

## Article

# A YOLOW Algorithm of Water-Crossing Object Detection

Shufang Xu <sup>1,2,\*</sup>, Hanqing Tang <sup>1,2</sup>, Jianni Li <sup>1,2</sup>, Longbao Wang <sup>1,2</sup>, Xuejie Zhang <sup>1,2</sup> and Hongmin Gao <sup>1,2</sup>

<sup>1</sup> School of Computer and Information, Hohai University, Nanjing 211100, China; 211307040014@hhu.edu.cn (H.T.); lijanni@hhu.edu.cn (J.L.); wlb@hhu.edu.cn (L.W.); xuejie\_zh@hhu.edu.cn (X.Z.); gaohongmin@hhu.edu.cn (H.G.)

<sup>2</sup> Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing 211100, China

\* Correspondence: xushufang@hhu.edu.cn

**Abstract:** Accurately identifying and locating water-crossing objects is of utmost importance for environmental protection. However, traditional detection algorithms often exhibit poor anti-interference performance and low detection accuracy in complex environments. To address these issues, this paper proposes a YOLOW algorithm based on a one-stage object detection algorithm for the automatic identification of water objects. The proposed algorithm incorporates two new modules, namely the SPDCS module and the SPPAUG module, to improve the model's performance. Specifically, the SPDCS module retains all information in the channel dimension, thereby enhancing the model's detection accuracy and recognition ability for water-crossing objects. The SPPAUG module performs multiscale feature fusion, which further improves the model's detection accuracy and recognition ability. Moreover, the C2f module is introduced from YOLOv8 to increase the detection speed. Experimental results on a water-floating object dataset demonstrate that the improved YOLOW model outperforms the standard YOLOv5s algorithm, especially in water-crossing object detection. This research has significant implications for environmental monitoring and protection.

**Keywords:** object detection; water-crossing objects; deep learning; YOLOv5



**Citation:** Xu, S.; Tang, H.; Li, J.; Wang, L.; Zhang, X.; Gao, H. A YOLOW Algorithm of Water-Crossing Object Detection. *Appl. Sci.* **2023**, *13*, 8890. <https://doi.org/10.3390/app13158890>

Academic Editor: José Salvador Sánchez Garreta

Received: 30 June 2023

Revised: 18 July 2023

Accepted: 22 July 2023

Published: 2 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Object detection is a pivotal task within the field of computer vision, focusing on the identification and localization of specific objects in images or videos. Its significance extends to domains like security and environmental protection [1,2]. With the increasing attention on water environmental protection, more and more research on object detection has turned to water-crossing object detection. Floating debris on the water surface is a serious concern that poses a threat to ecosystems and requires urgent attention. By accurately identifying and localizing floating objects on the water surface, object detection establishes a foundation for real-time monitoring and management, thereby fostering the safeguarding and restoration of the water environment.

In the research on traditional object detection algorithms for detecting water-crossing objects, the following problems still exist:

As shown in Figure 1a, images are often affected by multiple factors that can negatively impact object detection. Constantly various lighting conditions can cause objects to exhibit different features, such as shadows and highlights. Due to occlusion between objects, it is difficult to distinguish overlapping objects. Differences in object sizes and poses can further complicate object detection, as objects may appear differently depending on their orientation and distance from the camera. Figure 1b highlights another common challenge faced by object detection algorithms: low resolution. In many cases, the areas of the image that are relevant to object detection may be poorly resolved, making it difficult to identify objects accurately. Blurring caused by camera shake further reduces the clarity of the image and makes it harder to detect objects. Figure 1c illustrates the impact of water ripples and

reflections on object detection accuracy. These factors can introduce significant noise into the image, making it harder to distinguish between objects and their surroundings.



**Figure 1.** Challenges in detecting water-crossing objects.

Radar [3,4] and infrared [5] detection are commonly used techniques for detecting objects related to water. Compared with RGB imaging, these methods have limitations. Radar detection relies on reflected echo signals, while infrared detection relies on the thermal radiation emitted by objects. Both technologies lack the ability to provide visual information, such as the shape and color of objects. In addition, radar and infrared detection may not provide high-resolution images and may be challenged to distinguish different types of objects in water. Environmental factors such as weather conditions and water turbulence can also affect their accuracy. Another limitation of these technologies is the high equipment cost compared with conventional RGB cameras.

Since the introduction of the AlexNet [6] model, CNNs (convolutional neural networks) have shown remarkable performance in various computer vision tasks. In image classification, several famous CNN models have emerged, such as AlexNet, VGGNet [7], and ResNet [8]. In terms of object detection, a series of excellent models have been developed, including R-CNN series [9–14], YOLO series [15–24], SSD [25], EfficientDet [26], and FCOS [27,28]. These algorithms have advantages such as automatic learning, good feature representation, and high-precision detection.

However, existing object detection algorithms are not effective in extracting features relevant to water-crossing environments and detecting objects in low-resolution or blurry areas. To address these challenges, a new algorithm called YOLOW is proposed in this paper. YOLOW builds upon YOLOv5 by introducing an SPDCS (Space-to-Depth-Conv-Se) module, proposing an SPPAUG module based on the YOLOv5 SPPF module, and introducing a C2f module from YOLOv8. The contributions of this paper can be summarized as follows:

1. A new module called SPDCS has been designed based on the existing SPD (Space-to-Depth) module. Compared with the SPD module, the SPDCS module downsamples the original features and performs convolution and attention operations on the original features to further obtain more channel count features. In this way, a balance is achieved between obtaining new features and preserving initial features, thus improving data analysis and processing capabilities. Compared with the original YOLOv5 algorithm, using this innovative module can achieve five times the number of channels. By using the SPDCS module, YOLOW can extract more information features from low-resolution and blurry images in water-crossing environments. The advantage of this module is that it can retain original features and extract new features, thereby obtaining richer information. In this way, we can more accurately classify and identify objects, thereby improving data processing and analysis capabilities.
2. The SPPAUG module is an improvement on the SPPF module. The SPPF module is a spatial pyramid pooling module which can pool features at different scales, thus improving the expression ability of the model. SPPAUG has added more convolutional layers and a staggered network structure on top of the SPPF module. By

connecting features across multiple dimensions, more diverse features can be extracted. Compared with the original SPPF module, the SPPAUG module can more comprehensively capture information features in images, thereby improving the classification and recognition capabilities of the model. By adding the SPPAUG module, we can enable the model to better adapt to different scenarios and tasks, thereby improving its robustness and generalization ability.

