remote sensing

MDPI

Article

Multiple-Oriented and Small Object Detection with Convolutional Neural Networks for Aerial Image

Chao Chen, Jiandan Zhong * and Yi Tan

College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China; cchen@cuit.edu.cn (C.C.); tanyi@cuit.edu.cn (Y.T.)

* Correspondence: jdzhong@std.uestc.edu.cn; Tel.: +86-138-8049-5638

Received: 28 August 2019; Accepted: 17 September 2019; Published: 18 September 2019

Abstract: Detecting objects in aerial images is a challenging task due to multiple orientations and relatively small size of the objects. Although many traditional detection models have demonstrated an acceptable performance by using the imagery pyramid and multiple templates in a sliding-window manner, such techniques are inefficient and costly. Recently, convolutional neural networks (CNNs) have successfully been used for object detection, and they have demonstrated considerably superior performance than that of traditional detection methods; however, this success has not been expanded to aerial images. To overcome such problems, we propose a detection model based on two CNNs. One of the CNNs is designed to propose many object-like regions that are generated from the feature maps of multi scales and hierarchies with the orientation information. Based on such a design, the positioning of small size objects becomes more accurate, and the generated regions with orientation information are more suitable for the objects arranged with arbitrary orientations. Furthermore, another CNN is designed for object recognition; it first extracts the features of each generated region and subsequently makes the final decisions. The results of the extensive experiments performed on the vehicle detection in aerial imagery (VEDAI) and overhead imagery research data set (OIRDS) datasets indicate that the proposed model performs well in terms of not only the detection accuracy but also the detection speed.

Keywords: object detection; aerial image; convolutional neural network; deep learning; multiple orientations

1. Introduction

The rapid breakthrough of remote sensing satellite and unmanned aerial vehicle (UAV) technologies has ensured the easy availability of a large number of aerial images. Recently, such aerial images have been widely used in the daily activities of human society. For example, in urban planning, the UAV systems can promptly obtain the ground information including the outer frame information and vector information of city houses, and using this information, digital line graphics can be rapidly generated [1]. In the field of intelligent transportation, the high-resolution remote sensing images obtained by UAV systems are used to detect traffic accidents or vehicle flows [2]. Additionally, it is convenient to monitor the growth of crops in agricultural regions by using UAVs [3]. Therefore, extracting and mining valuable information from aerial images have attracted considerable attention. Object detection is a crucial task in computer vision, which focuses on understanding the image information efficiently and widely applied in some practical applications, such as in surveillance [4] and security systems [5,6]. However, object detection in aerial images is a challenging task because of the obscure and cluttered backgrounds and the presence of arbitrarily oriented and relatively small objects. In addition, in the aerial images, the objects always exhibit an overhead appearance, which may cause difficulty in extracting the discriminative features, thereby leading to confusion or false

detection. Practically, an object detection model is required to exhibit a high performance in terms of detection accuracy as well as speed.

Generally, the object detection technology focuses on two key sub-tasks: positioning the objects and extracting the features for recognizing. Object positioning aims to find the location of an object, which is usually defined by a rectangular bounding box. In the last few decades, the mainstream technique involves a sliding-window fashion, which means that the detection windows are shifted at different locations over the image with a specific stride. Dalal et al. [7] designed a pedestrian detector based on this strategy. Additionally, the classical DPM (deformable parts model, proposed by Felzenszwalb et al. [8]) employed a set of part components and adopted this strategy to detect an object over an image pyramid, which demonstrated an acceptable performance. Subsequently, many variants or improved approaches were designed based on this technique [9–12]. However, the sliding-window strategy is time-consuming because too many candidate regions are generated when traversing over an image pyramid, which is expected to reduce the computational complexity in object positioning. Consequently, extensive studies [13–26] proposed techniques called region proposal methods to reduce the number of candidate regions. Uijlings et al. [13] developed a regional proposal approach to generate candidate regions; the authors adopted a hierarchical segmentation and grouping strategy and measured the generated regions in terms of the objectness (which is defined as a probability value that determines the image region to be an object of any class). This approach is called selective search, and it generates only thousands of candidate regions from an image, which is considerably less than the number of regions generated by the sliding-window technique (approximately 10^5 for a medium-sized image). Although the selective search method reduces the number of candidate regions, it spends a large amount of time on the similarity calculation and grouping of each small segmented region. To reduce the time consumption in the process of region generation, a previous work [26] proposed a super-pixel segmentation approach named simple linear iterative clustering (SLIC), which exhibited high efficiency and simplicity; however, the segmented regions were irregular polygons, which are coarse and lead to inaccurate object positioning. With the development of deep learning, convolutional neural networks (CNNs) have presently become the most important method to generate candidate regions; Ren et al. [27] proposed the region proposal network (RPN), which adopts the anchor scheme to generate a set of candidate regions at each position of the feature map output from the CNN. Although the RPN obtained a better position accuracy compared to that of the traditional methods such as selective search and SLIC, its performance is not as satisfactory for small sized object. Therefore, Kong et al. [28] improved the RPN and proposed the HyperNet which generates multiple feature maps into one uniform feature space and exhibits better performance in terms of the small-sized object positioning. Recently, a large number of detection models [29–33] have adopted CNN-based methods to generate region proposals.

For the feature extraction and recognition, some researchers [7,8,34–39] attempted to design the discriminative features and train a classifier. A study [7] proposed a novel gradient feature named the histogram of oriented gradients (HOGS), based on which the authors trained a support vector machine (SVM) classifier to detect pedestrians, exhibiting satisfactory performance. In [8], the authors compressed the original HOG feature into 31 dimensions and trained a latent SVM, which demonstrated high efficiency for detection of objects in multiple categories. Additionally, various features were designed to deal with object detection problems, such as the scale-invariant feature transform (SIFT) [34] and its accelerated version named the speed up robust feature (SURF) [35]; these features are called hand-crafted features. Recently, with the rapid progress and development of deep learning, CNN has demonstrated a powerful feature representation capability, owing to which it can replace the aforementioned hand-crafted features to become the most important method of feature extraction. In another study [36], Krizhevsky et al. designed a CNN model called AlexNet and trained it on a dual GPU (graphics processing unit) architecture. In 2012, this model obtained excellent results [37] in the image classification challenge ILSVRC2012. Since then, an increase in the amount of CNN architectures have been proposed to handle various tasks in the field of computer vision, such as

the Z&F-Net [37], VGG-16, VGG-19 [38], and Inception [39]. Furthermore, combined with the soft-max layer, the trained CNN model can be used not only for feature extraction but also for classification. This type of "end-to-end" configuration is superior to the traditional two-stage approach [7,8].

Furthermore, some studies [40–42] proposed a type of uniform detection framework, which adopted only one CNN architecture to combine the generation of candidate regions and object recognition into one stage. Specifically, the authors in [40] handled the detection task as a regression problem in which the object's bounding box and its probabilities of the associated class were directly predicted. Liu et al. [41] proposed the single-shot multi-box detection model (SSD), which could directly output the confidence score of the object-like regions. Compared with the RPN-based method [27], these studies [40–42] reported improvement in the detection speed due to the less complicated architecture.

In the context of the object detection in aerial images, most works are focused on detecting vehicles because vehicles are common objects in such imagery and play a key role in many applications such as traffic analysis and reconnaissance missions. Traditionally, the methods are based on handcrafted features and a slide window fashion; in addition, certain road information is always used as auxiliary information, such as the orientation information. Recently, CNN has become the main approach in object detection. Although the existing CNN based detectors [27–33] have achieved promising results in general images, their performance is not satisfactory for aerial images. The small size of the objects in an aerial image (the average size of an object is approximately 40×20 pixels) makes it challenging for the RPN [27] to generate small-sized regional proposals because the RPN adopts the feature map generated from the deeper convolutional layer of the CNN, which is too coarse to enable positioning. Furthermore, the objects in aerial images are aligned in multiple orientations [43]; however, the proposals generated from the CNN model are always horizontal, which means that the position of an object may not be accurate. Consequently, the horizontal region proposals can lead to detection loss and cannot be applied in practice for aerial images.

To this end, we propose a CNN-based model to overcome the problems of detection of small-sized objects and objects with multiple orientations in aerial images. The proposed model includes two CNNs, which divide the detection task into two stages: first, one of the CNNs is trained to generate the candidate region proposals with multiple orientations; subsequently, the generated region proposals are fed into another CNN for object recognition. The contributions of this paper can be described as follows:

- We designed a CNN-based detection model for the objects in aerial images, which is different from the recent CNN-based models and traditional models (that adopt hand-crafted features and the sliding-window scheme). Our model consists of two independent CNNs: MORPN (multiple orientation regional proposal network) and ODN (object detection network): the MORPN is applied to generate multiple orientation region proposals, and the ODN is used to extract the features and make decisions.
- To deal with the objects with a small size and multiple orientations in the aerial image, we proposed the MORPN. For the small-sized objects, the proposed MORPN employs a hierarchical structure that combines the feature maps of multiple scales and hierarchies to generate the region proposals. For the objects with multiple orientations, to improve the positioning accuracy, the angle information is adopted to generate the oriented candidate regions, unlike the classical CNN-based models that generate only horizontal region proposals. Moreover, an object detection network named ODN is trained to extract deep features and make decisions.
- The proposed detection model was tested on two real-world aerial image datasets: VEDAI (vehicle detection in aerial imagery) [44] and OIRDS (overhead imagery research data set) [45]. Extensive experiments were conducted, and the evaluation results indicated that the proposed model achieved significant improvement in the detection performance compared to that of its counterparts.

The rest of this paper is organized as follows. In Section 2, we summarize the related works pertaining to object detection in aerial images. Section 3 describes the basic theories and analyses of the MORPN and ODN. In Section 4, we describe the extensive experiments conducted on the two datasets and discuss the testing results. In Section 5, we conclude this paper and describe the scope for future work.

