathrm{p}) & \text{otherwise} \end{cases} \tag{7}$$

where γ represents the weight factor, α represents the ratio factor for balancing positive and negative sample losses, p represents the classification score, y represents the sample class, where $y = 1$ corresponds to the object class and 'otherwise' refers to the background class.

However, the modulation factor in Focal Loss is fixed and does not adaptively adjust for different samples. In contrast, Varifocal Loss function utilizes the predicted confidence score as a dynamic modulation factor to adaptively adjust the loss weight of each sample. Specifically, it shown in Eq. 8.

$$\mathrm{VFL}(p,q)=\begin{cases}-q(q\log(p)+(1-q)\log(1-p)) & q>0\\ -\alpha p^{\gamma}\log(1-p) & q=0\end{cases}\tag{8}$$

where p represents the predicted score for object classification, and q represents the predicted score for object confidence. As a result, as the confidence increases, the dynamic modulation factor q becomes larger, leading to a higher loss weight. This allows the model to pay more attention to high-quality positive samples and prioritize their optimization.

The overall loss function of the PAR-YOLO model is shown in Eq. 9:

$$Loss=L_{cls}+\beta L_{reg}+\frac{L_{obj}}{N_{pos}}\tag{9}$$

where L_{cls} represents the classification loss, which uses the Varifocal Loss function. L_{reg} represents the regression loss, which uses the IoULoss function. L_{obj} represents the confidence loss, which uses the binary cross-entropy loss function. N_{pos} represents the number of Anchor Points classified as positive samples. β represents the weight coefficient of L_{obj}.

Experiment

Experimental environment and evaluation indicators

The software environment used in this study is Python 3.7 and PyTorch 1.7.1+cu110. The hardware environment consists of a GeForce RTX 3090 24G graphics card. During the training process of the PAR-YOLO model, a batch size of 16 was used, with a maximum training epoch of 300. The resolution of the input image is uniformly resized to 640×640. Data augmentation techniques such as mosaic, flipping, and cropping were employed in the experiments. After extensive experimentation, we set the hyperparameters α and γ in Eq. 8 to 2.0 and 0.75, respectively. The parameter β in Eq. 9 was set to 5.0. All other hyperparameters were kept at the default values of YOLOX without any modifications.

In the experiments, FLOPs, Params, and Frames Per Second (FPS) were used as evaluation metrics for model lightweightness, while mean Average Precision (mAP) was used as the evaluation metric for model detection accuracy. The mAP metric represents the average precision(AP) across all classes in the dataset, and it is calculated as the mean of individual class average precision values. This can be represented by Eq. 10.

$$mAP=\frac{1}{m}\sum_{i=1}^{m}AP_i\tag{10}$$

where m represents the total number of classes, and the AP is the area under the precision-recall (P-R) curve generated with recall on the horizontal axis and precision on the vertical axis. The calculation formulas for precision and recall are shown in Eq. 11:

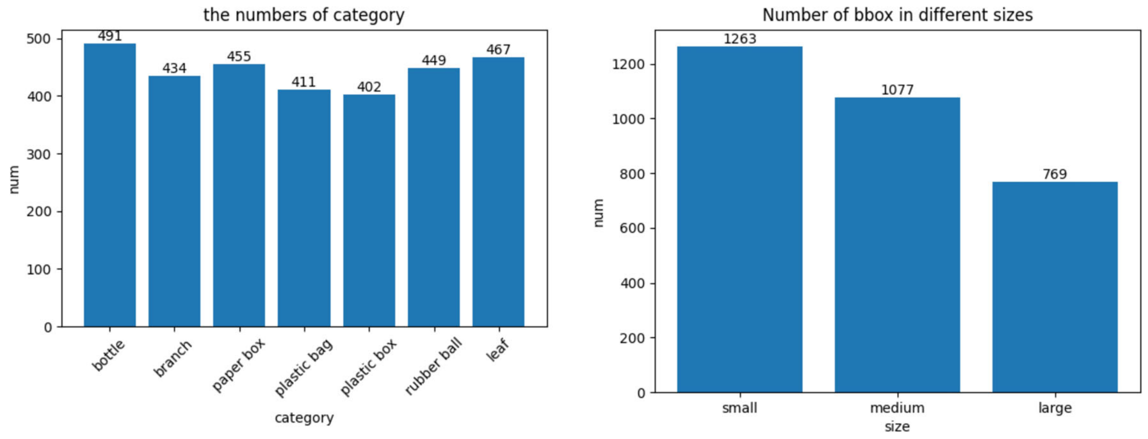
$$\begin{aligned}\text{Precision}&=\frac{TP}{TP+FP},\\ \text{Recall}&=\frac{TP}{TP+FN}\end{aligned}\tag{11}$$

