



Article

Absorption Pruning of Deep Neural Network for Object Detection in Remote Sensing Imagery

Jielei Wang ¹, Zongyong Cui ¹, Zhipeng Zang ², Xiangjie Meng ² and Zongjie Cao ^{1,*}

¹ School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

² Beijing Huahang Radio Measurement Research Institute, Beijing 102445, China

* Correspondence: zjcao@uestc.edu.cn

Abstract: In recent years, deep convolutional neural networks (DCNNs) have been widely used for object detection tasks in remote sensing images. However, the over-parametrization problem of DCNNs hinders their application in resource-constrained remote sensing devices. In order to solve this problem, we propose a network pruning method (named absorption pruning) to compress the remote sensing object detection network. Unlike the classical iterative three-stage pruning pipeline used in existing methods, absorption pruning is designed as a four-stage pruning pipeline that only needs to be executed once, which differentiates it from existing methods. Furthermore, the absorption pruning no longer identifies unimportant filters, as in existing pruning methods, but instead selects filters that are easy to learn. In addition, we design a method for pruning ratio adjustment based on the object characteristics in remote sensing images, which can help absorption pruning to better compress deep neural networks for remote sensing image processing. The experimental results on two typical remote sensing data sets—SSDD and RSOD—demonstrate that the absorption pruning method not only can remove 60% of the filter parameters from CenterNet101 harmlessly but also eliminate the over-fitting problem of the pre-trained network.



Citation: Wang, J.; Cui, Z.; Zang, Z.; Meng, X.; Cao, Z. Absorption Pruning of Deep Neural Network for Object Detection in Remote Sensing Imagery. *Remote Sens.* **2022**, *14*, 6245. <https://doi.org/10.3390/rs14246245>

Academic Editor: Konstantinos Topouzelis

Received: 21 October 2022

Accepted: 6 December 2022

Published: 9 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: remote sensing imagery; object detection; network pruning; object characteristics; deep convolutional neural network (DCNN)

1. Introduction

Due to the advantages of remote sensing imagery (e.g., all-day and all-weather), remote sensing technology has been widely used for meteorological observation, disaster monitoring, navigation safety monitoring, and other fields [1,2]. In recent years, open access to remote sensing imagery has become more and more convenient [3], and these massively available remote sensing images directly promote the application of DCNN in remote sensing imagery processing [4–6]. For object detection in remote sensing imagery, researchers have used DCNNs with more and more parameters in order to achieve a better detection performance [7–9]. One of the main problems with DCNNs is that they are often over-parametrized, which means that their inference and updating require a large amount of computing, storage, power, and communication bandwidth resources [10]; however, spaceborne or airborne remote sensing imaging systems are generally limited in communication bandwidth, computing, power, storage resources, and so on. Thus, applying deep learning techniques on these platforms leads to problems associated with insufficient resources when performing multiple tasks simultaneously. In addition, spaceborne or airborne platforms usually do not have high-speed Internet, and transmitting many remote sensing images to ground servers for processing will significantly consume valuable communication bandwidth. Therefore, the amount of parameters in DCNNs has become a bottleneck for object detection on resource-constrained platforms [11–16].

Series networks, such as Faster-RCNN [7], YOLO [8], SSD [9], and CenterNet [17], have achieved excellent detection performance in remote sensing object detection tasks in recent

years. However, these general deep detection neural networks are not specially designed for processing remote sensing data. Most researchers have taken object detection tasks in common high-resolution optical images as network performance evaluation benchmarks. Due to limitations in the imaging band, signal-to-noise ratio, and other factors, remote sensing images have lower spatial resolution and greater noise impact than general high-resolution optical images. Therefore, the available feature information of objects in remote sensing imagery is lower, and the network complexity required for object detection in remote sensing images should be much lower than that in traditional optical images; in other words, the object detection networks designed based on traditional optical data sets are too complex for the object detection tasks in remote sensing imagery. Applying such complex networks for remote sensing object detection not only wastes a lot of resources but also poses a serious risk of over-fitting.

Fortunately, researchers have found that the parameters of most DCNNs are significantly redundant and, in the best case, the authors predicted the remaining 95% with only 5% of the weights [18]. This phenomenon demonstrates that it is feasible to compress remote sensing object detection neural networks. Researchers have recently explored knowledge distillation [19], parameter quantization [20], low-rank decomposition, and network pruning to achieve model parameter compression [21]. Among such approaches, network pruning has been widely studied by researchers. By removing the redundant parameters in the network, we can not only significantly reduce the resource consumption in remote sensing object detection but can also eliminate the over-fitting problem in the network.

To obtain lightweight remote sensing object detection neural networks, we propose a network pruning method that takes into account the characteristics of remote sensing objects, which we call absorption pruning. Our previous work has shown that, as an anchor-free end-to-end differentiable object detection network, CenterNet is more effective in remote sensing object detection tasks than other detection networks [17]. Therefore, we take CenterNet as the pre-trained network for the absorption pruning method. The absorption pruning method is designed as a four-stage pruning method, as shown in Figure 1b. First, the absorption pruning method selects filters that are easily absorbed from the pre-trained network, according to the filter selection criteria proposed in this paper. Second, the knowledge absorption training algorithm designed in this paper smoothly transfers the “knowledge” of the selected filters to the rest of the network. Third, the selected filters are removed from the network. Fourth, fine-tuning is conducted to recover the network performance. Finally, the absorption pruning method outputs a light object detection network. In this process, according to the characteristics of remote sensing objects and the characteristics of the deep neural network feature extractor, we design a pruning ratio adjustment method to allow the absorption pruning method to better adapt to the remote sensing object detection task.

The contributions of this paper can be summarized as follows:

- We propose a new pruning pipeline that does not require the iterative mini-batch pruning used in the classic pruning pipeline.
- The absorption pruning method provides a knowledge absorption training method, which can smoothly transfer the “knowledge” in some filters to the remaining filters.
- A filter selection criterion that can identify filters that are easily absorbed is proposed.
- We design a pruning ratio adjustment method based on the remote sensing object characteristics, such that absorption pruning can better facilitate remote sensing object detection network compression. To the best of our knowledge, this is the first pruning ratio adjustment method designed for remote sensing object detection network pruning, according to remote sensing object characteristics.

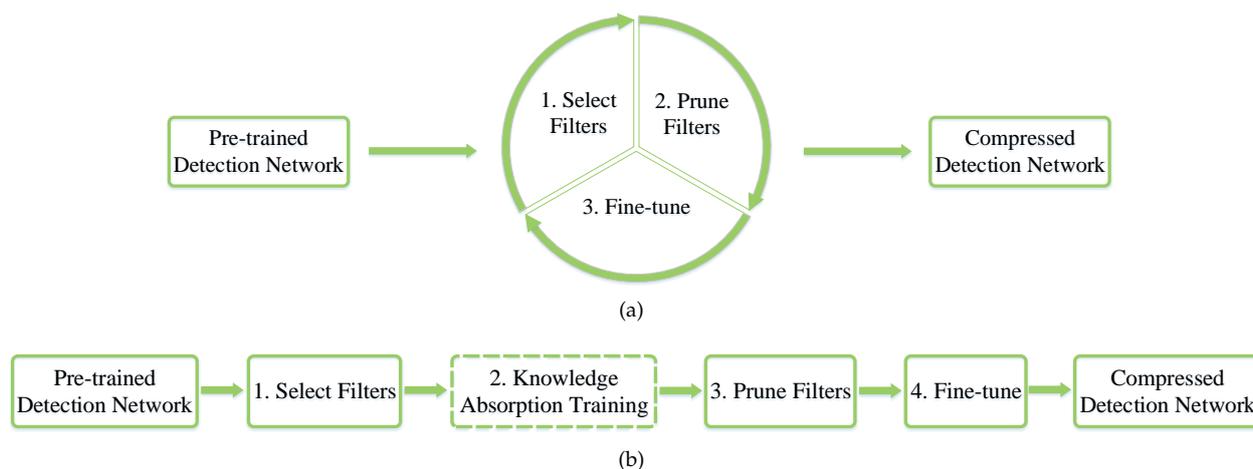


Figure 1. Comparison between the pipeline of the absorption pruning algorithm proposed in this paper and the pipeline of the traditional pruning algorithm: (a) Pipeline of the classic pruning method, and (b) pipeline of the absorption pruning method.

