remote sensing

MDPI

Article

SAR Target Recognition in Large Scene Images via Region-Based Convolutional Neural Networks

Zongyong Cui, Sihang Dang, Zongjie Cao *, Sifei Wang and Nengyuan Liu

Center for Information Geoscience, University of Electronic Science and Technology of China, Chengdu 611731, China; zycui@uestc.edu.cn (Z.C.); dangsihang@outlook.com (S.D.); surf_wang@126.com (S.W.); nengyuanliu@outlook.com (N.L.)

* Correspondence: zjcao@uestc.edu.cn

Received: 31 March 2018; Accepted: 17 May 2018; Published: 17 May 2018

Abstract: In this paper, a new Region-based Convolutional Neural Networks (RCNN) method is proposed for target recognition in large scene synthetic aperture radar (SAR) images. To locate and recognize the targets in SAR images, there are three steps in the traditional procedure: detection, discrimination, classification and recognition. Each step is supposed to provide optimal processing results for the next step, but this is difficult to implement in real-life applications because of speckle noise and inefficient connection among these procedures. To solve this problem, the RCNN is applied to large scene SAR target recognition, which can detect the objects while recognizing their classes based on its regression method and the sharing network structure. However, size of the input images to RCNN is limited so that the classification could be accomplished, which leads to a problem that RCNN is not able to handle the large scene SAR images directly. Thus, before the RCNN, a fast sliding method is proposed to segment the scene image into sub-images with suitable size and avoid dividing targets into different sub-images. After the RCNN, candidate regions on different slices are predicted. To locate targets on large scene SAR images from these candidate regions on small slices, the Non-maximum Suppression between Regions (NMSR) is proposed, which could find the most proper candidate region among all the overlapped regions. Experiments on 1476 × 1784 simulated MSTAR images of simple scenes and complex scenes show that the proposed method can recognize all targets with the best accuracy and fastest speed, and outperform the other methods, such as constant false alarm rate (CFAR) detector + support vector machine (SVM), Visual Attention+SVM, and Sliding-RCNN.

Keywords: SAR target recognition; large scene; region-based convolutional neural networks

1. Introduction

Spaceborne and airborne synthetic aperture radar (SAR) is able to operate in all-weather all-time conditions to generate high resolution SAR images; thus, SAR has been widely used both in military and civil fields. SAR image interpretation is the inevitable way to fully obtain the information of a specific SAR image. However, because of scattering imaging mechanism and speckle noise in SAR images, interpretation and understanding of SAR images is much more difficult than that of optical photos [1]. Several years ago, many algorithms based on deep learning have been well established for SAR automatic target recognition (ATR) [2,3], which is focused on in this work.

To achieve automatic target recognition (ATR) systems for SAR interpretation, MIT Lincoln Laboratory put forward a standard SAR ATR architecture. The structure contains three stages: detection, discrimination and classification/recognition [4]. The detection part is to extract candidate regions that might include targets from SAR images with a constant false alarm rate (CFAR) detector. However, these regions include not only targets that we want to recognize, such as tanks and vehicles,

but also background objects, for example, buildings and trees. Then, in the discrimination part, a discriminator trained by several manually designed features is used to divide regions into two classes, target or not target, to reduce false alarms. Output of discriminator will be sent to the classifier to give out the type of targets, and this operation is called classification/recognition. Performance of classification will be greatly influenced if targets are in extended operating conditions. The accuracy will also decrease significantly if any stage of SAR ATR is not well designed or not suitable for the current operating condition [5]. To provide enough samples for these SAR ATR models, a project called model-based Moving and Stationary Target Acquisition and Recognition (MSTAR) was carried out [6]. In the past few years, most researchers just focused on one part of these three stages, put forward theories such as scene segmentation, target discrimination, feature extraction, classifier design and so on. However, these theories and algorithms just out-perform in specific operating conditions, which makes them not able to be applied universally. This will also result in procedure isolation and increased difficulty of detection and recognition connection.

A reliable and universal system requires effective connection between detection and recognition, so End-to-End models were proposed [7,8], and apply robust trainable classifiers, such as Adaboost and support vector machine (SVM) [9–11] to realize SAR ATR. Though these End-to-End SAR ATR models can realize the connection among three stages, they are still not as efficient as we expect. Problems like size difference and target position mismatch between training samples and interpreting images still need to be solved.

Neural networks could extract features automatically and have obtained remarkable achievements in optical image detection. Regions with convolutional neural network (R-CNN), Fast R-CNN and Faster R-CNN were proposed, which can recognize different kinds of objects with different sizes in optical images with high accuracy [12–14]. You Only Look Once (YOLO) was proposed, which achieved fast detection, but the accuracy is lower than Faster R-CNN [15]. Liu proposed Single Shot MultiBox Detector (SSD), which was a compromise in accuracy and speed between Faster R-CNN and YOLO [16].

Inspired by these advanced methods, many researchers tried to introduce deep learning methods into the field of SAR target detection and recognition to solve problems in End-to-End models. Morgan realized SAR target recognition with CNN and verified its ability to extract SAR target features [17]. Hansen and Ding solved problems of target displacement, random speckle noise and pose missing with CNN, and proved that CNN has a favorable robustness compared with other classifiers [18,19]. Chen built an all-convolutional network called A-ConvNets to realize recognition of SAR targets and solved the over-fitting problem caused by lack of training samples [20]. Researchers above have verified that convolutional neural networks can be realized in every process of SAR image interpretation, but they are still in the range of three-stage process, a universal model that can interpret large scene SAR images at once is still under exploration.

In order to break the bottleneck of traditional three-stage process, these research results gave us the inspiration for applying deep learning methods and strategies in optical image detection to fields of SAR target detection and recognition with consideration of unique features of SAR images. In this paper, we have increased randomness of target distribution with strategies of segmentation to avoid over-fitting problems cased by small training data sets, and Non-maximum Suppression among Regions (NMSR) is proposed to choose the most proper candidate boxes among adjacent regions. The CNN network realized not only integration of detection and recognition, but also an effective and efficient performance dealing with large scene SAR images.

The remainder of this paper includes an introduction to the structure and components of our convolutional neural network as well as training and testing details in Section 2, exhibition of experimental results in Section 3, analyses of experimental results in Section 4 and conclusions in Section 5.

2. SAR Target Recognition Based on Region-Based Convolutional Neural Networks

At first, the flow chart of the model that we used to realize interpretation of large scene SAR images is shown in Figure 1.

