Journal of *Marine Science and Engineering*

MDPI

Article

A High-Precision Detection Model of Small Objects in Maritime UAV Perspective Based on Improved YOLOv5

Zhilin Yang, Yong Yin, Qianfeng Jing * and Zeyuan Shao

Navigation College, Dalian Maritime University, Dalian 116026, China; dmu_zly@dlmu.edu.cn (Z.Y.); bushyin@dlmu.edu.cn (Y.Y.); szy@dlmu.edu.cn (Z.S.)
* Correspondence: jqf_dlmu@dlmu.edu.cn

Abstract: Object detection by shipborne unmanned aerial vehicles (UAVs) equipped with electro-optical (EO) sensors plays an important role in maritime rescue and ocean monitoring. However, high-precision and low-latency maritime environment small-object-detection algorithms remain a major challenge. To address this problem, this paper proposes the YOLO-BEV ("you only look once"–"bird's-eye view") model. First, we constructed a bidirectional feature fusion module—that is, PAN+ (Path Aggregation Network+)—adding an extremely-small-object-prediction head to deal with the large-scale variance of targets at different heights. Second, we propose a C2fSESA (Squeeze-and-Excitation Spatial Attention Based on C2f) module based on the attention mechanism to obtain richer feature information by aggregating features of different depth layers. Finally, we describe a lightweight spatial pyramid pooling structure called RGSPP (Random and Group Convolution Spatial Pyramid Pooling), which uses group convolution and random channel rearrangement to reduce the model's computational overhead and improve its generalization ability. The article compares the YOLO-BEV model with other object-detection algorithms on the publicly available MOBDrone dataset. The research results show that the $mAP_{0.5}$ value of YOLO-BEV reached 97.1%, which is 4.3% higher than that of YOLOv5, and the average precision for small objects increased by 22.2%. Additionally, the YOLO-BEV model maintained a detection speed of 48 frames per second (FPS). Consequently, the proposed method effectively balances the accuracy and efficiency of object-detection in shipborne UAV scenarios, outperforming other related techniques in shipboard UAV maritime object detection.

Keywords: shipborne UAV scenarios; object detection; attention mechanism; space pyramid pool; YOLOv5

Citation: Yang, Z.; Yin, Y.; Jing, Q.; Shao, Z. A High-Precision Detection Model of Small Objects in Maritime UAV Perspective Based on Improved YOLOv5. *J. Mar. Sci. Eng.* **2023**, *11*, 1680. https://doi.org/10.3390/jmse11091680

Academic Editor: Alessandro Ridolfi

Received: 11 July 2023
Revised: 11 August 2023
Accepted: 24 August 2023
Published: 25 August 2023

1. Introduction

Object detection from the perspective of unmanned aerial vehicles (UAVs) is an efficient and flexible recognition method which has been widely used in various fields. For example, in man overboard (MOB) search and rescue [1], urban road traffic control [2], synthetic aperture radar monitoring [3], and building structure defect detection [4], UAV image detection has unique advantages and developmental potential. As an emerging means of maritime monitoring, shipborne UAVs have the advantages of high flexibility, rapid response, and low costs. Consequently, object detection based on shipborne UAVs has become a popular research topic in the field of maritime monitoring.

In maritime environments, images captured by UAVs often contain small objects—for example, MOB situations, life rafts, life buoys, and small boats, which often represent emergencies. Using UAVs to detect these small objects, they can be quickly located, effectively improving rescue efficiency and success rates, which is of great importance in the field of maritime rescue and monitoring. However, visible light images of the maritime environment captured by electro-optical (EO) sensors often contain complex maritime environmental information [5,6], as shown in Figure 1.

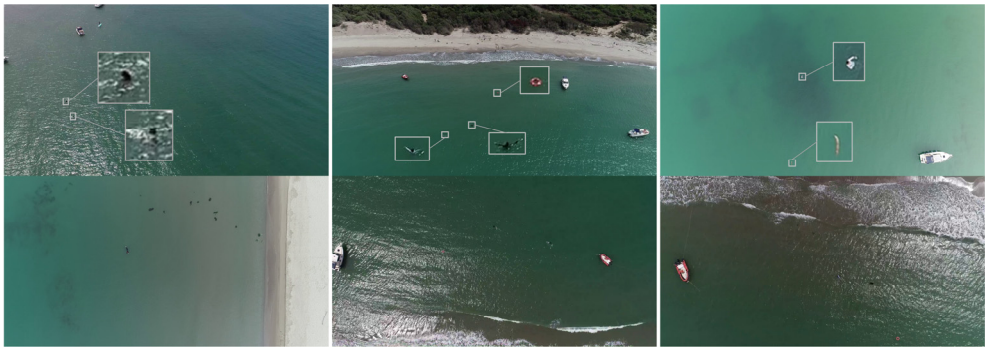


Figure 1. Main features of maritime UAV images. The square shows small objects such as man overboard and life buoy taken by the drone at a height of 60 m.

First, the shapes and sizes of small objects in the image can change because of changes in the flying height and angle of the UAV. In complex maritime environments, small objects can be difficult to observe with the naked eye, and can be easily affected by disturbance factors, such as waves. Second, changes in illumination and maritime texture can blur the boundaries and outlines of small objects in maritime areas, thereby reducing detection accuracy. Consequently, considering changes in the object size at different heights and angles, different illumination interference, and maritime textures, among others, an object-detection algorithm must exhibit strong adaptability and robustness to improve the detection performance of small objects in maritime environments from a UAV perspective.

Developing a detection algorithm with high precision and low latency in detecting small maritime objects from a UAV perspective is a major challenge. Traditional small-object-detection methods, such as those based on template matching, feature extraction, and biological vision, can achieve small-object detection to a certain extent but can be affected by environmental changes and object positions. For example, background subtraction [7] is effective in detecting small objects with static backgrounds but performs poorly with dynamic backgrounds. Consequently, to improve detection accuracy, researchers have attempted to integrate a variety of features and filtering methods—such as histogram of oriented gradients (HOG) features [8], scale-invariant feature transform (SIFT) features [9], and Gabor filtering [10]. However, these methods still suffer from problems of false positives when the object and background have similar features. Additionally, researchers have exploited biological vision mechanisms, such as the graph-based visual saliency (GBVS) algorithm [11] and biological eagle-eye vision adaptive methods [12], to identify small-object regions. However, these methods usually perform poorly in terms of detection accuracy for extremely small objects.

In recent years, researchers have begun to explore the use of deep-learning methods to solve small-object-detection problems. Such methods based on deep learning can be divided into one-stage and two-stage methods. The one-stage method directly outputs the position and category of an object through a forward pass, making it suitable for scenarios with high real-time requirements; however, its accuracy is relatively low. Representative algorithms include the single shot multiBox detector (SSD) [13], focal loss for dense object detection (RetinaNet) [14], you only look once version 3 (YOLOv3) [15], and you only look once version 7 (YOLOv7) [16] algorithms. The two-stage approach involves splitting the object-detection task into two stages—that is, first extracting a set of candidate object regions from an image through a selective search or candidate region extraction, and then classifying and regressing these regions. The two-stage method can obtain better detection accuracy, but its real-time performance is poor. Representative algorithms include region-based convolutional neural networks (R-CNNs) [17], faster region-based convolutional neural networks (Faster R-CNNs) [18], and an IoU-aware dense object detector (VarifocalNet) [19]. These deep-learning methods have achieved remarkable progress in small-object detection based on UAV scenarios, owing to their excellent feature extraction capabilities and end-to-end training methods.

To further meet the accuracy and real-time requirements of small-object detection from a UAV perspective, researchers have made a series of improvements and optimizations using deep-learning technology. First, in response to the large-scale variance problem of small objects caused about by changes in the flying height and angle of UAVs, researchers have proposed a series of new network modules to improve the detection accuracy of small objects. Yang et al. [20] designed a multi-scale semantic enhancement module to adapt to the large-scale variance problem of small objects, but it achieved a small improvement in accuracy. Liu et al. [21] improved the Darknet structure of YOLOv3 by optimizing the ResBlock module and adding convolution operations to a shallow layer. This method effectively expanded the receptive field but did not consider the impact of complex backgrounds on small objects.

Second, researchers have tried to address the problem of complex backgrounds and illumination changes in the images captured by UAVs. Ye et al. [22] proposed a global-local feature enhancement network to solve the problem of dense-object detection in complex backgrounds; however, they could not meet the real-time requirements. Wang et al. [23] solved the problem of an object being submerged by background clutter during the detection and tracking process by adding a channel attention mechanism to the model; however, they did not discuss the detection performance of small objects. Chen et al. [24] proposed a weather-sensing object-detection method, which realizes high-precision object detection through dynamic selection of machine learning models and regular training, and shows significant performance improvement under rainy and foggy conditions.