2. Related Work

As remote sensing imagery has become easier to obtain, many researchers have tried to use remote sensing data to drive intelligent applications based on DCNNs [3]. However, the over-parametrization problem of DCNNs hinders their application on resource-constrained remote sensing platforms [12–14]. Moreover, the work in the paper [22] has shown that the redundant parameters of DCNNs are necessary during network training. To reduce the hardware resources required to run deep neural networks, researchers have used network pruning to remove redundant parameters from the pre-trained network, thus reducing the risk of over-fitting. This section introduces the work related to this paper from two aspects: remote sensing object detection and network pruning.

2.1. Remote Sensing Object Detection

In recent years, detectors based on deep convolutional neural networks have achieved great success in object detection in remote sensing imagery. These deep detection networks have achieved significant advantages over traditional detectors. Therefore, as more and more excellent deep detection networks are proposed, researchers have used these detection networks for object detection tasks in remote sensing imagery. For example, Zhang et al. used R-CNN to solve the task of ship detection in high-resolution remote sensing images [23]. In [24], the authors used a rotating cascade R-CNN to detect objects in optical remote sensing images. With the continuous development of R-CNN series networks, the papers [25–27] used the more advanced Faster R-CNN and Mask R-CNN to solve the ship detection problem in remote sensing imagery and achieved good performance.

Next, single-stage detection networks, which are simpler and faster than the two-stage networks, began to shine in the field of remote sensing object detection. In [28], the authors successfully used the improved Retinanet for ship detection in optical remote sensing imagery. Liu et al. proposed an improved SSD network to detect small objects with complex backgrounds [29]. The papers [30–32] have further explored the application of YOLO series networks for object detection in remote sensing maps. In [33], the authors improved CenterNet to obtain a better small object detection capability in remote sensing imagery.

Most of the existing remote sensing object detection networks have been improved, based on the original detection network in order to achieve better performance. However, during the design of these original detection networks, they were verified on traditional high-resolution optical data sets, such as PASCAL VOC [34] and COCO [35]. Due to the difference in imaging distance and principle, remote sensing images contain fewer features than traditional high-resolution optical images. Therefore, detection and recognition tasks in remote sensing images require less network complexity. Using deep neural networks

designed on high-resolution optical data sets for remote sensing object detection leads to the problems of wasting hardware resources and over-fitting. Pruning redundant parameters from these networks can effectively solve these problems.

2.2. Network Pruning

Network pruning has been proven to be effective in reducing network complexity while preserving network performance [11,12,36]. Moreover, under an appropriate pruning ratio, network pruning can effectively eliminate the over-fitting problem of the neural network, thus improving network performance. Therefore, in recent years, more and more researchers have devoted themselves to studying network pruning methods for model parameter compression. Han et al. have proposed that parameters below a certain threshold should be selected and pruned to achieve network compression [37]. They used a classic iterative three-stage pipeline to achieve this process (i.e., select parameters, prune parameters, and fine-tune the pruned network). Since then, this classic pruning pipeline has become the basis of network pruning research. Most subsequent studies have focused on proposing more accurate parameter importance evaluation criteria. For instance, Li et al. proposed to use the sum of the absolute values of all parameters in the filter as the importance score, pruning filters with lower scores [14]. Luo et al. have considered that filters with the same output for different inputs are useless [38]. They described this feature by computing the entropy of a filter's mean output sequence for a large number of input samples. Wang et al. input training samples into a DCNN and then calculated the average information entropy of all feature maps and, finally, pruned filters with lower average information entropy [39]. The disadvantage of the pruning method based on the classic three-stage pruning pipeline is that it requires iterative pruning and may remove filters that still contain a lot of "knowledge".

He et al. also noted that it is unreasonable for the classical pruning pipeline to directly prune the selected filters [40]. They proposed a soft pruning method that uses an initialized network as input: instead of pruning the filters directly after selecting them, they set the filter parameters to zero, allowing them to be re-activated in subsequent training. The network will be pruned using some criteria until the end of training. In [21], Zhang et al. optimized the soft pruning process by initializing the parameters of selected filters to non-zero values that are more easily re-activated. However, these soft pruning methods still destroy the "knowledge" in these filters by resetting the parameters in the selected filters.

From the perspective of remote sensing object detection network pruning, Wang et al. designed a method for obtaining a lightweight two-stage detection network using triple pruning, which was proven to effectively reduce false alarms and achieve network acceleration [41]. However, there are still few pruning algorithms for remote sensing image object detection networks; therefore, it is crucial to carry out research in this area to enrich the pruning methods for remote sensing object detection algorithms under different task scenarios.

3. Methods

A unique pruning method for remote sensing object detection network compression is proposed in this paper. In Section 3.1, we first introduce the overall framework of the proposed method in detail, focusing on the difference in pipeline compared to existing pruning methods. Then, the unique filter selection criterion and absorption pruning training approach are introduced in Sections 3.2 and 3.3, respectively. Furthermore, a method for pruning ratio adjustment that considers object features in remote sensing imagery is proposed in Section 3.4. Finally, we introduce the pruning strategy of pruning CenterNet in Section 3.5.

3.1. Overall Framework

Most traditional network pruning methods adopt a classical three-stage iterative pruning pipeline, as shown in Figure 1a. Specifically, these methods first select unimportant

parameters from the network, according to the designed criteria. Then, these selected parameters are removed from the network. Finally, fine-tuning is conducted, in order to recover the performance damage caused by removing parameters. To avoid removing too many parameters at a time, which may cause irreversible damage to network performance, researchers typically choose to prune a small batch of parameters from the network each time, removing redundant network parameters by repeating the above three-stage operation many times. However, there are two significant unreasonable aspects in the classic pipeline: first, the iterative selection–pruning–fine-tuning pipeline consumes a lot of time and computing resources; second, the classic process simply removes parameters that still contain a lot of “knowledge” and then hopes to re-learn that knowledge through fine-tuning.

Therefore, the filter pruning pipeline shown in Figure 1b is proposed in this paper in order to avoid the above two problems. The pipeline proposed in this paper also starts with a pre-trained network, but abandons the traditional three-stage iterative operation and replaces it with a four-stage operation that only needs to be performed once: filter selection, knowledge absorption training, pruning, and fine-tuning.

Different from the classical pipeline, we no longer directly remove a selected filter but design a knowledge absorption training method to smoothly transfer the “knowledge” in the selected filter to the rest of the network and then prune these filters. This process of transferring “knowledge” allows the “knowledge” to be absorbed by the rest of the network. Therefore, we call this method the knowledge absorption training method, which is also the origin of the word “absorption” in this paper. Ideally, pruning performed after the knowledge in the filter has been fully absorbed will not cause any damage to model performance; however, in practice, due to the limitations of training techniques, the knowledge in the selected filter cannot be completely absorbed, and the pruned filters still contain a small amount of knowledge. Therefore, fine-tuning is required to recover the performance damage.

3.2. Filter Selection Criteria

Classic pruning methods select filters and prune them directly; therefore, unimportant filters should be selected to avoid a significant drop in network performance. However, the proposed absorption pruning method requires the rest of the network to absorb the knowledge in the selected filters, and so we need to choose easily absorbed filters. In other words, the knowledge contained in the selected filters should be easier to learn.

Assume that the parameter tensor of the j th filter of the i th layer of the network is $P_{i,j} \in c \times o \times w \times h$, where c , o , w , and h represent the number of input channels, the number of output channels, the width of the convolution kernels, and the height of the convolution kernels, respectively. In the process of training a complex network as a pre-trained network, the distance between the filter parameter tensor $P_{i,j}$ and its final converged parameter tensor $P_{i,j}^*$ can be defined as follows:

$$D_{i,j} = \left\| P_{i,j} - P_{i,j}^* \right\|_1, \quad (1)$$

where $\|\cdot\|_1$ represents the L1-norm. To measure the learning difficulty of different filter parameters, we monitor the parameter change during the training of the CenterNet network for the object detection task. Figure 2 shows the changes in the parameters of 10 randomly selected filters from CenterNet. The horizontal axis represents the training epoch, while the vertical axis represents $D_{i,j}$. Specifically, in the training process of the pre-trained network, we randomly select 10 filters as observation objects and record the parameters of these filters in different epochs. After the pre-trained network is trained to convergence, the $D_{i,j}$ values of the 10 filters at different epochs are calculated, according to Equation (1), and the final $D_{i,j}$ value variation curves with epochs are plotted in Figure 2.

