remote sensing

Article

Ship Detection under Complex Backgrounds Based on Accurate Rotated Anchor Boxes from Paired Semantic Segmentation

Xiaowu Xiao, Zhiqiang Zhou *, Bo Wang, Linhao Li and Lingjuan Miao

School of Automation, Beijing Institute of Technology, Beijing 100081, China; 3120170438@bit.edu.cn (X.X.); wangbo@bit.edu.cn (B.W.); lilinhao@bit.edu.cn (L.L.); miaolingjuan@bit.edu.cn (L.M.)

* Correspondence: zhzhzhou@bit.edu.cn

Received: 9 September 2019; Accepted: 22 October 2019; Published: 26 October 2019

Abstract: It is still challenging to effectively detect ship objects in optical remote-sensing images with complex backgrounds. Many current CNN-based one-stage and two-stage detection methods usually first predefine a series of anchors with various scales, aspect ratios and angles, and then the detection results can be outputted by performing once or twice classification and bounding box regression for predefined anchors. However, most of the defined anchors have relatively low accuracy, and are useless for the following classification and regression. In addition, the preset anchors are not robust to produce good performance for other different detection datasets. To avoid the above problems, in this paper we design a paired semantic segmentation network to generate more accurate rotated anchors with smaller numbers. Specifically, the paired segmentation network predicts four parts (i.e., top-left, bottom-right, top-right, and bottom-left parts) of ships. By combining paired top-left and bottom-right parts (or top-right and bottom-left parts), we can take the minimum bounding box of these two parts as the rotated anchor. This way can be more robust to different ship datasets, and the generated anchors are more accurate and have fewer numbers. Furthermore, to effectively use fine-scale detail information and coarse-scale semantic information, we use the magnified convolutional features to classify and regress the generated rotated anchors. Meanwhile, the horizontal minimum bounding box of the rotated anchor is also used to combine more context information. We compare the proposed algorithm with state-of-the-art object-detection methods for natural images and ship-detection methods, and demonstrate the superiority of our method.

Keywords: convolutional neural networks; paired semantic segmentation; ship detection; magnified convolutional features; context information

1. Introduction

Ship detection in high-resolution optical remote-sensing images has been widely used in military and civilian fields. Accurate and efficient ship detection can benefit many practical applications such as marine traffic management, harbor dynamic surveillance, and maritime management, and so on. However, there also exist many difficulties for accurate ship detection. For example, ships can be in various shapes, sizes, colors, and orientations. It is very hard to recognize ships from some complex backgrounds. Ships docked very closely can be easily missed by many detection methods.

In recent decades, hand-designed features have usually been used to perform accurate ship detection [1–7]. However, this type of feature cannot adapt well to complex and changeable background environments in optical remote-sensing images. In recent years, convolutional neural networks (CNN)-based object-detection methods for natural images show great advantages in both detection accuracy and running efficiency. Generally, there are two kinds of CNN-based object-detection approaches, i.e., two-stage detectors and one-stage detectors. The two-stage detectors first predefine some default anchors (referring to some rectangles with various sizes) with different scales and aspect ratios to generate horizontal region proposals, and then the region proposals are classified and regressed to more accurately locate objects [8,9]. The one-stage detectors also define some anchors, and then these anchors are used to directly predict the final detection results [10–12]. Generally, the two-stage detection methods can produce better detection performance than one-stage methods, while one-stage detectors are usually faster and simpler [13].

Based on the detection methods for natural images, many CNN-based ship detectors are proposed to improve the ship-detection performance in optical remote-sensing images. However, the horizontal bounding boxes used for detecting objects in natural images are very easy to miss out some inclined ships docked side by side [14–18], and these ships are very common in remote-sensing images. Thus, many CNN-based ship-detection methods propose to use rotated bounding boxes to locate ships in remote-sensing images. Specifically, rotated bounding boxes are predicted from rotated anchors for one-stage detectors [14,17], and from rotated region proposals for two-stage detectors [16].

Both object detectors for natural images and ship detectors for remote-sensing images perform accurate object (ship) detection by predefining some default anchors. However, these anchor-based detectors may have some drawbacks:

(1) The detectors usually take each pixel in convolutional feature maps as center point to define anchors. Moreover, in order to robustly detect objects with various sizes and shapes, multiple scales and aspect ratios (and angles for ship detectors) are usually needed to be set, so that the defined anchors are able to adapt as many objects in real scenarios as possible. In this way, many anchors would be generated. However, objects usually only occupy a small area of an image and their numbers are relatively small, especially for the optical remote-sensing image. As a result, massive anchors are useless and highly overlap with each other. In this case, non-maximum suppression (NMS) needs to perform to reduce redundancy, which would inevitably increase computational cost.

(2) The generated anchors by predefining some scales, aspect ratios (and angles) are usually only applicable to one or several particular object-detection datasets. When the sizes and shapes of objects or image sizes in a new data set have relatively large difference compared with original data set, it is necessary to reset the values of scales, aspect ratios of anchors for adapting new object-detection dataset. Thus, researchers need to pay more attention to the definition of anchors.

To avoid the drawbacks from the previous anchor-based detectors, in this paper, we propose a novel anchor generation algorithm. The proposed method can generate anchors with smaller number and higher accuracy, and does not need to readjust the parameters of anchors when the detector is applied to other new data sets. We first design an encoder-decoder network to perform paired semantic segmentation for per-pixel prediction. Through segmentation, we predict four parts of ships, i.e., top-left part, down-right part, top-right part, down-left part. By combining paired top-left and bottom-right parts (or top-right and bottom-left parts), we can generate a rotated minimum bounding rectangle. Taking this kind of rectangle as anchor would greatly reduce the number of anchors and increase their accuracy. Since the anchors are directly produced from the segmentation results, there are a few hyper parameters that need to be set. In addition, we output magnified convolutional features from the decoder of segmentation network to effectively use the high-resolution detail information and the low-resolution semantic information, which is good for the detection of multi-scale ship objects. Finally, we use the horizontal minimum bounding box of the rotated anchor to extract convolutional features for the following classification and regression of the anchors, which can extract not only ship object features, but its context features. In this way, the feature extraction process can be simplified, and the detection performance is able to be improved. Experimental results demonstrate that the proposed method can accurately locate ships in remote-sensing images with complex backgrounds.

2. The Proposed Method

Figure 1 shows network architecture of the proposed ship-detection method. An encoder-decoder semantic segmentation network is first employed to predict the four parts (i.e., top-left, bottom-right, top-right, bottom-left parts) of each ship (In Figure 1, we only show top-left part and bottom-right part). Then, using segmentation results, a small number of rotated anchors with higher accuracy can be generated. To combine both fine-scale detail information and coarse-scale semantic information, so that the generated anchors can be classified and regressed more accurately, unlike some common object-detection methods [9,16], we take advantage of magnified convolutional features to implement classification and regression for the generated anchors. Furthermore, the horizontal minimum bounding box of the rotated anchor is used to extract magnified convolutional features for each anchor in RoIPooling. This can extract more context information and is simpler. Finally, the extracted features are fed into two fully connected layers (fc) to output classification results and bounding box regression results for the rotated anchors.