Finally, to improve the real-time stability of small-object detection, researchers have also focused their attention on lightweight module construction and network pruning. Ye et al. [25] designed a lightweight feature extraction module to build a backbone network to control the model parameters and calculations, realizing the real-time detection of small objects in UAV images; however, their method did not achieve high-precision performance. Sharafaldeen et al. [26] relied on convolutional neural technology to detect the sea-surface object from the top view and deployed the training model on the embedded edge device, achieving a reasoning performance of more than 80 frames per second, but the detection accuracy needs to be improved. Cai et al. [27] adopted a network model pruning algorithm to solve the problem that real-time performance of the algorithm cannot be satisfied, being compromised owing to the limited resources of the computing platform. However, network pruning degraded the overall performance of the model. In summary, these improvements provide many ideas and methods for ongoing research into small-object detection in maritime environments from a UAV perspective, promoting the continuous development in this field.

To achieve high-precision and real-time detection of small objects, this study proposes a high-precision maritime detection system for small objects based on shipborne UAV scenarios. We call this the YOLO-BEV ("you only look once"–"bird's-eye view") model, and its network model is shown in Figure 2. Moreover, considering the limited computing resources of shipborne UAV platforms, we used the you only look once version 5 (YOLOv5) framework as the basic framework for the proposed model.

Experimental results on the MOBDrone dataset demonstrated that the proposed method could realize good real-time performance while achieving high-precision detection. The main contributions of this study are as follows:

This article constructed a bidirectional feature fusion module—that is, PAN+—which added an extremely small-object prediction head to the network to deal with the large-scale variance of objects at different heights and improve the detection accuracy of small maritime objects.

This article proposed a C2fSESA module based on the attention mechanism that obtained richer gradient flow information by aggregating the features of different depth layers, improving the model's perception of key features.

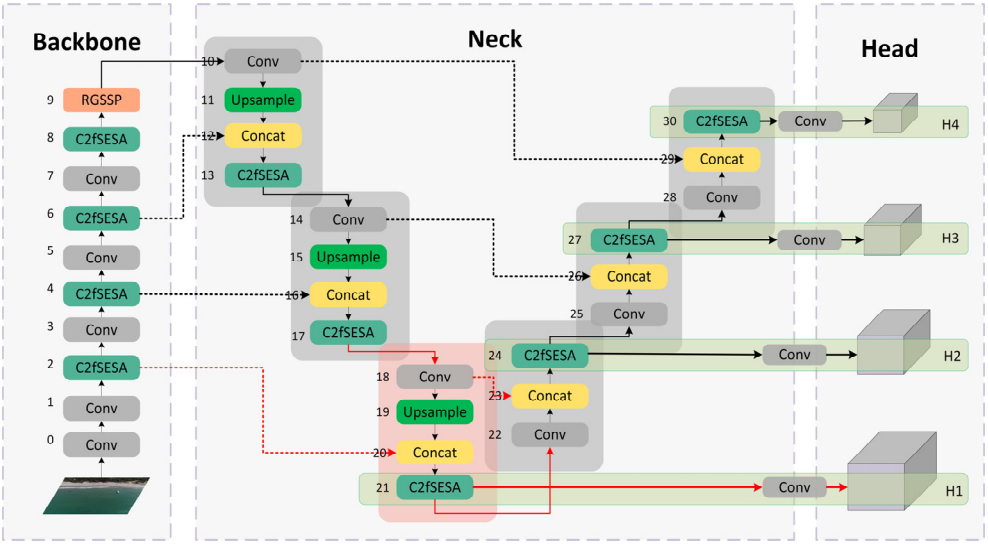


Figure 2. Network model of YOLO-BEV algorithm.

This article designed a lightweight spatial pyramid pooling structure—that is, RGSPP—which reduced the parameters and computational overhead of the model by grouping the convolutions and randomly rearranging the channels, improving the model detection accuracy and generalization performance.

This article proposed a maritime small-object-detection algorithm—that is, the YOLO-BEV model—suitable for a bird's-eye view (BEV), which realized high-precision and real-time detection of small objects from a shipborne UAV perspective.

The remainder of the paper is structured as follows: Section 2 describes related work on detector-related modules. Section 3 analyzes the BEV network model. Section 4 discusses the effectiveness of the proposed algorithm via analysis and discussion of the experimental results. Section 5 presents the conclusions.

2. Related Work

In this section, we provide an overview of related work from three aspects—the feature fusion structure, attention mechanism module, and spatial pyramid pooling structure.

2.1. Feature Fusion Structure

You only look once (YOLO) is an advanced object-detection algorithm. YOLOv1 [28] and YOLOv2 [29] improved the accuracy of object detection using methods such as deep convolutional neural networks and anchor mechanisms; however, they could not effectively detect objects of different scales. To solve these problems, YOLOv3 [15] introduced a feature fusion network and feature pyramid network (FPN) strategy; you only look once version 4 (YOLOv4) [30] and YOLOv5 later used the FPN strategy.

A feature fusion network is used to fuse the feature maps at different levels. The FPN [31] processes feature maps of different scales by constructing feature pyramids, enabling the network to simultaneously process objects at different scales, thus improving the performance of the detector. However, when an FPN performs feature fusion on multiscale feature maps, problems remain, such as the loss of small-object information. To solve these problems, researchers have proposed a series of optimization methods for FPN. Ghiasi et al. [32] designed a neural architecture search feature pyramid network (NAS-FPN) using a neural network structure search method to solve the problem of small-information loss in an FPN, automatically selecting the scale, number of layers, and feature fusion method for a feature map. Liu et al. [33] proposed a path aggregation network (PANet) based on adaptive feature pooling and feature weighting to solve the problem of low FPN accuracy. By performing adaptive feature pooling and feature weighting on feature maps at

different levels and then performing cascade aggregation, the network could better capture the characteristics of small objects. Tan et al. [34] proposed a bidirectional feature pyramid network (BiFPN) that introduced feature fusion paths in both the top-down and bottom-up directions. With this structure, the BiFPN could achieve adaptive feature fusion between different scales and levels. Additionally, Liu [35] et al. proposed an attentional semantic feature fusion (ASFF) method. It adaptively selected and fused features of different scales by computing the importance scores of feature maps, thereby improving the performance and robustness of object detection. In summary, these methods improved the performance and range of FPNs by introducing different feature fusion paths and mechanisms.

2.2. Attention Mechanism Module

The attention mechanism provides the ability to focus on and sift through information clutter. It can find small-object features in massive information, so that it can process and understand complex perception tasks more effectively.

Consequently, researchers have introduced different attention mechanisms in convolutional neural networks to improve their performance in vision tasks by learning attention weights to highlight important features and suppress noise effects. The squeeze-and-excitation (SE) [36] module is a local attention mechanism based on channel attention that can calculate channel weights through its global average pooling layer and two fully connected layers. The efficient channel attention (ECA) [37] module is a lightweight local attention mechanism that focuses only on the channel dimension and learns the interaction and feature representation capabilities between channels through one-dimensional convolution operations. Additionally, the coordinate attention (CA) [38] module is a local attention mechanism based on location coordinates and can be implemented by performing a spatial transformation network on the input feature map. It can learn the importance weights for each location, and capture location information and spatial relationships. However, the convolutional block attention module (CBAM) [39] also pays attention to spatial attention while considering channel attention, performing attention operations in both the channel and spatial dimensions. As distinct from these local attention mechanisms, the global attention module (GAM) [40] is a global attention mechanism that generates a global context vector by weighted summation over the entire input sequence to capture the key information of the entire sequence.

In conclusion, the application of the attention mechanism in convolutional neural networks can provide greater representational ability and adaptability to the model. However, choosing an appropriate attention mechanism requires considering many factors such as the application scenario and the computational overhead of the model.

2.3. Spatial Pyramid Pooling Structure

Spatial pyramid pooling (SPP) is a technique for mapping an input image of any size into a fixed-size feature vector in a convolutional neural network, which can adaptively divide the input image into a pyramid and perform pooling at each division level so that the network can handle images of any size and scale.

He et al. [41] introduced an SPP module into a convolutional neural network for the first time. The SPP module divided the feature map into multiple grids of different scales and performed a pooling operation within each grid, thereby combining the features of different scales to improve the accuracy of the model. The spatial pyramid pooling—fast (SPPF) [42] module greatly improved the calculation speed by using one pooling layer instead of multiple pooling layers in the SPP module and by adding two pooling operations of different sizes to obtain more contextual information. To further improve the calculation speed, the simplified spatial pyramid pooling—fast (SimSPPF) [43] module used the rectified linear unit (ReLU) activation function instead of the sigmoid-weighted linear unit (SiLU) activation function of the SPPF module. The spatial pyramid pooling cross-stage partial connection (SPPCSPC) [16] module introduced a cross-stage partial (CSP) connection to improve the feature transfer capability of the SPP module; however, the

computational overhead was substantial. In convolutional neural networks, SPP technology provides an effective solution for processing images of arbitrary sizes and scales and has been continuously optimized and improved.

3. Methods: The YOLO-BEV Model

This section describes the proposed methodology—that is, a YOLO-BEV model based on the YOLOv5 framework. For the backbone network, this article proposes an RGSPP structure; for the feature fusion network, we designed a PAN+ structure; and for the entire YOLO-BEV model, we designed C2fSESA as a CSP network. In this section, the paper provides a detailed introduction to the key modules—namely, the YOLO-BEV, PAN+, C2fSESA, and RGSPP modules.