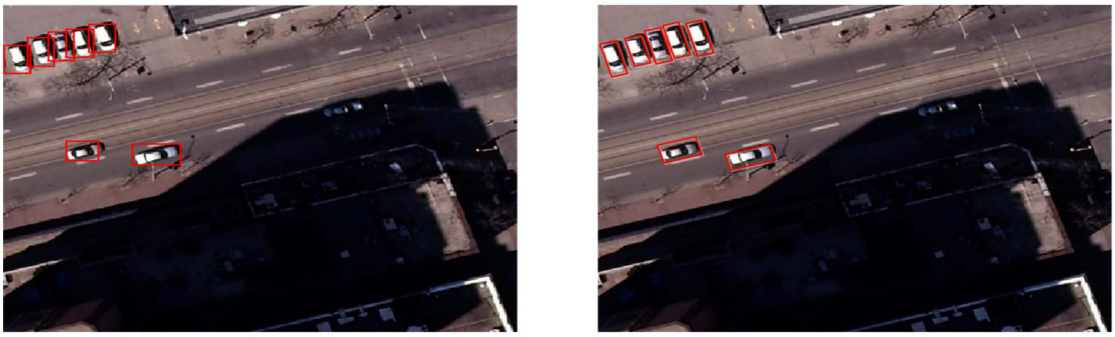
2. Related Work

The detection of objects in aerial images has been extensively studied over the last few decades, although most works [46–56] focused on extracting the discriminative feature and generating accurate region proposals. Xu et al. [46] employed the Viola–Jones approach, the histogram of oriented gradients (HOG) features and the linear support vector machine (SVM) to design a model for vehicle detection in UAV imagery, which exhibited a high performance. However, this scheme has a limitation, that is, the roadway information is required to adjust the orientation of the images. In addition, this scheme is not suitable for the objects arranged in multiple orientations, for example, in the aerial images without any roadway information. To deal with the rotated object detection problem in optical remote sensing images, Wang et al. [51] proposed the rotation-invariant matrix (RIM) features. These features were first encoded into fisher vectors, and subsequently, a spatial pyramid pooling strategy was adopted to obtain richer information. This model exhibited satisfactory performance in detecting certain medium-sized objects such as airplanes; however, it only outputs the horizontal bounding box, which is not suitable for the densely arranged and oriented objects. In another study [52], the researchers developed a vehicle detection method for aerial images, which utilized the GIS road vector map and morphological method and obtained an overall accuracy of 91.5% for 17 highway scenes; however, this method is not suitable for the objects in other scenes such as those of urban, suburban and residential areas.

Recently, CNNs have become the most prominent technique for object detection in aerial images, and most of the related approaches are developed based on two schemes. In some cases, the CNN is applied to replace the traditional hand-crafted features for feature extraction [47,48]. Ammour et al. [47] adopted a CNN as a feature extractor to detect objects in UAV images; they first employed a segmentation approach to generate the candidate regions from the input image and later fed the regions into a CNN model (which was pre-trained) for feature extraction. Finally, an SVM was trained to make the decisions. One study [48] adopted a feature extraction method that fused the spatial information of each candidate region. Ševo et al. [53] fine-tuned an Inception [39] model on the UCMerced dataset and U.S. Geological Survey (USGS) image dataset to realize object detection. Al-Najjar et al. [54] trained a CNN model to classify seven types of land covers on fused datasets (unmanned aerial vehicle imagery combined with the digital surface model (DSM)), and it was noted that the combination could help improve the accuracy. Because the extracted features from a CNN are more discriminative than the traditional hand-crafted features (such as HOG), the aforementioned models exhibit a satisfactory performance in aerial images. However, the detection phases of the models are complex, involving feature extraction, feature encoding and model fine-tuning.

Furthermore, a CNN model can be used for not only feature extraction but also the generation and classification of the candidate regions. Yan et al. [55] proposed a method that used the adaptive intersection-over-union (IoU) information to guide the detection of small-sized objects in aerial imagery. In addition, they designed a type of IoU-based weighted loss, which further improved the detection accuracy. Tang et al. [49] and Kong et al. [28] respectively proposed the hyper region proposal network (HRPN) and HyperNet, to locate small-sized objects accurately. Their work employed stacked multi-feature maps, which yielded a better positioning accuracy compared to that of the classical RPN. Deng et al. [50] improved the RPN and Fast R-CNN and proposed a two-way CNN model to realize small-sized object detection. Although the above CNN-based detection models attempted to address the problem of small-sized object detection and could demonstrate satisfactory performance for aerial images, the orientation information was always ignored in these studies [49,50,55] as the RPN-based models can generate only horizontal regions. In contrast, the oriented region proposals are more

similar to the real conditions and can help improve the detection accuracy. From the illustrations in Figure 1, it can be clearly seen that the horizontal bounding boxes (in Figure 1a) always overlap each other when the objects are arranged in a dense manner, which leads to the generation of inaccurate region proposals; in contrast, the oriented bounding box (in Figure 1b) can avoid such problems.



(a) (b)

Figure 1. (**a**) The bounding boxes without orientations (horizontal bounding box) may lead to overlapping or even missing, which can decrease the detection accuracy. (**b**) The bounding boxes with orientations can avoid such problems.

3. Methods

The proposed model is illustrated in Figure 2. The model consists of two CNNs: the first CNN, called the multiple orientation region proposal network (MORPN) is designed to propose multiple orientation region proposals; the second CNN is the object detection network (ODN), which performs feature extraction and recognition for the oriented region proposals generated by the first network. The detection process can be described as follows: 1) The image is input to the MORPN to generate the region proposals; 2) the image and its corresponding generated region proposals are fed to the ODN as input, and the confidence score of each region is obtained; 3) the detected regions are determined. Usually, we employ a threshold for the evaluation; if the detection score of detection is greater than the threshold, it is considered a true detection. The proposed model has the following differences with the existing CNN-based detection models [27,28]: 1) Unlike the RPN and HyperNet, which can only generate horizontal region proposals, the multiple angle information is combined with the region proposals, and the models detect the objects with an oriented bounding box; 2) unlike the HyperNet, which constructs a stacked feature map for generating the region proposals, we design a hierarchical structure that combines the feature map from various scales (output of the shallow and deep convolutional layers) to detect small-sized objects. Although the output of the deep convolutional layers exhibits an inferior performance in terms of the object positioning, it is associated with a higher detection recall; furthermore, the output of the shallow layers has an inferior detection recall but can help in the accurate positioning of small-sized objects. To exploit the advantages of both the layers, the feature maps output from the deep and shallow layers are combined in the proposed model; 3) we train the two networks independently, which is different from the approach adopted in previous studies [27,28]. This aspect implies that we do not need to train the sharing and un-sharing layers of the two networks. During the training stage, the two CNNs are trained independently. The detailed description of this model is provided in the following subsections.

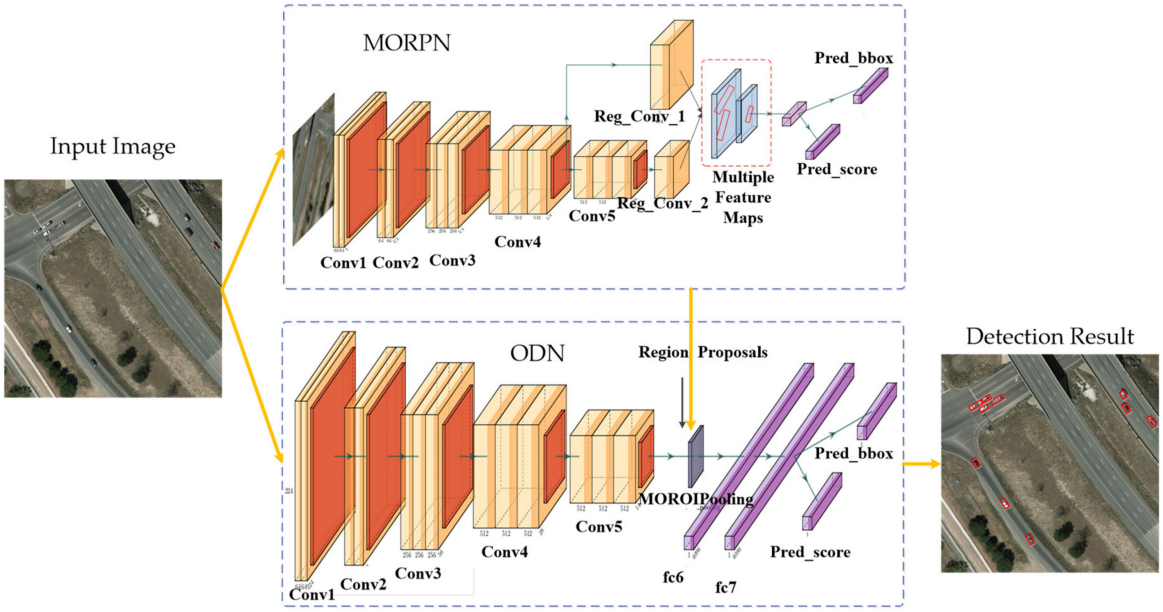


Figure 2. The complete architecture of the proposed model.

3.1. Backbone Architecture

The proposed MORPN and ODN adopt the VGG-16 model [39] as the backbone architecture. The VGG-16 is a deep CNN model, which has achieved significant success in the field of computer vision. The architecture of the VGG-16 model is shown in Figure 3. The model consists of 13 convolutional layers, 5 max-pooling layers and 3 fully connected layers. The convolutional layers are used to generate the feature maps, and the pooling layers down-sample the input feature and maintain sufficient information. The generated deep feature maps are input into the fully connected layers fc6, fc7 and fc8. The fc6 and fc7 layers generate a feature vector of 4096D (where D denotes the dimensions), and the fc8 layer generates a feature vector of 1000D. Finally, a soft-max layer takes this 1000D feature vector as the input to make the final decision.