Figure 2 contains two essential pieces of information: first, all filter parameters were initialized very well. The initialized filter parameters were very close to the optimized

parameters (i.e., the $D_{i,j}$ values of all filters are close to 0). The second is that the direction of the minimum gradient of the filter parameters during the training process was not consistent with the direction of the shortest distance (the lower curve is closer to the shortest distance than the upper curve). Therefore, we realized that this phenomenon can be used to measure the learning difficulty of filter parameters in order to construct a simple and unique filter selection criterion. Specifically, only the model at a certain stage and the final model need to be stored. Then, the value of $D_{i,j}$ for each filter is obtained according to Equation (1). A larger $D_{i,j}$ indicates that the filter $P_{i,j}$ is more challenging to learn. In addition, it can be seen, from Figure 2, that when the epoch is within the interval (5, 70), the convergence distance curves of different filters have obvious differences and rarely cross. Thus, when training the pre-trained network, we only save the $epoch = 10$ and the final converged model parameters to evaluate the filter absorption difficulty and select the filter from the converged pre-trained network according to the absorption difficulty score.

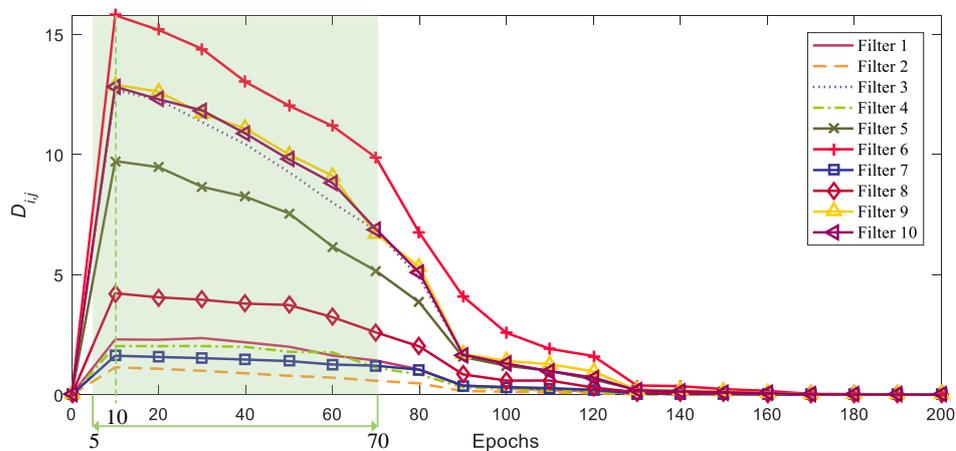


Figure 2. The change curves of filter parameters during network training. The fluctuation of the curves reflects the difficulty of learning different filter parameters.

3.3. Knowledge Absorption Training

For object detection tasks, the goal is to accurately predict the location and size of the object bounding box. Taking CenterNet as an example, the training loss function consists of three parts [42]:

$$Loss = loss_{hm} + loss_{wh} + loss_{off}, \tag{2}$$

where $loss_{hm}$ represents the prediction loss of the object centre, $loss_{wh}$ represents the prediction loss of the width and height of the object bounding box, and $loss_{off}$ represents the bias loss of the predicted object center.

Knowledge absorption training aims to transfer the knowledge from the selected filters to the rest of the network such that the selected filters can be removed harmlessly. In the absorption pruning algorithm proposed in this paper, we design an absorption loss $loss_{ab}$ to absorb the knowledge in the selected filters. Therefore, the new loss function in the absorption training is:

$$Loss^* = loss_{hm} + loss_{wh} + loss_{off} + \lambda loss_{ab}, \tag{3}$$

where λ is the weight of $loss_{ab}$.

Note that $loss_{ab}$ is defined to disable the selected filters gradually, while $loss_{hm}$, $loss_{wh}$, and $loss_{off}$ jointly ensure that the detection performance of the network does not decline during the process of the selected filters gradually losing their effect.

It should be noted that, in most existing convolutional neural networks, the convolutional layer is followed by a batch normalization layer to address the internal covariate shift problem [43]:

$$y_i = \gamma_i \frac{x_i - \mu_i}{\sqrt{\delta_i^2 + \varepsilon}} + \beta_i, \quad (4)$$

where x_i represents the output of the i th convolutional layer; y_i represents the output of the normalization layer after the i th convolutional layer; μ and δ^2 represent the expectation and variance of x_i calculated over the training set, respectively; and r_i and β_i are the scaling and offset parameters of the normalized x_i , respectively. The pair r_i and β_i serve as a gate for the output of a filter in the i th convolutional layer: if both $r_{i,j}$ and $\beta_{i,j}$ are 0, then the filter parameter $P_{i,j}$ will be entirely useless in the network.

Therefore, a new absorption loss is designed for gradually disabling selected filters by penalizing specific r_i and β_i combinations:

$$loss_{ab} = \sum_{i=1}^n mask_i (\|\gamma_i\|_1 + \|\beta_i\|_1), \quad (5)$$

where $mask_i$ is a 0,1 sequence, representing the index of the selected filter of the i th convolutional layer. An element 1 in $mask_i$ indicates that the corresponding filter is selected, while an element 0 indicates that the corresponding filter is not selected. It is worth mentioning that the work in [44] used a similar loss function design to regularize the parameter γ and proved its effectiveness. As such, the design of the loss function $loss_{ab}$ used in this paper was inspired by the article [44].

3.4. A Method for Pruning Ratio Adjustment Based on Remote Sensing Image Features

Compared with traditional optical images, the objects in remote sensing images have two remarkable characteristics: first, the size of the objects is tiny. This is because remote sensing imagery is usually imaged very far away from the object, and so the spatial resolution of the object is low. Second, the object features are typically very simple. For optical remote sensing imagery, this is mainly due to the long imaging distance. For SAR remote sensing imagery, in addition to the long imaging distance, it is also mainly affected by factors such as the low frequency of electromagnetic waves (compared to visible light) used by SAR imaging technology, the lack of rich color information, and strong background noise. Therefore, the objects in SAR images are only composed of a set of scattered points with different brightnesses and positions, and the provided object features are relatively simple.

In a related manner, DCNN also has two key characteristics. One is that higher convolutional layers have larger receptive fields, and the other is that the features extracted by higher convolutional layers are also more complex. Therefore, two inferences are easily drawn when DCNNs are used for object detection on remote sensing imagery. First, as the object size in the remote sensing image is small, a large receptive field is not required. Second, the features of remote-sensing objects are simple, and so the features extracted by the lower convolutional layers are more important. Based on such prior knowledge, we designed a method according to each layer's receptive field in order to adjust the absorption pruning ratio flexibly.

Traditional network pruning methods usually adopt the same pruning ratio η in each convolution layer. Considering the characteristics of the remote sensing object and DCNN analyzed above, the absorption pruning method used for remote sensing object detection networks should remove more filters with large receptive fields and retain more filters with small receptive fields. Equations (6) and (7) detail the calculation methods for the receptive field:

$$R_i = R_{i-1} + \left[(k_i - 1) * \prod_{i=1}^{i-1} s_i \right], \quad (6)$$

$$\text{sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}, \tag{7}$$

where R_i represents the receptive field of the i th convolutional layer, k_i is the convolution kernel size (or pooling size) of the i th convolutional layer, and s_i is the convolution stride.

We define the adjustment factor of the pruning ratio as α and define the minimum receptive field necessary for extracting remote sensing object features as z (i.e., the threshold to judge whether the receptive field is too large or too small). Then, the adjusted pruning ratio η_i^* for the i th convolution layer can be calculated, according to Equation (8).

$$\eta_i^* = \eta + \alpha \times \text{sign}(R_i - z) * \frac{R_i}{\sum_{i=1}^N R_i}. \tag{8}$$

Figure 3 illustrates the adjustment process for the pruning ratio. As shown in Figure 3, the adjustment of the pruning ratio is similar to a “lever”, where the parameter z controls the “fulcrum” of the lever, and the parameter α controls the adjustment range.

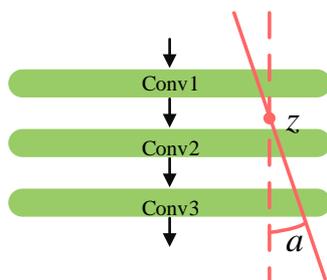


Figure 3. Illustration of the pruning ratio adjustment method based on the characteristics of remote sensing objects.

3.5. Pruning Strategy for CenterNet

CenterNet101 uses the feature extractor of ResNet101 as the backbone of the network. ResNet uses an innovative residual structure to fuse the output features of different network layers. This innovation facilitates the excellent performance of the ResNet structure network, but the complex network structure also brings challenges for network pruning.