3.1. PAN+ Module

The feature fusion module of YOLOv5 fuses feature maps at different levels and outputs feature maps at three different scales, the three feature-map sizes being 80 × 80, 40 × 40, and 20 × 20, corresponding to the detection of small, medium, and large objects, respectively. However, owing to changes in the flying height and angle of UAVs, extremely small objects are often present in the captured images. Consequently, we used a method of adding large-scale feature maps to solve the problem of low detection accuracy for small objects.

During object detection, in addition to paying attention to the semantic information of high-level features, we also need to consider shallow texture information, such as image pixels. Therefore, the network layers in the backbone should be fused to obtain both deep semantic and shallow texture information. To achieve this goal, the PANet [33] uses a top-down and bottom-up bidirectional fusion backbone network and adds a "shortcut" between the bottom and top layers to shorten the path between layers. To improve the detection accuracy of small maritime objects from the UAV perspective, as shown in Figure 3, we designed PAN+ as a feature fusion module. With the deepening of the backbone network, the feature map is downsampled five times with a step size of two. As the size of the feature map continues to decrease, the semantic information becomes more abundant. Then, upsampling in the FPN three times generates a larger 160 × 160 feature map (the green feature map shown in Figure 3), so that the underlying features also contain rich semantic information. Finally, a bottom-up feature aggregation method in the PAN is adopted, further enriching the location information of the object, thus alleviating the problem of small maritime object loss during the feature fusion process.

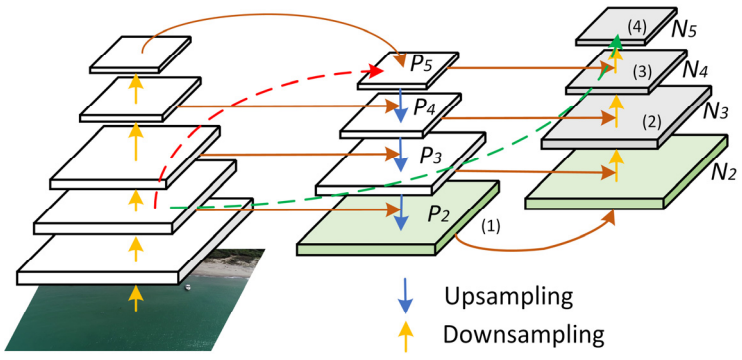


Figure 3. Schematic diagram of the PAN+ module.

In the proposed algorithm, we used the last 160 × 160 *feature map (1)* and output the predicted *feature maps (2)*, *(3)*, and *(4)* after the three PAN+ structures. The added 160 × 160 large-scale feature map provides rich feature information for the detection of small objects in maritime environments, improving the detection accuracy of small objects from a shipborne UAV perspective.

3.2. C2fSESA Module

Maritime images captured by shipborne UAVs always contain interference information—such as illumination reflections and texture changes—making it difficult to detect objects. Consequently, we used an attention mechanism to solve the problem of background interference. The attention mechanism allows the model to focus selectively on important features and suppress noise and redundant information, thereby improving its perception of key features.

3.2.1. The SESA Attention Mechanism

The CBAM [39] is a channel-spatial attention mechanism based on a convolutional neural network, that can learn and adaptively adjust the importance of feature maps. The channel attention mechanism module of the CBAM compresses the dimensions of the feature map into two different feature descriptions by performing global maximum pooling and global average pooling on the input feature map. The two feature maps share a multilayer perceptron network comprising a fully connected layer that reduces the number of channels, and a fully connected layer that restores the number of channels. The weight of each channel is obtained by stacking two feature maps and using a sigmoid activation function for normalization. The channel attention can be calculated using Equation (1) [39]:

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(AvgPool(F))) \\ &= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{1}$$

where F^c_{avg} and F^c_{max} denote the average and maximum pooled feature maps, respectively; W_0 and W_1 denote the two-layer weights of the multilayer perception, respectively; and σ denotes the sigmoid activation function.

In the spatial attention mechanism module of the CBAM, maximum and average pooling are first performed on the output feature map of the channel attention mechanism, and the two obtained feature maps are stacked on the channel dimension. Subsequently, a 7×7 convolutional layer is used to fuse the channel information of these two feature maps, and a sigmoid function is used to normalize the channel weights at each spatial position to obtain the spatial attention of the feature maps. The calculation method is shown in Equation (2) [39]:

$$\begin{aligned} Mc(F) &= \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7\times7}([F^s_{avg}; F^s_{max}])) \end{aligned} \tag{2}$$

where $f^{7\times7}$ denotes a convolution operation with a filter size of 7×7.

The SE [36] module is a channel attention module based on the squeeze-and-excitation mechanism. In the compression operation, the channel information is generated by global average pooling and compressed into channel descriptors. The global average pooling can be calculated using Equation (3) [36]:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{3}$$

where u_c denotes the input feature, $H \times W$ denotes the dimension of the $f^{7\times7}$ information, z_c denotes the pooling result, and F_{sq} denotes the squeeze operation.

In the excitation operation, the vector z obtained in the previous step can be processed through two fully connected layers (W_1 and W_0) to obtain the channel weight value (s), as shown in Equation (4) [36]:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z)) \tag{4}$$

where the weight vector s is used to assign weights to the feature u_c to obtain the final feature map $\widetilde{X}$, which can be calculated using Equation (5) [36]:

$$\widetilde{X} = F_{scale}(u_c, s_c) = s_c \cdot u_c \tag{5}$$

SE adds a squeeze-and-excitation module after each convolutional layer to learn the weight of each channel, strengthen the response of important channels, and reduce the response of unimportant channels, thereby improving the model's attention to important features. By contrast, the channel attention module of the CBAM processes only feature maps through max pooling and average pooling, which may fail to capture the more complex dependencies between channels. Consequently, we combined SE with the spatial attention module of the CBAM to propose a new SESA attention mechanism, as shown in Figure 4.

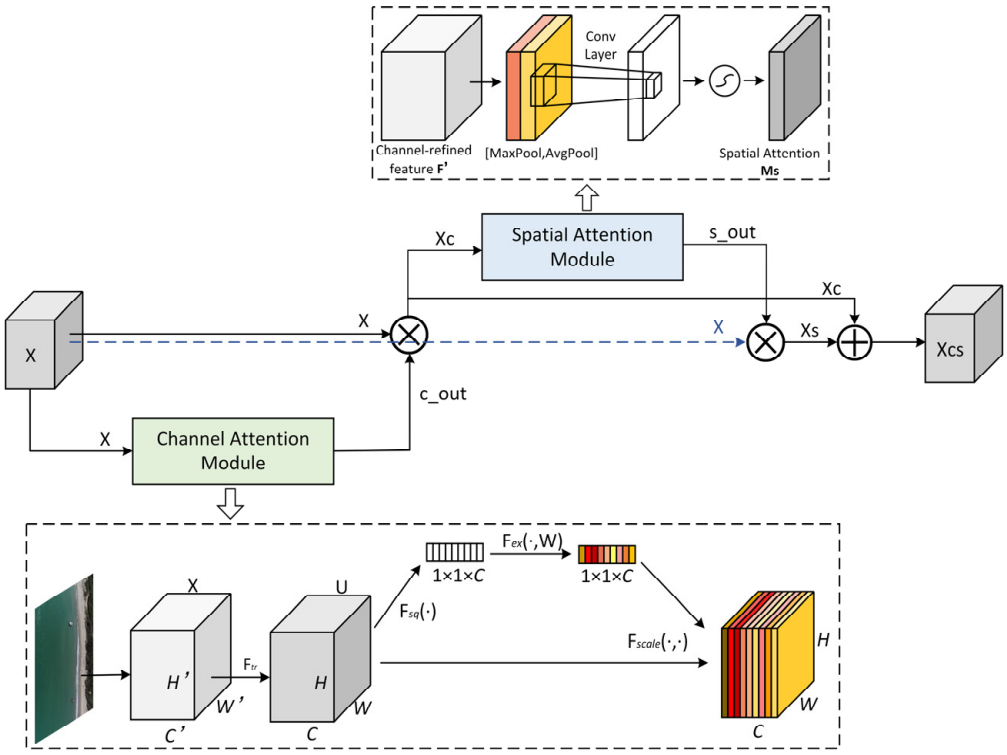


Figure 4. Channel–Spatial attention module.