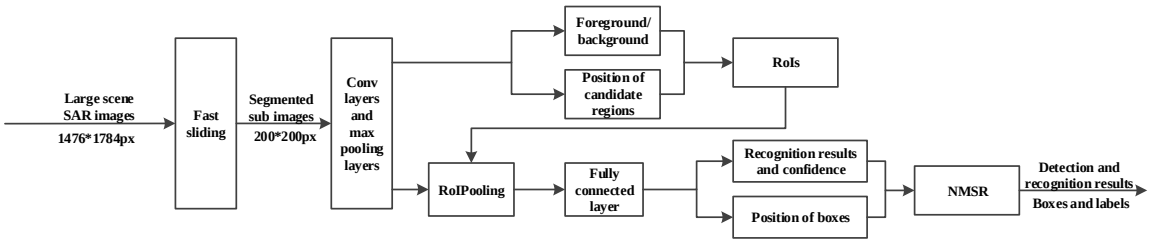


Figure 1. Flow chart of our model. (NMSR: non-maximum suppression among regions; SAR: synthetic aperture radar).

Then, structure of the CNN used to realize integration of target detection and recognition is given out, and every component of this CNN is described in detail to show how it works.

Finally, a nonlinear softmax function is used to realize classification and a fully connected layer is used to locate bounding boxes of targets [21].

Components of the CNN network are presented in details as below.

2.1. Fast Sliding

Before images are sent into the CNN, they will be re-sized to a certain size so that feature maps of them after CNN are of the same size, and classification could be finished accurately. With application of this strategy, the network is endowed with a certain degree of scale invariance for small images of different sizes and scales. However, as for large scene images, once they are re-sized to a small size, disappearance of most pixels will lead to information lost and result in a sharp drop of target detection [22], such as missing most targets, inaccuracy of target location and lower confidence.

Thus, it is necessary to cut large scene images into sub-images, and then send these sub-images into convolutional neural network to realize detection and recognition accurately.

Compared with recognition of SAR targets, location and detection of SAR targets are much more time-consuming because of random distribution of sparse targets. Furthermore, sliding operation is time-consuming as well. A sliding window with stride of one pixel can achieve the best detection performance, but it has the lowest efficiency. Using a sliding window with a larger stride will decrease time consumption. Since randomly distributed targets will appear in any place in an image, if one sliding window covers only a part of the target, detection and recognition results of this target in this sliding window will be not accurate at all. To solve the problem, we need to make sure that any potential target in a large scene image will be completely covered by at least one sliding window. If the size of the target is $w_t * h_t$, the size of sliding window is z; then, overlap k among adjacent slices should be limited as follows:

$$k \geq \frac{\max(w_t, h_t)}{z}. \tag{1}$$

This strategy could ensure an accurate detection result and a smaller time consumption. A diagram of this fast sliding strategy is shown in Figure 2.

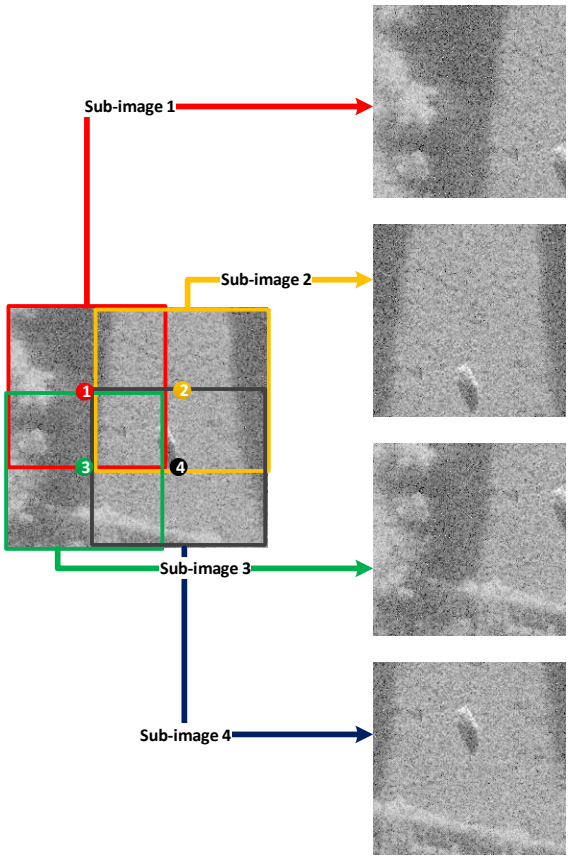


Figure 2. Diagram of fast sliding.

2.2. Structure of Region-Based Convolutional Neural Networks

It has been proved that the number of convolutional layers plays an important role in the performance of CNN [23]; as for SAR images, a network with a simple structure and only several layers could get a satisfying performance. As shown in Figure 3, a CNN network with five convolutional layers (conv layers) and two max pooling layers is used to extract features in SAR images. The number after the symbol @ is convolution kernel size.

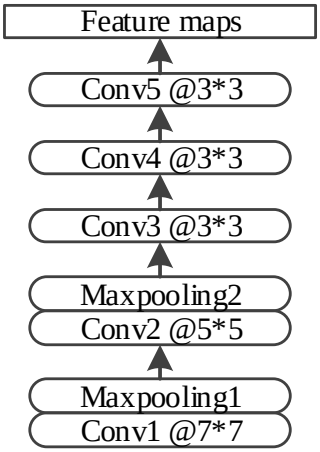


Figure 3. The convolutional neural network used to generate feature maps. (conv: number of feature maps; @: filter size).

During convolution operation, each convolution kernel is set to share the same weights so that this kernel could detect the only specific feature at different positions of the previous layer. Strategy of weight sharing can significantly reduce the number of free parameters, which can make training faster and more efficient [14]. Then, the input image is sent to do convolution operations with several convolution kernels, and outcomes of each convolution operation are organized as a set of 2D arrays [24]. Thus, a specific convolution kernel can be treated as a feature detector as well. However, a useful feature is more likely to distribute in a small part of the input, so it is necessary to make use of obvious features and drop useless parts. In our network, there is always one max pooling layer after each of the first two conv layers [25]. The max pooling layer is to find out the largest unit in a local patch and realize a subsample on the output of the conv layer to extract local optimal features—the 2D array output of this is called the feature map [26].

