remote sensing

MDPI

Technical Note

SG-Det: Shuffle-GhostNet-Based Detector for Real-Time Maritime Object Detection in UAV Images

Lili Zhang [1], Ning Zhang [1], Rui Shi [2,*], Gaoxu Wang [2], Yi Xu [2] and Zhe Chen [1]

[1] College of Computer and Information Engineering, Hohai University, Nanjing 211100, China; 211307040026@hhu.edu.cn (N.Z.)

[2] State Key Laboratory of Hydrology—Water Resources and Hydraulic Engineering, Nanjing Hydraulic Research Institute, Nanjing 210029, China

* Correspondence: rshi@nhri.cn

Abstract: Maritime search and rescue is a crucial component of the national emergency response system, which mainly relies on unmanned aerial vehicles (UAVs) to detect objects. Most traditional object detection methods focus on boosting the detection accuracy while neglecting the detection speed of the heavy model. However, improving the detection speed is essential, which can provide timely maritime search and rescue. To address the issues, we propose a lightweight object detector named Shuffle-GhostNet-based detector (SG-Det). First, we construct a lightweight backbone named Shuffle-GhostNet, which enhances the information flow between channel groups by redesigning the correlation group convolution and introducing the channel shuffle operation. Second, we propose an improved feature pyramid model, namely BiFPN-tiny, which has a lighter structure capable of reinforcing small object features. Furthermore, we incorporate the Atrous Spatial Pyramid Pooling module (ASPP) into the network, which employs atrous convolution with different sampling rates to obtain multi-scale information. Finally, we generate three sets of bounding boxes at different scales—large, medium, and small—to detect objects of different sizes. Compared with other lightweight detectors, SG-Det achieves better tradeoffs across performance metrics and enables real-time detection with an accuracy rate of over 90% for maritime objects, showing that it can better meet the actual requirements of maritime search and rescue.

Keywords: object detection; UAV images; lightweight network; maritime search and rescue

check for updates

Citation: Zhang, L.; Zhang, N.; Shi, R.; Wang, G.; Xu, Y.; Chen, Z. SG-Det: Shuffle-GhostNet-Based Detector for Real-Time Maritime Object Detection in UAV Images. *Remote Sens.* **2023**, *15*, 3365. https://doi.org/10.3390/rs15133365

Academic Editor: Shou Feng

Received: 11 May 2023
Revised: 28 June 2023
Accepted: 29 June 2023
Published: 30 June 2023

1. Introduction

In recent years, maritime accidents that have occurred globally have imposed a considerable toll on human society. Since 2014, maritime accidents have gradually increased, with approximately 4000 fatalities estimated per year [1]. Maritime search and rescue, a vital part of the national emergency response system, faces the main challenge of locating and finding objects at sea quickly and accurately. With the development of UAV technology, UAVs are highly effective in detecting objects for maritime SAR due to their advantages such as agility, portability, and air accessibility [2].

With the enhancement of computer hardware performance and the expansion of data volume, deep learning [3] has evolved into a powerful machine technique, which is extensively applied in domains such as video monitoring [4], self-driving [5], and facial recognition [6]. With the rapid development of deep learning, UAVs are increasingly integrated with object detection technology, making them more intelligent and efficient and widespread in fields such as disaster search and rescue [7], agricultural monitoring [8], and land surveying [9]. Deep learning-based object detection is a crucial task for computer vision and a vital technical enabler for the development of UAVs.

Deep learning-based object detectors are commonly categorized into one-stage and two-stage object detectors. The standard two-stage object detectors include R-CNN [10], SPP-Net [11], Fast R-CNN [12], and Faster R-CNN [13]. The R-CNN method employs the selective search algorithm to extract proposals from the original image, followed by feature extraction and support vector machine classification for each region proposal. SPP-Net adds a spatial pyramid pooling layer to the end of the CNN network, enabling the network to accept images of arbitrary sizes and pool feature maps of different sizes into a pyramid structure, thus ensuring consistent input sizes for the fully connected layer. Fast R-CNN introduces a Region of Interest pooling layer based on the spatial pyramid pooling module for feature mapping and utilizes a multi-task loss function to train for classification and localization tasks simultaneously. Faster R-CNN introduces an innovative approach by incorporating a region proposal network (RPN) into the network architecture, eliminating the need for the selective search algorithm to generate region proposals. This advancement enables end-to-end training, seamlessly and efficiently integrating the entire network. The standard one-stage object detectors include YOLO [14] and SSD [15]. YOLO utilizes the whole image as the input to the network and divides the image into several grid cells, followed by each grid cell predicting the position of the bounding box and the corresponding classification confidence. SSD uses a set of multi-scale feature maps to predict objects of different sizes, with shallow feature maps used for predicting smaller objects and deeper feature maps for larger ones. Additionally, it generates prior boxes for each pixel on the feature map to aid in the prediction process.

Although these detectors generally perform well, it is difficult for them to effectively solve the challenge of marine object detection on UAV platforms. The poor performance of general target detectors on UAV platforms may be due to the scale difference of UAV images, the complex background of UAV images, and the hardware constraints of UAV platforms. Researchers have addressed this problem in recent years by proposing corresponding object detection algorithms and models. Wang [16] et al. designed a UAV visual navigation and control system for maritime search and rescue, which provides stable and accurate position and target estimation based on the space Softmax layer and a specially designed convolutional layer. Zhao [17] et al. proposed YOLOV7-SEA for target detection of marine UAV images based on YOLOv7. Detection heads were added to detect small-size targets and attention mechanisms were used to emphasize important features. Tran [18] et al. proposed a method for detecting bottle marine waste based on machine learning and UAV and added a data enhancement method and image processing for background removal to optimize waste detection. Lu [19] et al. proposed the improved YOLOv5 algorithm to enhance the effectiveness of UAV marine fishery law enforcement and effectively integrated features of different scales by changing the feature extraction scale and increasing the path aggregation network.

Due to UAVs' hardware limitations and application scenarios, UAV image object detectors must possess lightweight and real-time capabilities. Generally speaking, pursuing speed can lead to a loss of accuracy and vice versa. As for how to improve the efficiency of object detection in UAV images, this study mainly focuses on the following two aspects: (1) reducing the size of the detector through lightweight design; (2) improving the detection accuracy of small objects. Based on the above discussion, this paper introduces a novel one-stage lightweight object detector called SG-Det, specifically designed to detect objects in UAV images for maritime SAR. Firstly, we propose Shuffle-GhostNet as the backbone of the detector. Considering the impact of the model on detection speed, we choose a lightweight classification network as the detector's backbone. We reconstruct the module based on GhostNet [20] with a stricter design concept and add a channel shuffle operation to form Shuffle-GhostNet. Secondly, we propose BiFPN-tiny and integrate it with the ASPP [21] module to form the neck of the detector. To extract small object features from UAV images and reduce model complexity, we propose BiFPN-tiny by modifying the original five feature extraction layers into three feature extraction layers and one feature enhancement layer. The feature enhancement layer only enhances small object scale features and does not participate in prediction. To complement the feature extraction capability of the model, we replace the 1 × 1 convolutional adjustment module before the Bidirectional Feature

Pyramid Network (BiFPN) [22] with the ASPP module. Finally, in the head of the detector, we generate three sets of bounding boxes at different scales—large, medium, and small—to detect objects of different sizes.