where TP refers to true positive samples, which in the equation represents the number of detection boxes with IoU $>$ IoU$_{threshold}$. FP refers to false positive samples, which represents the number of detection boxes with IoU$\leq$ IoU$_{threshold}$. FN refers to false negatives, which in the experiment represents the number of missed object objects.

Datasets

For the convenience of conducting research on water surface garbage detection, we constructed our own water surface garbage dataset. The dataset consists of a total of 2380 images, including images of water surface garbage from lakes and images of water surface garbage sourced from the internet. We manually annotated the dataset and divided it into training, validation, and test sets in an 8:1:1 ratio. The dataset covers seven main categories, namely bottle, branch, plastic bag, paper box, rubber ball, plastic box, and leaf. Figure 6 demonstrates the distribution of object quantities for each category in the dataset and the distribution of large, medium, and small objects. From Fig. 6(a), it can be observed that the number of objects in each category ranges from 400 to 500, with a small difference, achieving a balanced distribution of samples for each category.

In the official definition of the COCO dataset (Lin et al. 2014), objects are categorized into large, medium, and small based on their spatial coverage in the image. Specifically, when the area of the object's bounding box is less than 32^2 pixels, the object is classified as a small object; if the bounding box area is between 32^2 and 96^2 pixels, the object is classified as a medium object; and when the bounding box area exceeds 96^2 pixels, the object is classified as a large object. We follow the official classification criteria of the COCO dataset and have meticulously categorized the objects



(a)

(b)

Fig. 6 Data analysis of self-made water surface garbage dataset. (a) The distribution of the number of objects in each category in the self-made water surface garbage dataset; (b) The distribution of large, medium, and small objects in the self-made water surface garbage dataset

in our self-made water surface garbage dataset by size, which is visually presented in Fig. 6(b). It can be seen that in the dataset, small objects account for the majority with 1463 objects, followed by medium-sized objects with 1177, and large objects with 870.

To validate the effectiveness of our model, we also utilized the publicly available water surface garbage detection dataset called Flow (Cheng et al. 2021). This dataset is captured from the perspective of unmanned boats, which aligns more closely with the camera angles of water surface cleaning robots. The Flow dataset consists of 2000 images, with 1200 images in the training set and 800 images in the test set. This dataset contains only one category of bottles, and contains a total of 5271 annotated objects.The performance of our model can be further evaluated using this dataset.

To validate the generality of our model, we conducted experiments on the Pascal VOC2007 (Everingham et al. 2010) public dataset. This dataset comprises three parts: a training set, a validation set, and a test set. Specifically, the training and validation sets consist of 5,011 images, while the test set contains 4,952 images. These images cover a wide range of object categories within 20 natural scenes, including humans, animals, vehicles, and household items. By conducting experiments on this dataset, we ensure that our model not only performs well on specific tasks but also demonstrates good generality and adaptability.

Comparative experiments

We first compared the detection accuracy of PAR-YOLO and the baseline model YOLOX for each category in our self-made water surface garbage dataset, as shown in Table 1. PAR-YOLO demonstrates varying degrees of improvement in the detection accuracy for each category. Among them, the bottle category shows the largest improvement with an increase of 2.7% in the AP metric. These improvements are attributed to the NSM and the Varifocal Loss function. The NSM module effectively reduces the interference of environmental factors on the recognition of object features. The Varifocal Loss function enhances the model's focus on difficult samples, thereby improving the model's detection accuracy.