First, the input feature map obtains the channel attention weight *c_out* of the feature map using the SE attention module. Subsequently, *c_out* is multiplied by input feature map X to generate a new feature map Xc based on the channel attention weight. The calculation process is shown in Equation (6):

$$X_c = X \otimes c\_out \tag{6}$$

The feature map Xc generated by the SE module is input into the spatial attention module of the CBAM to obtain the spatial attention weight *s_out* of the feature map. Subsequently, X can be considered to be the feature map multiplied by *s_out*. Because of the illumination reflection and texture changes in maritime images, the pixel value of the image changes spatially; therefore, the spatial information in the image can be difficult to weight accurately, making it difficult to determine the weight distribution corresponding to the spatial information of the image. If Xc is multiplied by *s_out*, this weighting method overemphasizes the information of certain spatial positions, resulting in the compression or overemphasis of the reflection and texture change information in the image. We chose to use X as the feature map multiplied by *s_out* such that the

spatial attention weight *s_out* only controls the weighting of spatial information, thereby preserving more spatial detail information without overemphasizing specific spatial locations, which can be calculated as follows:

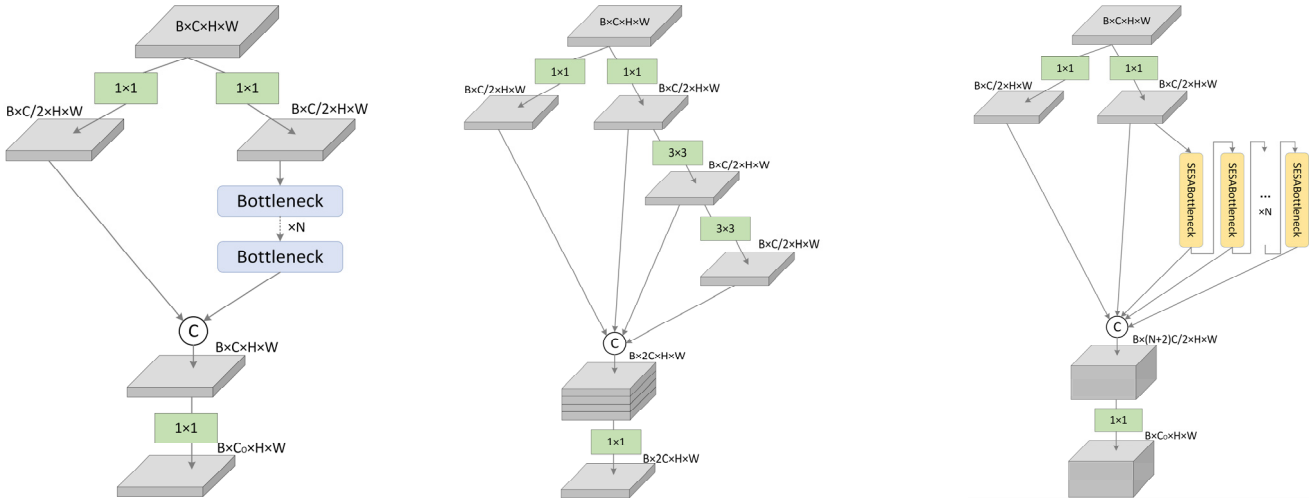
$$X_s = X \otimes s\_out \tag{7}$$

Finally, we applied the sigmoid activation function to the feature map after summing *Xc* and *Xs* to output the final feature map, the calculation formula of which is shown in Equation (8):

$$X_{cs} = sigmoid\ (Xc \oplus Xs) \tag{8}$$

3.2.2. The C2fSESA Module

The C3 module of the YOLOv5 model has two branches—that is, one bottleneck is stacked with n modules, the other using only a basic convolution module—and finally performs the *concat* operation on the two branches, as shown in Figure 5a.



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

Figure 5. Construction process of the C2fSESA module. (**a**) Schematic diagram of C3 module; (**b**) schematic diagram of ELAN module; (**c**) schematic diagram of C2fSESA module.

In deep neural networks, gradients constitute important information for the training process, as they can guide the updating of model parameters. However, when the depth increases, the gradient information may become extremely sparse or discontinuous, making it difficult to train and converge. The efficient layer-aggregation network (ELAN) [16] module is an efficient layer-aggregation technology for deep neural networks. It can obtain richer gradient flow information by aggregating the features of different depth layers, and can solve the problem of difficult convergence when the depth model is extended. Its structure is shown in Figure 5b.

Consequently, the C2f module is designed by integrating the ideas of the C3 module and ELAN, and is able to obtain more abundant gradient flow information than the C3 module. The structure of the C2f module is as shown in Figure 5c.

The C2f module uses the *split–apply–concatenate* method for feature fusion. Thus, the features of the different branches can be fully integrated, which is conducive to the extraction of richer features. Additionally, we introduce the attention mechanism SESA into the bottleneck of the C2f module and propose the C2fSESA module. This module helps the model adaptively adjust the importance of each feature map, capture the local and global features of the object more accurately, and improve the detection accuracy of the model.

3.3. RGSPP Structure

Maritime images captured by shipborne UAVs often have different resolutions and scales, and changes in the height and angle of the UAVs can cause different representations of the same object at different scales.

This study used SPP layers to overcome the limitations of fully connected layers, which require fixed-size inputs. First, we added a residual edge of size 1×1. Subsequently, the three parallel *MaxPool* layers of sizes 5×5, 9×9, and 13×13 were transformed into three serial *MaxPool* layers of size 5×5 to reduce the model parameters and computational overhead. Subsequently, the SiLU activation function of *Conv* was changed to a ReLU activation function, simplifying the convolution. Finally, the simplified convolution was grouped again, each convolution kernel only acting on the channels in each group instead of all the input channels, thereby reducing the model's computational overhead and memory consumption.

However, group convolution can also slightly reduce the expressive ability of the model, as the number of channels in each group is the same. This may limit the model from learning the combination of features between channels, resulting in a certain loss of information. Introducing random operations based on group convolution is a means of enhancing the expression and generalization abilities of convolutional neural networks. Consequently, shuffled group convolution [44] was used to shuffle the output of each group to enhance the interaction between different groups. The number of calculations was relatively large, and the potential relationship between the channels could not be determined, as shown in Figure 6.

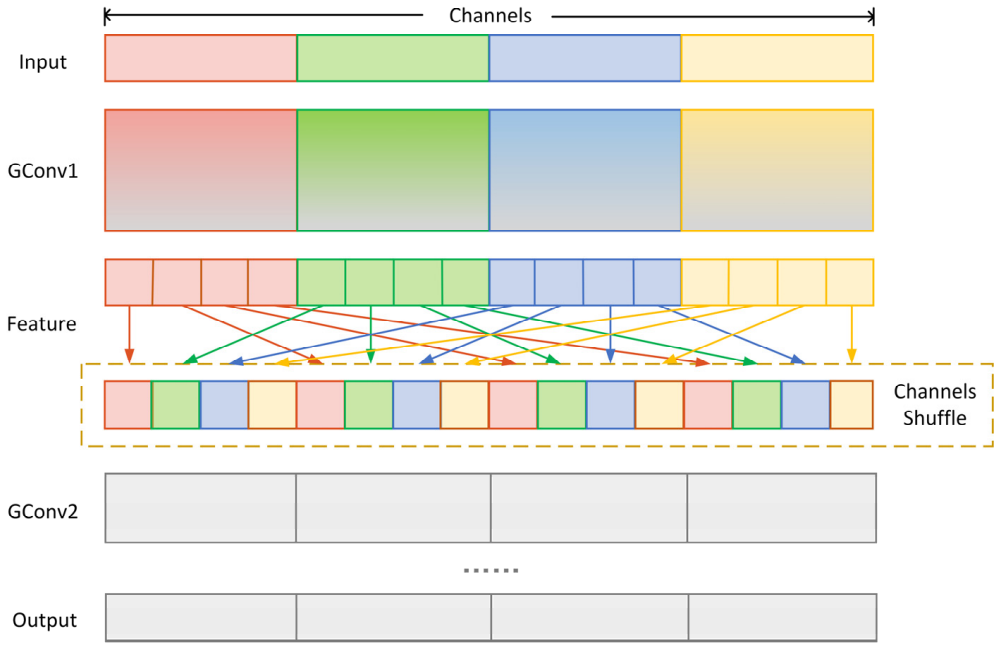


Figure 6. Schematic diagram of shuffled group convolution.

However, grouped convolution [45] and random channel rearrangement divide the input feature map into multiple groups and randomly rearrange the channels in each group to enhance the interaction between different channels while simultaneously reducing the dependence of the model on the channel order, as shown in Figure 7.

Therefore, based on the group convolution and random channel rearrangement methods, we designed a new spatial pyramid pooling structure—that is, RGSPP—as shown in Figure 8. Here, the convolution kernel is divided into four groups, each group being responsible for processing the corresponding input layer to enhance the generalization performance of the model.