2.2.1. Convolutional Layer

Conv layers are to extract different features in the input feature maps with plenty of convolution kernels. Input feature maps are connected to output feature maps through a specific designed strategy, so convolution kernels are forced to extract different features. If some input feature maps $O_i^{(l-1)}$ $(i = 1, ..., I)$ are connected to one output feature map $O_j^{(l)}$ $(j = 1, ..., J)$, and $O_i^{(l-1)}(x, y)$ is a unit of the ith output feature map in layer $l-1$ at position (x, y), $O_j^{(l)}(x, y)$ is the unit of the jth output feature map in layer l at position (x, y). Let manually configured parameters $k_{ji}^{(l)}(u, v)$ present the convolution kernel connecting these input feature maps and output feature maps, $b_j^{(l)}$ presents trainable bias of the output feature map. The convolution is computed as below:

$$O_j^{(l)}(x, y) = f(G_j^{(l)}(x, y)), \tag{2}$$

$$G_j^{(l)}(x, y) = \sum_{i=1}^{I} \sum_{u,v=0}^{K-1} k_{ji}^{(l)}(u, v) \cdot O_i^{l-1}(x - u, y - v) + b_j^{(l)}, \tag{3}$$

where $f(\cdot)$ is the nonlinear activation function, and here we used the ReLu function. $G_j^{(l)}(x, y)$ denotes the weighted sum of inputs to the output feature map at position (x, y). Other parameters include the number of input feature maps I, kernel size $K \times K$, convolution stride S and zero padding P. Stride S specifies the intervals while using convolution kernels to the input feature maps, and lead to lower dimensional outputs. Padding P is used to preserve the spatial size of input feature maps so that features appearing in edges can be extracted as well. A common strategy is to pad the input with zeros on each side of the input. A conv layer with kernels of size $K \times K$, stride of S and I feature maps of size $W_1 \times H_1$ as input will get output composed of J feature maps of size $W_2 \times H_2$, where

$$W_2 = (W_1 - K + 2P)/S + 1, \tag{4}$$

$$H_2 = (H_1 - K + 2P)/S + 1. \tag{5}$$

Recent literature has indicated that using filters with small sizes, such as 3×3 or 5×5, with a stride of 1 usually produces satisfying performance [23]. The number of feature maps on each layer is determined through cross validation. The common setting is that lower layers tend to have fewer feature maps and higher layers tend to have more. However, in some deep models, middle layers have much more feature maps and highest layer have fewer to reduce redundancy [27]. With strategy of weight sharing and a specially designed connection between layers, kernels are forced to extract a certain feature in a different position of feature maps, and the number of parameters is reduced as well.

2.2.2. Activation Function

The relationship between input image and output box and label is supposed to be a highly nonlinear mapping, so nonlinear activation function needs to be added at each conv layer [28]. In traditional convnets, a hyperbolic tangent function $f(x) = \tanh(x)$ or a sigmoid function $f(x) = 1/(1 + \exp(-x))$ are used as the nonlinear activation function at conv layer [29]. However, hyperbolic function is not good enough because when its output is near -1 or 1, the gradient of this function tends to zero, and this might terminate training. Thus, it used to be hard to train deep models due to these saturating nonlinearities. Recent research has found a nonsaturating nonlinearity that often works well in training, the rectified linear unit (ReLU) [30]. This function is given by

$$f(x) = \max(0, x). \tag{6}$$

ReLU can significantly reduce training time and deep networks using ReLU usually can reach their best performance with only supervised training on large labeled data sets without requiring any unsupervised pretraining.

2.2.3. Pooling Layer

In our model, there is only one pooling layer after each of the first two conv layers. Max pooling operation is used to select the pixel with maximum value in a certain group of pixels as the most significant and efficient features. Though this operation will throw away some information, it achieves shift and distortion invariance somehow. If the input image shifts a small amount, this operation can make sure that most outputs of pooling will not change [31].

The max pooling operation is defined as

$$O_i^{(l+1)}(x, y) = \max_{u,v-0,\ldots,M-1} O_i^{(l)}(x \cdot s + u, y \cdot s + v), \tag{7}$$

where M is the pooling size. Stride s determines distance between neighbor pooling windows.

As said before, pooling operation throws away some information during subsampling, so a larger pooling size is going to result in worse performance of feature extraction. This is why actually only two hyperparameter settings: 2×2 as pooling size with a stride of 2 and 3×3 as pooling size with a stride of 2 are used in practice [32].

A typical connection between conv layer and pooling layer is shown in Figure 4. As we have shown before, f_x is convolution operation, b_x and b_{x+1} are biases, W_{x+1} is the weight, S_{x+1} is the output feature map.

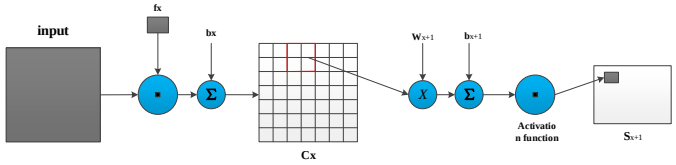


Figure 4. Typical connection between conv layer and pooling layer.

2.2.4. Softmax Classification

To deal with multi-class classification problems, the softmax nonlinearity is used in the output layer to generate posterior probabilities over each class. Due to that, after one fully connected layer, a two-dimensional vector is mapped into a K-dimensional weighted vector, and the value of each

element represents the relative probability of class of this input $p_i = P(y = i|x)$, for $i = 1, ..., K$. The softmax function can be presented as

$$p(i) = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)}, \tag{8}$$

where $\theta_j^T x$ is the jth element of output vector.

As for the classification procedure, with a set of N_{cls} labeled training samples, the loss function can be defined as

$$L_{cls}(\theta_j^T) = -\frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} \log P(p_i^* | p_i ; \theta_j^T), \tag{9}$$

where p_i is the predicted label of the ith sample and p_i^* refers to the ground-truth label.

With these ground-truth training samples, a better performed classifier can be obtained automatically through adjusting the trainable parameter θ_j^T, while minimizing this loss function, which means increasing the probability of giving out correct labels.

2.2.5. Region Proposal

To realize the integration of classification and location, a small network is used to slide on the final feature maps obtained from previous layers [14]. As shown in Figure 5, the network is fully connected with a $N \times N$ window on feature maps to generate a certain number of anchor boxes with different sizes and scales.

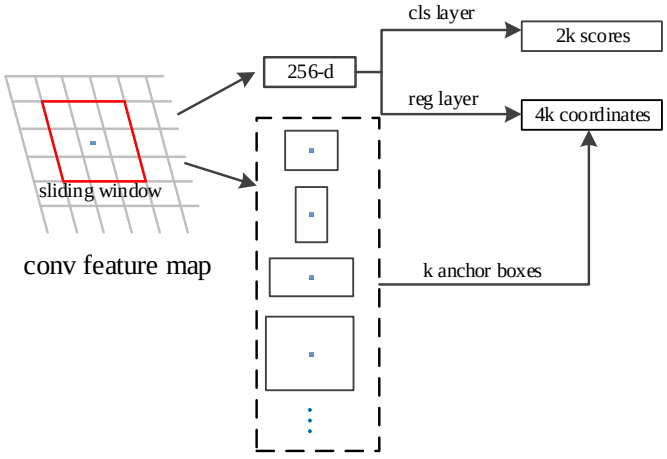


Figure 5. Region proposal network.