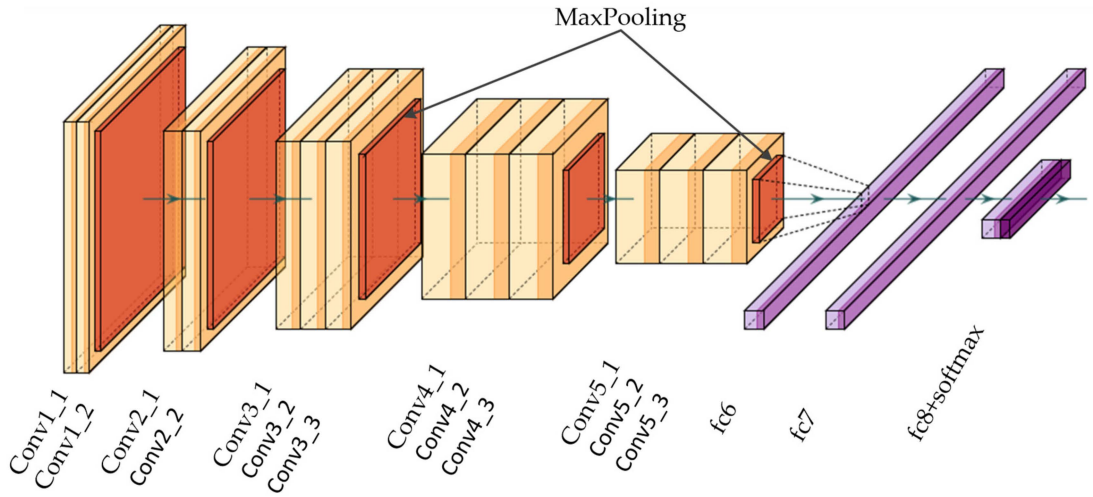


Figure 3. The architecture of the VGG-16 model.

The VGG-16 model adopts the convolutional kernel with a size of 3×3 and a stride of 1 for all the convolutional layers. Moreover, it adopts the max-pooling strategy with a 2×2 pixel window and stride of 2 for each pooling layer.

3.2. Region Proposal Approach

The region proposal approach is used to generate the object-like regions, and the RPN [27] is the first heuristic approach that adopts a CNN to accelerate the generation progress of the region proposals and increase the positioning accuracy. The RPN (shown in Figure 4) is designed based on the VGG-16, which reserves the convolutional layers (from *conv1_1* to *conv5_3*) and deletes the fc6, fc7 and fc8 layers; subsequently, the model slides a small convolutional network over the feature map generated by the deepest convolutional layer. Two sibling layers named *reg_layer* and *cls_layer* are connected behind the small network to perform the box regression and classification, respectively.

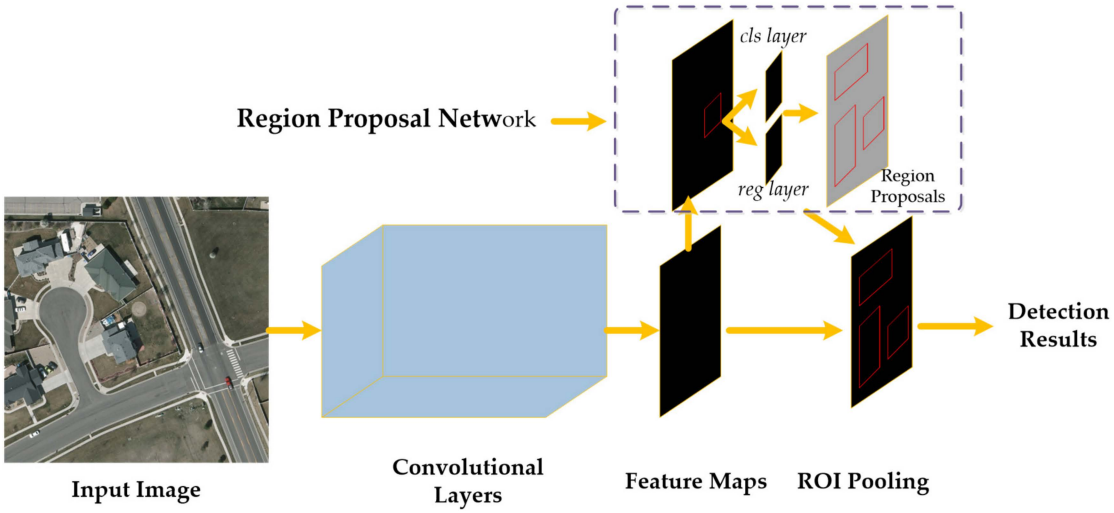


Figure 4. The architecture of the region proposal network (RPN).

The RPN adopts the anchor scheme to generate the region proposals. To account for the various object sizes, the sliding window generates k proposals at each sliding position of the feature map. As reported in [27], the parameter k is controlled by the scale and aspect ratio. Consequently, the *reg_layer* and *cls_layer* produce 4k outputs and 2k scores, respectively. Each region has 4 coordinates and 2 scores to represent its position and confidence, respectively.

However, in aerial images, the objects usually have a small size with multiple orientations. The horizontal proposals generated by the RPN are not suitable for such objects. Therefore, a more efficient and robust object detection model should not only generate small-sized region proposals but also provide proposals with the orientation information corresponding to the regions.

3.3. MORPN

The MORPN utilizes the VGG-16 as the backbone, and it is designed to address the problem of small-sized and oriented objects in aerial images. The MORPN is used to generate the object-like regions and their corresponding objectness scores from an input image. First, to propose the small candidate regions, the architecture with hierarchical feature maps is constructed instead of using the deepest feature map of the CNN model. Second, the oriented anchor scheme is employed to generate the candidate regions with various orientations.

(1). Hierarchical feature map.

Similar to the hyper feature map (which stacks multi convolutional layers together) proposed in [28,46], we construct a hierarchical structure by employing the output of the shallow and deep features from the convolutional layers. The feature map output from the shallow convolutional layer can efficiently locate the position of the object but has a reduced recall; furthermore, the feature map generated from the deeper layers can obtain a higher detection recall but exhibits inferior performance in terms of object positioning.

Similar to the RPN, we use the VGG-16 model as the backbone of the proposed MORPN, and subsequently, we make the following changes: 1) The soft-max layer and 3 fully connected layers from *fc_6* to *fc_8* are deleted; 2) two small networks named *reg_conv_1* and *reg_conv_2* are added behind the convolutional layers *conv4_3* and *conv5_3*; these two small networks are the two convolutional layers; 3) 512 convolutional kernels with a size of $3 \times 3 \times 512$ are adopted in these two small networks, which are employed to produce the 512D feature vectors over the hierarchical feature maps; 4) the 512D feature vector is simultaneously fed into the *pred_bbox* and *pred_score* layers. The details of this architecture are shown in Figure 5.

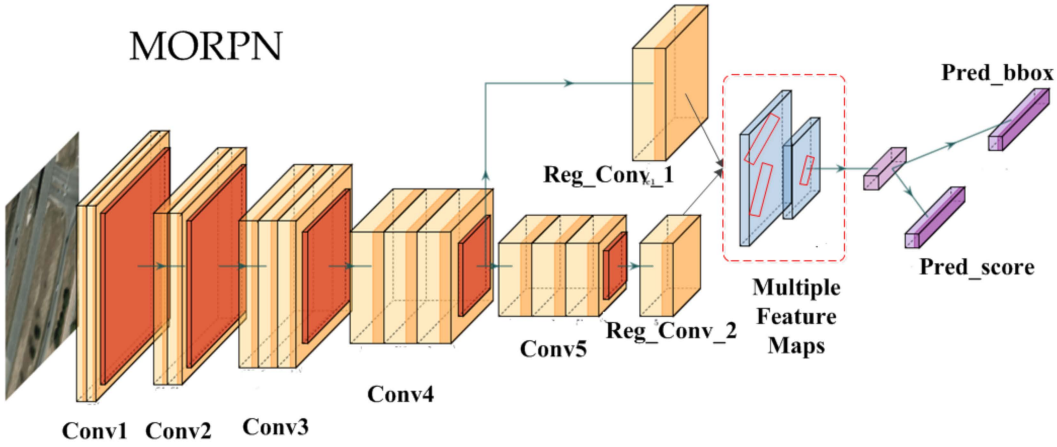


Figure 5. The architecture of multi-feature map.

(2). Oriented anchors.

We propose the oriented anchor scheme to generate the proposals of regions with multiple orientations and adopt the regression technique for the bounding box, which makes the proposals more suitable for the object regions. During the training phase, the ground truth of the object is defined as (x, y, w, h, θ), which represents an oriented bounding box with 5 tuples. The coordinate (x, y) is the centroid of the bounding box, w represents the long side (width) of the bounding box, h represents the height of the bounding box, and θ is the angle from the positive direction of the x-axis to the direction parallel to the long side of the oriented bounding box. Only half of the angular space $[0, \pi)$ can cover the entire orientation of the generated proposals because the generated proposal and its contrast proposal represent the same detected region, which does not reduce the recall rate. For θ in the range of $[\pi, 2\pi)$, its value is updated as $\theta := \theta - \pi$.

Compared with the traditional anchor scheme, the oriented anchor scheme is suitable for objects in aerial images. The angle parameter θ of the proposal is adopted to determine the orientation of the object. To avoid the computational load of the orientation coverage, we establish a balance; therefore, only 6 orientations $(0, \pi/6, \pi/3, \pi/2, 2\pi/3$ and $5\pi/6)$ are employed in our experiments. Moreover, the other two parameters (aspect ratio and scale) used in the traditional RPN are adopted as well. We employ 3 aspect ratios [1:1, 2:1, and 1:2] and 4 scales $[8^2, 16^2, 32^2$, and $64^2]$. Through this scheme, 72 generated regions are obtained at each position of the feature map. The two sibling output layers *pred_bbox* and *pred_score* generate 360 outputs (72×5) and 144 scores (72×2), respectively. The oriented anchor scheme is illustrated in Figure 6.

