Light-SAR-ShipNet (LSSNet): A Lightweight SAR Image Ship Detection System with Enhanced Deep Supervision

Lianzhong Zhang*1, Aihua Cai2, Wenna Li3, Panzhou Hao4
1University of Electronic Science and Technology of China (UESTC) Chengdu, China
2University of Electronic Science and Technology of China (UESTC) Chengdu, China
3University of Electronic Science and Technology of China (UESTC) Chengdu, China
4China Electronic Science Research Institute Beijing, China
*zlz0615@qq.com

Abstract—Applying deep learning methods to achieve synthetic aperture radar (SAR) image ship target detection has become a current hot research topic. However, most of the existing detection networks are designed based on optical images, so when they are directly applied to SAR image target detection, there will be the following two problems: 1) There are big differences between SAR image and optical image in imaging mechanism, geometric characteristics and radiation characteristics, etc, which makes the detection network have more redundant information; 2) Real-time performance and accuracy are both very important in SAR image target detection task, and applying a network designed based on optical image cannot balance the performance of the two well. In response to the above problems, this paper proposes a lightweight network LSSNet specially designed for SAR image ship detection task. In earlier layers, we use depthwise separable convolution instead of conventional convolution to design dense block; in deeper layers, stacked dense blocks with shortcuts are used to enhance deep supervision, and finally achieve high-speed and high-accuracy detection. This paper uses SSDD dataset as baseline for experiment and results show that LSSNet have higher accuracy and detection speed. Its lightweight structure will also help transplant to hardware devices in the future.

1. INTRODUCTION

SAR image target detection is divided into two categories: traditional methods and deep learning methods. The traditional ship target detection algorithm is mainly composed of three parts: detection window design, feature selection, and classifier design. Among them, the constant false alarm rate detection(CFAR) algorithm is one of the most widely used target detection algorithms. It detects the target by modeling the statistical distribution of the background clutter. However, with the increasing maturity of SAR image acquisition technology, traditional SAR target detection methods with complex calculations and high manual participation are no longer enough to meet people's needs for SAR image processing.

In recent years, SAR image ship detection mainly uses deep learning methods to achieve detection, but many deep learning network models have complex structures and many parameters, resulting in poor real-time performance and not conducive to the transplantation of hardware devices. So there has been a rising interest in running high-quality CNN models under strict constraints on memory and computational budget. Therefore, some experts and scholars have carried out research on the...
lightweight detection network[1-7], and have achieved certain results. Traditional lightweight CNN network structure includes ShuffleNet[1], GhostNet[2], MobileNet[3], Xception[4], PeleeNet[5], DenseNet[6], SqueezeNet[7], etc.

However, the above lightweight network designs are all based on optical images, which may not be applicable when migrating to the SAR image domain[8]. The experimental results from this paper will also prove this point. At present, the lightweight network applied to ship detection in SAR images[11,12,13,14] can be summarized as the following problems: first, there are more redundancy in the network model. This is because optical images are three-channel while SAR images are single-channel, and there are big differences between the two in imaging mechanism, geometric characteristics, radiation characteristics, etc[9,10]. The lightweight network designed for optical images is not well qualified for the detection task of SAR images; second, the balance of accuracy and speed is not ideal[15,16] since that high-speed detection network is often accompanied by the cost of sacrificing accuracy. However, real-time performance and accuracy are both important in SAR image detection tasks[17,18]. Therefore, the detection network needs to improve the utilization of feature maps.

In order to solve the above problems, this paper proposes the LSSNet for the SAR image ship target detection task, which realizes high-speed and high-precision ship target detection in SAR images. This lightweight model is also easy to transplant to hardware devices with limited computing resources to achieve real-time detection. The innovations of this paper are as follows:

First, compared with the lightweight networks suitable for optical images, a new lightweight network LSSNet specially designed for SAR image ship detection task is proposed. The model size of LSSNet is only 11.7M, and the mAP reaches 98.6%. Besides, it has a significantly lower false alarm rate and missed alarm rate. Second, the convolutional layer in LSSNet adopts a dense connection method to design, so that it can have a narrower layer to decrease the amount of parameters. And it adopts a stacked dense blocks structure with shortcut in the deeper network to enhance deep supervision, ensuring the efficient use of feature maps and improve the detection accuracy.