Then, these anchor boxes are sent to a fully connected layer called the intermediate layer and mapped into a lower dimensional vector. With these vectors as input, one fully connected layer called the classification layer (cls layer) is used to output scores of being an object or not an object to pick foregrounds out of backgrounds, and another fully connected layer called the regression layer (reg layer) is used to generate parameters of each anchor box and fine-tune the boundaries so that these boxes are accurate enough to locate objects through feature maps.

The network operates as a sliding window, so the fully connected layers are shared among all spatial locations. Anchor boxes with different sizes and scales ensures that they are able to locate objects with different sizes and shapes.

2.2.6. Loss Function

Anchors generated by that small network for region proposal are divided into two kinds: the positive and the negative. Anchors that have the highest Intersection-over-Union (IoU) overlap

with one ground-truth box or have an IoU overlap higher than 0.7 with any ground-truth box will be treated as positive anchors. In some previous references, the ratio is set to 0.7, and 0.7 is an empirical value. In general, this score >0.5 can be considered a good result. This means that any other bounding boxes with IoU higher than 0.7 will be treated as similar bounding boxes, and they will be deleted since they are not of the highest classify score. Those that have IoU overlap lower than 0.3 with all ground-truth boxes will be treated as negative anchors. The number of positive anchors chosen randomly for training equals that of negative anchors in each training sample. Other anchors do not participate in training.

A multi-task loss function is used to realize integration of classification and location. It is described as below:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*), \tag{10}$$

where p_i is the predicted probability of the ith anchor being an object. If this anchor is positive, then the ground-truth label p_i^* will be set to 1, or will be set to 0 if the anchor is negative. t_i is a vector with four parameters that represents the coordinates of the predicted bounding box. t_i^* is the parameters of the ground-truth box associated with a positive anchor.

A log loss L_{cls} over two classes (being an object or not) is used as classification loss, which is described in formula 8. For regression, a smooth L1 function defined as

$$S_{L1}(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & \left|t_i - t_i^*\right| < 1 \\ \left|t_i - t_i^*\right| - 0.5 & others, \end{cases} \tag{11}$$

which is also a robust loss function, is used as regression loss L_{reg}. The regression loss is activated only for positive anchors and disabled when anchors are negative. Classification term and regression term are normalized with N_{cls}, L_{reg} and a balancing weight λ.

Those four coordinates in t_i and t_i^* are designed as below:

$$\begin{cases} t_x = (x - x_a)/w_a, \\ t_y = (y - y_a)/h_a, \\ t_w = \log(w/w_a), \\ t_h = \log(h/h_a), \end{cases} \tag{12}$$

$$\begin{cases} t_x^* = (x^* - x_a)/w_a, \\ t_y^* = (y^* - y_a)/h_a, \\ t_w^* = \log(w^*/w_a), \\ t_h^* = \log(h^*/h_a), \end{cases} \tag{13}$$

where x and y denote the coordinates of the center of the box, and w and h denote the width and height of the box. x, x_a and x^* are variables of predicted box, anchor box and ground-truth box, respectively. It can be seen that this loss function can transform an anchor box to a nearby ground-truth box. Unlike previous feature-map-based bounding regression methods, the features on feature maps that we used for regression have the same spatial size. Then, a set of k bounding-box regressors are learned so that it can work for different sizes. Every regressor is independent and responsible for a specific scale and aspect ratio. Thus, even if features are of a fixed size or scale, it is still possible to predict boxes of various sizes.

2.3. Non-Maximum Suppression between Regions

After fast sliding, each slice is sent to the CNN network sequentially to detect and locate targets. Once the predicted bounding boxes and classification scores are generated by regression layer and classification layer respectively, the next task is to find out the most proper bounding box. Bounding boxes in the common area of adjacent slices are generated by different slices, so the traditional

Non-maximum Suppression (NMS) method could not be simply used to deal with these bounding boxes. The strategy of Non-maximum Suppression between Regions (NMSR) is proposed and used to find out the most proper bounding boxes, which also achieved admiring performance. The method of NMSR is described as below.

As each slice has a fixed position in large scene SAR image, if a slice of large scene image is the *i*th from left to right and the *j*th from top to bottom, the absolute position (x^*, y^*) of a pixel in this slice can be calculated through the function below:

$$\begin{cases} x^* = x + (1 - 0.3) \times w \times (i - 1), \\ y^* = y + (1 - 0.3) \times h \times (j - 1), \end{cases} \tag{14}$$

where w and h is the size of each slice, and (x, y) is the coordinate of a pixel in this slice.

To reduce computation complexity, the operation of NMS is executed in every slice before NMSR.

At first, the bounding box with the highest classify score is found out and set as a compared box. Any other bounding boxes have an intersection over the union (IoU) higher than a certain ratio with this compared box will be deleted.

Then, bounding boxes with the highest classify score in the rest of the bounding boxes are set as the compared box. Calculate IoU with all the rest of the bounding boxes.

Repeat this operation until bounding boxes with the top N classify scores are discovered. After this, these N bounding boxes with the highest classify scores are left after NMS in each slice. Any other bounding boxes are ignored and will not be calculated in the next step.

Finally, all coordinates of these left bounding boxes in each corresponding slice are replaced by their absolute positions, and operation of NMS is applied again among all these bounding boxes with absolute positions. NMS of bounding boxes in common areas is also achieved. Furthermore, the most proper bounding boxes with their coordinates on large scene images are determined after these operations.

A diagram that shows each step of this large scene SAR image detection and recognition method is shown in Figure 6.

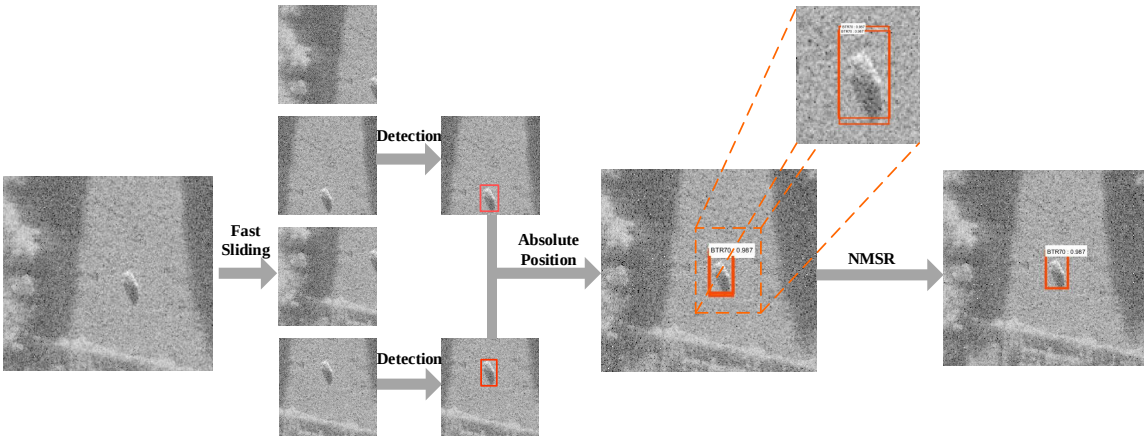


Figure 6. Non-maximum suppression between regions.