The ResNet structure network usually consists of multiple convolution blocks, as shown in Figure 4. In order to avoid the trouble of pruning filters in down-sampling layers, traditional pruning algorithms usually only prune the convolution layers shown in green [21]. The absorption pruning algorithm proposed in this paper also includes the red convolutional layers in the pruning range. The specific approach is as follows: treat the red convolutional layers in each convolutional block as a convolutional layer, and these layers share the same sequence of filter learning difficulty. In the filter selection stage, the learning difficulty score of each filter in the red layer is calculated as follows:

$$D_i = \{ \text{Max}(D_{i,1}), \text{Max}(D_{i,2}), \text{Max}(D_{i,3}), \dots, \text{Max}(D_{i,o}) \}, i = 1, 2, \dots, n, \tag{9}$$

where o represents the number of filters in the i th convolutional layer, and n represents the number of coupled convolutional layers (i.e., red convolutional layers in Figure 4) in each convolutional block. Therefore, the filters in all red convolutional layers in each convolutional block have the same sequence of filter importance scores, and the filters with the same index in these layers will be absorbed, pruned, or retained simultaneously. The absorption pruning algorithm can also implement pruning for all layers of the CenterNet101 backbone.

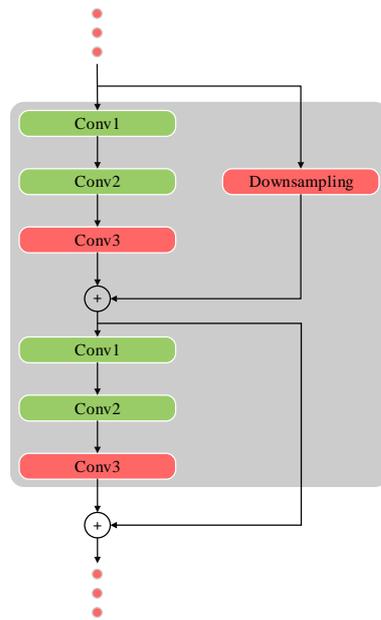


Figure 4. Illustration of the pruning strategy for CenterNet101.

The algorithm proposed in this paper is summarized as Algorithm 1:

Algorithm 1 Detailed process of the absorption pruning method.

Input: Pre-trained network with parameter $P_{i,j}^*$, network with parameter $P_{i,j}$ when epoch = 10, training set T , pruning ratio x , absorption training epochs E

Output: Lightweight network

1: Calculate the learning difficulty score of all filters in the pre-trained network:

$$D_{i,j} = \|P_{i,j} - P_{i,j}^*\|_1$$

2: Sort all training difficulty scores of the i th convolutional layer in ascending order to obtain the score set D_i^*

3: Calculate the pruning ratio η_i^* of the i th convolution layer according to Equation (8)

4: Obtain the filter score threshold $D_{i,\theta}^*$, where $\theta = \lceil \eta_i^* \text{card}(D_i^*) \rceil$

5: Select all filters with scores lower than the threshold $D_{i,\theta}^*$ to obtain the masks of each layer,

$$\text{mask}_{i,j} = \begin{cases} 1, & D_{i,j} < D_{i,\theta}^* \\ 0, & D_{i,j} \geq D_{i,\theta}^* \end{cases}$$

6: **for** epoch = 1; epoch < E ; epoch ++ **do**

7: $\text{Loss}^* \leftarrow \text{forward}(P_{i,j}^*, T)$ according to Equation (3).

8: $\text{Backward}(\text{Loss}^*, P_{i,j}^*)$

9: Update $P_{i,j}^*$

10: Prune all selected filters.

11: Fine-tune to restore network performance

12: **return** $P_{i,j}^*$

4. Experiments and Discussions

Experiments were carried out to verify the effectiveness of the proposed absorption pruning method. The detailed experimental settings are introduced in Section 4.1. Section 4.2 details the performance of the pre-trained network and the pruning results of the absorption pruning method at different pruning ratios. Sections 4.3–4.5 describe the effectiveness verification experiments for different functional modules of the absorption pruning method. In Section 4.6, the absorption pruning method is compared with other state-of-the-art network pruning methods in order to further verify its effectiveness. Hy-

perparameter selection experiments for the pruning ratio adjustment method are detailed in Section 4.7. In addition, partial object detection visualization results of the pruned lightweight network are shown and analyzed in Section 4.8.

4.1. Implementation Details

4.1.1. Data Sets

According to the imaging band, visible light remote sensing and SAR remote sensing are the most important means of earth observation for spaceborne or airborne platforms. In order to verify the effectiveness of the proposed absorption pruning algorithm on the object detection network in remote sensing imagery, we conducted experiments on two typical remote sensing object detection data sets: the SAR image data set SSDD [25] and the visible light remote sensing image data set RSOD [45].

The SSDD data set contains 1160 images and 2456 ships, with an average of 2.12 ships per image. The data are mainly from slices of RadarSar-2, TerraSAR-X, and Sentinel-1. The spatial resolution of the images is 1–15 m, and the average size of the images is 500×500 pixels. Figure 5 (Right) shows an image from the SSDD data set. Furthermore, in the experiments in this paper, the samples in the SSDD data set were randomly divided into training, test, and validation sets in a ratio of 8:1:1.

The RSOD data set was released by Wuhan University in 2015. It contains four types of objects: oil tank, aircraft, overpass, and playground. Researchers often use it as a validation data set for object detection algorithms based on optical remote sensing imagery. Figure 5 (Left) shows an image from the RSOD data set. We mixed the four classes of samples in the RSOD data set and randomly divided them into training, test, and validation sets in a ratio of 8:1:1.



Figure 5. Typical remote sensing images. (Left): Optical remote sensing image with airport objects; (Right): SAR image with ship objects.

4.1.2. Pre-Training

CenterNet101 was trained from scratch on the SSDD and RSOD data sets to obtain the pre-trained networks. We used the Adam optimizer during training. In addition, the maximum number of training epochs was set to 200, and the initial learning rate was set to 1.25×10^{-4} and decayed by a factor of 0.1 at the 90th, 130th, and 170th epochs.

4.1.3. Absorption Training and Fine-Tuning

For absorption training, the Adam optimizer was still adopted. The total number of training epochs was set to 60, and the initial learning rate was set to 1.25×10^{-4} and decayed by 0.1 at the 30th epoch. In the fine-tuning stage, we also used the Adam optimizer and set the total number of training epochs to 60. It should be emphasized that fine-tuning usually adopts a lower learning rate. Therefore, the initial learning rate was set to 1.25×10^{-6} , and was also decayed by 0.1 at the 30th epoch. It is worth noting that the total number of training epochs was set to 60 based on our observation of experimental phenomena. In other words, the selection of these hyperparameters was based on empirical evidence.

4.1.4. Other Experimental Parameter Settings

In all experiments, the sample batch size for training and testing was set to 8. The CenterNet101 network was used as the pre-trained network for all experiments. Considering the average size of images in the SSDD data set, the hyperparameter z of the pruning ratio adjustment method employed in the experiments shown in Sections 4.4 and 4.6 was set to 500, and the pruning ratio adjustment method hyperparameter α was set to 1. The weight parameter λ of $Loss_{ab}$ in Equation (3) was set to 1×10^3 for all experiments. In addition, the detection accuracy mAP used in this paper refers to mAP50.

4.2. Pre-Trained Networks and Pruning Results

According to the training settings in Section 4.1.2, we trained CenterNet101 on the SSDD and RSOD data sets to obtain the pre-trained networks. The pre-trained CenterNet101 networks achieved an mAP of 91.08% and 88.29% on the SSDD and RSOD data sets, respectively.

We input these networks into the absorption pruning method for pruning experiments with different pruning ratios. The pruning results are detailed in Table 1. From the experimental results on the SSDD data set, when the filter pruning ratio was not more than 60%, the detection performance of the pruned CenterNet101 was significantly better than that of the pre-trained network. This was because pruning alleviated the over-parametrization problem of CenterNet101, which reduced over-fitting. In addition, even when 85% of filters were removed from the pre-trained network, the object detection accuracy of the pruned network was only 0.58% lower than that of the pre-trained network. However, the pre-trained network parameters and storage space requirements are ten times larger than those of pruned networks. These results demonstrate that the absorption pruning method can effectively compress the DCNNs in SAR ship detection.

From the experimental results on the RSOD data set, it was found that when the pruning ratio did not exceed 60%, the pruned model networks also obtained better object detection performance than the pre-trained network. When the pruning ratio was increased to 85%, the network performance significantly declined after pruning. This phenomenon indicates that, compared with the SAR object detection task on the SSDD data set, the task on the RSOD data set—containing four types of optical objects—was more complicated and required higher network complexity. The overall results demonstrated that the absorption pruning algorithm proposed in this paper can effectively compress DCNNs for optical remote sensing object detection.

