applied sciences

MDPI

Article

YOLOv5-FF: Detecting Floating Objects on the Surface of Fresh Water Environments

Xiaohong Zhang [1,†], Changzhuo Min [1,†], Junwei Luo [1,†] and Zhiying Li [2,*]

[1] School of Software, Henan Polytechnic University, Jiaozuo 454000, China; xh.zhang@hpu.edu.cn (X.Z.); changz_min@163.com (C.M.); luojunwei@hpu.edu.cn (J.L.)
[2] School of Information, Renmin University of China, Beijing 100872, China
* Correspondence: zhiyingli@ruc.edu.cn
† These authors contributed equally to this work.

Abstract: Real-time detection and timely treatment of floating objects on rivers, lakes and reservoirs is very essential to protect water environment and maintain the safety of navigation and water projects. YOLOv5, as a one-stage object detection solution, is very suitable for real-time floating object detection. However, it suffers from the problem of the false detection and missed detection of floating objects especially of small floating objects. In this paper, we conducts a series of improvements on YOLOv5 to alleviate the problem. Concretely, we propose a hybrid attention mechanism supporting the interaction among channels over a long distance while preserving the direct correspondence between channels and their weights. Base on the attention mechanism, we propose an adaptive feature extraction module to capture the feature information of objects in the case of the feature loss caused by downsampling operations. Based on the attention mechanism and dilated encoder, we construct a feature expression enhancement module to cover large objects while not losing small objects in the same certain scale range. We also add a detection layer for small objects to improve the performance in detecting small floating objects. The experiments on the data set verify the usefulness and effectiveness of our work.

Keywords: floating objects; YOLOv5s; attention mechanism; dilated encoder; K-Mediods clustering

Citation: Zhang, X.; Min, C.; Luo, J.; Li, Z. YOLOv5-FF: Detecting Floating Objects on the Surface of Fresh Water Environments. *Appl. Sci.* **2023**, *13*, 7367. https://doi.org/10.3390/app13137367

Academic Editor: Oscar Reinoso García

Received: 16 May 2023
Revised: 14 June 2023
Accepted: 16 June 2023
Published: 21 June 2023

1. Introduction

Rivers, lakes and reservoirs play an important role in water supply and ecological balance. However, floating objects often occur on the surface of rivers, lakes and reservoirs [1–4]. The frequent floating objects can be divided into two categories, one is plant-related floating objects consisting of floating algae and fallen branches and leaves, and the other is garbage-related floating objects mainly including plastic products like plastic bottles, bags and so on. Plant-related floating objects come from nature, while garbage-related floating objects result from human activities. These floating objects can damage water quality, affecting water supply and aquaculture, as well as posing a threat to navigation and water projects [5,6]. Therefore, it is very important and necessary to detect floating objects in time and treat them effectively.

Remote sensing image analysis has been utilized to detect floating objects. Compared with remote sensing images, the images collected by handheld cameras [7] or airborne/shipborne cameras [8,9] are more accessible and suitable for detecting the small objects floating on rivers, lakes and reservoirs. In recent years, deep learning has exhibited a distinct advantage in floating object detection [10,11]. You Only Look Once (YOLO) [12] is eminently suitable for real-time object detection considering its high detection speed. YOLOv5 is an improved version of YOLO. It has attracted more and more attention and applications because it emphasizes both accuracy and speed. YOLOv5 has been applied to detect floating objects [13]. However, it suffers from the problem of the false detection and missed detection of floating objects especially of small floating objects. The problem is

mainly caused by the interference of background, the lack of obvious visual features and the loss of features in downsampling operations.

Many efforts have been made to improve the performance of YOLOv5 in detecting floating objects. Enhancing the feature extraction ability of YOLOv5 is one of those efforts. A common way to enhance the ability is to apply attention mechanism. Yang et al. [14] and Lin et al. [3] have adopted coordinate attention mechanism [15] and an attention layer in the process of feature extracting, respectively. Optimizing the feature fusion of YOLOv5 is another effort devoted to improving the performance in detecting floating objects. There are multiple ways to optimize the feature fusion. For example, Yang et al. use bidirectional feature pyramid network(BiFPN) [16] to replace the feature pyramid network(FPN) [17], and Huang et al. [18] apply attention mechanism to the feature fusion. Moreover, the introduction of a dedicated detection layer for small floating objects also helps to improve the performance of YOLOv5 [18].

Besides floating object detection, a lot of work has been carried out to improve the performance of YOLOv5 in other areas. Liu et al. [19] employed attention mechanism and transformer to extract the features of objects in aerial images. Wan et al. [20] proposed Shuffle-ECANet to extract the features of road damages. Ren et al. [21], Liu et al. [22] and Wan et al. exploited BiFPN to optimize the feature fusion of foreign objects on airport runways, faces and road damages, respectively. Yan et al. [23] introduced wavelet transform into PANet to deal with the feature loss problem of traffic signs after multiple downsampling and pooling operations. Jiang et al. [24] and Liu et al. [19] added a small target detection layer in their works. Although a lot of work has been devoted to improving the performance of YOLOv5 in object detection, how to improve the accuracy of YOLOv5 in detecting floating objects especially small even extra small floating objects on rivers, lakes and reservoirs in real time is still an open problem.

In this work, we propose an improved version of YOLOv5 to alleviate the problem. The improved version is named as YOLOv5-FF. Yolov5-FF is implemented on the basis of YOLOv5 by enhancing the feature extraction capability, strengthening the feature expression ability and introducing a special detection layer for small even extra small floating objects. We also construct a floating object dataset, and generate the appropriate prior boxes matching the dataset with the K-Mediods algorithm [25]. The main contributions of this work are described as follows:

(1) a channel attention mechanism is proposed to support the interaction of channels in different ways. It aims to supporting the interaction of channels over a long distance while preserving the direct correspondence between channels and their weights.
(2) an adaptive feature extraction convolution module is proposed to focus on the feature information of objects in the process of feature extraction. It has been applied to the neck to alleviate the impact of the feature loss caused by downsampling operations.
(3) a feature expression enhancement module is proposed to expand the received fields of feature maps without losing small objects. It has been applied to the neck to ensure that a feature map can cover the objects of different scales in a certain range.
(4) a new detection layer is constructed by exploiting all the feature maps generated by the backbone. It is designed specially for detecting small even extra small objects.

The rest of this paper is organized as follows. Section 2 introduces the theoretical basis. Section 3 elaborates the improved YOLOv5. Section 4 discusses the experimental results. After presenting the intelligent detection system we developed for floating objects in Section 5, this paper is concluded in Section 6.

