FPA-DNN: A Forward Propagation Acceleration based Deep Neural Network for Ship Detection

Feng Wang
School of Computer Science
Wuhan University
Wuhan, China
fengwang@whu.edu.cn

Fanshu Liao
School of Computer Science
Wuhan University
Wuhan, China
2019202110067@whu.edu.cn

Huiqing Zhu
School of Computer Science
Wuhan University
Wuhan, China
474931806@qq.com

***Abstract*—Ship detection in optical satellite images has played an important role in the field of remote sensing for a long time. Many detection methods have been proposed to address the ship detection problem, and most of them mainly focus on the improvement of detection accuracy but rarely pay attention to the detection speed. In this paper, we not only consider the improvement of detection accuracy, but also try to speed up the detection process. Based on the YOLOv2 model, we propose a forward propagation acceleration-based deep neural network model (FPA-DNN) to enhance the performance of the ship detection. The FPA-DNN model is a hybrid learning model, in which the deep neural network model LSDN can effectively reduce the number of parameters and improve the detection speed with no accuracy loss, and the pruning based forward propagation acceleration algorithm can remove the redundant convolution kernels and further speed up the detection process. Experimental results on the optical remote sensing image dataset show that, compared with several state-of-the-art deep learning models, 1) the LSDN model outperforms the others on the detection accuracy and detection speed; and 2) the FPA-DNN model can further improve the detection accuracy and speed up the detection process significantly.**

Keywords: Convolution neural network; Ship detection; Network Acceleration

I. Introduction

Automatic ship detection in optical satellite images has played an important role in port management, cargo transportation, and military defense. With the rapid development of optical remote sensing satellites, ship detection and identification based on large-scale optical remote sensing images has become a significant research topic [1]. Although many ship detection methods have been proposed, it still poses a great challenge due to the existence of uncertainties such as light, disruptors, the density of the ship, and so on.

In the past few decades, some traditional methods which segment the sea-land through the features of the texture and the shape have been proposed for ship detection [2][3]. However, due to the existence of huge noises such as light, disruptors, the density of the ship in the remote sensing images, these methods fail to be applied in the complex scenarios. The optical remote sensing images including different scenarios and intensive objects, i.e., images with calm seas, images with waves and fog, and images with housing, roads, and other complex backgrounds, are extremely difficult to be detected efficiently.

With the development of artificial intelligence techniques, the deep neural network models (DNNs) have been proposed, such as R-CNN [4], SPP-NET [5], Fast R-CNN [6], Faster R-CNN [7] and YOLO [8], and many of them have been applied for ship detection [9][10][11]. Although these methods have shown promising performance, most of them mainly focus on the improvement of detection accuracy but rarely pay attention to the detection speed. For the ship detection in real applications, it is required to get an accurate detection result in a short time. However, these deep learning networks with complex network structures and a large number of redundant parameters require a long time for detection, which are not suitable for ship detection. How to quickly detect ships from optical satellite images with high accuracy has become a challengeable research issue.

Recently, many methods which aim at reducing the size of the deep neural network model have been proved to shorten detection time efficiently. The most common one is to use a compact network model with fewer parameters, i.e., lightweight network model [12][13][14][15][16]. Another effective approach is to compress the network by pruning a trained dense network [17][18][19][20][21][22]. These methods are suitable for the network that contains a large number of redundant parameters. However, these methods rarely accuracy loss during the detection process.

In this paper, we propose a forward propagation acceleration-based deep neural network model (FPA-DNN) to enhance the performance of the ship detection with optical remote sensing images. The FPA-DNN model is a hybrid learning model which includes two acceleration methods: the first one is the deep neural network model LSDN which based on the YOLOv2 model and can effectively reduce the number of parameters and improve the detection speed with no accuracy loss; the second one is the pruning based forward propagation acceleration algorithm which can remove the redundant convolution kernels and further speed up the detection process. Then we evaluate our proposed models on the optical remote sensing image dataset including 800 sea surface images in different scenarios. Experimental results show that, compared with several state-of-the-art models, the proposed LSDN model and FPA-DNN model outperform

*Corresponding author: Feng Wang, fengwang@whu.edu.cn.

the other deep neural network models on both detection accuracy and detection speed, and the FPA-DNN model shows significant improvement on the detection speed.

The rest of this paper is organized as follows. Section II gives an introduction of related work about target detection and network acceleration methods. Section III presents the detailed information of our FPA-DNN model. Experimental results and discussion are reported in Section IV. The conclusion and future work are given in Section V.