Table 1. Overall pruning results of the absorption pruning method on remote sensing data sets.

Data Set	Pruned Filters (%)	Parameters (M)	Required Storage Space (MB)	mAP(%)
SSDD	0	53.58	214.3	91.08
	10	43.95	175.8	93.51
	35	25.57	102.3	92.28
	60	12.45	49.8	92.13
	85	4.6	18.4	90.50
RSOD	0	53.58	214.3	88.29
	10	43.95	175.8	91.54
	35	25.57	102.3	92.70
	60	12.45	49.8	89.43
	85	4.6	18.4	73.50

4.3. Absorption Training Validation Experiment

To verify the effectiveness of the absorption training approach, we used the CenterNet101 trained on the SSDD data set as the target and selected 10% of the filters, according to the proposed filter selection criterion, for knowledge absorption training. In order to

exclude the influence of other factors, the pruning ratio adjustment and fine-tuning were not enabled in this experiment.

Figure 6 shows the change curves of the loss function during absorption training. According to the analysis of Equation (3) in Section 3.3, the $loss_{hm}$, $loss_{wh}$, and $loss_{off}$ are indicators of network detection performance, while $loss_{ab}$ indicates how much the selected filters contribute to the network. As shown in Figure 6, during the absorption training process, $loss_{ab}$ decreased rapidly while $loss_{hm}$, $loss_{wh}$, and $loss_{off}$ only fluctuated within a small range. These results indicate that the selected filters rapidly lost their effect while the network performance was almost unchanged. It appears that the rest of the network absorbed the knowledge in the selected filters. It is worth mentioning that, in order to clearly show the changes in the four losses in the same figure, $loss_{ab}$ was multiplied by a factor of 1×10^{-3} .

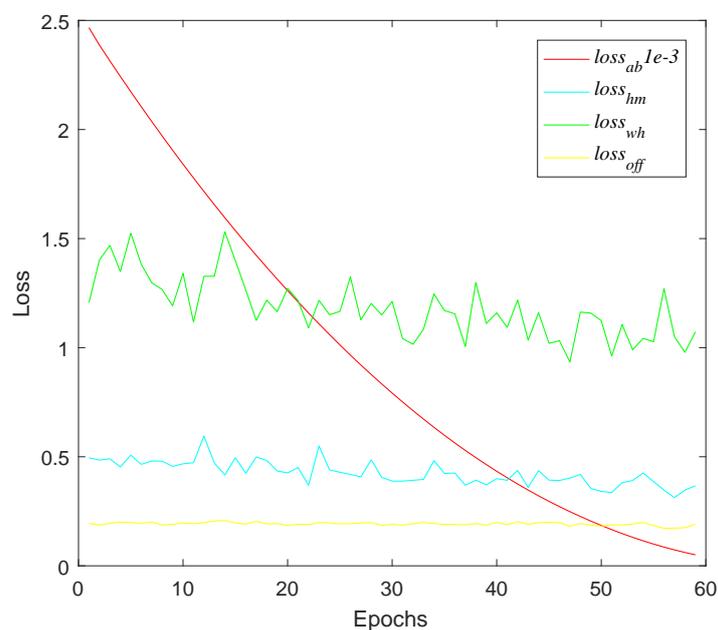


Figure 6. Changes in training loss during absorption training on SSDD.

In addition, Table 2 provides a comparison of the pruned network detection accuracy with and without absorption training under different filter pruning rates. It can be seen that the pruned networks after absorption training could retain most of the detection accuracy after pruning, while the pruned network without absorption training only retained part of the detection performance in small-scale pruning; however, when the pruning ratio was higher than 35%, the pruned network almost completely lost the ability to detect objects.

Table 2. Comparison of pruned network detection accuracy with and without absorption training under different filter pruning ratios (without fine-tuning).

Pruning Ratio	10%	35%	60%	85%
With absorption training	90.53	88.64	85.20	41.50
Without absorption training	81.08	1.57×10^{-6}	0	0

These results demonstrate that knowledge absorption training can effectively transfer the “knowledge” in the selected filter(s) to the rest of the network before pruning, thus effectively avoiding the performance damage caused by filter pruning.

4.4. Filter Selection Criteria Validation Experiment

To verify the effectiveness of the proposed filter selection criterion, we used the CenterNet101 trained on the SSDD data set as the target, and selected 10% of the filters in ascending, descending, or random order of learning difficulty for knowledge absorption training. In order to exclude the influence of other factors, the pruning ratio adjustment method and fine-tuning were not enabled in this experiment.

Absorption training on randomly selected 10% of filters was used as a benchmark. Theoretically, absorption training on the 10% filters with the lowest learning difficulty should provide the highest absorption efficiency (i.e., the contribution index $loss_{ab}$ of the selected filter decreases rapidly under the premise that the object detection performance indicators $loss_{hm}$, $loss_{wh}$, and $loss_{off}$ of the network remain stable). Meanwhile, absorption training on the 10% most difficult-to-learn filters is expected to lead to the lowest efficiency. The experimental results presented in Figure 7 show that the ordering of filter learning difficulty conformed to our theoretical expectations, proving that the filter selection criterion proposed in this paper can effectively score the learning difficulty of filter parameters. In addition, it can be seen, from Figure 7, that the $loss_{hm}$, $loss_{wh}$, and $loss_{off}$ values representing the network detection performance also showed a downward trend as the selected filters gradually lost their function during absorption training. This phenomenon indicates that the over-fitting problem of the network is gradually alleviated. Among them, the network (Figure 7a) that conducts knowledge absorption training in ascending order of filter parameter learning difficulty obtained the highest loss reduction, the network (Figure 7b) that conducted knowledge absorption training in random order of the difficulty of filter parameter learning was in second place, and the network (Figure 7c) that conducted knowledge absorption training in descending order of filter parameter learning difficulty obtained the worst loss reduction. This phenomenon also demonstrates that the filter selection criteria proposed in this paper can, indeed, select the most easily absorbed filter.

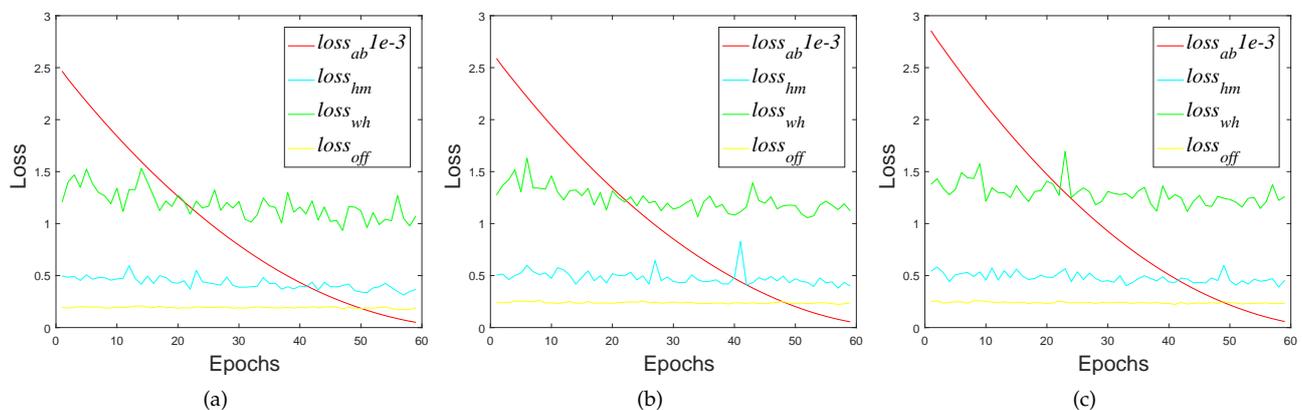


Figure 7. The absorbing training results when selecting filters in ascending, descending, and random order of learning difficulty: (a) Ascending order; (b) Random order; and (c) Descending order.

In order to further verify the effectiveness of the filter selection criteria proposed in this paper, the filters of the pre-trained network were pruned in the same ratios according to the increasing, random, and descending order of parameter learning difficulty, respectively. Except for the filter selection method, the rest of the experimental settings were kept the same, and the experiment was carried out on the SSDD data set. After selecting the filters, the pre-trained network went through the process of absorption, pruning, and fine-tuning. The accuracies of the obtained lightweight detection networks are given in Table 3.

