applied sciences

MDPI

Article

EYOLOv3: An Efficient Real-Time Detection Model for Floating Object on River

Lili Zhang [1], Zhiqiang Xie [1], Mengqi Xu [1], Yi Zhang [1] and Gaoxu Wang [2,*]

[1] School of Computer and Information, Hohai University, Nanjing 211100, China
[2] State Key Laboratory of Hydrology–Water Resources and Hydraulic Engineering,
Nanjing Hydraulic Research Institute, Nanjing 210029, China
* Correspondence: gxwang@nhri.cn

Abstract: At present, the surveillance of river floating in China is labor-intensive, time-consuming, and may miss something, so a fast and accurate automatic detection method is necessary. The two-stage convolutional neural network models appear to have high detection accuracy, but it is hard to reach real-time detection, while on the other hand, the one-stage models are less time-consuming but have lower accuracy. In response to the above problems, we propose a one-stage object detection model EYOLOv3 to achieve real-time and high accuracy detection of floating objects in video streams. Firstly, we design a multi-scale feature extraction and fusion module to improve the feature extraction capability of the network. Secondly, a better clustering algorithm is used to analyze the size characteristics of floating objects to design the anchor box, enabling the network to detect objects more effectively. Then a focus loss function is proposed to make the network effectively overcome the sample imbalance problem, and finally, an improved NMS algorithm is proposed to solve the object suppressed problem. Experiments show that the proposed model is efficient in detection of river floating objects, and has better performance than the classical object detection method and the latest method, realizing real-time floating detection in video streams.

Keywords: real-time object detection; video streaming; multi-scale feature

check for updates

Citation: Zhang, L.; Xie, Z.; Xu, M.; Zhang, Y.; Wang, G. EYOLOv3: An Efficient Real-Time Detection Model for Floating Object on River. *Appl. Sci.* **2023**, *13*, 2303. https://doi.org/10.3390/app13042303

Academic Editor: Andrés Márquez

Received: 21 December 2022
Revised: 6 February 2023
Accepted: 7 February 2023
Published: 10 February 2023

1. Introduction

With the rapid development of industrialization and urbanization, the ecological environment has been seriously damaged, especially the water environment. A large number of floating objects such as plastic bottles, plastic bags, and aquatic plants appear on the water surface of many rivers and lakes. The presence of floating objects on the water surface will not only affect the aesthetics of the water body, cause water environment pollution, destroy the ecological balance, and even pose a threat to people's drinking water safety [1]. In addition, a large number of floating objects may affect equipment working in the river. For example, the propeller of a ship may be entangled in floating objects, and floating objects gathered in front of the barrage also have varying degrees of impacts on power generation, shipping, and other aspects [2].

In order to address the pressing water environmental problems, we should develop more intelligent methods to identify and clean up floating objects on the river. At present, the surveillance of the river floating in China is labor-intensive, time-consuming, and may miss something, so a fast and accurate automatic detection method is necessary [3–5].

With the improvement of science and technology, the performance of computer hardware, especially graphics processing unit, has been greatly improved, thus promoting the rapid development of pattern recognition, computer vision and other deep learning technologies. Therefore, object detection methods based on deep learning have been applied in many fields [6,7], such as face recognition [8,9], vehicle detection [10], automatic driving [11,12], and achieved good results. However, there is little research in the detection of floating objects on rivers and lakes. The purpose of this paper is to detect the floating objects in the images through deep learning, realize the intelligent monitoring of floating objects on rivers or lakes, so as to reduce the pollution of the water environment caused by the accumulation of floating objects, and lay a foundation for the realization of early warning of water pollution, the improvement of drinking water safety, and good water environment detection system in the future.

The early object detection methods, such as the combination of artificial design features and support vector machines, or background difference method, are mainly used for object detection. However, due to the complex environment and diverse forms of floating objects on river, the expression ability of artificial design features on floating object features is insufficient, and the robustness and generalization performance are poor, which cannot achieve the desired detection accuracy [13,14]. Deep learning methods show more advantages over prior methods in object detection. However, the object detection based on deep learning has high requirements on the quantity and quality of training dataset, while there are few open floating object dataset currently. Therefore, it is a challenging task to create a floating object dataset, and design a network model with high detection accuracy and fast detection speed.

In order to solve the above problems, we propose a model named EYOLOv3, an efficient real-time detection model for floating detection based on YOLOv3, and the contributions of this model can be summarized as follows:

i. In order to make full use of the information of different scale features of floating objects, a multi-scale feature extraction and fusion network is constructed to obtain more scale feature maps. At the same time, the high-level feature maps and low-level detection feature maps are fused to enrich the semantics of feature maps and improve the detection accuracy of our model;
ii. In view of the mismatching between the large difference of floating size and the fixed anchor box sizes in YOLOv3, an adaptive anchor box clustering method is proposed to design the anchor box sizes that conforms to the floating objects on the river;
iii. The cross entropy loss function is optimized, and the weight factor is added to adjust the weight of the different samples adaptively, so that the training is focused on the hard samples. While solving the problem of sample imbalance, the detection accuracy of our model is improved; and
iv. In order to reduce the wrongly suppressed problem, the non-maximum suppression (NMS) algorithm is improved. The prediction boxes whose overlapping area with the prediction box with the highest confidence is greater than the threshold will not be directly suppressed, instead, we will set the confidence for each prediction box base on the overlapping area, which can reduce false positive samples and effectively detect the floating objects with high overlapping simultaneously.

2. Related Works

2.1. Traditional Object Detection Methods

Generally, different objects have differences in color, texture, shape and other features. Traditional object detection methods start from these features, extract features based on manual-designed features and combine them with classifiers to achieve object detection. Detection methods have been studied for a long time. Different objects usually have differences in color, texture, shape, and other features. Common manual features include: Haar-like features [15], histogram of oriented gradient (HOG) features [16], Edgelet features and mixed features [17,18]. In 2000, Papageorgiou proposed Haar-like features, which describe the features of an object in an image using Haar wavelet. The advantage of Haar-like features is that it can effectively obtain the significant areas in the image, but it is less sensitive to the contour information of the object. It is vulnerable to the influence of object shape and image lighting. In 2005, Dalal proposed the HOG features based on the scale-invariant feature transform (SIFT) [19], mainly by calculating the gradient direction vector of the local area in the image and making it into a histogram to form the feature, which is distinctive and invariant to translation, rotation, scale, and other features. Then, based on HOG features, there are some improved features. The commonly used Local Binary Pattern (LBP) was proposed by Ojala [20], which has the advantages of gray scale invariance and rotation invariance. Girshick [21] combines the HOG feature with the LBP feature to achieve object detection when an object is occluded in the image. In the case of complex background, the detection effect of HOG features will be significantly reduced, but LBP features can filter out background noise, thus improving the detection effect. In general, all the models need to design the corresponding features for different detection objects, and the detection accuracy is not high for floating detection.

2.2. Deep-Learning-Based Methods

In 2006, Geoffrey Hinton, published a paper in Science, and proposed the idea of Deep Belief Networks [22] for the first time, which opened the prelude to the study of deep learning. In 2012, Hinton proposed AlexNet [23], which won the first place in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). This fully demonstrates the effectiveness of deep learning in image recognition tasks and its great potential in image processing.

With the development of deep learning, the performance of object detection is also improving. R-CNN was proposed in 2014 as a pioneering work in the field of object detection, which improved the accuracy of object detection greatly by combining the good automatic feature extraction and classification prediction of convolution neural network (CNN) with selective search. Subsequent work has improved this method too. In 2015, Ross Girshick proposed Fast R-CNN [24], which use the idea of SPP-Net [25] to propose ROI pooling layer, and a multitask loss function to incorporate both classification prediction and bounding box prediction, thus greatly improving the accuracy and speed of object detection. However, as Fast R-CNN uses a selective search strategy for region suggestions, the time required to generate candidate regions rises. Ren proposed the Faster R-CNN network [26], which designs a new Region Proposal Network (RPN), which hands over the task of region proposals to convolution network and improves the detection accuracy and speed. Since small objects have always been the difficulty of object detection, Yi et al. [27] proposed a Faster R-CNN algorithm with Class Activation for the detection and localization of floating objects on the water surface. From the results, the algorithm has strong adaptability when detecting small objects. However, it does not combine the statistical characteristics of floating objects in the region for further analysis. In the instance segmentation, the large receptive field pays more attention to the detail information, while the small receptive field pays more attention to the semantic information. Zhang et al. [28] proposed a Mask refined R-CNN network based on the object detection network Mask R-CNN. By constructing a feature pyramid network, the network can sum the forward and backward transmission of feature maps with the same resolution to achieve feature fusion, so as to balance the network performance. The above methods belong to two-stage object detectors and are less real-time than one-stage methods. In 2016, the end-to-end phase of YOLO [29] was proposed, abandoning the region proposals phase, so that object detection can achieve real-time detection speed, but detection accuracy is lower. In view of the problems of YOLO, YOLOv2 [30] and YOLOv3 [31] are proposed subsequently.