3. In order to obtain more gradient flow information, this paper introduces the C2f module in YOLOv8. The function of this module is to enhance gradient flow by combining additional convolutional layers and pooling operations. By doing so, the C2f module enables the model to capture finer-grained details and better distinguish similar features. By integrating C2f modules, YOLOv8 can obtain more comprehensive and detailed information.

By introducing new modules and expanding existing ones, we have significantly enhanced the feature extraction ability of neural networks. These modules enable the model to capture more diverse and informative features, thereby improving the performance of various data analysis and processing tasks. Especially in complex water environments, our model can effectively handle objects of different scales and solve the challenge of object detection. Compared with traditional detection algorithms, our method provides a promising solution for accurate detection and recognition in aquatic scenes. Our method demonstrates the potential for improving the performance and reliability of object detection systems in water-crossing applications.

The remainder of this paper is organized as follows. In Section 2, we provide a brief introduction of relevant background knowledge. Section 3 describes the detailed pipeline of the proposed YOLOv8 method. The validity and superiority of the proposed method are demonstrated through extensive experiments and analysis in Section 4. Finally, we summarize the work presented in this paper in Section 5.

## 2. Related Work

### 2.1. Real-Time Object Detectors

The emergence of the convolutional neural network (CNN) has brought revolutionary changes to the field of object detection. In object detection tasks, CNN models are usually divided into two types: one-stage and two-stage. The one-stage detector directly detects locations with dense sampling, skipping the region recommendation step. This method typically uses a single network to predict the location and category of an object. The two-stage detector first provides rough region suggestions and then uses a head network to refine them to obtain the accurate position and category of the object. This method typically involves two stages of operation, namely generating region suggestions and classifying and regressing region suggestions.

Generally speaking, a one-stage object detection model uses a CNN-based backbone network to extract image features and uses detection heads to predict the category and bounding box of each object. In addition, in order to detect objects of different sizes, additional layers will be added to combine features of multiple scales, thereby generating feature representations with rich semantics. Currently, real-time object detectors based on YOLO and FCOS are in a leading position, including models such as YOLOv5, YOLOv7 [23], and YOLOv8 [24].

Typically, a two-stage object detector consists of two main steps: generating candidate regions and classifying and refining bounding boxes. In the first stage, effective techniques such as selective search, R-CNN series, or edge boxes are used to generate candidate regions. These candidate regions typically have multiple scales and aspect ratios to represent areas where objects may appear.

Usually, a one-stage detector is faster than a two-stage detector. These one-stage detector models need to have faster and stronger network architecture, more effective feature integration [29–35], more accurate detection methods [27,28,36], more robust loss function [37–42], more effective tag allocation methods [43–47], and more effective training

strategies to become the most advanced real-time object detectors. Therefore, a one-stage detector is more suitable for real-time object detection.

YOLOv5 is an object detection algorithm based on deep learning, whose core idea is to divide the entire image into several grid units, each of which is responsible for detecting the presence of objects and predicting information such as object position, category, and confidence. Compared with previous versions, YOLOv5 further improves detection performance and speed by improving network structure and training strategies.

YOLOv5 uses CSPDarknet as the backbone network. It is a lightweight and efficient network structure that can extract high-level semantic information from images. YOLOv5 also uses specific anchor boxes to improve detection accuracy. These anchor boxes are predefined rectangular boxes with different sizes and aspect ratios, which are used to detect objects of different sizes and shapes.

In addition, YOLOv5 also adopts various techniques for enhancing feature representation, such as adaptive convolution and multiscale training strategies. Adaptive convolution is a technique that can automatically adjust the size of convolution kernels, enabling the extraction of finer features on objects of different scales. The multiscale training strategy is to train on images at different scales, making the network have better scale invariance and generalization ability.

YOLOv5 has achieved remarkable results in various application domains, including object detection, pedestrian detection, and vehicle detection. It possesses low computational complexity and high real-time performance, with great potential for practical applications.

This paper proposes a water-crossing environment object detection algorithm called YOLOW based on the YOLOv5 algorithm. It combines the special object morphology and lighting conditions of the water-crossing environment, as well as the specific features of related objects, further improving detection performance compared with YOLOv5.

## 2.2. Multiscale Object Detection

Multiscale object detection poses a significant challenge, as it requires detecting objects of various sizes. Two common approaches for addressing this challenge are image pyramids [48] and FPNs (Feature Pyramid Networks) [49].

The former adjusts the input image to multiple scales through image pyramids and train a separate detector for each scale. However, constructing and processing image pyramids requires additional computational resources, and the features of each scale are processed independently, which may lead to insufficient information dissemination.

To address these issues, the SNIP [50] algorithm has been proposed. SNIP selectively performs backpropagation based on the object size in each detector, reducing training time and improving efficiency. This selective backpropagation method can reduce unnecessary calculations while improving the accuracy of the model. In addition, in order to further improve efficiency, the SNIPER [51] algorithm was proposed, which further improves the efficiency of SNIP by processing the contextual area around each object instance instead of every pixel in the image pyramid.

On the other hand, FPN (Feature Pyramid Network) utilizes the inherent multiscale features in convolutional layers and combines them through horizontal connections in a top-down structure. This enables FPN to better preserve and utilize fine-grained details from lower-level features. The core idea of FPN is to combine lower-level detail features with higher-level semantic features through top-down paths and horizontal connections. This method can promote cross-scale feature propagation and integration of contextual information, thereby improving the accuracy of object detection. FPN combines horizontal connectivity and upsampling operations to fuse features of different scales together, generating feature maps with different scale information, thereby helping detection algorithms better adapt to objects of different scales.

In order to enhance the information flow in FPNs, the PANet and BiFPN algorithms were introduced. They utilize shorter paths to improve the exchange of feature information, thereby enhancing the multiscale representation ability of features and the transmission

ability of semantic information. In addition, the SAN (Scale Aggregation Network) algorithm maps multiscale features to the scale-invariant subspace, enhancing the robustness of the detector to scale changes, and thus mapping features of different scales to the same space, further improving the generalization ability and robustness of the model.

In contrast, image pyramids require additional computational resources and independently process features at each scale, which may limit the dissemination and utilization of information. FPN retains fine-grained details of lower-level features, promotes cross-scale feature propagation, and integrates contextual information through horizontal connections.

