remote sensing

MDPI

Article

Geospatial Object Detection in High Resolution Satellite Images Based on Multi-Scale Convolutional Neural Network

Wei Guo [1,2], Wen Yang [1,2,*], Haijian Zhang [1] and Guang Hua [1]

[1] School of Electronic Information, Wuhan University, Wuhan 430072, China; weige@whu.edu.cn (W.G.); haijian.zhang@whu.edu.cn (H.Z.); ghua@whu.edu.cn (G.H.)

[2] The CETC Key Laboratory of Aerospace Information Applications, Shijiazhuang 050081, China

* Correspondence: yangwen@whu.edu.cn; Tel.: +86-27-68754367

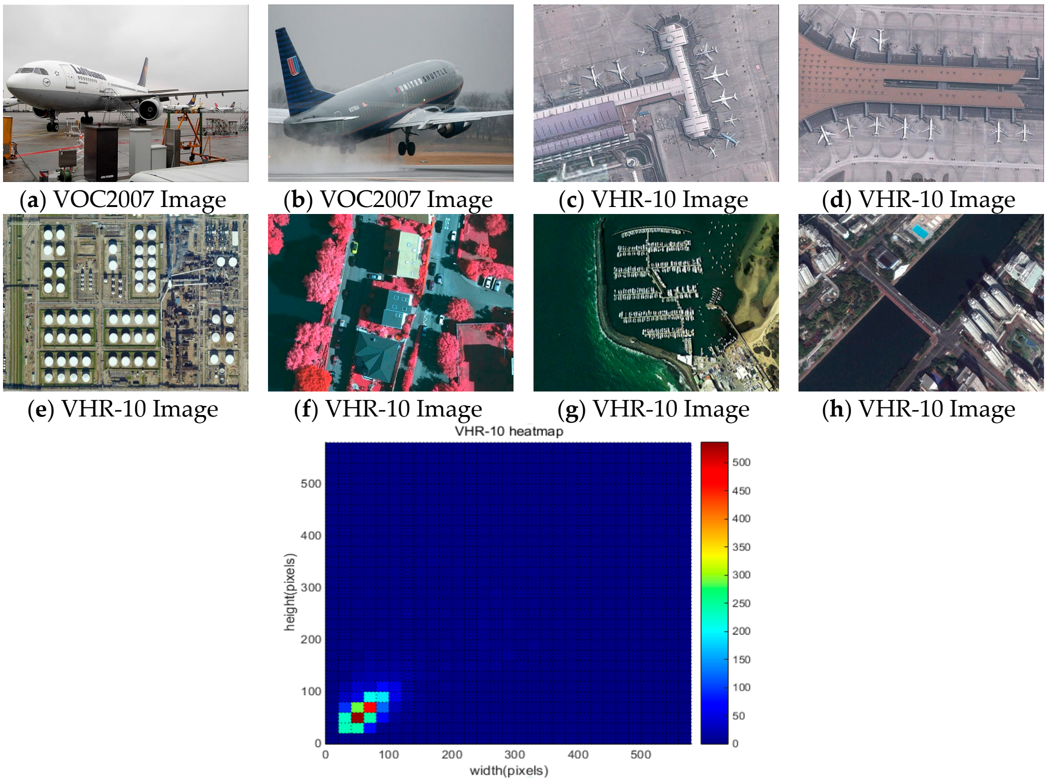
Received: 9 November 2017; Accepted: 16 January 2018; Published: 18 January 2018

Abstract: Daily acquisition of large amounts of aerial and satellite images has facilitated subsequent automatic interpretations of these images. One such interpretation is object detection. Despite the great progress made in this domain, the detection of multi-scale objects, especially small objects in high resolution satellite (HRS) images, has not been adequately explored. As a result, the detection performance turns out to be poor. To address this problem, we first propose a unified multi-scale convolutional neural network (CNN) for geospatial object detection in HRS images. It consists of a multi-scale object proposal network and a multi-scale object detection network, both of which share a multi-scale base network. The base network can produce feature maps with different receptive fields to be responsible for objects with different scales. Then, we use the multi-scale object proposal network to generate high quality object proposals from the feature maps. Finally, we use these object proposals with the multi-scale object detection network to train a good object detector. Comprehensive evaluations on a publicly available remote sensing object detection dataset and comparisons with several state-of-the-art approaches demonstrate the effectiveness of the presented method. The proposed method achieves the best mean average precision (mAP) value of 89.6%, runs at 10 frames per second (FPS) on a GTX 1080Ti GPU.

Keywords: high resolution satellite images; geospatial object detection; object proposal network; object detection network

1. Introduction

The rapid development of remote sensing technologies has created a large amount of high-quality satellite and aerial images for research and investigation. High resolution satellite (HRS) images, compared to ordinary low- and medium-resolution images, have some special properties: (1) the structure of geospatial objects is clear; (2) the spatial layout is distinct; and (3) the entire image is a collection of multi-scale objects [1]. Automated object detection in HRS images is a core requirement for large range scene understanding and semantic information extraction [2]. Over the past decades, considerable efforts have been made to develop various methods for the detection of different types of objects in satellite and aerial images [3], such as buildings [4,5], storage tanks [6,7], vehicles [8,9], and airplanes [10–12]. Object detection in HRS images determines whether there are one or more objects belonging to the classes we are looking for and locates the position of each object using a bounding box. Learning efficient image representations is the core task for object detection [13]. To solve the object detection problem, the traditional methods based on either coding of handcrafted features or unsupervised feature learning can only generate shallow to middle features with limited representative ability [14,15]. Recently, with the rapid development of convolutional neural network (CNN), several design variations using region based CNN have generated the state-of-the-art performance against traditional multi-class object detection benchmarks [16–20]. These benchmark datasets typically present target objects with "friendly" or dominant scales because those images in a large pool of available images and objects with significant scales, could be more easily selected [21]. Unlike objects on these benchmark datasets, objects on HRS images are much smaller, including fixed shape objects (e.g., airplanes, ships, vehicles, etc.) and diverse shape objects (e.g., harbors, bridges, etc.) that have vastly different scales, which makes object detection in HRS images a very difficult problem. Besides, large variations in the visual appearance of objects caused by viewpoint variation, resolution variation, occlusion, background clutter, illumination, shadow, etc., cause much larger challenges for object detection in HRS images [3]. Figure 1 gives the object scale comparison of the Pascal Visual Object Classes 2007 (VOC2007) benchmark with Northwestern Polytechnical University very-high-resolution 10-class (NWPU VHR-10) benchmark and the scale distribution of NWPU VHR-10 benchmarks. We can find that airplanes on the VOC2007 images occupy a dominant position, while objects on the VHR-10 benchmark images are much smaller, with significant differences among them. In fact, most objects have sizes less than 150 pixels, while very small objects such as vehicles as well as large objects such as track fields make up a large proportion of objects. Despite the progress made in traditional many-class object detection benchmarks, the complex object distribution makes it difficult to directly deal with the object detection task in HRS images [22].


(**a**) VOC2007 Image (**b**) VOC2007 Image (**c**) VHR-10 Image (**d**) VHR-10 Image

(**e**) VHR-10 Image (**f**) VHR-10 Image (**g**) VHR-10 Image (**h**) VHR-10 Image

(**i**) VHR-10 Scale Distribution

Figure 1. Scale Comparison of VOC2007 Image with VHR-10 Image and Scale Distribution of NWPU VHR-10 Benchmark.

Object detection in HRS images has been extensively studied over recent years [23]. The existing methods can be generally divided into four main categories: template matching-based methods, knowledge-based methods, object based image analysis (OBIA)-based methods, and machine learning-based methods [3]. Template matching-based methods can be further divided into two classes, i.e., rigid template matching and deformable template matching. Such types of methods usually have two steps: template generation and similarity measure [4,24]. For knowledge-based methods, the most widely leveraged types of knowledge are geometric and context [25–27]. OBIA-based methods mainly involve two steps: image segmentation and object classification [28]. For machine learning-based methods, three processing steps are needed: feature extraction, feature fusion dimension reduction, and classifier training [29,30]. Taking the advantages of the powerful feature extraction and classification techniques in machine learning area, object detection tasks have been formulated as feature extraction and classification problems, whose results have been shown to be promising. In the past decade, various feature extraction approaches have been developed for the representation of objects [31]. Among them, histogram of oriented gradients (HOG) feature, local binary pattern (LBP) feature, bag of words (BoW) feature and sparse coding based features are four widely used features and have greatly advanced the development of object detection. HOG feature was first proposed by Dalal and Triggs, since its edges or gradient structure describes the characteristics of local shape and is very appropriate for human detection [32]. LBP is an operator used to describe the local texture features of an image [33]. It has remarkable advantages such as rotation invariance and gray scale invariance. Face recognition with LBP features has shown superiority and efficiency over some other methods [34]. BoW feature represents the image of a scene by a collection of local regions, denoted as code-words obtained by unsupervised learning, and each region is represented as a part of a "theme" [35,36]. It has been widely used in geospatial object detection with excellent performance. Sparse coding is a kind of unsupervised method for learning sets of over-completed bases to represent data efficiently. Leveraging the mature theory in compressive sensing, sparse coding has been widely used in remote sensing image analysis, such as image de-noising, image classification and object detection, yielding promising performance [37–40]. Besides feature extraction, the subsequent classification is also very important in the process of object detection. A classifier can be trained using many different approaches by minimizing the misclassification error on the training dataset, including support vector machine (SVM), k-nearest neighbors (KNN), random forest (RF) and so on [41–43].