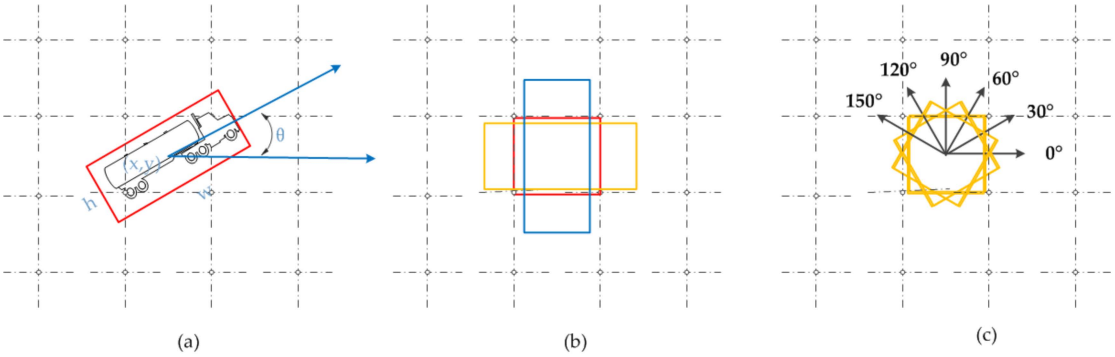


Figure 6. (**a**) The oriented bounding box with 5 tuples; the (**b**) 3 aspect ratios and (**c**) 6 orientations of the rotated bounding box.

(3). Loss function.

The MORPN has two output layers, *pred_bbox* and *pred_score*, which are employed to output the position of the region proposal and its corresponding objectness score. To train the MORPN in a single stage, a multi-task loss function [57] is employed during training. As shown in Equation (1), the function L consists of two sub-functions adopted for the bounding-box regression and classification.

$$L(p^t, l^t) = L_{cls}(p^t, p^g) + \lambda * p^g * L_{br}(l^t, l^g) \tag{1}$$

In Equation (1), p^t represents the probability that the region is an object, which is the output of the *pred_score* layer. The label p^g indicates the ground truth (1 and 0 for the positive and negative regions, respectively). L_{cls} represents the log loss between the background and object-like region. In addition, the parameter λ is the balance parameter in this loss function, which is set as 10 in the training stage.

The label l^t is the output of the *pred_bbox layer*, which is a vector of 5 parameters *(x, y w, h, θ)* used to predict the bounding box; l^g represents the ground truth. For the loss function L_{br}, we adopt the smooth *L1 loss function* [57]. Equations (2) and (3) provide a detailed description of these aspects.

$$L_{br}(l^t, l^g) = S_{L1}(l^t - l^g) \tag{2}$$

$$S_{L1}(z) = \left\{ \begin{array}{l} 0.5z^2 \ \text{if}|z| < 1 \\ |z| - 0.5 \ \text{others} \end{array} \right\} \tag{3}$$

(4). Settings of model training.

During the training stage, positive labels are assigned to the regions that simultaneously satisfy the following conditions: a) The highest intersection-over-union (IoU) ratio (defined in Equation (4)) between the ground truth and the regions is larger than 0.7, and b) the intersection angle between the ground truth and the regions is less than π/12. An anchor is defined as a positive sample when a ground truth box lies in its fit domain. In terms of the angle range [0, π) and the six orientations defined in our work, the fit domain is divided into six equal parts. Therefore, the arbitrary ground truth box fits the anchors with an adaptive fit domain. The negative labels are assigned to the regions that satisfy the following conditions: a) the IoU ratio is less than 0.3, or b) the intersection angle between the ground truth and the regions is larger than π/12 (with the condition that the IoU ratio is larger than 0.7).

$$IoU_{ratio} = \frac{A_{reg} \cap A_{gt}}{A_{reg} \cup A_{gt}} \tag{4}$$

In Equation (4), A_{reg} is the area of the region proposals, and A_{gt} is the area of the ground truth bounding box. The IoU computation of the oriented proposals is different from that of the horizontal proposals because the overlap parts are always polygonal regions. Therefore, in this study, we compute the IoU ratio considering the triangulation [58]. By using the triangulation, the polygonal regions can

be divided into a set of triangle regions, and the polygonal area can be computed by summing the triangles; subsequently, the IoU ratio can be computed.

The MORPN is initialized by the VGG-16 model, which is first pre-trained on the ILSVRC dataset, and we adopt the stochastic gradient descent (SGD) technique as the optimization method. Additionally, we adopt a mini-batch of 256 for an input image, and the numbers of positive and negative examples have a ratio of 1:1. Once the number of positive examples is less than 128 in an image, the negative ones are padded.

3.4. ODN

(1). Architecture

The ODN also adopts the VGG-16 model as the backbone. To extract the feature vector with a fixed length from the region proposals (having multiple sizes, scales and orientations), a MOROI (multiple orientation region of interest) polling layer is added. Behind the MOROI pooling layer, we add the *fc_6* and *fc_7* layers, which are the two fully connected layers. The ODN is required to output the position and corresponding confidence score of the object. Hence, we add two *fc* (fully connected) layers behind the *fc_7* layer. The detailed illustrations of these aspects are shown in Figure 7.

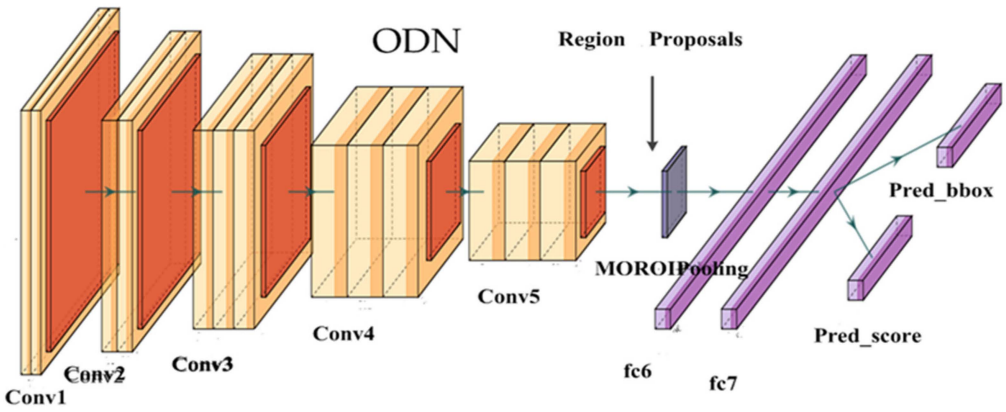


Figure 7. The architecture of the object detection network (ODN).

The input of the ODN includes 1) the image similar to the input of the MORPN, and 2) the region proposals output from the MORPN; these region proposals are directly mapped into the MOROI pooling layer.

Unlike the ROI pooling layer in the RPN, which can only deal with the horizontal region proposals, the MOROI pooling layer in the ODN is used to handle the oriented region proposals. From each region proposal, a fixed-length feature vector is generated and later input into a fully connected layer *fc_6*, which generates a feature vector of 4096D. The *fc_7* layer adopts the similar settings as those of *fc_6*. In addition, we add two sibling *fc* (fully connected) layers behind the *fc_7* layer, namely, the *pred_bbox* and *pred_score* layers. These two layers yield the bounding box and confidence score of the predicted object, respectively.

(2). Settings of the model training.

The loss function of the ODN is equivalent to that of the MORPN (shown in Equation (1)). In the training stage as well, we adopted the same pre-trained VGG-16 model and the hyper parameters (trained on ImageNet) as those described in Section 3.3.

(3). Multiple orientation region of interest (MOROI) layer.

The MOROI pooling layer is used to extract the fixed-length feature vectors. Practically, the region proposal is divided into $H \times W$ sub-regions. Therefore, each sub-region has the same orientation as that of the region proposal, and its size is $\frac{h}{H} \times \frac{w}{W}$. Additionally, the standard max pooling is performed

independently in each sub-region and feature map channel. The computation of the MOROI pooling is described in Table 1.

$$\begin{cases} x\prime \leftarrow (x_0 - x)\cos\theta + (y_0 - y)\sin\theta + x \\ y\prime \leftarrow (y_0 - y)\cos\theta - (x_0 - x)\sin\theta + y \end{cases} \tag{5}$$

$$\begin{cases} p_x \leftarrow x\prime \times S_a + n\cos\theta + m\sin\theta + \frac{1}{2} \\ p_y \leftarrow y\prime \times S_a - n\sin\theta + m\cos\theta + \frac{1}{2} \end{cases} \tag{6}$$

where $m \in \{0, \ldots S_h \times S_a - 1\}, n \in \{0, \ldots S_w \times S_a - 1\}$

Table 1. Max pooling process of the multiple orientation region of interest (MOROI) layer.

For an input-oriented region proposal defined by (x, y, h, w, θ) with a spatial scale S_a
1. Calculate the sub-region size: $S_w \leftarrow \frac{w}{W}, S_h \leftarrow \frac{h}{H}$
2. Find the top-left coordinate of each sub-region as follows: $x_0, y_0 \leftarrow x - \frac{w}{2} + jS_w, y - \frac{h}{2} + jS_h$
Where, $i \in \{0, \ldots H-1\}, j \in \{0, \ldots W-1\}$
3. Calculate the rotated coordinate of (x_0, y_0) by using Equation (5) (presented below)
4. Perform max-pooling on each sub-region
a. set $v = 0$;
b. compare the value of the feature map with v at each position (p_x, p_y) by using Equation (6) (presented below)
c. set v as the max value of this region
d. set the position (i,j) of the output feature map as $Feature_{(i,j)} = v$
5. Repeat the steps from 2 to 5 until the $Feature_{(i,j)}$ at each sub-region is calculated.