SPD [52] is a multiscale object detection method different from image pyramids and FPNs. It eliminates nonstep convolution and nonmaximum pooling operations, thereby preserving the original features. These methods have demonstrated significant improvements in multiscale object detection performance in various real-world applications.

On this basis, this paper proposes an improved SPDCS module that introduces convolutional attention operation on the basis of SPD, while processing the original image to further preserve the original features. Compared with SPD, SPDCS can retain more channel count features, thereby improving the multiscale representation ability of features and the transmission ability of semantic information.

### 2.3. Attention Mechanism

The attention mechanism is a widely used technique in machine learning and deep learning that has gained significant attention in recent years. It assigns different weights or levels of attention to different parts of the input data, enabling the model to selectively focus on information relevant to the current task. In contrast to traditional neural networks, which treat each input feature equally regardless of its importance to the task, the attention mechanism can adaptively allocate attention based on the different features of the data. By learning the weights, the attention mechanism can prioritize features that are more important to the task, thereby improving the model's performance.

One common form of attention mechanism is self-attention, also known as multihead attention. Self-attention can establish relationships between different positions in the input sequence and dynamically adjust weights based on input relevance. This approach has been widely applied in natural language processing tasks and has shown significant improvements in performance.

Notably, SEs (Squeeze-and-Excitation Networks) [53] and the CBAM (Convolutional Block Attention Module) [54] are two attention modules that have achieved remarkable success in computer vision tasks. SE can adaptively adjust the channel weights of convolutional features to pay attention to local features, while CBAM combines spatial attention and channel attention to more effectively capture local structural information of images.

Moreover, the Transformer [55] structure, which utilizes self-attention mechanisms, has revolutionized natural language processing and has been increasingly applied to object detection tasks. Its success has further highlighted the importance of attention mechanisms in improving the performance of deep learning models.

The proposed SPDCS module in this paper leverages the SE (Squeeze-and-Excitation) attention mechanism to extract crucial original features more effectively. In contrast to using standalone SPD modules, the integration of SE attention within the SPD process yields superior results.

## 3. Proposed Method

In this section, we first overview the design of YOLOV (see Figure 2). The module highlighted in the red box is proposed in this paper to effectively preserve more original features, leading to significant improvements in detection performance. Then, we describe details within an SPDCS block (see Figure 3), an SPPAUG block (see Figure 4), and a C2f block (see Figure 5).