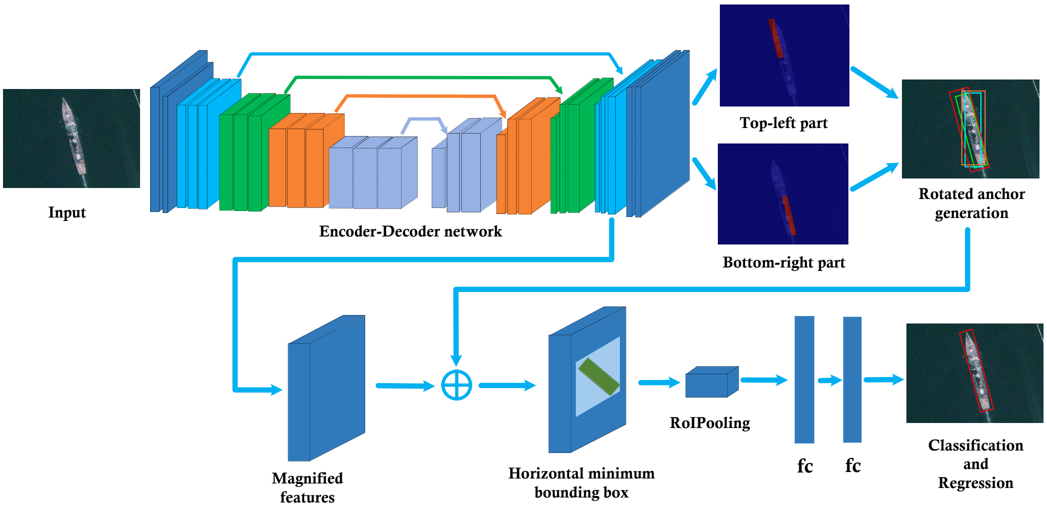


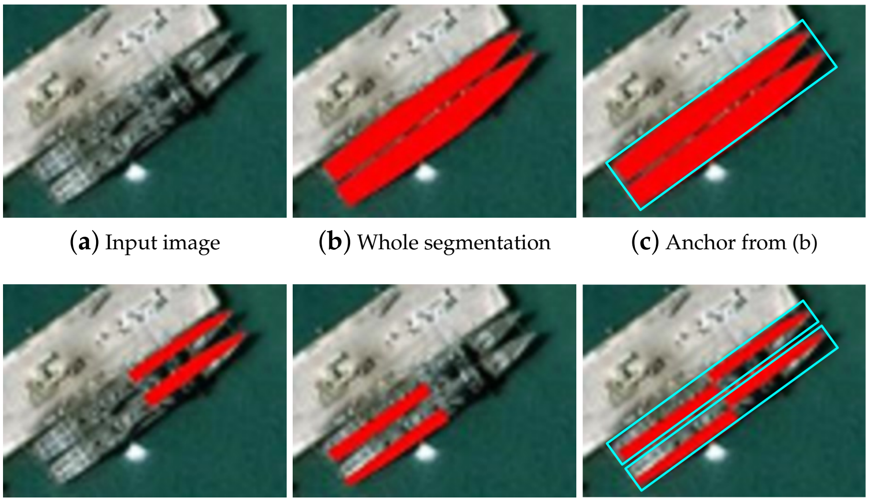
Figure 1. The network architecture of the proposed ship-detection method.

2.1. *Encoder-Decoder Network for Paired Semantic Segmentation*

The early CNN-based object-detection methods first uses some traditional proposal generation algorithms [19–23] to extract region proposals, and then train a CNN model for classification [24–26] (and regression [15,27]) of each proposal. These proposal generation methods are usually time-consuming and in this way, the whole detection method cannot be trained and tested end-to-end. Therefore, Faster R-CNN [9] designs an anchor-based region proposal network (RPN) to extract region proposals. RPN uses each point of the feature map as center point to preset many anchors with different sizes, and region proposals can be predicted from the predefined anchors with CNN. With RPN, the detector can perform end-to-end training and testing, and the detection accuracy is also improved. Since then, most of the two-stage detectors [8,28,29] and the one-stage detectors [10,11,30] (including ship detectors [14,16]) introduce anchor strategy, and produce promising detection performance. However, these anchor-based detectors may suffer from some limitations [31,32].

First, multiple fixed scales, aspect ratios (and angles for ship detection) need to be preset for anchor generation, to accurately detect objects (ships) with various sizes and shapes. To produce accurate and relatively few anchors, many validation experiments would be necessary, which is tedious and time-consuming. Even if many experiments are carefully performed, it is still very hard to find optimal setting for anchor generation. Secondly, the sizes and shapes of objects in different detection datasets can be quite different. The carefully designed anchors for a dataset are usually inappropriate for another dataset. Finally, since anchors are extracted from each pixel of the feature map, a large number of anchors would be generated in an image for accurate object detection. In this case, NMS is usually used to reduce overlaps between anchors and select more accurate anchors; however, this process is time-consuming.

In this paper, we propose to perform semantic segmentation for rotated anchor generation. Without carefully designed experiments for setting scales, aspect ratios, and angles, we can generate anchors with higher accuracy and smaller number. In addition, the proposed method does not need to reset anchors for a new dataset. For generating anchors via semantic segmentation, a direct and natural idea is to segment the whole ship in the remote-sensing image, and then we can take the rotated minimum bounding box of ship segmentation result as the anchor. However, the ships docked side by side are very common in optical remote-sensing images, and Figure 2a,b show these kinds of ships and their segmentation results, respectively. It can be seen that the generated anchor (Figure 2c) from these segmentation results is inaccurate. To solve this problem, we propose to perform paired segmentation for rotated anchor generation. The paired segmentation means to segment the top-left part, the bottom-right part, the top-right part, and the bottom-left part of a ship. Figure 2d,e show a pair of segmentation results, i.e., the top-left and the bottom-right segmentation of ships. As shown in Figure 2f, through combining these two paired parts, we can generate more accurate anchors by computing the minimum bounding boxes of the paired segmentation parts.


(**a**) Input image (**b**) Whole segmentation (**c**) Anchor from (b)

(**d**) Top-left segmentation (**e**) Bottom-right segmentation (**f**) Anchors from (d) and (e)

Figure 2. The whole ship segmentation result (**b**) would produce inaccurate anchor box (see (**c**)). Using top-left and bottom-right segmentation parts (see (**d**,**e**)) would generate more accurate anchors (see (**f**)).

In recent years, using fully convolutional neural networks to implement semantic segmentation has attracted much attention. Among them, the encoder-decoder networks (a kind of fully convolutional neural networks) [33–36] achieve state-of-the-art performance for semantic segmentation. The encoder-decoder network consists of the encoder and the decoder. The encoder extracts convolutional features and downsamples the input image with a range of convolutional and pooling layers. The role of the decoder is to map the low-resolution encoder features to input-resolution features with a series of deconvolutional and convolutional layers, and output dense pixel-wise classification. The architectures of encoder and decoder can be symmetric [34,37] or asymmetric [38–41].

In this paper, we design an asymmetric encoder-decoder semantic segmentation network for paired ship segmentation. Figure 3 shows the architecture of encoder-decoder network, and the left part shows encoder, while the right part is decoder. As shown in Figure 3, "conv" denotes a convolutional layer followed by a batch normalization layer [42] and a ReLU activation function [43]. "7 × 7" or "3 × 3" are the kernel size. "$(64, 64)$" denotes the channels of input and output, respectively. "s2" denotes that the stride is 2 and "s1" denotes that the stride is 1. "deconv" denotes a deconvolutional layer (The deconvolution is also called transposed convolution [35], and responsible to upsample the feature map.) [36] followed by a batch normalization layer and a ReLU activation function.