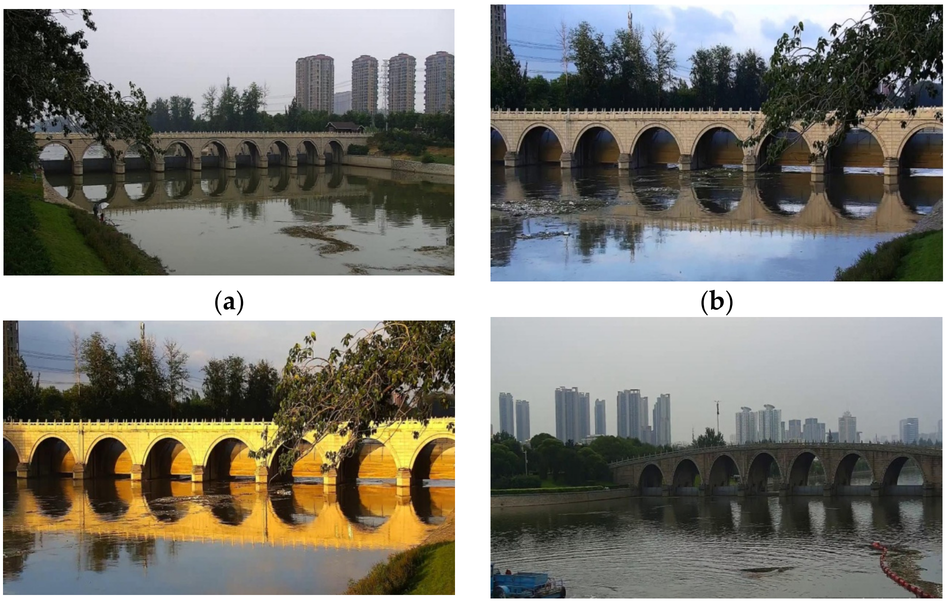
YOLOv3 has higher detection accuracy and detection speed, but is still not effective for floating objects on river which including small objects, and having dense object distribution and diverse shapes. Yang et al. [32] proposed a coordinated attention mechanism into the YOLOv5 model, which enabled the model to extract features quickly and efficiently, and improved the recall rate, average accuracy and F1 score on both public and private datasets. However, it also brings higher computational complexity and longer training time. Due to the real-time requirement, Lin et al. [33] proposed an improved YOLO for floating debris in waterway based on a one-stage algorithm. As the floating objects on river in China are different from the floating debris in waterway, we propose our method for the detection of floating objects on river, from the creation of the dataset to the design of the model, and to the end-to-end real-time floating detection in the video stream.

3. Proposed Method

YOLOv3 has become the preferred method for object detection, but the deep feature extraction network in YOLOv3 will reduce or even miss the features of smaller objects. While the floating objects on river have the various aspect ratios and the floating with the same size usually is dense distribution, so it is not easy to balance the detection accuracy and feature scale, and the detection accuracy of objects with dense distribution is lower. In addition, YOLOv3 is prone to gradient disappearance after it finishes continuous convolution operations, and the expression ability and generalization of feature extraction are not strong. In this paper, we will create the dataset of floating objects on river at first, then design an efficient real-time detector for floating detection, and carry out the end-to-end model training and real-time floating detection in video stream.

3.1. Dataset Creation of Floating Objects on River

We selected monitoring video images of a river gate and a flood diversion gate in Beijing in August 2018. The video frame rate is 25 frames per second. The floating objects mainly consist of water and grass. First, we picked out the video clips that contain floating objects and set an interval of 2 s to capture images. In order to ensure the validity of the floating object detection model and the convergence of the convolution neural network, we choose images with different lightings, as shown in Figure 1. We get 2065 images containing at least one floating object per image. We use rotation and clipping to expand the dataset, and the final dataset contains 4320 pictures, and the dataset is randomly divided into training set and test set with the ratio of 7:3.


(**a**) (**b**)

(**c**) (**d**)

Figure 1. Images in dataset. (**a**) Image with common light, (**b**) image with dark light, (**c**) image with bright light; (**d**) distance view image.

In order to facilitate the object detection model to read data, we use the constructed dataset as VOC2007 format. We used the annotation tool LabelImg to annotate all images in the dataset. Each image uses a rectangular bounding box to surround the objects, and the category information is set to flotage. After annotating, Labelmg automatically generates an XML file containing coordinate information and category information of the rectangular bounding box. Divide all file names into two randomly according to the ratio of 7:3, write them into the train.txt file and the test.txt file by line. All images recorded in train.txt are used for model training, and all images recorded in test.txt are used for model testing.

3.2. Training

First, the network is initialized with the model parameters of ImageNet pre-training; secondly, based on this, the EYOLOV3 network model is used to continuously tune the network according to the training dataset. The training process of the model mainly includes two stages: forward propagation and reverse propagation. The forward propagation mainly refers to the transfer from the input layer to the convolution layer, the convolution operation of the convolution kernel, the activation function operation, the pooling operation and the full connection calculation. The output result is calculated through forward propagation, and it is handed over to the back propagation algorithm to update the model parameters. The back-propagation algorithm will be combined with the gradient-based optimization algorithm during training, and the error or loss of the network will be transferred back to each network layer, and each layer of network will update the model parameters by calculating the gradient iteration. When the back-propagation reaches the input layer, the network will start the forward propagation calculation again and repeat the above process until the network converges or reaches the iteration number.

3.3. EYOLOv3: Optimized YOLOv3 Model

The EYOLOv3 network is shown in Figure 2. The backbone network structure of YOLOv3 is shown in Figure 3. The main improvement parts are as follows: (1) Build multi-scale feature extraction and fusion network to obtain feature of different scales of the floating, and more effectively fuse and re-extract shallow and deep features to improve the expression and generalization ability of network for feature extraction; (2) aiming at the variable object size, k-means++ is used to design the superior anchor box for floating objects on river; (3) the NMS is improved to reduce the missed detection of objects with high overlaps; (4) in order to improve the effect of nonlinear features and increase the expression ability and generalization of the network, the weight factor is added to adjust the weight of the different samples adaptively based on the cross-entropy loss function.