2. RELATED WORK

2.1. Depthwise Convolution
In order to achieve the purpose of designing a lightweight detection network, we are inspired by the depthwise convolution and using this method can greatly decrease the amount of network parameters. At present, the research of lightweight detection network has made progress, such as MobileNet series[1,20], ShuffleNet series[3,19], etc., which all use depthwise convolution to design the network and achieve a lightweight one. Similarly, Xception[4] also draws on the idea of depthwise convolution to decrease network parameters. In the SAR image ship detection task, depthwise convolution can be introduced to redesign the feature extraction network to achieve the purpose of lightening the network.

Figure 1 shows the structure of depthwise convolution. Unlike conventional convolution, each convolution kernel is convolved with only one channel, which greatly reduce the amount of network parameters. However, there are two problems with using depthwise convolution alone. One is that the number of feature channels cannot be expanded, and the other is that the correlation between channels is ignored. These reasons will lead to the loss of detection accuracy. Research on optical images[21,22] has proved that introducing a 1×1 convolution kernel after depthwise convolution can achieve almost the same accuracy as conventional convolution, but the amount of parameters is greatly decreased.
2.2. Dense Connection

In order to achieve high-precision detection, we are inspired by a dense connection, experiments in this paper can also verify the effectiveness of the method. In the deep learning network, as the depth of the network continues to deepen, gradient disappearance becomes more and more obvious\cite{23}. In order to alleviate the problem, scholars try to connect the shallow features with the latter features through shortcuts. The densely connected DenseBlock is introduced for the first time in DenseNet\cite{6}. This structure has very excellent performance and PeleeNet\cite{5}, DSOD\cite{8}, etc\cite{24,25,26}, have been successively proposed based on this structure. Figure 2 shows the densely connected structure. It concatenates the feature maps obtained in each layer to realize the efficient use of features and alleviate the gradient disappearance problem. Since the feature map can be utilized to the greatest extent, the channel of each layer can be narrow enough to reduce the amount of network parameters. In the SAR image ship detection task, efficient using feature maps can improve the detection performance and decrease the redundant parameters to a certain extent, which is beneficial to the design of lightweight networks.
3. Overview of Proposed Method

Based on the existing research results of lightweight network and drawing on its lightweight module design method, we design LSSNet for the SAR image ship targets detection task, which achieves high-precision and high-speed detection. Figure 3 shows the structure of LSSNet. It consists of three sub-modules, DenseBlock, Transition layer and stacked denseblock with shortcut (ShortcutDB). In order to realize the detection of different sizes’ ship targets, the network sets two different detection scales. One is the feature map obtained at the end of LSSNet, and the other is the feature map obtained in the penultimate downsampling.

3.1. Transition Layer

It is designed to down-sample the feature maps and increase the number of feature map channels. Shallow features have a small receptive field, but contains rich image informations, which is conducive to the detection of small targets. As the number of network layers continues to deepen, the receptive field gradually becomes larger and the feature map scale becomes smaller and smaller. In order to prevent more ship features loss, the number of convolutional layers and deep feature channels in the deep position of the network should be set to increase, and the transition layer achieves this.

Figure 4 shows the structure of transition layer. The convolution layer uses a 3×3 convolution kernel with stride 2. Then we provide batch normalization (BN)\[27\] operations after the convolutional layer, and finally use the activation function LeakReLU to activate.

![Figure 4 Structure of transition layer](image)

3.2. DenseBlock

The traditional denseblock module structure[6] is shown in Figure 5(a). For each layer, all the feature maps in previous layers are used as the input of the current layer, and its own feature maps are used as the input of the subsequent layers, forming a full interconnection. The denseblock module proposed in this paper is shown in Figure 5(b). It replaces the conventional 3×3 convolutional layer in the traditional denseblock with 3×3 depthwise convolution and 1×1 pointwise convolution. Related papers[21,22]
point out this convolution method can ensure that detection network parameters is greatly decreased without greatly impairing the accuracy. After that, the SE module is introduced. And finally the feature maps obtained in the previous layer are concated with the feature maps currently obtained at the output.

![Figure 5](image_url) (a) Traditional DenseBlock; (b) Our Lightweight DenseBlock.