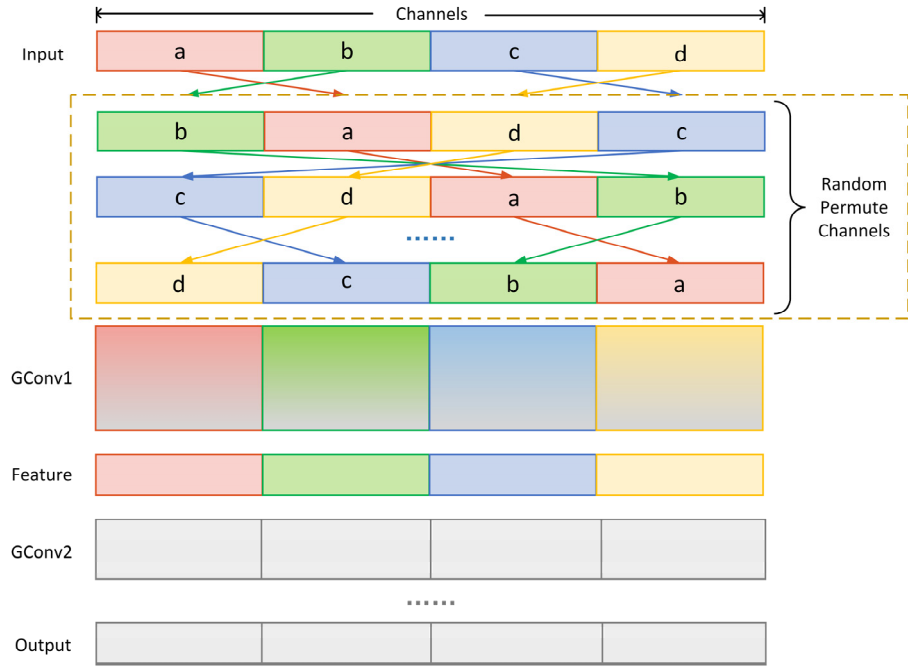


Figure 7. Schematic diagram of group convolution and random channel rearrangement.

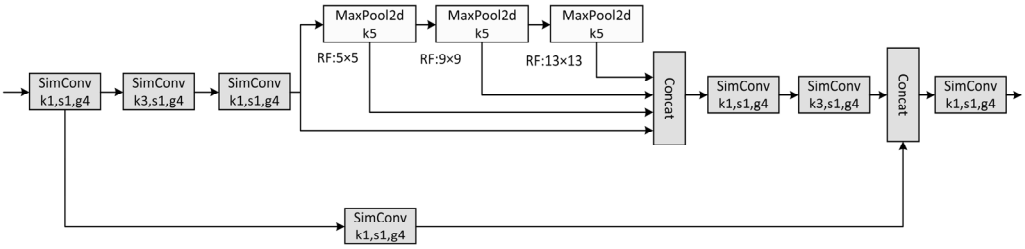


Figure 8. Schematic diagram of RGSPP structure.

We compared different SPP structures at baseline, with the parameters and GFLOPS (Giga Floating-point Operations Per Second) listed in Table 1. It is evident that the RGSPP parameters and calculations are reduced. Our experiments also proved that RGSPP is not only a lightweight spatial pyramid pooling structure but also improves the accuracy of object detection.

Table 1. Comparison of spatial pyramid pooling structure parameters and calculation.

Model	Parameters	GFLOPS
SPPF	7,033,114	16.0
SimSPPF	7,033,114	16.0
SPPFCSPC	13,461,274	21.1
RGSPP	6,888,218	15.9

4. Experiments and Results

This section briefly describes the datasets used in the study. We then present the details of the experimental design and results and analyze the performance of the proposed method by comparing it with other object-detection algorithms. For an objective and fair comparison, our experiments were conducted on an NVIDIA RTX 2060 GPU, using CUDA v11.0, cuDNN v8.1.1, PyTorch 1.7.1, and other configurations.

4.1. MOBDrone Dataset Based on an EO Sensor

The MOBDrone dataset [46] is a large-scale drone image dataset for MOB detection, collected using the DJI FC6310 camera of the Phantom 4 Pro V2 UAV. The dataset comprises 66 video clips. We divided the 66 videos into frames based on specific intervals and obtained 7805 images. After removing the empty maritime data, we divided the remaining data into training, verification, and test datasets at a ratio of 8:1:1. The tags included five types (person, boat, wood, lifebuoy, and surfboard); the number of labels for each category is listed in Table 2.

Table 2. Statistics and samples of object annotation information.

Class	Annotations	Samples
Person	11,687	
Boat	10,269	
Wood	2459	
Lifebuoy	1712	
Surfboard	3793	

To train and evaluate the mainstream object-detection network models better, we counted the large, medium, and small objects in each category based on the metrics of the Microsoft COCO challenge dataset (Table 3). As shown in Figure 9, the total number of small objects is 12,817 (42.8%); the total number of medium objects is 11,301 (37.7%); and the total number of large objects is 5802 (19.5%).

Table 3. Definition of large, medium, and small objects in the COCO dataset.

Objects	Metric (Square Pixels)
Small	Area < 32^2
Medium	32^2 < area < 96^2
Large	Area > 96^2

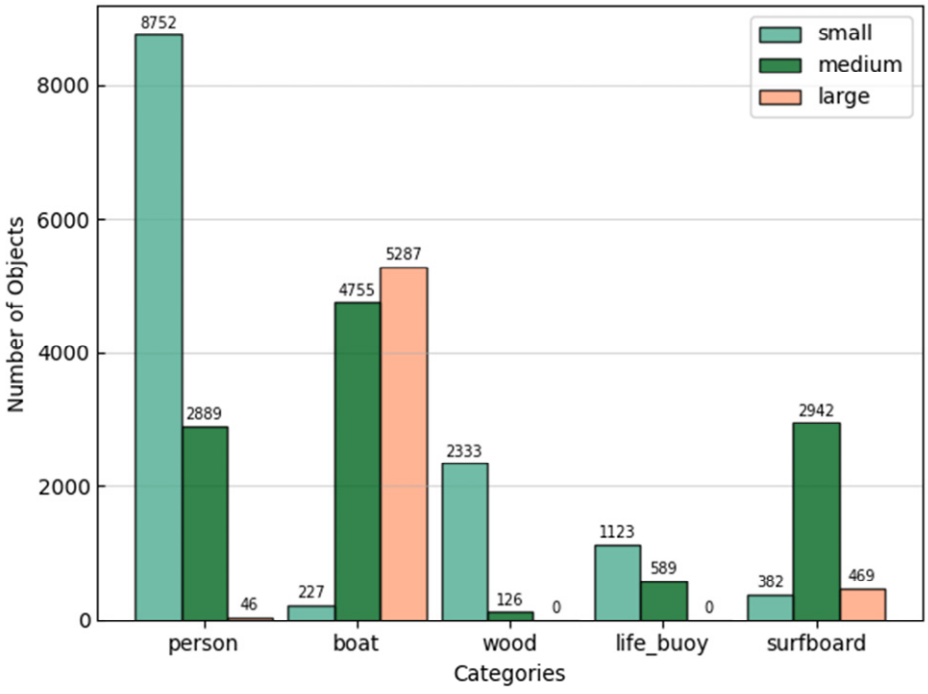


Figure 9. Statistics on the number of large, medium, and small objects of five label types.

4.2. The Evaluation Index

We used the MOBDrone dataset to test the performance of the YOLO-BEV model, comprehensively considering the performance of the model from different aspects, including detection accuracy and speed. Precision and recall were adopted as the accuracy indices. The calculation formula for precision can be expressed as shown in Equation (9):

$$P = \frac{TP}{TP + FP} \tag{9}$$

Recall can be calculated using Equation (10), as follows:

$$R = \frac{TP}{TP + FN} \tag{10}$$

where TP denotes a positive sample that is correctly predicted, FP is a negative sample that is incorrectly predicted, TN is a negative sample that is correctly predicted, and FN is a positive sample that is incorrectly predicted.

The aim is to obtain a model which can simultaneously achieve high precision and recall; therefore, these two factors must be considered comprehensively. One method uses the harmonic mean F1 for measurement, and the other uses the area under the P-R curve (AUC)—that is, the average precision (AP). The AP can be calculated as follows:

$$AP = \int_0^1 P(R)dR \tag{11}$$

The AP is for one category, and we can calculate the average precision of all categories according to Equation (12):

$$mAP = \frac{1}{|N|} \sum_{i \in N} AP(i) \tag{12}$$

where N denotes the number of label categories in the MOBDrone dataset.

Consequently, in this study, we used $mAP_{0.5:0.95}$, $mAP_{0.5}$, $mAP_{0.75}$, AP_S, AR_S, and frames per second (FPS) as the evaluation indices of the model to comprehensively evaluate its actual performance. Among them, $mAP_{0.5:0.95}$ is the average precision of all categories with 10 thresholds between 0.5 and 0.95 (step = 0.05), $mAP_{0.5}$ is the average precision of all categories with a threshold of 0.5, and $mAP_{0.75}$ is the average precision of all categories with a threshold of 0.75; AP_S and AR_S are the average precision and average recall for small objects with 10 thresholds between 0.5 and 0.95.

Additionally, the FPS is an important index for measuring the real-time performance of a computer vision system, indicating the number of frames processed per second. In the object-detection of shipborne UAV images, a high FPS value ensures that the system can respond to object changes in a timely manner, thereby ensuring real-time and accurate detection.

4.3. Performance of the Object-Detection Algorithm

This article conducted a series of simulation experiments on the MOBDrone dataset to demonstrate the effectiveness of the YOLO-BEV algorithm for small-object detection in shipboard UAV scenarios.