As can be seen from Table 3, under the condition of the same filter selection ratio, selecting the easiest-to-learn filters can provide the best lightweight network while selecting

the most difficult-to-learn filters led to the worst network. These results demonstrate that the filter learning difficulty evaluation criterion proposed in this paper is reasonable.

Table 3. Verification experiment results regarding the validity of the filter selection criteria.

Order	Pruning Ratio (%)			
	10	35	60	85
Ascending	93.51	92.28	92.13	90.50
Random	93.29	91.76	90.55	88.98
Descending	92.14	90.81	87.28	85.36

4.5. Pruning Ratio Adjustment Method Validation Experiment

Experiments were also carried out to verify the effectiveness of the proposed pruning ratio adjustment method. The CenterNet101 trained on the SSDD data set was again used as the test object. The pre-trained CenterNet101 underwent absorption training, pruning, and fine-tuning with and without the pruning ratio adjustment method, and finally, lightweight compressed networks were obtained. Figure 8 depicts the performance of the compressed networks with and without the pruning ratio adjustment method at different pruning ratios. It can be seen that the pruning ratio adjustment method specially designed for remote sensing image object detection enhanced the absorption pruning method. By analyzing the number of filters in each layer of the compressed networks obtained with and without absorption pruning, we observed that the method for the pruning ratio adjustment proposed in this paper can, indeed, help absorption pruning to focus on pruning top convolutional layers and reduce the pruning of bottom convolutional layers.

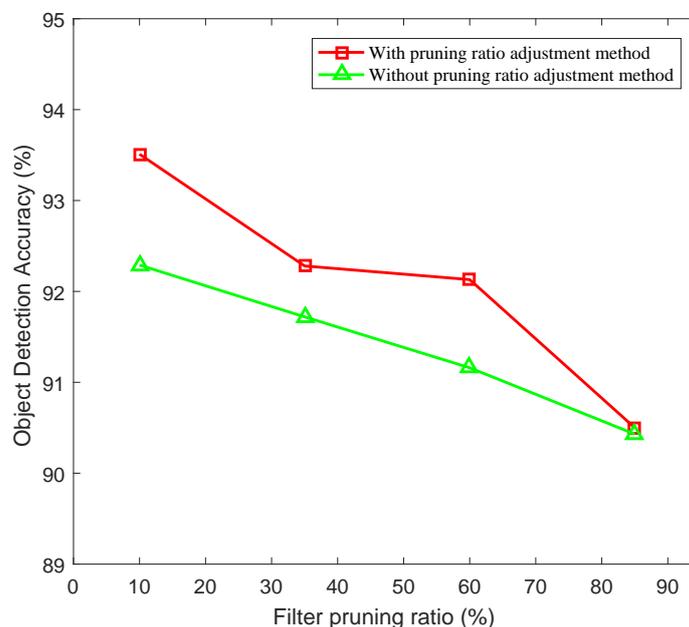


Figure 8. Comparison of CenterNet101 in the absorption pruning with and without the pruning ratio adjustment method.

4.6. Comparison with Other Pruning Methods

To further verify the effectiveness of the absorption pruning method proposed in this paper, four network pruning methods were used to prune CenterNet101 for remote sensing object detection: the classic pruning method L1 [14], APoZ [15], Taylor [16], and the state-of-the-art (SOTA) pruning method HRank [12]. Under the same pruning rate, L1, APoZ, Taylor, and HRank adopt the classic pruning pipeline to prune filters in a layer-by-layer

manner. In this process, L1, APoZ, Taylor, and HRank uniformly pruned eight times and fine-tuned for 30 epochs after each pruning to restore network performance.

The detailed experimental results are shown in Table 4. It can be seen that the performance of the absorption pruning method proposed in this paper was better than that of the SOTA method and significantly better than those of the classic pruning methods. Moreover, as L1, APoZ, Taylor, and HRank were designed for pruning optical object classification networks, they perform worse than expected in remote sensing object detection network compression tasks. These results demonstrate that the absorption pruning method proposed in this paper is more suitable for the network compression task of remote sensing object detection and can achieve SOTA performance.

Table 4. Results for experimental comparison of absorption pruning with other pruning methods.

Data Set	Method	Pruning Ratio (%)			
		10	35	60	85
SSDD	L1 [14]	90.10	88.12	85.46	81.82
	APoZ [15]	89.44	85.31	76.74	60.40
	Taylor [16]	90.12	88.64	86.19	83.28
	HRank [12]	91.30	90.35	89.94	85.86
	Proposed absorption pruning	93.51	92.28	92.13	90.50
RSOD	L1 [14]	85.54	83.22	76.20	62.20
	APoZ [15]	83.10	74.72	63.25	51.64
	Taylor [16]	86.74	84.35	80.18	70.62
	HRank [12]	89.90	86.01	80.69	72.06
	Proposed absorption pruning	91.54	92.70	89.43	73.50

4.7. Hyperparameter Experiment for Proposed Pruning Ratio Adjustment Method

In order to explore the influence of the values in the hyperparameter combination of z and α on the absorption pruning method, we fixed the pruning ratio as 60% and conducted experiments on both the SSDD and RSOD data sets. From the experimental results shown in Table 5, it can be seen that the absorption pruning method performed better when the pruning ratio adjustment magnitude α was five. A too-small value of α cannot take full advantage of the pruning ratio adjustment method, while an excessively large α will degrade the network performance. When the pruning ratio adjustment fulcrum z was 300, it was obviously better than when z was 500 and 700. This indicates that, in the object detection task on remote sensing images, the lightweight network should retain more filters with a receptive field below 300 while removing more filters with a receptive field greater than 300. Considering that the input size of the CenterNet101 network is 512×512 , the size of objects in remote sensing images usually does not exceed 300×300 . Therefore, filters with receptive fields exceeding 300 are unnecessary in remote sensing object detection networks. This phenomenon is consistent with our analysis in Section 3.4 regarding the small size and simple features of remote sensing objects.

Table 5. Effects of different hyperparameter combinations on the absorption pruning method.

Data Set	$z = 500$			$z = 300$	$z = 700$	
	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$		$\alpha = 10$	
SSDD	92.13	92.34	91.38	91.93	91.38	91.16
RSOD	89.43	90.07	86.26	90.18	86.26	87.10

4.8. Detection Results of the Pruned CenterNet101

To verify the effectiveness of the absorption pruning method proposed in this paper in eliminating the over-fitting problem, we compared the object detection results of the lightweight network with 60% of filters pruned against the pre-trained network.

Figures 9 and 10 show the visualized comparison results. As seen from the left image of Figure 9b, the pre-trained network presented very good detection performance with a pure sea surface background. In the middle image of Figure 9b, it can be seen that the pre-trained network had misdetection (only five out of seven ships are detected) and false alarm (the lines on land are wrongly detected as a ship) problems in the dense near-shore ship detection scenario. The right image of Figure 9b shows that the pre-trained network incorrectly detected lakes and lines on land as ships. A similar phenomenon can also be observed in Figure 10a, where the pre-trained network failed to detect two closely aligned aircraft and wrongly detected the shadow of an oil tank as the oil tank. However, it can be seen from Figures 9c and 10b that the lightweight detection networks avoided all of the above problems and successfully completed the object detection tasks in different complex scenes. These results prove that the absorption pruning method proposed in this paper not only can retain the detection ability of the pre-trained network but also effectively alleviates the over-fitting problem of the pre-trained network.

