Article

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], Zhe Chen [1]

[1] College of Computer and Information Engineering, Hohai University, Nanjing 211100, China;
[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 currently mainly relies on Unmanned Aerial Vehicles (UAVs) to detect the objects. Most traditional object detection methods focus on boosting the detection accuracy while neglecting the detection speed of the heavy model. However, it is also essential to improve the detection speed 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 while being capable of reinforcing small object features. Furthermore, we incorporate the atrous spatial pyramid pooling module (ASPP) to 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, which shows that it can be better meet the actual requirements of maritime search and rescue.

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

1. Introduction

In recent years, maritime accidents that have occurred globally impose a huge toll on human society. Since 2014, the incidence of maritime accidents has gradually increased, with an estimate of approximately more than 4000 fatalities per year [1]. Maritime search and rescue, which is an important part of the national emergency response system, faces the main challenge of how to locate and find 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 potent 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 use in fields such as disaster search and rescue [7], agricultural monitoring [8], and land surveying [9]. Deep learning-based object detection is not only a crucial task for computer vision but also 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 common 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

 doi:10.20944/preprints202305.1132.v1

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 that is based on the Spatial Pyramid Pooling module for feature mapping, and utilizes a multi-task loss function to simultaneously train for classification and localization tasks. Faster R-CNN proposes a Region Proposal Network (RPN) based on Fast R-CNN for generating region proposals, instead of relying on the selective search algorithm. This allows for true end-to-end training, as the RPN is integrated into the entire network architecture. The common one-stage object detectors include YOLO [14], and SSD [15]. YOLO utilizes the whole image as the input to 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 usually perform well, they often struggle to be effective when used to detect UAV images. General object detectors may not work well for UAV images due to differences in viewpoint, object size, background interference, and lighting conditions. In recent years, researchers have started to address this issue by proposing target detection algorithms and models specifically designed for UAV images. Tan [16] et al. proposed an improved version of the YOLOv4 object detection model called YOLOv4_Drone, which uses hollow convolution, ultra-lightweight subspace attention mechanism, and soft non-maximum suppression for feature extraction, multi-scale feature representation, and object detection, respectively. Yang [17] et al. proposed a new approach for detecting objects in aerial images using clustering techniques. The proposed approach uses a hierarchical clustering algorithm to group objects that are close to each other in space and have similar properties, such as color and texture. The resulting clusters are then processed by a detector to identify individual objects within the clusters. Xu [18] et al. proposed a new method for detecting small objects in aerial images, which utilizes the dot distance algorithm to effectively identify isolated object in clutter backgrounds.

Due to the hardware limitations and application scenarios of UAVs, lightweight and real-time performance are essential requirement for UAV image object detectors. Generally speaking, pursuing speed excessively can lead to a loss of accuracy, and vice versa. As for how to improve the efficiency of the 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) Improve the detection accuracy of small objects. Building on the aforementioned discussion, this paper introduces a novel one-stage lightweight object detector, called SG-Det, which is specifically designed for detecting 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 [19] 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 [20] module to form the neck of the detector. To effectively 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) [21] 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:

 doi:10.20944/preprints202305.1132.v1

(1) We propose a lightweight object detector name SG-Det, which simultaneously meets the requirements of high precision and high-speed detection of UAV images in SAR.
(2) We design a lightweight classification network name 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 dense small 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 the large number of parameters. To address the issue of large-scale neural network models, many scholars have proposed their own lightweight network architectures. The lightweight neural network is a specialized neural network structure designed for efficient computation and low-delay reasoning, making it particularly suitable for scenarios with limited computing resources, such as mobile devices. MobileNet [22] introduces the concept of depthwise separable convolution, which decomposes standard convolution into depthwise convolution and pointwise convolution. This decomposition allows MobileNet to significantly reduce computation and parameters while maintaining high accuracy. SqueezeNet [23] utilizes two distinct types of convolutional layers: squeeze layers and expand layers. The squeeze layers are employed to decrease the number of channels, while the expand layers increase the number of channels and also augment the depth of the feature map. ShuffleNet [24] utilizes a combination of group convolution and channel shuffling operations to achieve a high level of accuracy while simultaneously maintaining low computational costs. 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. To address this issue, the feature pyramid network was proposed, 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) [25] introduces a top-down approach for integrating high-level features with low-level features, enabling the combination of low-resolution feature maps with rich semantic information and high-resolution feature maps with rich spatial information. The Path Aggregation Network (PANet) [26] builds upon the foundation of FPN and further 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) [27] 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. In this paper, we propose a lightweight

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

BiFPN designed specifically for UAV image detection, with a focus on enhancing the features of small objects.