4.3.1. Ablation Experiment

To explore the impact of different components of the model on the performance of the algorithm, we conducted ablation experiments based on YOLOv5. We adjusted and changed several key modules in the algorithm and retrained and tested it to evaluate the contributions of the different modules. Table 4 lists the performance of the three improved methods based on the YOLOv5 model.

Table 4. YOLOv5 combines precision and recall performance of three improved methods.

Methods	P	C	R	AP_1	AP_2	AP_3	AP_S	AR_S
YOLOv5	×	×	×	0.526	0.931	0.483	0.333	0.431
MI	√	×	×	0.547	0.964	0.510	0.373	0.492
MII	√	√	×	0.557	0.969	0.513	0.390	0.498
MIII	√	√	√	0.564	0.971	0.536	0.407	0.513

Note: √: used; ×: not used; MI: Method I; MII: Method II; MIII: Method III; P: PAN+; C: C2fSESA; R: RGSPP; AP_1: $mAP_{0.5:0.95}$; AP_2: $mAP_{0.5}$; AP_3: $mAP_{0.75}$.

With the improvement in the method, the model achieves a substantial improvement in detection performance compared to the YOLOv5 baseline. In Method I, the $mAP_{0.5:0.95}$ value is 0.547, which is 4.0% higher than that at the baseline. Additionally, the AP_S value of Method I is 0.04 higher than that of YOLOv5, achieving a substantial increase of 12.0%, proving that the PAN+ module plays an important role in detecting small objects. Compared with Method I, the $mAP_{0.5:0.95}$, AP_S, and AR_S indicators of Method II improve by 1.8%, 4.6%, and 1.2%, respectively, proving that the C2fSESA module has rich gradient information and the ability to focus on feature maps. Method III adds the RGSPP module based on Method II, and the values of all evaluation indexes reach their highest values, with the RGSPP module exhibiting greater competitiveness.

As shown in Table 5, the $mAP_{0.5:0.95}$ value of the YOLO-BEV algorithm using the MOBDrone test dataset is 0.564, which is 7.2% higher than the $mAP_{0.5:0.95}$ value of YOLOv5. In particular, the YOLO-BEV algorithm performs even better for the detection of small objects. The AP_S value of the small object is 0.407, which is 22.2% higher than that of YOLOv5. Moreover, the AR_S of the YOLO-BEV algorithm is 19.0% higher than that of YOLOv5, showing a very strong recall ability for small objects. The $mAP_{0.5}$ and $mAP_{0.75}$ of the YOLO-BEV algorithm also increases by 4.3% and 11.0%, respectively, proving that its regression ability to the object-bounding box is more powerful. Additionally, the detection speed of the YOLO-BEV algorithm is maintained at 48 FPS, greatly improving the detection accuracy of small objects in shipborne UAV scenarios under the premise of satisfying real-time performance.

Table 5. Performance evaluation of YOLOv5 and YOLO-BEV on the MOBDrone dataset.

Methods	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.75}$	AP_S	AR_S
YOLOv5	0.526	0.931	0.483	0.333	0.431
YOLO-BEV	0.564	0.971	0.536	0.407	0.513

The precision, recall, $mAP_{0.5:0.95}$, and $mAP_{0.5}$ of the YOLOv5 and YOLO-BEV algorithms are as shown in Figure 10. It is generally believed that the closer the algorithm curve is to the top and the smaller the fluctuation, the better the overall performance. As shown in Figure 10, the four curves (precision, recall, $mAP_{0.5:0.95}$, and $mAP_{0.5}$) of the YOLO-BEV algorithm are all above those of the YOLOv5 algorithm and converge faster. In particular, for the $mAP_{0.5:0.95}$ curve, the increase is evident, verifying the high-precision performance of the YOLO-BEV algorithm proposed in this study.

Table 6 presents the comparison results of the average precision values for each category on the MOBDrone dataset for 10 thresholds between 0.5 and 0.95. From the analysis of the YOLOv5 and YOLO-BEV algorithms on the $AP_{0.5:0.95}$ value change of each category, the YOLO-BEV algorithm greatly improved the detection accuracy of small objects; in particular, for categories with many small objects, such as people and wood, the detection results show an increase of 22.2% and 7.9%, respectively. The $AP_{0.5:0.95}$ value for large-boat objects is only reduced by 0.4%. For categories with more medium-sized objects, such as life buoys and surfboards, the $AP_{0.5:0.95}$ value of the YOLO-BEV algorithm improves by 5.7% and 6.5%, respectively. Based on analysis of these results, we can conclude that the

YOLO-BEV algorithm maintains its accuracy for large objects while greatly improving its detection accuracy of small objects.

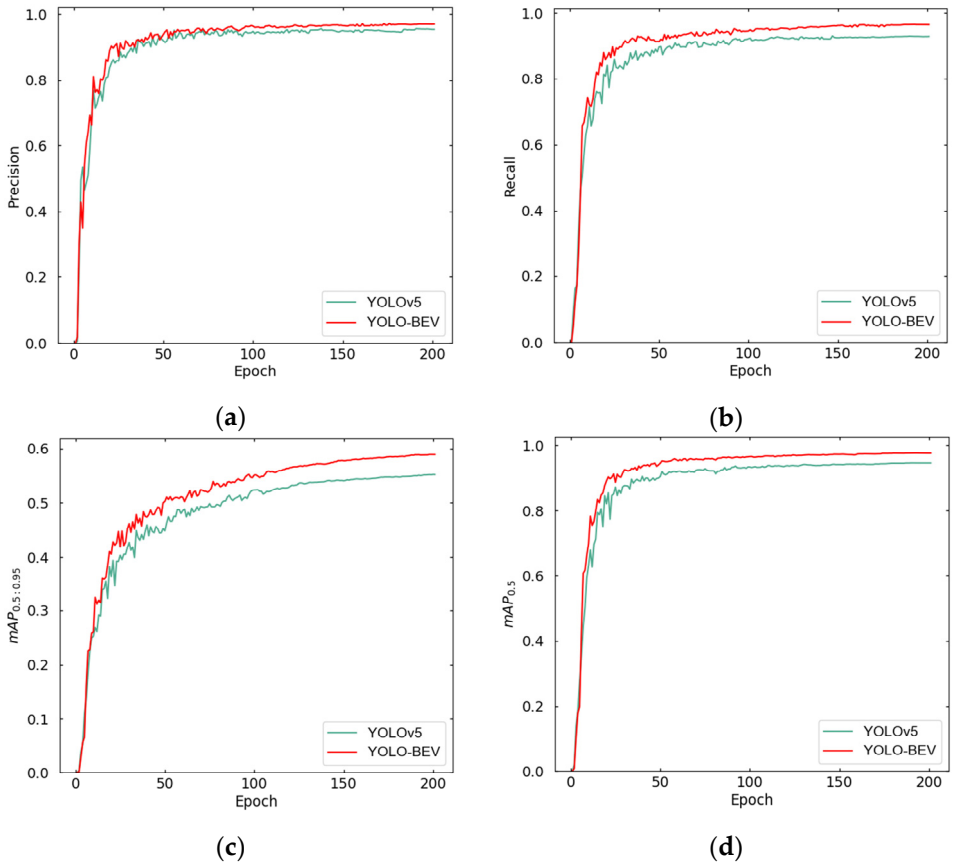


Figure 10. Comparison of YOLOv5 and YOLO-BEV evaluation indicators; (**a**–**d**) respectively show the comparison curves of *Precision*, *Recall*, $mAP_{0.5:0.95}$, and $mAP_{0.5}$.

Table 6. Comparison of average precision per category between YOLOv5 and YOLO-BEV on MOBDrone dataset.

Methods	Person	Boat	Wood	Surfboard	Life_Buoy
YOLOv5 ($AP_{0.5:0.95}$)	0.387	0.838	0.598	0.506	0.435
YOLO-BEV ($AP_{0.5:0.95}$)	0.473	0.835	0.645	0.539	0.460

4.3.2. Comparisons of the YOLO-BEV Algorithm with Other Object-Detection Algorithms

To further verify the high-precision performance of the YOLO-BEV algorithm, we compared it with the SSD, RetinaNet, Faster R-CNN, VarifocalNet, YOLOv3, and YOLOv7 algorithms. To objectively compare the performance of each detector on the MOBDrone dataset, the input resolution of the one-stage detector was adjusted to 416 × 416 px. The two-stage detector used the default resolution of the official release code. Additionally, the backbone network of the SSD algorithm was VGG16; that of RetinaNet, Faster R-CNN, and VarifocalNet algorithms was ResNet50; that of the YOLOv3 algorithm was DarkNet53; and that of the YOLOv5 and YOLOv7 algorithms was CSPDarkNet53.