2. Theoretical Basis

2.1. YOLOv5

YOLOv5 has four variant versions. YOLOv5s [26] is the version with the simplest network structure and the fastest detection speed. Hence, it is very suitable for real-time object detection scenarios. YOLOv5s consists of backbone, neck, and prediction as Figure 1 shows. The backbone uses convolution and pooling operations to continuously

downsample a 640 × 640 input image to generate feature maps of four different scales. The neck uses Feature Pyramid Network (FPN) and Pixel Aggregation Network (PAN) [27] to fuse these three feature maps of the lowest resolutions so that each feature map contains richer features. Prediction uses the fused feature maps to detect objects of different scales.

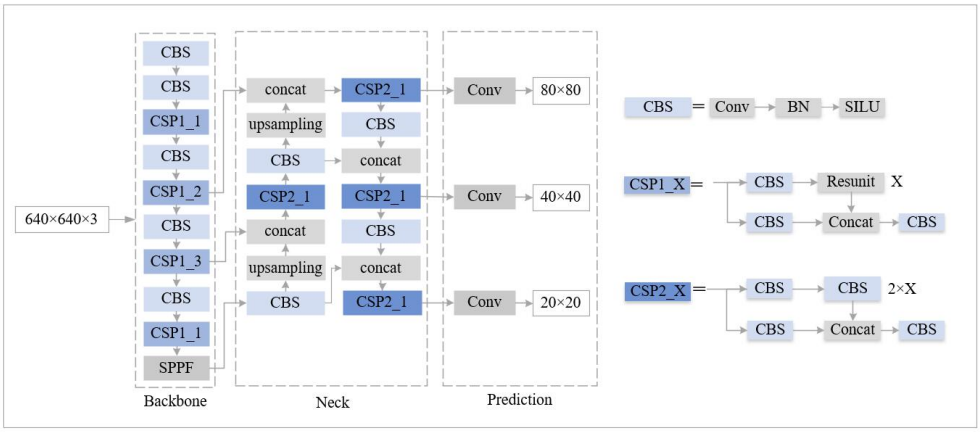


Figure 1. Structure of YOLOv5s.

2.2. Attention Mechanism

The Convolutional Block Attention Module (CBAM) [28] is a hybrid attention mechanism. It consists of a channel attention module and a spatial attention module. The channel attention module is shown in Figure 2. According to the figure, it first performs global maximum pooling and global average pooling separately on an input feature map-F to generate two 1 × 1 × C feature maps. And then, it exploits multilayer perceptron (MLP) to process these two feature maps. After fusing the output of MLP, it generates a channel attention-M_c which is calculated according to Equation (1). In the equation, σ depicts the sigmoid function, and F^C_{avg} and F^C_{avg} denote the feature maps generated by average pooling operations and max pooling operations, respectively. $W_0(W_0 \in \mathbb{R}^{C/r \times C})$ and $W_1(W_1 \in \mathbb{R}^{C \times C/r})$ are the MLP weights.

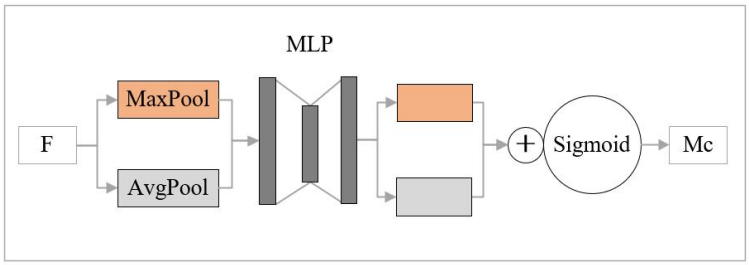


Figure 2. Channel attention module in CBAM.

$$\begin{aligned} M_C(F) &= \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \\ &= \sigma\left(W_1\left(W_0\left(F^C_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^C_{\text{max}}\right)\right)\right) \end{aligned} \tag{1}$$

The spatial attention module first generates two H × W × 1 feature maps by performing maximum pooling and average pooling separately on the output of the channel attention module, and then combines these two feature maps into one H × W × 2 feature map. After that, it executes convolution operations on the H × W × 2 feature map and creates an H × W × 1 feature map. It finally outputs a spatial attention by performing a sigmoid operation on the H × W × 1 feature map.

2.3. Dilated Encoder

Feature pyramid improves the performance of object detection by assigning objects of different scales to the corresponding detection layers. However, Chen et al. [29] found that a feature map cannot cover large objects due to the limitation of receptive field. Although downsampling can increase the receptive fields of feature maps to cover large objects, it also leads to the loss of features of small objects, which improves the detection performance of large objects but degrades the detection performance of small objects. To solve this problem, Chen et al. proposed dilated encoder.

Dilated encoder consists of a projector and a residual block group. The projector first performs a 1×1 convolution on an input feature map to reduce the number of channels, and then executes a 3×3 convolution to enrich the contextual information of the feature map. The residual block group iteratively performs the operations defined by residual blocks four times. The first iteration takes the output of the projector as the input feature map, while each of the other iterations takes the output of the previous iteration as its input feature map. Each iteration processes an input feature map sequentially with a 1×1 convolution, a 3×3 convolution and a 1×1 convolution, and outputs a new feature map with a larger receiver field. After that, it fuses the input feature map with the output feature map with the BN layer and the ReLU function.

2.4. K-Mediods Clustering Algorithm

Generating appropriate prior boxes not only improves the accuracy of the prediction, but also accelerates the convergence speed of the prediction. YOLOv5 utilizes the K-means algorithm and the genetic algorithm to generate prior boxes. The K-means algorithm takes the average of the points in a cluster as the centroid of the cluster. In the case of the outliers in a dataset, the K-means algorithm generates the results with significant errors from the actual results. Different from the K-means algorithm, the K-Mediods algorithm takes the point with the smallest sum of the distances from each of the other points in the same cluster as the centroid of the cluster, which indicates that outliers have less influence on the K-Mediods algorithm. Therefore, the K-Mediods algorithm produces more stable clustering results.

3. Improved Floating Object Detection Solution

In this section, we elaborate the improvements proposed to YOLOv5 to meet the speed and accuracy requirements for real-time floating object detection. The improvements include proposing a hybrid attention mechanism, constructing an adaptive feature extraction convolution module and a feature expression enhancement module, adding a special detection layer for small floating objects and generating appropriate prior bounding boxes. The hybrid attention mechanism and the adaptive feature extraction convolution module are designed to highlight features of objects and suppress the interference of irrelevant information. The feature expression enhancement module is proposed to enhance the ability of each feature map to detect the objects in a certain scale range. The new detection layer is constructed to detect small even extra small objects. Prior box selection algorithm aims to select the prior boxes matching our dataset so as to improve the accuracy and the convergence speed of prediction. All the improvements have been implemented in YOLOv5-FF as Figure 3 shows.