This interconnected structure of upper and lower layer feature maps can make full use of the target information in each layer of feature maps. On the basis of retaining the feature maps of the previous layers, each layer can add new feature maps. Even if the number of new feature maps in each layer is small, the feature information can be fully utilized through the form of multi-layer interconnection. Therefore, the DenseBlock designed in this paper decreases the number of new feature maps for each layer, which not only ensures the full use of feature information, but also helps to achieve a lightweight design of the network.

The SE module[28] is proposed for the accuracy loss caused by the different importance of feature map channels in the process of convolution. Conventional convolution is that each feature channel has the same importance by default, but their importance is not the same in actual problems[4,18]. The introduction of the SE module can effectively solve this problem.

![Figure 6](image_url) Squeeze and Excitation module.

The structure of the SE module is shown in Figure 6. It mainly consists of compression and excitation. The compression operation is a global average pooling and the excitation operation is composed of two fully connected layers. The excitation operation can obtain the importance of each channel with a size of \(1 \times 1 \times C\) weight coefficient. Finally, the input feature map is multiplied by the weight coefficient to obtain a feature map considering the importance of different channels.
3.3. Stacked DenseBlocks With Shortcut (ShortcutDB)
In the shallow features, a small number of channels can still contain rich ship target information, and the DenseBlock structure designed above can effectively extract ship features from the shallow structure. However, as the number of layers deepens, the receptive field of the convolutional layer becomes larger and larger, which is prone to loss of ship feature information. Therefore, it is necessary to increase the number of convolutional layers and convolutional channels in the deep structure. Using the above-mentioned DenseBlock structure concating upper and lower layer features will make the width of the feature map wider and wider, which is not conducive to the realization of lightweight networks. The use of a 1×1 convolution kernel can compress the width of the feature map, but will lose a part of the feature information. In order to ensure a lightweight network structure with no damage to the feature information, a stacked dense block structure with shortcut is proposed later.

It named this stacked DenseBlocks with shortcut ShortcutDB. The ShortcutDB structure is shown in Figure 7(a). It uses two DenseBlocks designed above to stack, and then compresses the number of feature channels through a 1×1 convolution kernel, so that the number of feature maps added each time is the same as a single DenseBlock, and finally we directly connect input to the output through a 1×1 convolution kernel. The shortcut branch adopts the 1×1 convolution kernel because we have to ensure the same number of input and output channels, otherwise the input and output feature maps cannot be added directly. It is precisely because of the existence of shortcut that the deep features have an enhanced supervision. We also showed stacked DenseBlocks without shortcut as Figure 7(b) to prove the superiority of ShortcutDB module in later experiments.

4. DETAILS OF DETECTION

4.1. Activation Function
In order to make the network training more fully, batch normalization (BN) operation is performed after each convolutional layer, and the LeakReLU function is used for activation, which helps the gradient disappearance during the training process. The activation function LeakReLU shows as follows:

\[
LeakReLU(x) = \begin{cases} 
  x, & x \geq 0 \\
  0.1x, & x < 0 
\end{cases}
\]

(1)

4.2. Anchor Box
Two-scale feature maps are used for prediction, and sets three anchor boxes for each predicted feature map. For the smallest feature map, it has the largest receptive field and is suitable for detecting larger targets, so the largest anchor boxes are used: (81,82), (135,169), (344,319). For larger feature maps, its receptive field is smaller, which is suitable for detecting small and medium-sized targets. Therefore, corresponding anchor boxes are used: (10,14), (23,27), (37,58).
4.3. Loss Function

The loss function proposed in the detection network can be used for reference in YOLOv3[29]. Since there is only one type target in SAR image ship detection, the category loss function could be ignored.

To achieve SAR ship detection, it is necessary to obtain center coordinates, height and width of the ship target bounding box and its confidence score. Therefore, the overall loss function consists of two parts: the bounding box loss function and the confidence loss function.