Table 7 shows the detection performance comparison results of the YOLO-BEV algorithm and other deep-learning object-detection algorithms on the MOBDrone dataset. As is evident from Table 7, the $mAP_{0.5}$ value of the YOLO-BEV algorithm reaches 0.971, which is 4.3% higher than that of the YOLOv5 algorithm. The detection speed of the YOLO-BEV algorithm reaches 48 FPS. Although it is slightly lower than the 53 FPS of the YOLOv5

algorithm, it still meets the real-time requirements of 30 FPS for maritime automation. In the two-stage detection algorithm, both the Faster R-CNN and VarifocalNet algorithms achieve high detection accuracy; however, the detection speed is only 11 and 15 FPS, respectively, which does not meet real-time detection requirements. In the one-stage detection algorithm, although the SSD algorithm has an advantage in speed compared with the two-stage algorithm, it performs poorly in terms of accuracy. RetinaNet is a representative one-stage algorithm. Although the $mAP_{0.5:0.95}$ value reaches 0.549, the average precision and recall for small objects are low and cannot meet real-time detection requirements. Among the one-stage detection algorithms of the YOLO series, the YOLO-BEV algorithm achieves excellent competitive performance in terms of both detection accuracy and speed. Compared to the YOLOv3 algorithm, the $mAP_{0.5:0.95}$, $mAP_{0.5}$, and AP_S of the YOLO-BEV algorithm increases by 3.8%, 3.7%, and 15.3%, respectively, with the FPS rate increasing by 50%. Compared to the YOLOv7 algorithm, the $mAP_{0.5:0.95}$, $mAP_{0.5}$, and AP_S of the YOLO-BEV algorithm increases by 2.4%, 2.9%, and 13.4%, respectively, and the FPS rate increases by 30%. Three methods, namely YOLOv5, Fast R-CNN, and DETR, were used in reference [24]. Compared with *method 3* in reference [24], all evaluation indexes were improved, among which AP_S was increased by 12.7%. Although the method in reference [26] achieves 40 FPS in real-time, the average detection accuracy for small objects is only 0.350. Although the detection speed of the YOLO-BEV algorithm is slightly lower than that of the YOLOv5 algorithm, it is important to improve the detection accuracy of maritime objects to achieve real-time performance. In particular, when detecting a MOB in maritime conditions, one must accurately obtain the position of the MOB. In summary, the YOLO-BEV algorithm is a high-precision detection algorithm for small objects under maritime conditions and is suitable for shipborne UAV scenarios.

Table 7. Comparison of detection performance between YOLO-BEV and other deep-learning object-detection algorithms on MOBDrone dataset.

Detection	Backbone	AP_1	AP_2	AP_3	AP_S	AR_S	FPS
SSD	(1)	0.455	0.897	0.418	0.227	0.360	22
RetinaNet	(2)	0.549	0.939	0.522	0.346	0.437	26
Faster R-CNN	(2)	0.554	0.953	0.495	0.366	0.470	11
VarifocalNet	(2)	0.560	0.960	0.529	0.397	0.494	15
YOLOv3	(3)	0.544	0.936	0.519	0.353	0.450	32
YOLOv5	(4)	0.526	0.931	0.483	0.333	0.431	53
YOLOv7	(4)	0.551	0.944	0.528	0.359	0.465	37
[24]	(5)	0.551	0.957	0.522	0.361	0.474	23
[26]	(3)	0.547	0.940	0.491	0.350	0.463	40
YOLO-BEV	(4)	0.564	0.971	0.536	0.407	0.513	48

Note: (1): VGG16; (2): ResNet50; (3): DarkNet53; (4): CSPDarkNet53; (5): Transformer; AP_1: $mAP_{0.5:0.95}$; AP_2: $mAP_{0.5}$; AP_3: $mAP_{0.75}$.

4.4. Visualization of Some Detection Results

Figure 11 shows the object recognition results of the UAVs at different heights (30–60 m) in an actual sea area. As shown in Figure 11, even under conditions which include illumination and angle changes, the proposed algorithm exhibits excellent performance for small-object detection in maritime applications. However, given a novel algorithm, more test conditions need to be considered to evaluate the actual usability of the algorithm. Since it is extremely difficult and risky to record a maritime dataset in reduced visibility, we used simulated datasets [47] for evaluation. Figure 12 illustrates the visualization of object detection results using the YOLO-BEV algorithm under deteriorating visibility ((a) storms and heavy waves; (b) dim; (c) dark lighting; (d) fog). The proposed algorithm shows good performance under challenging conditions such as dim or dark light, fog, storm, or heavy waves.

Figure 11. Visualization of object detection results of YOLO-BEV at different heights (30–60 m) of UAVs. The red bar represents the detected person, the pink bar represents the detected boat, the yellow bar represents the detected surfboard, the orange bar represents the detected life_buoy, and the brown bar represents the detected wood.

Figure 12. Visualization of object detection results of YOLO-BEV under deteriorating visibility. The red bar represents the detected person, the pink bar represents the detected boat, the yellow bar represents the detected surfboard, the orange bar represents the detected life_buoy, and the brown bar represents the detected wood.

4.5. Architecture of the System

The shipborne UAV maritime object-detection system comprises a UAV terminal, data communication module, shipboard computing terminal, object-detection algorithm, interactive interface, and an airborne global positioning system (GPS) module. The UAV obtains a maritime image sequence through an EO sensor and transmits it to the ship via data communication. The object-detection algorithm on the shipboard computing terminal automatically detects and identifies objects in the image, and the airborne GPS module accurately locates them, after which the object information is displayed on the interactive interface. Figure 13 shows the graphical user interface (GUI), including three components: an image transmission video link address area, an airborne video recognition result display area, and an airborne GPS position return area.

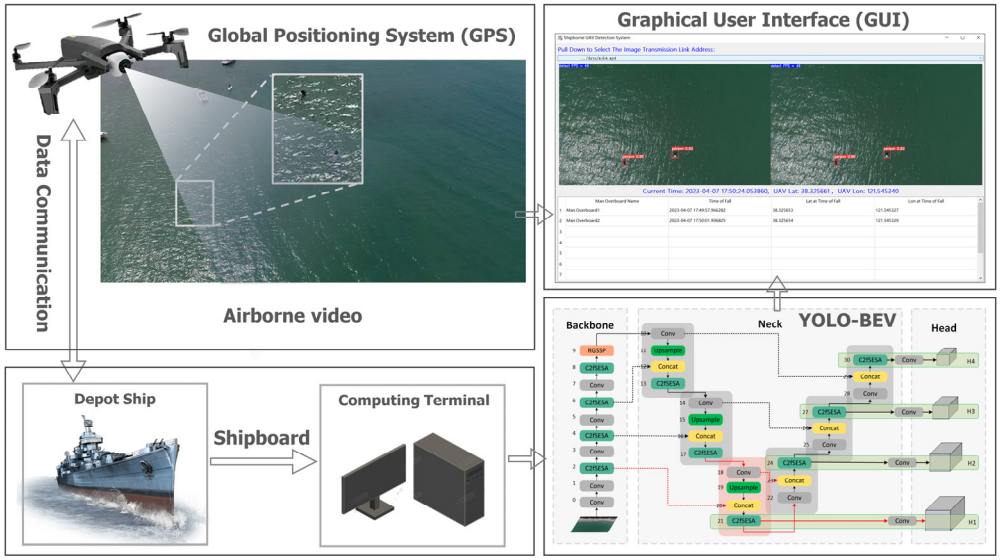


Figure 13. Architecture of shipborne UAV object-detection system.

5. Discussion

5.1. Discussion of Results

The iteration of AI intelligent algorithms is very important for the study of small-object detection from the perspective of shipborne UAVs. Although many studies have explored the problem of maritime object recognition by shipborne UAVs, these studies have not fully considered its accuracy and real-time. In order to solve this problem, a high-precision and real-time detection system is proposed in this study, which is suitable for small object recognition by shipborne UAVs. In the ablation experiment in Table 4, the influence of each improved module on the performance of the algorithm in this paper is verified, thus proving the effectiveness of the improvement of each key module. It can be seen from Tables 5 and 6 that through a series of improvements and optimization, the detection accuracy of the proposed algorithm reaches the highest level, which proves the superiority of the proposed algorithm in detecting targets from the perspective of maritime UAVs. In Table 7, we compare the algorithm with other methods. Experimental results show that the accuracy of the proposed algorithm achieves a leading edge. Although the real-time performance is slightly lower than that of YOLOv5, the speed of the proposed algorithm is still ahead of other algorithms, and it meets the requirement of 30 FPS for maritime automatic driving. In order to further prove the generalization performance of the algorithm, we carried out simulation tests under deteriorating visibility (dim or dark lighting and in the presence of fog, storms, or heavy waves), and prove the applicability of the algorithm under reduced visibility. While the test results of virtual datasets may not be fully representative of what happens under actual sea conditions, they can provide initial insights into how our methods perform in harsh environments.

5.2. Limitations

Given a novel algorithm, it is necessary to evaluate the actual usability of the algorithm in its real environment. The proposed algorithm was tested in the fields of sea surface glare reflection, sea water texture change, UAV height and angle change, etc., and the practical usability of the proposed algorithm was verified in these scenarios. However, under deteriorating visibility (dim or dark lighting and in the presence of fog, storms, or heavy waves), this kind of consideration would require many more test conditions and possible real implementation and testing over several seasons. In Section 4, although we used virtual datasets to test applicability under reduced visibility, we also acknowledge that the test results of virtual datasets cannot fully represent what happens under actual sea states. Therefore, it is necessary to further enhance the dataset under different sea conditions, enhance the scale, diversity, and representativeness of the dataset, and improve the generalization performance of the model.