3.1. A hybrid Attention Mechanism Supporting Two Ways of Channel Interaction

Both MLP and the 1-D convolution of K [30] can implement the interaction of channels. Although MLP supports the interaction among channels over a long distance, the dimension reduction operations it uses destroy the direct correspondence between channels and their weights. The 1-D convolution of K can preserve the direct correspondence. However, it only implements the interactions among k adjacent channels. In order to realize the interaction among channels over a long distance while mitigating the damage to the direct

correspondence between channels and weights, we propose a channel attention mechanism based on MLP and the 1-D convolution of K.

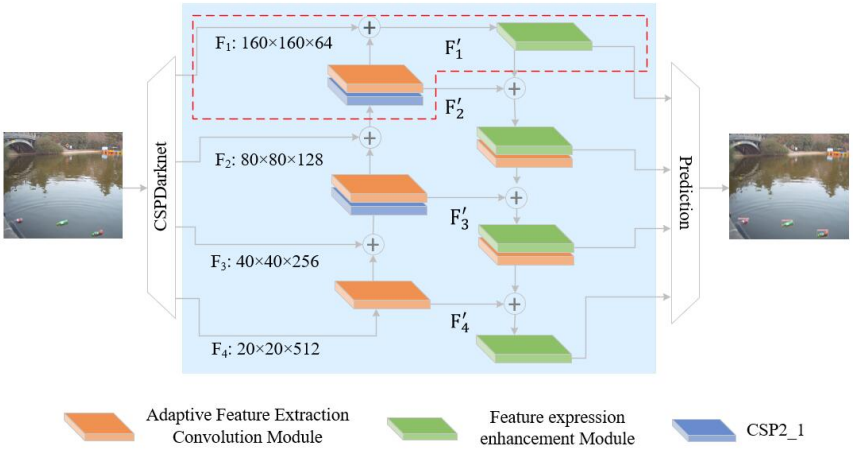


Figure 3. Structure of YOLOv5-FF.

Figure 4 shows the details of the channel attention mechanism. According to the figure, the channel attention mechanism exploits max pooling and average pooling separately to process an input feature map. And then, it utilizes MLP and the 1-D convolution of K to process the results of max pooling and the results of average pooling, respectively. After that, it merges the output of MLP and the 1-D convolution of K, and generates channel attention by applying sigmoid function to the merged results. We combine the proposed channel attention mechanism and the spatial attention mechanism in CBAM, and propose a hybrid attention mechanism named as $H^{2C}AM$.

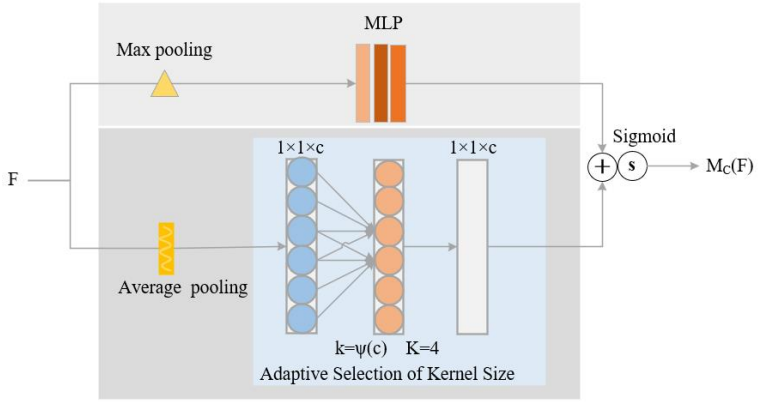


Figure 4. The channel attention module supporting channel interaction in two ways.

$$\begin{aligned} M_C^{'}(F) &= \sigma(\text{Conv}\,1D(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \\ &= \sigma\left(\text{Conv}\,1D\left(F_{\text{Avg}}^{C}\right) + W_1\left(W_0\left(F_{\text{Max}}^{C}\right)\right)\right) \end{aligned} \tag{2}$$

Given a feature map F, $M_C^{'}(F)$ is used to represent the corresponding channel attention. $M_C^{'}(F)$ is calculated according to Equation (2). In the equation, σ depicts the sigmoid function, F_{avg}^{C} and F_{avg}^{C} separately denote the results of average pooling and max pooling, and $W_0(W_0 \in \mathbb{R}^{C/r \times C})$ and $W_1(W_1 \in \mathbb{R}^{C \times C/r})$ represent the MLP weights. Based on the Equation (2), we use Equation (3) to calculate the output of $H^{2C}AM$, where M_S denotes the spatial attention of CBAM.

$$f_{H^{2C}AM}(F) = (M_C^{'}(F) \times F) \times (M_S(M_C^{'}(F) \times F)) \tag{3}$$

3.2. Adaptive Feature Extraction Convolution Module

The backbone generates multiple feature maps of different resolutions by performing downsampling operations. However, downsampling results in the loss of some critical features, e.g., position and color. The feature map of the lowest resolution loses the most features. In the worst case, it could lose all the features of some objects. Unfortunately, this feature map is the start point of the bottom-up feature fusion path in the neck. All the feature maps generated in the bottom-up fusion path are based on these feature maps. Therefore, they all suffer from the problem of feature loss. The top-down fusion path in the neck also exploits downsampling operations to generate feature maps of different resolutions, which also results in feature loss.

$$f_{FEB}(F_i) = \sigma^S(f_{BN}(f^{1\times1}(F_i))) \tag{4}$$

To reduce the impact of feature loss on detection results, we introduce an adaptive feature extraction convolution module (AFECM) to enhance the expression ability of each feature map. Figure 5 shows the structure of AFECM. AFECM consists of two blocks: feature extraction block (FEB) and enhance block (EnB). FEB is implemented based on CBS in Figure 1. It includes a convolution layer, a batch normalization layer and an activation layer. Given an input feature map F_i, the output of FEB is noted as $f_{FEB}(F_i)$ and calculated according to Equation (4), where σ^S represents Silu activation function, and f_{BN} denotes the batch normalization function [31]. EnB is introduced to emphasize features and suppress the interference of irrelevant information. It is implemented by $H^{2C}AM$. The output of AFECM is noted as $f_{AFEC}(F_i)$ and calculated according to Equation (5).