II. Related Work

A. DNNs for Target Detection

Deep learning has made even greater achievements in the field of target detection. In 2014, Girshick et al. proposed R-CNN [4] by using image feature information to determine target candidate regions, which greatly improved the accuracy of target detection. Subsequently, He et al. proposed SPP-NET [5], compared with R-CNN, it simplifies the normalization process of image input and greatly speeds up the target detection. However, as its basic structure is similar to R-CNN, it still has some problems such as tedious training steps and cannot be applied in the real-time field. In order to solve the existing problems of SPP-NET, Ren et al. proposed Fast R-CNN [6] and Faster R-CNN [7], which are both simple and Fast target detection frameworks based on R-CNN and SPP-NET, and adopted the multi-task loss function to simplify the steps of network training and reduce the training cost. However, both Fast-R-CNN and Faster-R-CNN are based on R-CNN, and the process of determining the candidate box is still time-consuming. Although the speed has been improved compared with SPP-NET, the detection speed still lags behind the requirement of real-time detection. After this, Redmon et.al proposed a new target detection framework YOLO [8]. Unlike R-CNN based models, YOLO can find the category of the target in the current region by regression on multiple candidate regions.

These deep neural network models have high accuracy in target detection tasks, but there are many parameters in these models which require large amount computation time and high-performance hardware devices. Therefore, these models cannot meet the speed requirement of ship detection.

B. Network Acceleration Methods

In order to improve the detection speed of deep neural network models, lots of network acceleration methods have been proposed to accelerate the detection speed of the network, and reduce the network storage space purposes. These network acceleration methods can be divided into the following two categories: lightweight network model-based methods and pruning methods.

1) Lightweight Network Model-based Methods: Lightweight neural network models mainly focus on building network models with small size and small computation cost. The most famous lightweight network model is Xception which adopts depthwise convolution and pointwise convolution in the network structure [12]. The main idea of Xception is to divide the convolution kernel into C groups and deal with each group separately, so that the convolution operation between different groups can be synchronized, which can help reduce the convolution operation time and speeding up the network prediction speed. In Xception, the pointwise convolution uses the convolution kernel of $1 * 1$ to convolve the input image. In this way, the number of channels of the feature map is reduced to compress the network structure. ShuffleNet proposed by the Face++ team employs the inter-group exchange information mechanism to the network's packet convolution and solves the problem that packet convolution cannot carry out information interaction [13]. ShuffleNet designs the network structure by means of packet convolution and channel shuffle, reducing network model parameters and making it lightweight. Mobile-net V1 [14] is a convolutional neural network with a small volume and small computation proposed by Google. This network adopts the idea of depthwise separable convolution instead of standard convolution to compress the network model and enable the network to be used on mobile devices. In addition, for the automatic design neural network based on the neural network architecture search, NASNet [15] is based on AutoML (automatic design machine learning model) method. This model first searches the neural network architecture on small data sets, then uses AutoML to find the optimal convolutional layer and automatically creates the final network through multiple stack optimization. In the research based on AutoML automatic model compression, traditional model compression technology relies on manual design and needs to balance network prediction time, accuracy, and memory size, which is a time-consuming process. Google proposed AMC [16] (AutoML for Model compression (AMC)), which uses reinforcement learning to provide model compression strategies and is suitable for model compression.

2) Pruning Methods: The network pruning methods can be divided into unstructured pruning methods [17] [18] [19] and structured pruning methods [20] [21] [22]. Unstructured inter-layer pruning methods directly prune the parameters of each layer of the network. Liu et al. proposed the networks slimming [23] method, which uses the scaling factor $gamma$ in the Batch Normal algorithm [24] as the factor to evaluate the contribution of the upper output. The smaller the $gamma$ is, the less important the corresponding neuron is, which means the neuron can be clipped. The pruning methods proposed by Anwar et al. determines whether the current filter is deleted by scoring different filters [25]. In order to avoid the impact of pruning on the accuracy of network prediction, this method iterates pruning and model training continuously to ensure that the model is compressed without loss of accuracy. The pruning algorithm achieves the goal of network compression by removing redundant connections in the network. And after removing some connections, the network detection speed can also be improved.

To sum up, compared with the traditional detection methods, the detection methods based on deep learning models have better generalization, but a large number of parameters and complex network structure may lead to huge time cost on ship detection. Although the acceleration methods can help improve the detection speed, they will involve accuracy loss in the detection process.

III. FPA-DNN

In order to enhance the performance of ship detection, we firstly develop a deep neural network model LSDN based on the YOLOv2 model, which can effectively reduce the number of parameters. To further improve the detection speed of the network model, we propose a forward propagation acceleration algorithm (FPA) and explore this FPA algorithm into the previous LSDN network to get a new deep neural network model (FPA-DNN) for ship detection.

A. Lightweight Ship Detection Network (LSDN)

Since ship detection requires fast detection speed, in order to reduce the number of network parameters and improve the speed of network detection, based on the YOLO V2 model, we firstly propose a new lightweight detection network model for ship detection, named as LSDN. The overall framework of LSDN is shown in Figure 1.