2.3. Group Convolution

Group convolution was initially used in AlexNet[28] 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 g groups based on the number of channels, followed by a conventional convolution on each group. Suppose the input feature map shape is (h,w,n) and the output feature map shape is (h',w',m), then the computation of conventional convolution is

$$N = h' \times w' \times n \times m \times k \times k \tag{5}$$

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

After dividing the input feature maps into g groups, the number of channels in each group of input feature maps is n/g and the number of channels in each output feature map group is m/g. Then the computation of the group convolution is

$$N' = g \times h' \times w' \times \frac{n}{g} \times \frac{m}{g} \times k \times k = \frac{1}{g} \times h' \times w' \times n \times m \times k \times k \tag{6}$$

The group convolution reduces the computation of the conventional convolution to 1/g, and also reduces the number of parameters to 1/g. However, it's important 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

Ghost convolution was introduced by GhostNet as a cost-effective linear operation to generate feature maps, which effectively reduces both 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 compared to conventional convolution, greatly reducing the model's complexity. The primary feature maps and Ghost feature maps are then combined to obtain output feature maps that are sufficient for feature extraction. In this approach, both the primary feature maps and Ghost feature maps are kept at the same size. In order to prevent an excessive number of parameters from being generated in our object detector, we employ ghost convolution multiple times throughout the network.

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 three parts: the backbone, neck, and head. In the backbone network secetion, we introduce a novel lightweight classification network called Shuffle-GhostNet, which consists of multiple Shuffle-Ghost bottlenecks. The structure of the Shuffle-Ghost Bottleneck bears some resemblance to 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 secetion, we construct a four-layer feature pyramid

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

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 excel at capturing intricate details such as local features and textures, whereas high-level features focus on extracting global semantic information and abstract features. To harness the strengths of both, the neck network integrates low-level and high-level features, resulting in improved performance. In the head network secetion, we generate three sets of boundary boxes at different scales - large, medium, and small - to enable detection of objects of varying sizes.

3.1. 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 amout 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 ShuffleGhostNet 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 serveral locations to significantly decrease the computataional complexity of the model compared to traditional object detection methods. Howerver, some modules lack coherence and appear to be designed in isolation rather than as a cohesive while. 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.

The original GhostNet Module utilizes the 1*1 convolution as the primary convolution to adjust the channel dimension, which effectively reduces the number of model parameters compared to 3*3 or 5*5 convolution. However, the design of the primary convolution and the cheap convolution were not well-correlated. To achieve more efficient feature extraction modules, we propose the Shuffle-Ghost Module, which is 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 in 2 or 4. When the number of group is limited to 2, the information contained within each group becomes too dense, making it difficult to fully leverage the advantages of group convolution. In this case, the benefits of the group convolution design may not be apparent. 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 enhance the information flow between the primary convolution and the cheap convolution, while simultaneously reducing the number of model parameters.

As depicted in Figure 6, we design the Shuffle-Ghost Bottleneck based on the Shuffle-Ghost Module, which is 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 one reducing the channle dimension to align with the residual connection. The original Ghost Bottleneck in GhostNet, when the stride is 2, depthwise convolution is added between two Ghost Modules for downsampling function. Depthwise convolution is a common means of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

lightweight models, is used in MobileNet and Xception [29], and requires less computation than the traditional 1*1 and 3*3 convolutions. However, the use of the depthwise convolution is not well 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 channle shuffle module is incorporated after the group convolution to boost information flow and enhance model representation across different channel groups.

When stacking multiple group convolutions, a problem arises where the the output of a certain part of the channel is derived from only a part of the input channel. To address this issue, we incorporate the channel shuffle operation into the Shuffle-Ghost Bottleneck, enabling the full utilization of each channel's features.Figure 7(a) shows the situation when the group convolution is stacked, here the group convolution with the number of groups is 3 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. Clearly, 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, and subsequently assigns different subgroups to each group in the next level. As depicetd in Figure 7(b), the first group is partitioned into three subgroups, which are 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.