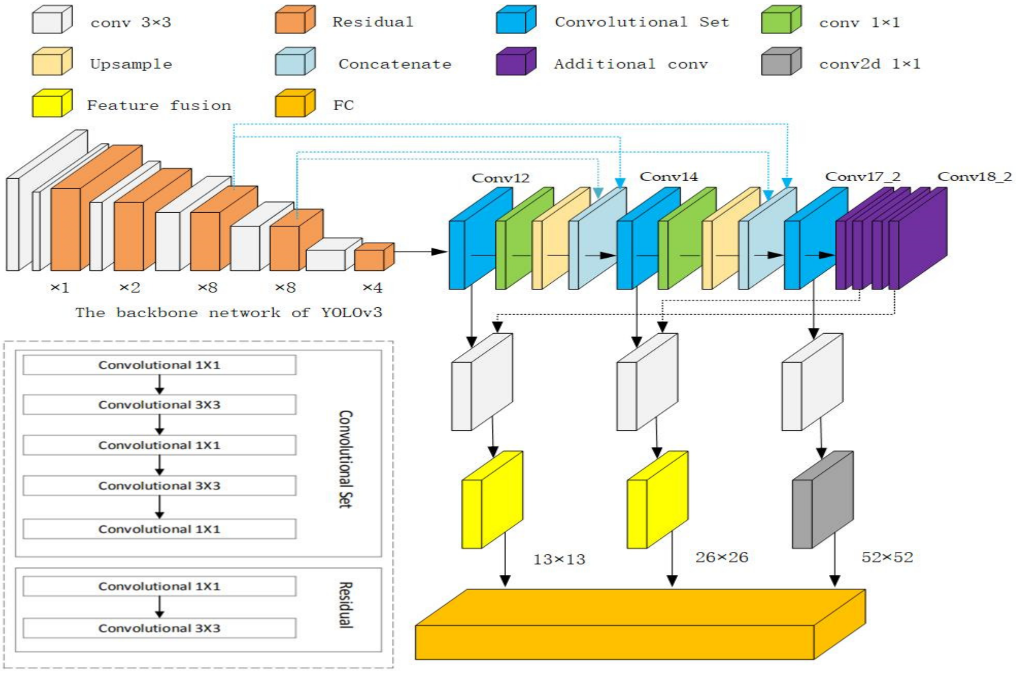
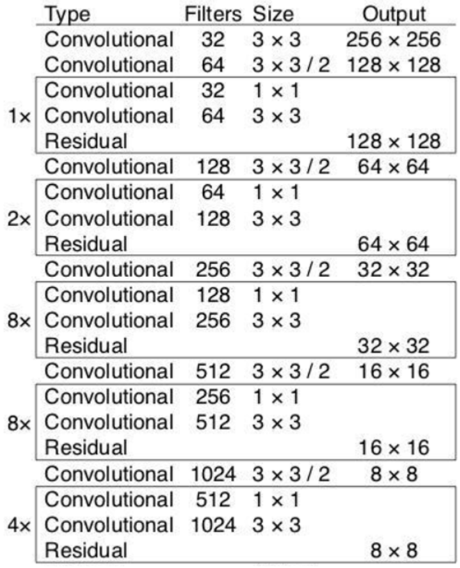


Figure 2. EYOLOv3 model for floating object detection.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8

Figure 3. Backbone network structure of YOLOv3.

3.3.1. Multi-Scale Feature Extraction and Fusion Network

The multi-scale feature extraction of EYOLOv3 uses the YOLOv3 backbone network and adds the auxiliary convolution layer at its end to obtain a variety of feature maps with different scales and resolutions. We use 3 × 3 and 1 × 1 convolution kernel, the activation function is Leaky Relu, and batch normalization (BN) is used to prevent overfitting after each 2D convolution calculation. The calculation formula of BN is shown in Formulas (1) and (2):

$$\hat{x}^{(k)} = \frac{x^{(k)} - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right]}} \tag{1}$$

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{2}$$

where $x^{(k)}$ denotes the linear activation of neurons in the current layer, γ and β is the regulatory parameter of neurons, E represents the mean within Batch, and *Var* is the variance.

The Feature Pyramid Network (FPN) can use feature fusion to achieve multi-scale object detection in multi-layer features. The lower-level features have higher resolution and contain more location and detail information. However, due to less convolutions, they have lower semantics and more noise. High level features have stronger semantic information, but their resolution is lower and their perception ability is poor. Hence, we combine the up-sampling of FPN and feature fusion of different layers, and propose a multi-scale jump connection method to achieve multi-scale feature fusion and improve the detection accuracy of multi-scale objects of the model.

In order to obtain more features and higher semantic, we add four new convolution (Cnv17_1, Conv17_2, Conv18_1, Conv18_2) layers at the end of feature extraction network, and a fusion branch is constructed to fuse the features output by the new convolution layers with multi-scale features to obtain feature maps of three sizes. One is the fusion of Conv12 layer and Conv18_2 layer, which is used to predict the three largest anchors and is suitable for large object detection; the second is the fusion of Conv14 layer and Conv17_2 layer, which is used to predict three anchors of medium size and is suitable for medium size object detection; the third is obtained by convolving the feature map of Conv16 layer, which is used to predict the minimum three anchors and is suitable for small object detection. After the down-sampling operations, the sizes of feature maps obtained by different convolutional layers are different, so it is impossible to directly fuse the features of convolutional neural networks. Take the fusion of Conv12 layer and Conv18_2 layer as an example, firstly, the feature map of the Conv18_2 layer is up-sampled; secondly, we use

L2-Norm to normalize the eigenvalue of the two layers to the same level, because Conv12 layer is shallow and its eigenvalue is large, and Conv18_2 layer is deep and its eigenvalue is small, so the direct connection is not feasible; thirdly, the two corresponding feature maps are merged to increase the number of channels; finally, through 1 × 1 convolution layer to fuse and reduce the dimension of the merged feature map, so the fused feature map contains both high-level feature semantic and low-level high-resolution detail information. The feature fusion module is shown in Figure 4. And Figure 5 shows the visualization process of the module. It can be seen from the Figure 5 that the fused feature maps have more information.

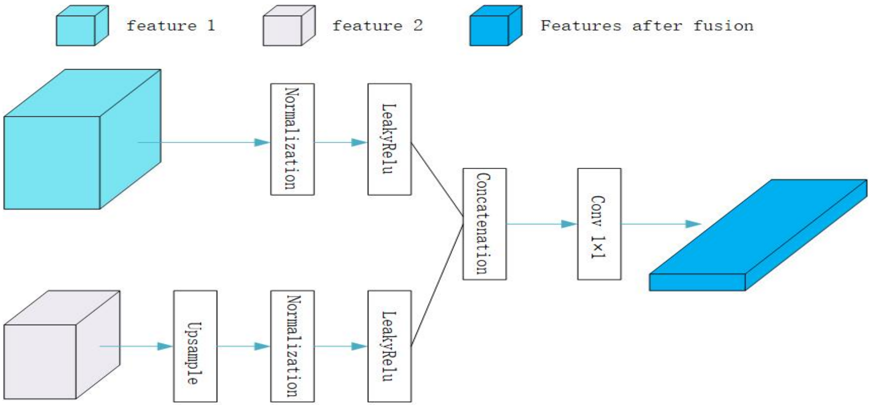


Figure 4. Feature fusion module.

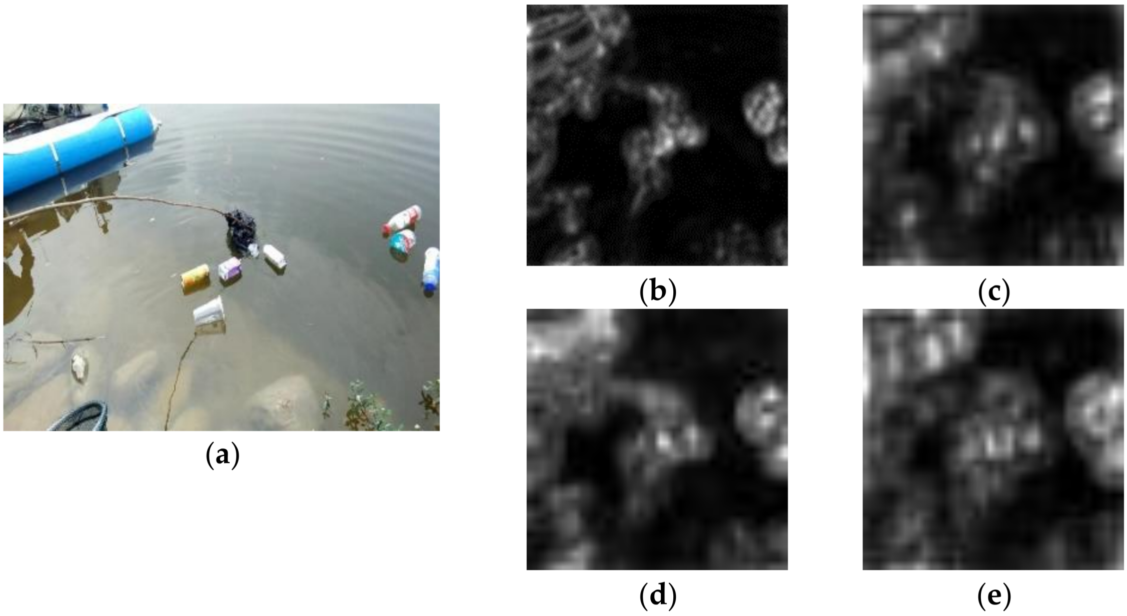


Figure 5. The visualization process of feature fusion module. (**a**) original image, (**b**) visualization of the feature in Conv14 layer, (**c**) visualization of the feature in Conv16 layer, (**d**,**e**) show the visualization of the fused feature.