We also conducted comparative experiments on our self-made water surface garbage dataset with other detection models, including one-stage models such as SSD, YOLOv5, and YOLOX, two-stage models such as Cascade R-CNN and Dynamic R-CNN, as well as transformer-based detection model DETR and Deformable-DETR. The experimental results are shown in Table 2. Compared to the other models, PAR-YOLO achieved the best results in all metrics. The mAP metric improved by 1.13% compared to the second-best model, Cascade R-CNN. The Params metric decreased

Table 1 The detection accuracy comparison of PAR-YOLO and the baseline model YOLOX for each category in self-made water surface garbage dataset

PAR-YOLO		YOLOX	
category	AP(%)	category	AP(%)
bottle	85.9	bottle	83.2
branch	80.9	branch	80.1
plastic bag	87.5	plastic bag	85.6
paper box	84.8	paper box	83.3
rubber ball	88.8	rubber ball	87.3
plastic box	84.8	plastic box	83.2
leaf	86.0	leaf	84.1

Table 2 The experimental results comparison of different models on our self-made water surface garbage dataset

Model	mAP(%)	Params(M)	FPS	FLOPs(G)
SSD (Liu et al. 2016)	79.11	26.29	72	353.21
YOLOv5 (Ultralytics 2020)	84.65	46.2	125	108.3
YOLOX (Ge et al. 2021)	83.83	8.94	229	26.65
Cascade R-CNN (Cai and Vasconcelos 2018)	84.15	41.75	47	206.47
Dynamic R-CNN (Zhang et al. 2020)	83.62	40.62	49	198.52
DETR (Carion et al. 2020)	81.86	36.74	50	100.94
Deformable-DETR (Zhu et al. 2020)	83.25	40.00	49	173.00
PAR-YOLO	**85.53**	**7.42**	**238**	**22.56**

Best results are in **bold**

by 1.64M, the FLOPs metric decreased by 4.57G and the FPS metric improved by 11 compared to the second-best YOLOX model.

We also conducted comparative experiments on the publicly available water surface garbage dataset called Flow, and the experimental results are shown in Table 3. As per the results in the table, the PAR-YOLO model achieved the best results in all metrics on the Flow dataset. The mAP metric improved by 0.4% compared to the second-best model, Cascade R-CNN.

PAR-YOLO excels in the task of water surface garbage detection, partly due to its adoption of a one-stage detection architecture, which offers a significant advantage in processing speed. In addition to this, we have implemented a series of targeted improvements, including reducing computational complexity, enhancing the model's noise resistance in water surface environments, and increasing attention to difficult samples. These enhancements have significantly boosted the overall performance of PAR-YOLO, demonstrating exceptional performance across all evaluation metrics. In contrast, other comparative methods have not been specifically optimized for water surface garbage detection scenarios, thus exhibiting certain limitations in detection accuracy. We believe that further specialized research based on these methods for the scenario of water surface garbage detection could also achieve detection accuracy comparable to that of PAR-YOLO.

The comparative experiments of the PAR-YOLO model with other advanced one-stage and two-stage detection models on the Pascal VOC2007 dataset are shown in Table 4. PAR-YOLO achieves the best results in all performance metrics. Compared to the second-best Cascade R-CNN model, PAR-YOLO achieves a precision improvement of 0.6% in mAP. Compared to the baseline model, PAR-YOLO achieves a precision improvement of 1.9% in mAP. PAR-YOLO has demonstrated outstanding performance in experiments, primarily because there is also a certain degree of environmental noise interference and difficult sample issues in general object detection scenarios. Therefore, the improvements we made for the water surface garbage scenario are also applicable to general object detection scenarios and can enhance the model's detection accuracy. At the same time, PAR-YOLO has been specifically optimized for model efficiency, enabling it to surpass other one-stage detection networks in detection speed, such as SSD, YOLOv5, and YOLOX. Moreover, due to its one-stage detection architecture, PAR-YOLO significantly outperforms two-stage networks in inference speed, such as Cascade R-CNN and Dynamic R-CNN, as well as Transformer-based detection models like DETR and Deformable-DETR. These models, due to their complex structural characteristics, struggle to match PAR-YOLO in inference speed. It is worth noting that due to the limited scale of the VOC2007 dataset, the excellent performance of

Table 3 The experimental results comparison of different models on the Flow dataset