With the recent rapid development of deep learning, CNNs have become a new approach for feature representation and greatly improved the performance of object detection [44]. Current CNN-based object detection algorithms could be roughly divided into two streams: the region-based CNN (R-CNN) methods (e.g., R-CNN, Fast R-CNN and Faster R-CNN) and the region-free methods (e.g., you only look once (YOLO) and single shot multi-box detector (SSD)) [16–20]. For each input image, R-CNN firstly extracts around 2000 region proposals using selective search algorithm. Then, it computes features for each proposal using a large CNN, followed by classifying each region using class-specific linear SVMs [16]. Since CNN could extract deep features, the R-CNN outperforms other handcrafted features-based methods by a large margin on the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013). Fast R-CNN builds on R-CNN, but replaces the SVM classifier with a region of interest (RoI) pooling layer and some fully connected (FC) layers to classify region proposals and adjust the position of region proposals, which not only improves training and testing speed, but also increases detection accuracy [17]. Faster R-CNN uses region proposal network (RPN) to generate high-quality region proposals. Then, these region proposals are used by Fast R-CNN for detection. RPN shares full image convolutional features with the detection network, thus enabling nearly cost-free region proposal generation [18]. The region-based methods utilize a classifier to perform object detection. By contrast, the region-free methods such as YOLO frame object detection as a regression problem so that a single network could predict bounding boxes and associated classes directly [19]. This method is extremely fast. SSD also considers object detection as a regression problem, but small convolutional filters are applied to feature maps to predict category scores and box offsets rather than fully connected layers. Besides, feature maps of different scales are used to make predictions of multi-scales, so the detection accuracy is greatly improved compared with YOLO [20].

It is important to note that these CNN-based object detection methods were designed somewhat specifically for general object detection challenges, which is not suitable for geospatial object detection in HRS images [45]. Besides, to handle the problem of multi-scale objects and small objects, some methods like Fast R-CNN and Faster R-CNN achieve this by up-sampling the input image at training phase or testing phase. It significantly increases the memory occupation and processing time. In this paper, we propose a multi-scale CNN for geospatial object detection. The main contributions of this paper are summarized as follows:

(1) A unified multi-scale CNN is proposed for geospatial object detection in HRS images. Objects with extremely different scales could be more efficiently detected than the state-of-the-art methods.
(2) A modified base network is designed to generate feature maps with high semantic information at each layer. Since feature maps of different layers can be assigned to objects of specific scales, the detection performances at all scales are correspondingly improved.
(3) An optimized object proposal network is presented to produce better object proposals. By adding multi-scale anchor boxes to multi-scale feature maps, the network could generate object proposals exhaustively, which could improve the recall rate of the detection. By adding proposal score layers behind the multi-scale feature maps, the network could suppress most of the negative samples, which could improve the precision of the detection.

The proposed method is evaluated on a publicly available remote sensing object detection dataset and then compared with several state-of-the-art approaches. The experimental results demonstrate the effectiveness and superiority of our method.

The rest of this paper is organized as follows. Section 2 presents the methodology of our multi-scale CNN, which consists of the multi-scale object proposal network and the multi-scale object detection network. Section 3 presents the dataset description and experimental details. Sections 4 and 5 present the analysis of the experimental results and a discussion of the results, respectively. Finally, the conclusions are drawn in Section 6. Important terms and their abbreviations are provided in Table 1.

Table 1. Important terms and their abbreviations.

HRS	High Resolution Satellite
CNN	convolutional neural network
FPS	frames per second
VOC2007	the Pascal Visual Object Classes 2007
NWPU VHR-10	Northwestern Polytechnical University very-high-resolution 10-class
OBIA	object based image analysis
HOG	histogram of oriented gradients
LBP	local binary pattern
BoW	bag of words
SVM	support vector machine
KNN	k-nearest neighbors
RF	random forest
R-CNN	region-based CNN
YOLO	you only look once
SSD	single shot multi-box detector
ILSVRC2013	the ImageNet Large Scale Visual Recognition Challenge 2013
RoI	region of interest
FC	fully connected
RPN	region proposal network
FCN	fully convolutional network
IoU	intersection-over-union
AP	average precision
PRC	precision-recall curve
mAP	mean AP
TP	true positive
FP	false positive
SSCBoW	spatial sparse coding BoW
COPD	collection of part detectors
RICNN	rotation-invariant CNN
NMS	non-maximum suppression

2. Methodology

Figure 2 provides an overview of the technical workflow, which displays the components of the multi-scale CNN. The proposed network consists of a multi-scale object proposal network and a multi-scale object detection network, both of which share a multi-scale base network for feature map generation. Details of these networks are provided in the following content.

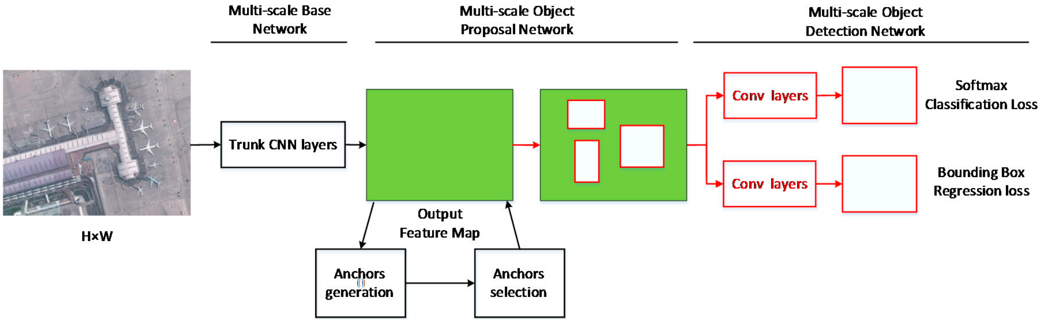

Multi-scale Base Network
Multi-scale Object Proposal Network
Multi-scale Object Detection Network
H×W
Trunk CNN layers
Output Feature Map
Anchors generation
Anchors selection
Conv layers
Conv layers
Softmax Classification Loss
Bounding Box Regression loss

Figure 2. Architecture of the unified multi-scale CNN.

2.1. Shared Multi-Scale Base Network

The coverage of multi-scale is an important problem for object detection [46]. R-CNN based methods (such as R-CNN, Fast R-CNN and Faster R-CNN) choose the output of the last layer as reference set of feature maps [3–5]. However, the single scale feature maps have a fixed receptive field of the input image, which can be mismatched to small or large objects [47]. SSD uses feature maps from multiple layers of the CNN [20]. As the semantic information in the low-layer is shallow, the detection result for small objects is relatively poor. To produce feature maps that have strong semantics at all scales, we propose a new shared multi-scale base network which combines high semantic information from higher layers with fine details from lower layers, as shown in Figure 3. The proposed base network produces feature maps through multiple branches, starting from the last layer of bottom-up feedforward network, which has very high semantic information but poor localization performance due to the coarseness of the feature maps [48,49]. Then, the feature maps of the last layer are transmitted back by the top-down network. Bottom-up feature maps at middle layers are combined with the top-down feature maps to produce final feature maps via lateral connections [50,51]. As the final feature maps are composed of feature maps from the top-down network and the bottom-up network, it can capture both pertinent fine details and high-level semantic information. We use modified VGG-16 as the bottom-up network, which is a standard architecture used for high quality image classification [52]. Because of the large range of HRS imagery and small size of geospatial objects on it, we do not use feature maps after conv1, conv2 and conv3 for lateral connections due to its weak semantic information and large memory overhead. Moreover, we discard feature maps after conv6, conv7, conv8 and conv9 because their feature maps are too small to distinguish objects on them. To ensure the number of feature maps with different sizes, we use feature maps after conv4, conv5, fc6 and fc7 to produce final feature maps. To use arbitrary size input images, we convert fc6 and fc7 to convolutional layers similar to the network architecture used in fully convolutional network (FCN) as the fc layers can be viewed as convolutional with kernels that cover the entire input regions [53]. After the bottom-up feedforward network, several feature maps with different scales are produced. Then, in the top-down layers and lateral connection layers, a de-convolutional layer and a 3×3 convolutional kernel are applied separately to guarantee the outputs of the top-down layers and lateral connection layers have the same size and dimension. Then, the two corresponding feature maps are merged by element-wise addition to produce the final multi-scale feature maps. In detail, feature maps from four different branches are produced. For convenience, we denote these feature maps as {F4, F5, F6, F7} for conv4, conv5, fc6, and fc7 outputs, which can be seen in Figure 3. As the stride of each max pooling layer in VGG-16 is 2, these output feature maps have receptive field of {8, 16, 32, 64} pixels with respect to the input image, with a size of $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, $\frac{H}{32} \times \frac{W}{32}$ and $\frac{H}{64} \times \frac{W}{64}$, respectively.

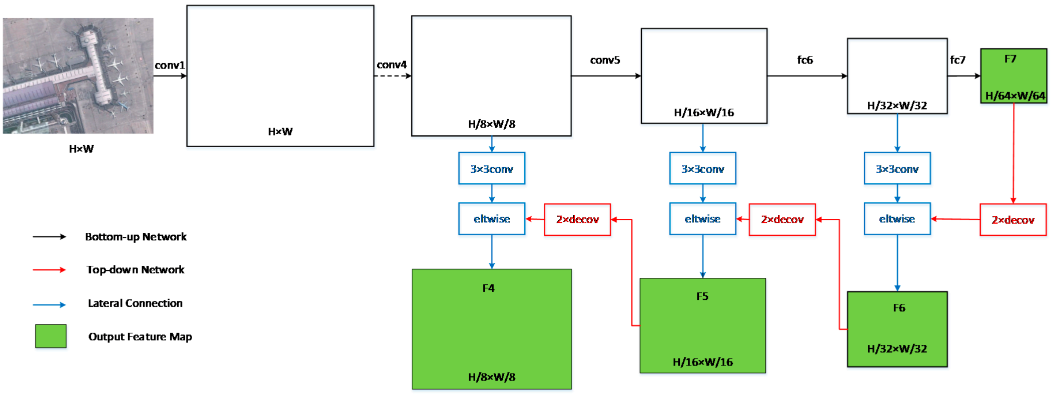


Figure 3. Architecture of the shared multi-scale base network.