The bounding box loss function is as follows:

$$\text{Loss}_{\text{bbox}} = \sum_{i=0}^{B} \sum_{j=0}^{S} I_{i}^{\text{ship}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \\
\lambda_w \sum_{i=0}^{B} \sum_{j=0}^{S} I_{i}^{\text{ship}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$  \hspace{1cm} (2)

Where $x_i, y_i$ represent the i-th ground truth coordinate, $\hat{x}_i, \hat{y}_i$ represent the i-th bounding box coordinate in the j-th grid, $\hat{w}_i, \hat{h}_i$ represent the i-th ground truth width and height in the j-th grid, $w_i, h_i$ represent the i-th bounding box width and height in the j-th grid. $I_{i}^{\text{ship}}$ is a 0-1 variable, if the i-th grid contains ships or part of ships, $I_{i}^{\text{ship}} = 1$, otherwise $I_{i}^{\text{ship}} = 0$. $\lambda_c$ is the coordinate loss weight coefficient, $\lambda_w$ is the width and height loss weight coefficient, $B$ is the number of bounding boxes, $S$ is the number of grids (The specific grid division method is detailed in Yolov3[29]).

The confidence loss function is as follows:

$$\text{Loss}_{\text{conf}} = \sum_{i=0}^{B} \sum_{j=0}^{S} I_{i}^{\text{ship}} \left[ \hat{s}_{i,j} - \text{GIOU}(B_p, B_G) \right]^2 + \\
\lambda_c \sum_{i=0}^{B} \sum_{j=0}^{S} (1 - I_{i}^{\text{ship}}) [\hat{s}_{i,j} - \text{GIOU}(B_p, B_G)]^2$$  \hspace{1cm} (3)

Where $\hat{s}_{i,j}$ represents the the j-th bounding box confidence score in the i-th grid, $\lambda_c$ represents the confidence score loss function weight coefficient, $\lambda_p$ is the punishment coefficient when undetecting ships.

The traditional method uses IoU to construct the loss function. IoU is the intersection ratio of the predicted box and the real box, which is defined as:

$$\text{IoU} = \frac{\text{area}(B_p \cap B_G)}{\text{area}(B_p \cup B_G)}$$  \hspace{1cm} (4)

Where $B_p$ reprents the prediction bounding box, $B_G$ reprents the ground truth.

This paper uses GIOU[30] to construct the loss function, as shown in Figure 8(b), which is defined as:
\[ \text{GIoU} = \text{IoU} - \frac{|C - (B_{\text{pre}} \cup B_{\text{ground}})|}{|C|} \] (5)

Where \( C \) is minimum bounding rectangle of prediction bounding box and ground truth. The loss function constructed with GIoU not only focuses on the overlapping area, but also on the non-overlapping area. Using GIoU solves the problem that the distance cannot be evaluated between two non-overlapping areas.

5. Experiment

This paper is based on the Ubuntu16.04 system for experiments, using Pytorch[31] as the framework of deep learning, and writing programs on the Pycharm software platform. The experimental hardware configuration CPU is Inter(R) i5-10400F, GPU is GeForce GTX 1660 SUPER, memory is 16G, and CUDA 11.1 is used to call GPU for training acceleration.

5.1. Dataset

The dataset used in this paper is SSDD, published by the Professor Li Jianwei et al. of Naval Aviation University[9,32,33]. SSDD comes from the satellite-borne radar system, and the target areas are cropped into about 500×500 pixel images. It has four polarization modes: HH, HV, VV and VH, with a 1-15m resolution, ship targets are distributed in open sea areas and coastal areas. The dataset contains a total of 1160 images, 2456 ship targets, and each image has an average of 2.12 ship targets. The dataset images have multiple polarization modes, different resolutions, far and near sea scenes, which can effectively verify the performance of the detection algorithm, having been used by many scholars[9,18,32,33,34,35]. Unfortunately, the SSDD dataset does not give a clear data division method[18], and the relevant literature[9,18,32,33,34,35] divides the dataset according to its own needs. This paper refers to the division method of related literature and uses SSDD dataset with the same division method for performance comparison.

### TABLE I. COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT METHODS

| Method     | GT | TP | \( F_p \) | \( F_N \) | \( P_d \) | \( P_f \) | \( P_m \) | Precision | Recall | mAP | Times (ms) |
|------------|----|----|----------|----------|---------|---------|---------|-----------|--------|-----|-----------|
| Yolov3     | 279| 269| 10       | 15       | 96.4    | 5.4     | 3.6     | 94.6      | 96.4   | 97.1| 138.2     |
| Yolov3-tiny| 279| 261| 18       | 23       | 93.5    | 8.4     | 6.5     | 91.6      | 93.5   | 93.9| 13        |
| ShuffleNet2| 279| 267| 12       | 20       | 95.7    | 7.0     | 4.3     | 93.0      | 95.7   | 96.9| 7.4       |
| MobileNet2 | 279| 262| 17       | 35       | 93.9    | 12.7    | 6.1     | 88.3      | 93.9   | 95.8| 20.8      |
| GhostNet   | 279| 264| 15       | 22       | 94.6    | 8.1     | 5.4     | 91.9      | 94.6   | 95.7| 6.9       |
| LSSNet     | 279| 273| 6        | 9        | 97.8    | 3.1     | 2.2     | 96.9      | 97.8   | 98.6| 10.1      |