Model	mAP(%)	Params(M)	FPS	FLOPs(G)
SSD (Liu et al. 2016)	26.9	26.29	72	353.21
YOLOv5 (Ultralytics 2020)	45.2	46.2	125	108.3
YOLOX (Ge et al. 2021)	43.3	8.94	229	26.65
Cascade R-CNN (Cai and Vasconcelos 2018)	46.8	41.75	47	206.47
Dynamic R-CNN (Zhang et al. 2020)	46.2	40.62	49	198.52
DETR (Carion et al. 2020)	40.4	36.74	50	100.94
Deformable-DETR (Zhu et al. 2020)	45.3	40.00	49	173.00
PAR-YOLO	**47.3**	**7.42**	**238**	**22.56**

Best results are in **bold**

Table 4 The experimental results comparison of different models on the Pascal VOC2007 dataset

Model	mAP(%)	Params(M)	FPS	FLOPs(G)
SSD (Liu et al. 2016)	22.4	26.29	72	353.21
YOLOv5 (Ultralytics 2020)	27.3	46.2	125	108.3
YOLOX (Ge et al. 2021)	26.6	8.94	229	26.65
Cascade R-CNN (Cai and Vasconcelos 2018)	27.9	41.75	47	206.47
Dynamic R-CNN (Zhang et al. 2020)	27.6	40.62	49	198.52
DETR (Carion et al. 2020)	23.4	36.74	50	100.94
Deformable-DETR (Zhu et al. 2020)	25.0	40.00	49	173.00
PAR-YOLO	**28.5**	**7.42**	**238**	**22.56**

Best results are in **bold**

Transformer models on large datasets was not fully demonstrated.

Ablation experiments

First, we conducted ablation experiments on our self-made water surface garbage dataset to determine which bottleneck modules in the backbone of YOLOX can be replaced by the Ghost Bottleneck module to improve model performance.The backbone of YOLOX consists of four bottleneck modules located in the C1, C2, C3, and C4 layers. We conducted ablation experiments at these four positions, and the results are shown in Table 5. After replacing the C1 layer, the FPS increased by 9 frames, but the mAP decreased by 0.5%. This phenomenon is mainly because there is a lot of noise in the early feature maps, and traditional convolutions can more effectively filter features. Subsequently, when only replacing the C2 layer, the mAP increased by 0.22%, while Params and FLOPs decreased by 0.59M and 0.95G respectively, and the FPS increased by 6 frames. Building on this, after further replacing the C3 and C4 layers, the mAP was slightly lower than the baseline model by 0.03%, but Params and FLOPs were significantly reduced by 1.8M and 4.66G respectively, and the FPS increased by 17 frames compared to the baseline model. Therefore, considering the trade-off between accuracy and computational efficiency, the final decision was to replace the bottleneck modules in the C2, C3, and C4 layers.

Table 5 The ablation experimental results of replacing bottleneck modules at different positions in the YOLOX backbone with Ghost Bottleneck on our self-made water surface garbage dataset

replacement	mAP(%)	Params(M)	FPS	FLOPs(G)
Baseline	83.83	8.94	229	26.65
C1	83.33	8.43	238	24.76
C2	**84.05**	8.35	235	25.70
C2&C3	84.01	7.78	242	23.04
C2&C3&C4	83.80	**7.14**	**246**	**21.99**

Best results are in **bold**

We then performed ablation experiments on our self-made water surface garbage dataset to evaluate the impact of three modules: Ghost Bottleneck, NSM, and Varifocal Loss. The experimental results are shown in Table 6. When we separately added the NSM and Varifocal Loss modules, the mAP metric improved. However, after adding the NSM module, the Params and FLOPs metrics increased, and the FPS metric decreased. After adding the Ghost Bottleneck module, both the Params and FLOPs metrics decreased, and the FPS metric increased, while the mAP metric decreased by 0.03%, which is within an acceptable range. Subsequently, we combined them pairwise and added them to YOLOX, resulting in optimization of all performance metrics without conflicts among them. Finally, by incorporating these three modules into YOLOX, we achieved mAP of 85.53%, Params of 7.42M, FPS of 238, and FLOPs of 22.56G. Although each individual metric is not optimal, considering the overall detection accuracy and real-time performance of the model, we reached the best possible result (Table 6).