2.2. Multi-Scale Object Proposal Network

Figure 4 gives the architecture of the RPN, which introduces novel anchor boxes that serve as object proposals at multiple scales and aspect ratios [5]. An RPN inputs feature maps of the same layer and outputs a set of rectangular anchor boxes, each with two objectness scores that estimate the probability of being an object or not and four coordinates encoding the position of the object. At each sliding-window location, RPN simultaneously predicts multiple region proposals, where the number of predicted region proposals is denoted as k. In Faster R-CNN, there are three scales and three aspect ratios, leading k = 9. anchor boxes at each sliding position. As shown in Section 2.1, we have obtained multi-scale feature maps {F4, F5, F6, F7} with different receptive field of {8, 16, 32, 64} pixels. There is no need to let an anchor box with a big receptive field match to a small object. Anchor boxes of different scales can be assigned to feature maps of different scales. Thus, SSD uses a single scale anchor box at feature maps of different scales. Moreover, SSD imposes five different aspect ratios for the default sliding position rather than three in the Faster R-CNN method. The experimental result of SSD shows that using a variety of default anchor box shapes achieves better prediction. However, as we can see that Faster R-CNN using a single scale feature map with multi-scale anchor boxes could detect most of the objects on images. In other words, a single scale feature map could be responsive for multi-scale objects with the help of multi-scale anchor boxes. Thus, we add multi-scale anchor boxes to multi-scale feature maps to improve the accuracy of detection. We tried four different options, as shown in Table 2. $\text{RF}_{\text{min}}(3 \times 8 = 24$ pixels) is the receptive field of sliding window in F4 feature maps. Table 3 shows the detection accuracy of the four different options. We can see that Option 2 is the best.

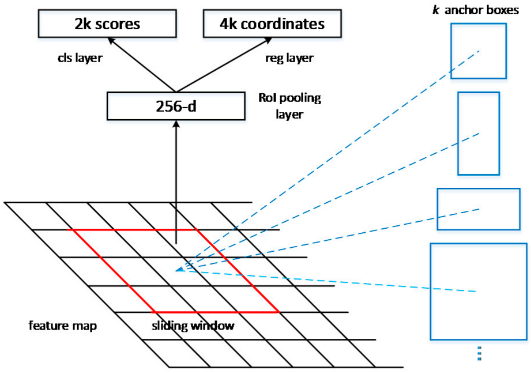


Figure 4. Architecture of the region proposal network (RPN).

Table 2. Options for Anchor boxes.

Receptive Field of Anchor Boxes	F4	F5	F6	F7
1 (Single Scale)	RF_{min}	$3RF_{min}$	$5RF_{min}$	$7RF_{min}$
2 (Two Scale)	RF_{min}, $2RF_{min}$	$3RF_{min}$, $4RF_{min}$	$5RF_{min}$, $6RF_{min}$	$7RF_{min}$, $8RF_{min}$
3 (Three Scale)	RF_{min}, $2RF_{min}$, $3RF_{min}$	$3RF_{min}$, $4RF_{min}$, $5RF_{min}$	$5RF_{min}$, $6RF_{min}$, $7RF_{min}$	$7RF_{min}$, $8RF_{min}$, $9RF_{min}$
4 (Four Scale)	RF_{min}, $2RF_{min}$, $3RF_{min}$, $4RF_{min}$	$3RF_{min}$, $4RF_{min}$, $5RF_{min}$, $6RF_{min}$	$5RF_{min}$, $6RF_{min}$, $7RF_{min}$, $8RF_{min}$	$7RF_{min}$, $8RF_{min}$, $9RF_{min}$, $10RF_{min}$

Table 3. The Mean AP values of the four different options.

Options	1	2	3	4
Mean AP	86.2%	**89.6%**	87.2%	77.9%

The architecture of the proposed multi-scale object proposal network is shown in Figure 5. We consider anchor boxes with two scales and five aspect ratios at each feature map, resulting in a total number of about 8000 anchor boxes. However, only a small part of the anchor boxes contains objects. This leads to a significant imbalance between the positive and negative samples. It is difficult for the object detector to suppress most of the negative samples and give a reasonable score and an appropriate position for the positive samples at the same time. Thus, we train a classifier to help the object detector suppress the negative samples. Instead of using all the anchor boxes for detection, we add a proposal score layer behind the feature maps to execute positive samples selection. A 3×3 convolutional layer followed by a softmax function layer is used to predict the proposal score of each anchor box. During training phase, an anchor box is assigned a positive label 1 if it has the highest Intersection-over-Union (IoU) for a given ground truth bounding box or an IoU over 0.5 with any ground-truth bounding box, and a negative label 0 if it has IoU lower than 0.3 for all ground truth bounding box. All the positive samples are used for backward propagation. Then, we randomly select the negative samples for backward propagation until the ratio between positive and negative samples is at least 1:3 [17,54]. As 0.5 is the median of the output of the softmax function, we use 0.5 as the threshold. Anchor boxes whose proposal scores are higher than 0.5 will be used for detection.

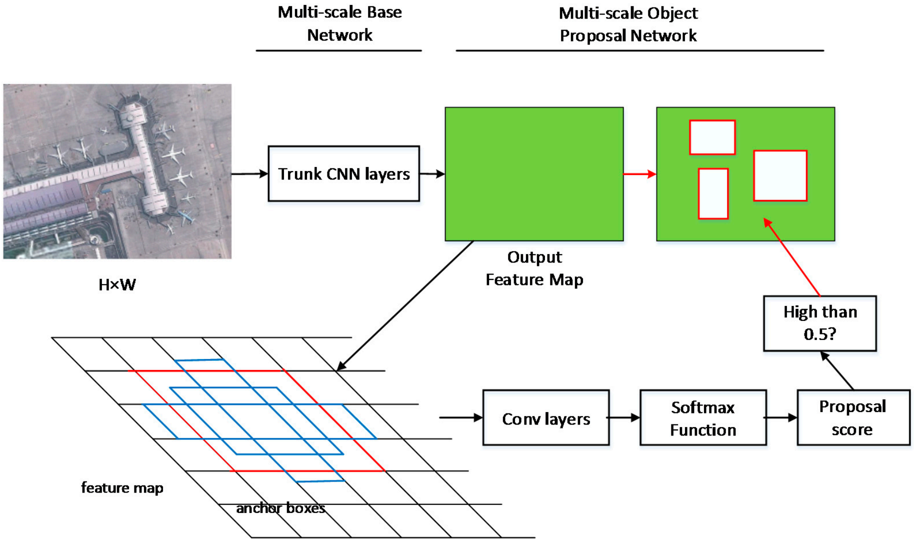


Figure 5. Architecture of the multi-scale object proposal network.

During the training process, the parameters W of the multi-scale object proposal network are learned from a set of training samples $S = \{(X_i, Y_i)\}_{i=1}^{N}$, where X_i is a training image and $Y_i = (y_i, b_i)$ is the combination of its class label $y_i \ \epsilon \ \{0, 1\}$ and corresponding ground truth bounding boxes coordinates $b_i = \left(b_i^x, b_i^y, b_i^w, b_i^h\right)$. This is achieved with a multi-task loss

$$L_{pro}(W) = \sum_{f=1}^{4} \alpha_f l^f(X_i, Y_i|W), \tag{1}$$

where f stands for the feature map used to produce anchor boxes, α_f stands for the weight of object proposal loss l^f. The loss for each feature map is

$$l(X_i, Y_i|W) = \sum_{i \ \epsilon \ S} L_{pro}(p(X_i), y_i), \tag{2}$$

where $p(X_i)$ = $(p_0(X_i),\ p_1(X_i))$ is the anchor box's probability distribution over being an object or background. $L_{pro}(p(X_i), y_i)$ is the cross-entropy loss.

2.3. Multi-Scale Object Detection Network

The positive and negative samples obtained in Section 2.2 could be used to train our multi-scale object detection network, shown in Figure 2. We use two 3 × 3 convolutional layers on feature maps to obtain the classification and bounding box regression result, which can be changed into a hierarchy of convolutional layers or more advanced blocks like residual or inception units. For simplicity, we only use two 3 × 3 convolutional layers. We have training samples $S_f = \{S_f^+, S_f^-\}$ for each feature map. The subscript f indicates the feature map used for training. $S_f^+ = \{S_f^{+1}, S_f^{+2}, S_f^{+3}, \cdots, S_f^{+M}\}$, $S_f^- = \{S_f^{-1}, S_f^{-2}, S_f^{-3}, \cdots, S_f^{-N}\}$. M is the number of positive samples and N is the number of negative samples in feature map f. For each sample $S_f^i = \{(X_i, Y_i)\}_{i=1}^{M+N}$, X_i is the input image of sample i and $Y_i = (y_i, b_i)$ is the combination of class label $y_i \epsilon \{0, 1, 2, 3, \cdots, K\}$ (0 stands for negative samples and K is the number of classes) and coordinates b_i of ground truth bounding boxes. The loss for learning the object detection parameters can be expressed as

$$L_{obj}(W) = \sum_{f=1}^{4} \sum_{i=1}^{M+N} \beta_f l^f(X_i, Y_i|W), \tag{3}$$

where β_f stands for the weight of object detection loss l^f. The loss for each sample is

$$l(X_i, Y_i|W) = L_{cls}(p(X_i), y_i) + \mu[y_i > 0] L_{loc}(b_i, b), \tag{4}$$

where $p(X_i) = (p_0(X_i),\ p_1(X_i),\ p_2(X_i), \cdots,\ p_K(X_i))$ is sample's probability distribution over each class, μ is a trade-off coefficient for balancing the weight of classification loss and bounding box regression loss, $L_{cls}(p(X_i), y_i)$ is the cross-entropy loss, b is the regressed bounding box, and $L_{loc}(b_i, b)$ is a smoothed loss. In accordance with the above definitions, the overall loss function for our method can be given by

$$L(W) = L_{pro}(W) + L_{obj}(W) \tag{5}$$

By stochastic gradient descent, we can learn the optimal parameters W^{opt} for the whole network.