The main contributions of this paper are as follows:

(1) We propose a lightweight object detector named SG-Det, which meets the requirements of high precision and high-speed detection of UAV images in SAR.
(2) We design a lightweight classification network named Shuffle-GhostNet, refactor the original GhostNet, and introduce the channel shuffle operation to enhance the flow between information groups and the robustness of the network model.
(3) We design a lightweight feature fusion architecture named BiFPN-tiny, which enhances the corresponding features to capture the characteristics of small, dense objects in UAV images.
(4) We validate the effectiveness of the proposed network on the aerial-drone floating objects (AFO) dataset, demonstrating its ability to achieve real-time detection with high accuracy.

2. Related Work

2.1. Lightweight Neural Network

Although the one-stage detector reduces the model size, it still cannot achieve real-time detection of UAV images due to many parameters. To address the issue of large-scale neural network models, many scholars have proposed their lightweight network architectures. The lightweight neural network is a specialized neural network structure for efficient computation and low-delay reasoning. It is particularly suitable for scenarios with limited computing resources like mobile devices. MobileNet [23] introduces the concept of depthwise separable convolution, which decomposes standard convolution into depthwise and pointwise convolution. This decomposition allows MobileNet to reduce computation and parameters while significantly maintaining high accuracy. SqueezeNet [24] utilizes two t types of convolutional layers: squeeze and expand. The squeeze layers decrease the number of channels, while the expand layers increase the number of channels and augment the depth of the feature map. ShuffleNet [25] combines group convolution and channel shuffling operations to achieve high accuracy while maintaining low computational costs. Leveraging the parallel design principles of both MobileNet and Transformer, Mobile-Former [26] achieves a seamless fusion of local and global features, exhibiting remarkable prowess in classification and various downstream tasks. ConvNext [27] harnesses the immense potential of multi-scale feature information by employing parallel combination, group convolution, and cross-path design, significantly enhancing efficiency and scalability. GhostNet introduces the Ghost module to extract redundant features from the original features using cost-effective operations, allowing the model to effectively utilize and embrace these redundant features while minimizing computational cost. In this paper, we introduce a modified version of GhostNet as the backbone of our network.

2.2. Feature Pyramid Network

In object detection and semantic segmentation tasks, objects and backgrounds typically appear at varying scales, requiring the image to be processed at multiple scales for optimal detection and segmentation results. The Feature Pyramid Network was proposed to address this issue, which can extract rich feature information at different scales and fuse this information to achieve superior object detection and semantic segmentation performance. The Feature Pyramid Network (FPN) [28] introduces a top-down approach for integrating high-level features with low-level features, combining low-resolution feature maps with rich semantic information and high-resolution feature maps with rich spatial information. The Path Aggregation Network (PANet) [29] builds upon the foundation of FPN. Furthermore, it optimizes the feature pyramid structure, introducing a novel feature aggregation strategy to achieve more accurate and efficient target detection. The Neural Architecture Search Feature Pyramid Network (NAS-FPN) [30] employs a neural network automatic search method to discover the optimal feature pyramid structure by exploring various network structures, enabling the feature pyramid structure to exhibit superior performance in object detection and semantic segmentation tasks. The Bidirectional Feature Pyramid Network (Bi-FPN) improves the feature pyramid's capability by enabling top-down and bottom-up information flow, resulting in more accurate and efficient object detection. The Channel Enhancement Feature Pyramid Network (CE-FPN) [31], inspired by subpixel convolution, proposes a sub-pixel skip fusion method that realizes both channel enhancement and upsampling. In this paper, we propose a lightweight BiFPN explicitly designed for UAV image detection, focusing on enhancing the features of small objects.

2.3. Group Convolution

Group convolution was initially used in AlexNet [32] to address the issue of insufficient video memory, and it is currently utilized in various lightweight modules to minimize the number of operations and parameters, as shown in Figure 1. This method splits the input feature map evenly into multiple groups based on the number of channels, followed by a conventional convolution on each group, supposing that the input feature map $X \in R^{C\times H\times W}$, C denotes the number of channels of the input feature map and H and W denote the height and width of the input feature map, respectively. At the same time, the input feature map $Y \in R^{C'\times H'\times W'}$, C' denotes the number of channels of the output feature map and H' and W' denote the height and width of the output feature map, respectively. The computation of conventional convolution is computed as:

$$N = H' \times W' \times C \times C' \times k \times k \tag{1}$$

where k denotes the height and width of the convolution kernel.

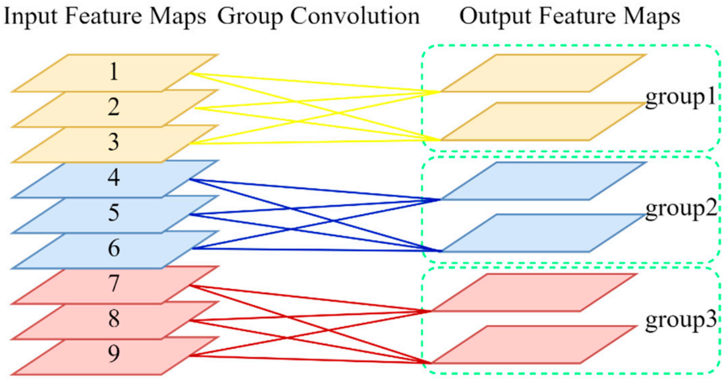

Figure 1. Group convolution.

The computation of group convolution is computed as:

$$N' = g \times H' \times W' \times \frac{C}{g} \times \frac{C'}{g} \times k \times k = \frac{1}{g} \times H' \times W' \times C \times C' \times k \times k \tag{2}$$

where g denotes the number of groups the input feature map is divided into, C/g denotes the number of channels in each group of the input feature map, and C'/g denotes the number of channels in each group of the output feature map. The group convolution reduces the computation of the conventional convolution to $1/g$ and reduces the number of parameters to $1/g$. However, it is essential to note that each group's convolution kernel only convolves with the input feature map of the same group, not with the input feature map of other groups. We leverage group convolution in several components of our object detector to reduce the number of parameters and computational complexity, resulting in faster training and inference.

2.4. Ghost Convolution

GhostNet introduced Ghost convolution as a cost-effective linear operation to generate feature maps, effectively reducing model parameters and computational workload. Figure 2 illustrates the concept of Ghost convolution, which involves dividing the traditional convolution operation into the primary convolution and the cheap convolution. The primary convolution is essentially the same as conventional convolution, but it strictly limits the total number of convolution kernels to be much smaller than that of conventional convolution. In contrast, cheap convolution utilizes the original feature map obtained by primary convolution to perform group convolution, generating redundant feature maps known as Ghost feature maps. Group convolution involves less computation and operates faster than conventional convolution, significantly reducing the model's complexity. The primary and Ghost feature maps are combined to obtain output feature maps sufficient for feature extraction. In this approach, the primary and Ghost feature maps are kept the same size. In order to prevent an excessive number of parameters from being generated in our object detector, we employed Ghost convolution multiple times throughout the network.