In Figure 1, the blue box represents the convolutional layer, where $Conv_i$ represents the convolutional layer of the i^{th} layer, $13 * 13(16)$ means the size of the convolutional kernel of the current convolutional layer is $13 * 13$, and the number of convolutional kernels is 16. The red box represents the maximum pooling layer, where $2 * 2 + 2(s)$ represents the $2 * 2$ pooling with a step size of 2. And the yellow box represents the route layer, which is also the feature fusion layer.

The LSDN model makes the following improvements on the YOLOv2 model. Firstly, the multi-layer convolutional layers in YOLOv2 are replaced by the combined structure of convolution and maximum pooling. In the YOLOv2 network model, it adopts multi-layer convolution before pooling, which is suitable for feature extraction of high-resolution images. In the convolutional neural network, each point in the feature graph obtained from the input image after multiple convolutions contains more image information. However, the combined method of multi-layer convolution and re-pooling is not suitable for ship detection.

As shown in Figure 2, the target in the red rectangle is the ship in the satellite image. It can be seen that the ship in the satellite image is very small. Therefore, if multi-layer convolution is used for feature extraction, with the increase of the convolution layer, the pixel information in the convolution image is much larger than the size of the ship in the image. And the information contained in each pixel may not only contain the ship, such as the ship beside the island, which will decrease of detection accuracy. Figure 3 shows the detection results interfered by the island information. Since the features of small islands and ships are similar, it is easy to get false detection results.

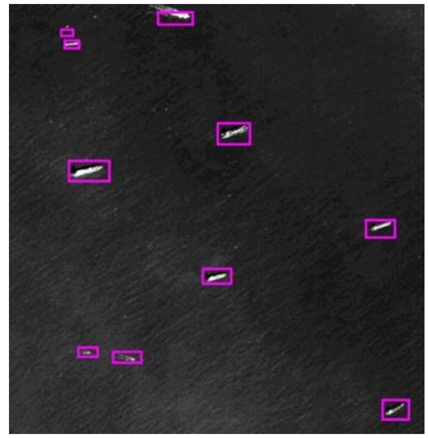

Fig. 2
Ships in satellite images.

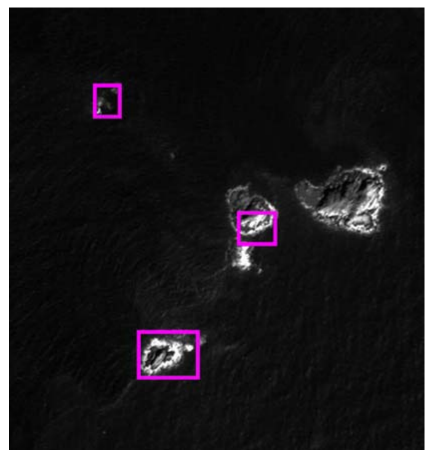

Fig. 3
Detection results of island information interference.

In LSDN, we employ a combined structure of convolution and maximum pooling method so as to avoid misidentification caused by continuous feature extraction of multiple convolutional layers. And the maximum pooling used in LSDN would have certain denoising effects, which can improve the efficiency of feature extraction to avoid the detection on islands or spray and get more accurate detection results.

Secondly, the general convolution is replaced by the depthwise separable convolution. From the 13^{th} layer to the 16^{th} layer in LSDN, we use depth separable convolution in addition to combining the convolution layer and pooling layer. The differences between the general convolution process and the depthwise separable convolution process are shown in Figure 4 and Figure 5.

For the general convolution operation, when the size of the input image is $h * h * c$, where h is the length and width of the image, and c is the number of channels of the image, the number of channels of the output image is n after convolving with n convolution kernels by the size of $k * k * c$. For the depthwise separable convolution operation, it can be divided

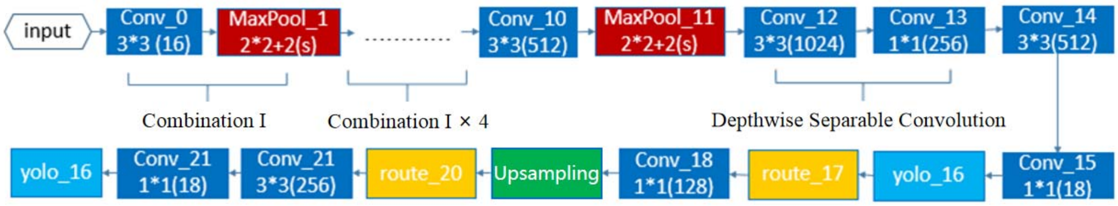

Fig. 1
Framwork of LSDN.