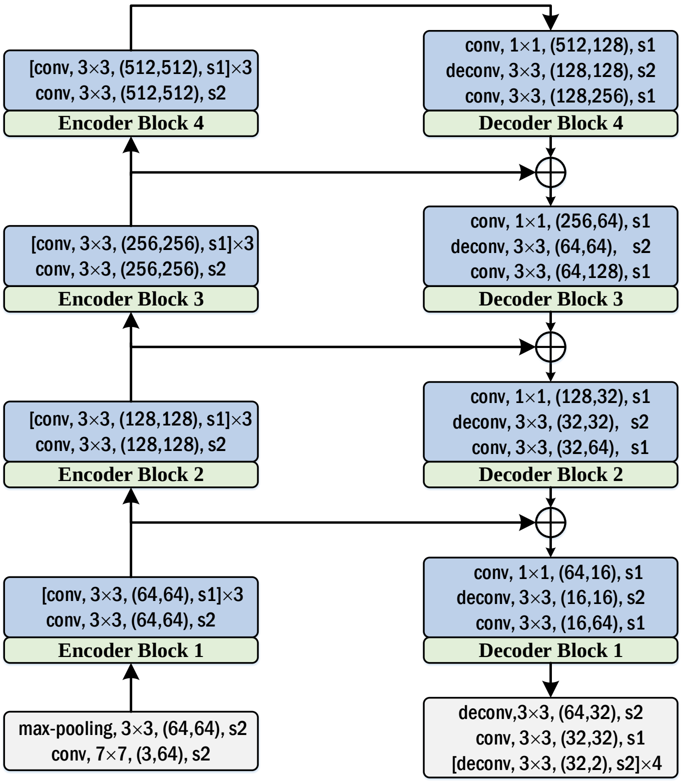


Figure 3. The encoder-decoder semantic segmentation network architecture.

Encoder: Trading off between inference efficiency and segmentation accuracy, we use ResNet18 [44] as our encoder to extract convolutional features. The ResNet18-based encoder first reduces the input resolution to 1/4 of the original size using a convolutional layer and a max-pooling layer. Then, four encoder blocks are added to extract input features and perform downsampling. Each block includes two shortcut connections as described in [44].

Decoder: Performing deconvolution operation to upsample feature maps is relatively time-consuming, especially for large input and output feature maps. To reduce computational cost of deconvolution operation as much as possible, in each decoder block, the input channel is first reduced 4 times with the 1 × 1 convolution operation. Then, the deconvolution operation is performed on the reduced features. Finally, a 3 × 3 convolutional layer is used to double the channel of feature maps. Through four decoder blocks, the feature maps are upsampled to 1/4 of input resolution. After that, we add a deconvolutional layer followed by a convolutional layer and four deconvolutional layers. The final four deconvolutional layers are parallel, respectively predict four paired segmentation parts of ships (i.e., top-left part, bottom-right part, top-right part, bottom-left part). As shown in Figure 3, we link the output of each encoder block to the input of its corresponding decoder block, to recover lost spatial information due to downsampling.

The encoder-decoder paired ship segmentation network has four outputs, which predict four parts of ships, respectively. There are two categories (denoted as 0 and 1) in each output. For example, for the ship in Figure 4a, the ground-truth segmentation label of the output for the top-left part is shown in Figure 4b, in which we take the pixels in top-left part of the ship (denoted as red color) as positive samples (i.e., 1), and the pixels in other areas (denoted as blue color) as negative samples (i.e., 0). The top-left part segmentation label Figure 4b can be obtained from the whole ship segmentation label. However, compared with the detection label (a rectangle determined by four points), it is much difficult to annotate the segmentation label (an object area determined by a large number of points). Although there exist some modern segmentation annotation tools (e.g., LabelMe https://github.com/CSAILVision/LabelMeAnnotationTool), obtaining accurate segmentation label for a remote-sensing image dataset is still relatively time-consuming.

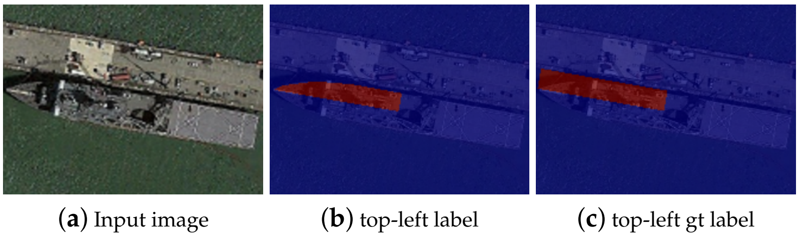

(**a**) Input image (**b**) top-left label (**c**) top-left gt label

Figure 4. For the input image (**a**), taking (**b**) as the top-left part segmentation label would result in time-consuming annotation process. To avoid this problem, as shown in (**c**), we propose to use ground-truth box of the ship to generate ground-truth (gt) segmentation label.

To reduce annotation work, we propose to use the ground-truth box of the ship to generate segmentation label for four-part prediction. As shown in Figure 4c, we take the pixels in the top-left part of the ground-truth box of the ship as the positive samples, and those in the other areas as negative samples. This can be reasonable due to the two facts. First, since ships are usually in narrow-rectangle shapes, the two kinds of segmentation labels (Figure 4b,c) have almost the same accuracy. Secondly, the paired ship segmentation aims to generate rotated anchors, and not to produce final detection results. Hence, a segmentation label with a bit lower accuracy is acceptable. Through using the ground-truth box to generate segmentation label, we can save a lot of human resources without reducing detection accuracy.

From Figure 4c, we can see that positive and negative samples are much imbalanced. To avoid the class-imbalanced problem, we employ weighted cross-entropy loss function [39,45,46] to train the paired semantic segmentation network. The loss function L_p for the whole paired segmentation network is defined as

$$L_p = L_{tl} + L_{br} + L_{tr} + L_{bl}, \tag{1}$$

in which L_{tl}, L_{br}, L_{tr}, L_{bl} are four weighted cross-entropy loss functions for top-left segmentation, bottom-right segmentation, top-right segmentation, and bottom-left segmentation, respectively. L_{tl} is defined as

$$L_{tl}\left(p,u,v\right) = -\frac{|Y_-|}{|Y_+|}\sum_{u\in Y_+}\log p_u - \sum_{v\in Y_-}\log p_v, \tag{2}$$

in which Y_+ is positive sample set and Y_- is negative sample set in top-left segmentation, and $|Y_+|$ and $|Y_-|$ are numbers of Y_+ and Y_-, respectively. The probability p_u denotes the confidence score that the pixel u is classified as the positive sample, and the probability p_v is the confidence score that the pixel v is classified as the negative sample. L_{br}, L_{tr}, L_{bl} have similar definitions to L_{tl}.

2.2. Rotated Anchor Generation

Using encoder-decoder semantic segmentation network, we can get four segmentation parts of ships, i.e., top-left part, bottom-right part, top-right part, and bottom-left part. In this section, we detail the strategy for generating rotated anchors by combining top-left and bottom-right parts of ships. This kind of strategy is also applicable to top-right and bottom-left parts. Figure 5a,b show the top-left segmentation result and the bottom-right segmentation result of a remote-sensing image, respectively. Please note that for better display, the segmentation result (only containing 0 and 1) is shown on the input image, and the pixels in the red regions denote positive samples (i.e., 1) and the pixels in other regions are negative samples (i.e., 0).