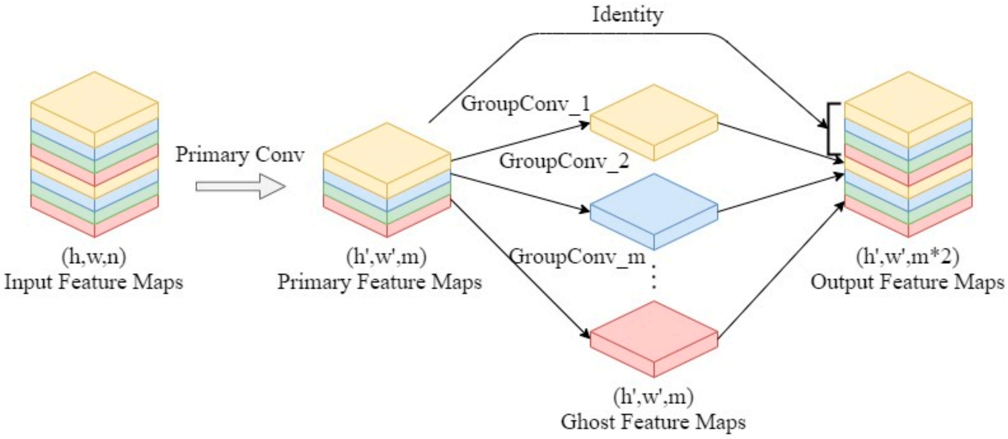

Figure 2. Ghost convolution.

3. Methods

3.1. Overall Framework

Figure 3 illustrates the architecture of our proposed SG-Det, which follows the one-stage detection principle by dividing the network into three parts: the backbone, neck, and head. In the backbone network section, we introduced a novel lightweight classification network called Shuffle-GhostNet, which consists of multiple Shuffle-Ghost bottlenecks. The structure of the Shuffle-Ghost bottleneck resembles a residual network consisting of two stacked Shuffle-Ghost modules. Additionally, the channel shuffle operation is introduced to improve information flow between different groups of features. In the neck network section, we constructed a four-layer feature pyramid by combining the BiFPN-tiny and ASPP modules, with three layers used for feature extraction and one layer dedicated to enhancing features related to small objects. Low-level features capture intricate details, such as local features and textures, whereas high-level features focus on extracting global semantic information and abstract features. To harness both strengths, the neck network integrates low-level and high-level features, resulting in improved performance. In the head network section, we generated three sets of boundary boxes at different scales—large, medium, and small—to enable the detection of objects of varying sizes.

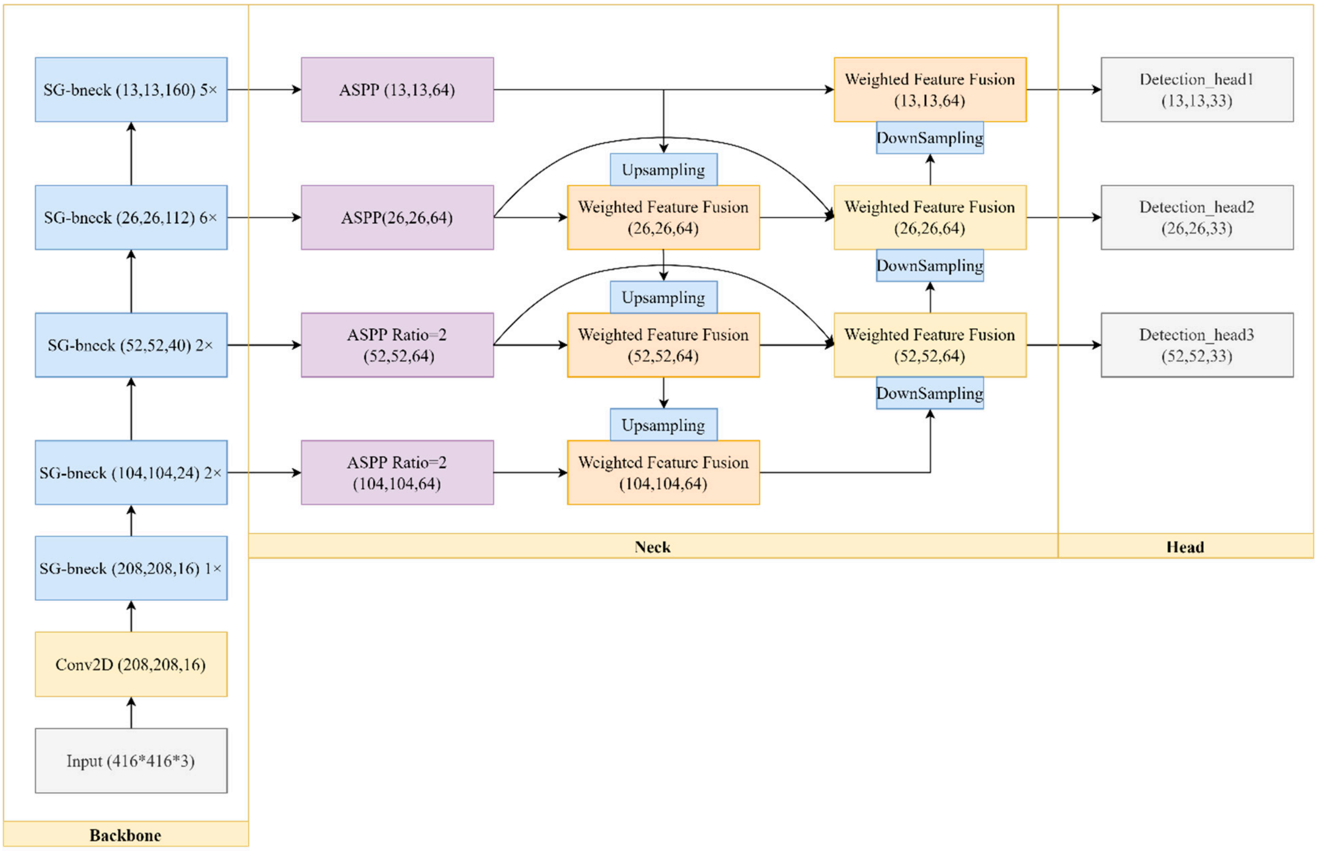

Figure 3. Overall framework diagram.