3.3.2. Anchor Box Design

The YOLOv3 model used nine priori anchor boxes obtained by the dataset COCO, which are not suitable for floating objects on river. In the real scene, the shapes of the floating objects are varying and have the different aspect ratios from the objects in dataset COCO. We also manually set for anchor boxed may not be able to match the ground truth box of the floating, which affects the final detection as a result. Therefore, we design appropriate anchor boxes to improve the model faster and more accurate. The clustering algorithm is to maximize the IOU overlapping value between the anchor box and the ground truth box. The measurement function is shown in Formula (3):

$$d(box,\ centroid) = 1 - \text{IoU}(box, centroid) \tag{3}$$

Here, *box* is the ground truth box, and *centroid* is the cluster center. The value of the distance *d* decreases with the increase of the IOU value. According to the above method, K-means++ [34] is used to re-cluster the anchor boxes of the floating on river.

The steps of the K-means++ in detail are as follows:

1. Randomly select a sample from our dataset and take it as the initial clustering center;
2. Calculate the shortest distance between each data point and the existing clustering center;
3. Select a new data point as the clustering center. The selection principle is as follows: the greater the distance calculated in step 2, the greater the probability that it will be selected as the next clustering center. Then the next cluster center is determined by the method of polling scheme;
4. Repeat step 2 and step 3 until *k* clustering centers are selected;
5. Associate the each remaining sample point in the data set d with its nearest clustering center respectively and take it into the same clustering;
6. Calculate the mean value of each clustering, and take it as the clustering center of each group; and
7. Repeat step 5 and step 6 until the new clustering center is the same as the initial clustering center or less than a pre-set threshold.

The K-means++ algorithm is shown in Algorithm 1.

Algorithm 1 K-means++ algorithm

```
Repeat {
        for $i$ =1 to $m$ & $k$ = 1 to $K$
                $D^{(i)} := \|x^{(i)} - \mu_k\|^2$
                $p^{(i)} := \frac{D^{(i^2)}}{\sum_{i=1}^{m} D^{(i)^2}}$
        for $i$ = 1 to $m$
                $c^{(i)} := arg{\cdot}min_k\|x^{(i)} - \mu_k\|^2$
        for $k$ = 1 to $K$
                $\mu^{(i)} := \frac{\sum_{i=1}^{m} I\{c^{(i)}=k\}\cdot x^i}{\sum_{i=1}^{m} I\{c^{(i)}=k\}}$
}
```

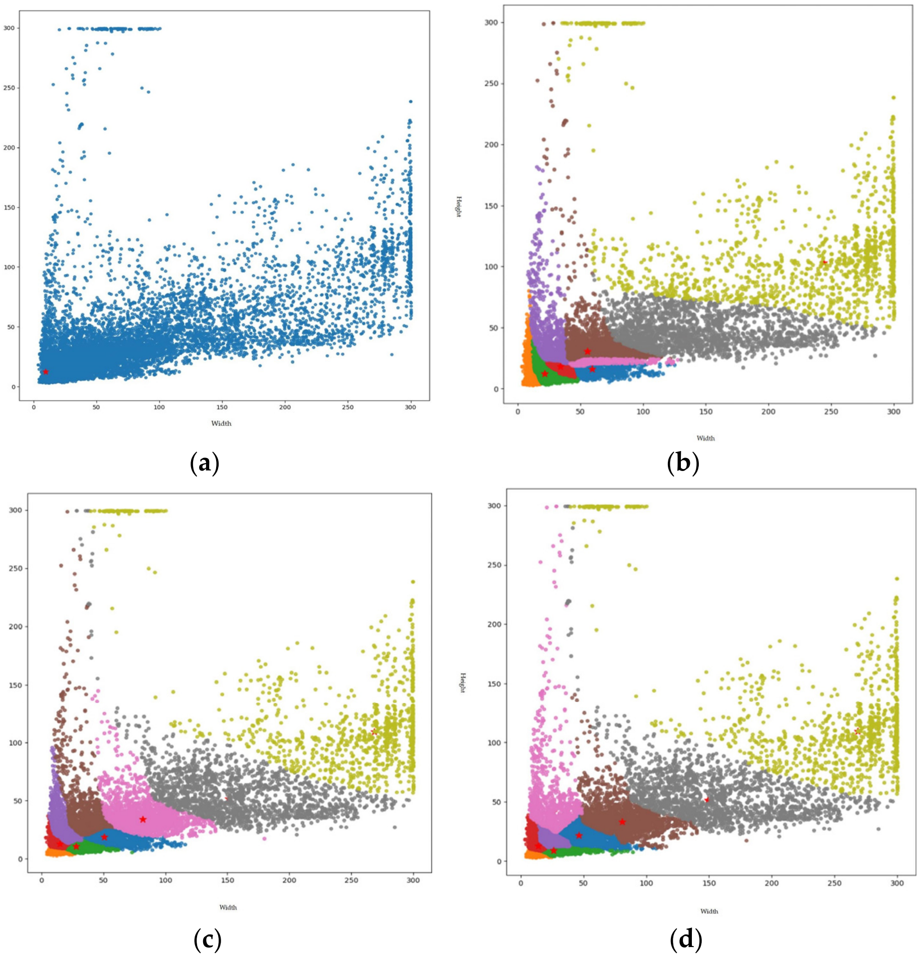
Where $D^{(1)},\ D^{(2)}, \ldots, D^{(m)}$ represents the shortest distance between each data point and the existing clustering center, and $P^{(1)}, P^{(2)}, \ldots, P^{(m)}$ represents the probability that each data point is selected as the next clustering center.

The label format of floating object dataset created in this paper is shown in Table 1. Here, object-class indicates the object category label, (*x*,*y*) indicates the center coordinate of the ground truth box, and width and height indicate the width and height of it, respectively.

Table 1. Dataset label format.

Dataset Label Format
<object-class> <*x*> <*y*> <width> <height>

We use the values of width and height to get the clustering centers of the floating objects, and set the coordinates *x* and *y* to 0 at initialization phase. Figure 6 shows the aspect ratio distribution and the clustering process of the ground truth boxes of floating, where the horizontal ordinate is the width of the ground truth boxes, and the ordinate is the height of the ground truth boxes.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Aspect ratio distribution and the clustering process of the ground truth boxes of floating. (**a**) Aspect ratio distribution, (**b**) clustering result of 10 iterations, (**c**) clustering result of 30 iterations, (**d**) final clustering result.

Considering the balance between calculation efficiency and accuracy of the network, we choose nine anchor boxes, as shown in Table 2:

Table 2. Anchor boxes.

Anchor Boxes
Anchor boxes = (14,8) (18,19) (28,21) (37,10) (48,25) (57,31) (91,33) (148,51) (262,109)

3.3.3. Improved NMS Algorithm

The traditional NMS only selects the prediction box with the highest confidence in an area, so the prediction boxes whose overlapping area with the prediction box with the highest confidence is greater than the threshold will be directly suppressed, then rise the missed detection. We set the confidence for each prediction box based on the overlapping area, which can reduce false positive samples and effectively detect the floating objects with high overlapping simultaneously. The improved NMS algorithm is as follows:

- First, sort the prediction boxes according to their confidence scores, and select the detection box M with the highest score;
- Traverse the remaining prediction boxes of the object. If the overlapping of the prediction box b_i and M is greater than the IOU threshold, the attenuation function is used to adjust the score of the prediction box. In order to prevent score faulting,

the Gaussian weighted attenuation function is used here. The calculation formula is shown in Formula (4):

$$S_i = S_i e^{-\frac{iou(M,b_i)^2}{\sigma}}, \forall b_i \notin D \quad (4)$$

where D represents the set of high score detection boxes that is reserved.