We further validated the effectiveness of our modules on the Flow dataset. In the experiments, we integrated the Ghost Bottleneck, NSM, and Varifocal Loss modules into the base model in two ways: individual integration and progressive integration. The detailed experimental results are shown in Table 7. The results demonstrate that both the NSM and Varifocal Loss modules effectively improve the detection accuracy of the model. When these two modules are combined, the model achieves the best performance in terms of mAP, with a significant improvement of 4.1% compared to the baseline method. Although there is a slight decrease in mAP after integrating the Ghost Bottleneck module, it significantly reduces the number of parameters and computational complexity while greatly improving the inference speed, aligning with our design intention of balancing detection accuracy and real-time performance.

To comprehensively demonstrate the universality and excellence of our modules, we conducted ablation experiments on the Pascal VOC2007 dataset. In the experiments, we

Table 6 Ablation experiments of the Ghost Bottleneck, NSM, and Varifocal Loss modules on our self-made water surface garbage dataset

Ghost Bottleneck	NSM	Varifocal Loss	mAP(%)	Params(M)	FPS	FLOPs(G)
			83.83	8.94	229	26.65
√			83.80	**7.14**	**246**	**21.99**
	√		85.12	9.10	204	27.53
		√	84.73	8.94	229	26.65
√	√		84.75	7.42	238	22.56
	√	√	**85.68**	9.10	204	27.53
√		√	84.47	**7.14**	**246**	**21.99**
√	√	√	85.53	7.42	238	22.56

Best results are in **bold**. Second best results are in underline

followed the same approach of individually adding and progressively integrating the modules into the baseline model. The detailed experimental results are presented in Table 8. The experimental data clearly indicate that each module not only contributes significantly on its own but also achieves optimal performance through synergistic effects. Ultimately, by incorporating these three modules into the baseline model, we achieved the best result with an mAP of 28.5%, a parameter count of 7.42M, an inference time of 4.17ms, and FLOPs of 22.56.

It is worth noting that this series of experiments not only confirms the individual contributions of our modules but also demonstrates their potential when working together, providing valuable insights for optimizing model performance and improving efficiency.

Results visualization and analysis

PAR-YOLO optimizes the computational complexity of the model, suppresses the interference of noise features on feature maps, and introduces a superior loss function, effectively enhancing the detection efficiency and improving detection accuracy. Partial detection results on our self-made water surface garbage dataset are shown in Fig. 7. The PAR-YOLO model can accurately identify the type of garbage and accurately frame the objects. Moreover, thanks to the noise suppression module, even in the presence of aquatic plant obstruction, strong lighting, and reflection interference, the model can achieve good detection results.

Although the PAR-YOLO model has demonstrated certain potential on our self-made water surface garbage dataset, its performance is still subject to several limitations. By analyzing the detection results of the model in the Flow dataset, we found that the model performs poorly when dealing with special types of objects and scenes. For instance, the visualization results in column 1 of Fig. 8 show that the model is unable to effectively identify extremely small objects. Moreover, the model has limited detection capabilities for objects partially submerged in water, as shown in column 2 of Fig. 8. For stacked objects, the model also failed to accurately box out each individual, as shown in column 3 of Fig. 8. Under poor lighting conditions, due to the inconspicuous features of the objects, the model similarly failed to achieve effective detection, as shown in column 4 of Fig. 8. These failure cases reflect the limitations of our model and the insufficiencies of the self-made dataset. To address these issues, continuous

Table 7 Ablation experiments of the Ghost Bottleneck, NSM, and Varifocal Loss modules on Flow dataset

Ghost Bottleneck	NSM	Varifocal Loss	mAP(%)
			43.3
√			43.1
	√		45.9
		√	45.5
√	√		45.4
	√	√	**47.4**
√		√	45.4
√	√	√	47.3

Best results are in **bold**. Second best results are in underline

Table 8 Ablation experiments of the Ghost Bottleneck, NSM, and Varifocal Loss modules on Pascal VOC2007 dataset

Ghost Bottleneck	NSM	Varifocal Loss	mAP(%)
			26.6
√			26.5
	√		27.7
		√	27.5
√	√		27.5
	√	√	**27.7**
√		√	27.4
√	√	√	28.5