3. Experiments

3.1. Experimental requirements

In this paper, the experimental data are from the MSTAR dataset, which is widely used in testing and comparing the performance of SAR detection and recognition algorithms. The collection of the MSTAR dataset was supported by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency. A large amount of SAR images were collected, including different target

types, aspect angles, and depression angles. An x-band, HH polarized SAR in a 0.3-m resolution is used to collect these images. However, only a small part of these images are publicly released. The type of targets we used are BMP2 (tank), BTR70 (armored car), and T72 (tank). The images were captured at two different depression angles 15° and 17° with $190 \sim 300$ different aspect versions, which are full aspect coverage over 360°. Optical images and SAR images of these three types of targets with similar aspect angles are shown in Figure 7.

In our experiment, overlap among adjacent sliding windows is set as 0.3 since widths and heights of all targets in training samples are smaller than 60 pixels. A sliding window of 200×200 is used to cut large scene images into small slices before detection in our model. With this strategy, every potential target will appear completely in at least one slice. As for slices occurring at the edges of large scene SAR images, the excess part will be padded with zero. This strategy has achieved an efficient segmentation, much faster than that with a stride of one pixel. Furthermore, the larger the sliding window we use, the faster detection and recognition will be accomplished. Because of the size and scale invariance of our network, even though the size of the sliding window we use is about twice that of training samples in our experiment, performance of detection and recognition is still of high quality. The final result has achieved balance between time consumption and accuracy of detection and recognition.

Figure 7. Optical images (**top**) and corresponding synthetic aperture radar (SAR) images (**bottom**). From left to right: BMP2, BTR70, T72.

In total, 2268 slices of BMP2, BTR70 and T72 are chosen randomly and used for generating training patches, as shown in Table 1. One-hundred slices of each type of target above are left for testing.

Table 1. Numbers of BMP2, BTR70, and T72 slices for generating training samples.

Class	No. Images
BMP2	1185
BTR70	329
T72	754

Before training, every 128×128 SAR target slice is randomly sampled into 90×90 patches. Because there is only one target in the center of each slice, the patch size of 90×90 could ensure that the target appears in each patch completely. This operation will increase the number of training samples as well as the randomness of the target position, which ensures that the CNN model after training could find targets in different positions of every slice. With this strategy, each slice can be increased at most $(128 - 90 + 1) \times (128 - 90 + 1) = 1521$ times. However, actually, considering the redundancy of these 1521 patches and the amount of training samples required, five randomly chosen patches of each slice are generated and chosen as training samples. Thus, there are $2268 \times 5 = 11,340$ training samples in total. The example of the original slice and corresponding five randomly generated patches are shown in Figure 8.

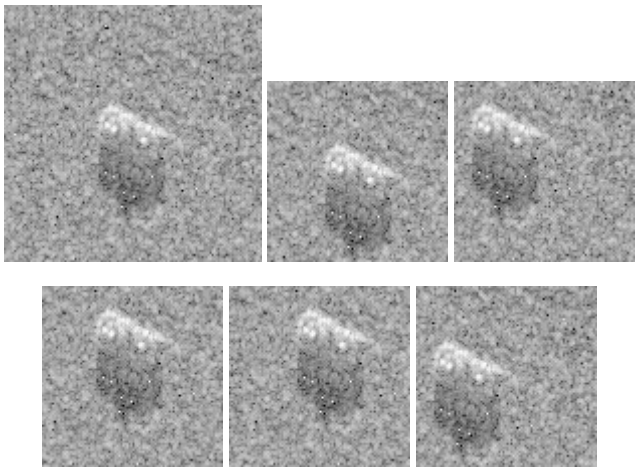


Figure 8. Original image and five randomly generated patches.

Ground-truth boxes are marked manually, which just cover the target in these 11,340 90 × 90 patches of three different targets. The information saved includes coordinates of ground-truth box and target type in each patch. These patches are re-sized to 800 × 800 so that features in these patches are well extracted. These patches along with their corresponding coordinates of ground-truth box and target type are used as training samples to train the CNN network.

To show performance of the network that we used for integration of SAR detecting and recognizing, confirmatory experiments with test samples are done and the results that we get are inspiring.

3.2. Accuracy of Detection and Recognition

In order to determine the detection and recognition accuracy of the CNN network that achieve integration of detection and recognition after training, the remaining 300 SAR images of three different types are used as test samples. Since the size of test samples is similar to that of training samples, there is no need to segment test samples before detection and recognition. Boxes with the highest score from the softmax classifier are shown for these test samples. If the target along with its shadow is surrounded by this box correctly, then the class on this box is regarded as the recognition result. The confusion matrix of this test is shown in Table 2. The true target class is listed on the left and predicted target class is shown on top.

Table 2. Number of chosen images.

Class	BMP2	BTR70	T72	Accuracy(%)
BMP2	93	2	5	93
BTR70	2	97	1	97
T72	5	1	94	94
Average	-	-	-	94.67

3.3. Anti-Noise Performance

Another experiment is done after we get detection and recognition accuracy. The experiment is to explore anti-noise performance of our CNN network. Unlike other well-established work for anti-noise [33], values of a certain proportion of pixels are replaced by that of a Gaussian distribution noise. The replaced proportions are 2%, 4%, 8%, and 16%. An example of original slice and its modified images with noise are shown in Figure 9. The network trained previously on three-target detection and recognition problem is used to deal with these noise added slices. Only if over 80% area of the target

along with its shadow is surrounded by the predicted box with the highest score, and the predicted class for this box is correct can the slice be regarded as a correctly detected and recognized one.