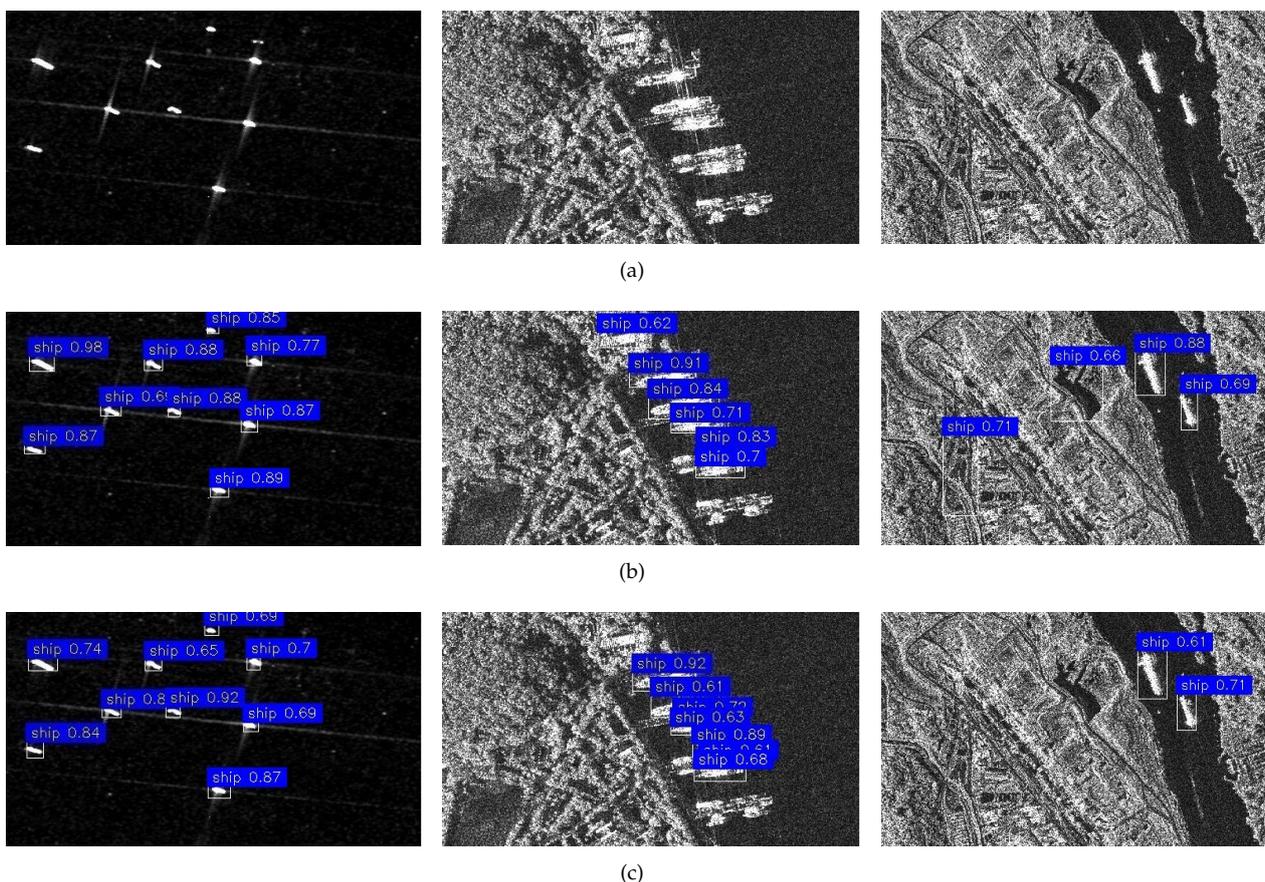


Figure 9. Comparison of object detection results between pre-trained network and pruned lightweight network on SDD data set: (a) input images; (b) object detection results for the pre-trained network; and (c) object detection results for the pruned lightweight network.

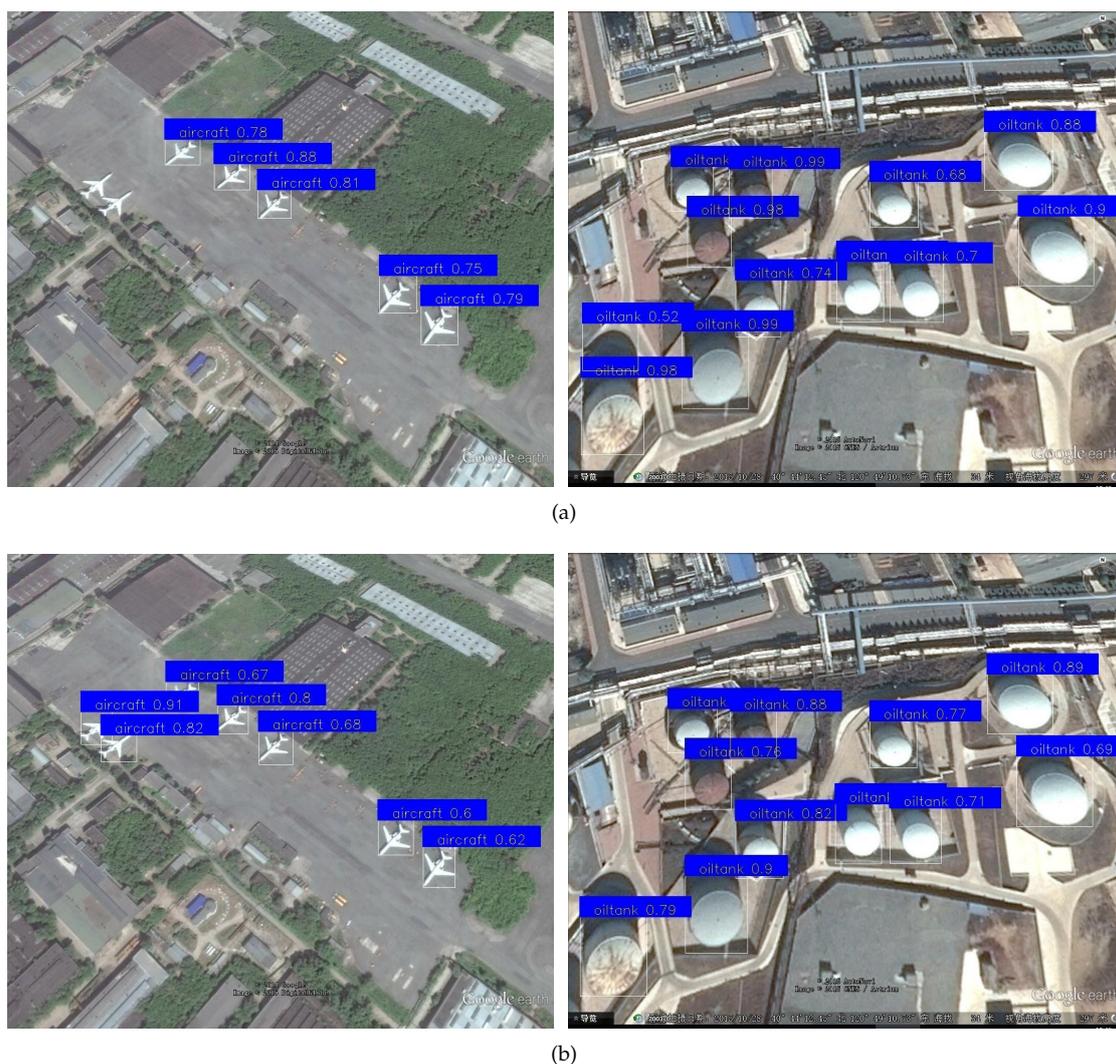


Figure 10. Comparison of object detection results between pre-trained network and pruned lightweight network on RSOD data set: (a) object detection results for the pre-trained network, and (b) object detection results for the pruned lightweight network.

5. Conclusions

In this paper, we proposed a filter pruning method specifically for object detection networks in remote sensing imagery. Unlike the classical iterative pruning pipeline used by existing pruning methods, we innovatively proposed a four-step pruning pipeline that only needs to be executed once. Considering the particularity of the proposed pruning pipeline, we designed a criterion for the selection of filters that are easy to learn rather than selecting unimportant filters, as in existing pruning methods. In addition, we innovatively proposed a pruning ratio adjustment method based on the object characteristics in remote sensing images in order to optimize the design of the pruning method. The experimental results of the absorption pruning method on the SSDD data set indicated that the parameters of the pruned network were less than 10% of that in the pre-trained network, while the object detection accuracy was only reduced by 0.58%. In addition, on both the SSDD and RSOD data sets, the network performance was improved by more than 1% after the absorption pruning method removed 60% of the filters from the pre-trained network. These results demonstrate that the proposed absorption pruning method can effectively remove the redundant parameters in the remote sensing object detection network, thus eliminating the over-fitting problem.

Author Contributions: Conceptualization, J.W.; methodology, J.W.; software, J.W.; validation, Z.Z. and X.M.; formal analysis, Z.C. (Zongyong Cui); investigation, J.W.; resources, Z.C. (Zongyong Cui) and Z.C. (Zongjie Cao); data curation, Z.C. (Zongyong Cui); writing—original draft preparation, J.W.; writing—review and editing, Z.C. (Zongjie Cao); visualization, Z.Z. and X.M.; supervision, Z.C. (Zongyong Cui); project administration, Z.C. (Zongjie Cao); funding acquisition, Z.C. (Zongjie Cao) and Z.C. (Zongyong Cui). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under grants 61971101 and 62271116.

Data Availability Statement: All data sets used in this article are public data sets.

Acknowledgments: The author would like to thank the Naval Aeronautical and Astronautical University for providing the SSDD data set, as well as Wuhan University for providing the RSOD data set.