Best results are in **bold**. Second best results are in underline

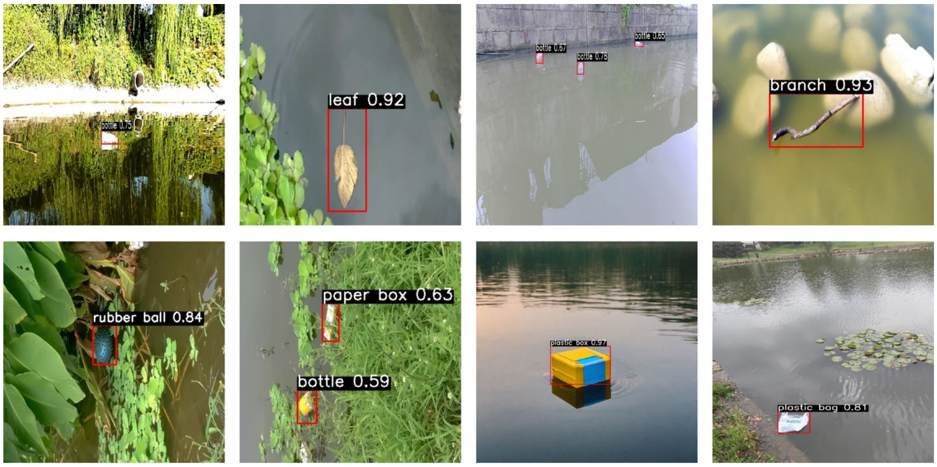


Fig. 7 Detection results of the PAR-YOLO model on the self-made water surface garbage dataset.

optimization of model performance is needed to ensure better results in real scenarios. At the same time, our self-made dataset also needs to be further enhanced in terms of diversity and breadth, especially by incorporating samples from different environments and conditions, in order to more accurately reflect the complexity of real-world scenarios.

Conclusion

In the task of water surface garbage detection, accuracy and real-time performance are two key challenges. To address these challenges, we propose a water surface garbage detection model named PAR-YOLOX. The improvements of PAR-YOLOX mainly include the following aspects: (1) In the backbone part of the model, a more efficient Ghost Bottleneck module is designed to reduce computational burden; (2) In the neck part of the model, NSM is designed to suppress water surface noise interference, thereby improving detection accuracy; (3) In the head part of the model, the Varifocal Loss function is adopted to enhance the model's focus on difficult samples. Finally, the effectiveness of the PAR-YOLOX model is demonstrated through experiments on water surface garbage detection datasets and general object detection datasets, providing certain assistance for the development of water surface garbage detection tasks in terms of accuracy and real-time performance.

Although our research has made some progress, it still faces several challenges. First, the performance of the PAR-YOLO algorithm in handling occlusions between objects, extremely small object recognition, and object detection under low-light conditions is still insufficient and requires further optimization. Second, the quality of our self-made water surface garbage dataset needs to be improved, and the diversity of samples and scenarios in the dataset should be enhanced through more extensive data collection. Future research directions will delve into these issues with the aim of advancing the technology in the field of water surface garbage detection.