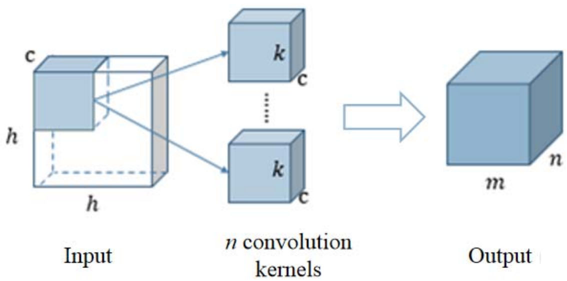

Fig. 4
General convolution.

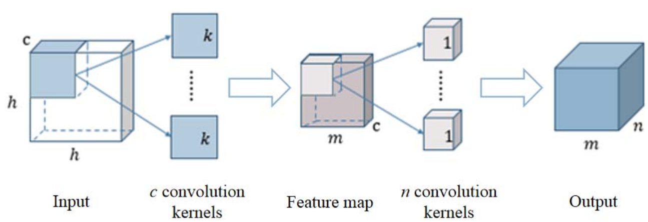

Fig. 5
Depthwise separable convolution.

into two steps. The first step is doing convolution on the input image with c convolution kernels by the size of $k*k*1$ and get the feature image by the size of $m*m*c$. In the second step, the convolution is carried on the feature image with n convolution kernels by the size of $1*1*c$. In this way, the computation cost of convolution can be greatly decreased.

As shown in Figure 1, in the 13^{th} layer, for example, the input feature image size is $13*13*512$, and the output image size is $13*13*256$, which is exactly the output of the 15^{th} layer in the network. If using the general convolution, it requires at least 256 convolution kernels by the size of $13*13*512$, and the total number of calculations in the convolution process is $13*13*512*256*13*13$. However, if the depthwise separable convolution is adopted, the original convolution operation is divided into two parts. The first part uses a convolution kernel of $13*13*1$, and the second part uses a convolution kernel of $1*1*256$. In the second part, the convolution kernel of size $1*1$ is used to reduce the feature dimension. In the 13^{th} layer of the ship detection network, it can be seen that the input of the original 13^{th} layer is $13*13*1024$ and the feature dimension is 1024. After the convolution, the output feature dimension is 256, which is determined by the number of convolution kernels. The size of the feature image obtained by the depthwise separable convolution is same with that obtained by the general convolution, but the number of parameters of the convolution kernels has reduced from the original $13*13*512*256$ to $13*13*1*1024+1*1*1*256$, which means the depthwise separable convolution can effectively compress the network model and reduce the calculation cost.

Finally, in order to guarantee the detection accuracy, we use a feature fusion method in the ship network LSDN, which can combine shallow features with deep features. In the 20^{th} layer and 21^{th} layer of LSDN, the upsampling and route method are used which can splice the deep and shallow features together. In this way, different features can be fused and help the network model get higher detection accuracy.

B. Forward Propagation Acceleration (FPA)

In LSDN, each layer uses a multi-dimensional convolutional kernel with many convolutional parameters. Take the 15^{th} convolutional layer as an example, there are 512 convolutional kernels by the size of $3*3*256$ in this layer with a total of 1179684 weight parameters. In order to further improve the detection speed of the network, we propose a forward propagation acceleration algorithm and explore it into the previous LSDN model for ship detection.

As mentioned above, there are two main ways to accelerate the network model, i.e., lightweight network model-based methods and pruning methods. Here, the forward propagation acceleration algorithm FPA is proposed based on the pruning methods. There are two steps in FPA. The first step is analyzing the weight distribution of the parameters in convolution layers by measuring the importance of the convolution kernels, which can help find the redundant weight parameters in LSDN. These redundant weight parameters could be deleted in the second step by the pruning method.

1) Weight Distribution Analysis in LSDN: As we know, different pruning algorithms may have different experimental effects on different network models. For example, static pruning is more suitable for sparse network structures. Therefore, it is reasonable to analyze the weight distribution of the convolution layers before designing the network pruning method.

According to the weight distribution analysis on the parameters in convolution layers of LSDN, we find that there are a large number of redundant parameters. Most weights in LSDN are small, and nearly 90% weights belong to the $[0.5, 0.5]$ interval, in which nearly 30% are 0. From the above analysis on the pruning algorithm, using a single weight for pruning may result in an incomplete network connection. Meanwhile, in the LSDN model, a large number of continuous weight values are 0, which means the weights in some convolution kernels are all 0 or in the interval $[0.01, 0.01]$. Hence, it is very necessary to do the pruning on the convolution kernels in LSDN.