$$f_{AFEC}(F_i) = f_{H^{2C}AM}(f_{FEB}(F_i)) \tag{5}$$

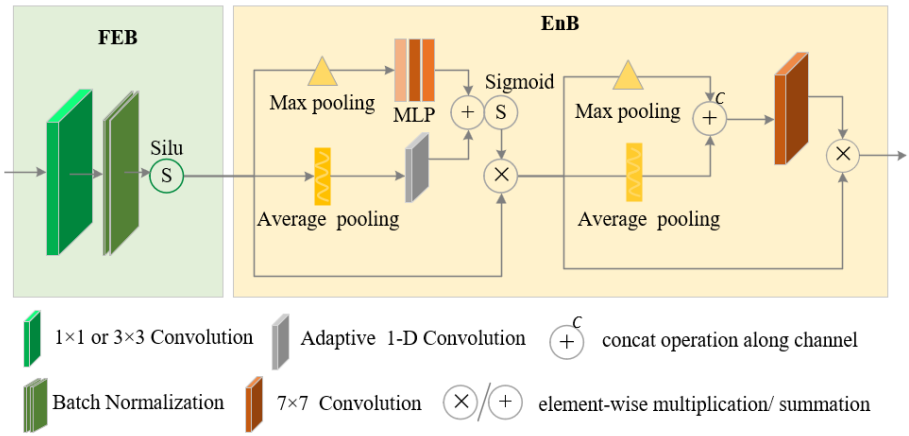


Figure 5. Structure of AFECM.

3.3. Feature Expression Enhancement Module

Although YOLOv5 constructs multiple feature maps to detect objects of different scales, each feature map is still expected to detect the objects of the different scales in a certain range. Downsampling is always utilized to expand the received field of a feature map to cover the relatively large objects. However, it results in the loss of features. Comparing with the relatively large objects, the features of the relatively small objects in the same scale range are more possible to be lost. Once the critical features of these small objects are lost, they could be probably missed by the corresponding feature map, which limits the ability of the feature map to detect objects of the scales in a certain range. In order to enlarge the received field to cover large objects without missing small objects, we introduce a feature expression enhancement module (FEEM) in this work.

$$f_{EBEF}(F_i) = f_{FEB}(f_{FEB}(F_i)) + f_{FEB}(F_i) \tag{6}$$

$$\begin{cases} f_{RE}(F_i) = f_u(f_u(f_u(f_u(F_i)))) \\ f_u(F_i) = f^{1\times1}(f^{3\times3}(f^{1\times1}(f_p(F_i)))) \\ f_p(F_i) = f^{3\times3}(f^{1\times1}(F_i)) \end{cases} \tag{7}$$

Figure 6 shows the structure of FEEM. According to the figure, FEEM consists of two parts: enhanced feature extraction block (EFEB) and received field expansion block (RFEB). EFEB is implemented based on CSP2_1 in Figure 1. Given an input feature map-F_i, it generates a new feature map according to Equation (6), where f_{FEB} is calculated according to Equation (4). RFEB is implemented based on DE. It takes the output feature map of EFEB as the input feature map, and generates a new feature map with a larger received field according to Equation (7). Based on Equations (6) and (7), FEEM generates a feature map with a stronger feature expression ability according to Equation (8).

$$f_{FEEM}(F_i) = f_{RE}(f_{EBEF}(F_i)) \tag{8}$$

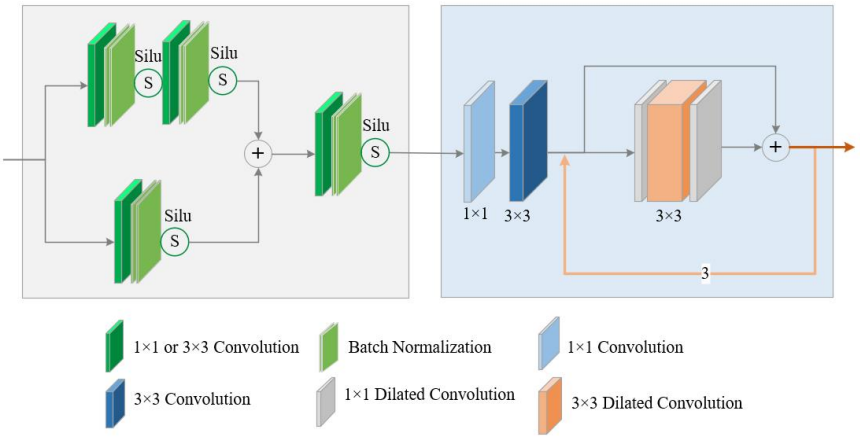


Figure 6. Structure of FEEM.

3.4. Special Detection Layer for Small Objects

The backbone generates four feature maps of four different resolutions, that is, 160 × 160, 80 × 80, 40 × 40 and 20 × 20. The feature map of a high resolution contains more details than the feature map of a low resolution. Details are very important for detecting small objects. However, in the neck, only the three feature maps with the least details are utilized, while the feature map with the most details, i.e., the feature map of the highest resolution, is not used. To improve the performance in detecting small objects, we exploit the feature map to create a detection layer special for small objects.

$$F_{final} = f^{1\times1}(f_{FEEM}(f^{+}(F_1, F_2^{''}))) \tag{9}$$

Here, we illustrate the detection layer with Figure 1. We upsample $F_2^{'}$ and merge the upsampled result with F_1 to create $F_1^{'}$. After that, we utilize a FEEM to process $F_1^{'}$ and generate the final feature map for detecting small objects. Equation (9) shows the details to calculate the final features. In the equation, $F_2^{''}$ denotes the upsampling result of $F_2^{'}$, and f^{+} describes the concat operation. The area surrounded by the red dotted lines in Figure 3 shows the detection layer.

3.5. Prior Box Generation with K-Mediods

The K-means algorithm is utilized by YOLOv5 to generate prior boxes. It calculates the centroid of a cluster by averaging all the points in the same cluster, which inevitably increases the error between the computed centroid and the true centroid, especially in the presence of outliers. Unlike the K-means algorithm, the K-Mediods algorithm calculates the sum of the distances between a point and all the other points in the same cluster, and takes the point with the smallest sum of distances as the centroid of the cluster. It can provide more stable results than the K-means algorithm. Therefore, it is exploited to replace the K-means algorithm to generate prior boxes in this work. Table 1 shows the prior boxes generated to match each feature map.

Table 1. Prior boxes generated by K-Mediods.

Feature Map	Receptive Field	Anchor
20 × 20	Large	(105, 42), (114, 88), (209, 159)
40 × 40	Medium	(25, 29), (48, 25), (52, 50)
80 × 80	Small	(10, 6), (15, 10), (29, 15)
160 × 160	Tiny	(3, 2), (4, 5), (7, 4)

4. Experiments

In this section, we present a detailed evaluation of YOLO-FF. First, we introduce the experimental environments and evaluation metrics. And then, we describe the dataset constructed in this work. Finally, we discuss the results of the ablation experiments and the comparison experiments.