3.2. Backbone

The backbone of a neural network often employs operations such as convolution and pooling to extract features of various levels from input images. To achieve a practical and efficient backbone, we propose a novel lightweight classification network called Shuffle-GhostNet. Mainstream convolutional networks tend to generate a considerable amount of redundant intermediate feature maps during the calculation process. Figure 4 displays a visualization of some intermediate feature maps from Shuffle-GhostNet. Similar feature maps are marked with boxes of the same color, indicating that these pairs of feature maps are redundant. Feature map pairs that exhibit similarity are referred to as “ghosts”, and these redundant intermediate feature maps play an indispensable role in enhancing the feature extraction ability of the model during actual reasoning tasks. The core concept behind Shuffle-GhostNet is that, while redundant intermediate feature maps are necessary and cannot be eliminated, the convolution operations required to generate these feature maps can be accomplished using lighter methods.

In the original GhostNet, group convolution and depthwise convolution are utilized in several locations to significantly decrease the computational complexity of the model compared to traditional object detection methods. However, some modules lack coherence and appear to be designed in isolation rather than as a cohesive whole. To address these issues, this paper introduces the Shuffle-GhostNet, which enhances GhostNet through a more meticulous design approach, fully exploiting the benefits of group convolution to reinforce the exchange and circulation of information within each group.

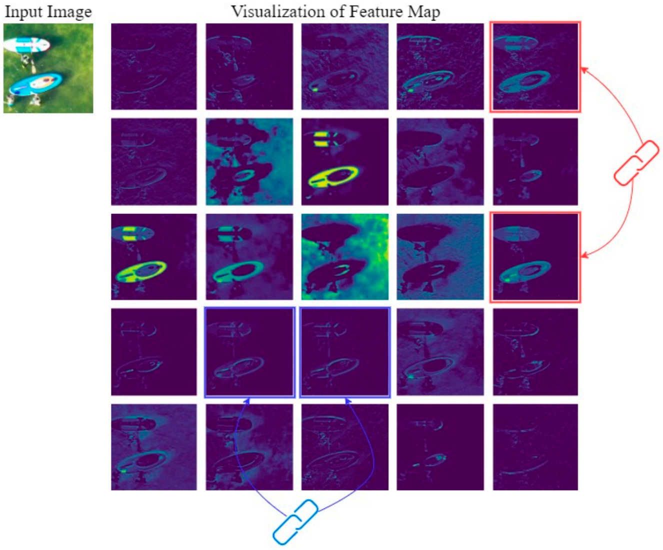

Figure 4. Feature map visualization.

The original GhostNet Module utilizes the 1 × 1 convolution as the primary convolution to adjust the channel dimension, effectively reducing the number of model parameters compared to the 3 × 3 or 5 × 5 convolution. However, the design of the primary convolution and the cheap convolution needed to be better correlated. To achieve more efficient feature extraction modules, we propose the Shuffle-Ghost module, depicted in Figure 5. The primary convolution is optimized as a 1 × 1 group convolution, with two variations based on the number of channels in the network. Specifically, the group convolution is implemented in two cases: when the number of groups is 2 or 4. When the number of groups is limited to 2, the information within each group becomes too dense, making it challenging to leverage the advantages of group convolution. In this case, the benefits of the group convolution design may be obscure. However, by increasing the number of feature groups to 4, we can better distribute the information and achieve more efficient and effective convolutions. By designing the Shuffle-Ghost module from the primary convolution to the group convolution, we enhanced the information flow between the primary convolution and the cheap convolution while reducing the number of model parameters.

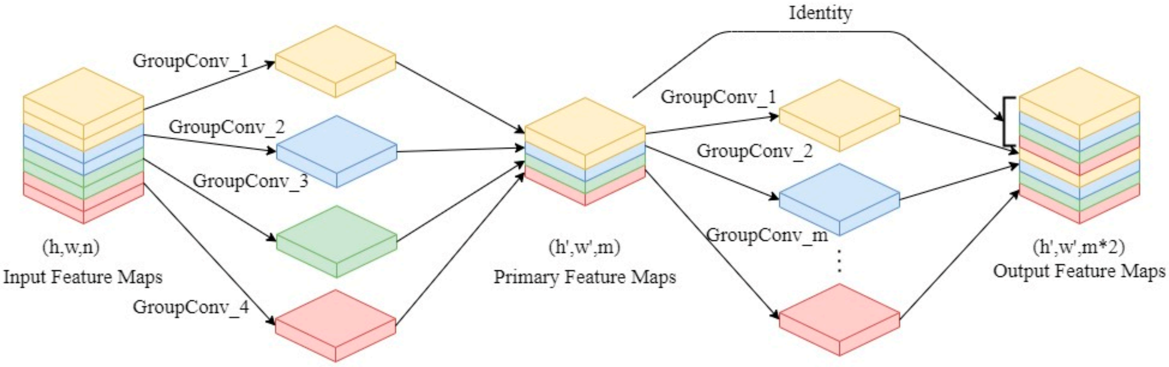

Figure 5. Shuffle-Ghost module.

As depicted in Figure 6, we designed the Shuffle-Ghost bottleneck based on the Shuffle-Ghost module, similar to the basic residual block in ResNet. The bottleneck is primarily composed of two stacked Ghost modules, with the first one serving as an expansion layer that increases the channel dimension, and the second reduces the channel dimension to align with the residual connection. In the original Ghost bottleneck in GhostNet, when the stride is 2, depthwise convolution is added between two Ghost modules for the downsampling function. Depthwise convolution is a standard means of lightweight models used in MobileNet and Xception [33] and requires less computation than the traditional 1 × 1 and 3 × 3 convolutions. However, the depthwise convolution needs to be better adapted to the group convolution used in many parts of GhostNet, which can result in feature loss when downsampling the feature map. To address this issue, we propose the Shuffle-Ghost bottleneck, which replaces depthwise convolution with group convolution and sets the number of groups to 4 to ensure consistency with the primary convolution. Furthermore, a channel shuffle module is incorporated after the group convolution to boost information flow and enhance model representation across different channel groups.

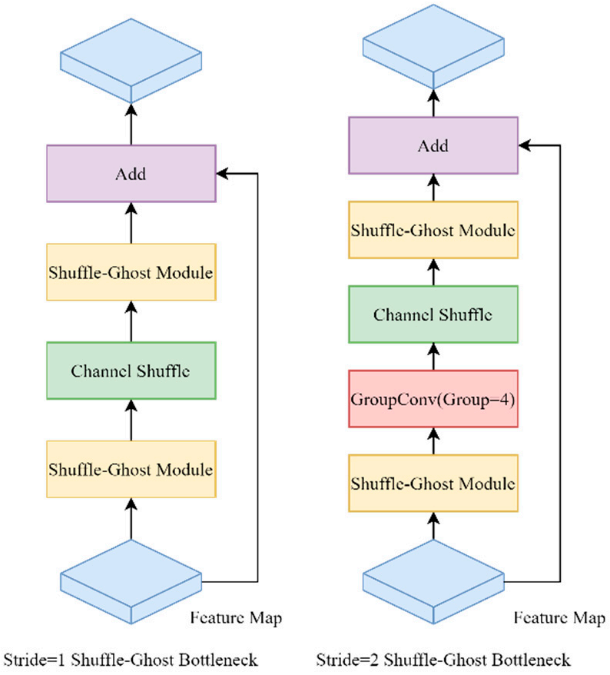

Figure 6. Shuffle-Ghost bottleneck.