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

References

1. Zhang, J.; Xing, M.; Sun, G.C.; Li, N. Oriented Gaussian Function-Based Box Boundary-Aware Vectors for Oriented Ship Detection in Multiresolution SAR Imagery. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–15. [[CrossRef](#)]
2. Wang, X.; Li, G.; Zhang, X.P.; He, Y. Ship Detection in SAR Images via Local Contrast of Fisher Vectors. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *58*, 6467–6479. [[CrossRef](#)]
3. Wei, S.; Zeng, X.; Qu, Q.; Wang, M.; Su, H.; Shi, J. HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation. *IEEE Access* **2020**, *8*, 120234–120254. [[CrossRef](#)]
4. Zhang, T.; Zhang, X.; Ke, X.; Zhan, X.; Shi, J.; Wei, S.; Pan, D.; Li, J.; Su, H.; Zhou, Y.; et al. LS-SSDD-v1.0: A Deep Learning Dataset Dedicated to Small Ship Detection from Large-Scale Sentinel-1 SAR Images. *Remote. Sens.* **2020**, *12*, 2997. [[CrossRef](#)]
5. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2017**, *55*, 3639–3655. [[CrossRef](#)]
6. Chen, H.; Qi, Z.; Shi, Z. Remote Sensing Image Change Detection With Transformers. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 1–14. [[CrossRef](#)]
7. Lin, Z.; Ji, K.; Leng, X.; Kuang, G. Squeeze and Excitation Rank Faster R-CNN for Ship Detection in SAR Images. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 751–755. [[CrossRef](#)]
8. Jiang, S.; Zhu, M.; He, Y.; Zheng, Z.; Zhou, F.; Zhou, G. Ship Detection with Sar Based on Yolo. In Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September 2020–2 October 2020; pp. 1647–1650. [[CrossRef](#)]
9. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote. Sens.* **2019**, *11*, 765. [[CrossRef](#)]
10. Wang, H.; Qin, C.; Zhang, Y.; Fu, Y. Neural Pruning via Growing Regularization. In Proceedings of the International Conference on Learning Representations, Online, 3–7 May 2021. [[CrossRef](#)]
11. Dai, C.; Liu, X.; Cheng, H.; Yang, L.T.; Deen, M.J. Compressing Deep Model with Pruning and Tucker Decomposition for Smart Embedded Systems. *IEEE Internet Things J.* **2021**, *9*, 14490–14500. [[CrossRef](#)]
12. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. HRank: Filter Pruning Using High-Rank Feature Map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1526–1535. [[CrossRef](#)]
13. Verma, V.K.; Singh, P.; Namboodiri, V.P.; Rai, P. A “Network Pruning Network” Approach to Deep Model Compression. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 2–5 March 2020; pp. 2998–3007. [[CrossRef](#)]
14. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017. [[CrossRef](#)]
15. Hu, H.; Peng, R.; Tai, Y.; Tang, C. Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures. *arXiv* **2016**, arXiv:1607.03250. [[CrossRef](#)]
16. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Inference. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017. [[CrossRef](#)]
17. Cui, Z.; Wang, X.; Liu, N.; Cao, Z.; Yang, J. Ship Detection in Large-Scale SAR Images Via Spatial Shuffle-Group Enhance Attention. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 379–391. [[CrossRef](#)]
18. Denil, M.; Shakibi, B.; Dinh, L.; Ranzato, M.; de Freitas, N. Predicting Parameters in Deep Learning. In Proceedings of the Twenty-seventh Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013.
19. Zhang, Y.; Lan, Z.; Dai, Y.; Zeng, F.; Bai, Y.; Chang, J.; Wei, Y. Prime-aware adaptive distillation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 658–674.

20. Courbariaux, M.; Bengio, Y. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1. *arXiv* **2016**, arXiv:1602.02830. [[CrossRef](#)]
21. Zhang, K.; Liu, G.; Lv, M. RUFFP: Reinitializing unimportant filters for soft pruning. *Neurocomputing* **2022**, *483*, 311–321. [[CrossRef](#)]
22. Luo, J.H.; Zhang, H.; Zhou, H.Y.; Xie, C.W.; Wu, J.; Lin, W. ThiNet: Pruning CNN Filters for a Thinner Net. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2525–2538. [[CrossRef](#)]
23. Zhang, S.; Wu, R.; Xu, K.; Wang, J.; Sun, W. R-CNN-Based Ship Detection from High Resolution Remote Sensing Imagery. *Remote Sens.* **2019**, *11*, 631. [[CrossRef](#)]
24. Zhang, C.; Xiong, B.; Kuang, G. Ship Detection and Recognition in Optical Remote Sensing Images Based on Scale Enhancement Rotating Cascade R-CNN Networks. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 3545–3548. [[CrossRef](#)]
25. Li, J.; Qu, C.; Shao, J. Ship detection in SAR images based on an improved faster R-CNN. In Proceedings of the 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA), Beijing, China, 13–14 November 2017; pp. 1–6. [[CrossRef](#)]
26. Chen, Z.; Gao, X. An Improved Algorithm for Ship Target Detection in SAR Images Based on Faster R-CNN. In Proceedings of the 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP), Wanzhou, China, 9–11 November 2018; pp. 39–43. [[CrossRef](#)]
27. Nie, X.; Duan, M.; Ding, H.; Hu, B.; Wong, E.K. Attention Mask R-CNN for Ship Detection and Segmentation From Remote Sensing Images. *IEEE Access* **2020**, *8*, 9325–9334. [[CrossRef](#)]
28. Yin, R.; Xu, Q.; Ding, Y. Ship Detection from Optical Remote Sensing Imagery Based on Scene Classification and Saliency-Tuned Retinanet. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 3553–3556. [[CrossRef](#)]
29. Lu, X.; Ji, J.; Xing, Z.; Miao, Q. Attention and Feature Fusion SSD for Remote Sensing Object Detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–9. [[CrossRef](#)]
30. Guo, Y.; Chen, S.; Zhan, R.; Wang, W.; Zhang, J. LMSD-YOLO: A Lightweight YOLO Algorithm for Multi-Scale SAR Ship Detection. *Remote Sens.* **2022**, *14*, 4801. [[CrossRef](#)]
31. Xu, D.; Wu, Y. Improved YOLO-V3 with DenseNet for Multi-Scale Remote Sensing Target Detection. *Sensors* **2020**, *20*, 4276. [[CrossRef](#)]
32. Shi, P.; Jiang, Q.; Shi, C.; Xi, J.; Tao, G.; Zhang, S.; Zhang, Z.; Liu, B.; Gao, X.; Wu, Q. Oil Well Detection via Large-Scale and High-Resolution Remote Sensing Images Based on Improved YOLO v4. *Remote Sens.* **2021**, *13*, 3243. [[CrossRef](#)]
33. He, Z.; Huang, L.; Zeng, W.; Zhang, X.; Jiang, Y.; Zou, Q. Elongated Small Object Detection from Remote Sensing Images Using Hierarchical Scale-Sensitive Networks. *Remote Sens.* **2021**, *13*, 3182. [[CrossRef](#)]
34. Everingham, M.; Eslami, S.M.A.; Gool, L.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
35. Lin, T.Y.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755. [[CrossRef](#)]
36. Liu, S.; Yu, G.; Yin, R.; Yuan, J.; Shen, L.; Liu, C. Joint Model Pruning and Device Selection for Communication-Efficient Federated Edge Learning. *IEEE Trans. Commun.* **2022**, *70*, 231–244. [[CrossRef](#)]
37. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
38. Luo, J.; Wu, J. An Entropy-based Pruning Method for CNN Compression. *arXiv* **2017**, arXiv:1706.05791. [[CrossRef](#)]
39. Wang, J.; Jiang, T.; Cui, Z.; Cao, Z. Filter pruning with a feature map entropy importance criterion for convolution neural networks compressing. *Neurocomputing* **2021**, *461*, 41–54. [[CrossRef](#)]
40. He, Y.; Kang, G.; Dong, X.; Fu, Y.; Yang, Y. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 2234–2240. [[CrossRef](#)]
41. Wang, S.; Wang, M.; Zhao, X.; Liu, D. Two-Stage Object Detection Based on Deep Pruning for Remote Sensing Image. In *International Conference on Knowledge Science, Engineering and Management*; Liu, W., Giunchiglia, F., Yang, B., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 137–147.
42. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
43. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 448–456.
44. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2755–2763.
45. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [[CrossRef](#)]