In LSDN, only the part whose convolution kernel weight value is close to 0 is considered to be pruned. After pruning, we testify the performance of the new network model. If the detection accuracy loss of this new network model is within a certain range, it is not necessary to do retraining. Since the ship images for detection are satellite images, the waves or fog would affect the detection results. If the network is too complicated, it might get overfitting. The pruning network can reduce the network complexity, which is helpful to keep the detection accuracy to some extent.

2) RCKP Pruning: In order to further improve the detection efficiency, the second step of the FPA algorithm adopts the weighted pruning method RCKP to simplify the model and improve the detection speed. In RCKP, we do the pruning by using the sum of the absolute value of the convolution kernel weights by setting a threshold to determine whether the network convolution kernels need to be pruned or not. It can help avoid deleting the convolution kernels whose weights are small in the corresponding layer but is relatively large to other convolution kernels in other layers.

The threshold t in RCKP is determined by the pruning parameter sen [26], which can be calculated by Formula 1. Considering the high cost of retraining, in order to avoid the retraining of the network model, we choose t with a small pruning sensitivity as the pruning threshold at the beginning of the pruning process.

$$sen(t) = \frac{OA}{PA} PR(t) * TRR = \frac{\Delta Acc}{PR(t) * TRR} \tag{1}$$

where OA represents the detection accuracy of the original network, and PA represents the detection accuracy of the network after pruning. $PR(t)$ refers to the proportion of pruning in the network when the threshold value is t, that is, the ratio of the weight of pruning to the ownership weight in the network, which can be calculated by Formula 2.

$$PR(t) = \frac{N_P}{N_A} \tag{2}$$

where N_p refers to the number of pruning convolution kernel when the threshold value is t, and N_A refers to the weight of convolutional kernel in the original network.

In Formula 1, TRR is the ratio of time reduction after pruning, that is, the time of network acceleration and the time of the original network model to detect a picture, which can be calculated by Formula 2. And T_{org} represents the time taken by the original network model to detect each picture, and T_{acc} represents the time taken by the network model to detect each image after pruning.

$$TRR = \frac{T_{org} - T_{acc}}{T_{org}} = \frac{\Delta T}{T_{org}} \tag{3}$$

After determining the network threshold t according to the pruning sensitivity, the convolutional kernel pruning is carried out in LSDN. Suppose there are N convolution kernels in the i^{th} layer of the network, the first j in the i^{th} layer convolution a convolution kernels can be expressed as $F_{i,j} \in R^{n_i * k * k}$, where k is the size of the convolution kernel, n_i is the dimension for convolution kernels. During the pruning process, we make a statistics on the L_1 normalization of each convolution kernel $F_{i,j}$, and get $\sum |F_{i,j}|$. Then we remove those convolution kernels whose statistical value is smaller than the threshold t.

The pruning method RCKP can be summarized in Algorithm 1.

Algorithm 1 RCKP Pruning Process

Require: the network to be pruned;
Ensure: pruned network;
1: $\Sigma|F_{i,j}| \leftarrow$ Calculate the sum of the absolute values of weights in all convolutional layers;
2: Sort $\Sigma|F_{i,j}|$ of all convolution kernels;
3: $t \leftarrow$ Take the top 5% of sorted results in Line 2 as the initial threshold;
4: Prune the network according to t;
5: Calculate the pruning sensitivity of current threshold t according to Formula 2;
6: Increase t by $\Delta t = 0.5$;
7: Repeat 3-6, until t reaches maximal value, and the pruning sensitivity gets to the minimal value;
8: Set the current threshold t_max as the threshold of network pruning;
9: Delete all the convolutional kernels whose $\Sigma|F_{i,j}|$ is less than t_{max};

After the RCKP pruning process, we can get the pruned network model. We then explore this FPA algorithm in the previous LSDN model and get a new deep neural network model FPA-DNN for ship detection. The construction process of the FPA-DNN model can be summarized in Algorithm 2.

Algorithm 2 FPA-DNN

Require: the deep neural network based on YOLOv2;
Ensure: the network after acceleration, FPA-DNN;
//LSDN
Construct the network model by the combined structure of convolution and maximum pooling;
Do the convolution operation by depthwise separable convolution method;
Train the parameters by the fusion features of shallow features and deep features;
//FPA
Analyze the weight distribution of the parameters in convolution layers of LSDN;
Prune LSDN by **Algorithm 1**;
Return the final ship detection network FPA-DNN.

IV. Experimental Study

In order to verify the effectiveness of the proposed convolution network models, we use several evaluation metrics to compare the results with some traditional deep neural network models in practical applications. All experiments are conducted on a PC with an Intel Core i7-770 running at 3.6GHz with 8GB of memory. The experimental platforms are 64-bit Microsoft Visual Studio 2015 and OpenCV 3.4.0.

