*remote sensing*

*Article*

# Geospatial Object Detection on High Resolution Remote Sensing Imagery Based on Double Multi-Scale Feature Pyramid Network

**Xiaodong Zhang *, Kun Zhu , Guanzhou Chen , Xiaoliang Tan, Lifei Zhang, Fan Dai, Puyun Liao and Yuanfu Gong**

State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; zkun@whu.edu.cn (K.Z.); cgz@whu.edu.cn (G.C.); xl_tan@whu.edu.cn (X.T.); lifeizhang@whu.edu.cn (L.Z.); daifan@whu.edu.cn (F.D.); LiaoPuyun@whu.edu.cn (P.L.); gongyuanfu@163.com (Y.G.)

* Correspondence: zxdlmars@whu.edu.cn; Tel.: +86-27-6877-8033

Received: 17 February 2019; Accepted: 23 March 2019; Published: 28 March 2019

**Abstract:** Object detection on very-high-resolution (VHR) remote sensing imagery has attracted a lot of attention in the field of image automatic interpretation. Region-based convolutional neural networks (CNNs) have been vastly promoted in this domain, which first generate candidate regions and then accurately classify and locate the objects existing in these regions. However, the overlarge images, the complex image backgrounds and the uneven size and quantity distribution of training samples make the detection tasks more challenging, especially for small and dense objects. To solve these problems, an effective region-based VHR remote sensing imagery object detection framework named Double Multi-scale Feature Pyramid Network (DM-FPN) was proposed in this paper, which utilizes inherent multi-scale pyramidal features and combines the strong-semantic, low-resolution features and the weak-semantic, high-resolution features simultaneously. DM-FPN consists of a multi-scale region proposal network and a multi-scale object detection network, these two modules share convolutional layers and can be trained end-to-end. We proposed several multi-scale training strategies to increase the diversity of training data and overcome the size restrictions of the input images. We also proposed multi-scale inference and adaptive categorical non-maximum suppression (ACNMS) strategies to promote detection performance, especially for small and dense objects. Extensive experiments and comprehensive evaluations on large-scale DOTA dataset demonstrate the effectiveness of the proposed framework, which achieves mean average precision (mAP) value of 0.7927 on validation dataset and the best mAP value of 0.793 on testing dataset.

**Keywords:** very-high-resolution (VHR) remote sensing imagery; object detection; multi-scale pyramidal features; multi-scale strategies

## 1. Introduction

Object detection on very-high-resolution (VHR) optical remote sensing imagery has attracted more and more attention. It not only needs to identify the category of the object, but also needs to give the precise location of the object [1]. The improvements of earth observation technology and diversity of remote sensing platforms have seen a sharp increase in the amount of remote sensing images, which promotes the research of object detection. However, the problems of the complex backgrounds, the overlarge images, the uneven size and quantity distribution of training samples, illumination and shadows make the detection tasks more challenging and meaningful [2–4].

The optical remote sensing image object detection has made great progress in recent years [5]. The existing detection methods can be divided into four main categories, namely, template matching-based methods, knowledge-based methods, object image analysis-based (OBIA-based) methods and machine learning-based methods [2]. The template matching-based methods [6–8] mainly contain rigid template matching and deformable template matching, which includes two steps, specifically, template generation and similarity measure. Geometric information and context information are the two most common knowledge for knowledge-based object detection algorithm [9–11]. The key of the algorithm is effectively transforming the implicit connotative information into established rules. OBIA-based image analysis [12] principally contains image segmentation and object classification. Notably, the appropriate segmentation parameters are the key factors, which will affect the effectiveness of the object detection. In order to more comprehensively and effectively characterize the object, machine learning-based methods [13,14] are applied. They first extract the features (e.g., histogram of oriented gradients (HOG) [15], bag of words (BoW) [16], Sparse representation (SR)-based features [17], etc.) of the object, then perform feature fusion and dimension reduction to concisely extract features. Finally, those features are fed into a classifier (e.g., Support vector machine (SVM) [18], AdaBoost [19], Conditional random field (CRF) [20], etc.) trained with a large amount of data for object detection. In conclusion, those methods rely on the hand-engineered features, however, they are difficult to efficiently process remote sensing images in the context of big data. In addition, the hand-engineered features can only detect specific targets, when applying them to other objects, the detection results are unsatisfactory [1].

In recent years, the deep learning algorithms emerging in the field of artificial intelligence (AI) are a new kind of computing model, which can extract advanced features from massive data and perform efficient information classification, interpretation and understanding. It has been successfully applied to the fields of machine translation, speech recognition, reinforcement learning, image classification, object detection and other fields [21–25]. Even in some applications, it has exceeded the human level [26]. Compared with the traditional object detection and localization methods, the deep learning-based methods have stronger generalization and features expression ability [2]. It learns effective representation of features by a large amount of data, and establishes relatively complex network structure, which fully exploits the association among data and builds powerful detectors and locators. Convolutional neural network (CNN) is a kind of deep learning model specially designed for two-dimensional structure images inspired by biological visual cognition (local receptive field) and it can learn the deep features of images layer by layer. The local receptive field of CNN can effectively capture the spatial relationship of the objects. The characteristics of weight sharing greatly reduces the training parameters of the network and the computational cost. Therefore, the CNN-based methods are being widely used when automatically interpreting images [2,27–30].