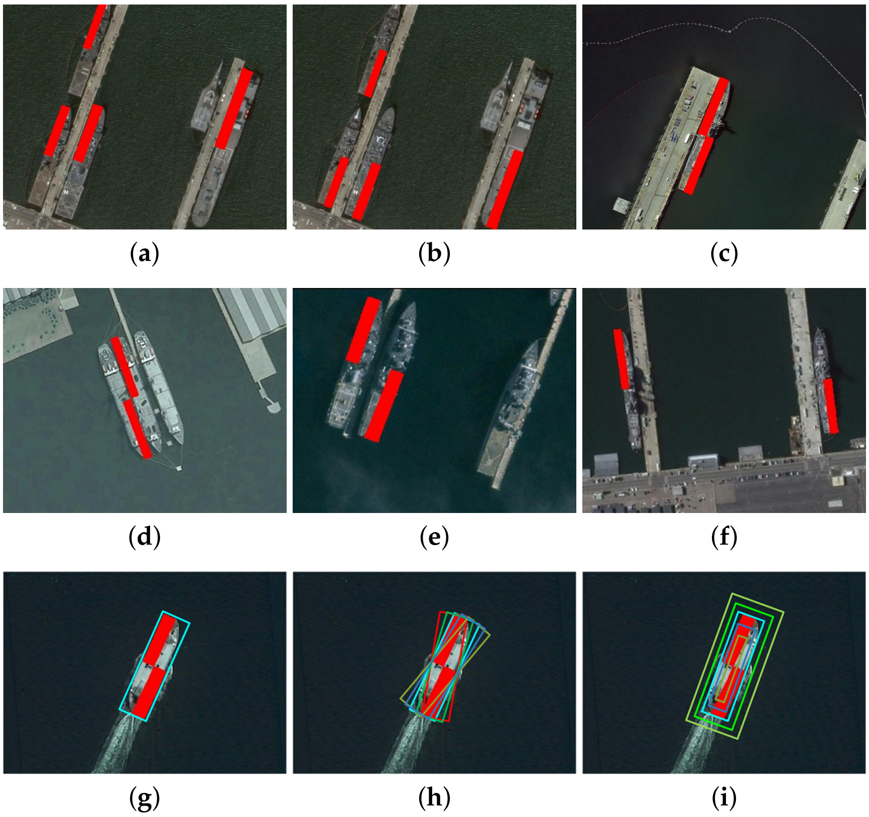


Figure 5. The illustration for rotated anchor generation. (**a**,**b**) are the top-left and bottom-right segmentation results, respectively. (**c**–**f**) show four different combination cases of the connected regions in the two segmentation results. (**g**) The initial rotated anchor can be generated via computing the minimum bounding box of the two connected regions. In (**h**,**i**), rotation and scaling are respectively performed to increase the number of the rotated anchor.

The strategy for generating rotated anchors mainly includes four steps:

(1) We regard each red region in segmentation results (Figure 5a,b) as a connected region. Please note that we regard predicted each top-left part (see Figure 5a) or each bottom-right part (see Figure 5b) as a connected region, and Figure 5c–f show the different combination cases of two connected regions.

(2) Each connected region in the top-left segmentation result (Figure 5a) is combined with each connected region in the bottom-right segmentation result (Figure 5b). There mainly exist four different combination cases for two connected regions from different segmentation results (see Figure 5c–f). When the shortest distance between the pixels in the two connected regions is smaller than a threshold d, we deem that these two connected regions are from the same ship, and these two regions should be reserved (see Figure 5c). When the distance is smaller than d, there may exist another case (see Figure 5d). In this case, although two connected regions are not from the same ship, since the ships are docked very closely, we still reserve these two regions for higher recall rate. When the shortest distance is larger than threshold d (see Figure 5e,f), we abandon these two connected regions. In our experiments, $d \in [5, 15]$ can be satisfactory.

(3) The initial rotated anchor can be generated via computing the rotated minimum bounding box of the reserved two connected regions (see Figure 5g).

(4) To improve the recall rate of rotated anchors, as shown in Figure 5h, we rotate the initial anchor with five small angle values (i.e., $-4^\circ, -2^\circ, 0^\circ, +2^\circ, +4^\circ$). In this way, each initial anchor would produce four other anchors around the same ship. Then, we scale all these anchors with five factors (i.e., 0.6, 0.8, 1.0, 1.2, 1.4) to generate more rotated anchors (see Figure 5i).

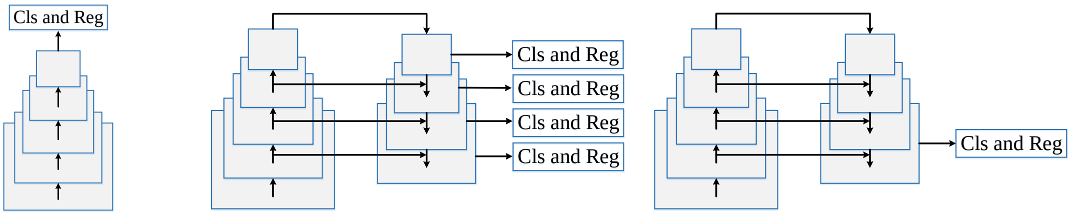
About 20 pairs of connected regions can be reserved in a remote-sensing image, and thus we can generate 20 initial rotated anchors. Through step (4), an optical remote-sensing image would generate about 500 (20 × 5 × 5) rotated anchors.

2.3. RoIPooling Based on Magnified Convolutional Features and Context Information

RoIPooling aims to extract a fixed-size feature for each anchor from one or several convolutional layers, and then the extracted feature is used for classification and regression of each rotated anchor to produce final detection results. In this paper, we propose to extract the feature containing context information from the magnified (upsampled) convolutional layers. In this way, we can effectively improve the ship-detection accuracy, especially for small ship object.

2.3.1. Magnified Convolutional Features

Generally, RoIPooling often extracts single-scale [9,47] or multi-scale [8,48] features for classification and regression of each anchor. As shown in Figure 6a, the single-scale features are usually obtained from the last layer of some base convolutional networks (e.g., VGG16 [49] and ZF [50]). The last layer is always acquired through many downsampling operations. In this case, the single-scale features would contain relatively more semantic information, while lacking detail information that is beneficial to small object detection. As a result, only using single-scale features leads to low detection accuracy for small ship objects. On the other hand, the multi-scale features are from multiple convolutional layers (see Figure 6b), which effectively combine high-resolution detail information and low-resolution semantic information. Thus, this can accurately detect ship objects with different sizes. However, the process obtaining multi-scale features and performing classification and regression many times may be tedious and time-consuming.



(**a**) Single-scale features (**b**) Multi-scale features (**c**) Magnified features

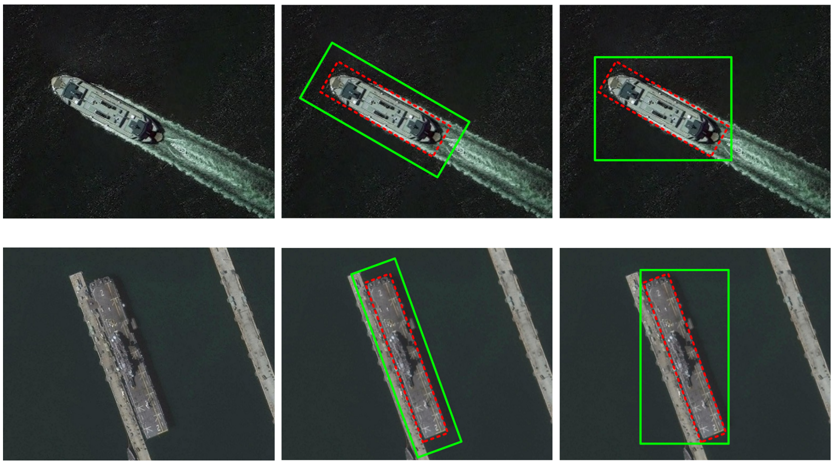
Figure 6. (**a**) The single-scale features are used to perform classification and regression (Cls and Reg). (**b**) The multi-scale features are used to perform classification and regression. (**c**) The magnified features are used to perform classification and regression.