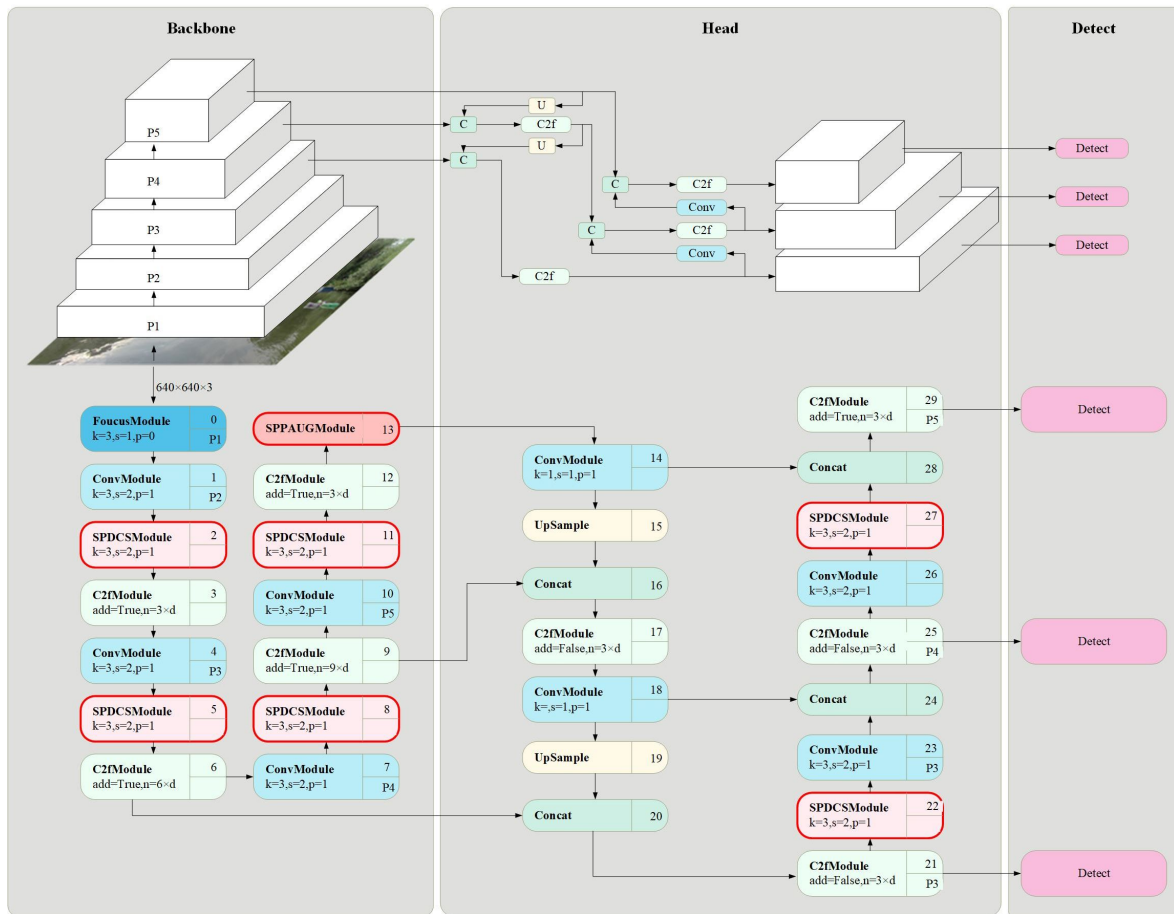


**Figure 2.** Architecture of YOLOv2.

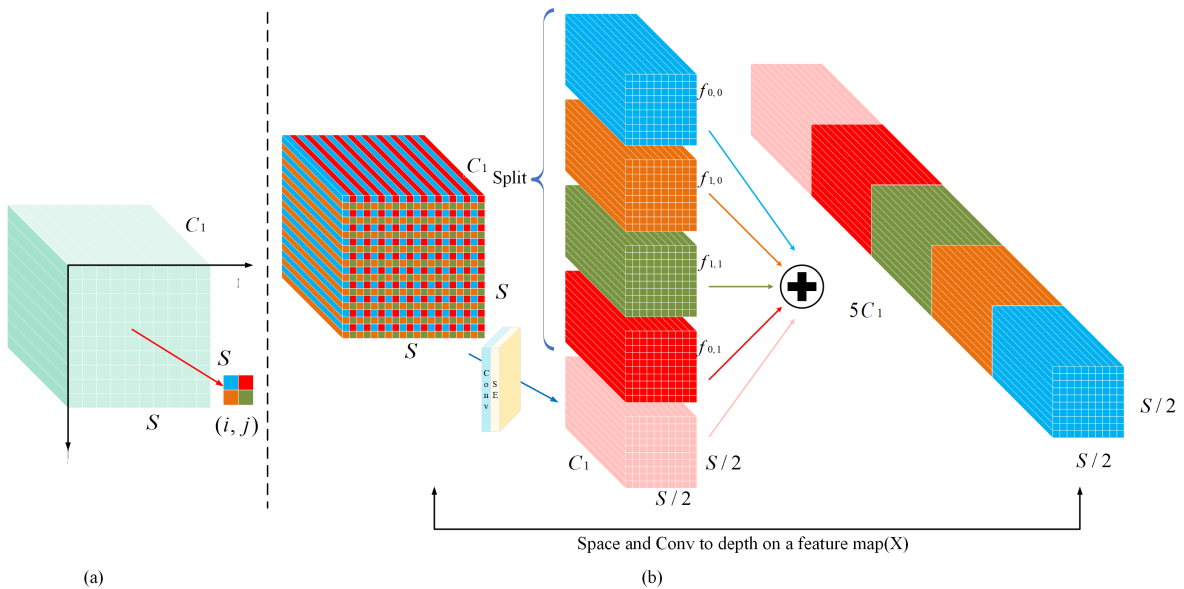


**Figure 3.** Architecture of SPDCS when scale = 2. (a) original features. (b) the process we do on the features.

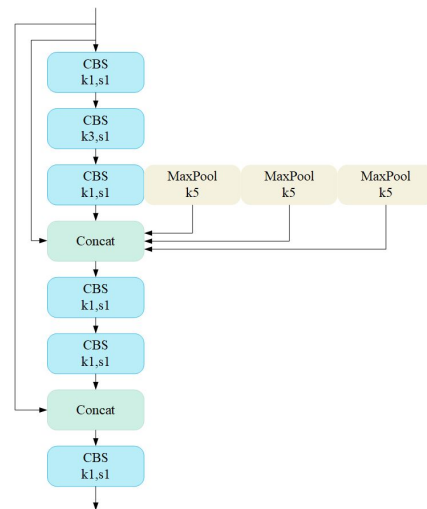


**Figure 4.** Architecture of SPPAUG.

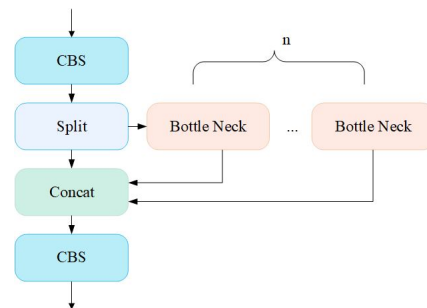


**Figure 5.** Architecture of the C2f block.

### 3.1. SPDCS

In object detection, preserving important information in feature maps is crucial for the accuracy of the model. This paper expands on the SPD (Space-to-Depth) module and creates the SPDCS (Space-to-Depth-Conv-Se) module. Operations such as step convolution and pooling may result in the loss of important details, so the SPD module avoids step convolution and pooling by using downsampling, thereby preserving the original features to a greater extent. On the basis of SPD, the SPDCS module further adds convolutional operations and attention mechanisms to process the original image, greatly preserving key information in the channel dimension, which helps to prevent the loss of basic information. In this paper, we use SPDCS blocks in the backbone and header to preserve the original features and further improve the representation ability of features and the transmission ability of semantic information through convolutional operations and attention mechanisms.

The Conv-Se module, as described in the literature, can be expressed as follows:

$$SE(Conv(X)) = \sigma(FC(f_{gap}(Conv(X)))) \cdot Conv(X) \quad (1)$$

$$Conv(X) = W * X + b \quad (2)$$

In these formulations, the input feature map is denoted as  $X$ . The  $Conv(X)$  operation represents the convolution operation on the input feature map using weight parameters  $W$  and a bias term  $b$ .  $f_{gap}$  (Global average pooling) calculates the average value of each channel in the feature map  $X$ , yielding a compressed global feature representation, and  $FC$  (fully connected layer) performs a linear transformation on the global feature representation.  $\sigma$  (sigmoid activation function) maps the output of the fully connected layer to the range  $[0, 1]$ , producing channel-wise attention weights.

For any intermediate feature map  $X$  of size  $S \times S \times C_1$ , the subfeature map sequence can be sliced as follows [52]:

$$f_{x,y} = X[x : S : scale, y : S : scale], x, y \in [0, scale) \quad (3)$$

To explain the process in more detail, given any feature map  $X$ , the submap  $f_{x,y}$  is formed by selecting all entries  $X(i, j)$ , where  $i + x$  and  $j + y$  are divisible by the scale factor. This results in each the submap being downsampled from  $X$  by the scale factor. Simultaneously,  $X$  is convolved and refined through an attention mechanism to obtain a more enhanced feature with dimensions of  $\frac{S}{2} \times \frac{S}{2} \times C_1$ . Figure 3b presents an example with  $scale = 2$ , where five subgraphs are obtained, namely  $f_{0,0}$ ,  $f_{1,0}$ ,  $f_{0,1}$ ,  $f_{1,1}$ , and  $Conv - Se$ , and  $X$  is downsampled by a factor of 2.

Next, these subfeature maps ( $f_{0,0}$ ,  $f_{1,0}$ ,  $f_{0,1}$ ,  $f_{1,1}$ ) and  $conv-se$  are concatenated along the channel dimension to obtain the feature map  $X'$ , which has reduced spatial dimensions by the  $scale$  factor and increased channel dimensions by  $scale^2$ . In other words, the SPDCS layer transforms the feature map  $X(S, S, C_1)$  into the intermediate feature map  $X'(\frac{S}{scale}, \frac{S}{scale}, (scale^2 + 1)C_1)$ . This approach effectively preserves critical information and can improve the accuracy of object detection models.