We firstly testify the performance of the LSDN model, and then evaluate the performance of the FPA-DNN model.

A. Data Set

The data set used in the experiment includes 800 sea surface images collected by satellites, which have the characteristics of high resolution and high noise, and the image size is $1024*1024$. The 800 images are categorized into three kinds of scenarios: a) images with calm sea, b) images with fog, and c) images with housing, roads, and other complex backgrounds.

B. Evaluation Metrics

Performances of algorithms are evaluated by some widely used metrics, including $Precision$, $Recall$, $F1$, AP (Average Precision), $AverageIoU$ (Intersection over Union), and $Detectionspeed$ (Detection speed).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Here, TP and TN refer to the number of true positives and true negatives, FP and FN represent the number of false positives and false negatives.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

AP (Average Precision) is another metric on the network model's detection accuracy.

$$AP = \frac{1}{11} \sum\nolimits_{r \in (0,0.1,\ldots,1)} Pinterp(r) \tag{7}$$

$$Pinterp(r) = \max_{\tilde{r}, r \geq r} P(\tilde{r}) \tag{8}$$

$AverageIoU$ is an evaluation metric that can reflect the degree of overlap between the actual location of the target to be detected and the predicted location of the network model. IoU can be calculated by the following formula.

$$IoU = 2 * \frac{A_O}{A_U} \tag{9}$$

where A_O is the area where the area of the external rectangular box predicted by the single network model overlaps with the area of the rectangular box predicted by the network model, and A_U is the sum of the area of the external rectangular box predicted by the network model and the external rectangular box at the correct position of the ship. As shown in Figure 6, the green box is the ground-truth bounding box for the ship, and the red area is the network model's bounding box for the ship.

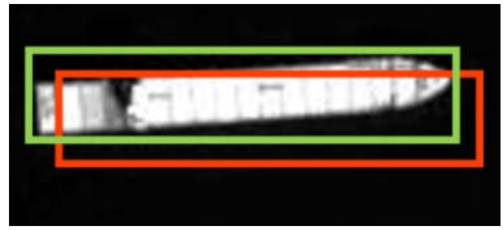

Fig. 6
IoU metric.

The detection speed $Dspeed$ is calculated by Formula 10.

$$Dspeed = \frac{T_{all}}{N_S} \tag{10}$$

where N_S represents the total number of test samples, and T_{all} represents the time taken to test N_S.

To evaluate the accelerating performance of FPA-DNN, we propose an evaluation metric $Ratio$ to measure the compression ratio of the models.

$$Ratio = \frac{N_{TP}}{N_{TW}} \tag{11}$$

where N_{TP} is the total number of pruned weights, and N_{TW} is the total number of weights in the original network. $Ratio$ can well reflect the Ratio of network compression. It should be noticed that $Ratio$ is used to measure the compress ratio of the network model.

C. Results and Discussion

1) Results Comparison on LSDN: In order to validate the performance of the LSDN model, we select some convolution neural networks with different structures which have already been widely used in target detection tasks for comparisons, such as YOLOv2 [27], YOLOv2-Tiny [27], and YOLOv3-Tiny [28].

Figure 7 shows the detection results using different network models in different scenarios. From Figure 7, we can see that each detection network model can correctly detect ships in the calm sea scenario in most cases. And compared with other detection network models, the LSDN model can obtain a more accurate ship location, that is, the detection box marked in the detection results is closer to the standard box. For the ship detection in the complicated scenario, when the image is included with similar background information, such as waves, fog, etc., the YOLOv2-Tiny model is easy to recognize the complex background as ships, which is the same as the YOLOv2 model and YOLOv3-Tiny model. But the LSDN model cannot be affected by these kinds of noise and has shown better performances than the others. When the image contains buildings such as houses and roads, although the YOLOv2 model and the YOLOV3-Tiny model can detect only some of the ships, while the LSDN model can detect all ships. By comparing the detection results of different test images, it can be seen that the LSDN model is less disturbed