Since we upsample the low-resolution features to recover lost detail information in decoder of segmentation network (see Figure 3), the convolutional layers in decoder already contain both semantic information and detail information. To make the proposed detection network simpler and faster without reducing detection accuracy, we only take advantage of a single-scale convolutional layer in decoder to perform classification and regression (see Figure 6c). Specifically, we add a 512-channel convolutional layer with kernel size 3 × 3 after the output of decoder block1 (see Figure 3). Then, we take the features from the added convolutional layer as magnified (upsampled) features to classify and regress the rotated anchors. In this simple way without using multi-scale features, we can effectively improve the detection performance of the multi-scale ship objects.

2.3.2. Context Information

RoIPooling consists of two stages. In the first stage, we extract the features in the area covered with the rotated anchor box from the magnified convolutional layer. The extracted features are then implemented max-pooling operation to produce fixed-size feature representations in the second stage.

It is worth noting that context information is good for ship object detection. For the ship in optical remote-sensing images, context information includes sea, harbor, ship, dock, building, and so on. To make extracted features contain more context information, one way is to directly enlarge rotated anchor box, and then the features containing context information can be obtained by using enlarged anchor box. As shown in Figure 7b, the red rectangle denotes the original rotated anchor, and the green rectangle is the enlarged anchor box, which contains more context information. With the help of context information, the detection network can recognize the ship object easier.



(**a**) Input image (**b**) Enlarged anchor (**c**) Horizontal rectangle

Figure 7. (**a**) is the input image. In (**b**), enlarged rotated anchor box (the green rectangle) is used to extract the features containing context information. However, this way is relatively complicated. In this paper, we use horizontal minimum bounding box (the green rectangle in (**c**)) of the original rotated anchor (the red rectangle in (**c**)) to extract features containing context information.

However, compared with horizontal anchor box, extracting features for rotated anchor box is more complicated [51]. To both reduce computation and use context information, in this paper, we propose to take advantage of horizontal minimum bounding box of rotated anchor box instead of using the enlarged anchor box. As shown in Figure 7c, the green rectangle denotes horizontal minimum bounding box. We can see that this kind of box can also effectively cover more context information, and extracting convolutional features based on the horizontal box can be simpler and faster.

2.4. Loss Function and Training Strategy

2.4.1. Loss Function

The rotated anchor is assigned as the positive sample (ship) if its Intersection-over-Union (IoU) overlap with any ground-truth box is more than 0.5. In this way, some ground-truth boxes may have no matched anchors. To avoid this problem, we also assign the anchor with the highest IoU overlap with a ground-truth box as a positive sample. The rest anchors are set as the negative samples (background).

RoIPooling outputs a fixed-size convolutional feature (512 × 7 × 7) for each rotated anchor. Then, the features are fed into two fully connected layers to produce final classification result (ship or background) and bounding box regression result (location of ship) of each anchor. Each fully connected layer includes 1024 neurons followed by a ReLU activation function. Similar to many object-detection methods [9,16,27,51], we adopt log loss function (denoted as L_c) for classification and smooth-L_1 loss function (denoted as L_r) for regression. Hence, by combining paired semantic segmentation loss L_p, classification loss L_c and regression loss L_r, the total loss function L of the proposed detection network can be defined as

$$L = \lambda_p L_p + \lambda_c L_c + \lambda_r L_r, \tag{3}$$

in which λ_p, λ_c, λ_r are balancing parameters, and set as 1, 2, 2, respectively.

2.4.2. Assignment of Anchor

As described in Section 2.2, we generated about 500 rotated anchors. In the training stage, performing classification and regression for all anchors is undesirable. On the one hand, this would consume more training time. On the other hand, this can cause serious sample imbalance problem, because negative samples are much more than positive samples. Hence, we sample 32 positive samples, and make the ratio between positive samples and negative samples at 1:3. When the number of the positive sample is smaller than 32, we use the negative sample to make up the positive sample. In the testing stage, to reduce inference time, we sample 300 anchors to perform classification and bounding box regression.

2.4.3. End-to-End Training

The detection model is initialized with ResNet-18 pre-trained weights for ImageNet classification [44]. The proposed detection network is trained end-to-end using SGD optimizer [52] on a NVIDIA GTX 1080 Ti GPU. We set the initial learning rate to 0.001 and decrease the learning rate by 0.1 at 50k-th and 80k-th iterations, and the total iteration number is set as 100k. The momentum and weight decay is set to 0.9 and 5×10^{-5}, respectively. The network trains one image at each iteration. Our code is based on Pytorch [53].

3. Experiments

3.1. Dataset

The ship data set is collected from Google Earth. In the data set, there are 900 optical remote-sensing images, including 3584 ships. The image size is 1024 × 768. We generate ground-truth boxes for each image by using annotation tool LabelMe https://github.com/CSAILVision/LabelMeAnnotationTool. We randomly sample 300 optical remote-sensing images as the testing set, and the rest 600 images are set as the training set. Some images in the data set are shown in Figure 8. We can see that ships are in arbitrary orientations, and usually closely docked side by side (see Figure 8a–d). The background in some images can be very complex (see Figure 8e–f).

To reduce overfitting and improve detection accuracy, we perform data augmentation for the training set. Four data augmentation strategies are used in this paper. First, we implement horizontal flip and vertical flip for each image in the training set. Secondly, a gaussian filter with standard deviation of 2 is used to smooth the training images. Finally, we rotate each training image with five angles $(30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ)$. With these four strategies, we augment the training set from 600 images to 5400 images.

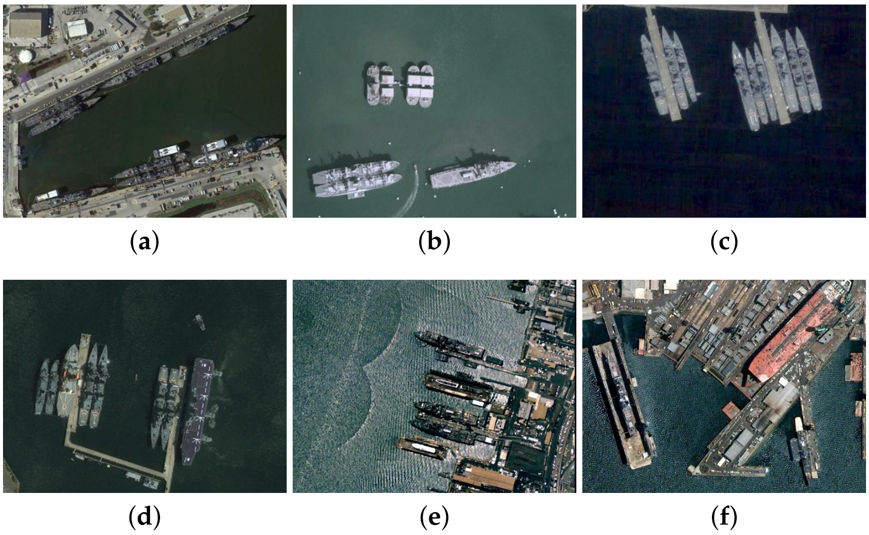


Figure 8. Some images in the ship data set. The ships are in arbitrary orientations, and usually closely docked side by side (see (**a**–**d**). The background in some images can be very complex (see (**e**–**f**)).