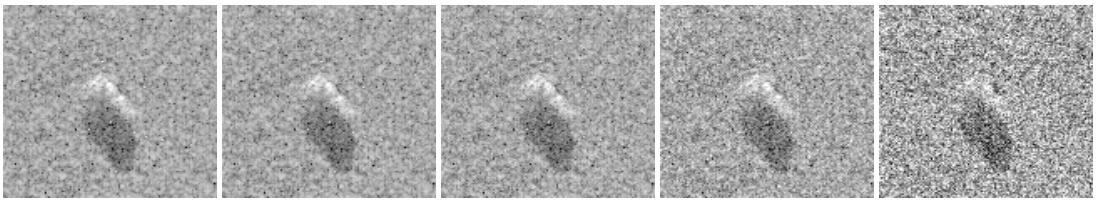


Figure 9. Example of original slice (**left**) and modified images with noise. Proportions are 2%, 4%, 8% and 16%, respectively, from left to right.

Every group of modified images with noise are sent to the CNN network trained previously successively, and accuracy of each group of noise added images is listed in Table 2.

3.4. Performance of Region Proposal Network and Non-Maximum Suppression

In order to evaluate the accuracy of candidate regions generated by the region proposal network in our CNN network, an untrained slice of T72 in MSTAR data set is used as test sample, as Figure 10 (left) shows. Seven candidate regions with the highest scores after detection and recognition are shown in Figure 10. In order to show boundaries of these candidate regions clearly, predicted class and scores are hidden.

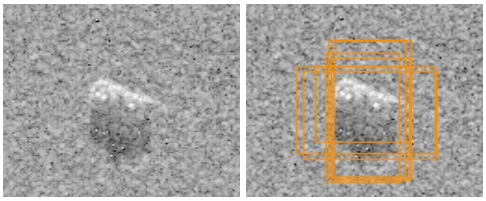


Figure 10. Test sample and the top seven candidate regions.

After generating the candidate regions, feature maps surrounded by these candidate regions are used as inputs to fully connected layer in CNN. Then, the strategy of NMS is used to find out and show the most proper box with a label on it. The final result of test samples used above is shown in Figure 11 to show its performance and verify that the integrated network can give the correct labels of candidate regions.

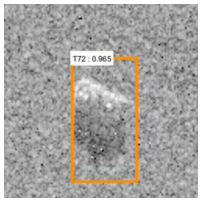


Figure 11. Interpretation result after Non-Maximum Suppression (NMS).

3.5. Detection and Recognition Performance on Large Scene Images

Interpretation of large scene SAR images has also been a difficult problem to solve. Almost all SAR interpretation methods use strategy of segmentation to deal with this problem. However, traditional SAR detection and recognition methods are pretty sensitive to segmentation results, and this would lead to a sharp decrease in interpretation results. In our CNN network, what we need to do is to make it so every target can appear in one segmented slice at least.

To get test samples of the similar size with training samples, a large scene SAR image is segmented into small slices (200 × 200 is used in this paper) to reach balance between interpretation accuracy and interpretation time. An overlap rate (30%) is set in different adjacent slices, so that each target is bound to appear in one slice completely at least. Then, each slice is re-sized to 800 × 800 so that the feature maps are of the same size. These re-sized images are sent to the CNN network sequentially.

At first, performance of this segmentation strategy and NMSR is evaluated through a large scene SAR image with a simple background. This SAR image is not contained in the MSTAR dataset. We embed targets of different types in a 1476 × 1784 large simple scene image in MSTAR. Since targets and scene images are all acquired by SAR sensor with resolution of 0.3 m, embedding targets in scene images is a reasonable simulation.

Then, this SAR image with 14 untrained targets (4 BTR70s, 5 BMP2s and 5 T72s) is interpreted by this CNN network to verify the ability of this segmentation strategy and CNN network. Figure 12 shows the detection and recognition results on this large scene SAR image, and boxes with confidence higher than 0.9 after NMSR are shown. These 14 targets are placed in three lines, but the center of them are not actually on a line so that the sensibility of target position and distribution of this CNN network could be verified. There are five BMP2s in the first line, five BTR70s in the second line and four T72s in the third line. With observation of segmented slices, 10 targets appear in at least two slices. The red boxes represent "BMP2", orange represent "BTR70", and yellow represent "T72". In addition, all classification confidence probabilities are shown besides the boxes.

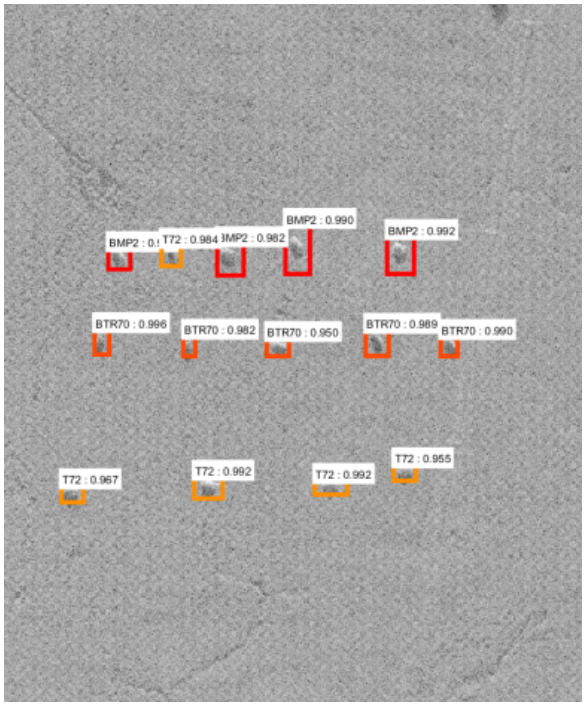
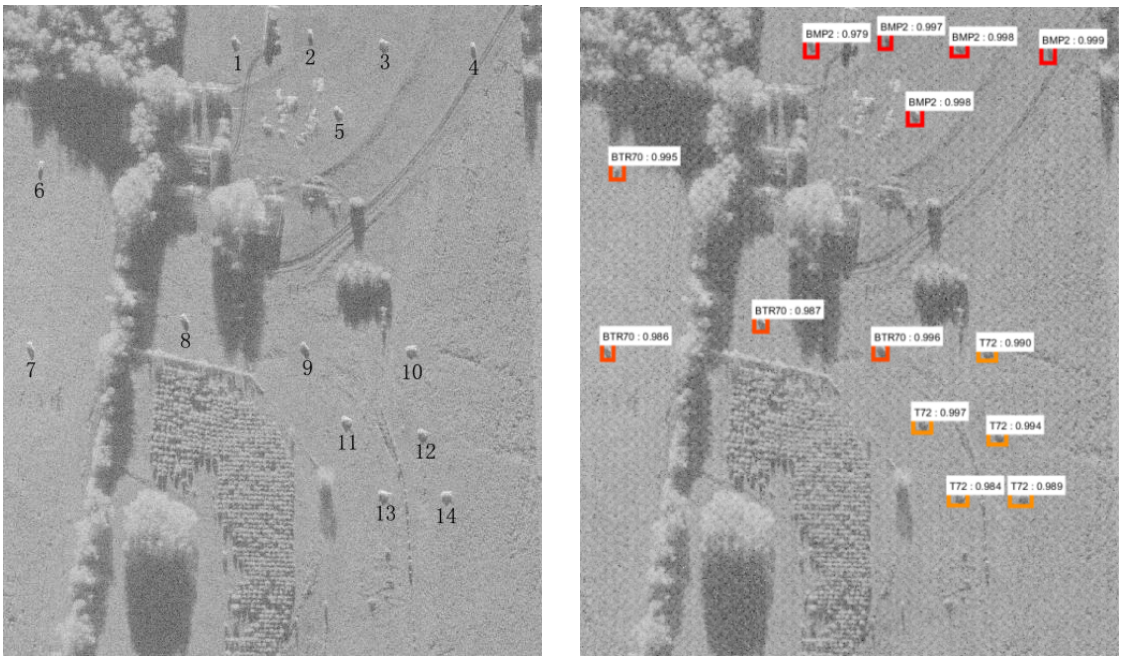