- Finally, select the box with the highest score again from the rest and set it to M, repeat the above, and successively reduce the scores of the prediction boxes with high overlapping, sort the final scores from high to low in sequence, and select all the prediction boxes with higher score than the threshold as the final.

The above process not only completes the non-maximum suppression to filter the redundant boxes, but also improves the detection accuracy of the model while there are multiple floating objects with higher overlapping area.

3.3.4. Balanced Focal Loss Function

The YOLOv3 does not make candidate box extraction, so it faces the imbalance between the foreground and background during training. To solve this problem, this paper uses a focal loss function replace the cross entropy loss function, and the binary classification cross entropy loss function is shown in Formula (5):

$$E(p,y) = \begin{cases} -log(p)\ y = 1 \\ -log(1-p)\ others \end{cases} \quad (5)$$

where $y \in \{\pm 1\}$ indicates whether it is a foreground, and $p \in [0,1]$ refers to the probability classified as a foreground, that is y = 1. The p_t is shown in Equation (6):

$$p_t = \begin{cases} p\ y = 1 \\ 1-p\ others \end{cases} \quad (6)$$

Therefore, the cross entropy function can be defined as shown in Equation (7):

$$CE(p,y) = CE(p_t) = -log(p_t) \quad (7)$$

The cross entropy loss function will always produce a loss value, even if the samples can be easily classified and identified. When the loss values are accumulated, the classes with fewer samples will be missed. In order to solve the category imbalance, a weighting factor $\alpha_t \in [0,1]$ is introduced into the cross entropy loss function. The definition of α_t is shown in Equation (8):

$$\alpha_t = \begin{cases} \alpha\ y = 1 \\ 1-\alpha\ others \end{cases} \quad (8)$$

Therefore, the balanced loss function can be optimized as shown in Equation (9):

$$CE(p_t) = -\alpha_t log(p_t) \quad (9)$$

This function will be greatly affected by the imbalance of foreground and background samples due to the negative samples dominating the gradient and loss value. In addition, we need to reduce the weight of simple samples in training process, in order to solve hard sample mining. Therefore, we use a weight factor $(1-p_t)^\gamma$, where γ represents an adjustable parameter. We obtained the final focus function as shown in Equation (10):

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t)\ \gamma \geq 0 \quad (10)$$

It can be concluded that the focal loss function can reduce the weight of simple samples, thus increasing the weight of hard samples. Therefore, the contribution of hard samples can be improved during model training, so as to solve the problem of imbalance between foreground and background samples and improve the detection accuracy of the model.

4. Experiments

4.1. Parameters Setting

All the Experiments in this paper is completed on Nvidia GeForce 1080 Ti, and the simulation platform is Ubuntu 16.04. We use the dataset we created in Section 3.1, and the parameter settings are shown in Table 3. In the experiment, the random gradient descent method is used to optimize 40,000 iterations of training. The initial learning rate is set to 0.0001, and the learning rate of 10,000 iterations per iteration is reduced by 10 times.

Table 3. Experimental parameters setting.

Parameter	Value
Batch Size	64
Image Size	416 × 416
Initial learning Rate	0.0001
Maximum Learning Rate	0.001
Minimum Learning Rate	0.000001
Momentum	0.9
Iterations	40,000

4.2. Evaluation Index

In order to evaluate the performance of the network and prove the effectiveness of our method, the following indicators are selected

- Precision (P) and recall (R). Precision refers to the ratio of the number of positive samples to the number of all samples; recall rate refers to the ratio of positive samples to the number of truth samples. The calculation formulas of precision and recall are shown in Equations (11) and (12):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

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

- where TP denotes the number of positive samples detected correctly, FP denotes the number of negative samples detected as positive samples, and FN denotes the number of positive samples detected as negative samples false negatives. Average Precision (AP) and mean Average Precision (mAP). AP measures the detection accuracy of the model. We can draw the P-R curves according to the precision and recall of the model, integrate the curve function, and calculate the area to obtain the average precision. The mAP is the mean value of APs of all the categories in one dataset. In this paper, since there is only one floating object, the AP is the same as the mAP.
- Frames Per Second (FPS). The frame rate refers to the number of pictures that can be detected per second. This indicator is used to evaluate the detection speed of the object detection of the network.

4.3. Experiments and Analysis

4.3.1. Experiments of Our Method

Firstly, Figure 7 shows the loss curve of our model in the training process. We selected test data with different environments to prove the robustness of our method, which is very important in real application, because the quality of the images in surveillance system is worse than the images in the public dataset. Figure 8 shows the detection results of floating objects under different light conditions and small object density. Our method can detect floating objects accurately, with high confidence, and has high robustness for floating objects in different scenes and lightings. As shown in Figure 8e,f, EYOLOv3 also has excellent detection effect in the case of dense small objects. Although there is a possibility of missed detection of small floating objects, compared with YOLOv3 model, it greatly improves the missed detection of small objects. The comparative experiments will be described later.

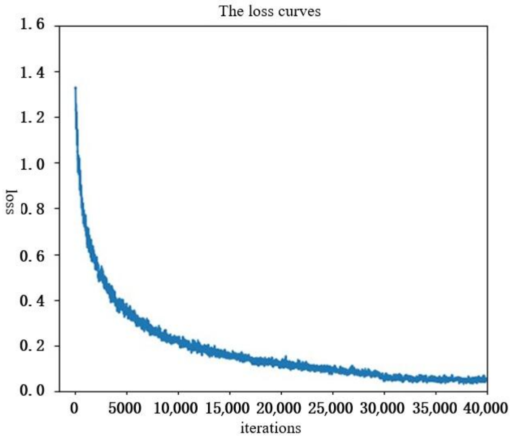
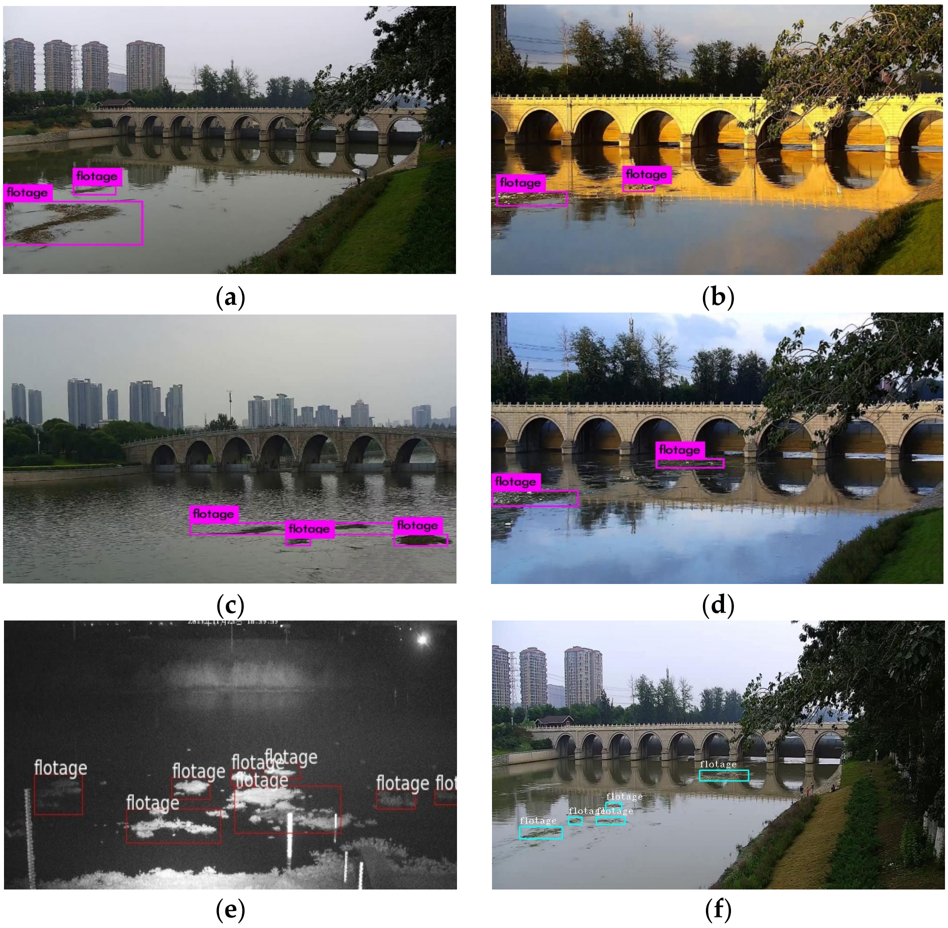


Figure 7. The loss curve of our model in the training process.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 8. Floating object detection of our method under different lightings. (**a**) Detection result with common light, (**b**) detection result with bright light, (**c**) detection result with dark light, (**d**) detection result with moderate light, (**e**) 0 and (**f**) detection result under dense small objects.