4. Experimental Results and Discussions

We validate the proposed model on two datasets: the VEDAI dataset [44] and OIRDS [45]. The model performance is evaluated in terms of speed and accuracy. Section 4.1 describes the evaluation metrics. All the programs in our experiments are implemented based on MATLAB2014a and Caffe. The running environment includes a (Titan X) GPU with a 12 GB memory and a multi-core (Intel Core i7) CPU.

4.1. Evaluation Metrics

To evaluate the proposed detection model, we employ the following four metrics in our experiments:

(1) Recall rate.

The recall rate [59], as defined in Equation (7), indicates the ratio of the number of detected objects to the number of all related objects. Here, TP denotes the true positive, and FN denotes the false negative.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

(2) Average precision (AP)

The average precision [60] is a value that intuitively indicates the performance of the detectors, and it is defined as in Equation (8).

$$ap = \int_0^1 p(r)dr \tag{8}$$

Here, r represents the recall, and p represents the precision [55], which is defined in Equation (9). FP in Equation (9) refers to the false positive. The average precision (AP) is calculated by evaluating the area under the precision–recall curve.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

(3) Precision–recall curve (PRC).

The precision–recall curve is determined considered the precision and recall; the x-axis is defined by the recall, and the y-axis is defined by the precision.

(4) F1-Score

The F1-Score [55] can be defined as in Equation (10).

$$F1\_Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

Overall, the F1-Score and AP are the equilibrium indicators, which reveal the performance of the detectors. The F1-Score and AP values are positively correlated with the performance. During the testing, if the IoU_{ratio} of the detection greater than 0.5, it is defined as true, or else, it is false.

4.2. Baselines

To validate the performance of the proposed model, we compared the models that employed different region-proposal schemes, including the selective search, segmentation, RPN, multi-feature maps without orientation (this approach adopted the similar framework as that of the MORPN but the orientation information was not used for region generation) and "one-stage" detection model (such as SSD and Yolo V2, which adopt only one CNN to handle the proposal generation and object recognition). Moreover, different CNN architectures such as the VGG-16 and Z&F models were considered as well. The following baselines were selected for comparison.

1. Fast RCNN [57]: The model adopted in the experiments employed the selective search scheme as the region proposal technique, and the generated region proposals were fed into a CNN for box regression and classification. We employed the VGG-16 as the backbone. This method is referred to as $Fast_{VGG\text{-}16}$.

2. Faster RCNN [27]: This model adopted the RPN as the region proposal approach, and approximately 300 region proposals were generated by the RPN. We adopted the VGG-16 and Z&F models as the backbone, and they are referred to as $Faster_{VGG\text{-}16}$ and $Faster_{Z\&F}$, respectively.

3. Segmentation and detection approach [26]: First, we segmented the testing image into approximately 700 candidate proposals without overlap and later fed these regions into the CNN-based model for classification. In our experiments, a super-pixel segmentation approach named the simple linear iterative clustering was adopted. The VGG-16 and Z&F models were adopted as the backbone, and they were respectively referred to as $SLIC_{VGG\text{-}16}$ and $SLIC_{Z\&F}$.

4. Cascaded model: This model employed two cascaded CNNs. Similar to the proposed model, this model adopted multiple and hierarchical feature maps to locate the small objects. The difference is that this model generated only the horizontal region proposals. The VGG-16 model was adopted as the backbone, and it was referred to as $Cascaded_{VGG\text{-}16}$.

5. SSD [41]: The model adopted in the experiments employed only one CNN for the proposal generation and object detection. The SSD model adopted the VGG-16 as the backbone, which added a set of extra feature layers behind the Conv5_3 layer to improve the accuracy for multi-scale objects. In the experiments, we referred to this model as the SSD.

6. Yolo V2 [42]: The model adopted in the experiments employed the darknet-19 as the backbone, which consists of 19 convolutional layers and 5 max-pooling layers. The darknet-19 mainly adopts the convolutional kernel with a size of 3×3 similar to that used in the VGG-16. In our experiments, this model was referred to as Yolo V2.

4.3. VEDAI Dataset

The VEDAI dataset is a benchmark dataset, which includes aerial images having two sizes: VEDAI 1024 (1024×1024 pixels) and VEDAI 512 (512×512 pixels). These images were recorded in Utah, US during spring 2012 and comprise nine classes of vehicles. The ground sampling distance of VEDAI 1024 is 12.5 cm/pixel, and it is 25 cm/pixel for VEDAI 512. Figure 8 shows some examples of the images in this dataset, which illustrates various backgrounds of the images, such as highways, crops and

residential areas. The detailed data distribution of this dataset is presented in Table 2. Nine classes are present in this dataset, however, the objects of some categories such as planes, boats and others are scarce. In our experiments, we discarded these categories and merged the remaining categories (cars, pick-ups, trucks, camping cars, tractors and vans) into a class named vehicle. Hence, approximately 3300 vehicles were adopted for training and testing.

Figure 8. Examples from the vehicle detection in aerial imagery (VEDAI) dataset [40].

Table 2. Data distribution of the VEDAI dataset.

Type	Tag	Number	Type	Tag	Number	Type	Tag	Number
Car	car	1340	Plane	pla	47	Tractor	tra	190
Pick-up	pic	950	Boat	boa	170	Van	van	100
Truck	tru	300	Camping car	cam	390	Other	oth	200

In the training phase, we randomly selected 1000 images from the VEDAI dataset as the training set. Each input image was pre-processed by resizing its shorter side to 600 pixels. Additionally, we employed the equivalent parameters and settings for the MORPN and ODN during the training stage. Four parameters, as described in Table 3, including the weight decay, momentum, iterations and learning rate were considered. Both the networks were trained in 40,000 iterations. To make the trained model more accurate, we adopted the variable learning rate during various stages: in the first 30,000 iterations, a learning rate of 0.001 was adopted; for the remaining 10,000 iterations, we adopted a finer learning rate (0.00001).

Table 3. Hyper parameters for the training stage.

Parameter	Value
Weight decay	0.0005
Momentum	0.9
Iterations	40,000
Learning rate	first 30,000 iters: 0.001 remaining 10,000 iters: 0.0001

During the testing phase, we used the remaining 247 images of the VEDAI dataset to evaluate the performance. We tested the proposed model and the baselines mentioned in Section 4.2. From the test results presented in Tables 4 and 5, it could be noted that the proposed model outperforms the other baselines on the VEDAI 1024 set in terms of the recall rate (75.1), AP (63.2%) and F1-score (0.469). Moreover, on the VEDAI 512 set, the proposed model still demonstrates the best performance, yielding the best recall rate (73.7%), AP (59.5%) and F1-Score (0.451). The Cascaded$_{\text{VGG-16}}$ model demonstrates the second-best performance (the AP is 72.3% and F1-Score is 0.320), which indicates that the hierarchical feature map is superior to the models that adopt only one feature map (Fast$_{\text{VGG-16}}$, Faster$_{\text{VGG-16}}$ and Faster$_{\text{Z\&F}}$). The SSD and Yolo V2 models demonstrate a worse performance compared

to that of our model and the Cascaded$_{\text{VGG-16}}$ model because the feature maps used for generating the region proposal are not fine for small-sized objects. The segmented approaches (SLIC $_{\text{VGG-16}}$ and SLIC$_{\text{Z\&F}}$) demonstrate the most inferior performance because the region proposals generated from the SLIC are coarse and irregular. Figure 9a,b respectively illustrates the PRC on the VEDAI 1024 and 512 sets. Clearly, the areas under the orange PRCs (proposed method) are the largest. This finding indicates that the proposed method demonstrates the best performance on the VEDAI 1024 and 512 sets, which is consistent with the results presented in Tables 4 and 5, where the best results are marked in bold.

Table 4. Detection results of different models on the VEDAI 1024 set.

Evaluation Metric	VEDAI 1024								
	Fast$_{\text{VGG-16}}$	Faster$_{\text{VGG-16}}$	Faster$_{\text{Z\&F}}$	SLIC $_{\text{VGG-16}}$	SLIC$_{\text{Z\&F}}$	Cascaded $_{\text{VGG-16}}$	SSD [41]	Yolo V2 [42]	Our Model
Recall Rate	72.2%	73.9%	63.5%	58.8%	58.3%	72.3%	70.5%	73.8%	**75.1%**
AP	39.8%	42.1%	30.8%	23.2%	25.4%	54.6%	46.1%	50.3%	**63.2%**
F1-Score	0.229	0.232	0.216	0.064	0.066	0.320	0.295	0.313	**0.469**

Table 5. Detection results of different models on the VEDAI 512 set.

Evaluation Metric	VEDAI 512						
	Fast$_{\text{VGG-16}}$	Faster$_{\text{VGG-16}}$	Faster$_{\text{Z\&F}}$	Cascaded$_{\text{VGG-16}}$	SSD [41]	Yolo V2 [42]	Our Model
Recall Rate	69.4%	71.4%	60.9%	69.7%	69.1%	70.3%	**73.7%**
AP	37.3%	40.9%	32.0%	50.2%	43.1%	46.9%	**59.5%**
F1-Score	0.224	0.225	0.212	0.305	0.292%	0.309%	**0.451**

The results are different between VEDAI 1024 and VEDAI 512 mainly based on the following reasons. Firstly, the testing set of VEDAI 1024 and VEDAI 512 have different size. Although the testing images are resized to the same size when fed into the proposed model, the detection results are mapped to the images of different sizes (testing images are 1024×1024 and 512×512 for VEDAI 1024 and VEDAI 512 respectively). This mapping involves the bounding box regression [8], and the regression parameter leaned from the *pred_bbox* layer. For different dataset, the learned regression parameters are different. Secondly, the IoU$_{\text{ratio}}$ decide whether the detections are defined as true. Because the objects in the aerial image have relatively small size, a little offsite (generated by the aforementioned mapping) of the bounding box will lead to the larger change of IoU$_{\text{ratio}}$, which also causes the differences between two testing sets with different image size.