### 3.2. SPPAUG

In order to retain and utilize more original features, this paper creates the SPPAUG module as an extension of the SPPF module. The objective is to enhance the performance and efficiency of object detection algorithms while preserving the fast detection speed characteristic of the YOLO series. By striking a balance between speed and performance, the SPPAUG module improves the accuracy of object detection algorithms while maintaining their efficiency.

The module operates multiple convolutional kernels and max-pooling layers with a size of  $5 \times 5$  in a serial manner (see Figure 4), which is equivalent to a max-pooling operation with a larger size, such as  $9 \times 9$  or  $13 \times 13$ . This process leads to the extraction of more original features. Furthermore, the SPPFAUG module introduces spatial pyramid pooling and cross-stage feature fusion to further enhance the feature representation, resulting in improved detection accuracy and robustness.

In summary, the SPPAUG module is designed to balance speed and performance, and its implementation leads to an improvement in the efficiency and accuracy of object detection algorithms. We employ the SPPAUG block in the 13th layer in the backbone.

### 3.3. C2f

The C2f module is designed by drawing inspiration from both the C3 module and the ELAN architecture, aiming to provide richer gradient flow information while maintaining a lightweight design. Similar to the C3 module, the C2f module uses factorized convolutions to reduce computational complexity. However, instead of using channel-wise convolutions like C3, the C2f module factorizes the spatial and channel dimensions separately to further reduce the number of parameters.

Furthermore, as shown in Figure 5, the C2f module incorporates the ELAN design philosophy of utilizing multiple levels of feature maps to improve network performance. Specifically, it includes a feature fusion operation that aggregates feature maps from different levels with different scales to enhance the feature representation capability. This helps the network to capture both low-level and high-level features, making it more robust and effective for various object detection tasks. Overall, the C2f module strikes a balance between efficiency and effectiveness by leveraging the best practices from both C3 and ELAN architectures in a manner suitable for scientific writing. We employ C2f blocks in the backbone and heads. The illustration of the C2f block was inspired by [51].

## 4. Experimental Results and Analysis

Extensive experiments were conducted on the FLOW dataset [56] to demonstrate the effectiveness and superiority of the proposed improved YOLOv5 algorithm. Sections 4.1 and 4.2, respectively, introduce the dataset used in our experiments and the principle of the evaluation metrics. Section 4.3 presents the compared detectors, experimental details, and parameter settings. Model analysis and ablation analysis are performed in Section 4.4. The experimental hardware environment consisted of a Xeon(R) Platinum 8255C 12-core CPU, 2.50 GHz, and 40 GB RAM, NVIDIA GEFORCE RTX 3080 GPU, and the operating system was Linux. The experiments were implemented using Python 3.8.10, Conda 4.10.3, and Pytorch 1.9.0.

### 4.1. Experimental Dataset

The dataset for this paper is the FLOW dataset, which is a publicly available dataset provided by Ouke Intelligence, which consists of two subdatasets: FLOW-Img and FLOW-Radar-Img (FLOW-RI), the latter being a multimodal subdataset. FLOW-Img contains 2000 images with 5271 annotated objects, with small objects accounting for more than half of them. The dataset was collected under different lighting and wave conditions, observing the objects from different directions and angles. It is specifically designed for waterborne object detection, suitable for complex inland water scenes such as lakes, rivers, and reservoirs. The dataset has the following characteristics: The images in the dataset were taken from different angles, making it have a certain degree of view difference. Additionally, the image scenes in the dataset are diverse, such as different weather, time, and environments, which provide various data samples.

### 4.2. Experimental Indicators

In this paper, four widely used evaluation metrics in the field of computer vision are applied to quantitatively analyze the detection performance of the proposed YOLOv5 improved algorithm. The analysis is conducted from an academic perspective.

#### 4.2.1. Precision

Precision is a widely used evaluation metric in the field of computer vision, especially in object detection tasks. It measures the accuracy of positive predictions made by the model, that is, the proportion of true positive predictions among all positive predictions. Precision can be expressed by Equation (4). In other words, precision reflects the model's ability to avoid false positives. Precision can be calculated as the ratio of true positive predictions to the sum of true positive and false positive predictions. A higher precision value indicates a lower false positive rate, which means the model is better at correctly identifying positive samples.

#### 4.2.2. Recall

Recall, also known as sensitivity, is a measure of the ability of a model to correctly identify all positive samples. It is defined as the ratio of true positive samples to the total number of actual positive samples. Recall can be expressed by Equation (5). A higher recall indicates that the model is able to detect a higher proportion of positive samples, which is desirable in tasks where missing positive samples is more costly than false alarms.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Among them, true positive (TP) represents the number of samples that the model correctly predicts as positive, while false positive (FP) represents the number of samples that the model incorrectly predicts as positive.

#### 4.2.3. Average Precision

The average precision (AP) is a widely used evaluation metric in object detection, which measures the area under the precision–recall curve. The precision–recall curve is created by varying the confidence threshold of the model’s predictions and calculating precision and recall at each threshold. AP takes into account the precision at all recall levels and is more informative than only reporting precision or recall at a single threshold. AP ranges from 0 to 1, with higher values indicating better detection performance.

#### 4.2.4. Frames Per Second

Frames Per Second (FPS) is a metric used to measure the speed of object detection algorithms. Specifically, it refers to the number of frames or images that a detector can process in one second. A higher FPS indicates that the detector is able to process more images in a shorter amount of time and is therefore more efficient. In real-time applications, a high FPS is important to ensure that the detector can keep up with the speed of the input video stream.

### 4.3. Experiment Details and Settings

To ensure fairness, all training results are the average of 20 training runs. The reliability of the model is verified by randomly selecting images from the dataset for validation.

#### 4.3.1. Comparison Algorithms and Parameter Settings

To validate the effectiveness of the improved YOLOv5 algorithm, we conducted a comparative study with the original algorithm. We kept the detector settings consistent and adjusted the platform-related parameters to ensure fairness. Through these measures, we can eliminate the influence between different algorithms and platforms and more accurately compare the performance differences between the improved algorithm and the original algorithm.

#### 4.3.2. Parameter Settings of YOLOv5

In YOLOv5, the stochastic gradient descent (SGD) optimizer and the early stop strategy are employed, of which the learning rate and momentum are set to 0.01 and 0.937, respectively. In the training phase of YOLOv5, 60% of the raw images are captured as the training set and 40% as the validation set. The epoch and batch size of the network are set to 300 and 20, respectively.