Then to obtain Shuffle-GhostNet, we stack the proposed Shuffle-Ghost Bottleneck. It's worth noting that, in the original GhostNet, a 1x1 convolution was add after the final Ghost Bottleneck to increase dimensionality. However, for the lightweight object detector, simply increasing the number of channels to six times the original number has a significant impact. In the proposed Shuffle-GhostNet, we don't employ the operation of generating channels with excessively high dimensions. Instead, we connect 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 name 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 emplorys five feature extraction layers for efficient feature extraction, the design is somewhat redundant and overlooks the characteristics of UAV images. In our initial BiFPN-tiny design, we aimed to created a lightweight model by reducing the number of feature extraction layers to 3. However, this decision had a downside as the removal of the shallow feature extraction layer made it challenging for the model to effectively extract small object features. 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 rather serves as 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

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. By 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 accuracy of the model. When 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.

3.3. 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 generate three sets of bounding boxes at different scales – large, medium and small – to detect objects of different sizes. Therefore, the predicted tensor size is $N \times M \times [3 \times (4 + 1 + 6)]$ for four bounding box offsets, one object confidence, and six class probabilities, where N and M represent the tensor's height and width, respectively. After predicting the bounding boxes, the final detection results are obtained by filtering the predicted boxes using non-maximum suppression (NMS).

4. Experiment

4.1. Model training

The hardware information of this experiment is: the CPU of the computer is AMD R7-5800H, the processor benchmark frequency is 3.2Ghz, the memory is 16GB, the graphics card type is NVIDIA RTX 3060, and the video memory is 6G. 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 batchsize is 8 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 [30] specifically 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 be directly used for network training. To address this issue, we cropperd the original images into multiple 416*416 images to facilitate the training process. Additionally, we adopted the cropping method proposed in [31], which involves leaving a 30% overlap when there is an object at the edge of the crop. This ensures the integrity of object information, as shown in Figure 10. After cropping, we obtained a total of 33,391

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

416*416 images. These images are 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.

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 [32], MobileNetv3 [33], ShuffleNetv2 [34], GhostNet, YOLOv3-tiny [35], YOLOv4-tiny [36], 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's 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	33.29G	29.91M
MobileNetv2		**85.86%**	26.31	61.23G	82.38M
MobileNetv3		82.93%	27.03	21.34G	33.94M
ShuffleNetv2		74.77%	25.64	52.14G	62.32M
GhostNet		83.15%	28.57	58.37G	60.49M
Shuffle-GhostNet		84.81%	**30.30**	**21.16G**	**14.17M**

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 compared to other lightweight networks. The number of output channels in Shuffle-GhostNet is only 1/6 of that in GhostNet, which could potentially lead to a decrease in model accuracy, as hypothesized. However, our experimental results showed an increase of 1% in mAP. This suggests that channel shuffling successfully enhances the information exchange between channel groups, and that the set number of channels is sufficient to effectively complete the detection task. 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 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, and EfficientDet. 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, resulting in a significant improvement in both accuracy and speed. In comparison 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. This highlights the robustness and versatility of our overall framework, beyond just the effectiveness of Shuffle-GhostNet. In comparison to other lightweight object detectors, our proposed approach has a slightly lower FPS than YOLOv4-tiny. However, it still provides real-time detection capabilities, making it a suitable option for various applications.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

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

Model	mAP	FPS	GFLOPs	Param
Shuffle-GhostNet + Faster R-CNN	84.81%	30.30	21.16G	14.17M
YOLOv3-tiny	79.23%	29.78	5.71G	9.09M
YOLOv4-tiny	81.80%	**34.75**	6.83G	5.89M
EfficientDet	73.22%	27.92	4.62G	3.83M
Our method	**87.48%**	31.90	**2.34G**	**3.32M**

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 respectively represent the average accuracy of detecting small, medium, and large targets. 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 accurate detection of such objects. 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 accurate detection of such objects. From the results shown in Table 3, it can be found that different models have different detection capabilities for objects of different scales. The Faster R-CNN-based lightweight detector has a deeper and wider network layer, which has a better detection effect for medium and large-scale objects, but as the network layer deepens, the features of small objects become weaker and weaker, and it is difficult to guarantee the detection accuracy of small 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%
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 13, 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. In summary, our proposed method achieves a better trade-off between performance index parameters, which is more advantageous for real-world maritime SAR applications.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 May 2023 doi:10.20944/preprints202305.1132.v1

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 while ensuring that the experimental environment and configuration remained the same. The results of the ablation experiment are presented in Table 4.