In the field of object detection, with the development of the large public natural image datasets (e.g., Pascal VOC [31], ImageNet [32]), and the significantly improved graphics processing units (GPUs), the CNN-based detection frameworks have achieved outstanding achievements [33]. The existing CNN-based detection methods can be roughly divided into two groups: the region-based methods and the region-free methods. The region-based methods first generate candidate regions and then accurately classify and locate the objects existing in these regions, and these methods have higher detection accuracy but slower speed. Conversely, the region-free methods directly regress the object coordinates and object categories in multiple positions of the image, and the whole detection process is one-stage. These region-free methods have faster detection speed but relatively poor accuracy [34]. Among numerous region-based methods, Region-based CNN (R-CNN) [35] is a pioneering work. It utilizes the selective search algorithm [36] to generate the region proposals, and then extracts features via CNN on these regions. The extracted features are fed into a trained SVM classifier, which classifies the category of the object. Finally, bounding box regression is used to correct the initial extracted coordinates and non-maximum uppression (NMS) is used to delete highly redundant bounding boxes to obtain accurate detection results. R-CNN [35] demands to perform feature extraction at each region proposal, so the process is time-consuming [37]. Besides, the forced image resizing process on the candidate regions before they are fed into the CNN also caused information loss. To solve the above problems, He et al. proposed Spatial Pyramid Pooling Network (SPP-Net) [38], which adds a spatial pyramid layer, namely, Region-of-Interest (RoI) pooling layer, on the top of the last convolutional layer. The RoI pooling layer divides the features and generates fixed-length outputs, therefore it can deal with the arbitrary-size input images. SPP-Net [38] performs one-time features extraction to obtain an entire-image feature map, and the region proposals share the entire-image feature map, which greatly speeds up the detection. On the basis of R-CNN, Fast-RCNN [39] adopts the multi-task loss function to carry out classification and regression simultaneously, which improves the detection, positioning accuracy and greatly improves the detection efficiency. However, using the selective search algorithm to generate region proposals is still very time-consuming because the algorithm implements on the central processing unit (CPU). In order to take advantage of the GPUs, Faster R-CNN [37], consisting of a region proposal network (RPN) and Fast R-CNN, was proposed. The two networks share convolution parameters, and they have been integrated into a unified network. Thus, the region-based object detection network achieves end-to-end operation. Feature pyramids play a crucial role in multi-scale object detection system, which combine resolution and semantic information over multiple scales. Feature pyramid network (FPN) [40] was proposed to simultaneously utilize low-resolution, semantically strong features and high-resolution, semantically weak features, it is superior to single-scale features for a region-based object detector and shows significant improvements in detecting small objects. In addition to the region-based object detection frameworks, there are many region-free object detection networks, including Over-Feat [41], you only look once (YOLO) [42] and single shot multi-box detector (SSD) [43], etc. These one-stage networks consider object detection as a regression problem, they do not generate region proposals and predict the class confidence and coordinates directly. They greatly improve the detection speed, although sacrificing some precision.

The CNN-based natural imagery object detection has made great progress, but high-precision and high-efficiency object detection for remote sensing images still has a long way to go. Different from natural images, remote sensing images usually show the following characteristics:

1. The perspective of view. Remote sensing images are usually obtained from a top-down view while natural images can be obtained from different perspectives, which greatly affects how objects are rendered on the images [1].
2. Overlarge image size. Remote sensing images are usually larger in size and range than natural images. Compared with natural image processing, remote sensing image processing is more time-consuming and memory-consuming.
3. Class imbalances. The imbalances mainly include category quantity and object size. Objects in natural scene images are generally uniformly distributed and not particularly numerous, but a single remote sensing image may contain one object or hundreds of objects and it may also simultaneously include large objects such as playgrounds and small objects like cars.
4. Additional influence factors. Compared with natural scene image, remote sensing image object detections are affected by illumination condition, image resolution, occlusion, shadow, background and border sharpness [33].

Therefore, constructing a robust and accurate object detection framework for remote sensing images is very challenging, but it is also of much significance. To overcome the size restrictions of the input images, the problem of small objects loss and retain the resolution of the objects, Chen et al. [1] put forward MultiBlock layer and MapBlock layer based on SSD [43]. The MultiBlock layer divides the input image into multiple blocks, the MapBlock layer maps the prediction results of each block to the original image. The network achieves a good effect on airplane detection. Considering the complex distribution of geospatial objects and the low efficiency for remote sensing imagery, Han et al. [33] proposed the P-R-Faster R-CNN, which achieves multi-class geospatial object detection by combining the robust properties of transfer mechanism and the sharable properties of Faster R-CNN. Guo et al. [3] proposed a unified multi-scale CNN for multi-scale geospatial object detection, which consists of a multi-scale object proposal network and a multi-scale object detection network. The network achieves the best precision on the Northwestern Polytechnical University very high spatial resolution-10 (NWPU VHR-10) [44] dataset. However, for small and dense objects detection on remote sensing images, they did not propose an effective solution, and did not make full use of the resolution and semantic information simultaneously, which may lead to unsatisfactory results in the case of more complex backgrounds, numerous data and overlarge image size [4,40]. Some frameworks [1,45–47] only have effects for certain types of objects. Besides, RoI pooling layer in these networks will cause misalignments between the inputs and their corresponding final feature maps, these misalignments affect the object detection accuracy, especially for small objects.

To solve the above problems, we presented an effective framework, namely, Double Multi-scale Feature Pyramid Network (DM-FPN), which makes full use of semantic and resolution features simultaneously. We also put forward some multi-scale training, inference and adaptive categorical non-maximum suppression (ACNMS) strategies. The main contributions of this paper are summarized as follows:

1. We have constructed an effective multi-scale geospatial object detection framework, which achieves good performance by simultaneously utilizing low-resolution, semantically strong features and high-resolution, semantically weak features. Accordingly, the RoI Align layer used in our framework can solve the misalignment caused by RoI pooling layer and it improves the object detection accuracy, especially for small objects.
2. We proposed several multi-scale training strategies, including the patch-based multi-scale training data and the multi-scale image sizes used during training. To overcome the size restrictions of the input images, we divided the image into blocks with a certain degree of overlap. The patch-based multi-scale training data strategy both enhance the resolution features of the small objects and integrally divide the large objects into a single patch for training. In order to increase the diversity of objects, we adopt multiple image sizes strategy for patches during training.
3. During the inference stage, we also proposed a multi-scale strategy to detect as many objects as possible. Besides, depending on the intensity of the object, we adopt the noval ACNMS strategy, which can effectively reduce redundancy among the highly overlapped objects and slightly overcome the uneven quantity distribution of training samples, enabling the framework preferably to detect both small and dense objects.