3.2. Compared Methods

The proposed detection method is compared with three other detection algorithms, including Faster R-CNN [9], SSD [10], and Rotation Dense Feature Pyramid Network (R-DFPN) [16].

Faster R-CNN: Faster R-CNN [9] is designed for detecting objects with horizontal bounding boxes. Faster R-CNN first uses VGG16 model [49] to extract convolutional features. Then, the features are fed into region proposal network to generate many region proposals. Finally, the features are used again to perform classification and bounding box regression for region proposals. The size of the input image of Faster R-CNN is set as 1024 × 768, and we adopt the default settings for other parameters. The code https://github.com/rbgirshick/py-faster-rcnn is implemented based on Caffe [54].

SSD: SSD [10] produces horizontal bounding boxes for objects. Taking advantage of multi-scale convolutional features, SSD directly produces detection results based on many preset anchors. The input image of SSD is resized into 512 × 512. The code of SSD https://github.com/weiliu89/caffe/tree/ssd is published with Caffe.

R-DFPN: R-DFPN [16] aims to detect arbitrary-oriented ships in optical remote-sensing images with rotated bounding boxes. R-DFPN proposes a dense feature pyramid network, which builds high-level semantic feature maps for all scales by means of dense connections, to effectively detect ships in various complex scenes. The input image is set as 1024 × 768. The authors run R-DFPN https://github.com/yangxue0827/R-DFPNFPNTensorflow based on Tensorflow [55].

3.3. Comparison

Figure 9 shows some ship-detection results from the four detection methods including Faster R-CNN [9], SSD [10], R-DFPN [16], and ours. From the first row to the fourth row, we can see that the images contain many closely docked ships side by side, and this case is very common in optical remote-sensing images. Since Faster R-CNN and SSD output the horizontal bounding box for the ship object, it is very easy to miss out some inclined closely docked ships [14–17]. As shown in the second column and the third column, these two methods produce very inaccurate detection results. R-DFPN outputs the rotated bounding box for the ship object, but the rotated anchors in R-DFPN are predefined using several fixed angle values, which leads to relatively inaccurate anchor boxes compared with our method. Hence, from the fourth column we can see that some rotated boxes are inaccurate and some ships are missed out. In contrast, our detection method (see the last column) gives more accurate locations for those closely docked ships. For the optical remote-sensing images in the last two rows, the backgrounds and the ship objects have very similar textures and appearances. In this case, our method outputs more robust detection results than other three detection methods.

We also test the proposed method on two large-scale optical remote-sensing images. We partition the large-scale image into multiple small-scale images of resolution 1024 × 768, and run each small-scale image using the proposed detection network to output the location of each ship. We use sliding-window strategy with overlap of 20% to partition large-scale images. Figure 10 shows a large-scale remote-sensing image with resolution 4356 × 2589, and the red boxes in the image denote the detection results from our method, and the yellow boxes denote the missed ship objects, and the green boxes indicate the wrong detections. We can see that the detection results are very accurate, and there only exist several missed and wrong bounding boxes. Figure 11 gives another large-scale image with resolution 2605 × 1475, and the image contains many closely docked ships. In this situation, the proposed method can still give satisfactory detection performance.

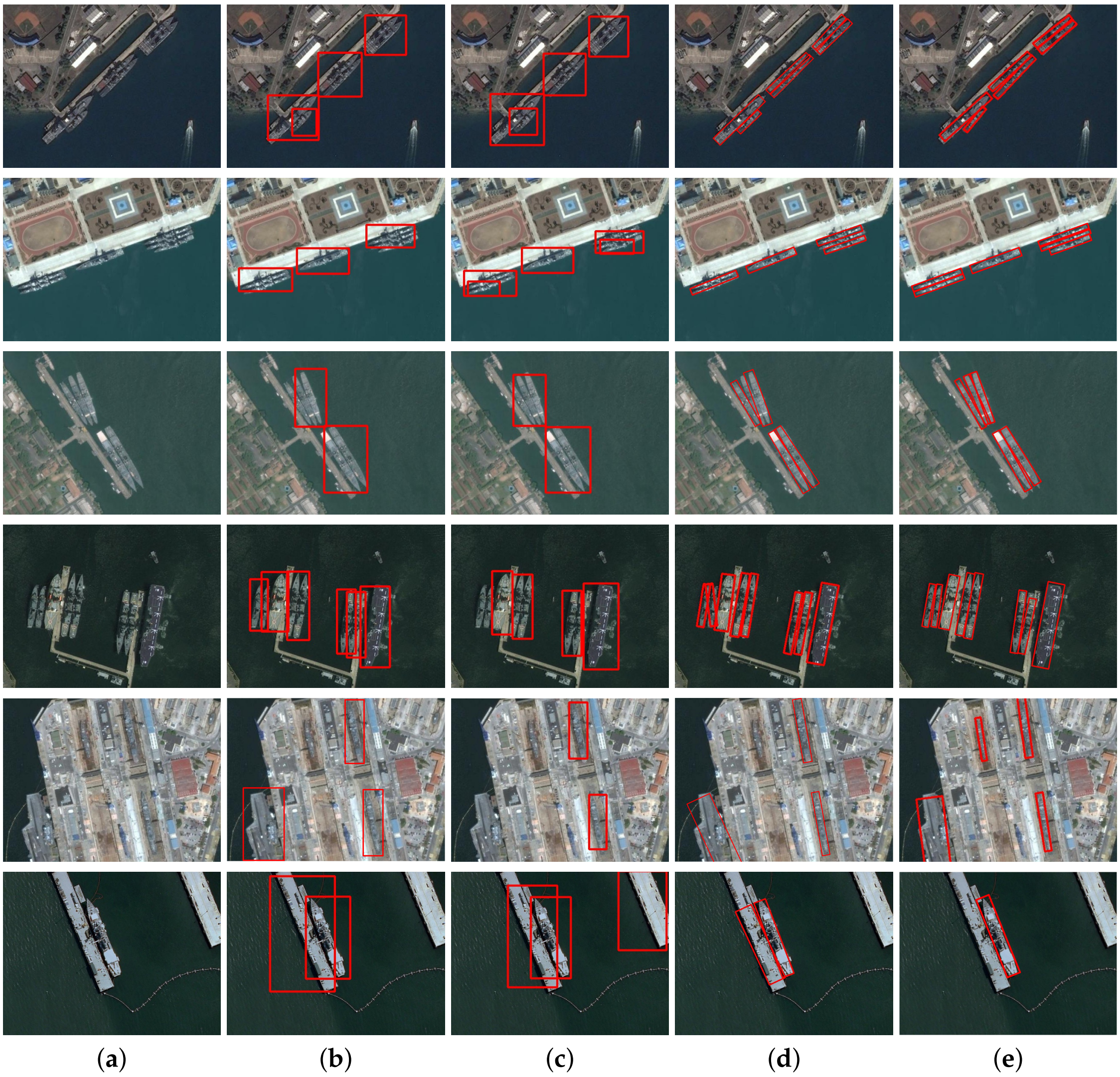


Figure 9. The ship-detection results from different methods. The images from the first row to the fourth row contain many closely docked ships. For the optical remote-sensing images in the last two rows, the backgrounds and the ship objects have very similar textures and appearances. Our method gives accurate detection results for all these cases. (**a**) Input; (**b**) Faster R-CNN [9]; (**c**) SSD [10]; (**d**) R-DFPN [16]; (**e**) Ours.