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 [37], 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 [38], to highlight the most important 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 only complicated the network structure without improving its performance.

And then we added group convolution of 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 obvious. 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 design of the model with 4 groups is more reasonable. Lastly, we incorporate 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.

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	**1.65G**	**2.65M**
✓	✓						84.20%	25.94	3.20G	4.02M
✓		✓					83.15%	21.14	1.92G	2.89M
✓	✓		✓				77.98%	25.97	3.21G	4.03M
✓	✓			✓			85.67%	25.54	2.68G	3.35M
✓	✓				✓		85.02%	25.43	2.73G	3.42M
✓	✓			✓		✓	**87.48%**	**31.90**	2.34G	3.32M

4. Conclusion

In this paper, we propose a lightweight detector name SG-Det to tackle the challenge of detecting objects in UAV images for maritime SAR. First, we develope 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 introduce 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 generate 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. In comparison 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:

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: Not applicable.

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

References

1. EMSA. Annual overview of marine casualties and incidents[J]. 2018.
2. Lin L, Goodrich M A. UAV intelligent path planning for wilderness search and rescue[C]//2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009: 709-714.
3. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
4. Chen J, Li K, Deng Q, et al. Distributed deep learning model for intelligent video surveillance systems with edge computing[J]. IEEE Transactions on Industrial Informatics, 2019.
5. Xu Y, Wang H, Liu X, et al. Learning to see the hidden part of the vehicle in the autopilot scene[J]. Electronics, 2019, 8(3): 331.
6. Goswami G, Ratha N, Agarwal A, et al. Unravelling robustness of deep learning based face recognition against adversarial attacks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).
7. Guo, S., Liu, L., Zhang, C., & Xu, X. (2020). Unmanned aerial vehicle-based fire detection system: A review. Fire Safety Journal, 113, 103117.
8. Zhang, J., Liu, S., Chen, Y., & Huang, W. (2020). Application of UAV and computer vision in precision agriculture. Computers and Electronics in Agriculture, 178, 105782.
9. Ke, Y., Im, J., Son, Y., & Chun, J. (2020). Applications of unmanned aerial vehicle-based remote sensing for environmental monitoring. Journal of Environmental Management, 255, 109878.
10. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
11. He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.
12. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
13. Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
14. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
15. Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.
16. Tan L, Lv X, Lian X, et al. YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm[J]. Computers & Electrical Engineering, 2021, 93: 107261.
17. Yang F, Fan H, Chu P, et al. Clustered object detection in aerial images[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 8311-8320.
18. Xu C, Wang J, Yang W, et al. Dot distance for tiny object detection in aerial images[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 1192-1201.
19. Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.
20. Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834-848.
21. Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.
22. Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

 doi:10.20944/preprints202305.1132.v1

23. Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size[J]. arXiv preprint arXiv:1602.07360, 2016.
24. Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848-6856.
25. Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
26. Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.
27. Ghiasi G, Lin T Y, Le Q V. Nas-fpn: Learning scalable feature pyramid architecture for object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 7036-7045.
28. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
29. Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.
30. Gąsienica-Józkowy J, Knapik M, Cyganek B. An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance[J]. Integrated Computer-Aided Engineering, 2021, 28(3): 221-235.
31. Van Etten A. You only look twice: Rapid multi-scale object detection in satellite imagery[J]. arXiv preprint arXiv:1805.09512, 2018.
32. Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.
33. Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314-1324.
34. Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116-131.
35. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
36. Wang C Y, Bochkovskiy A, Liao H Y M. Scaled-yolov4: Scaling cross stage partial network[C]//Proceedings of the IEEE/cvf conference on computer vision and pattern recognition. 2021: 13029-13038.
37. Liu S, Huang D. Receptive field block net for accurate and fast object detection[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 385-400.
38. Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.