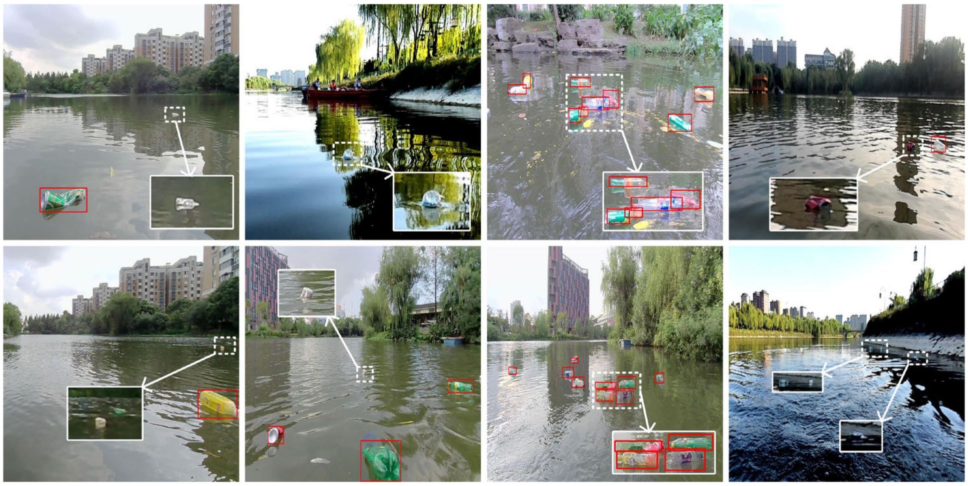


Fig. 8 Some failure cases of detection results on the Flow dataset

Author Contributions Ning Li and Mingliang Wang have led the conception and design of the work, as well as the acquisition and interpretation of data. They've also been instrumental in drafting and revising the content to ensure intellectual value. The final version has been reviewed by He Huang, Bo Li, Baohua Yuan, and Shoukun Xu.

Funding This work was supported by Jiangsu Petrochemical Process Key Equipment Digital Twin Technology Engineering Research Center Open Project (DTEC202103).

Data Availability The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

No datasets were generated or analysed during the current study.

Declarations

Conflicts of Interest All authors of this research paper declare that they have no conflict of interest.

Competing interests The authors declare no competing interests.

References

Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767

Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934

Ge Z, Liu S, Wang F, Li Z, Sun J (2021) Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp 21–37. Springer

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88:303–338

Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6154–6162

Zhang H, Chang H, Ma B, Wang N, Chen X (2020) Dynamic r-cnn: Towards high quality object detection via dynamic training. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, pp 260–275. Springer

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30

Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European Conference on Computer Vision, pp 213–229. Springer

Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159

Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni L.M, Shum HY (2022) Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605

Wen L, Ding J, Xu Z (2021) Multiframe detection of sea-surface small target using deep convolutional neural network. IEEE Trans Geosci Remote Sens 60:1–16

Zhang L, Wei Y, Wang H, Shao Y, Shen J (2021) Real-time detection of river surface floating object based on improved refinedet. IEEE Access 9:81147–81160

Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4203–4212

Zhang H, Wang Y, Dayoub F, Sunderhauf N (2021) Varifocalnet: An iou-aware dense object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8514–8523

Yang X, Zhao J, Zhao L, Zhang H, Li L, Ji Z, Ganchev I (2022) Detection of river floating garbage based on improved yolov5. Math 10(22):4366

Yi N, Luo W (2023) Research on water garbage detection algorithm based on gfl network. Front Comput Intel Syst 3(1):154–157

Pang Y, Qin B (2021) Gdt-net: A model based on deep learning for water surface garbage detection. In: Proceedings of the 2021 5th International Conference on Deep Learning Technologies, pp 1–7

Cheng Y, Zhu J, Jiang M, Fu J, Pang C, Wang P, Sankaran K, Onabola O, Liu Y, Liu D et al (2021) Flow: A dataset and benchmark for floating waste detection in inland waters. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10953–10962

Ma L, Wu B, Deng J, Lian J (2023) Small-target water-floating garbage detection and recognition based on unet-yolov5s. In: 2023 5th International Conference on Communications, Information System and Computer Engineering (CISCE), pp 391–395. IEEE

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp 234–241. Springer

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4510–4520

Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1314–1324

Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6848–6856

Ma N, Zhang X, Zheng HT, Sun J (2018) Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 116–131

Wen L, Ding J, Xu Z (2021) Multiframe detection of sea-surface small target using deep convolutional neural network. IEEE Trans Geosci Remote Sens 60:1–16

Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1580–1589

Yang X, Zhao J, Zhao L, Zhang H, Li L, Ji Z, Ganchev I (2022) Detection of river floating garbage based on improved yolov5. Math 10(22):4366

Yi N, Luo W (2023) Research on water garbage detection algorithm based on gfl network. Front Comput Intel Syst 3(1):154–157

Zhang L, Wei Y, Wang H, Shao Y, Shen J (2021) Real-time detection of river surface floating object based on improved refinedet. IEEE Access 9:81147–81160

Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13713–13722

Misra D, Nalamada T, Arasanipalai AU, Hou Q (2021) Rotate to attend: Convolutional triplet attention module. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 3139–3148

Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2980–2988

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp 740–755. Springer

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88:303–338

Ultralytics (2020) ultralytics/yolov5. https://github.com/ultralytics/yolov5.com, https://doi.org/10.5281/zenodo.7347926. Accessed: 26 Jun 2020

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