When stacking multiple group convolutions, a problem arises where the output of a particular part of the channel is derived from only a part of the input channel. To address this issue, we incorporated the channel shuffle operation into the Shuffle-Ghost bottleneck, fully utilizing each channel's features. Figure 7a shows the situation when the group convolution is stacked. Here, the group convolution with the number of groups is three; as an example, GroupConvN_M, where N indicates the number of group convolution, M indicates which group in the group convolution. For instance, GroupConv1_1 indicates the convolution operation of the first group in the first group convolution. The convolution result of the first output group is exclusively linked to the first input group, and likewise for the other groups, leading to a hindrance in information exchange across different groups. To enhance the flow between channel groups, the channel shuffle operation divides each set of channels into multiple subgroups. Subsequently, it assigns different subgroups to each group in the next level. As depicted in Figure 7b, the first group is partitioned into three subgroups and then allocated to the three groups in the subsequent layer, following the same procedure for the other groups. With this method, the output result of group convolution will then come from the input data of different groups, enabling information flow between different groups.

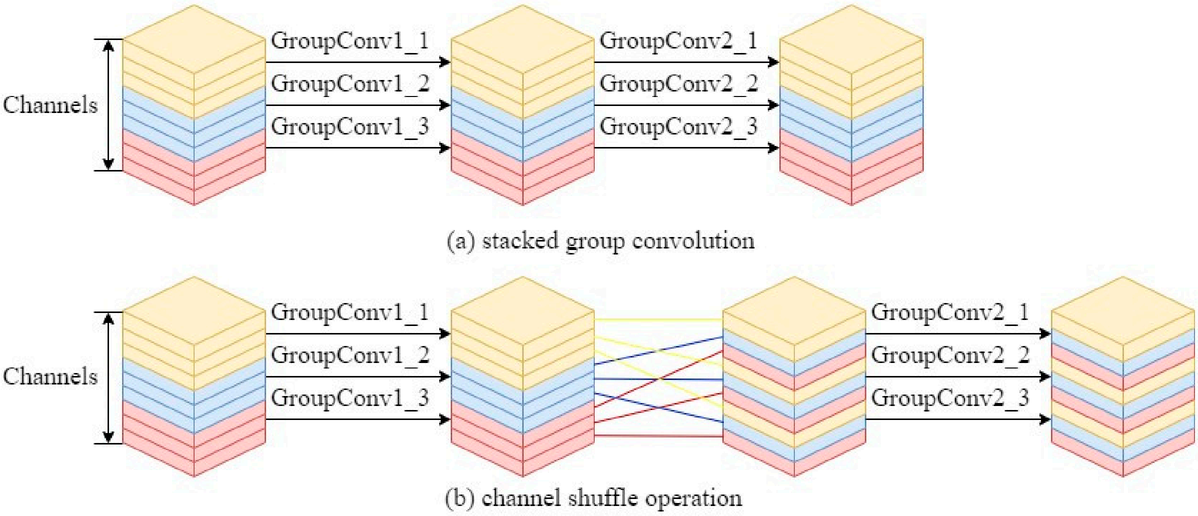

Figure 7. Channel shuffle.

Then, to obtain Shuffle-GhostNet, we stacked the proposed Shuffle-Ghost bottleneck. It is worth noting that, in the original GhostNet, a 1 × 1 convolution was added after the final Ghost bottleneck to increase dimensionality. However, for the lightweight object detector, simply increasing the channels to six times the original number has a significant impact. In the proposed Shuffle-GhostNet, we did not employ generating channels with excessively high dimensions. Instead, we connected the feature pyramid to the last four Shuffle-Ghost bottlenecks to reinforce the features and compensate for the absence of channel information.

3.3. Neck

Functioning as a component linking the backbone and head, the neck plays a pivotal role in processing and merging the features extracted by the backbone to better suit the object detection tasks. In this paper, we propose a lightweight feature pyramid named BiFPN-tiny and integrate it with the ASPP module to serve as the network's neck section. Figure 8 shows the original BiFPN and the proposed BiFPN-tiny. While the original BiFPN employs five feature extraction layers for efficient feature extraction, the design could be somewhat redundant and overlooks the characteristics of UAV images. In our initial BiFPN-tiny design, we aimed to create a lightweight model by reducing the number of feature extraction layers to 3. However, this decision had a downside, as removing the shallow feature extraction layer made it challenging for the model to extract small object features effectively. To enable effective feature extraction at various scales, we have introduced a shallow feature layer solely dedicated to fusing features of small objects. The layer is not involved in the final inference work but is an intermediate step to ensure optimal feature extraction. Furthermore, unlike the original BiFPN, which repeats the BiFPN module several times based on different resource constraints, our model only uses the BiFPN-tiny module once to meet the lightweight and inference time requirements. However, this has resulted in insufficient feature extraction ability, leading to a decline in accuracy.

To enhance the feature extraction capability of the model, we have replaced the 1 × 1 convolution adjustment module before BiFPN-tiny with the ASPP module, as depicted in Figure 9. The ASPP module leverages atrous convolution with different sampling rates for input to obtain multi-scale features, thereby improving the model's ability to extract features. Furthermore, to account for the varying heights and widths of input features at each layer of BiFPN-tiny, we have implemented a larger sampling rate (12, 24, 36) for shallow features with larger heights and widths and a smaller sampling rate (6, 12, 18) for deep features with smaller heights and widths. Using different receptive fields to extract different feature information for different scale feature layers, we can adjust the number of channels while fusing features to improve the model's accuracy. Compared to the original BiFPN, the feature pyramid obtained from the fusion of the proposed BiFPN-tiny and ASPP module exhibits a more robust feature extraction ability and faster reasoning speed.

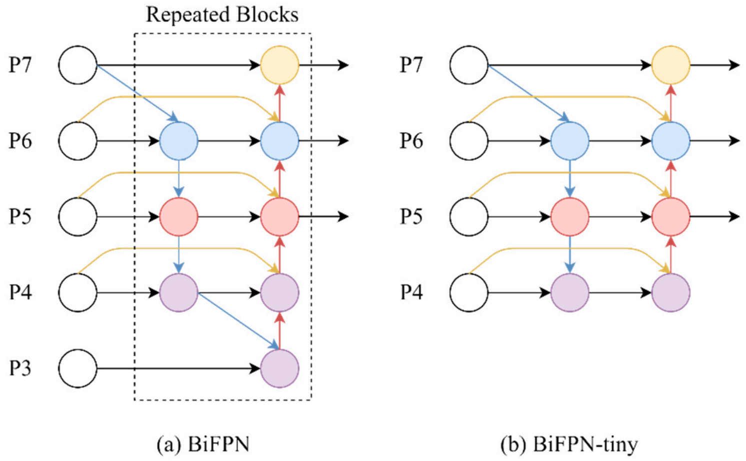

Figure 8. Structure of BiFPN and BiFPN-tiny.