4.3.2. Experiment on Improved Strategies

In order to quantitatively analyze the impact of each improvement strategy in our method, the designs of the ablation experiments are shown in Table 4, and the experiments of different improvement strategies are displayed in the form of P-R curve in Figure 9, and the experiments are in Figure 10.

Table 4. The ablation experiments of the improved strategies.

Model Name	Balanced Focal Loss Function	K-Means++	Improved NMS Algorithm	mAP
Model 1	×	×	×	78.6%
Model 2	√	×	×	81.1%
Model 3	×	√	√	80.7%
Model 4	√	√	√	**82.3%**

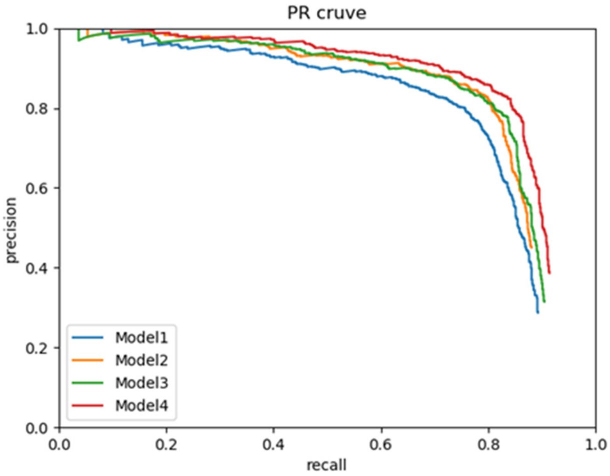
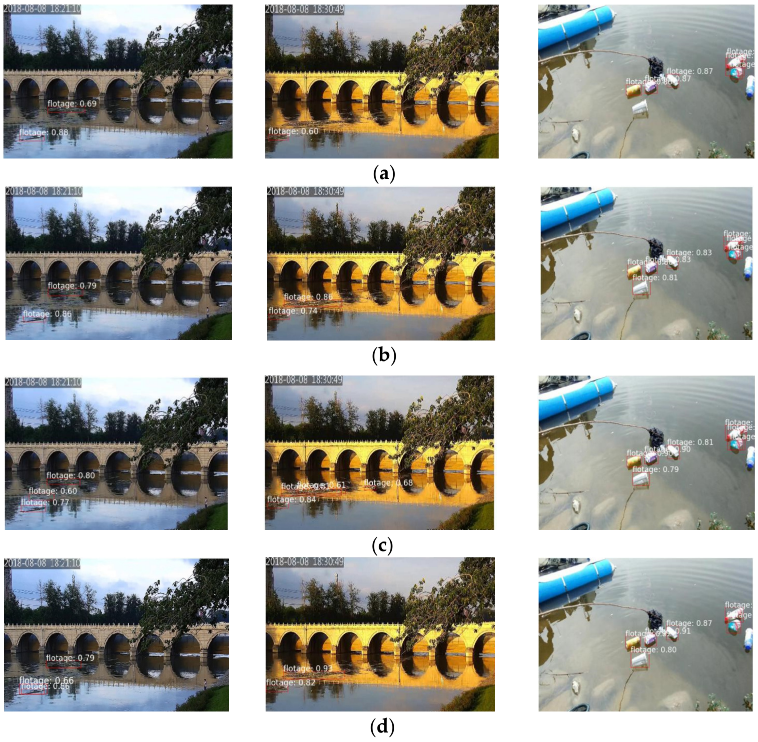


Figure 9. Comparison of PR curves of different improvement strategies.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 10. The ablation experiments with different improvement strategies. (**a**) Model 1, (**b**) Model 2, (**c**) Model 3, (**d**) Model 4.

It is seen from Table 4, the Model 1 does not use any improvement strategy, that is YOLOv3. Model 2 uses the focal loss function. Compared with Model 1, the detection accuracy of Model 2 has increased by 2.5%, indicating that the focus loss improvement strategy is effective. Model 3 introduces k-means++ clustering optimization algorithm and improved NMS algorithm, and the detection accuracy is improved by 2.1% compared with Model 1. It is verified that these two improved strategies can avoid missed detection and reduce the number of false positive samples, and improve the performance of the model. Model 4 adopts the focal loss function on the basis of Model 3, and the detection accuracy is 1.6% higher than Model 3, which indicates that the focal loss function effectively solves the imbalance problem of sample categories and improves the model performance. Model 4 integrates three improved strategies, and its detection accuracy is up to 82.3%, which is 3.7% higher than the YOLOv3, indicating that our proposed improvement strategy can effectively detect the floating objects on river automatically.

4.3.3. Comparison with Other YOLOs

We compared the detection performance of EYOLOv3 with other YOLOs, and Table 5 shows the detection performance of several models. The experiments show that, compared with YOLOv3, EYOLOv3 increases the mAP by 3.7%, and only increases the memory consumption of the graphics card; compared with YOLOv5, the mAP of EYOLOv3 is 1.8% higher, the detection speed of YOLOv5 is faster than that of EYOLOv3. However, the graphics card memory consumption of YOLOv5 is 1.54 times that of EYOLOv3, which will consume more computing resources, increase hardware costs, and have a negative impact on practical applications.

Table 5. Comparison with other YOLOs.

Model Name	Precision	Recall	mAP	FPS	Graphics Card Memory Consumption
YOLOv3 [31]	83.2%	80.4%	78.6%	35	**1218 M**
YOLOv5	85.8%	82.5%	80.5%	**40**	1925 M
EYOLOv3	**87.4%**	**85.7%**	**82.3%**	35	1250 M