3. Experiments

Remote Sensing datasets from Google Map have received extensive research attention in the recent years and are recognized as a valid source for remote sensing research [55]. To evaluate the performance of the proposed multi-scale CNN, we performed ten-class object detection experiments on a publicly available dataset: NWPU VHR-10 dataset acquired from Google Earth [56]. The dataset description, evaluation metrics, baseline methods and implementation details are described in this section.

3.1. Dataset Description

The NWPU VHR-10 dataset is a ten-class geospatial object detection dataset used for multi-class object detection. This dataset contains airplanes, ships, storage tanks, baseball diamonds, tennis courts, basketball courts, ground track fields, harbors, bridges, and vehicles. It contains a total of 800 very high resolution satellite images, with 715 images acquired from Google Earth with a resolution of 0.5–2.0 m, and 85 pan-sharpened color infrared images with a resolution of 0.08 m. Two image sets are contained in this dataset: a positive dataset, with 650 images each containing at least one target to be detected, and a negative dataset of 150 images, without any targets of the given classes to be detected. From the positive image set, 757 airplanes, 302 ships, 655 storage tanks, 390 baseball diamonds, 524 tennis courts, 150 basketball courts, 163 ground track fields, 224 harbors, 124 bridges, and 477 vehicles were manually annotated with bounding boxes used for ground truth. For the comparison with baseline method, we divide the positive dataset into 20% for training, 20% for validation and 60% for testing, namely 130 images for training, 130 images for validation and 390 images for testing.

3.2. Evaluation Metrics

We incorporate the widely used average precision (AP) and precision-recall curve (PRC) to quantitatively evaluate the performance of the proposed multi-scale CNN. The AP computes the average value of the precision over the interval from recall = 0 to recall = 1, i.e., the area under the PRC; hence, the higher the AP, the better the performance [8]. In addition, mean AP (mAP) computes the average value of all the AP values for all the classes. The precision metric measures the proportion of detections that are true positives, and the recall metric measures the proportion of positives that are correctly detected. The precision and recall metrics can be formulated as follows:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{6}$$

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

A detecting anchor box is considered to be true positive (TP) if the area IoU between it and the ground truth is larger than 0.5, otherwise, it will be considered as false positive (FP). In addition, if the area overlap ratio between several detecting anchor boxes and the ground truth are bigger than 0.5, only the bounding box with the largest area IoU is considered as TP, others are considered as FP.

3.3. Baseline Methods

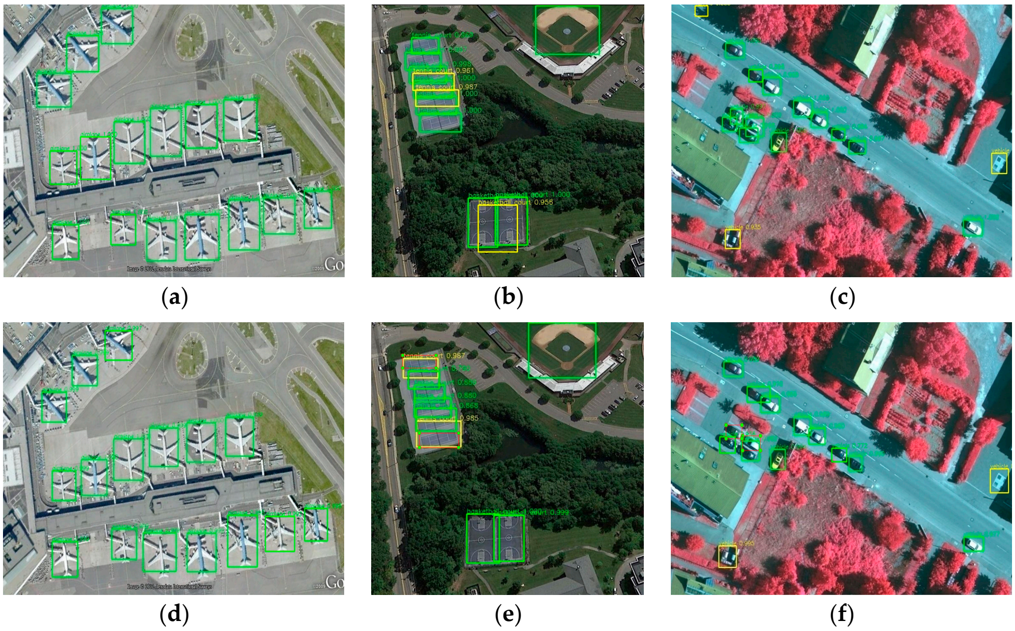
To evaluate the proposed multi-scale CNN quantitatively, we compared it with three state-of-the-art methods and four state-of-the-art CNN-based methods: (1) the BoW feature based method in which each image region is represented as a histogram of visual words generated by the k-means algorithm [57]; (2) the spatial sparse coding BoW (SSCBoW) feature based model in which visual words are generated by the sparse coding algorithm [36]; (3) the collection of part detectors (COPD) based method which is composed of 45 seed-based part detectors trained in HOG feature space. Each part detector is a linear SVM classifier corresponding to a particular viewpoint of a particular object class, hence, the collection of them provides an approximate solution for rotation-invariant object detection [58]; (4) a transferred CNN model fine-tuned from AlexNet, which is used as a universal CNN feature extractor [59]; (5) a rotation-invariant CNN (RICNN) model which considers rotation-invariant information with a rotation-invariant layer and other fine-tuned layers [59]; (6) the SSD model with an input image size of 512 × 512 pixels; and (7) the faster R-CNN model with an input image size about 1000 × 600 pixels.

3.4. Implementation Details

Our method is trained end-to-end on the NWPU VHR-10 trainval dataset, and tested on NWPU VHR-10 test dataset. To make the model more robust to various input object sizes and shapes, each training image is randomly sampled by one of the following options: (1) using the original/flipped input image; and (2) randomly sampling a patch whose edge length is 0.4, 0.5, 0.6, 0.7, 0.8 or 0.9 of the original image. We keep the patch only if at least one object's center is in the sampled patch. During the training phase, we initialize the same parameters with VGG-16 by the model pre-trained with ImageNet dataset. For other newly added layers, we initialize the parameters by drawing weights from a zero-mean Gaussian distribution with standard deviation of 0.01. The learning rate is 10^{-3} for the first 30,000 iterations and then decayed to 10^{-4} for other 10,000 iterations. We use a weight decay of 0.0005 and a momentum of 0.9. We resize the input image so that it has an input size of 320 × 320 at the training stage and an input size of 512 × 512 at the testing stage as detection often requires fine-grained visual information [19]. The hyper-parameters α_f, β_f, and μ in Section 2 are set to 1 in all experiments. We adopt stochastic gradient descent with a mini-batch of 10 images.

4. Results

Figure 6 shows airplanes, tennis courts, basketball courts, baseball diamonds and vehicles detected by using our method, Faster R-CNN and SSD. The predicted bounding boxes that match the ground truth bounding boxes with IoU > 0.5 are plotted in green color, while the false alarms and missing targets are plotted in yellow and red color, respectively. Our method is better in the given scenes, since it successfully detects all the objects with a small number of false alarms, while Faster R-CNN detects most of the objects with a small number of false alarms and missing targets, and SSD with many false alarms and missing targets. The detection results for vehicles and tennis courts show that our method could generate better bounding boxes that cover most of the objects even when they are closely aligned and with a small size. The detection results for airplanes and vehicles show that our method could exclude most of the false bounding boxes and detects with a small number of false alarms. This is because our method could generate better bounding boxes that cover most of the objects by using multi-scale base network. Moreover, our method could suppress most of the false alarms by using positive samples selecting.



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

(**d**) (**e**) (**f**)

Figure 6. *Cont.*

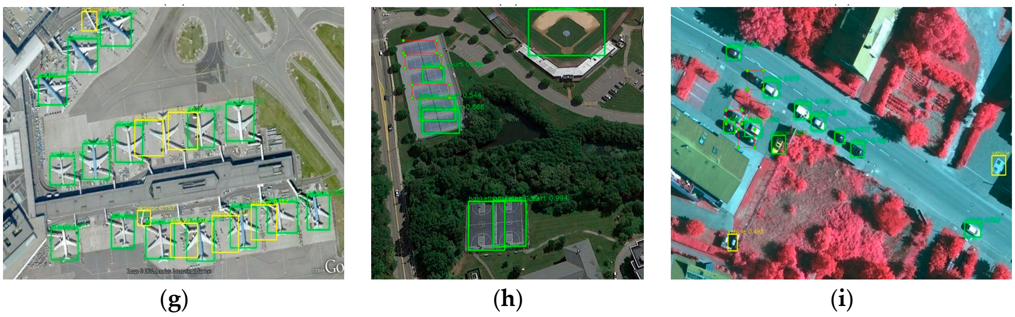

(**g**) (**h**) (**i**)

Figure 6. Detection results using our method, Faster R-CNN and SSD. The first row are results of our method; the second row are results of Faster R-CNN; the third row are results of SSD.