Figure 12. Interpretation result of a large simple scene SAR image.

Finally, interpretation of a complex large scene SAR image of 1476 × 1784 is completed with this CNN network with NMSR. We embedded targets randomly in an image of fields, trees and bushes in MSTAR dataset to make up this complex large scene SAR image. The corresponding class of each target with a given number in Figure 13 (top) is listed in Table 3. The threshold is set as 0.9.



(**a**) (**b**)

Figure 13. Experiment on complex background image (**a**) target distribution; (**b**) interpretation results.

Table 3. Number of chosen images.

Target Class	Corresponding Number
BMP2	1, 2, 3, 4, 5
BTR70	6, 7, 8, 9
T72	10, 11, 12, 13, 14

3.6. Comparison Experiments on Complex Background Image

Contrast experimental results based on two large scene SAR images above are completed and results are listed in Tables 4 and 5. Here, we have listed number of ROIs (Regions of Interest) after detection (No. ROIs), number of correctly detected targets (No. Det), false alarm rate (F.A. Rate), proportion of detected targets in all targets (Det Rate), number of correctly recognized targets (No. Rec), proportion of correctly recognized targets in detected targets (Rec Rate) and time consumption. A Constant False Alarm Rate (CFAR) detector in [34] and visual attention algorithm in [35] are used as detectors and SVM is used to finish classification.

Table 4. Results on a large scene synthetic aperture radar (SAR) image with simple background. (CFAR: constant false alarm rate; SVM: support vector machine; RCNN: region-based convolutional neural networks).

Methods	No. ROIs	No. Det	F.A. Rate	Det Rate	No. Rec	Rec Rate	Time(s)
CFAR + SVM	16	14	12.5%	100%	13	92.86%	37.24
Visual attention + SVM	17	14	17.64%	100%	13	92.86%	15.82
Segmentation + RCNN	9	9	0%	64.29%	9	100%	8.63
Our method	14	14	0%	100%	14	100%	29.32

Table 5. Results on large scene SAR image with complex background.

Methods	No. ROIs	No. Det	F.A. Rate	Det Rate	No. Rec	Rec Rate	Time(s)
CFAR + SVM	21	9	57.14%	64.29%	9	100%	42.73
Visual attention + SVM	24	10	58.33%	71.43%	9	90%	23.92
Segmentation + RCNN	11	11	0%	78.57%	11	100%	10.25
Our method	14	14	0%	100%	14	100%	31.18

4. Discussion

4.1. Analysis on Detection and Recognition Accuracy

It can be seen from Table 2 that all these 300 test samples are surrounded by boxes with highest scores correctly. Furthermore, recognition accuracy of BTR70 achieves the best performance, but BMP2 and T72 are more likely to be recognized as each other. The reason might be that T72 and BMP2 have similar turrets and gun barrels.

As for the speed of this integrated system, these 300 128 $\times$ 128 untrained slices are interpreted, and the total time these slices cost is about 52 seconds from when it starts to deal with the first image until all results of these images are shown on a computer with GTX750Ti.

4.2. Analysis on Anti-Noise Performance

Table 6 has shown a phenomenon that the more pixels are replaced by noises in the original image, the worse the detection and recognition accuracy are. From the experiment, we can find that when the replace ratio is less than 8 percent, the accuracy of recognition is acceptable. However, when the ratio is larger than 8 percent, recognition accuracy will face a sharp drop. To some extent, the result shows that the CNN network can resist the influence of noise. That is to say, compared with training samples, test samples are not allowed to be badly polluted or have too many imaging differences.

Table 6. Accuracy of each group of noise added images.

Proportion	2%	4%	8%	16%
Accuracy	89.33%	83.67%	72.00%	44.33%

4.3. Analysis on Performance of Region Proposal Network and Non-Maximum Suppression

With observation of Figure 10, we find it obvious that all of these top seven candidate regions cover the target, and all of them have surrounded the entire target and its shadow correctly. This means that features of T72 in SAR images are well extracted to generate candidate regions accurately.

From Figure 11, we can see that the CNN network uses NMS to reduce redundancy. It shows a box in the correct place as detection results and the highest recognition confidence of 0.965. The result has exhibited the interpretation performance of this integrated SAR interpretation system. The integrated network can give out correct labels of candidate regions.

4.4. Analysis on Detection and Recognition Performance of Large Scene Images

In Figure 12, it is obvious that all 14 of these targets are well surrounded by boxes, and there is only one box around each target. This means that all other predicted boxes with lower scores are suppressed, and predicted boxes in adjacent slices are suppressed as well by NMSR. Although one BMP2 in the first line is recognized as T72, results verify that CNN with NMSR performs well dealing with large scene images of a simple background.

The results illustrate that this CNN network that realizes integration of target detection and recognition works well; all 14 targets of the three different kinds in a complex large scene SAR image are detected correctly. In addition, all degrees of classification confidence labeled in Figures 12 and 13

are over 0.97. No trees or bushes are interpreted as targets, which have proved the effect and usefulness of features extracted by CNN. Features among these three kinds of targets are well extracted and used to realize interpretation. There is only one box surrounding each target in this large scene SAR image, even though many of these targets appear on more than two slices. This result has proved that NMSR is a useful strategy dealing with predicted boxes among adjacent slices.

Thus, the results above have verified that this CNN network with NMSR has a satisfactory performance dealing with large scene SAR images regardless imaging background.