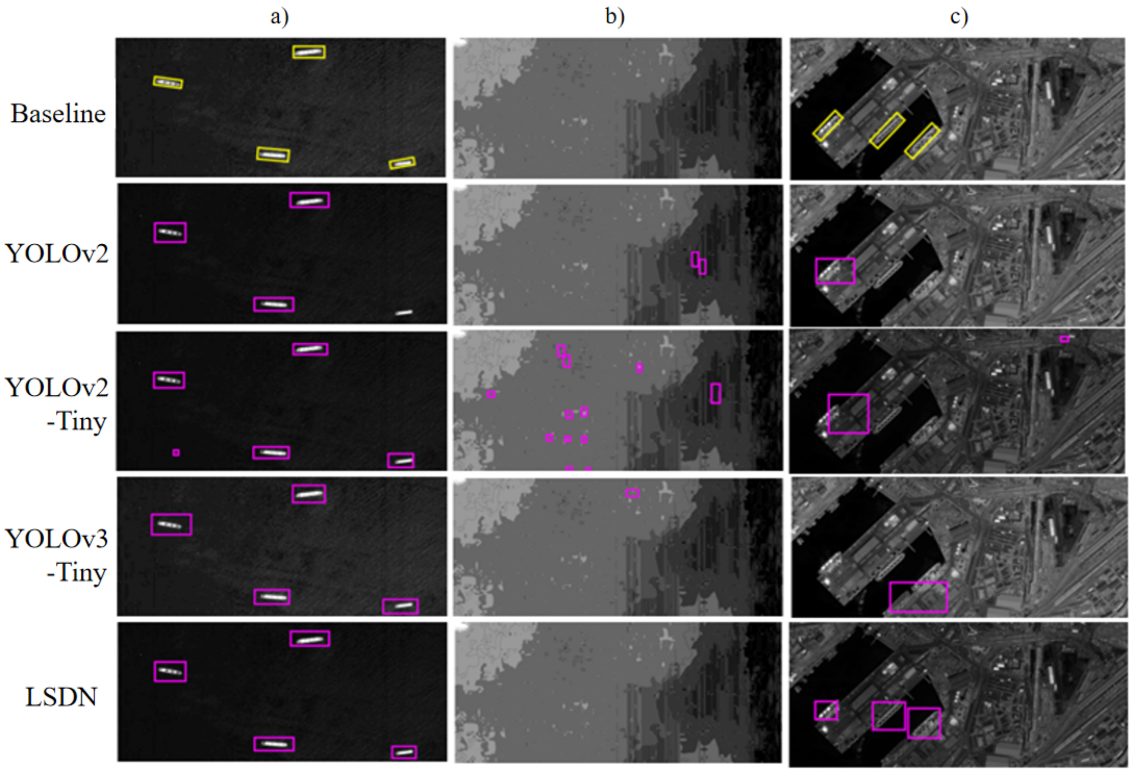

Fig. 7
Detection results of different network models in different scenarios.

by noises and can detect the position of ships more accurately than other network models.

Table I shows the detection results of different network models. From Table I, we can see that the YOLOv2 model has the slowest detection speed. It might because YOLOv2 contains a large number of convolutional layers, which increases the time of network detection. The detection speeds of YOLOv2-Tiny, YOLOv3-Tiny, and LSDN combined by the convolutional layer and pooling layer are faster than that of YOLOv2. And the detection accuracy of the LSDN model is much higher than that of YOLOv2 and YOLOv2-Tiny. A possible reason is that the feature fusion technology is adopted in LSDN, which can extract the edge features and the texture features of the target well.

2) Results Comparison on FPA-DNN: In order to show the performance of the proposed FPA algorithm, we compare the FPA-DNN model with some state-of-the-art network acceleration models with different pruning algorithms, including maximum pruning algorithm, and convolution kernel pruning algorithm [29].

Table II shows the experimental results on different network acceleration models. From Table II, we can find that for the trained network model, there is no significant loss of detection accuracy after network pruning. Moreover, the network model FPA-DNN which adopts the FPA algorithm has the best results. Even after pruning, compared with the LSDN model, the detection accuracy and Average IoU of the FPA-DNN are improved slightly. The model using the maximum pruning algorithm shows the worst detection accuracy results, which confirms that the convolution kernel with maximum absolute weight value has a greater influence than other convolution kernels. The convolution kernel pruning algorithm performs better than the maximum pruning algorithm, but is a little worse than LSDN. The reason is that the convolution kernel pruning algorithm removes the convolution kernel with fixed proportion for each layer, without considering the weight parameter distribution in the network.

V. Conclusion and Future Work

In this paper,in order to get an effective detection result on ship detection, a new deep neural network named model FPA-DNN has been developed. The FPA-DNN model is a hybrid learning model, in which the deep neural network model LSDN can effectively reduce the number of parameters and improve the detection speed without accuracy loss, and the pruning based forward propagation acceleration algorithm can remove the redundant convolution kernels and further speed up the detection process. The experimental results on 800 optical satellite images in different scenarios show that, compared with several state-of-the-art models, the LSDN model and FPA-DNN model outperform the other deep neural network models on both detection accuracy and detection speed, and the FPA-DNN model shows significant improvement on the detection speed.

In the future, we will continue our study on improving

TABLE I

Detection results of different network models.