4.1. Experimental Setup

Environments. All experiments are conducted on the same server equipped with an Intel Core i7-8700 CPU, an NVIDIA TITAN Xp GPU and 12 GB of memory. The server is deployed with Ubuntu 20.04.4, Torch 1.11.0, CUDA 11.3 and Python 3.7 development environment.

$$AP = \int_0^1 P(R)dR = \sum_{K=0}^{N} P(K)\Delta R(K) \tag{10}$$

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1score = \frac{2 \times P \times Recall}{P + Recall} \tag{13}$$

$$P = \frac{TP}{TP + FP} \tag{14}$$

Evaluation metrics. We use five evaluation metrics, namely Average Precision (AP), mean Average Precision (mAP), Recall, F1 score, and Frame Per Second (FPS), to evaluate YOLOv5-FF from the perspective of detection accuracy and detection speed. AP is calculated according to Equation (10), where R is the horizontal coordinate of the PR curve, P is the vertical coordinate of the PR curve, K is the Kth point on the PR curve, N is the total number of points on the PR curve, and $P(K)$ represents the value of the vertical coordinate of the Kth point on the PR curve. mAP can be obtained by averaging the APs for each category as Equation (11) shows. In the equation, C is the total number of categories in the dataset, and AP_i is the AP of the ith category. Recall is calculated according to Equation (12), where TP and FN represent the number of the true positive objects and that of the false negative objects, respectively. F1 score is calculated according to Equation (13), where P is calculated according to Equation (14). In Equation (14), FP depicts the number of the false positive objects.

4.2. Dataset

We constructed a floating object dataset and named it as FODS-00-01. FODS-00-01 consists of 6240 images totaling 12.9 GB. All these images were captured by handheld devices with different resolutions or extracted from videos collected by the devices installed near a local river and lake. Figure 7 shows the number of the images in various resolutions. All these images contain a total of 21,891 floating objects in total. We calculated the ratio of the total number of pixels per floating object to the total number of pixels in the corresponding image, and depicted the cumulative distribution function (CDF) of the ratio in Figure 8. According to the figure, about 20% of the floating objects occupy no more than 0.0058% of the pixels in an image, which means that about 20% of the objects occupy no more than 24 pixels if the resolutions of images are 640 × 640.

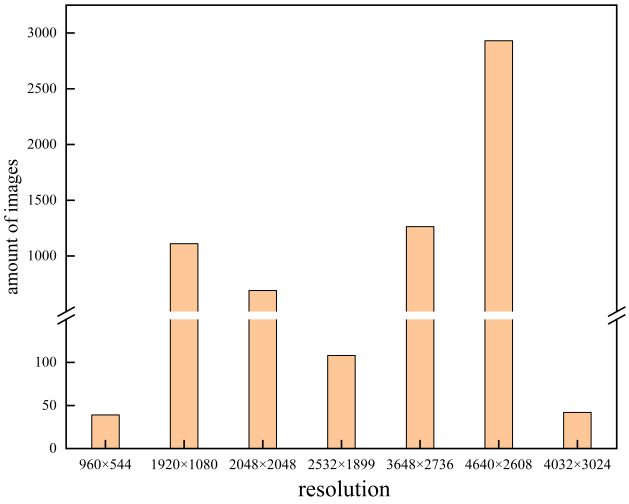


Figure 7. Number of the images in various resolutions.

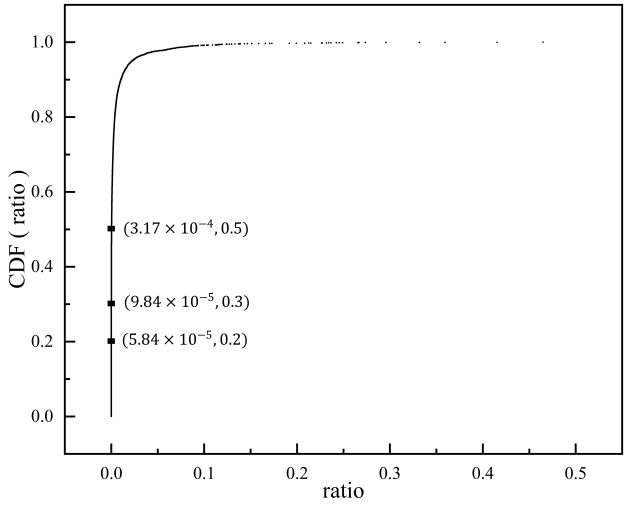


Figure 8. CDF of the pixel ratio of a floating object to the corresponding image.

FODS-00-01 includes floating objects such as abandoned garbage, rootless aquatic plants and deciduous leaves and branches. The abandoned garbage includes plastic bags, plastic bottles, foam blocks and other plastic products. They were labeled as “white_trash”. Rootless aquatic plants and deciduous leaves and branches were considered as the same class of floating objects. They were labeled as “plant_mixture”. Figure 9 shows some of the floating objects contained in FODS-00-01.

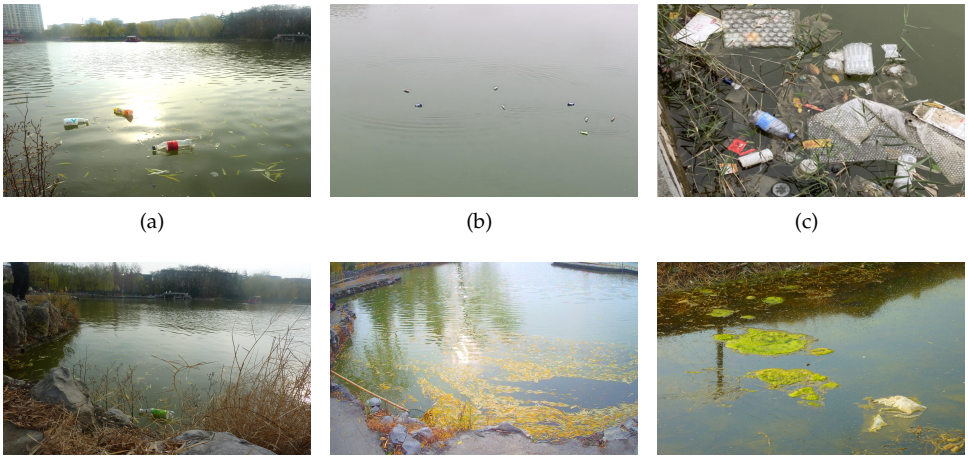


Figure 9. Some floating objects in FODS-00-01. (**a**–**e**) include only white_trash floating objects or plant_mixture floating objects, while (**f**) includes both classes of the floating objects.