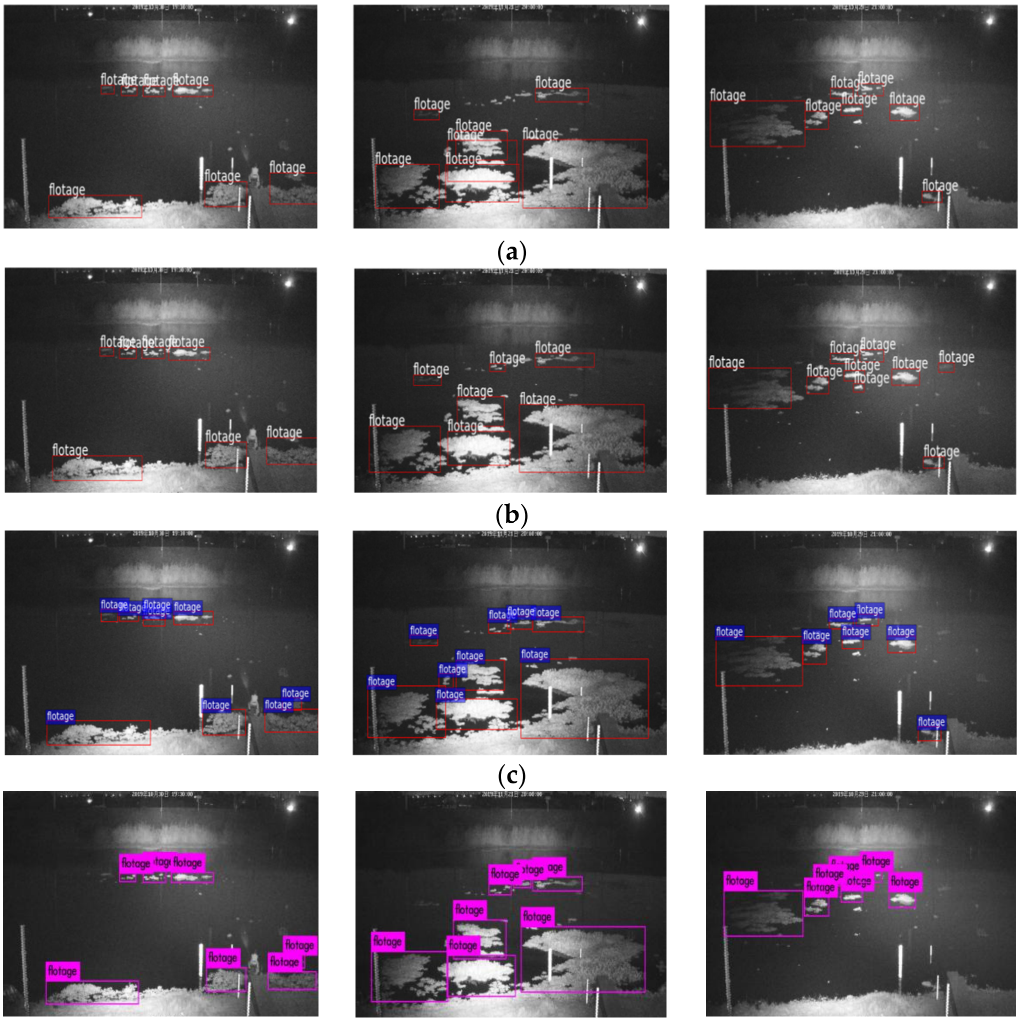
4.3.4. Comparison with Other Methods

We compare our model with other object detectors including improved RefineDet model we proposed in [35]. YOLOv3 uses the same parameter settings as our model in this paper, and Fast R-CNN and Faster R-CNN use the default parameter settings. Mask-Refined R-CNN, CA-faster R-CNN and Improved RefineDet train on the dataset we build until the model converges. We compare EYOLOv3 and the above methods on the same test dataset. The detection accuracy, the recall rate, the mAP, and FPS of the different detectors are shown in Table 6.

Table 6. Comparison with other models.

Model Name	Precision	Recall	mAP	FPS
Fast R-CNN [24]	79.8%	78.5%	75.3%	4
Faster R-CNN [26]	88.6%	84.9%	81.2%	13
YOLOv3 [31]	83.2%	80.4%	78.6%	**35**
EYOLOv3	87.4%	**85.7%**	**82.3%**	**35**
Mask-Refined R-CNN [28]	86.9%	85.1%	81.8%	16
CA-faster R-CNN [27]	**89.5%**	83.6%	82.0%	20
Improved RefineDet [35]	88.3%	85.0%	81.5%	28

It can be seen from Table 5, Fast R-CNN model, as a two-stage object detector, using the selective search strategy and high-level feature map, has the lowest detection accuracy and slowest detection speed, and missed detection for small objects is obvious. The detection accuracy of the Faster R-CNN network is slightly higher than Fast R-CNN and YOLOv3, and its detection speed is also faster than Fast R-CNN, but slower than YOLOv3. YOLOv3 is a single-stage detection model, so its detection speed is fast. However, YOLOv3 uses the traditional NMS algorithm, so there is a problem of missed detection of floating objects and no candidate box extraction process, so the detection accuracy is low. Our proposed model achieves the highest detection accuracy, which is 82.3%, and the detection speed reaches 35 FPS, which meets the real-time detection requirements of video streams. Both Mask-Refined R-CNN and CA-faster R-CNN are improved by adding auxiliary network on the basis of R-CNN, which improves the detection accuracy. However, due to its two-stage object detection network, the detection speed is slower than EYOLOv3. Compared with the Improved RefineDet we proposed in the previous work, EYOLOv3 uses the idea of skip connection for reference to fuse low-level features into multiple high-level features and uses soft-NMS algorithm to effectively reduce the missed detection rate of highly overlapping objects. Consequently, the final feature map generated for detection in EYOLOv3 contains rich detail information, which is conducive to small object detection. And the experiments show EYOLOv3 has higher detection accuracy and speed than Improved RefineDet. The experiments are shown in Figure 11.



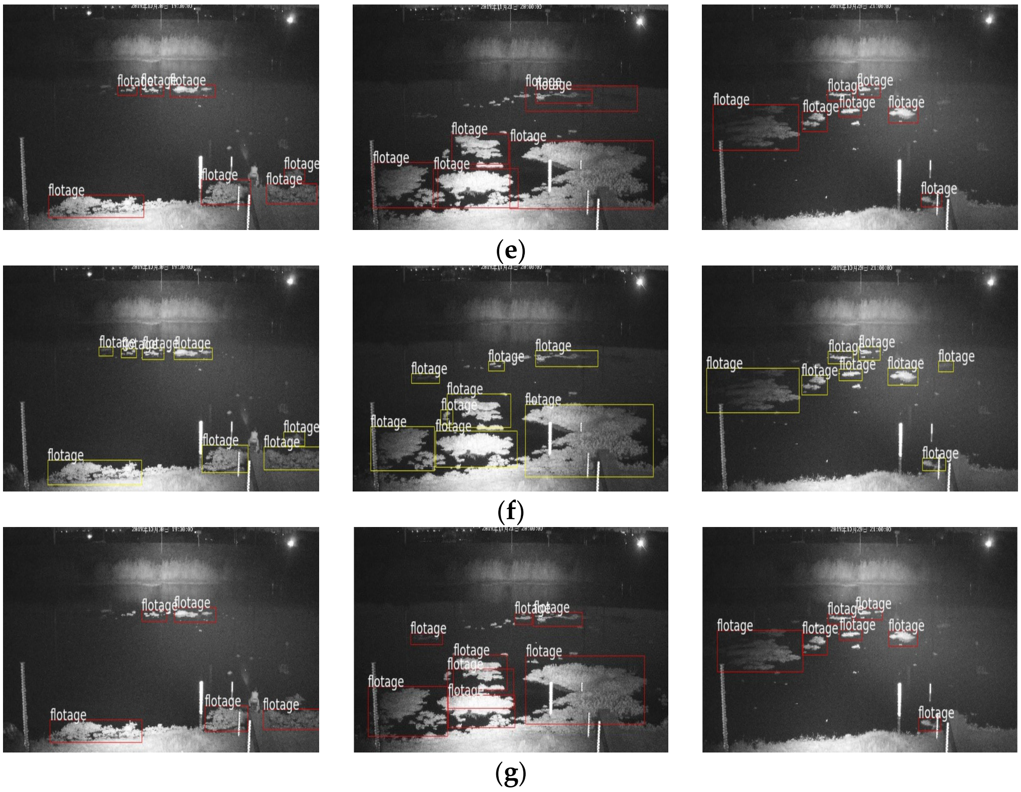
(**a**)

(**b**)

(**c**)

(**d**)

Figure 11. *Cont.*



(**e**)

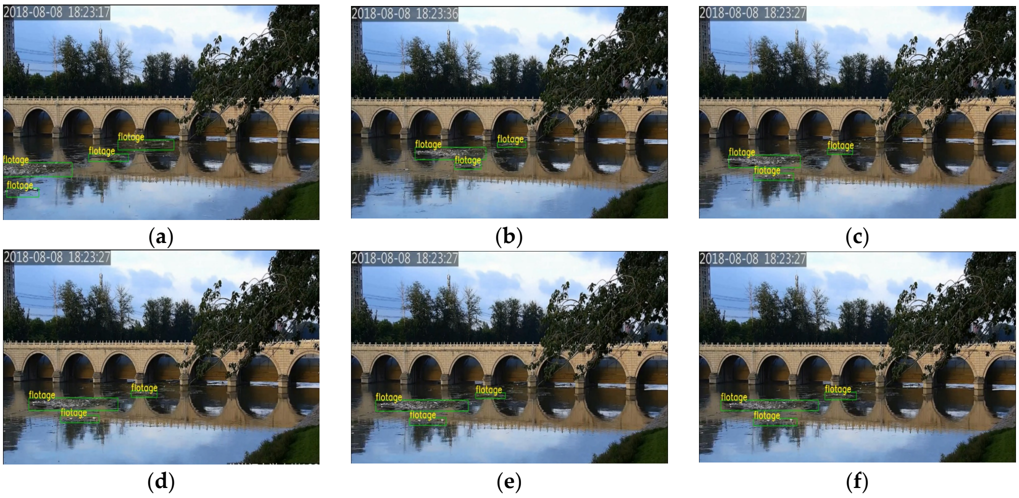
(**f**)

(**g**)

Figure 11. Comparison with other methods. (**a**) Fast R-CNN [24], (**b**) Faster R-CNN [26], (**c**) YOLOv3 [31], (**d**) EYOLOv3, (**e**) Mask-Refined R-CNN [28], (**f**) CA-faster R-CNN [27], (**g**) Improved RefineDet [35].