5.2. Training Strategy

SSDD dataset is randomly divided into training set, validation set and test set according to the ratio of 7:2:1. During the training process, the Adam optimizer is used for parameter updating, which can adaptively adjust the learning rate, avoiding the problem of a sharp drop in the learning rate during model training. The initial learning rate set in the experiment is 0.001, all models are trained from scratch for 800 epochs, and the training batch size is 8. During training, save the optimal model until the training is completed; at the same time, save the last trained model so that it can be loaded and trained again.
5.3. Experiment Results

Figure 9 shows part of the prediction results of LSSNet on the SSDD test set. The green boxes in the figure are the predicted results of LSSNet, and the white boxes are the ground truth. It can be seen from the figure that the detection method we proposed has a good detection effect for ship targets of different scales on the far and near coasts. The detection accuracy rate reaches 96.9%, the recall rate reaches 97.8%, and the mAP reaches 98.6%. Compared with the classical lightweight network, its performance has a significant improvement. The mAP of LSSNet is 98.6%, which is 1.7% higher than Shufflev2, the lightweight network with the highest average precision. And its model size is 11.7M, only 1.9M larger than the lightest network model GhostNet. Due to the lightweight structure of the network, it only takes 10.1ms to realize the detection of a SAR image, so it can meet the basic real-time requirements.

![Figure 9 Results of LSSNet on SSDD test set.](image)

In order to compare and analyze the performance of the LSSNet, the following table II shows the evaluation indexes obtained by testing different detection models on the SSDD data set. From the results in the table, it can be seen that the lightweight networks ShuffleNetv2, MobileNetv2, GhostNet, etc. proposed based on optical images all have an mAP more than 95%, but their precision rate is lower than 95% and false alarm rate is high. Compared with the classic Yolov3, they even have lower precision rate. The LSSNet proposed in this paper can achieve 96.9% precision rate and 97.8% recall rate, and has the highest mAP of 98.6% in these comparison algorithms, which proves the effectiveness of the detection model for SAR ship detection tasks.

5.4. Algorithm Comparison Analysis

In order to show the real-time performance of different detection algorithms more intuitively, table II provides multiple evaluation indexes for different detection models, including network parameters, FLOPs, and model size. The amount of network parameters and FLOPs directly determine the inference speed, which are two important indexes for judging real-time performance. It can be seen from table II that GhostNet has the least amount of network parameters and FLOPs. Its model size is only 9.8M, but its false alarm rate is 5.0% higher than LSSNet, which is also a lightweight network and have higher mAP. Table I shows that the inference speed of LSSNet is only 3.2ms per SAR image slower than GhostNet, but LSSNet have a significantly better performance on false alarm rate 3.1% and missed alarm rate 2.2%. The model size of ShuffleNetv2 and LSSNet are almost the same. As is shown in table I, although ShuffleNetv2 has the best precision performance in classical lightweight network, its precision performance is still worse than our LSSNet.
Figure 10 Speed and precision performance of different lightweight networks.

LSSNet is 1.9M larger than the smallest model GhostNet, but it has significantly better precision performance than GhostNet. Compared with ShuffleNetv2, a lightweight network with the best precision performance in traditional lightweight networks, the model size of LSSNet is decreased by 0.1M and mAP is increased by 1.7%. Figure 10 shows the speed and precision performance of different lightweight networks. The superior performance of LSSNet can also be seen from the figure. It can well balance the speed and precision of SAR ship detection. Besides, it also reflects the necessity of redesigning a backbone network for specific task.