The MORPN is the key component of the proposed model, and the positioning accuracy of the generated region proposal is a key factor that determines the final results. We evaluated the performance of the positioning corresponding to the proposed MORPN and other RPN-based region proposal approaches. The "one-stage" detection models such as the SSD and Yolo V2 were not selected for comparison because these models employed only one CNN for not only object positioning but also detection. In the RPN-based model and the proposed model, the positioning and detection are performed by two independent CNNs. The following experiments focused mainly on the positioning performance of the region proposal networks. In [27], the VGG-16 and Z&F models were adopted as the backbone of the RPN, and the recall–IoU curve was proposed as the evaluation criterion. Therefore, we employed the same criterion (as shown in Figure 10). From the results shown in Figure 10, it can be noted that the proposed model yields the best recall rate with the IoU rate ranging from 0 to 1; therefore, the MORPN exhibits the best positioning performance.

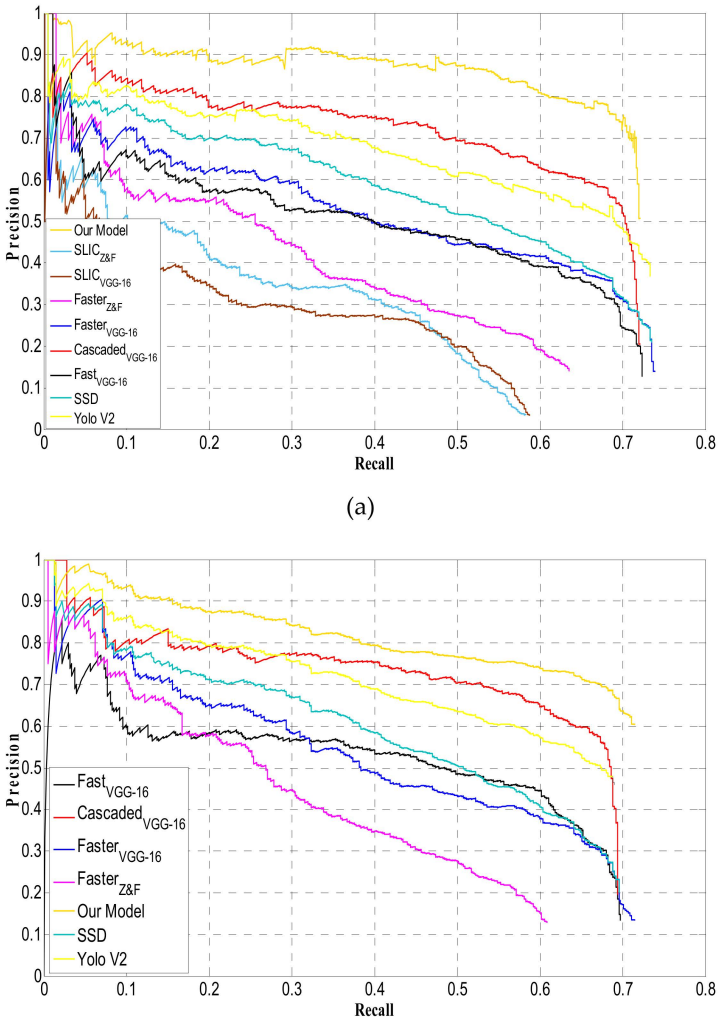
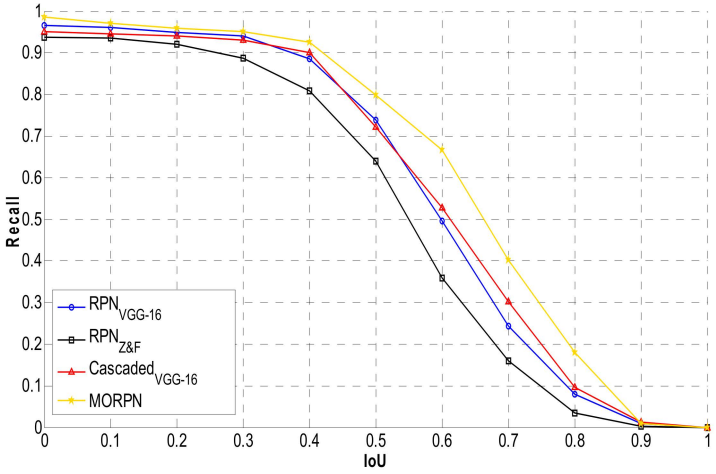


Figure 9. Precision–recall curves generated on two sub-testing sets: (**a**) VEDAI 1024 (**b**) VEDAI 512.

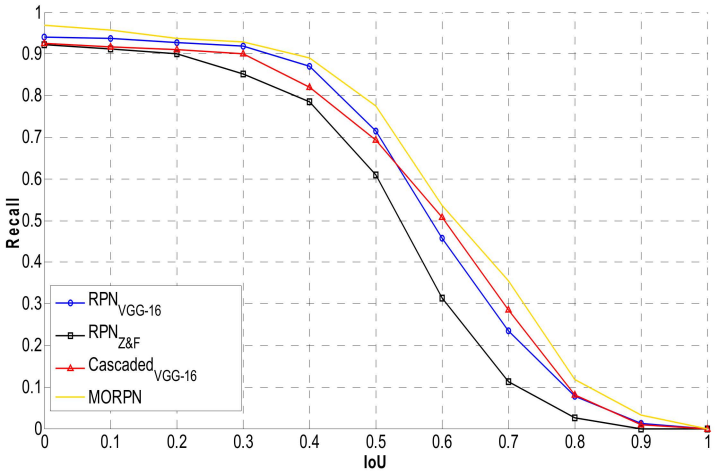
Furthermore, we evaluated the detection speed and training speed in our experiments. The detection time was evaluated in frames per second (fps), and the training time was evaluated in hours (h). Figures 11 and 12 show the results on different datasets; the detection time and training time are referred to as *Det. Time* and *Tra. Time*, respectively. In terms of the detection time, the $\text{Fast}_{\text{VGG-16}}$ model demonstrates the worst performance, the Yolo V2 and SSD models demonstrate the top two best performances, and the other models exhibit comparable performances. Because the $\text{Fast}_{\text{VGG-16}}$ adopts the selective search [13] scheme as the proposal generation approach, it consumes more time for proposal generation than the other models. The Yolo V2 and SSD models perform well in terms of the detection speed because the two models adopt only one CNN architecture to perform the bounding box regression and objection detection. However, the detection accuracy of the Yolo V2 (50.3% on VEDAI 1024, 46.9% on VEDAI 512) and SSD (46.1% on VEDAI 1024, 43.1% on VEDAI 512) models are worse than those of the $\text{Cascaded}_{\text{VGG-16}}$ model and the proposed model, which adopt two CNN architectures to realize the positioning and detection. The $\text{SLIC}_{\text{VGG-16}}$ and $\text{SLIC}_{\text{Z\&F}}$ models exhibit a detection speed that is similar to that of our model; however, their detection accuracies are extremely low (23.2% and 25.4%) because the region proposals generated by the adopted segmentation technique are extremely coarse, which degrades the positioning accuracy. The proposed model and $\text{Cascaded}_{\text{VGG-16}}$ exhibited nearly similar performances because these two models adopt a similar architecture. The $\text{Faster}_{\text{Z\&F}}$ and $\text{Faster}_{\text{VGG-16}}$ models have a slightly higher detection speed than those of our model. This phenomenon occurs because, in the case of the $\text{Faster}_{\text{Z\&F}}$, the shallower CNN

model (Z&F model) yields less computation complexity; however, the detection accuracy of this model (30.8% on VEDAI 1024, and 32% on VEDAI 512) is extremely poor. The $\text{Faster}_{\text{VGG-16}}$ does not use the hierarchical architecture to generate the region proposals, leading to less time consumption. However, this gap is not remarkably large in practical use.

In terms of the training time, the training stage of CNN is always time-consuming. The $\text{Fast}_{\text{VGG-16}}$, $\text{SLIC}_{\text{VGG-16}}$ and $\text{SLIC}_{\text{Z\&F}}$ models incur less time because these models adopt the CNN architecture only for classification, and this process is relatively simple. In contrast, the other models adopt a CNN for not only proposal generation but also object detection; therefore, they consume considerably more time during training. In particular, the $\text{Faster}_{\text{Z\&F}}$ and $\text{Faster}_{\text{VGG-16}}$ models incur a large time consumption because they alternatively train two CNN models two times. In practical use, the detection time is considered more than the training time because if a model is trained and deployed, it will rarely be re-trained.



(a)



(b)

Figure 10. Recall–IoU (intersection-over-union) curves of two sub-testing sets: (**a**) VEDAI 1024 (**b**) VEDAI 512.