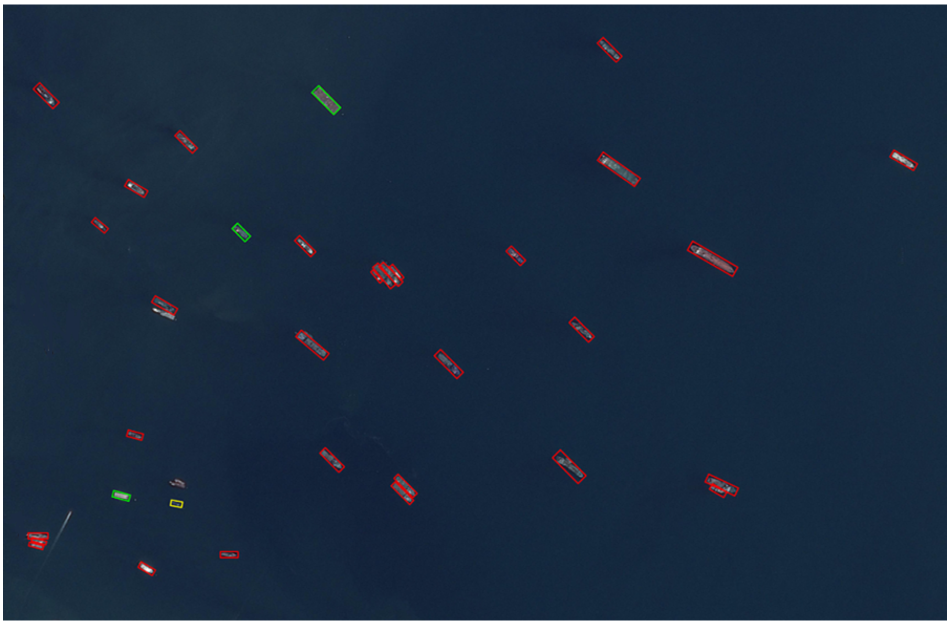


Figure 10. The detection results of one large-scale remote-sensing image (4356 × 2589). The red boxes denote the detection results from our method. The yellow boxes denote the missed ship objects. The green boxes indicate the wrong detections.

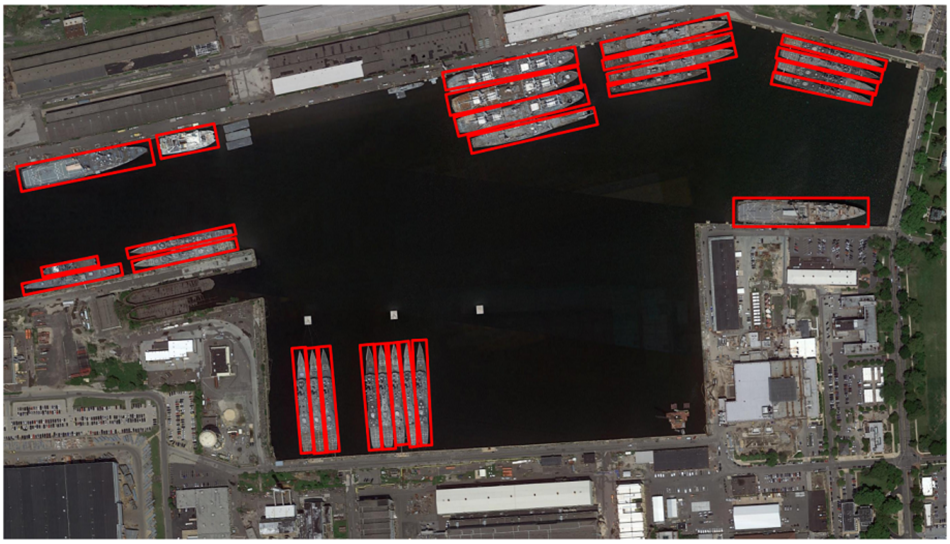


Figure 11. The detection results of one large-scale remote-sensing image (2605 × 1475).

We use Average Precision (AP) to quantitatively evaluate the performance of different detection algorithms. Generally, many detection methods usually set IoU of 0.5 to compute AP. To provide more comprehensive comparison in this paper, we compute and compare AP under different IoU values (i.e., 0.5, 0.6, 0.7) for the four detection methods. Figure 12 shows comparison of AP of different detection methods. The x-axis denotes the value of IoU, and the y-axis denotes the value of AP. We can see that our method is able to produce the highest AP under different IoU values, which indicates the superiority of the proposed method.

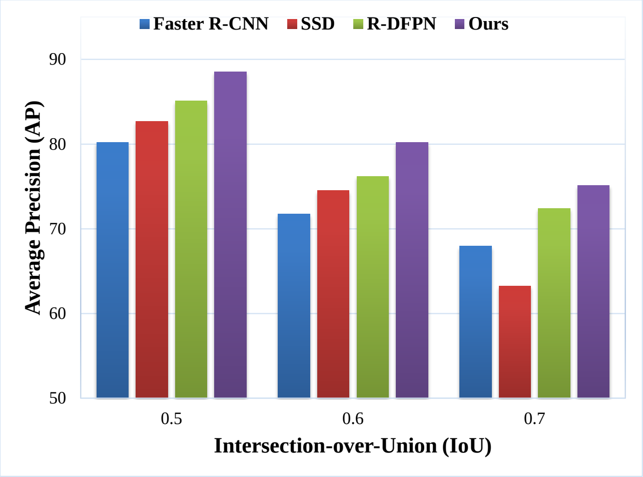


Figure 12. Comparison of AP under different IoU values from the four detection methods.

4. Discussion

4.1. Paired Semantic Segmentation

In paired semantic segmentation network, we predict four parts (i.e., top-left, bottom-right, top-right, bottom-left parts) of each ship to generate rotated anchor boxes. Please note that we can also only predict two parts (i.e., top-left, bottom-right parts, or top-right, bottom-left parts) for each ship. As shown in Table 1, the detection accuracy based on two parts is also relatively satisfactory. Predicting four parts can extract more useful information, and increase recall rate, resulting in better detection performance (see the last row in Table 1).

Table 1. The AP for different outputs of paired semantic segmentation network.

Output	AP
2 parts (top-left, bottom-right)	87.3%
2 parts (top-right, bottom-left)	87.1%
4 parts (top-left, bottom-right, top-right, bottom-left)	88.5%

4.2. Segmentation Label

As described in Section 2.1, to reduce annotation work, we use the ground-truth box of the ship to generate segmentation label instead of using real segmentation label. We test these two settings in Table 2. We can see that the detection accuracy from these two settings is almost the same, which indicates the effectiveness of the proposed strategy.

Table 2. The AP for different segmentation labels.

Setting	AP
Label based on box	88.5%
Real label	88.4%

4.3. Setting of Anchor

In Section 2.2, we set 5 angle values and 5 scale values to respectively rotate and resize anchors to improve the detection performance. We also test several other settings for rotating and resizing anchor boxes. A small number of anchors would decrease the recall rate, resulting in relatively low detection accuracy (see the second and the third rows of Table 3). On the other hand, when the number of the anchor becomes relatively large, the recall rate would be near saturation. In this situation, the detection accuracy would almost no longer increase (see the last two rows of Table 3). Thus, 5 angle values and 5 scale values are used in this paper.

Table 3. The AP for different settings of generating anchors.