**TABLE II. NETWORK PARAMETERS, FLOPS AND MODEL SIZE OF DIFFERENT METHODS**

| Network Parameters |
|--------------------|
| **FLOPs**         |
| **Model Size**    |

| Method   | FLOPs       | Model Size |
|----------|-------------|------------|
| Yolov3   | 61,497,430  | 123.4M     |
| Yolov3-tiny | 8,666,692  | 17.4M      |
| ShuffleNetv2 | 5,740,644  | 11.8M      |
| MobileNetv2  | 8,096,724  | 65.4M      |
| GhostNet    | 4,710,796  | 9.8M       |
| **LSSNet** | **5,673,904** | **11.7M**  |

Using the above detection algorithms all can achieve efficient detection of far-shore targets. However, for some near-shore ship targets, the detection effects of different algorithms are obviously different. The average precision of the detection network is often determined by the ability to detect difficult ship samples. Therefore, even if the average precision is increased by about 1-2%, it can be considered as a significant improvement in the detection ability of difficult ship samples. The figure 11 below is a representative of some difficult samples. We show the detection capabilities of different detection algorithms on the SAR image samples. The green box represents the prediction result, and the white box represents the ground truth. It can be seen from the figure that the LSSNet proposed in this paper has better detection capabilities for difficult targets than classic lightweight networks, and it can even detect two ship targets that are very close while classical lightweight network just regard it as one ship target. Besides, we can see from the Figure 11, the results of LSSNet shows less false alarm targets and missing alarm targets.
Compared with the classic lightweight network method, LSSNet better balances the speed and precision performance, which adapts to the specific tasks of SAR ship detection. In addition, LSSNet has lower network parameters, FLOPs and smaller model size compared with a method in literature [18], which is also designed for SAR ship detection. Besides, the detection speed and precision of LSSNet are faster and higher. LSSNet achieved such excellent performance, the reasons may be:

5.4.1. This method densely concats the deep and shallow features to achieve efficient use of ship feature maps. In the deep network, shortcuts are introduced in the stacked denseblocks, which enhance the deep supervision and strengthen the information flow between upper and lower layers;

5.4.2. In the basic structure of denseblock, the conventional convolution is replaced with a depthwise separable convolution, and the SE module is added later. The new structure improves the detection speed and decreases the model size while ensuring that the precision performance is not compromised.

5.5. Ablation Experiment
LSSNet is a lightweight network designed specifically for SAR ship detection tasks. The above experimental results have also proved the great performance of LSSNet for SAR ship detection. However, in order to pursue a more excellent detection model, this paper also explores its structural variants. The improvement of its variants is mainly based on the following two directions:

5.5.1. When downsampling in LSSNet, the conventional convolution is used. We can consider using depthwise separable convolution for downsampling instead of conventional convolution to decrease the model size, and introduce SE module to reduce the loss of ship feature information during downsampling;

5.5.2. LSSNet uses directly stacked denseblocks in the shallow layers, and the stacked denseblocks with shortcuts(ShortcutDB) in the deeper layers. We can consider removing the shortcuts and explore the impact of the shortcuts on LSSNet detection performance.
TABLE III. ACCURACY PERFORMANCE OF LSSNet’s Variants

| Downsampling with traditional convolution | Downsampling with depthwise separable convolution | Downsampling with SE module | With shortcuts | Precision (\%) | Recall (\%) | mAP (\%) | Time |
|-----------------------------------------|-----------------------------------------------|----------------------------|---------------|----------------|-------------|----------|------|
| √                                      | √                                             | ×                          | √             | 94.7           | 95.3        | 96.1     | 8.9  |
| ×                                      | √                                             | √                          | √             | 94.4           | 96.0        | 96.8     | 9.1  |
| √                                      | ×                                             | ×                          | ×             | 89.8           | 93.5        | 94.2     | 8.7  |
| √                                      | ×                                             | ×                          | √             | 96.9           | 97.8        | 98.6     | 10.1 |

TABLE IV. REAL-TIME PERFORMANCE OF LSSNet’s Variants

| Downsampling with traditional convolution | Downsampling with depthwise separable convolution | Downsampling with SE module | With shortcuts | Network Parameters | FLOPs | Model Size |
|-----------------------------------------|-----------------------------------------------|----------------------------|---------------|-------------------|-------|------------|
| √                                      | √                                             | ×                          | √             | 5,380,244         | 10,156,234 | 11.1M      |
| ×                                      | √                                             | √                          | √             | 5,397,860         | 10,365,896 | 11.2M      |
| √                                      | ×                                             | ×                          | ×             | 4,977,848         | 9,824,783  | 10.2M      |
| √                                      | ×                                             | ×                          | √             | 5,673,904         | 11,163,958 | 11.7M      |