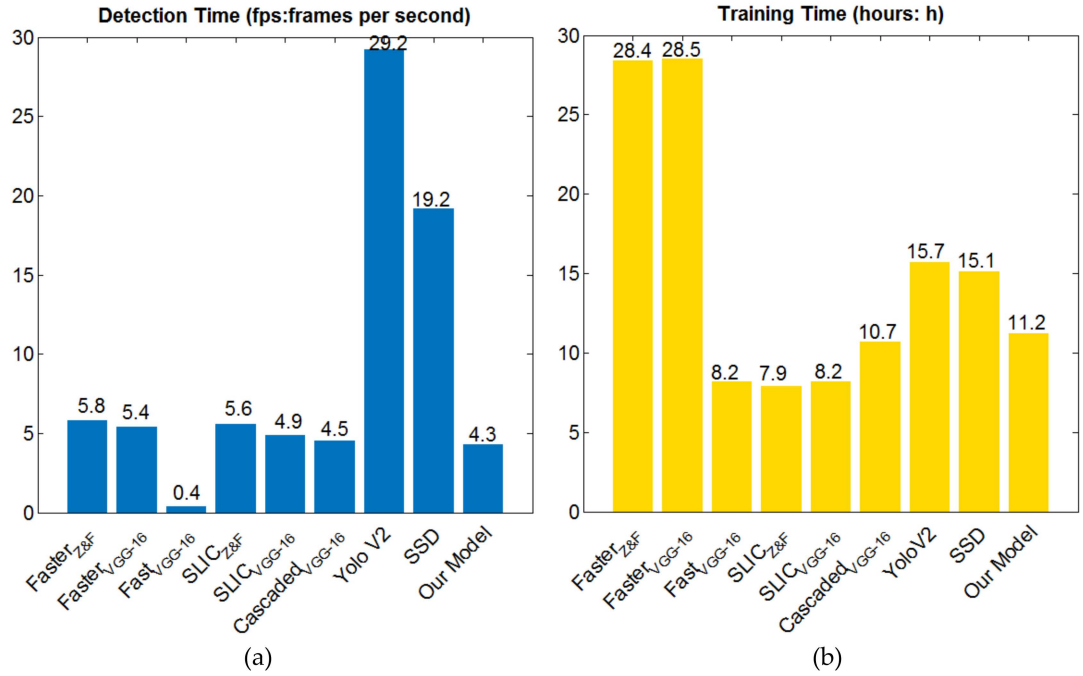


Figure 11. Detection time and training time on the VEDAI 1024 set (**a**) detection time (**b**) training time.

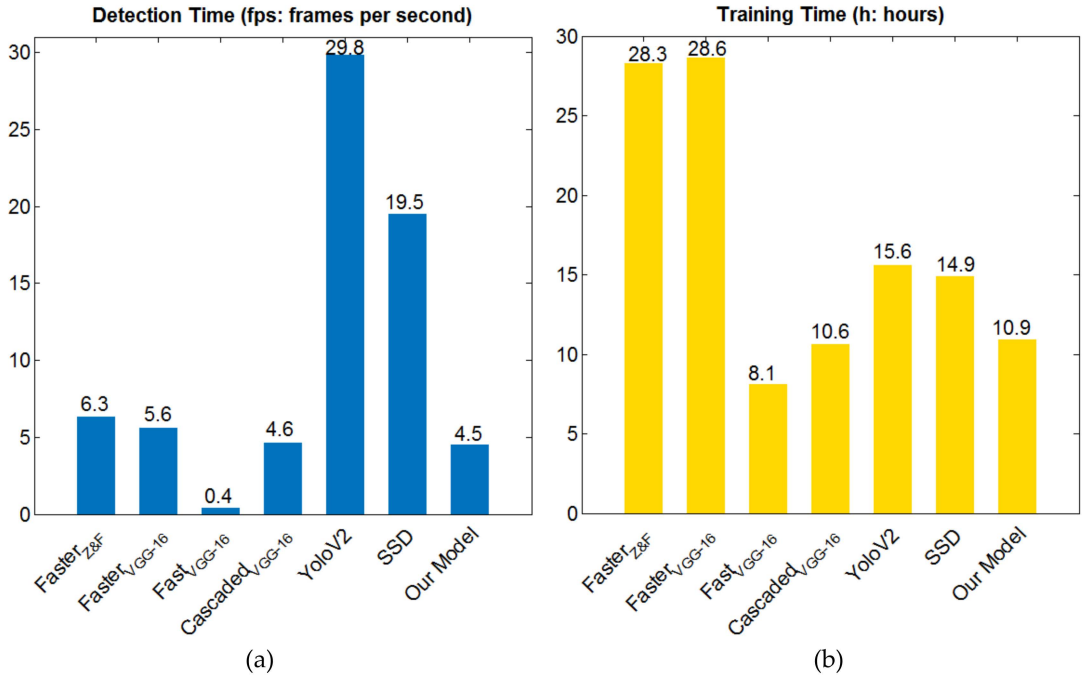


Figure 12. Detection time and training time on the VEDAI 512 set (**a**) detection time (**b**) training time.

Some detection examples of the VEDAI 1024 set are shown in Figure 13, in which the bounding boxes are denoted in red, blue and yellow. The red boxes represent the detected objects, the blue boxes represent the false detections, and the yellow ones represent the undetected objects. In Figure 13, some long trucks are detected incorrectly because the ground truths of these trucks are considerably larger than those of the generated candidate regions, which makes the IoU rate smaller than 0.5. Additionally, some objects are undetected owing to the cluttered background, which leads to missing information.

Figure 13. Examples of detection results on the VEDAI [44] dataset. The red boxes (bounding box) are the truly detected objects, the blue boxes are the false detections and the yellow boxes are the undetected objects.

4.4. OIRDS

The OIRDS project [45] provides an aerial imagery dataset captured from the existing sources of freely available imagery. The dataset includes approximately 900 labeled images, which contain approximately 1800 labeled objects that are various types of vehicles. These images have a resolution ranging from 256 × 256 pixels to 640 × 480 pixels. Additionally, the ground sampling distance is approximately 15 cm/pixel.

We performed the validation on four models, including the $\text{Faster}_{\text{VGG-16}}$, $\text{Faster}_{\text{Z\&F}}$, and $\text{Cascaded}_{\text{VGG-16}}$ models and the proposed model. Approximately 500 images were randomly selected from the OIRDS and combined with the VEDAI 1024 training set for training. The hyper parameters and settings were equivalent to those employed in the former section.

In the training phase, we defined all the labeled objects as the vehicle category and discarded the challenging examples, which were annotated with a lower probability (the annotated images in the OIRDS were tested by many raters; the object agreed to by all the raters was defined as having a probability of 100%. We discarded the objects with a probability of less than or equal to 25%). During the testing, approximately 300 images were selected for testing. Table 6 presents the comparison results, where the best results are marked in bold. In terms of the detection accuracy, the proposed model demonstrates the best recall rate (82.9%), AP (75.9) and F1-Score (0.795), which indicates that the proposed method achieves the best performance on the OIRDS. The PRC (precision–recall curve) is shown in Figure 14. The area under the orange PRC (the proposed method) is the largest. Additionally, we compared the positioning performance of several RPN-based models by considering the recall–IoU curve, as shown in Figure 15. Clearly, the proposed model yields the best recall rate at the full range ([0, 1]) of the IoU values. Several detection examples are shown in Figure 16, in which the bounding boxes are denoted in red, blue and yellow. The red boxes represent the detected objects, the blue boxes represent the false detections, and the yellow boxes represent the undetected objects. From the results shown in Figure 16, it can be noted that most of the objects are truly detected. However, there still

remain certain background regions that have an appearance similar to that of the vehicles, leading to incorrect detection. Due to the cluttered backgrounds, some objects are undetected.

Table 6. Detection results of different models on the overhead imagery research data set (OIRDS).

Evaluation Metric	$\text{Faster}_{\text{VGG-16}}$	$\text{Faster}_{\text{Z\&F}}$	$\text{Cascaded}_{\text{VGG-16}}$	SSD [41]	Yolo V2 [42]	Our Model
Recall Rate	76.5%	67.9%	79.3%	77.9%	78.5%	**82.9%**
AP	63.8%	53.6%	72.7%	65.3%	69.2%	**75.9%**
F1-Score	0.765	0.657	0.783	0.769	0.775	**0.795**

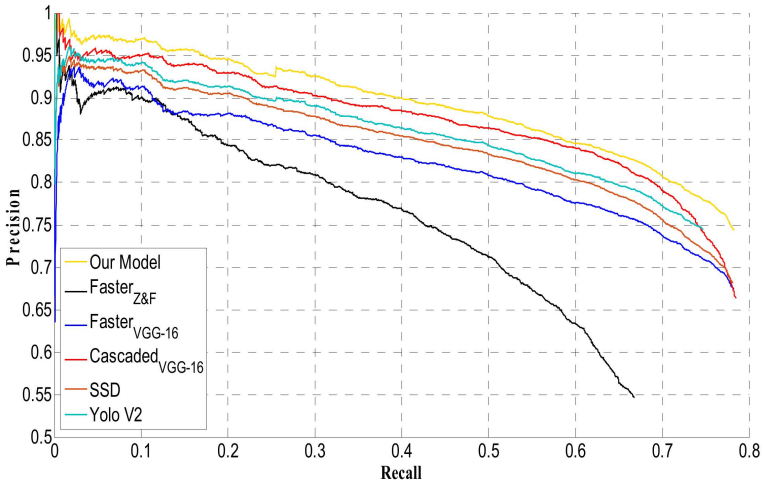


Figure 14. Precision–recall curve and comparison results on the OIRDS.

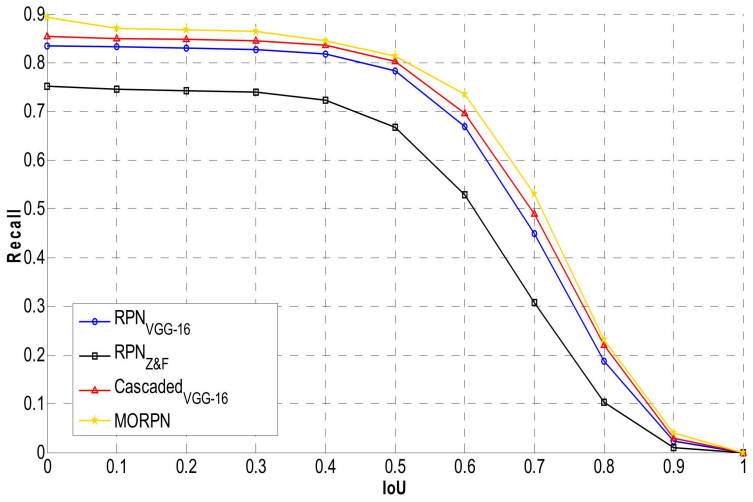


Figure 15. Recall–IoU curve and the comparison results on the OIRDS.

Figure 16. Examples of detection results on the OIRDS [45]. The red boxes (bounding box) denote the truly detected objects, the blue boxes are the false detections and the yellow boxes are the undetected objects.