Network Models	AP	Precision	Recall	F1-score	AverageIoU	Dspeed(s/pages)
YOLO$v2$[27]	54.40%	0.68	0.65	<u>0.67</u>	43.26%	15.82
YOLO$v2-tiny$[27]	**81.80%**	0.26	**0.91**	0.40	16.54%	16.29
YOLO$v3-tiny$[28]	62.82%	<u>0.72</u>	0.73	0.50	<u>47.73%</u>	<u>12.73</u>
LSDN	<u>78.17%</u>	**0.87**	<u>0.88</u>	**0.88**	**56.71%**	**11.24**

TABLE II

Detection results on different network acceleration models.

Network Models	*Ratio*	Precision	Recall	F1-score	AverageIoU	Dspeed(s/pages)
LSDN	-	**0.87**	<u>0.88</u>	<u>0.88</u>	<u>56.71%</u>	11.24
Maximum pruning	40%	0.68	0.41	0.51	34.44%	8.24
Convolution kernel pruning[28]	40%	0.76	0.70	0.73	44.17%	<u>8.06</u>
FPA-DNN	**42.16%**	**0.87**	**0.89**	**0.89**	**57.50%**	**7.63**

the efficiency of FPA-DNN with more data sources. And we will also try to research on the network compressing method which can further optimize the convolution operation and reduce detection time.

VI. ACKNOWLEDGEMENT

This work is supported by the National Nature Science Foundation of China [Grant No. 61773296], and the National Key Research and Development Plan [Grant No. 2017YFC1502902].

REFERENCES

[1] L. Zhai, Y. Li, and Y. Su, "Inshore ship detection via saliency and context information in high-resolution sar images," *IEEE Geoscience & Remote Sensing Letters*, pp. 1–5, 2016.

[2] C. Zhu, H. Zhou, R. Wang, and J. Guo, "A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 48, no. 9, pp. 3446–3456, 2010.

[3] C. Wang, F. Bi, W. Zhang, and L. Chen, "An intensity-space domain cfar method for ship detection in hr sar images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 4, pp. 529–533, 2017.

[4] G. R., D. J., and D. T., "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014, abs/1311.2524.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–16, 2014.

[6] G. R., "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 06 2015.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–787.

[9] H. Luo, K. Wu, Z. Guo, L. Gu, and L. M. Ni, "Ship detection with wireless sensor networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 23, no. 7, pp. 1336–1343, 2012.

[10] Y. Wang, X. Ning, B. Leng, and H. Fu, "Ship detection based on deep learning," in *ICMA*, 2019, pp. 275–279.

[11] K. Miao, X. Leng, L. Zhao, and K. Ji, "A modified faster r-cnn based on cfar algorithm for sar ship detection," in *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, 2017.

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1800–1807.

[13] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018, pp. 6848–6856.

[14] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, abs/1704.04861.

[15] I. Bello, B. Zoph, V. Vasudevan, and Q. Le, "Neural optimizer search with reinforcement learning," 2016, abs/1611.01578.

[16] Y. He, J. Lin, Z. Liu, H. Wang, L. J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *ECCV*, 2018, pp. 815–832.

[17] A. Abdelhafiz, O. Hammi, A. Kwan, and F. M. Ghannouchi, "Digital predistortion of lte-a power amplifiers using compressed sampling based unstructured pruning of volterra series," *IEEE Transactions on Microwave Theory & Techniques*, vol. 62, no. 11, pp. 2583–2593, 2014.

[18] Y. Lecun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in Neural Information Processing Systems*, vol. 2, pp. 598–605, 2000.

[19] L. I. Shia and L. I. Qiu-Shib, "Research on pruning algorithm of product feature mining in chinese review," *Computer Engineering*, vol. 37, no. 23, pp. 43–45, 2011.

[20] N. Yu, S. Qiu, X. Hu, and J. Li, "Accelerating convolutional neural networks by group-wise 2d-filter pruning," in *IJCNN*, 2017, pp. 2502–2509.

[21] R. Sredojevic, S. Cheng, L. Supic, R. Naous, and V. Stojanovic, "Structured deep neural network pruning via matrix pivoting," *CoRR*, 2017, abs/1712.01084.

[22] W. H., Z. Q., and W. Y., "Structured deep neural network pruning by varying regularization parameters," *CoRR*, 2018, abs/1804.09461.

[23] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *ICCV*, 2017, pp. 2755–2763.

[24] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[25] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, pp. 18–32, 2015.

[26] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017, pp. 1–13.

[27] D. Li, X. Wang, and D. Kong, "Deeprebirth: Accelerating deep neural network execution on mobile devices," in *AAAI*, 2018, pp. 2322–2330.

[28] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless cnns with low-precision weights," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[29] W. Shengsheng, M. Wang, and G. Wang, "Two stage remote sensing image target detection based on deep neural network pruning," *Journal of Northeastern University*, vol. 40, no. 2, pp. 174–179, 2019.