Table III, IV shows the experimental results obtained by the above improved method. From the results in the tables, it can be seen that when only using the depthwise separation convolution to replace the conventional convolution for downsampling, the network parameters, FLOPs and the model size are all decreased to a certain extent. However, the average precision is also decreased by 2.5%. No matter whether SE module is introduced behind the depthwise separation convolution, the average precision cannot be effectively improved, even if it is decreased by 1.9%. The use of depthwise separable convolution for downsampling slightly improves the detection speed, but the average precision is greatly reduced. This also proves the rationality of LSSNet using conventional convolution for downsampling. When the shortcuts in LSSNet are removed, the average precision is decreased from 98.6% to 94.2%, and the inference time of each SAR image is only 1.4ms faster, which proves the necessity of adopting shortcuts. Obviously, the above exploration experiments have proved the superiority of the LSSNet designed. Perhaps LSSNet still has room to explore, but considering our limited energy, it is impossible to traverse the exploration of all network variants. Based on the above exploration experiment, the structure of LSSNet has the best performance in terms of balancing speed and precision in SAR ship detection. And previous experimental results also well prove this point.

5.6. Supplementary Explanation
The baseline used in this paper is SSDD data set. Many scholars have used this baseline to verify the designed detection algorithm. However, many papers pointed out that the precision performance will be different because the data set does not have a clear data division method. The SSDD data set has 1160 SAR ship images, of which simple samples in the far shore areas account for the majority, and difficult samples with offshore targets only account for a minority. If there are too many simple samples in the test set generated by random division, the average precision of the detection network will naturally increase. Therefore, in order to compare and analyze the performance of different detection algorithms,
the data set needs to use the same division method. The LSSNet proposed and its comparison detection algorithms all use the same train set, validation set and test set. The previous experimental results have proved that LSSNet has good generalization ability and strong robustness. Compared with the classical lightweight detection algorithm, it has higher precision and also meets the basic needs of real-time performance.

LSSNet is a lightweight network specifically designed for SAR ship detection tasks. It can achieve high-speed and high-precision detection of SAR ship images. Its model size has been decreased to 11.7M, which will help future FPGA and DSP hardware transplant. We believe that the lightweight network model is already relatively small, and the focus of subsequent research work should be on how to improve the average precision of the lightweight network. Specifically, how we improve the average precision, enhance the network generalization capability and strengthen the ability to detect difficult samples on the basis of not significantly changing the model size. We think lightweight networks with better performance can be studied from two aspects. One is to redesign the network structure to find a more efficient backbone, and the other is to design more suitable anchors and a better loss function for SAR image targets.

6. CONCLUSION

This paper proposes a lightweight detection network LSSNet for SAR ship detection tasks, which can achieve high-speed and high-precision precision detection of ship targets. This method uses a combination of denseblock and stacked denseblocks with shortcut to improve the average precision and uses depthwise separable convolution to achieve the purpose of lightweight. Verified by SSDD data set, LSSNet has good generalization ability and strong robustness. Besides, the model size of LSSNet is only 11.7M, which is helpful for the transplantation of resource-scarce hardware devices in the future.

REFERENCES

[1] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. ECCV, 2018.
[2] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. GhostNet: More Features from Cheap Operations. CVPR, 2020.
[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. CVPR, 2018, pp. 4510-4520.
[4] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. CVPR, 2017.
[5] Robert J. Wang, Xiang Li, and Charles X. Ling. Pelee: A Real-Time Object Detection System on Mobile Devices. NIPS, 2018.
[6] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis., Jul. 2017, pp. 609–616.
[7] 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.
[8] Z. Shen, Z.Liu, J.Li, Y.-G.Jiang, Y. Chen, and X. Xue, “DSOD: Learning deeply supervised object detectors from scratch,” in Proc. IEEE Int. Conf. Comput. Vis., Aug. 2017, pp. 1–10.
[9] LI Jianwei, QU Changwen, and SHAO Jiaqi. Ship detection in SAR images based on an improved faster R-CNN[C]. 2017 SAR in Big Data Era: Models, Methods and Applications, Beijing, China, 2017: 1–6. doi: 10.1109/BIGSAR DATA.2017.8124934.
[10] Du Lan, Liu Bin, Wang Yan, et al. SAR image target detection algorithm based on convolutional neural network [J]. Journal of Electronics and Information Technology, 2016,38 (12): 3018-3025.
[11] P. Zhou, G. Cheng, Z. Liu, S. Bu, and X. Hu, “Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping,” Multidimensional Syst. Signal Process., vol. 27, no. 4, pp. 925–944, Oct. 2016.
[12] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, “Advanced deep learning techniques for salient and category-specific object detection: A survey,” IEEE Signal Process. Mag., vol. 35, no. 1, pp. 84–100, Jan. 2018.