In the following ablation experiments and comparison experiments, FODS-00-01 is divided into training set, test set and validation set according to the ratio of 8:1:1.

4.3. Ablation Study

To analyze the effectiveness of the improvements in this work, we perform ablation experiments and describe the results in Table 2. In the table, Model1 denotes the original version of YOLOv5s, while Model2, Model3, and Model4 denote the variant version with a new detection layer, the variant with a new detection layer and AFECM and the variant with a new detection layer, AFECM and FEEM, respectively.

According to the experimental results, Model2 achieved a performance improvement compared to Model1, indicating that the introduction of the new detection layer helped to improve the performance of YOLOv5s in detecting floating objects. Model3 and Model4 also achieved performance improvements separately compared to Model2 and Model3, which means the effectiveness of the collaboration of AFECM, FEEM and the detection layer in improving the performance of YOLOv5s.

Model5 represents the variant with a new detection layer, AFECM, FEEM and the prior box generation method with K-mediods algorithm. It achieved a performance improvement comparing with Model4, which indicates the effectiveness of all the improvements proposed in this work.

Table 2. Ablation results.

Models	New Detection Layer	AFECM	FEEM	Prior Boxes with K-Mediods Algorithm	mAP(%)0.5	Recall	F1-Score
Model1	×	×	×	×	78.00	71.80	79.88
Model2	✓	×	×	×	78.30	76.90	80.61
Model3	✓	✓	×	×	78.40	72.50	79.46
Model4	✓	✓	✓	×	78.80	74.90	80.24
Model5 (YOLO-FF)	✓	✓	✓	✓	80.80	79.60	82.35

4.4. Performance Comparison

We conduct experiments to evaluate the detection accuracy and speed of YOLOv5-FF by comparing YOLOv5-FF with several important object detection solutions including the two lightest versions of YOLOv8 [32], i.e., YOLOv8n and YOLOv8s, YOLOv7 [33], YOLOv5 [26], YOLOX [34], YOLOv4 [35], YOLOv3 [36], SSD [37], Faster-RCNN [38] and YOLOv5-CB [39]. Faster-RCNN is a two-stage detection solution, while all the other

solutions are single-stage detection solutions. To ensure the fairness of training processes, the training parameters of batch_size and the number of iterations are separately set to 16 and 300, while all the other training parameters are configured with their default values.

We evaluate the accuracy of YOLOv5-FF by four metrics, namely, AP, mAP, recall, and F1 score, and record the values of the four metrics for each solution in Table 3. According to the table, YOLOv5-FF detects plant_mixture objects with the highest AP. Compared with YOLOv8n, YOLOv8s, YOLOv7, YOLOX, YOLOv5, YOLOv4, YOLOv3, SSD, Faster-CNN and YOLOv5-CB, it improves the AP in detecting plant_mixture objects by 6.5%, 5.3%, 22.2%, 12.5%, 2.3%, 30.7%, 3.9%, 25.94%, 12.06% and 7.7%, respectively.

Table 3 also indicates that YOLOv5-FF detects white_trash objects in a higher AP than detecting plant_mixture objects. In the scenario of detecting white_trash objects, although YOLOv5-FF shows a slight degradation in AP compared to YOLOv3, it achieves a significant improvement in AP over all the other solutions. Concretely, it improves the AP in detecting white_trash objects by 16.9%, 16.6%, 9.2%, 8.2%, 3.2%, 22.7%, 52.42%, 50.3% and 2.3% comparing with YOLOv8n, YOLOv8s, YOLOv7, YOLOX, YOLOv5, YOLOv4, SSD, Faster-CNN and YOLOv5-CB, respectively.

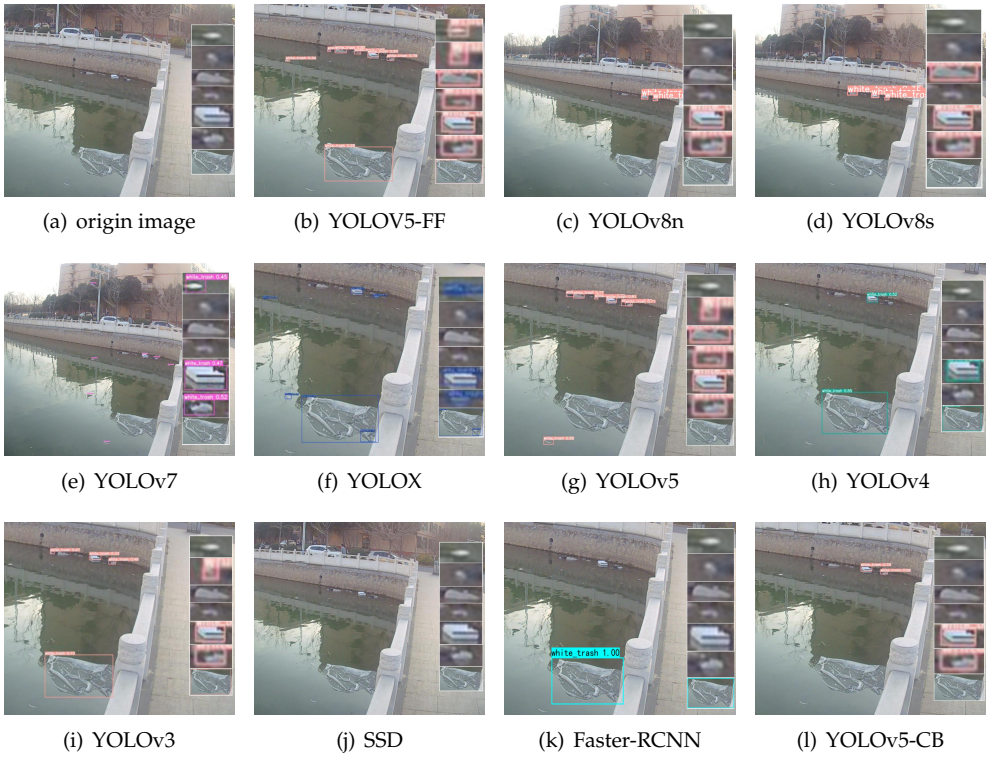
Table 3. Detection accuracy comparison.