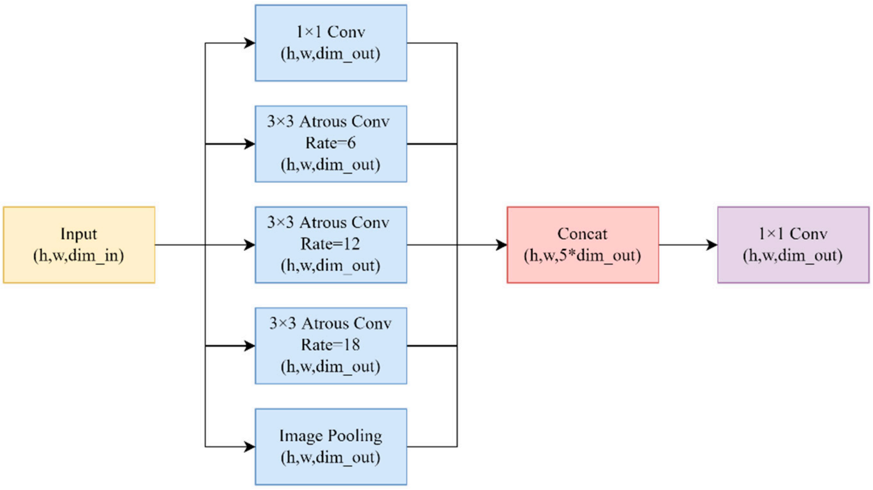

Figure 9. ASPP module.

3.4. Head

The head serves as the final layer of the object detection model, responsible for detecting the object from the feature map. Typically, the head includes a classifier that identifies the object's category and a regressor that predicts the object's location and size information. In the head network section, we utilized three prediction heads to generate accurate predictions for features at various scales. For each prediction head, we generated three anchor boxes of different proportions and sizes to adapt to objects of different sizes to generate the most accurate object bounding box. Therefore, the tensor size of the prediction head in this paper can be expressed as N × M × [3 × (4 + 1 + 6))], where N and M represent the length and width of the tensor, respectively, 3 represents the number of preset anchor boxes, 4 represents bounding box offsets for location prediction, 1 represents object confidence for category prediction, and 6 represents the category of objects in the data set. Then, we decoded the position and size of the prediction bounding box according to the object information in the prediction header and finally carried out non-maximum suppression (NMS) with the true bounding box to filter out the inaccurate prediction results.

4. Experiment and Discussion

4.1. Model Training

The hardware information of this experiment is as follows: the CPU of the computer is AMD R7-5800H, the processor benchmark frequency is 3.2 Ghz, the memory is 16 GB, the graphics card type is NVIDIA RTX 3060, and the video memory is 6 G. The operating system is 64-bit win11, the deep learning framework is Pytorch 1.8.2, and the parallel computing architecture is CUDA11.1. When the model starts training, the training batch size is eight, and the initial learning rate is 0.001.

4.2. Dataset

The dataset used in this experiment is the aerial-drone floating objects (AFO) dataset proposed by [34] precisely for the object detection work of floating objects, and the object category contains six categories: human, surfboard, boat, buoy, sailboat, and kayak. The dataset contains 3647 images and mostly large-size UAV images like 3840 × 2160, with more than 60,000 annotated objects. The original images used in this experiment are too large to use directly for network training. To address this issue, we cropped the original images into multiple 416 × 416 images to facilitate the training process. Additionally, we adopted the cropping method proposed in [35], which involves leaving a 30% overlap when an object is at the edge of the crop. This operation ensures the integrity of object information, as shown in Figure 10. After cropping, we obtained a total of 33,391 416 × 416 images. These images were divided into the training set, validation set, and test set in the ratio of 8:1:1 for the training and testing of the model, respectively.

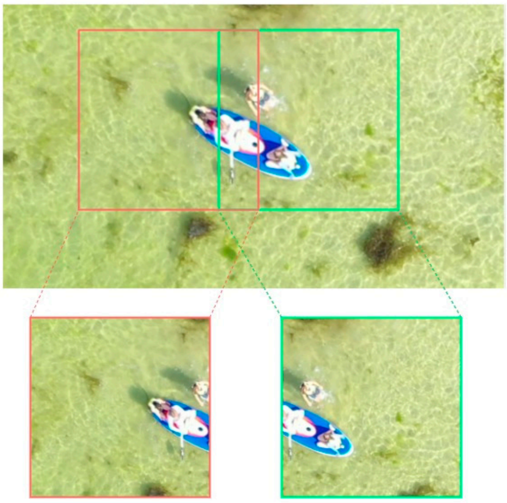

Figure 10. Cropping overlap display.

4.3. Comparison Experiment

To validate the effectiveness of our proposed method, we compared SG-Det with a range of commonly used lightweight object detectors, including SqueezeNet, MobileNetv2 [36], MobileNetv3 [37], ShuffleNetv2 [38], GhostNet, YOLOv3-tiny [39], YOLOv4-tiny [40], YOLOv5-s [41], YOLOv7-tiny [42], and EfficientDet. The evaluation metrics used in this experiment include mean average precision (mAP), frames per second (FPS), giga floating-point operations per second (GFLOPs), and Param, which are used to assess the accuracy, inference speed, computational complexity, and parameter amount of the model, respectively. It is worth noting that SqueezeNet, MobileNetv2, MobileNetv3, ShuffleNetv2, and GhostNet are lightweight classification networks, not end-to-end object detectors. In our experiment, we removed their fully connected layer, following literature recommendations, and replaced the backbone network of Faster R-CNN to achieve object detection. We also conducted the same experiment on our proposed Shuffle-GhostNet to verify its effectiveness. The experimental results are shown in Table 1.

Table 1. Lightweight backbone detection results based on Faster R-CNN.

Model	Detection Framework	mAP	FPS	GFLOPs	Param
SqueezeNet	Faster R-CNN	84.46%	24.99	33.29 G	29.91 M
MobileNetv2		**85.86%**	26.31	61.23 G	82.38 M
MobileNetv3		82.93%	27.03	21.34 G	33.94 M
ShuffleNetv2		74.77%	25.64	52.14 G	62.32 M
GhostNet		83.15%	28.57	58.37 G	60.49 M
Shuffle-GhostNet		84.81%	**30.30**	**21.16 G**	**14.17 M**

Due to the decay of the number of output channels and the use of group convolution in its design, Shuffle-GhostNet has significantly lower computational complexity and parameter count than other lightweight networks. The number of output channels in Shuffle-GhostNet is only 1/6 of that in GhostNet, potentially leading to a decrease in model accuracy. However, our experimental results showed an increase of 1% in mAP. The result suggests that channel shuffling successfully enhances the information exchange between channel groups and that the set number of channels is sufficient to complete the detection task effectively. The high FPS achieved by Shuffle-GhostNet demonstrates that our proposed method meets the timeliness requirements for maritime SAR. Although there is a slight accuracy gap compared to MobileNetv2, Shuffle-GhostNet achieves a balance between multiple performance parameters to meet the practical needs of a production implementation.