More results of our method on images from the VHR-10 test dataset are shown in Figure 7. It can be seen that objects with extremely small scales could also be detected well, e.g., the storage tanks in Figure 7b and the vehicles in Figure 7f. Besides, the detection performance for other objects, like airplanes, ships and baseball diamonds are also very promising. However, when objects are small and closely aligned, there may be some false positives, as shown in Figures 6b and 7b,g . This is because we add multi-scale anchor boxes to multi-scale feature maps to improve the accuracy of detection. Although our method could cover most objects, there exist a small number of repeated bounding boxes that cannot be suppressed by non-maximum suppression (NMS) operator. If we decrease the threshold of non-maximum suppression operator, more repeated bounding boxes will be restrained, but at the same time, some small objects will also be missed. To solve this problem, we replace the traditional NMS operator by a Soft NMS operator. Figure 8 shows some detection results using our method with NMS and Soft NMS, respectively. It could be seen that these false alarms are suppressed.

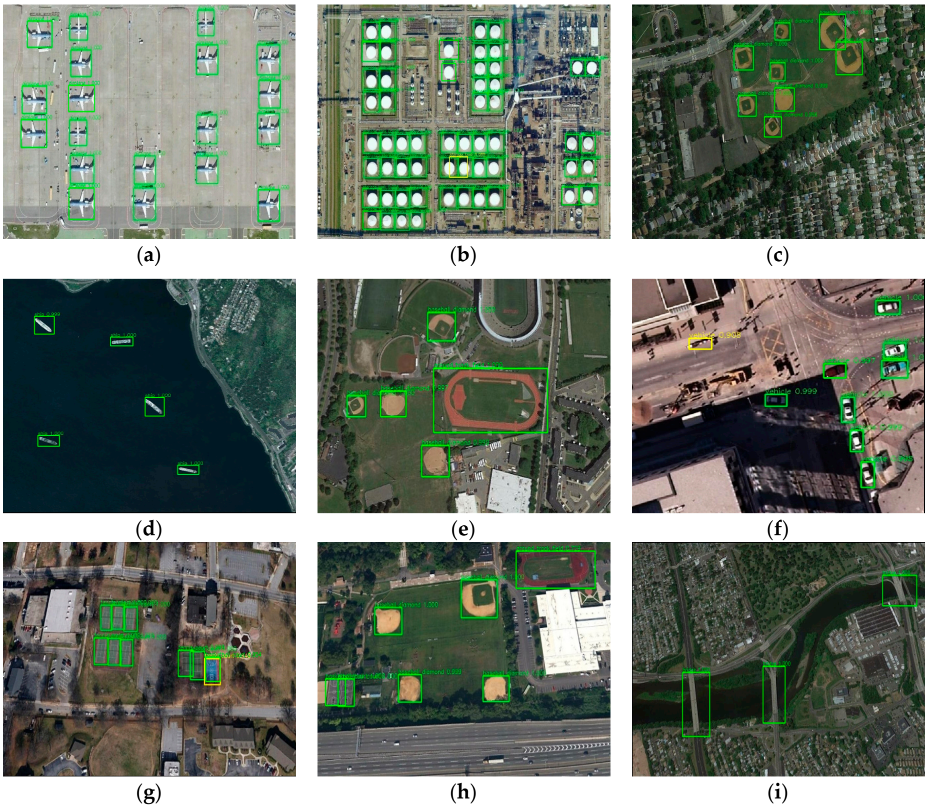
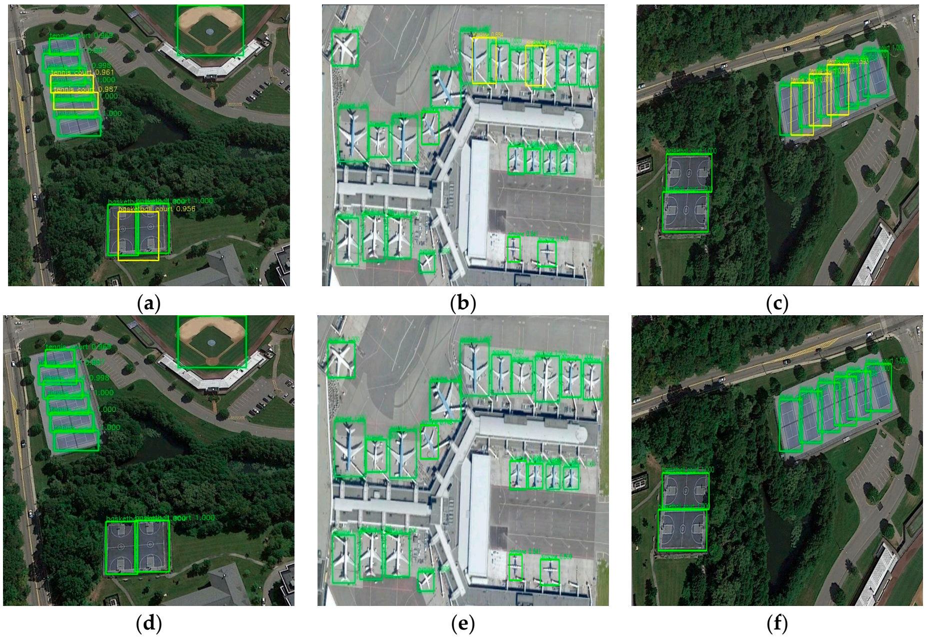

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)
(**g**) (**h**) (**i**)

Figure 7. Detection results using our method.



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

(**d**) (**e**) (**f**)

Figure 8. Detection results using our method with NMS and Soft NMS. The first row are results of our method with NMS; the second row are results of our method with Soft NMS.

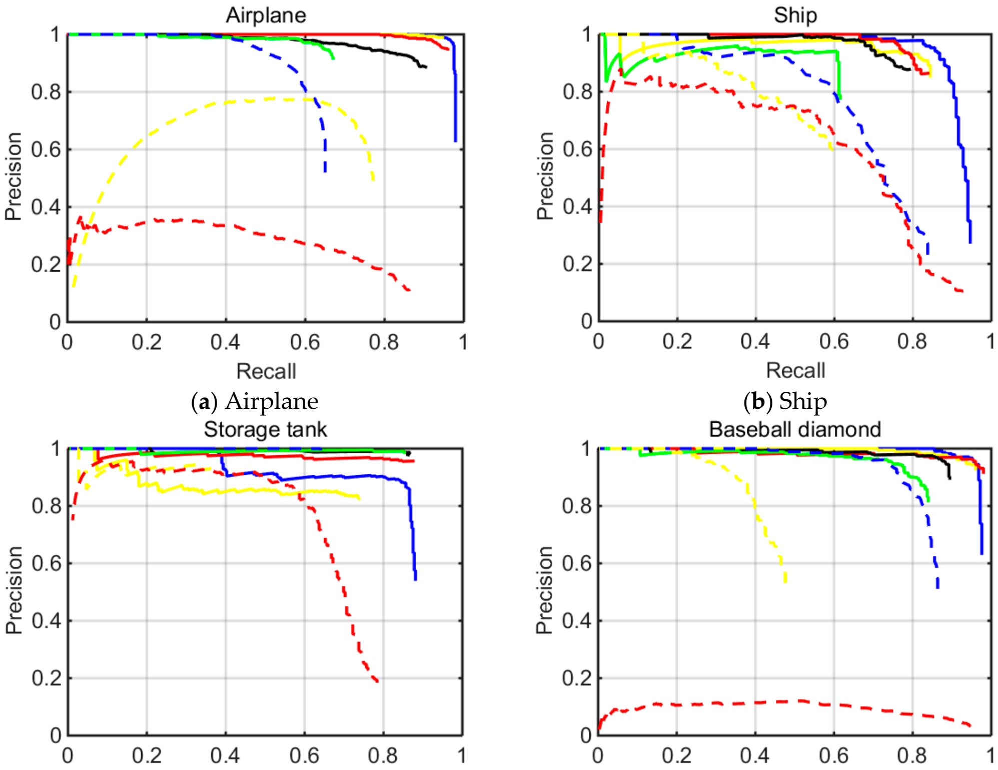
Quantitative comparisons of the eight different methods are shown in Tables 4 and 5, and Figure 9. The PRC over 10 testing classes are plotted in Figure 9. The recall ratio evaluates the ability of detecting more targets, while the precision evaluates the quality of detecting correct objects rather than containing many false alarms. In this figure, we can see that our multi-scale CNN achieves the best recall for almost all classes except bridges. It shows that our multi-scale object proposal network could produce anchor boxes which cover most objects. In particular, the recall rates of small objects like storage tanks and vehicles increase more than other objects, which further illustrate the good performance of our methods for small objects detection. On the other hand, it can be seen that our method can usually achieve higher precision than other methods, in detecting airplanes, ships, storage tanks and so on. This is because we have made use of an object proposal score layer to execute positive samples selection, which means that our anchors boxes have a higher probability of predicting the correct bounding boxes. At the same time, it decreases the number of bounding boxes, so the recall rate on bridges decreased. Table 3 lists the AP and mAP for each method. Based on these statistical data, we can see that the proposed multi-scale CNN obtains the best mAP value of 89.6% among all the object detection methods. In addition, it obtains the highest AP values for almost all classes except storage tanks. Compared with the second best method of Faster R-CNN, there is a 4.97% increase for airplanes, a 11.79% increase for ships, a 42.68% increase for ships, a 1.78% increase for baseball diamonds, a 10.87% increase for tennis courts, a 3.23% increase for basketball courts, a 6.17% increase for ground track fields, a 17.54% increase for harbors, a 25.04% increase for bridges, and a 10.41% increase for vehicles. Table 4 presents the average testing time per image for each method. It is seen that the computation time needed in our method is a little bit more than SSD, but much less than Faster R-CNN.

Table 4. The AP values of the eight object detection methods.