Experiment results evaluated on DOTA [48] dataset, a large-scale dataset for object detection in aerial images, indicating the effectiveness and superiority of the proposed framework. The rest of this paper is organized as follows. Section 2 introduces the related work involved in the paper. Section 3 elaborates the proposed framework in detail. Section 4 mainly includes the description of the datasets, evaluation criteria and experiment details. Section 5 implements ablation experiments and makes reliable analyses to the results. Section 6 discusses the proposed framework and analyzes its limitations. Finally, the conclusions are drawn in Section 7.

## 2. Related Works

In this section, we will first review some outstanding region-based object detection frameworks, they have achieved remarkable accomplishments on natural image object detection. Then we will introduce RoI Align layer, which can significantly improve the detection performance of small objects.

### 2.1. Region-Based Object Detection Networks

The region-based object detection networks are mainstream frameworks for high-precision object detection, including R-CNN, SPP-Net, Fast R-CNN and Faster R-CNN [35,37–39]. Their common process is to first generate numerous candidate areas by the region proposal algorithms [36,49,50]. Then, the networks employ CNN to extract abundant features from these candidate regions and infer the category and coordinates of objects on each region. Finally, a bounding box algorithm is utilized to get precise coordinates. Faster R-CNN integrates these steps to form a unified network and realizes end-to-end object detection. It consists of two modules, formally, RPN and Fast R-CNN, and the two tasks share convolutional features. Figure 1 shows the overall architecture of Faster R-CNN.

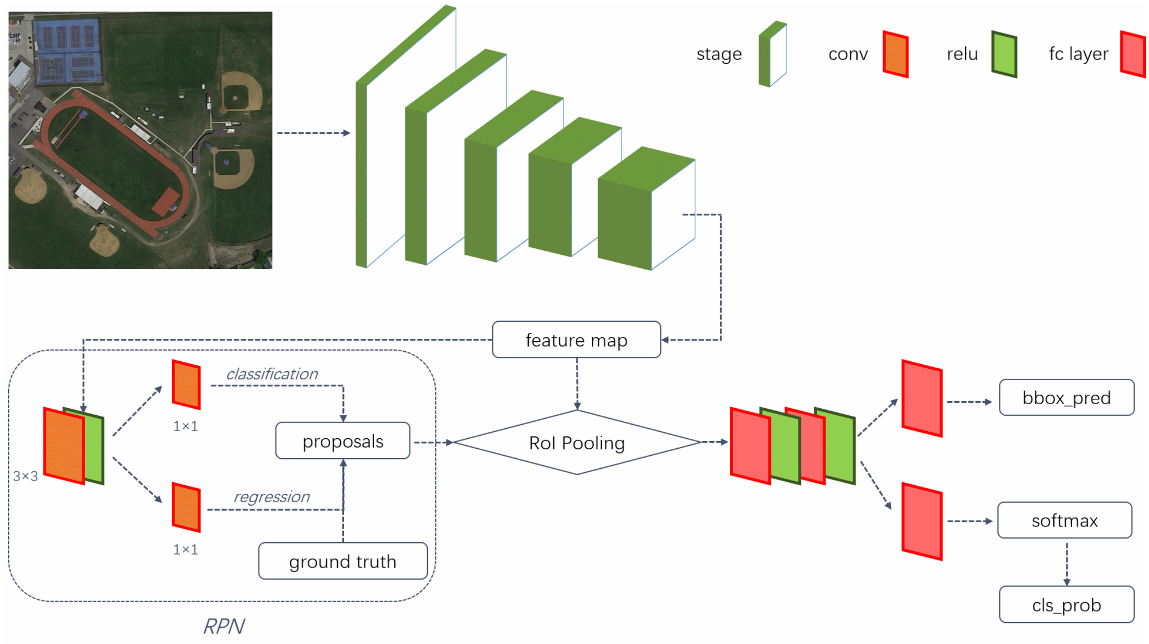



**Figure 1.** The architecture of Faster R-CNN. The "conv" represents convolutional layer, the "relu" represents activation function and the "fc layer" represents fully connected layer. The network outputs intermediate layers of the same size in the same "stage". The "bbox_pred" represents the position offset of the object and the "cls_prob" represents the probability of the category.

RPN is a kind of fully convolutional network [51], it deals with the arbitrary-size input image and outputs a set of region proposals with an objectness score. These candidate regions will be fed into the following Fast R-CNN for precise detection. The core scheme of RPN is "anchors", which simultaneously predicts multiple region proposals of diversiform scales and aspect ratios with a total number of $k$ at each sliding window in the last shared convolutional layer. The features obtained from each sliding window will be imported into two sibling $1 \times 1$ convolutional layers, specifically, the box-classification layer (*cls*) and the box-regression layer (*reg*). The *cls* layer is used to identify a binary class label of being an object or not while the *reg* layer is used to correct the coordinates of the object. Therefore, the *cls* layer has 2k outputs while the *reg* layer has 4k outputs.

After RPN processing, we got a mass of candidate regions with class-agnostic and coordinate attributes. These regions will be fed into the subsequent Fast R-CNN for further category judgment and coordinate regression. Fast R-CNN adopts RoI pooling layer to extract fixed-length feature vectors from arbitrary-size candidate regions and these feature vectors are fed into categorical classification and regression layers to obtain the final detection results. The RPN and Fast R-CNN employ the approximate joint training scheme to share convolution. As such, an efficient and end-to-end object detection framework is constructed.