We conducted various experiments to analyze the effectiveness of the proposed and introduced modules in YOLOv5. After conducting experiments on the Flow dataset, we obtained the model configurations and evaluation metrics shown in Table 1.

**Table 1.** Comparison of Object Detectors on Flow Dataset.

| Model   | #Param. | Size | C2f | SPD | SPDCS | SPPAUG | P     | R     | mAP <sub>0.5</sub> <sup>val</sup> | mAP <sub>0.75</sub> <sup>val</sup> | mAP <sub>0.5:0.95</sub> <sup>val</sup> | FPS   |
|---------|---------|------|-----|-----|-------|--------|-------|-------|-----------------------------------|------------------------------------|----------------------------------------|-------|
| 1       | 7.2M    | 640  | ×   | ×   | ×     | ×      | 81.8% | 73.4% | 78.7%                             | 27.2%                              | 36.0%                                  | 106.4 |
| 2       | 8.5M    | 640  | ✓   | ×   | ×     | ×      | 83.5% | 74.0% | 79.1%                             | 28.9%                              | 36.8%                                  | 126.6 |
| 3       | 8.8M    | 640  | ×   | ✓   | ×     | ×      | 84.9% | 74.3% | 80.4%                             | 29.2%                              | 37.3%                                  | 101.6 |
| 4       | 13.2M   | 640  | ×   | ×   | ✓     | ×      | 85.7% | 74.1% | 81.6%                             | 31.0%                              | 38.1%                                  | 82.7  |
| 5       | 13.7M   | 640  | ×   | ×   | ×     | ✓      | 84.5% | 72.9% | 79.3%                             | 27.4%                              | 36.6%                                  | 105.3 |
| 6       | 14.9M   | 640  | ✓   | ×   | ×     | ✓      | 85.6% | 72.6% | 79.6%                             | 29.4%                              | 37.0%                                  | 111.1 |
| 7       | 11.0M   | 640  | ✓   | ✓   | ×     | ×      | 82.9% | 75.4% | 81.0%                             | 29.2%                              | 37.7%                                  | 120.5 |
| 8       | 14.4M   | 640  | ✓   | ×   | ✓     | ×      | 84.4% | 76.6% | 81.9%                             | 30.5%                              | 38.7%                                  | 90.6  |
| 9       | 15.2M   | 640  | ×   | ✓   | ×     | ✓      | 82.9% | 74.7% | 81.1%                             | 30.3%                              | 38.2%                                  | 106.4 |
| 10      | 19.6M   | 640  | ×   | ×   | ✓     | ✓      | 83.8% | 75.6% | 81.8%                             | 32.0%                              | 38.8%                                  | 85.6  |
| 11      | 16.4M   | 640  | ✓   | ✓   | ×     | ✓      | 85.5% | 73.9% | 81.4%                             | 30.6%                              | 38.1%                                  | 107.6 |
| YOLOW   | 20.9M   | 640  | ✓   | ×   | ✓     | ✓      | 87.0% | 75.1% | 82.1%                             | 30.7%                              | 38.3%                                  | 93.2  |
| Improve | +13.7M  | 640  | ✓   | ×   | ✓     | ✓      | +0.52 | +0.17 | +0.34                             | +0.35                              | +0.23                                  | −13.2 |

Table 1 shows that the simplest Model-1 without the C2f, SPD, SPPFAUG, and SPDCS modules have the lowest performance. Model-2, adding the C2f module, achieves better performance compared with Model-1, demonstrating the effectiveness of the C2f module. Similarly, Models 3 to 5, solely utilizing the SPD, SPDCS, and SPPAUG modules, achieve better performance compared with Model-1, respectively, which demonstrates the effectiveness of these modules.

Furthermore, Model-5, adding the SPDCS module, achieves better performance than Model-4, demonstrating the superiority of the SPDCS module. Models 6 to 9 were trained by combining two modules to further demonstrate the improved performance obtained by combining modules compared with using a single module. Finally, our proposed method using a combination of the C2f, SPD, and SPPAUG modules outperformed the other models using different combinations of these three modules (Model-10). Although the YOLO algorithm proposed in this paper has some shortcomings compared with the basic YOLOv5s algorithm in terms of detection speed, it can still meet the requirements of real-time detection and has a significant improvement in accuracy compared with YOLOv5.

Table 2 showcases the current state-of-the-art one-stage object detection algorithms, YOLOv5s, YOLOv6s, YOLOv7-tiny, and YOLOv8s, as well as our proposed YOLO algorithm. The performance analysis confirms the effectiveness and accuracy of YOLO, further validating its superiority.

In recent years, real-time object detection has gained significant attention, leading to advancements in one-stage detection algorithms. YOLOv5s, an earlier version of the YOLO series, is renowned for its fast detection speed and satisfactory performance. YOLOv6s and YOLOv7-tiny have introduced improvements to enhance detection accuracy while maintaining efficiency. The latest version, YOLOv8s, incorporates advanced techniques for achieving even higher accuracy.

**Table 2.** Comparison of object detectors on Flow dataset.

| Model       | #Param. | Size | P     | R     | mAP <sup>val</sup> <sub>0.5</sub> | mAP <sup>val</sup> <sub>0.75</sub> | mAP <sup>val</sup> <sub>0.5:0.95</sub> | FPS   |
|-------------|---------|------|-------|-------|-----------------------------------|------------------------------------|----------------------------------------|-------|
| YOLOv5s     | 7.2M    | 640  | 81.8% | 73.4% | 78.7%                             | 27.2%                              | 36.0%                                  | 106.4 |
| YOLOv6s     | 17.2M   | 640  | 76.5% | 62.5% | 76.5%                             | 22.9%                              | 34.9%                                  | 106.1 |
| YOLOv7-tiny | 6.0M    | 640  | 81.3% | 66.9% | 73.6%                             | 23.1%                              | 32.6%                                  | 113.9 |
| YOLOv8s     | 11.2M   | 640  | 83.9% | 74.2% | 81.3%                             | 33.7%                              | 39.9%                                  | 106.4 |
| YOLOW       | 20.9M   | 640  | 87.0% | 75.1% | 82.1%                             | 30.7%                              | 38.3%                                  | 93.2  |

In comparison, our proposed YOLO algorithm demonstrates superior performance. It employs innovative approaches to address challenges in water-crossing environments, including reflection, occlusion, and distortion. By incorporating additional feature extraction techniques and optimization methods, YOLO outperforms the baseline YOLO models in terms of detection accuracy and robustness.