Methods	**BoW**	**SSC BoW**	**COPD**	**Transferred CNN**	**RICNN**	**SSD**	**Faster R-CNN**	**Multi-Scale CNN**
Airplane	0.2496	0.5061	0.6225	0.661	0.8835	0.957	0.946	**0.993**
Ship	0.5849	0.5084	0.6887	0.569	0.7734	0.829	0.823	**0.920**
Storage tank	0.6318	0.3337	0.6371	0.843	0.8527	**0.856**	0.6532	0.832
Baseball diamond	0.0903	0.4349	0.8327	0.816	0.8812	0.966	0.955	**0.972**
Tennis court	0.0472	0.0033	0.3208	0.350	0.4083	0.821	0.819	**0.908**
Basketball court	0.0322	0.1496	0.3625	0.459	0.5845	0.860	0.897	**0.926**
Ground track field	0.0777	0.1007	0.8531	0.800	0.8673	0.582	0.924	**0.981**
Harbor	0.5298	0.5833	0.5527	0.620	0.6860	0.548	0.724	**0.851**
Bridge	0.1216	0.1249	0.1479	0.423	0.6151	0.419	0.575	**0.719**
Vehicle	0.0914	0.3361	0.4403	0.429	0.7110	0.756	0.778	**0.859**
Mean AP	0.2457	0.3081	0.5458	0.597	0.7263	0.759	0.809	**0.896**

Table 5. The average testing time of the eight object detection methods.

Average Testing Time	**BoW**	**SSC BoW**	**COPD**	**Transferred CNN**	**RICNN**	**SSD**	**Faster R-CNN**	**Multi-Scale CNN**
	5.32	40.32	1.07	5.24	8.77	**0.09**	0.16	0.11



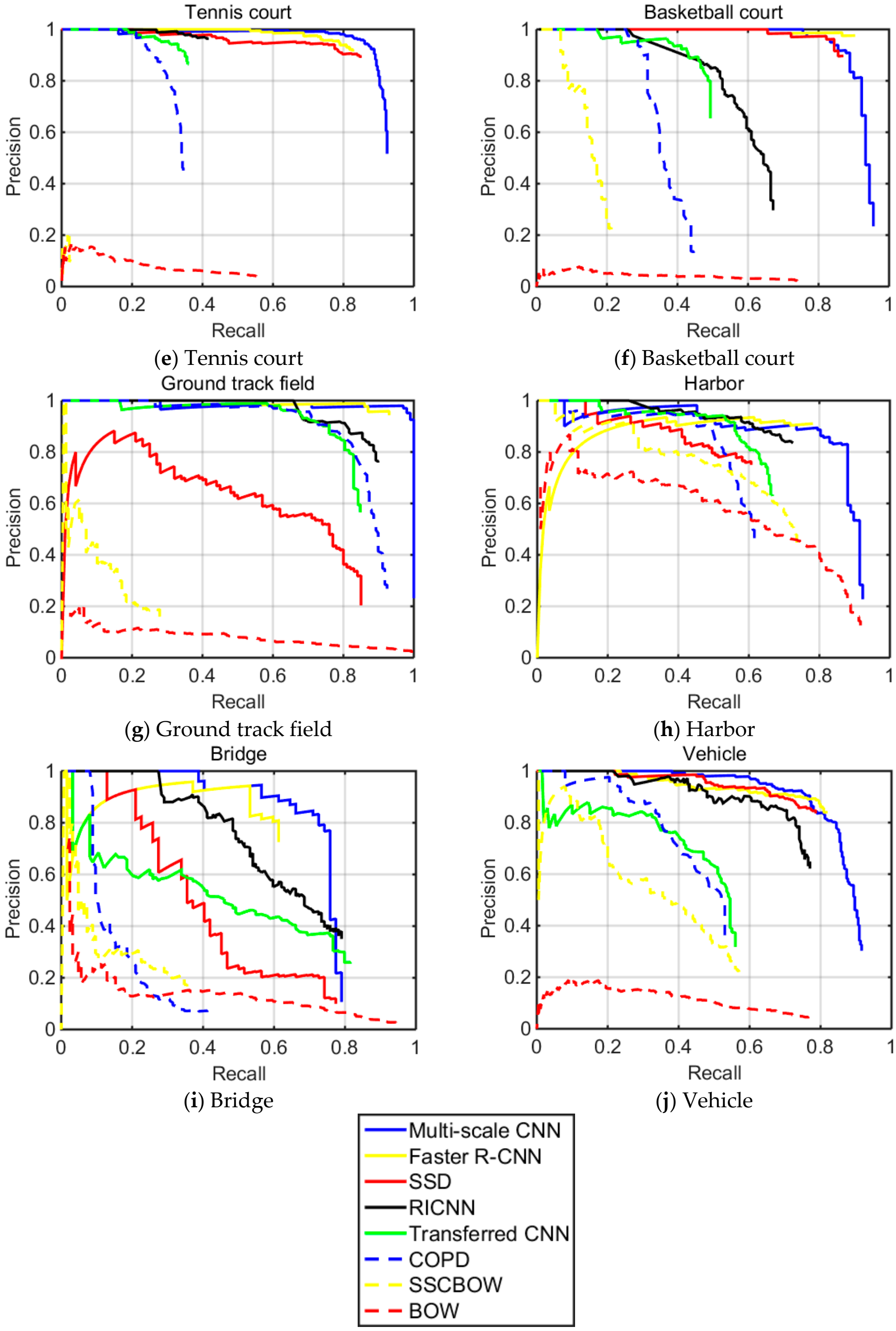
(**a**) Airplane

(**b**) Ship

(**c**) Storage tank

(**d**) Baseball diamond

Figure 9. *Cont.*



(**e**) Tennis court

(**f**) Basketball court

(**g**) Ground track field

(**h**) Harbor

(**i**) Bridge

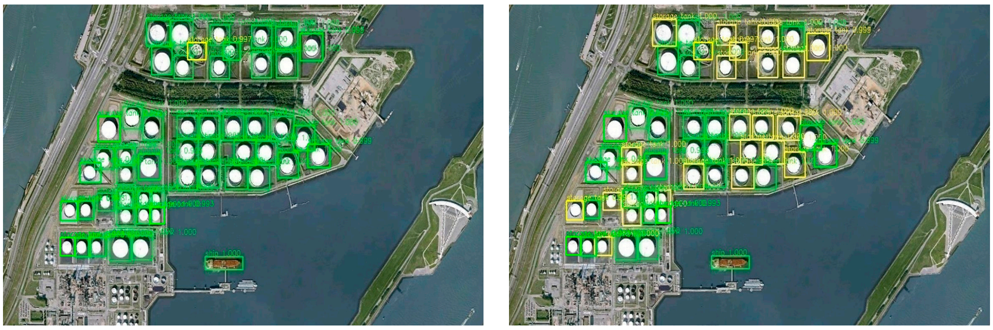
(**j**) Vehicle

Figure 9. Precision versus recall curve for each method.

The bounding box quality is evaluated in Table 6, which lists the AP and mAP for our method, Faster R-CNN and SSD under different IoU for all the test images. It can be easily seen from the table that the AP for each method drops when IoU increases. When IoU is equal to 0.3, we can see that our method obtains mAP value of 92.8%, and it obtains the highest AP values for all classes even for storage tanks. Figure 10 shows the different detection results of storage tanks using our method when IoU is set as 0.3 and 0.5. We can see that our method obtained a lower AP of storage tanks because many bounding boxes with targets inside it are considered as false alarms. If with a small IoU, the AP increased greatly. For fair comparison, we set IoU as 0.5 in this paper, the same as the IoU values in the baseline methods. However, these baseline methods are not for multi-scale geospatial objects detection in HRS imagery. As it is very important to determine a suitable IoU before detection, we can set a lower IoU in real remote sensing applications.

Table 6. The AP values of the three object detection methods under different IoU.

Methods	**SSD**			**Faster R-CNN**			**Multi-Scale CNN**		
IoU	0.3	0.4	0.5	0.3	0.4	0.5	0.3	0.4	0.5
Airplane	0.963	0.961	0.957	0.952	0.950	0.946	**0.993**	0.993	0.993
Ship	0.829	0.829	0.829	0.853	0.853	0.823	**0.922**	0.920	0.920
Storage tank	0.898	0.893	0.856	0.861	0.763	0.6532	**0.951**	0.936	0.832
Baseball diamond	0.966	0.966	0.966	0.964	0.960	0.955	**0.972**	0.972	0.972
Tennis court	0.854	0.847	0.821	0.860	0.857	0.819	**0.924**	0.918	0.908
Basketball court	0.887	0.887	0.860	0.897	0.897	0.897	**0.926**	0.926	0.926
Ground track field	0.743	0.662	0.582	0.924	0.924	0.924	**0.981**	0.981	0.981
Harbor	0.608	0.600	0.548	0.814	0.747	0.724	**0.897**	0.891	0.851
Bridge	0.454	0.454	0.419	0.733	0.716	0.575	**0.827**	0.735	0.719
Vehicle	0.782	0.782	0.756	0.824	0.814	0.778	**0.884**	0.876	0.859
Mean AP	0.798	0.788	0.759	0.868	0.848	0.809	**0.928**	0.915	0.896



(**a**) IoU = 0.3 (**b**) IoU = 0.5

Figure 10. Detection results of storage tanks under different IoU.

5. Discussion

5.1. Performance Analysis of the Proposed Multi-Scale CNN in Large Range HRS Imagery

To demonstrate the effectiveness of the proposed multi-scale CNN in large range HRS imagery application, we collect some large scale HRS images using the Google Earth and have done a great number of experiments on it. For simplicity, we choose one image and show the experimental results below. The image is collected from Charles de Gaulle International Airport in Paris, with a resolution of 0.597 m and a size of 8000× 24,000 pixels. Considering the limited GPU memory and processing speed, each image is cropped into several contiguous image blocks whose size is 1000 × 1000 pixels for testing. During the cropping phase, we set an overlap of 200 pixels for each contiguous image blocks, which is larger than the average airplane length, to ensure that an airplane at the boundary is not be ignored. After each image block is processed individually, the contiguous image blocks are spliced together according to their original position. For overlapping of the adjacent image blocks, we use a NMS operator to eliminate redundant bounding boxes.