5. Conclusions

To realize the detection of small-sized and oriented objects in aerial images, we propose an object detection model based on two improved CNNs: the MORPN and ODN, which are used to generate the region proposals and make decisions, respectively. Compared with the existing region proposal networks, the proposed MORPN has two improvements: first, a hierarchical architecture including multiple scales is established. Because the hierarchical architecture takes advantage of the feature map generated from the deep and shallow convolutional layers, it can generate more accurate region proposals, especially for the small-sized objects in aerial images. Second, we propose an oriented anchor scheme to address the detection problem of objects with multiple orientations because the region proposals generated based on this scheme are suitable for the objects with arbitrary orientations. Moreover, we train the ODN for classification. This ODN is combined with the MORPN to build a detection model. The results of the extensive experiments conducted indicate that the proposed model performs well in terms of both the detection accuracy and speed.

However, the proposed method still has some limitations. One of the limitations is in terms of the detection of challenging objects; for example, the detection of partially occluded or extremely small objects. The other limitation is the false alarm problem; for example, certain objects exhibit an appearance similar to that of the background, which can lead to confusion. In future work, we aim to design a deeper CNN to improve the performance of the feature extraction and object positioning. Moreover, compressing the model weights and making it suitable for a platform with less computational capacity is another meaningful research direction.

Author Contributions: Conceptualization, C.C. and J.Z.; Writing-Original Draft, C.C.; Writing-Review & Editing, J.Z.; Validation, J.Z. and Y.T.; Software, Y.T.

Funding: This research received no external funding.

Acknowledgments: The authors would appreciate the anonymous reviewers for their valuable comments and suggestions for improving this paper.

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

References

1. Guo, Z.; Du, S.; Zhao, W.; Lin, Y. A graph-based approach for the co-registration refinement of very-high-resolution imagery and digital line graphic data. *Int. J. Remote Sens.* **2016**, *17*, 4015–4034. [CrossRef]
2. Menouar, H.; Guvenc, I.; Akkaya, K. UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges. *IEEE Commun. Mag.* **2017**, *3*, 22–28. [CrossRef]
3. Gómez-Candón, D.; De Castro, A.I.; Granados, F. Assessing the accuracy of mosaics from unmanned aerial vehicle (UAV) imagery for precision agriculture purposes in wheat. *Precis. Agric.* **2013**, *1*, 44–56. [CrossRef]
4. Cheng, F.; Huang, S.; Ruan, S. Scene Analysis for Object Detection in Advanced Surveillance Systems Using Laplacian Distribution Model. *IEEE Trans. Syst. Man Cybern. Part C* **2011**, *41*, 589–598. [CrossRef]
5. Yin, J.; Liu, L.; Li, H. The infrared moving object detection and security detection related algorithms based on W4 and frame difference. The infrared moving object detection and security detection related algorithms based on W4 and frame difference. *Infrared Phys. Technol.* **2016**, *77*, 302–315. [CrossRef]
6. Trupti, M.; Jadhav, P.M.; Phadke, A.C. Suspicious object detection in surveillance videos for security applications. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 1–5.
7. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
8. Felzenszwalb, P.; Girshick, R.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef] [PubMed]
9. Lin, L.; Wang, X.; Yang, W.; Lai, J. Discriminatively Trained And-Or Graph Models for Object Shape Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 959–972. [CrossRef] [PubMed]
10. Huang, S. Discriminatively trained patch-based model for occupant classification. *IET Intell. Transp. Syst.* **2012**, *6*, 132–138. [CrossRef]
11. Cheng, G.; Han, J.; Guo, L.; Qian, X.; Zhou, P.; Yao, X.; Hu, X. Object detection in remote sensing imagery using a discriminatively trained mixture model. *ISPRS J. Photogramm. Remote Sens.* **2013**, *85*, 32–43. [CrossRef]
12. Yao, C.; Bai, X.; Liu, W.; Latecki, L. Human Detection Using Learned Part Alphabet and Pose Dictionary. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 251–266.
13. Uijlings, J.; Van de Sande, K.; Gevers, T.; Smeulders, A. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
14. Cheng, M.; Zhang, Z.; Lin, W.; Torr, P. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3286–3293.
15. Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *54*, 2189–2202. [CrossRef] [PubMed]
16. Carreira, J.; Sminchisescu, C. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1312–1328. [CrossRef] [PubMed]
17. Tian, W.; Zhao, Y.; Yuan, Y. Abing: Adjusted binarized normed gradients for objectness estimation. In Proceedings of the International Conference on Signal Processing, Hangzhou, China, 19–23 October 2014; pp. 1295–1300.
18. Hosang, J.; Benenson, R.; Dollar, P.; Schiele, B. What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 814–830. [CrossRef] [PubMed]
19. Chavali, N.; Agrawal, H.; Mahendru, A.; Batra, D. Object-Proposal Evaluation Protocol is 'Gameable'. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2578–2586.
20. Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; Malik, J. Multiscale combinatorial grouping. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 328–335.
21. Kuo, W.; Hariharan, B.; Malik, J. Deepbox: Learning objectness with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2479–2487.

22. Michael, V.; Xavier, B.; Gemma, R.; Benjamin, D. SEEDS: Superpixels extracted via energy-driven sampling. In Proceedings of the European Conference on Computer Vision, Firenze, Italy, 7–13 October 2012; pp. 13–26.
23. Vedaldi, A.; Soatto, S. Quick shift and kernel methods for mode seeking. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 705–718.
24. Veksler, O.; Boykov, Y.; Mehrani, P. Superpixels and supervoxels in an energy optimization framework. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 211–224.
25. Bergh, M.; Boix, X.; Roig, G.; Capitani, B.; Gool, L. SEEDS: Superpixels Extracted via Energy-Driven Sampling. *Int. J. Comput. Vis.* **2013**, *7578*, 1–17.
26. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
27. 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]
28. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853.
29. 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] [PubMed]
30. Zitnick, C.; Dollár, P. Edge Boxes: Locating Object Proposals from Edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
31. Cai, Z.; Fan, Q.; Rogerio, S.; Vasconcelos, F. A Unified Multi-Scale Deep Convolutional Neural Network for Fast Object Detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 354–370.
32. Li, J.; Liang, X.; Shen, S.; Xu, T.; Feng, J.; Yan, S. Scale-Aware Fast R-CNN for Pedestrian Detection. *IEEE Trans. Multimed.* **2018**, *20*, 985–996. [CrossRef]
33. Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
34. Bay, H.; Tuytelaars, T.; Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
35. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing System, Lake Tahoe, NV, USA, 3–6 December 2012.
36. Deng, J.; Berg, A.; Satheesh, S.; Su, H.; Khosla, A.; Li, F. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). Available online: http://www.image-net.org/challenges/LSVRC/2012 (accessed on 1 May 2019).
37. Zeiler, M.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. Szegedy, C.; Liu, W.; Jia, Y. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
41. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
42. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
43. Xie, H.; Wang, T.; Qiao, M.; Zhang, M.; Shan, G.; Snoussi, H. Robust object detection for tiny and dense targets in VHR aerial images. In Proceedings of the Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 6397–6401.
44. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]

45. Tanner, F.; Colder, B.; Pullen, C.; Heagy, D.; Eppolito, M.; Carlan, V.; Oertel, C.; Sallee, P. Overhead imagery research data set—An annotated data library & tools to aid in the development of computer vision algorithms. In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 14–16 October 2009; pp. 1–8.
46. Xu, Y.; Yu, G.; Wang, Y.; Wu, X.; Ma, Y. A Hybrid Vehicle Detection Method Based on Viola-Jones and HOG + SVM from UAV Images. *Sensors* **2016**, *16*, 1325. [CrossRef] [PubMed]
47. Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312. [CrossRef]
48. Qu, T.; Zhang, Q.; Sun, S. Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. *Multimed. Tools Appl.* **2016**. [CrossRef]
49. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. *Sensors* **2017**, *17*, 336. [CrossRef] [PubMed]
50. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Zou, H. Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3652–3664. [CrossRef]
51. Wang, G.; Wang, X.; Fan, B.; Pan, C. Feature Extraction by Rotation-Invariant Matrix Representation for Object Detection in Aerial Image. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 851–855. [CrossRef]
52. Zheng, Z.; Zhou, G.; Wang, Y.; Liu, Y.; Li, X.; Wang, X.; Jiang, L. A Novel Vehicle Detection Method with High Resolution Highway Aerial Image. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2338–2343. [CrossRef]
53. Ševo, I.; Avramović, A. Convolutional Neural Network Based Automatic Object Detection on Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 740–744. [CrossRef]
54. Yan, J.; Wang, H.; Yan, M.; Diao, W.; Sun, X.; Li, H. IoU-Adaptive Deformable R-CNN: Make Full Use of IoU for Multi-Class Object Detection in Remote Sensing Imagery. *Remote Sens.* **2019**, *11*, 286. [CrossRef]
55. Al-Najjar, H.; Kalantar, B.; Pradhan, B.; Saeidi, V.; Halin, A.; Ueda, N.; Mansor, S. Land Cover Classification from fused DSM and UAV Images Using Convolutional Neural Networks. *Remote Sens.* **2019**, *11*, 1461. [CrossRef]
56. Zhong, J.; Lei, T.; Yao, G. Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks. *Sensors* **2017**, *17*, 2720. [CrossRef]
57. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
58. Plaisted, D.; Hong, J. A heuristic triangulation algorithm. *J. Algorithms* **1987**, *8*, 405–437. [CrossRef]
59. Kahaki, S.; Nordin, M.; Ashtari, A.; Zahra, S. Invariant Feature Matching for ImageRegistration Application Based on New Dissimilarity of Spatial Features. *PLoS ONE* **2016**, *11*, e0149710.
60. Qin, T.; Liu, T.; Li, H. A general approximation framework for direct optimization of information retrieval measures. *Inf. Retr.* **2010**, *4*, 375–397. [CrossRef]