In addition, the algorithm parameters used in the paper will have an impact on the results. Different parameter settings will result in different detection performance, so more parameter tuning is required.

6. Conclusions

This paper presents a novel approach for detecting small maritime objects from the perspective of UAVs, utilizing an enhanced detection model based on YOLOv5, CSPDarknet53, FPN, and PAN. The proposed model incorporates three key improvements to address the unique characteristics of the maritime environment. Through experimental verification on the MOBDrone dataset, compared with YOLOv5, its detection accuracy of small objects is greatly improved by 22.2%, and the inference speed of the model reaches 57 FPS, which proves that the model can obtain higher detection accuracy and real-time performance. In addition, this paper provides an overview of the architecture of a shipborne UAV maritime object-detection system, equipping researchers with the necessary insight to better design shipborne UAV maritime object-detection systems. Finally, algorithmic applications for shipborne UAVs usually require real-time processing of large amounts of data, such as sensor data (e.g., images). Therefore, in future work, converting the model into TensorRT format can significantly improve the inference speed and performance of the model, and meet the engineering application requirements of maritime shipborne UAVs.

Author Contributions: Conceptualization, Y.Y. and Q.J.; methodology, Z.Y.; software, Z.Y.; validation, Y.Y., Z.Y. and Q.J.; data curation, Z.S.; writing—original draft preparation, Z.Y.; writing—review and editing, Z.S.; visualization, Z.Y.; supervision, Y.Y. and Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under Grant 2022YFB4301402, the 2022 Liaoning Provincial Science and Technology Plan (Key) Project: R&D and Application of Autonomous Navigation System for Smart Ships in Complex Waters under Grant 2022JH1/10800096, the Fundamental Research Funds for the Central Universities under Grant No.3132023139, and the International cooperation training program for innovative talents of Chinese Scholarships Council under Grant No. CSC [2022] 2260.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

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

References

1. Leira, F.S.; Helgesen, H.H.; Johansen, T.A.; Fossen, T.I. Object Detection, Recognition, and Tracking from UAVs Using a Thermal Camera. *J. Field Robot.* **2021**, *38*, 242–267. [CrossRef]
2. Chen, X.; Li, Z.; Yang, Y.; Qi, L.; Ke, R. High-Resolution Vehicle Trajectory Extraction and Denoising from Aerial Videos. *IEEE Trans. Intell. Transport. Syst.* **2021**, *22*, 3190–3202. [CrossRef]

3. Guo, Q.; Liu, J.; Kaliuzhnyi, M. YOLOX-SAR: High-Precision Object Detection System Based on Visible and Infrared Sensors for SAR Remote Sensing. *IEEE Sens. J.* **2022**, *22*, 17243–17253. [CrossRef]
4. Tan, Y.; Li, G.; Cai, R.; Ma, J.; Wang, M. Mapping and Modelling Defect Data from UAV Captured Images to BIM for Building External Wall Inspection. *Autom. Constr.* **2022**, *139*, 104284. [CrossRef]
5. Gonçalves, J.A.; Henriques, R. UAV Photogrammetry for Topographic Monitoring of Coastal Areas. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 101–111. [CrossRef]
6. Lyu, H.; Shao, Z.; Cheng, T.; Yin, Y.; Gao, X. Sea-Surface Object Detection Based on Electro-Optical Sensors: A Review. *IEEE Intell. Transport. Syst. Mag.* **2023**, *15*, 190–216. [CrossRef]
7. Stojnić, V.; Risojević, V.; Muštra, M.; Jovanović, V.; Filipi, J.; Kezić, N.; Babić, Z. A Method for Detection of Small Moving Objects in UAV Videos. *Remote Sens.* **2021**, *13*, 653. [CrossRef]
8. Wang, S.; Han, Y.; Chen, J.; He, X.; Zhang, Z.; Liu, X.; Zhang, K. Weed Density Extraction Based on Few-Shot Learning Through UAV Remote Sensing RGB and Multispectral Images in Ecological Irrigation Area. *Front. Plant Sci.* **2022**, *12*, 735230. [CrossRef]
9. Yahyanejad, S.; Rinner, B. A Fast and Mobile System for Registration of Low-Altitude Visual and Thermal Aerial Images Using Multiple Small-Scale UAVs. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 189–202. [CrossRef]
10. Kaljahi, M.A.; Shivakumara, P.; Idris, M.Y.I.; Anisi, M.H.; Lu, T.; Blumenstein, M.; Noor, N.M. An Automatic Zone Detection System for Safe Landing of UAVs. *Expert Syst. Appl.* **2019**, *122*, 319–333. [CrossRef]
11. Harel, J.; Koch, C.; Perona, P. Graph-Based Visual Saliency. In *Advances in Neural Information Processing Systems 19*; Schölkopf, B., Platt, J., Hofmann, T., Eds.; The MIT Press: Cambridge, MA, USA, 2007; pp. 545–552; ISBN 978-0-262-25691-9.
12. Duan, H.; Xu, X.; Deng, Y.; Zeng, Z. Unmanned Aerial Vehicle Recognition of Maritime Small-Target Based on Biological Eagle-Eye Vision Adaptation Mechanism. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3368–3382. [CrossRef]
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37.
14. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
17. 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.
18. 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]
19. Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. VarifocalNet: An IoU-Aware Dense Object Detector. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8510–8519.
20. Yang, J.; Xie, X.; Shi, G.; Yang, W. A Feature-Enhanced Anchor-Free Network for UAV Vehicle Detection. *Remote Sens.* **2020**, *12*, 2729. [CrossRef]
21. Liu, M.; Wang, X.; Zhou, A.; Fu, X.; Ma, Y.; Piao, C. UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective. *Sensors* **2020**, *20*, 2238. [CrossRef]
22. Ye, T.; Qin, W.; Li, Y.; Wang, S.; Zhang, J.; Zhao, Z. Dense and Small Object Detection in UAV-Vision Based on a Global-Local Feature Enhanced Network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
23. Wang, C.; Shi, Z.; Meng, L.; Wang, J.; Wang, T.; Gao, Q.; Wang, E. Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism. *Drones* **2022**, *6*, 149. [CrossRef]
24. Chen, M.; Sun, J.; Aida, K.; Takefusa, A. Weather-Aware Object Detection Method for Maritime Surveillance Systems. Available online: https://ssrn.com/abstract=4482179 (accessed on 1 August 2023).
25. Ye, T.; Qin, W.; Zhao, Z.; Gao, X.; Deng, X.; Ouyang, Y. Real-Time Object Detection Network in UAV-Vision Based on CNN and Transformer. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [CrossRef]
26. Sharafaldeen, J.; Rizk, M.; Heller, D.; Baghdadi, A.; Diguet, J.-P. Marine Object Detection Based on Top-View Scenes Using Deep Learning on Edge Devices. In Proceedings of the 2022 International Conference on Smart Systems and Power Management (IC2SPM), Beirut, Lebanon, 10 November 2022; pp. 35–40.
27. Cai, Y.; Luan, T.; Gao, H.; Wang, H.; Chen, L.; Li, Y.; Sotelo, M.A.; Li, Z. YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
28. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
29. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
30. Bochkovskiy, A.; Wang, C.Y.; Liao, H.J.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.

31. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
32. Ghiasi, G.; Lin, T.Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15 June 2019; pp. 7029–7038.
33. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
34. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
35. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
36. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
37. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
38. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 13708–13717.
39. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19; ISBN 978-3-030-01233-5.
40. Zhou, K.; Tong, Y.; Li, X.; Wei, X.; Huang, H.; Song, K.; Chen, X. Exploring Global Attention Mechanism on Fault Detection and Diagnosis for Complex Engineering Processes. *Process Saf. Environ. Prot.* **2023**, *170*, 660–669. [CrossRef]
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *ECCV Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1094–1916. [CrossRef]
42. Qiu, M.; Huang, L.; Tang, B.-H. ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion. *Remote Sens.* **2022**, *14*, 3498. [CrossRef]
43. 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. Computer Vision and Pattern Recognition (CVPR). *arXiv* **2022**, arXiv:2209.02976.
44. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
45. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM.* **2017**, *60*, 84–90. [CrossRef]
46. Cafarelli, D.; Ciampi, L.; Vadicamo, L.; Gennaro, C.; Berton, A.; Paterni, M.; Benvenuti, C.; Passera, M.; Falchi, F. MOBDrone: A Drone Video Dataset for Man OverBoard Rescue. In *Image Analysis and Processing—ICIAP 2022*; Sclaroff, S., Distante, C., Leo, M., Farinella, G.M., Tombari, F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2022; Volume 13232, pp. 633–644. ISBN 978-3-031-06429-6.
47. Kiefer, B.; Ott, D.; Zell, A. Leveraging Synthetic Data in Object Detection on Unmanned Aerial Vehicles. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2021.