### 2.2. Feature Pyramid Network

Most region-based object detection frameworks only use the single-scale features for faster detection, such feature representations are very unfriendly to small objects. In Faster R-CNN, the backbone adopts Visual Geometry Group 16 weight layers (VGG16 [52]) and the last feature map reduces to 1/32 compared to the original image after 5 convolutional layers (with a pooling step of 2), some small objects like cars and ships will lose a large proportion of features after such operations. In the deep convolutional networks, the low-level layers have poor semantic but strong resolution while the high-level layers have rich semantic but scarce resolution [40]. Although some frameworks [43,53] adopt multi-scale feature maps that already computed from different layers, they abnegate low-level features and therefore lose the opportunity to take advantage of higher-resolution features. Combining strong resolution and semantic information will enhance the detection performance, especially for small objects. In a pioneering way, FPN leverages the in-network features obtained from the last layer of each stage in the convolutional networks (ConvNets). It combines coarse-resolution, semantically strong features with high-resolution, semantically weak features to construct a multi-scale pyramidal hierarchy network without additional memory consumption. We note that if the output feature maps have the same size, they are in the same stage. As shown in the Figure 2, the core mechanism of the FPN mainly includes bottom-up pathway, top-down pathway and lateral connections.

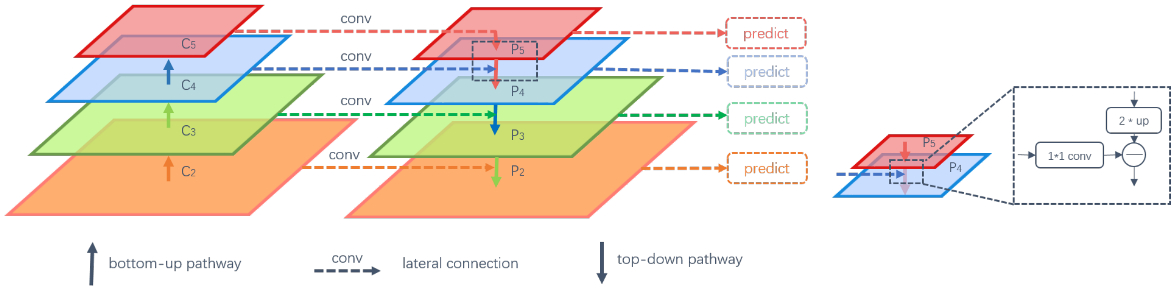
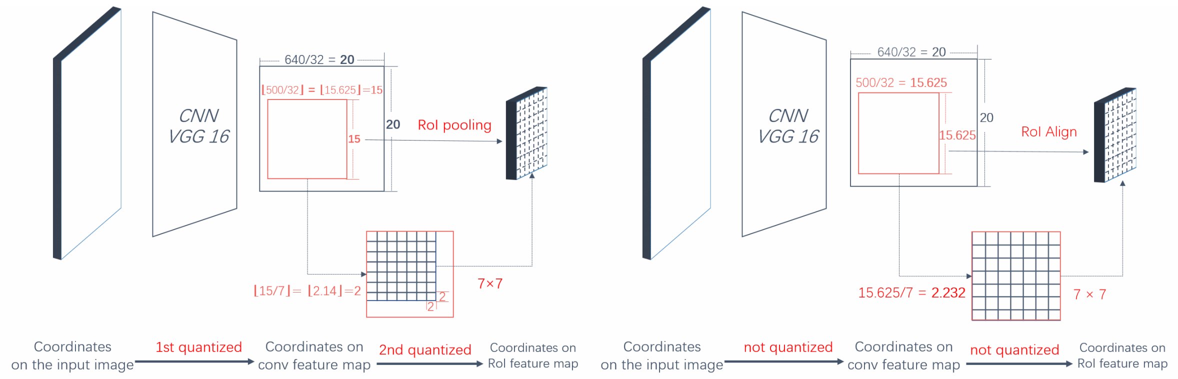



**Figure 2.** The core mechanism of the FPN mainly includes bottom-up pathway, top-down pathway and lateral connections.

- Bottom-up pathway. Actually, this operation is the forward propagation process of the network. During the operation, the last convolutional layer in each stage is extracted to establish a feature pyramid. Compared with other methods [54–56], this mechanism requires no additional memory footprint.
- Top-down pathway and lateral connections. The top-down pathway upsamples the feature map obtained from the bottom-up pathway to the same size as the semantically coarser, but spatially stronger feature maps. The lateral connections merge the same-size feature maps obtained from the bottom-up pathway and the top-down pathway respectively, which first undergoes a $1 \times 1$ convolutional layer to reduce channel dimensions. The mergence process is implemented by element-wise addition. Subsequently, a $3 \times 3$ convolution is executed on each merged feature map to eliminate the aliasing effect of upsampling.

### 2.3. ROI Align

ROI Align is a kind of regional feature aggregation method proposed in Mask R-CNN [57], which solves the problem of misalignment caused by RoI pooling during the two integer quantification operations. RoI pooling layer divides the region proposal on the last convolutional layer into a fixed-length (e.g., $7 \times 7$) feature map for subsequent classification and bounding box regression tasks. Since the coordinates of candidate regions are obtained by regression, generally speaking, they are floating-numbers. After rounding down, the data after the decimal point is abandoned. As shown in Figure 3a, there are two rounding operations during the pooling: the coordinates of candidate region are first quantified to integer, then the quantified RoI is divided into $k \times k$ bins on average, and each bin is quantified again thus introducing misalignments between the RoI and the final feature map. Such misalignments are harmful to objects detection task, especially for small objects.




(**a**) RoI pooling layer. (**b**) RoI align layer.