Figure 11 shows the detection results of the proposed multi-scale CNN in large range HRS imagery. Table 7 displays the performance of the proposed method accordingly. It can be seen that our method can achieve superior performance in large range HRS imagery. Figure 12 shows some typical false alarms and missing targets for analyzing the characteristics of false alarms and missing targets. Some false alarms are similar to true positive targets, others are brought by splicing of image blocks. As for missing targets, we can see that there is a big color difference between them and true positive targets. Thus, they are missed.

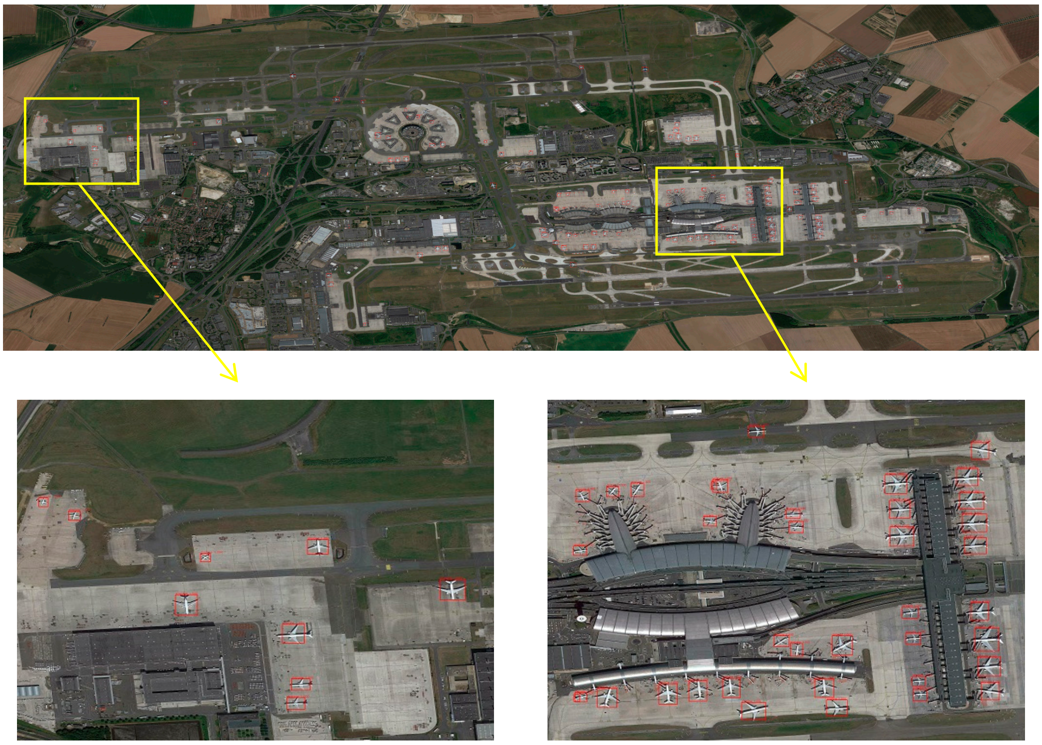


Figure 11. Detection results of our method in large range HRS imagery.

Table 7. Detection performance of our method in large range HRS imagery.

Size	Ground Truth	True Positive	False Positive	Recall	Precision	Testing Time
8000× 24,000	137	132	11	0.964	0.923	81.93

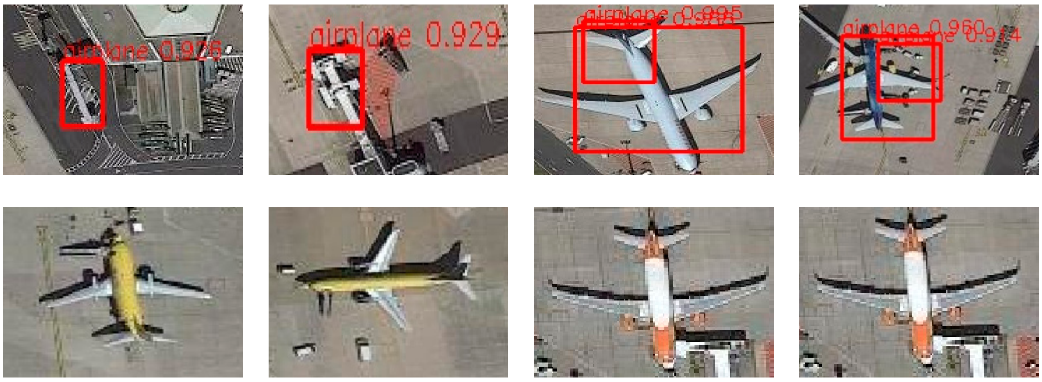


Figure 12. Some typical false alarms and missing targets among the detection results of our method: the first row are some typical false alarms, while the second row are some missing targets.

5.2. *Sufficiency Analysis of the Proposed Multi-Scale CNN*

To address the problem of detecting objects at multiple scales, we propose the multi-scale CNN to learn multiple classifiers by making use of multiple feature maps at different layers. Although great progress has been made in the field of multi-scale object detection in high resolution remote sensing images, we are interested in solving this problem more effectively. We discuss this problem in this section with two extra strategies to our algorithm.

There are two simple strategies to address the problem of detecting objects at multiple scales. The first is to learn a single classifier at the training stage and rescale the image multiple times at the testing stage, so that objects at all possible scales can be matched by the classifier, then a non-maximum suppression operator is applied to eliminate redundant bounding boxes, as shown in Figure 13. This strategy requires feature computation at multiple image scales, which tends to be very time-consuming. An alternative approach is to learn multiple classifiers by using multi-scale training at the training stage, and then use a single-scale image at the testing stage, as shown in Figure 14. It avoids the repeated computation of the feature map, but it is time-consuming to learn multiple classifiers and hard to produce good detectors with a single scale feature map. Here, we discuss the sufficiency of our algorithm by combining our algorithm with the two simple but useful strategies. The multi-scale CNN uses an input size of 320 × 320 at the training stage and an input size of 512 × 512 at the testing stage as detection often requires fine-grained visual information [19]. With multi-scale training, we change the resolution to sizes of {256, 320, 384, 448, 512} at the training stage. We also change the input resolution to sizes of {256, 320, 384, 448, 512} at the testing stage with multi-scale testing. The AP value and testing time are listed in Tables 8 and 9. It can be seen that the Mean AP has no improvement, but it decreases in multi-scale training or multi-scale testing. We can draw the conclusion that our method is sufficient to solve the problem of multi-scales objects detection in HRS images.

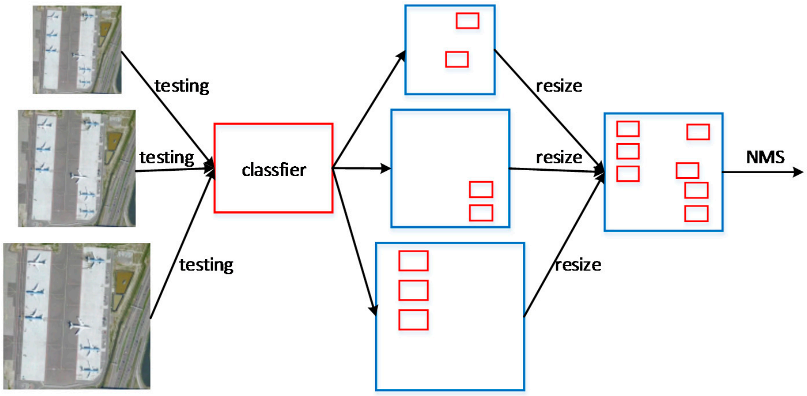


Figure 13. Strategy of the multi-scale testing.

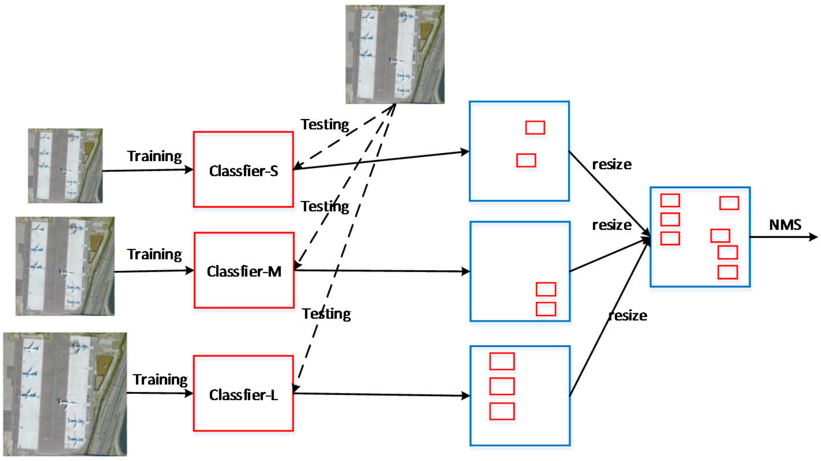


Figure 14. Strategy of the multi-scale training.

Table 8. The AP values of the object detection strategies.