#### 4.4. Visual Evaluation of Experimental Results

Figures 6–8 demonstrate that our proposed YOLO algorithm outperforms YOLOv5s in detecting objects in water-crossing environments. Compared with YOLO, YOLOv5s exhibits some missed detections in Figure 6a,b and some false detections in Figure 6c,d, highlighting the higher accuracy of YOLO. Underwater environments pose significant challenges for object detection, as objects may be subject to reflections, occlusions, and distortions that alter their shape and color, making detection more difficult. To overcome these challenges, our YOLO algorithm employs advanced feature extraction techniques and optimization methods to enhance detection accuracy and robustness. Through comparison experiments with YOLOv5s, we demonstrate that our algorithm achieves superior detection performance in water-crossing environments, which is of practical significance for underwater robotics and object detection applications.

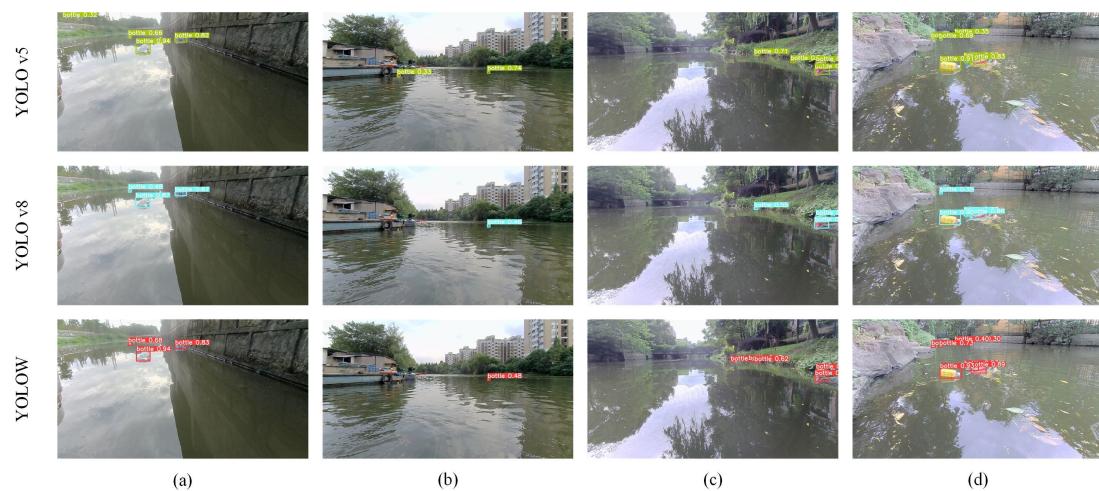


**Figure 6.** The detection results of YOLO v5, YOLO v8 and YOLOW on the existing data set FLOW. Sample (a–d) are four different experiments.

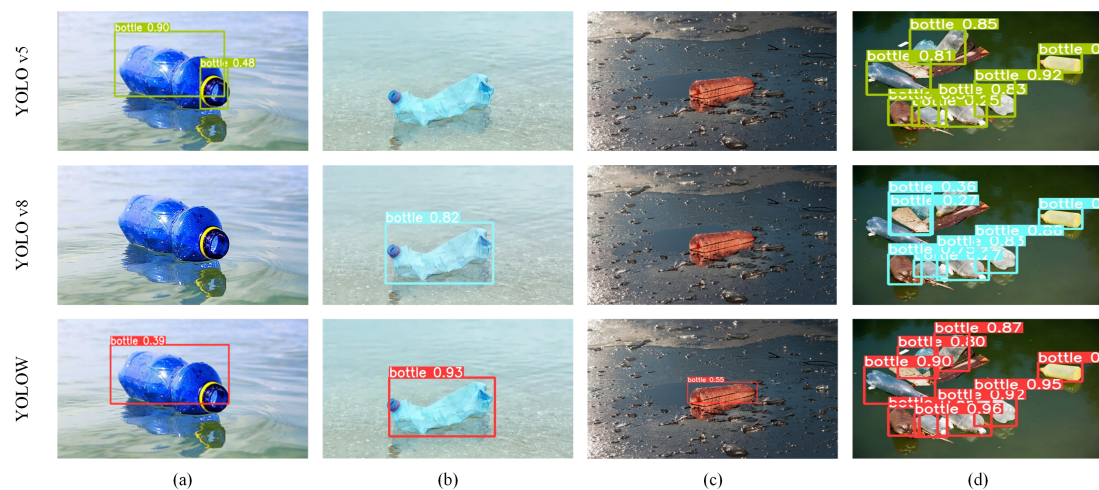


**Figure 7.** Use YOLO v5, YOLO v8 and YOLOW to collect the detection results on related pictures collected on the Internet respectively. Sample (a–c) are single object samples, and (d) is multiply objects sample.

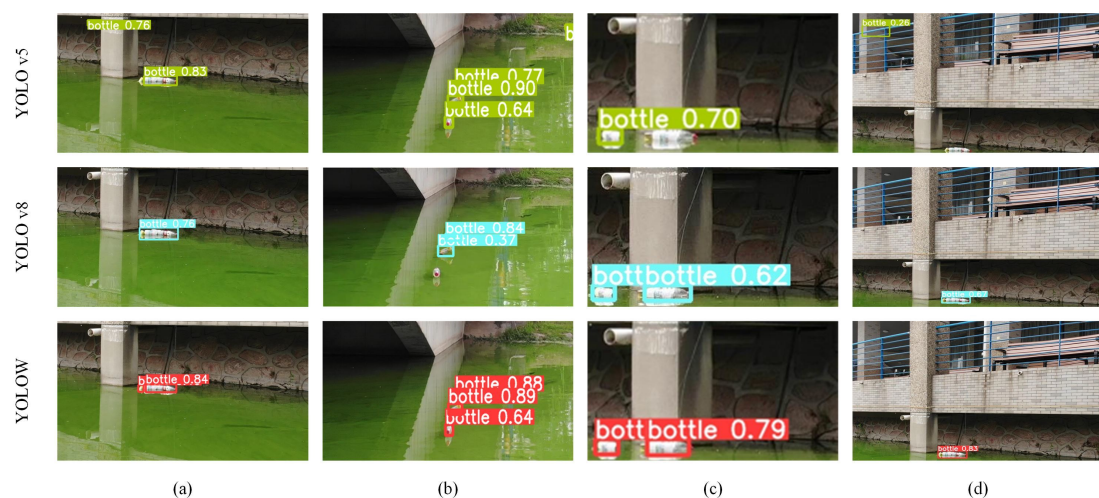


**Figure 8.** Use YOLO v5, YOLO v8 and YOLOW to detect the results of real-time images collected by drones. Sample (a–d) are four different experiments.