**Figure 3.** RoI align layer solves misalignments caused by RoI pooling layer.

RoI Align was proposed to solve the above deficiency of RoI Pooling, it abnegates all quantifications and utilizes bilinear interpolation to obtain the precise values. Formally, RoI Align retains the original floating-numbers instead of quantified integers. The alignment process is shown in Figure 3b. During the first quantification, the boundary coordinates of each candidate region are not round down to maintain floating-numbers. During the second quantification, each RoI is divided into $k \times k$ bins and this process is still not round down. Subsequently, four fixed sampled points are calculated by bilinear interpolation in each RoI bin, and the maximum or average pooling is performed to get align results. RoI Align solves the misalignments between the inputs and the extracted feature maps, which is significant for object detection on remote sensing images that contain numerous small objects.

## 3. Framework

In this section, we will elaborate the details of our proposed framework. In order to efficiently detect the objects on remote sensing images, we also propose some multi-scale training and inference strategies. Meanwhile, different ACNMS thresholds are selected according to the size and intensity of the category, which can improve the detector performance to some extent.

### *3.1. The Core Mechanism of the Proposed Network*

#### 3.1.1. The Overall Structure

The overall structure of the proposed framework named Double Multi-scale Feature Pyramid Network (DM-FPN) is shown in Figure 4.

The infrastructure of DM-FPN is based on Faster R-CNN [37] with FPN [40]. Formally, both the original region proposal network and the detection network were modified by FPN. DM-FPN combines coarse-resolution, semantically strong features with high-resolution, semantically weak features, and such operations have great advantages in detecting small objects. We adopt ResNet50 [58] as backbone of our framework. The convolution can be divided into 5 stages and the output of each stage's last residual block was selected as $\{C_2, C_3, C_4, C_5\}$, noting that they have strides of {4, 8, 16, 32} pixels corresponding to the original image. We do not utilize the first stage because it is memory-consuming. This process is called the bottom-up pathway, which has been described in Section 2.2. The corresponding $\{P_2, P_3, P_4, P_5\}$ were obtained by top-down path, lateral connections and mergence. Actually, to eliminate the aliasing effect of upsampling, a $3 \times 3$ convolution is executed on each merged feature map to obtain the final feature maps $\{P_2, P_3, P_4, P_5\}$, which are shared by the region proposal network and the class-specific detection network.

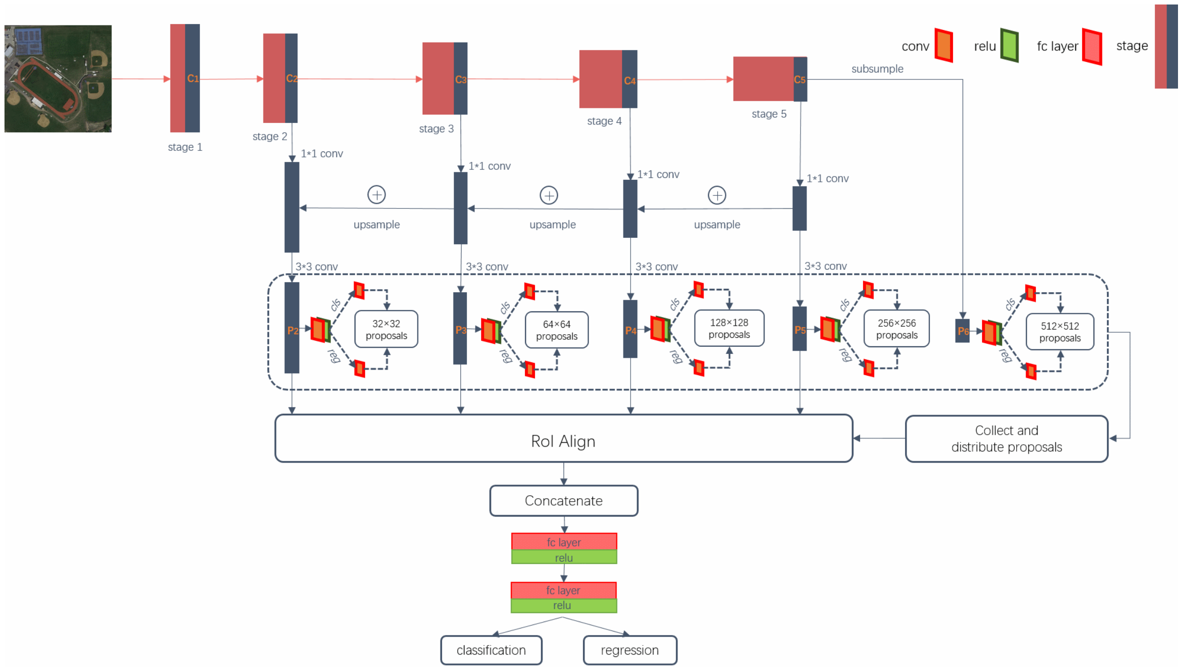



**Figure 4.** The overall structure of the proposed DM-FPN. It consists of a multi-scale region proposal network and a multi-scale object detection network. These two modules share convolutional layers.