Setting	AP
1 angle, 1 scale	71.4%
3 angles, 3 scales	82.7%
5 angles, 5 scales	88.5%
7 angles, 7 scales	88.5%

4.4. Magnified Convolutional Features

We use the output of the decoder block 1 to produce magnified convolutional features. Compared with other decoder blocks (i.e., 2,3,4), the decoder block 1 is through multiple upsampling operations (see Figure 3), and thus it contains more detail information and semantic information. The detail information is good for location of the small-scale object, and the semantic information is beneficial to the recognition of the large-scale object. Hence, the generated magnified features based on the decoder block 1 would lead to better detection performance (see Table 4).

Table 4. The AP for magnified convolutional features based on different decoder blocks.

Decoder Block	AP
Decoder Block 4	83.4%
Decoder Block 3	85.2%
Decoder Block 2	88.1%
Decoder Block 1	88.5%

5. Conclusions

In this paper, we propose a novel ship-detection method based on convolutional neural networks, which can accurately detect ships in optical remote-sensing images with complex backgrounds with rotated bounding boxes. To overcome the shortcomings of frequently used anchor generation methods, we design paired semantic segmentation network to predict the four parts (i.e., top-left, bottom-right, top-right, bottom-left parts) of each ship in the proposed detection network. With the predicted four parts, the generated rotated anchors can be more accurate and have smaller number. The proposed anchor generation method can also be more robust to different detection datasets. To effectively detect both large-scale and small-scale ship objects, the magnified convolutional features containing more detail information and semantic information are used to perform classification and regression of the anchor box. Furthermore, we propose to use the horizontal minimum bounding box of the rotated anchor to extract more context information, which is simpler and beneficial for accurately recognizing ships under complex backgrounds. Experiments demonstrate that our method can produce better detection performance compared with some state-of-the-art object-detection methods for natural images and ship-detection methods.

Author Contributions: Conceptualization, X.X.; methodology, X.X. and Z.Z.; software, X.X.; validation, X.X., Z.Z. and B.W.; investigation, B.W. and L.M.; resources, L.L. and L.M.; data curation, X.X.; writing–original draft preparation, X.X. and Z.Z.; writing–review and editing, X.X. and Z.Z.; visualization, B.W., L.L. and L.M.; supervision, B.W.

Funding: This research received no external funding.

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

References

1. Jiang, L.B.; Wang, Z.; Wei-Dong, H.U. An AIAC-based Inshore Ship Target Detection Approach. *Remote Sens. Technol. Appl.* **2007**, *22*, 88–94.
2. Lin, J.; Yang, X.; Xiao, S.; Yu, Y.; Jia, C. A Line Segment Based Inshore Ship Detection Method. In *International Conference on Remote Sensing*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 261–269.
3. Xu, J.; Fu, K.; Sun, X. An Invariant Generalized Hough Transform Based Method of Inshore Ships Detection. In Proceedings of the International Symposium on Image and Data Fusion, Tengchong, China, 9–11 August 2011; pp. 1–4.
4. Liu, G.; Zhang, Y.; Zheng, X.; Sun, X.; Fu, K.; Wang, H. A New Method on Inshore Ship Detection in High-Resolution Satellite Images Using Shape and Context Information. *IEEE Geosci. Remote Sens. Lett.* **2013**, *11*, 617–621. [CrossRef]
5. Tang, J.; Deng, C.; Huang, G.B.; Zhao, B. Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1174–1185. [CrossRef]
6. Jian, X.; Xian, S.; Zhang, D.; Fu, K. Automatic Detection of Inshore Ships in High-Resolution Remote Sensing Images Using Robust Invariant Generalized Hough Transform. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 2070–2074. [CrossRef]
7. Li, S.; Zhou, Z.; Wang, B.; Wu, F. A Novel Inshore Ship Detection via Ship Head Classification and Body Boundary Determination. *IEEE Geosci. Remote Sens. Lett.* **2017**, *13*, 1920–1924. [CrossRef]
8. Hong, S.; Roh, B.; Kim, K.H.; Cheon, Y.; Park, M. Pvanet: Lightweight deep neural networks for real-time object detection. *arXiv* **2016**, arXiv:1611.08588.
9. 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. [CrossRef]
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016, pp. 21–37.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA , 21–26 July 2017; pp. 6517–6525.
12. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
14. Liu, L.; Pan, Z.; Lei, B. Learning a rotation invariant detector with rotatable bounding box. *arXiv* **2017**, arXiv:1711.09405.
15. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 900–904.
16. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
17. Liu, W.; Ma, L.; Chen, H. Arbitrary-Oriented Ship Detection Framework in Optical Remote-Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 937–941. [CrossRef]
18. Xue, Y.; Hao, S.; Xian, S.; Yan, M.; Fu, K. Position Detection and Direction Prediction for Arbitrary-Oriented Ships via Multitask Rotation Region Convolutional Neural Network. *IEEE Access* **2018**, *6*, 50839–50849.
19. Uijlings, J.R.; Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
20. Zitnick, C.L.; Dollar, P. Edge Boxes: Locating Object Proposals from Edges. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 391–405.
21. Cheng, M.M.; Zhang, Z.; Lin, W.Y.; Torr, P. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In Proceedings of the Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 3286–3293.
22. Pont-Tuset, J.; Barron, J.; Marques, F.; Malik, J. Multiscale Combinatorial Grouping. In Proceedings of the Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014 ; pp. 328–335.

23. Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship Rotated Bounding Box Space for Ship Extraction From High-Resolution Optical Satellite Images With Complex Backgrounds. *IEEE Geosci. Remote Sens. Lett.* **2017**, *13*, 1074–1078. [CrossRef]
24. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
26. Zhang, R.; Jian, Y.; Zhang, K.; Chen, F.; Zhang, J. S-cnn-based ship detection from high-resolution remote sensing images. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B7*, 423–430. [CrossRef]
27. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
29. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
30. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. Ssh: Single stage headless face detector. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4875–4884.
31. Zhu, C.; He, Y.; Savvides, M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1903.00621.
32. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. *arXiv* **2019**, arXiv:1904.01355.
33. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
34. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
35. Iglovikov, V.; Shvets, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv* **2018**, arXiv:1801.05746.
36. Fu, J.; Liu, J.; Wang, Y.; Zhou, J.; Wang, C.; Lu, H. Stacked deconvolutional network for semantic segmentation. *IEEE Trans. Image Process.* **2019**, doi:10.1109/TIP.2019.2895460. [CrossRef] [PubMed]
37. Cheng, D.; Meng, G.; Cheng, G.; Pan, C. SeNet: Structured edge network for sea–land segmentation. *IEEE Geosci. Remote Sens. Lett.* **2016**, *14*, 247–251. [CrossRef]
38. Volpi, M.; Tuia, D. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 881–893. [CrossRef]
39. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
40. Sun, W.; Wang, R. Fully Convolutional Networks for Semantic Segmentation of Very High Resolution Remotely Sensed Images Combined with DSM. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 474–478. [CrossRef]
41. Masoud, M.; Bahram, S.; Mohammad, R.; Fariba, M. Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.* **2018**, *10*, 1119–1140.
42. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
43. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
45. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.

46. Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. Richer convolutional features for edge detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3000–3009.
47. Hara, K.; Liu, M.Y.; Tuzel, O.; Farahmand, A.m. Attentional network for visual object detection. *arXiv* **2017**, arXiv:1702.01478.
48. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
49. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
50. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 818–833.
51. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia* **2018**, *20*, 3111–3122. [CrossRef]
52. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.
53. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 20 October 2019).
54. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Girshick, R.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
55. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:1603.04467.