4.5. Analysis on Comparison Experiments

It can be seen from Tables 4 and 5 that our model can detect all targets in large scene images accurately and the performance of this model is not influenced by a change of background. This means that this fast sliding strategy could find all targets in different places, and targets appear at least one slice completely as we expected. As for results of recognition, since CNN extracts specific and effective features of SAR targets, and these features are used to realize detection and recognition at the same time, the best performance of both detection and recognition is achieved with our model.

5. Conclusions

Connecting the detection and recognition process to interpret large scene SAR images is difficult because of speckle noise and inefficient connection among these processes. Inspired by great success of deep convolutional neural networks, methods of DCNN are applied to SAR image interpretation to extract features automatically, and a method to integrate detection and recognition of large scene SAR images based on non-maximum suppression between regions (NMSR) is proposed in this paper. A model that is efficient in SAR image interpretation is built and a trained model that can generate a variety of accurate predicted boxes with confidence is obtained. Then, the performance of this system is evaluated, and 94.67% of three-class recognition accuracy on the MSTAR data set proved that it is efficient with high accuracy. Experiments on 1476×1784 simulated MSTAR images of a simple scene and complex scene show that the proposed method can recognize all targets with higher accuracy and faster speed, compared with the other methods, such as CFAR+SVM, Visual Attention+SVM, and Sliding-RCNN.

In the future, optimization algorithms should be researched to reduce the training time and test time. Other structures of deep learning networks can also be considered to implement real-time detection and recognition systems for SAR.

Author Contributions: Z.C. and S.D. put forward the method and designed the experiments. Z.C., S.W., and N.L. contributed to finishing experiments. All authors contributed to analysing experimental results and writing the paper.

Acknowledgments: This study is supported by the Fundamental Research Funds for the Central Universities No. 2672018ZYGX2018J013, the National Nature Science Foundation of China under Grant U1433113.

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

References

1. Geng, J.; Fan, J.; Wang, H. High-resolution SAR image classification via deep convolutional autoencoders. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2351–2355. [CrossRef]
2. Pei, J.; Huang, Y.; Huo, W.; Zhang, Y.; Yang, J.; Yeo, T.-S. SAR Automatic Target Recognition Based on Multiview Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2196–2210. [CrossRef]
3. Liu, H.; Yang, S.; Gou, S.; Zhu, D.; Wang, R.; Jiao, L. Polarimetric SAR Feature Extraction With Neighborhood Preservation-Based Deep Learning. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 1456–1466. [CrossRef]
4. Zhao, Q.; Principe, J.C.; Brennan, V. Synthetic aperture radar automatic target recognition with three strategies of learning and representation. *Opt. Eng.* **2000**, *39*, 1230–1245. [CrossRef]

5. Song, S.; Xu, B.; Yang, J. SAR target recognition via supervised discriminative dictionary learning and sparse representation of the SAR-HOG feature. *Remote Sens.* **2016**, *8*, 683. [CrossRef]
6. Lay, O.P.; Dubovitsky, S.; Peters, R.D. MSTAR: A submicrometer absolute metrology system. *Opt. Lett.* **2003**, *28*, 890–892. [CrossRef] [PubMed]
7. Novak, L.M.; Gregory, J.O.; William, S.B. Performance of 10-and 20-target MSE classifiers. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 1279–1289. [CrossRef]
8. English, R.A.; Rawlinson, S.J.; Sandirasegaram, N.M. ATR workbench for automating image analysis. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery X, Orlando, FL, USA, 12 September 2003; pp. 349–358. [CrossRef]
9. Wei, G.; Qi, Q.; Jiang, L.; Ping, Z. A New Method of SAR Image Target Recognition based on AdaBoost Algorithm. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Boston, MA, USA, 7–11 July 2008. [CrossRef]
10. Tison, C.; Pourthie, N.; Souyris, J.C. Target recognition in SAR images with Support Vector Machines (SVM). In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007. [CrossRef]
11. Wang, Y.; Han, P.; Lu, X.; Wu, R.; Huang, J. The Performance Comparison of Adaboost and SVM Applied to SAR ATR. In Proceedings of the International Conference on Radar, Shanghai, China, 16–19 October 2006. [CrossRef]
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
13. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Science, Santiago, Chile, 7–13 December 2015. [CrossRef]
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
16. Liu, W.; Anguelov, D.; Erhan, D. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37. [CrossRef]
17. Morgan, D.A.E. Deep convolutional neural networks for ATR from SAR imagery. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXII, Baltimore, MA, USA, 13 May 2015. [CrossRef]
18. Malmgren-Hansen, D.; Nobel-J, M. Convolutional neural networks for SAR image segmentation. In Proceedings of the 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Abu Dhabi, United Arab Emirates, 7–10 December 2015; pp. 231–236. [CrossRef]
19. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional Neural Network With Data Augmentation for SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [CrossRef]
20. Chen, S.; Wang, H.; Xu, F.; Jin, Y. Target Classification Using the Deep Convolutional Networks for SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [CrossRef]
21. Bridle, J.S. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. *Neurocomputing* **1990**, *8*, 227–236. [CrossRef]
22. Papson, S.; Narayanan, R.M. Classification via the Shadow Region in SAR Imagery. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 969–980. [CrossRef]
23. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. *arXiv* **2014**, arXiv:1311.2901.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [CrossRef]
26. Lecun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256. [CrossRef]
27. Arbib, M.A. *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1998.

28. Huang, G.; Haroon, A.B. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Trans. Neural Netw.* **1998**, *9*, 224–229. [CrossRef] [PubMed]
29. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
30. Hara, K.; Saito, D.; Shouno, H. Analysis of function of rectified linear unit used in deep learning. In Proceedings of the 2015 International Joint Conference on International Joint Conference on Neural Networks, Killarney, Ireland, 12–17 July 2015. [CrossRef]
31. Dong, L.; Wei, F.; Tan, C.; Tang, D. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), Baltimore, MD, USA, 23–25 June 2014; pp. 49–54. [CrossRef]
32. Kang, M.; Ji, K.; Leng, X. Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection. *Remote Sens.* **2017**, *9*, 860. [CrossRef]
33. Yuan, X.; Tang, T.; Xiang, D.; Li, Y.; Su, Y. Target recognition in SAR imagery based on local gradient ratio pattern. *Int. J. Remote Sens.* **2014**, *35*, 857–870. [CrossRef]
34. An, W.; Xie, C.; Yuan, X. An improved iterative censoring scheme for CFAR ship detection with SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 4585–4595. [CrossRef]
35. Shuo, L.; Zongjie, C. SAR image target detection in complex environments based on improved visual attention algorithm. *EURASIP J. Wirel. Commun. Netw.* **2014**, *1*, 125–138. [CrossRef]