### 3.1.2. Multi-Scale Region Proposal Network

The original RPN extracts region proposals on the last single-scale convolutional layer. In order to take advantage of the pyramid character of FPN, we need to extract candidate regions on multiple convolutional layers, namely, {$P_2$, $P_3$, $P_4$, $P_5$, $P_6$}, noting that $P_6$ is simply a stride 2 subsampling of $P_5$, which is only used in multi-scale region proposal network. The anchors own ranges of {$32^2$, $64^2$, $128^2$, $256^2$, $512^2$} pixels on {$P_2$, $P_3$, $P_4$, $P_5$, $P_6$} respectively. On each feature map, there are three aspect ratios, namely, {1:2, 1:1, 2:1}. As a result, there are a total of 15 anchors on these pyramidal feature maps. The selection of positive and negative samples is determined by the Intersection-over-Union (IoU) between the region proposal and ground-truth box. We note that IoU is defined as the ratio between the intersection and the union of two boxes. If an anchor has the highest IoU with a given ground-truth box or it has an IoU greater than 0.7 with any ground-truth box, then it will be assigned to the positive. Conversely, if an anchor has an IoU less than 0.3 for all ground-truth boxes, it's a negative sample. We abandon samples that are neither positive nor negative. In a mini-batch of 256, the ratio of positive to negative samples is 1:1. These rules apply to {$P_2$, $P_3$, $P_4$, $P_5$, $P_6$} indistinguishably. Specially, the common ground-truth boxes are equally participated in the calculation with the pyramid anchors located on five-level feature maps. With these definitions, the loss function for an image is defined as:

$$L\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i \left(p_i, p_i^*\right) + \lambda \cdot \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \quad (1)$$

where $i$ represents the index of an anchor in a mini-batch while $p_i$ is the predicted probability of anchor $i$ being an object. If the anchor is positive, the ground-truth label $p_i^*$ equals to 1, otherwise equals to 0. $t_i$ is a vector that consists of four parameterized coordinates of the predicted bounding box, and $t_i^*$ is that of the ground-truth box associated with a positive anchor. The classification loss $L_{cls}$ is represented by the log loss, which identifies a binary class label of being an object or not. And the regression loss $L_{reg}$ is constructed by the Smooth L1 loss. The above two loss functions are weighted by a balancing parameter $\lambda$. Usually, the *cls* term is normalized by the mini-batch size while the *reg* term is normalized by the number of anchors. In this paper, we specify that $N_{cls}$ and $N_{reg}$ are equal to 256 and 2000, respectively. We set $\lambda$ is equals to 9 and thus both *cls* and *reg* terms are roughly equally weighted.

We note that we reserve the top 2000 region proposals based on their *cls* scores on {$P_2$, $P_3$, $P_4$, $P_5$, $P_6$} respectively, then we concatenate these candidate boxes and adopt Non-Maximum Suppression (NMS) with a fixed IoU threshold of 0.7 to retain the final 2000 RoIs, which will be fed into the subsequen class-specific detection network for exact object detection.

#### 3.1.3. Multi-Scale Class-Specific Detection Network

Fast R-CNN [39] is a single-scale region-based object detection framework, which utilizes RoIs generated by RPN for object detection. Different from the previous networks that pooling RoI to single-scale feature map, we need to align RoIs from different scales to the multiple pyramidal feature maps. We assign an RoI of width $w$ and height $h$ (based on the input image) to the level $P_k$ by:

$$k = \left\lfloor \boldsymbol{k}_0 + \log_2(\sqrt{wh}/224) \right\rfloor \tag{2}$$

where 224 is the normative ImageNet pre-training size as FPN [40] does, and $k_0$ is the level that an RoI with a size of $w \times h = 224^2$ should be mapped into. Notably, we assigned $k_0$ equals to 4 as [40] does. These RoIs can be assigned to different levels according to their size. For example, if an anchor has a width of 188 and a height of 111, it should be mapped into the $P_3$ level. Subsequently, we adopt RoI align to extract $7 \times 7$ feature maps, which will be fed into two 1024-d fully-connected layers before the final classification and bounding box regression layers. Based on the above settings, both region proposal network and class-specific detection network can utilize multi-scale pyramidal features for object detection.

### *3.2. Multi-Scale Training Strategies*

Multi-scale training strategies mainly include the patch-based multi-scale training data and the multi-scale image sizes used during training. Their descriptions are as follows:

1. Patch-based multi-scale training data. The size restrictions of the input images cause a lot of semantic information will lost in the deep convolutional layers, especially for small objects. Therefore, we slice remote sensing images into patches with a certain degree of overlap, and then send these image blocks into the network for training. At the same time, considering the uneven distribution of objects on the remote sensing image, which may include large objects such as playgrounds, and may also include small objects like cars, we enlarge and shrink remote sensing images by a factor of 2 and 0.5 respectively. The enlarged remote sensing images enhance the resolution features of the small objects while the shrunken remote sensing images integrally divide the large objects into a single patch for training.
2. Multi-scale image sizes used during training. In order to enhance the diversity of objects, we adopt multiple scales for patches during training. Each scale is the pixel size of a patch's shortest side and the network uniformly select a scale for each training sample at random.

### *3.3. Multi-Scale Inference Strategies*

We scale images to detect as many objects as possible during inference, and the scaled images include enlarged and shrunken images, horizontally and vertically flipped images. Specifically, we first perform multi-scale process on each test image, then we slice it into patches with a certain degree of overlap according to its size and carry out detection on these image blocks. Finally, we apply ACNMS to these concatenate bounding boxes from each patch to get the final results.

### *3.4. Adaptive Categorical Non-Maximum Suppression (ACNMS)*

NMS is a post-processing module in the object detection framework, which is mainly used to delete highly redundant bounding boxes. A single remote sensing image may contain one big object or hundreds small objects, thus there exists a class imbalance between different categories. In the previous multi-class object detection works [3,4,33], the NMS thresholds for different categories are the same, but we find that different NMS thresholds for different categories based on the category intensity (CI) can improve the accuracy of object detection to a certain extent. We define CI as:

$$CI = N_{IoC} / N_{img} \tag{3}$$

where $N_{IoC}$ means the total number of instances for each category, $N_{img}$ means the total number of images. If the CI of a category is greater than the given threshold, we set this category a larger NMS threshold than the generic NMS threshold. In general, NMS thresholds for denser objects are larger because they overlap each other more commonly.