[13] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, “Multi-scale object detection in remote sensing imagery with convolutional neural networks,” ISPRS J. Photogramm. Remote Sens., vol. 145, pp. 3–22, Nov. 2018.

[14] H. Dai, L. Du, Y. Wang, and Z. Wang, “A modified CFAR algorithm based on object proposals for ship target detection in SAR images,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 12, pp. 1925–1929, Dec. 2016.

[15] X. Wang and C. Chen, “Ship detection for complex background SAR images based on a multiscale variance weighted image entropy method,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 2, pp. 184–187, Feb. 2017.

[16] J. Zhao, W. Guo, Z. Zhang, and W. Yu, “A coupled convolutional neural network for small and densely clustered ship detection in SAR images,” Sci. China Inf. Sci., vol. 62, Apr. 2019, Art. no. 42301.

[17] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. (2016). PVANET: Deep but lightweight neural networks for real-time object detection. arXiv preprint arXiv:1608.08021

[18] ZHANG Xiaoling, ZHANG Tianwen, SHI Jun, et al. High-speed and High-accurate SAR ship detection based on a depthwise separable convolution neural network[J]. Journal of Radars, 2019, 8(6): 841–851. doi: 10.12000/JR19111.

[19] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[20] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv preprint arXiv:1707.01083.

[21] Hoang V T, Jo K H. PydMobileNet: Pyramid Depthwise Separable Convolution Networks for Image Classification. 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE). IEEE, 2019: 1430-1434.

[22] Haase D, Amthor M. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 14600-14609.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren et al. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385, 2015.

[24] Yuxi Li, Jiwei Li, Weiyao Lin, et al. Tiny-DSOD: Lightweight Object Detection for Resource-Restricted Usages. arXiv preprint arXiv:1807.11013, 2018.

[25] Youngwan Lee, Joong-won Hwang, Sangrok Lee, et al. An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. CVPR, 2019.

[26] Youngwan Lee and Jongyoul Park. CenterMask: Real-Time Anchor-Free Instance Segmentation. arXiv preprint arXiv:1911.06667, 2020.

[27] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. IEEE Int. Conf. Mach. Learn., Feb. 2015, pp. 448–456.

[28] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[J]. arXiv preprint arXiv:1709.01507, 2017, 7.

[29] REDMON J and FARHADI A. YOLOv3: An incremental improvement[J]. arXiv: 1804.02767, 2018.

[30] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, et al. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. In CVPR, 2019.

[31] Adam Paszke, Sam Gross, Francisco Massa, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library, In NIPS, 2019.

[32] LI Jianwei, QU Changwen, and PENG Shujuan. A ship detection method based on cascade CNN in SAR images[J]. Control and Decision, 2019, 34(10): 2191–2197.
[33] LI Jianwei, QU Changwen, PENG Shujuan, et al. Ship detection in SAR images based on convolutional neural network[J]. Systems Engineering and Electronics, 2018, 40(9): 1953–1959. doi: 10.3969/j.issn.1001-506X.2018.09.09.

[34] YANG Long, SU Juan, LI Xiang. Ship detection in SAR images based on deep convolutional neural network[J]. Systems Engineering and Electronics, 2019, 41(9): 1990–1997. doi: 10.3969/j.issn.1001-506X.2019.09.11.

[35] HU Changhua, CHEN Chen, HE Chuan, et al. Ship small target detection based on deep convolution neural network in SAR image[J]. Journal of Chinese Inertial Technology, 2019, 27(3): 397–405, 414. doi: 10.13695/j.cnki.12-1222/o3.2019.03.0