## 5. Conclusions

To address the problems of weak anti-interference ability, difficulty in recognizing small objects, and inability to handle complex environments in traditional water-crossing object detection, an improved water-crossing object detection algorithm based on YOLOv5 is proposed. By introducing three modules into the network model, the algorithm greatly reduces the probability of missed detection while improving detection accuracy, and it also has a “correcting” mechanism. Experimental results show that the proposed method improves precision, recall, and average precision by 6.4%, 2.3%, and 4.3%, respectively. This method can handle water-crossing objects of different scales, solving the problem of object recognition in complex water environments. Although the improved model meets high precision requirements, its size is larger than the original YOLOv5 model. The next step is to develop a lightweight model based on this improved model for optimization.

**Author Contributions:** Conceptualization, S.X., H.T., J.L., L.W., X.Z. and H.G.; methodology, S.X., H.T., J.L., L.W., X.Z. and H.G.; validation, H.T. and S.X.; formal analysis, H.T. and S.X.; investigation, S.X. and H.T.; writing—original draft preparation, H.T. and S.X.; writing—review and editing, H.T.; supervision, S.X.; project administration, S.X. and H.G.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 62071168, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20211201, and in part by the China Postdoctoral Science Foundation under Grant 2021M690885.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within in the paper.

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

## References

1. Dai, H.; Huang, G.; Wang, J.; Zeng, H. VAR-tree model based spatio-temporal characterization and prediction of O<sub>3</sub> concentration in China. *Ecotoxicol. Environ. Saf.* **2023**, *257*, 114960. [[CrossRef](#)] [[PubMed](#)]
2. Zeng, H.; Shao, B.; Dai, H.; Yan, Y.; Tian, N. Prediction of fluctuation loads based on GARCH family-CatBoost-CNNLSTM. *Energy* **2023**, *263*, 126125. [[CrossRef](#)]
3. Cheng, Y.; Xu, H.; Liu, Y. Robust small object detection on the water surface through fusion of camera and millimeter wave radar. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15263–15272.
4. Behnamian, A.; Banks, S.; White, L.; Brisco, B.; Millard, K.; Pasher, J.; Chen, Z.; Duffe, J.; Bourgeau-Chavez, L.; Battaglia, M. Semi-automated surface water detection with synthetic aperture radar data: A wetland case study. *Remote. Sens.* **2017**, *9*, 1209. [[CrossRef](#)]
5. Shakmak, B.; Al-Habaibeh, A. Detection of water leakage in buried pipes using infrared technology; A comparative study of using high and low resolution infrared cameras for evaluating distant remote detection. In Proceedings of the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), IEEE, Amman, Jordan, 3–5 November 2015; pp. 1–7.
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
11. 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](#)]
12. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.

13. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
14. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [CrossRef] [PubMed]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
19. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
20. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 13029–13038.
21. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
22. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
23. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
24. YOLO v8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 10 January 2023).
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
26. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
27. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
28. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: A simple and strong anchor-free object detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1922–1933. [CrossRef]
29. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
30. Kirillov, A.; Girshick, R.; He, K.; Dollár, P. Panoptic feature pyramid networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6399–6408.
31. Hu, M.; Li, Y.; Fang, L.; Wang, S. A2-FPN: Attention aggregation based feature pyramid network for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 15343–15352.
32. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 7373–7382.
33. Li, Y.; Mao, H.; Girshick, R.; He, K. Exploring plain vision transformer backbones for object detection. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part IX; Springer: Berlin/Heidelberg, Germany, 2022; pp. 280–296.
34. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 10213–10224.
35. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
36. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 14454–14463.
37. Zhou, D.; Fang, J.; Song, X.; Guan, C.; Yin, J.; Dai, Y.; Yang, R. Iou loss for 2d/3d object detection. In Proceedings of the 2019 International Conference on 3D Vision (3DV), IEEE, Quebec City, QC, Canada, 16–19 September 2019; pp. 85–94.
38. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
39. Chen, K.; Lin, W.; Li, J.; See, J.; Wang, J.; Zou, J. AP-loss for accurate one-stage object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3782–3798. [CrossRef]
40. Oksuz, K.; Cam, B.C.; Akbas, E.; Kalkan, S. A ranking-based, balanced loss function unifying classification and localisation in object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 15534–15545.

41. Oksuz, K.; Cam, B.C.; Akbas, E.; Kalkan, S. Rank & sort loss for object detection and instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021; pp. 3009–3018.
42. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI conference on artificial intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
43. Zhu, B.; Wang, J.; Jiang, Z.; Zong, F.; Liu, S.; Li, Z.; Sun, J. Autoassign: Differentiable label assignment for dense object detection. *arXiv* **2020**, arXiv:2007.03496.
44. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, Virtual Conference, 11–17 October 2021; pp. 3490–3499.
45. Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; Sun, J. Ota: Optimal transport assignment for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 303–312.
46. Li, S.; He, C.; Li, R.; Zhang, L. A dual weighting label assignment scheme for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 9387–9396.
47. Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. End-to-end object detection with fully convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Conference, 19–25 June 2021; pp. 15849–15858.
48. Adelson, E.H.; Anderson, C.H.; Bergen, J.R.; Burt, P.J.; Ogden, J.M. Pyramid methods in image processing. *RCA Eng.* **1984**, *29*, 33–41.
49. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
50. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
51. Singh, B.; Najibi, M.; Davis, L.S. Sniper: Efficient multi-scale training. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. Available online: <https://proceedings.neurips.cc/paper/2018/hash/166cee72e93a992007a89b39eb29628b-Abstract.html> (accessed on 29 June 2023).
52. Sunkara, R.; Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. *arXiv* **2022**, arXiv:2208.03641.
53. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
54. 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.
55. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: [https://proceedings.neurips.cc/paper\\_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html) (accessed on 29 June 2023).
56. Cheng, Y.; Zhu, J.; Jiang, M.; Fu, J.; Pang, C.; Wang, P.; Sankaran, K.; Onabola, O.; Liu, Y.; Liu, D.; et al. Flow: A dataset and benchmark for floating waste detection in inland waters. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021; pp. 10953–10962.

**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.