## 4. Dataset and Experimental Settings

### *4.1. Dataset Description*

We evaluated our proposed framework on DOTA [48] dataset, which contains 2806 aerial images with pre-divided 1411 training images, 458 validation images and 937 testing images. We note that the testing images have no labels, however, you can submit the test results in a fixed format to DOTA Evaluation Server (http://captain.whu.edu.cn/DOTAweb/evaluation.html). Those DOTA images are obtained from different sensors and platforms with crowdsourcing and the size ranges from 800 $\times$ 800 to 4000 $\times$ 4000 pixels. DOTA consists of 15 common categories, namely, plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool. The fully annotated DOTA dataset contains 188,282 instances, each of which is labeled by an oriented quadrilateral instead of an axis-aligned one, which is typically used for object annotation in natural scene images. Another common geospatial object detection dataset is NWPU VHR-10 [44], which contains 800 images in 10 categories with a total of 3651 instances. The average size of NWPU VHR-10 is 1000 $\times$ 1000 pixels. Compared with NWPU, DOTA is a larger annotated dataset for multi-class geospatial object detection, which has more complex backgrounds, larger image size and denser object distribution thus more reflective of the real-world applications [48]. Therefore, the evaluation on DOTA can better verify the effectiveness and robustness of our proposed network.

The benchmark of DOTA contains two detection tasks. Task 1 uses the initial oriented bounding boxes as ground truth. Task 2 uses the converted horizontal bounding boxes as ground truth. In this work, we only focus on the horizontal bounding box detection task with $(xmin, ymin, xmax, ymax)$ format, so we need to convert the labeled oriented bounding box into the minimum bounding rectangle for each image. Figure 5 shows some examples about the original annotations and their minimum bounding rectangles.

### *4.2. Evaluation Criteria*

We adopted Precision-Recall Curve (PRC) and Average Precision (AP) as evaluation criteria in our experiments, which are widely used in the object detection works.


**Figure 5.** Examples of Annotated Images. The red quadrilaterals represent original annotations, the green rectangles represent minimum bounding rectangles.

#### 4.2.1. Precision-Recall Curve

The precision metric is the ratio of the correct identification quantity to the total identification quantity while the recall metric is the proportion of the correct identification quantity to the total labeled quantity, which can be illustrated by the following two formulas:

$$precision = TP/(TP + FP) \tag{4}$$

$$recall = TP/(TP + FN) \tag{5}$$

we note that if the IoU value between the predicted bounding box and the ground truth is larger than 0.5, it will be considered as true positive $(TP)$, otherwise, it will be considered as false positive $(FP)$. In addition, false negative $(FN)$ refers to the prediction boxes that overlap with ground truth but do not have the maximum overlap value. The precision-recall curve (PRC) describes the relationship between the precision metric and the recall metric, an object detector of a certain category is considered good if its prediction stays high as recall increases.

#### 4.2.2. Average Precision

Average Precision (AP) is the averaged precision across all recall values between 0 and 1, namely, the area under the PRC. A higher AP indicates a better detector. Mean average precision(mAP) represents the average AP over all categories.

### *4.3. Baseline Methods*

We compared the proposed framework with the classic region-based methods including Faster RCNN [37] and FPN [40] on DOTA validation dataset. For the testing dataset, we submitted the inference results to DOTA website because of lacking annotated labels, and we selected several current top-ranked results for comparison.

### *4.4. Implementation Details*

We implemented our network on the open source Caffe2 (https://caffe2.ai/) framework and executed on a 64-bit Ubuntu 16.04 computer with 8GB memory GeForce GTX1070Ti GPU. We note the comparison models were implemented in their original environments without any additions.

#### 4.4.1. Training

We first enlarged and shrunk the original images by a factor of 2 and 0.5 respectively, then we sliced the original and scaled images into patches of $1000 \times 1000$ pixels with an overlap of 500 pixels. All the original image patches, partial randomly selected enlarged and shrunken image patches were taken as our training samples with a total number of 31,396. These training samples will be fed into the network after data augmentation, which includes rotation and flip. We adopted three scales during training, they are $800 \times 800$, $900 \times 900$ and $1000 \times 1000$ pixels respectively. Each scale is the pixel size of a patch's shortest side and the network uniformly select a scale for each training sample at random. We adopted ResNet50 as our backbone, which was pre-trained on ImageNet dataset. We trained a total of 300k iterations with a learning rate of 0.0025 for the first 150k iterations, 0.00025 for the next 50k iterations, and 0.000025 for the remaining 100k iterations, which took us about 40 hours in total. The network was trained by stochastic gradient descent algorithm with a mini-batch of 2 images. Weight decay and momentum are 0.0001 and 0.9 respectively.

#### 4.4.2. Inference

We implemented inference based on the image patches in order to detect as many objects as possible. To accelerate the inference, we sliced validation images into patches of $1000 \times 1000$ pixels with an overlap of 200 pixels. We performed detection on each diced image and then concatenated the predicted results from each patch. We set CI threshold to 10, and the ACNMS threshold is 0.38. Specifically, if the intensity of a category is greater than CI threshold, then its NMS threshold is 0.38, otherwise we set its NMS threshold to 0.3. Meanwhile, to verify the effectiveness of the multi-scale inference strategies, we also performed the same detections on the shrunken images, the horizontal rotation and vertical rotation images simultaneously. We did not perform detections on the enlarged images because of their vastly time-consuming.

## 5. Results and Analysis

### 5.1. Ablation Experiments

Ablation experiments were carried out to verify the effectiveness of the proposed multi-scale training, inference and ACNMS strategies. In the following subsection, we will gradually verify the relevant strategies. The multi-scale training and inference strategies can be expressed as Equation (6):

$$(p)\_based(x) + (s)\_scale \tag{6}$$

where $p$ represents the patch sizes used for training, $x$ represents the patch sources used for training and $s$ represents the patch scales used for inference. For example, 800_based(4)+1_scale means that we resized the pre-divided patches into $800 \times 800$ pixels for training. These multi-scale training data include four data sources, specifically, the original images, the patches obtained from original images, enlarged and shrunken images. During inference, we performed detection on the patches only obtained from original images. The size of these patches is $1000 \times 1000$ pixels with an overlap of 200 pixels. Finally, we concatenated the bounding boxes from each patch and adopted ACNMS to get the final results. The detailed explanations are shown in Table 1.