	Airplane	Ship	Storage Tank	Baseball Diamond	Tennis Court	Basketball Court	Ground Track Field	Harbor	Bridge	Vehicle	Mean AP
Our method	**0.993**	0.920	**0.832**	0.972	**0.908**	0.926	0.981	0.851	0.719	**0.859**	**0.896**
Our method with multi-scale training	0.968	0.885	0.756	0.973	0.906	**0.937**	**0.990**	0.876	0.725	0.846	0.886
Our method with multi-scale testing	0.987	**0.928**	0.774	**0.974**	0.903	0.883	0.980	**0.880**	**0.786**	0.846	0.894

Table 9. The average running time of the object detection strategies.

Time (s)	Our Method	Our Method with Multi-Scale Training	Our Method with Multi-Scale Testing
	0.11	**0.11**	0.40

6. Conclusions

A multi-scale CNN for geospatial object detection in HRS images is proposed in this paper. The special design of the multi-scale CNN, i.e., the shared multi-scale base network and the multi-scale object proposal network, enables production of feature maps with high semantic information at different layers and generation of anchor boxes that cover most of the objects with a small amount of negative samples. Experiments on NWPU VHR-10 dataset and comparisons with state-of-the-art approaches demonstrate the effectiveness and superiority of the proposed method. Further, the proposed multi-scale CNN is evaluated and shown to be effective in dealing with large range HRS imagery. With the use of multi-scale training and multi-scale testing, our proposal is shown to be sufficient in detecting multi-scale objects. In our future work, we plan to investigate more refined network to produce anchor boxes with better locating capacity.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (No. 61331016) and the CETC key laboratory of aerospace information applications under Grant KX162600018.

Author Contributions: Wei Guo and Wen Yang provided the original idea for the study; Haijian Zhang and Guang Hua contributed to the discussion of the design; Wei Guo conceived and designed the experiments; Wen Yang supervised the research and contributed to the article's organization; and Wei Guo drafted the manuscript, which was revised by all authors. All authors read and approved the final manuscript.

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

References

1. Yu, H.; Yang, W.; Xia, G.; Liu, G. A Color-Texture-Structure Descriptor for High-Resolution Satellite Image Classification. *Remote Sens.* **2016**, *8*, 259. [CrossRef]
2. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Scalable multi-class geospatial object detection in high spatial resolution remote sensing images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 2479–2482.
3. Cheng, G.; Han, J. A Survey on Object Detection in Optical Remote Sensing Images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
4. Stankov, K.; He, D.C. Detection of buildings in multispectral very high spatial resolution images using the percentage occupancy hit-or-miss transform. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4069–4080. [CrossRef]
5. Sirmacek, B.; Unsalan, C. A probabilistic framework to detect buildings in aerial and satellite images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 211–221. [CrossRef]
6. Zhang, L.; Shi, Z.; Wu, J. A Hierarchical Oil Tank Detector with Deep Surrounding Features for High-Resolution Optical Satellite Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4895–4909. [CrossRef]

7. Ok, A.O.; Baseski, E. Circular oil tank detection from panchromatic satellite images: A new automated approach. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1347–1351. [CrossRef]
8. Wen, X.; Shao, L.; Fang, W.; Xue, Y. Efficient feature selection and classification for vehicle detection. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 508–517.
9. Yu, X.; Shi, Z. Vehicle detection in remote sensing imagery based on salient information and local shape feature. *Opt. Int. J. Light Electron Opt.* **2015**, *126*, 2485–2490. [CrossRef]
10. Cai, H.; Su, Y. Airplane detection in remote sensing image with a circle-frequency filter. In Proceedings of the International Conference on Space Information Technology, Beijing, China, 19–20 November 2005; p. 59852T.
11. Bo, S.; Jing, Y. Region-based airplane detection in remotely sensed imagery. In Proceedings of the 2010 3rd International Congress on Image and Signal Processing (CISP), Yantai, China, 16–18 October 2010; Volume 4, pp. 1923–1926.
12. An, Z.; Shi, Z.; Teng, X.; Yu, X.; Tang, W. An automated airplane detection system for large panchromatic image with high spatial resolution. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 2768–2775. [CrossRef]
13. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]
14. Hu, F.; Xia, G.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
15. Dai, D.; Yang, W. Satellite Image Classification via Two-layer Sparse Coding with Biased Image Representation. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 173–176. [CrossRef]
16. 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, 24–27 June 2014; pp. 580–587.
17. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
18. 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]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. SSD: Single Shot MultiBox Detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 21–37.
21. Gao, Y.; Guo, S.; Huang, K.; Chen, J.; Gong, Q.; Zou, Y.; Bai, T.; Overett, G. Scale Optimization for Full-Image-CNN Vehicle Detection. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 11–14 June 2017; pp. 785–791.
22. Han, X.; Zhong, Y.; Zhang, L. An Efficient and Robust Integrated Geospatial Object Detection Framework for High Spatial Resolution Remote Sensing Imagery. *Remote Sens.* **2017**, *7*, 666. [CrossRef]
23. Lin, H.; Shi, Z.; Zou, Z. Maritime Semantic Labeling of Optical Remote Sensing Images with Multi-Scale Fully Convolutional Network. *Remote Sens.* **2017**, *9*, 480. [CrossRef]
24. Jain, A.K.; Ratha, N.K.; Lakshmanan, S. Object detection using Gabor filters. *Pattern Recognit.* **1997**, *30*, 295–309. [CrossRef]
25. Leninisha, S.; Vani, K. Water flow based geometric active deformable model for road network. *ISPRS J. Photogramm. Remote Sens.* **2015**, *102*, 140–147. [CrossRef]
26. Ok, A.O. Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS J. Photogramm. Remote Sens.* **2013**, *86*, 21–40. [CrossRef]
27. Ok, A.O.; Senaras, C.; Yuksel, B. Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 1701–1717. [CrossRef]
28. Blaschke, T.; Hay, G.J.; Kelly, M.; Lang, S.; Hofmann, P.; Addink, E.; Feitosa, R.Q.; Vander Meer, F.; Vander Werff, H.; Van Coillie, F. Geographic object-based image analysis-towards a new paradigm. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 180–191. [CrossRef] [PubMed]
29. Li, Y.; Wang, S.; Tian, Q.; Ding, X. Feature representation for statistical-learning-based object detection: A review. *Pattern Recognit.* **2015**, *48*, 3542–3559. [CrossRef]

30. Li, X.; Cheng, X.; Chen, W.; Chen, G.; Liu, S. Identification of Forested Landslides Using LiDAR Data, Object-based Image Analysis, and Machine Learning Algorithms. *Remote Sens.* **2015**, *7*, 9705–9726. [CrossRef]
31. Xia, G.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L. AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
32. 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, 20–26 June 2005; pp. 886–893.
33. Liao, S.; Zhu, X.; Lei, Z.; Zhang, L.; Li, S. Learning multi-scale block local binary patterns for face recognition. In Proceedings of the International Conference on Biometrics (ICB), Seoul, Korea, 27–29 August 2007; pp. 828–837.
34. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [CrossRef] [PubMed]
35. Cheng, G.; Guo, L.; Zhao, T.; Han, J.; Li, H.; Fang, J. Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA. *Int. J. Remote Sens.* **2013**, *34*, 45–59. [CrossRef]
36. Sun, H.; Sun, X.; Wang, H.; Li, Y.; Li, X. Automatic Target Detection in High-Resolution Remote Sensing Images Using Spatial Sparse Coding Bag-of-Words Model. *IEEE Geosci. Remote Sens. Lett.* **2011**, *9*, 109–113. [CrossRef]
37. Zhao, Y.; Yang, J. Hyperspectral image de-noising via sparse representation and low-rank constraint. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 296–308. [CrossRef]
38. Yang, W.; Yin, X.; Xia, G. Learning High-level Features for satellite Image Classification with Limited Labeled Samples. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4472–4482. [CrossRef]
39. Du, B.; Zhang, L. A discriminative metric learning based anomaly detection method. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6844–6857.
40. Ren, X.; Ramanan, D. Histograms of Sparse Codes for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 25–27 June 2013; pp. 3246–3253.
41. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
42. Geva, S.; Sitte, J. Adaptive nearest neighbor pattern classification. *IEEE Trans. Neural Netw.* **2002**, *2*, 318–322. [CrossRef] [PubMed]
43. Tim, K. Random decision forests. In Proceedings of the International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; Volume 1, p. 278.
44. Kirzhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; Volume 25, pp. 1097–1105.
45. 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.* **2016**, *13*, 1074–1078. [CrossRef]
46. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S. Single-Shot Refinement Neural Network for Object Detection. *arXiv* **2014**.
47. Cai, Z.; Fan, Q.; Feris, R.; Vasconcelos, N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In Proceedings of the IEEE European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 354–370.
48. Lin, T.; 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 June 2017; pp. 2117–2125.
49. Fu, C.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**.
50. Kong, T.; Sun, F.; Yao, A.; Liu, H.; Lu, M.; Chen, Y. Ron: Reverse connection with objectness prior networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 5936–5944.
51. Shrivastava, A.; Sukthankar, R.; Malik, J.; Gupta, A. Beyond Skip Connections: Top-Down Modulation for Object Detection. *arXiv* **2016**.
52. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**.

53. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
54. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-based Object Detectors with Online Hard Example Mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
55. Hu, Q.; Wu, W.; Xia, T.; Yu, Q.; Yang, P.; Li, Z.; Song, Q. Exploring the use of Google Earth imagery and object-based methods in land use/cover mapping. *Remote Sens.* **2013**, *5*, 6026–6042. [CrossRef]
56. NWPU VHR-10 Dataset. Available online: http://www.escience.cn/people/gongcheng/NWPU-VHR-10.html (accessed on 26 June 2017).
57. Xu, S.; Fang, T.; Li, D.; Wang, S. Object classification of aerial images with bag-of-visual words. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 366–370.
58. Cheng, G.; Han, J.; Zhou, P. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote Sens.* **2014**, *98*, 119–132. [CrossRef]
59. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]