To validate the effectiveness of our proposed object detector, we compared it with other end-to-end lightweight object detectors such as YOLOv3-tiny, YOLOv4-tiny, EfficientDet, YOLOv5-s, and YOLOv7-tiny. Additionally, to observe the contribution of each module, we included Shuffle-GhostNet based on Faster R-CNN for comparison. The experimental results are presented in Table 2. Compared to the original BiFPN in EfficientDet, our proposed BiFPN-tiny combined with ASPP appears to be more focused and capable of fully utilizing the potential of multi-scale feature fusion, significantly improving accuracy and speed. Compared to the Faster R-CNN-based Shuffle-GhostNet detector, our approach not only achieves a slight improvement in accuracy and speed but also significantly reduces the number of model parameters and computational effort required. The result highlights the robustness and versatility of our overall framework beyond just the effectiveness of Shuffle-GhostNet. Compared to other lightweight object detectors, our proposed approach has a slightly lower FPS than some models. However, it still provides real-time detection capabilities, making it a suitable option for various applications.

Table 2. End-to-end lightweight object detector detection results.

Model	mAP	FPS	GFLOPs	Param
Shuffle-GhostNet + Faster R-CNN	84.81%	30.30	21.16 G	14.17 M
YOLOv3-tiny	79.23%	29.78	5.71 G	9.09 M
YOLOv4-tiny	81.80%	**34.75**	6.83 G	5.89 M
EfficientDet	73.22%	27.92	4.62 G	3.83 M
YOLOv5-s	84.47%	33.21	6.76 G	7.04 M
YOLOv7-tiny	85.78%	37.54	5.59 G	6.03 M
Our method	**87.48%**	31.90	**2.34 G**	**3.32 M**

To assess the detection performance of our proposed method on targets of varying sizes in UAV images, we listed the AP_s, AP_m, and AP_L scores for each model, which represent the average accuracy of detecting small, medium, and large targets, respectively. Table 3 illustrates that each model exhibits distinct detection capabilities for objects of varying scales. The lightweight detector based on Faster R-CNN employs a deeper and wider network layer, thereby achieving superior detection performance for medium and large-scale objects. However, with increasing network depth, the detector's ability to identify small objects weakens, which poses a challenge to ensuring the accurate detection of such objects. Our proposed method, leveraging the strengths of BiFPN-tiny and ASPP, preserves small-scale features to a great extent, as supported by experimental results that demonstrate its effectiveness in detecting small objects in UAV images.

Table 3. The model's ability to detect objects at all scales.

Model	AP_S	AP_M	AP_L
SqueezeNet	21.1%	41.1%	52.9%
MobileNetV2	19.8%	**47.1%**	**58.6%**
MobileNetV3	12.0%	35.4%	53.7%
ShuffleNetV2	13.6%	31.2%	45.4%
GhostNet	17.8%	38.0%	53.0%
Shuffle-GhostNet	18.3%	40.5%	54.9%
YOLOv3-tiny	15.3%	34.9%	50.9%
YOLOv4-tiny	19.1%	32.2%	41.3%
EfficientDet	13.5%	30.3%	34.9%
YOLOv5-s	27.0%	37.0%	51.1%
YOLOv7-tiny	16.8%	37.4%	52.7%
Our method	**29.8%**	37.2%	52.3%

Then, we conducted a thorough analysis of the experimental results of our proposed method, as presented in Figure 11, which displays the number and detection accuracy of various targets. Notably, the three targets with the lowest detection accuracy are also the least represented in the dataset. Furthermore, due to their high frequency, humans are prone to occlude and overlap other targets in real-world images, which can result in missed detections and misjudgments. Nonetheless, our proposed method achieves a detection accuracy of up to 91% for humans, the primary object of maritime SAR, which satisfies the requirements of practical applications. Our proposed method achieves a better trade-off between performance index parameters, which is more advantageous for real-world marine SAR applications.

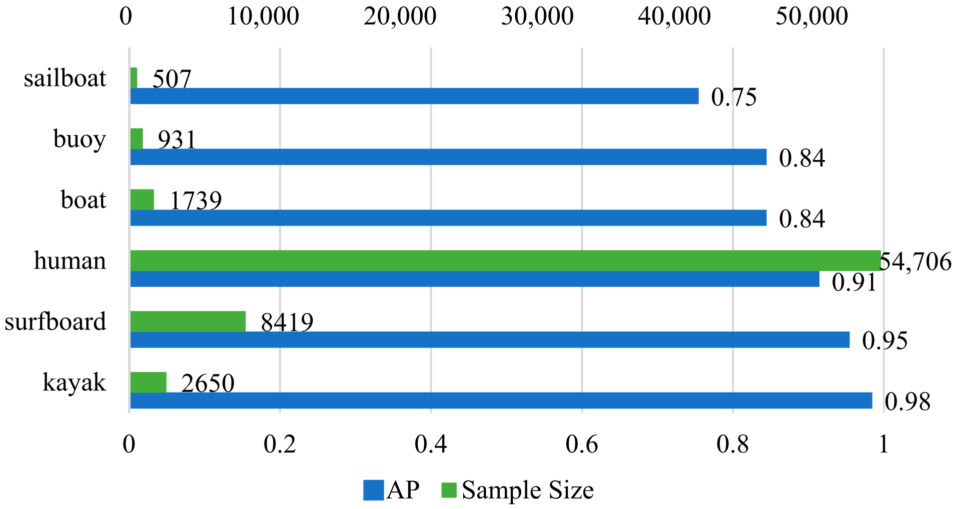

Figure 11. AP and sample size of all categories of objects in the proposed method.

4.4. Ablation Experiment

In order to verify the effectiveness of our proposed method and the contribution of each module, we conducted an ablation experiment. In this section, we added each module to the model step-by-step, ensuring that the experimental environment and configuration remained the same. The results of the ablation experiment are presented in Table 4.

Table 4. Results of ablation experiments.

BiFPN-Tiny	ASPP	RFB	CBAM	GroupConv (Group = 4)	GroupConv (Group = 2)	Channel Shuffle	mAP	FPS	GFLOPs	Param
✓							78.96%	27.48	**1.65 G**	**2.65 M**
✓	✓						84.20%	25.94	3.20 G	4.02 M
✓		✓					83.15%	21.14	1.92 G	2.89 M
✓	✓		✓				77.98%	25.97	3.21 G	4.03 M
✓	✓			✓			85.67%	25.54	2.68 G	3.35 M
✓	✓				✓		85.02%	25.43	2.73 G	3.42 M
✓	✓			✓		✓	**87.48%**	**31.90**	2.34 G	3.32 M

Our experimental results indicate that a single BiFPN-tiny has insufficient feature extraction capability, resulting in lower model accuracy. To address this limitation, we experimented with incorporating additional modules, such as ASPP and RFB [43], to enhance the feature extraction ability of the network. Among these, we found that the ASPP module, which combines the advantages of atrous convolution with different sampling rates, was more effective at achieving multi-scale feature fusion and extraction. We also attempted to improve the network's feature extraction capabilities by incorporating attention mechanisms, such as CBAM [44], to highlight the essential parts of the data. However, our experimental results showed that the feature extraction capability of the network was already close to saturation, and adding the CBAM module just complicated the network structure without improving its performance.