Models	AP		mAP(%)0.5	Recall	F1-Score
	Plant_Mixture	White_Trash			
YOLOv5-FF	72.90	88.70	80.80	79.60	82.35
YOLOv8n	66.40	71.80	69.10	61.60	70.09
YOLOv8s	67.60	72.10	69.80	62.40	72.32
YOLOv7	50.70	79.50	65.10	59.70	67.48
YOLOX	60.40	80.50	70.43	-	-
YOLOv5	70.60	85.50	78.00	71.80	79.88
YOLOv4	42.20	66.00	54.10	64.70	45.00
YOLOv3	69.00	88.90	79.00	71.40	79.63
SSD	46.96	36.28	41.62	18.99	31.46
Faster-RCNN	60.84	38.4	49.62	54.52	60.32
YOLOv5-CB	65.2	86.4	75.8	70.10	78.31

AP can only evaluate the accuracy of YOLOV5-FF in detecting a certain type of floating objects, while mAP has the ability to evaluate the accuracy of YOLOV5-FF in detecting all types of floating objects. According to the results in Table 3, YOLOV5-FF detects floating objects with the highest mAP in all the solutions. Compared to YOLOv8n, YOLOv8s, YOLOv7, YOLOX, YOLOv5, YOLOv4, YOLOv3, SSD, Faster-CNN and YOLOv5-CB, YOLOV5-FF improves the mAP in detecting floating objects by 11.7%, 11%, 15.7%, 10.37%, 2.8%, 26.7%, 1.8%, 39.18%, 31.18% and 5%, respectively.

Recall and F1 score are also utilized to evaluate the accuracy of YOLOv5-FF. Except YOLOvX, all the other solutions obtain valid Recall and F1 score, and YOLOV5-FF obtians the highest Recall and F1 score among all these solutions. Compared to YOLOv8n, YOLOv8s, YOLOv7, YOLOv5, YOLOv4, YOLOv3, SSD, Faster-CNN and YOLOv5-CB, YOLOV5-FF improves the Recall and F1 score in detecting floating objects by 18% and 12.26%, 17.2% and 10.03%, 19.9% and 14.87%, 7.8% and 2.47%, 14.9% and 37.35%, 8.2% and 2.72%, 60.61% and 50.89%, 25.08% and 22.03%, and 9.5% and 4.04%, respectively.

Figure 10 shows the detection results of different solutions on the same image in FODS-00-01. Since most of the solutions performed better at detecting white_trash floating objects than detecting plant_mixture floating objects, we chose the image only including white_trash floating objects. According to the figure, only YOLOv5-FF can produce the same results as the ground truth in the presence of large floating objects and small even extra small floating objects. These results confirm the advantage of YOLOv5-FF over the comparison solutions in detecting floating objects of different scales.



(a) origin image (b) YOLOV5-FF (c) YOLOv8n (d) YOLOv8s

(e) YOLOv7 (f) YOLOX (g) YOLOv5 (h) YOLOv4

(i) YOLOv3 (j) SSD (k) Faster-RCNN (l) YOLOv5-CB

Figure 10. Detection result comparison of different solutions. YOLOv5-FF detected all the floating objects correctly, while all the other solutions have missed detection.

According to the results of the above comparison, we think YOLOv5-FF can detect floating objects with higher accuracy than the other solutions.

YOLOv5-FF is expected to detect floating objects in real time. Therefore, it is necessary to evaluate the detection speed, which is performed by comparing the FPS metrics of each of the solutions. Table 4 shows the FPS results of different solutions. According to the results, YOLOv5-FF can detect 78 frames per second on average, which indicates that YOLOv5-FF satisfies the requirement of detecting floating objects in real time.

Table 4. Detection speed comparison.

Models	FPS
YOLOv5-FF	78
YOLOv8n	129
YOLOv8s	113
YOLOv7	26
YOLOX	36
YOLOv5	131
YOLOv4	36
YOLOv3	53
SSD	31
Faster-RCNN	24
YOLOv5-CB	120

5. PFWD—An Intelligent Detecting System Based on YOLOv5-FF

We have developed an intelligent detection system for floating objects on water surfaces based on YOLOv5-FF and named it as PFWD. PFWD is programmed in C++. It adopts a C/S architecture. The server side is implemented based on Opencv. It has the ability to read frames from video files and cameras, and call trained models to detect and

classify floating objects. The client side is implemented with Qt. It supports video preview and detected result display.

PFWD can process multiple videos simultaneously in real-time depending on the multi-threading technique. It utilizes tab widget as the container to preview multiple videos and display the detected and classified results. A tab in the tab widget is used to display the video data from a camera or a video file and the corresponding detected results. Each tab is set up with a pop-up menu that provides options such as detection on and off, warning on and off, etc. To enable the warning function, at least one warning area should be drawn beforehand. As soon as the floating objects in the warning areas exceed a predefined threshold, an immediate warning message is issued. Figure 11 shows interface PFWD.

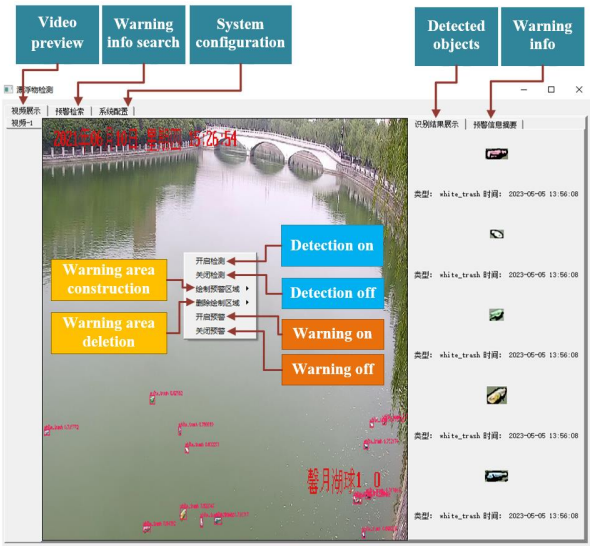


Figure 11. Interface and functions of PFWD.

6. Conclusions

In this work, we analyzed the performance problem of YOLOv5 in detecting floating objects in real time. To alleviate the performance problem, we proposed several improvements on YOLOv5. First, we proposed a hybrid attention mechanism which supported the interaction of channels over a long distance while preserving the direct correspondence between channels and their weights. Second, we constructed a feature extraction convolution module and a feature enhancement module based the proposed attention mechanism, and applied the two modules to the neck to optimize feature fusion. Third, we added a special detection layer to improve the performance in detecting small floating objects. Forth, we designed a prior box generation method to obtain the prior boxes matched our dataset. Finally, based on the above improvements, we implemented an intelligent floating object detection system.