4.3.5. Real-Time Detection of Floating Objects in Video Stream

To further test the real-time performance of our method, a monitoring video containing floating objects is collected from the flood diversion gate, and our method is applied to detect the floating objects in the video. The detection results are shown in Figure 12. It can be seen from the Figure 12a,b that floating objects in each frame can be accurately detected. Also, Figure 12c–f show that floating detections of the four consecutive frames at 18:23:27 in 8 August 2018. Therefore, our method can realize efficient real-time floating detection in video stream.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 12. Real-time detection of floating objects in video stream of our method. (**a**,**b**) Show that floating detections of different shapes in the video stream. (**c**–**f**) are the floating detections of four consecutive frames (25 frames/second).

5. Conclusions and Prospect

In this paper, we propose an EYOLOv3 model. Considering the characteristics of floating objects, a deep multi-scale feature extraction and fusion network is designed to make full use of multi-scale feature information to achieve the complementary effect of different features and enhance the feature expression ability; in the training stage, the balanced Focal Loss Function is used to dynamically adjust the weight of positive and negative samples, so that the training can focus on hard samples and solve the imbalance problem of positive and negative samples. K-means++ is used to cluster the anchor boxes of the floating objects, and the anchor boxes redesigned match the floating better; improved NMS algorithm is used to reduce the missed detection of objects with high overlap. The experiments show that the EYOLOv3 is robust to the floating detection under different lightings. The detection accuracy of our model is improved to 82.3%, and the detection speed can reach 35 FPS, which meet the requirements of real-time floating detection.

In future work, we consider further improvement of NMS or use of better NMS algorithm, such as Syncretic-NMS algorithm proposed by Chu et al. [36]. The algorithm takes the traditional NMS as the first step, processes the boundary boxes obtained by the traditional NMS, judges the adjacent boundary boxes of each boundary box, and combines them with the corresponding boundary boxes. Syncretic-NMS algorithm has achieved excellent performance when applied to instance segmentation. In addition, combining multiple object detection networks is also the trend of future research. Munteanu et al. [37] proposed a deep learning model based on YOLO, SSD and EfficientDet mainstream object detection network. It is applied to the detection of sea mines on the surface and underwater, which provides a useful idea for the detection of floating objects on the surface.

Author Contributions: Conceptualization, L.Z. and Z.X.; methodology, L.Z. and Z.X.; validation, Z.X., M.X. and Y.Z.; resources, G.W.; data curation, L.Z. and G.W.; writing—original draft preparation, L.Z., Z.X. and Y.Z.; writing—review and editing, L.Z. and Z.X.; supervision, L.Z.; project administration, L.Z. and G.W.; funding acquisition, L.Z. and G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Nation Nature Science Foundation of China (grant nos.420755191, 91847301, 92047203, and 52009080); The science and technology innovation project of Guangdong Province Water Resources Department (No.2021-07).

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. Zhang, X.; Gao, Q.; Yan, J.; Ji, D.; Luo, Y. Water quality affected by floating debris near the dam section of three gorges reservoir. *J. Lake Sci.* **2020**, *32*, 609–618.
2. Cai, Y.; Tang, X.; Jiang, W. Summary of treatment measures for floating debris in river course. *J. Yangtze River Sci. Res. Inst.* **2013**, *30*, 84–89.
3. Moore, C.J. Synthetic polymers in the marine environment: A rapidly increasing, long-term threat. *Environ. Res.* **2008**, *108*, 131–139. [CrossRef] [PubMed]
4. Jung, R.T.; Sung, H.G.; Chun, T.B.; Keel, S.I. Practical engineering approaches and infrastructure to address the problem of marine debris in Korea. *Mar. Pollut. Bull.* **2010**, *60*, 1523–1532. [CrossRef]
5. Chen, C.L.; Liu, T.K. Fill the gap: Developing management strategies to control garbage pollution from fishing vessels. *Mar. Policy* **2013**, *40*, 34–40. [CrossRef]
6. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014. [CrossRef]
7. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]

8. Li, J.; Wang, Y.B.; Wang, C.G.; Tai, Y.; Qian, J.J.; Yang, J.; Wang, C.J.; Li, J.L.; Huang, F.Y.; Soc, I.C. DSFD: Dual Shot Face Detector. In Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
9. Dahl, G.E.; Yu, D.; Deng, L.; Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 30–42. [CrossRef]
10. Ševo, I.; Avramović, A. Convolutional neural network based automatic object detection on aerial images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 740–744. [CrossRef]
11. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.H.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
12. Li, P.L.; Chen, X.Z.; Shen, S.J.; Soc, I.C. Stereo R-CNN based 3D object detection for autonomous driving. In Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
13. Cherkassky, V. The nature of statistical learning theory. *IEEE Trans. Neural Netw.* **1997**, *8*, 1564. [CrossRef]
14. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999. [CrossRef]
15. Papageorgiou, C.; Poggio, T. A trainable system for object detection. *Int. J. Comput. Vis.* **2000**, *38*, 15–33. [CrossRef]
16. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005. [CrossRef]
17. Bo, W.; Nevatia, R. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In Proceedings of the Tenth IEEE International Conference on Computer Vision, Beijing, China, 17–21 October 2005. [CrossRef]
18. Watanabe, T.; Ito, S.; Yokoi, K. Co-occurrence histograms of oriented gradients for pedestrian detection. In Proceedings of the 3rd Pacific-Rim Symposium on Image and Video Technology, Tokyo, Japan, 13–16 January 2009. [CrossRef]
19. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999. [CrossRef]
20. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994. [CrossRef]
21. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
22. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
24. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
26. 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]
27. Yi, Z.R.; Yao, D.Y.; Li, G.J.; Ai, J.Y.; Xie, W. Detection and localization for lake floating objects based on CA-faster R-CNN. *Multimed. Tools Appl.* **2022**, *81*, 17263–17281. [CrossRef]
28. Zhang, Y.Q.; Chu, J.; Leng, L.; Miao, J. Mask-Refined R-CNN: A network for refining object details in instance segmentation. *Sensors* **2020**, *20*, 1010. [CrossRef]
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
30. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
31. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
32. Yang, X.S.; Zhao, J.Y.; Zhao, L.; Zhang, H.Y.; Li, L.; Ji, Z.L.; Ganchev, I. Detection of river floating garbage based on improved YOLOv5. *Mathematics* **2022**, *10*, 4366. [CrossRef]
33. Lin, F.; Hou, T.; Jin, Q.N.; You, A.J. Improved YOLO based detection algorithm for floating debris in waterway. *Entropy* **2021**, *23*, 1111. [CrossRef]
34. Arthur, D.; Vassilvitskii, S. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, LA, USA, 7–9 January 2007.
35. Zhang, L.; Wei, Y.; Wang, H.; Shao, Y.; Shen, J. Real-Time Detection of River Surface Floating Object Based on Improved RefineDet. *IEEE Access* **2021**, *9*, 81147–81160. [CrossRef]

36. Chu, J.; Zhang, Y.Q.; Li, S.M.; Leng, L.; Miao, J. Syncretic-NMS: A merging Non-Maximum Suppression algorithm for instance segmentation. *IEEE Access* **2020**, *8*, 114705–114714. [CrossRef]
37. Munteanu, D.; Moina, D.; Zamfir, C.G.; Petrea, S.M.; Cristea, D.S.; Munteanu, N. Sea mine detection framework using YOLO, SSD and EfficientDet deep learning models. *Sensors* **2022**, *22*, 9536. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.