Then, we added group convolution for group 2 and group 4, respectively. The experimental results showed that when the number of groups is 2, there is too much information in a single feature group, so the advantage of group convolution is not apparent. On the other hand, when the number of groups is 4, the network can effectively utilize the features from different channels, resulting in better performance. Therefore, we concluded that the model design with four groups is more reasonable. Lastly, we incorporated channel shuffling operations into the network architecture to optimize the exchange of information between different groups of channels, resulting in a notable enhancement of the model's overall performance. To date, we have validated the effectiveness of all proposed methods and models, ensuring the balance between accuracy and speed.

5. Conclusions

In this paper, we proposed a lightweight detector named SG-Det to tackle the challenge of detecting objects in UAV images for maritime SAR. First, we developed a novel lightweight classification network called Shuffle-GhostNet which serves as the backbone of our detector. By redesigning the correlation group convolution and incorporating channel shuffle operation, Shuffle-GhostNet can significantly reduce the number of parameters and enhance information flow among different groups. Then, we introduced a lightweight feature pyramid called BiFPN-tiny and combined it with the ASPP module to create a four-layer feature pyramid. This architecture uses three layers for feature extraction and one layer to enhance small object features, resulting in an effective and efficient detection framework. Finally, we generated three sets of bounding boxes at different scales—large, medium, and small—to detect objects of different sizes. Extensive experimental results demonstrate that our proposed SG-Det achieves real-time object detection and surpasses a 90% accuracy rate for the primary task of SAR at sea. Moreover, our ablation experiments validate the effectiveness and contribution of each module, providing further evidence of the robustness and reliability of our approach. Compared to other lightweight object detectors, our proposed detector achieves superior trade-offs between performance index parameters, particularly in specific small object tasks. This feature makes it more capable of meeting the actual working requirements of SAR at sea, highlighting the superiority of our model.

Author Contributions: Methodology, L.Z.; Software, G.W.; Validation, Y.X.; Formal analysis, N.Z.; Data curation, G.W.; Writing—original draft, N.Z.; Writing—review & editing, N.Z.; Visualization, Z.C.; Supervision, R.S., Y.X. and Z.C.; Funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Guangdong Water Technology Innovation Project (grant number 2021-07), the Natural Science Foundation of Jiangsu Province (No. BK20201311), and National Natural Science Foundation of China (No. 62073120, 42075191, 91847301, 92047203, 52009080).

Data Availability Statement: The data used to support the findings of this study are obtained from [34].

Acknowledgments: The authors express their gratitude to the anonymous reviewers and the editor for their invaluable comments and insightful suggestions that have greatly contributed to enhancing the quality of the manuscript.

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

References

1. EMSA. *Annual Overview of Marine Casualties and Incidents*; EMSA: Lisboa, Portugal, 2018.
2. Lin, L.; Goodrich, M.A. UAV intelligent path planning for wilderness search and rescue. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 709–714.
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
4. Chen, J.; Li, K.; Deng, Q.; Li, K.; Yu, P.S. *Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing*; IEEE Transactions on Industrial Informatics: Piscataway, NJ, USA, 2019.
5. Xu, Y.; Wang, H.; Liu, X.; He, H.R.; Gu, Q.; Sun, W. Learning to See the Hidden Part of the Vehicle in the Autopilot Scene. *Electronics* **2019**, *8*, 331. [CrossRef]
6. Goswami, G.; Ratha, N.; Agarwal, A.; Singh, R.; Vatsa, M. Unravelling robustness of deep learning based face recognition against adversarial attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 1–3 February 2018; Volume 32.
7. Guo, S.; Liu, L.; Zhang, C.; Xu, X. Unmanned aerial vehicle-based fire detection system: A review. *Fire Saf. J.* **2020**, *113*, 103117.
8. Zhang, J.; Liu, S.; Chen, Y.; Huang, W. Application of UAV and computer vision in precision agriculture. *Comput. Electron. Agric.* **2020**, *178*, 105782.
9. Ke, Y.; Im, J.; Son, Y.; Chun, J. Applications of unmanned aerial vehicle-based remote sensing for environmental monitoring. *J. Environ. Manag.* **2020**, *255*, 109878.
10. 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 June 2014; pp. 580–587.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
12. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. 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.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]
14. 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.
15. 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 European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
16. Wang, Y.; Liu, W.; Liu, J.; Sun, C. Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning-based control. *ISA Trans.* **2023**, *137*, 222–235. [CrossRef] [PubMed]
17. Zhao, H.; Zhang, H.; Zhao, Y. Yolov7-sea: Object detection of maritime uav images based on improved yolov7. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 233–238.
18. Tran, T.L.C.; Huang, Z.-C.; Tseng, K.-H.; Chou, P.-H. Detection of Bottle Marine Debris Using Unmanned Aerial Vehicles and Machine Learning Techniques. *Drones* **2022**, *6*, 401. [CrossRef]
19. Lu, Y.; Guo, J.; Guo, S.; Fu, Q.; Xu, J. Study on Marine Fishery Law Enforcement Inspection System based on Improved YOLO V5 with UAV. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, China, 7–10 August 2022; pp. 253–258.
20. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
21. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]

22. 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, 13–19 June 2020; pp. 10781–10790.
23. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
24. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
25. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
26. Chen, Y.; Dai, X.; Chen, D.; Liu, M.; Dong, X.; Yuan, L.; Liu, Z. Mobile-former: Bridging mobilenet and transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5270–5279.
27. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
28. 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.
29. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
30. 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, 16–17 June 2019; pp. 7036–7045.
31. Luo, Y.; Cao, X.; Zhang, J.; Guo, J.; Shen, H.; Wang, T.; Feng, Q. CE-FPN: Enhancing channel information for object detection. *Multimed. Tools Appl.* **2022**, *81*, 30685–30704. [CrossRef]
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
33. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 1251–1258.
34. Gąsienica-Józkowy, J.; Knapik, M.; Cyganek, B. An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance. *Integr. Comput.-Aided Eng.* **2021**, *28*, 221–235. [CrossRef]
35. Van Etten, A. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv* **2018**, arXiv:1805.09512.
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
37. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
38. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 116–131.
39. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
40. 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, Nashville, TN, USA, 20–25 June 2021; pp. 13029–13038.
41. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Fang, J.; Michael, K.; Montes, D.; Nadar, J.; Skalski, P.; et al. ultralytics/yolov5: V6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. 2022. Available online: https://zenodo.org/record/6222936#.Y5GBLH1BxPZ (accessed on 20 December 2022).
42. 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.
43. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 385–400.
44. 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; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.

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.