Author Contributions: Conceptualization, X.Z. and J.L.; methodology, X.Z. and C.M.; formal analysis, X.Z., J.L. and Z.L.; investigation, X.Z., C.M. and J.L.; resources, X.Z. and Z.L.; data curation, X.Z. and C.M.; writing—original draft preparation, X.Z., C.M. and Z.L.; writing—review and editing, X.Z. and Z.L.; visualization, X.Z., C.M. and J.L.; supervision, X.Z. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by in part by the National Natural Science Foundation of China under Grant No. 61972134, Innovative and Scientific Research Team of Henan Polytechnic University under No. T2021-3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jin, S.J.; Kwon, Y.J.; Yoo, S.H. Economic Valuation of Reducing Submerged Marine Debris in South Korea. *Appl. Sci.* **2020**, *10*, 6086. [CrossRef]
2. Gall, S.C.; Thompson, R.C. The impact of debris on marine life. *Mar. Pollut. Bull.* **2015**, *92*, 170–179. [CrossRef]
3. Lin, F.; Hou, T.; Jin, Q.; You, A. Improved YOLO Based Detection Algorithm for Floating Debris in Waterway. *Entropy* **2021**, *23*, 1111. [CrossRef]
4. Biermann, L.; Clewley, D.; Martinez-Vicente, V.; Topouzelis, K. Finding Plastic Patches in Coastal Waters using Optical Satellite Data. *Sci. Rep.* **2020**, *10*, 5364. [CrossRef]
5. Zhang, X.; Gao, Q.; Yan, J.; Ji, D.; Luo, Y. Water quality affected by floating debris near the dam section of Three Gorges Reservoir. *Hupo Kexue/J. Lake Sci.* **2020**, *32*, 609–618.
6. Kumar, S.; Yadav, D.; Gupta, H.; Verma, O.P.; Ahn, C.W. A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management. *Electronics* **2020**, *10*, 14. [CrossRef]
7. Kylili, A. Identifying floating plastic marine debris using a deep learning approach. *Environ. Sci. Pollut. Res. Int.* **2019**, *26*, 17091–17099. [CrossRef] [PubMed]
8. Garcia-Garin, O.; Borrell, A.; Aguilar, A.; Cardona, L.; Vighi, M. Floating marine macro-litter in the North Western Mediterranean Sea: Results from a combined monitoring approach. *Mar. Pollut. Bull.* **2020**, *159*, 111467. [CrossRef]
9. Wolf, M.; Berg, K.V.D.; Garaba, S.P.; Gnann, N.; Zielinski, O. Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q). *Environ. Res. Lett.* **2020**, *15*, 114042. [CrossRef]
10. Yi, Z.; Yao, D.; Li, G.; Ai, J.; Xie, W. Detection and localization for lake floating objects based on CA-faster R-CNN. *Multimed. Tools Appl.* **2022**, *81*, 17263–17281. [CrossRef]
11. Gupta, H.; Verma, O. Normalization free Siamese network for object tracking. *Expert Syst.* **2022**, e13214.
12. Li, S.; Liu, S.; Cai, Z.; Liu, Y.; Chen, G.; Tu, G. TC-YOLOv5: Rapid detection of floating debris on raspberry Pi 4B. *J. Real-Time Image Process.* **2023**, *20*, 17. [CrossRef]
13. Armitage, S.; Awty-Carroll, K.; Clewley, D.; Martinez-Vicente, V. Detection and Classification of Floating Plastic Litter 391 Using a Vessel-Mounted Video Camera and Deep Learning. *Remote Sens.* **2022**, *14*, 3425. [CrossRef]
14. Yang, X.; Zhao, J.; Zhao, L.; Zhang, H.; Li, L.; Ji, Z.; Ganchev, I. Detection of River Floating Garbage Based on Improved YOLOv5. *Mathematics* **2022**, *10*, 4366. [CrossRef]
15. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, arXiv:2103.02907
16. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
17. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *IEEE Comput. Soc.* **2017**, *abs/1612.03144*, 2117–2125.
18. Huang, J.; Jiang, X.; Jin, G. Detection of River Floating Debris in UAV Images Based on Improved YOLOv5. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2022; pp. 1–8.
19. Liu, Y.; He, G.; Wang, Z.; Li, W.; Huang, H. NRT-YOLO: Improved YOLOv5 based on nested residual transformer for tiny remote sensing object detection. *Sensors* **2022**, *22*, 4953. [CrossRef] [PubMed]
20. Wan, F.; Sun, C.; He, H.; Lei, G.; Xu, L.; Xiao, T. YOLO-LRDD: A lightweight method for road damage detection based on improved YOLOv5s. *EURASIP J. Adv. Signal Process.* **2022**, *2022*, 98. [CrossRef]
21. Ren, M.; Wan, W.; Yu, Z.; Zhao, Y. Bidirectional YOLO: Improved YOLO for foreign object debris detection on airport runways. *J. Electron. Imaging* **2022**, *31*, 063047. [CrossRef]
22. Liu, H.; Sun, F.; Gu, J.; Deng, L. Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode. *Sensors* **2022**, *22*, 5817. [CrossRef]
23. Yan, B.; Li, J.; Yang, Z.; Zhang, X.; Hao, X. AIE-YOLO: Auxiliary Information Enhanced YOLO for Small Object Detection. *Sensors* **2022**, *22*, 8221. [CrossRef] [PubMed]
24. Jiang, X.; Hu, H.; Liu, X.; Ding, R.; Xu, Y.; Shi, J.; Du, Y.; Da, C. A smoking behavior detection method based on the YOLOv5 network. *J. Phys. Conf. Ser.* **2022**, *2232*, 012001. [CrossRef]
25. Alkoffash, M.S. Automatic Arabic Text Clustering using K-means and K-mediods. *Int. J. Comput. Appl.* **2012**, *51*, 5–8.
26. Ultralytics. YOLOV5-Master. 2021. Available online: https://github.com/ultralytics/yolov5.git/ (accessed on 22 December 2021).
27. Wang, W.; Xie, E.; Song, X.; Zang, Y.; Wang, W.; Lu, T.; Yu, G.; Shen, C. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8440–8449.
28. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

29. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You only look one-level feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13039–13048.
30. Zhou, R.; Li, X.; Jiang, W. SCANet: A Spatial and Channel Attention based Network for Partial-to-Partial Point Cloud Registration. *Pattern Recognit. Lett.* **2021**, *151*, 120–126. [CrossRef]
31. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the ICML'15: 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37; pp. 448–456.
32. Ultralytics. GitHub–Ultralytics/Ultralytics: YOLOv8. 2021. Available online: https://github.com/ultralytics/ultralytics.git (accessed on 10 June 2023).
33. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
34. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
35. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
36. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
37. Berg, A.C.; Fu, C.Y.; Szegedy, C.; Anguelov, D.; Erhan, D.; Reed, S.; Liu, W. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
38. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
39. Qiao, G.; Yang, M.; Wang, H. A Detection Approach for Floating Debris Using Ground Images Based on Deep Learning. *Remote Sens.* **2022**, *14*, 4161. [CrossRef]