**Table 1.** Details of multi-scale training and inference strategies.

| Parameters | Connotation | Values | Details |
|---|---|---|---|
| $p$ | Patch sizes used for training | 0 | Training with original images |
| | | 800 | Training with patches of $800 \times 800$ pixels |
| | | 900 | Training with patches of $900 \times 900$ pixels |
| | | 1000 | Training with patches of $1000 \times 1000$ pixels |
| | | (800, 900, 1000) | Training patches with a randomly selected size from ($800^2$, $900^2$, $1000^2$) pixels |
| $x$ | Patch sources used for training | 0 | Original images without slicing |
| | | 1 | Patches from original images |
| | | 4 | Original images, patches from original images, partial randomly selected enlarged and shrunken images simultaneously |
| $s$ | Patch scales used for inference | 0 | Inference on the original images |
| | | 1 | Inference on the patches from the original images |
| | | 4 | Inference on the patches from original images, shrunken images, horizontal and vertical rotation images simultaneously |

#### 5.1.1. Patch-Based Training and Inference Strategies

In this section, we conducted two sets of ablation experiments to illustrate the superiority of patch-based training and inference strategies. We adopted (a), (b), (c), etc. to represent each method in Table 2. In each column, the bold number indicates the best detection result, and the other tables are the same. Table 2(a) carried out training using the original images without patches. For fair comparison, we resized the original images to $1000 \times 1000$ pixels and the inference was also performed on the original images. The training strategies of Table 2(b) were the same as Table 2(a), however, it performed inference on the patches obtained from the original images. Both training and inference of Table 2(c) were performed on the patches obtained from the original images.

**Table 2.** The AP values of ablation experiments for patch-based training and inference strategies.

| Method | 0_based(0)+0_scale (a) | 0_based(0)+1_scale (b) | 1000_based(1)+1_scale (c) |
|---|---|---|---|
| plane | 0.7078 | 0.8015 | **0.8986** |
| ship | 0.6023 | 0.829 | **0.8886** |
| storage tank | 0.4213 | 0.5483 | **0.7808** |
| baseball diamond | 0.6478 | 0.4105 | **0.8112** |
| tennis court | 0.888 | 0.9064 | **0.9078** |
| basketball court | 0.4822 | 0.5279 | **0.6671** |
| ground track field | 0.4304 | 0.4104 | **0.7225** |
| harbor | 0.8391 | 0.7742 | **0.8894** |
| bridge | 0.2973 | 0.308 | **0.6326** |
| large vehicle | 0.675 | 0.7244 | **0.764** |
| small vehicle | 0.5571 | 0.6002 | **0.679** |
| helicopter | 0.3309 | 0.1027 | **0.654** |
| roundabout | 0.2943 | 0.3957 | **0.722** |
| soccer ball field | 0.4059 | 0.3982 | **0.6588** |
| swimming pool | 0.4472 | 0.5328 | **0.6153** |
| mAP | 0.5351 | 0.5513 | **0.7528** |

Comparing Table 2(a) and Table 2(b), we can observe that patch-based inference strategy has improved detection accuracy on most categories except baseball-diamond, ground-track-field, harbor, helicopter and soccer-ball-field. Through further experiments we found that the sizes of baseball-diamond, ground-track-field, harbor, and soccer-ball-field are so large that they often beyond the scope of a single patch, therefore, training with original images but prediction with patches are not conducive to these objects. However, the poor detection effect of helicopter is mainly caused by: (1) Quite a few samples, the sample number (630) of helicopter is far fewer than other categories; (2) Some helicopter samples are similar to airplane, and these two categories generally appear simultaneously. Nevertheless, the patch-based inference strategy is still slightly ascending.

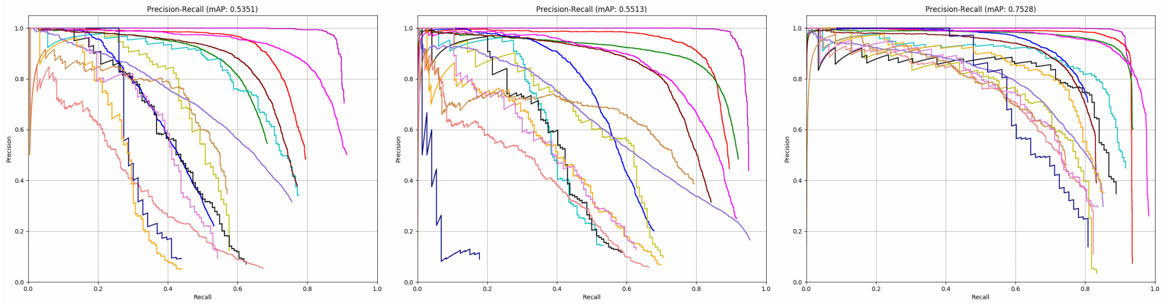
With the patch-based training strategy, Table 2(c) shows the superiority compared to Table 2(b), it not only has an overwhelming advantage in mAP (0.5513 to 0.7528), but also increases the AP value of each category, which illustrates that the patch-based training strategy is targeted and more adequately understand the characteristics of the objects. Besides, the patch-based training strategy implicitly increases the sample number of each category, especially for the sample-scarce categories.

Computational efficiency is also an important indicator in evaluating a framework's performance, so we calculated the average running time for each strategy. The results are shown in Table 3.

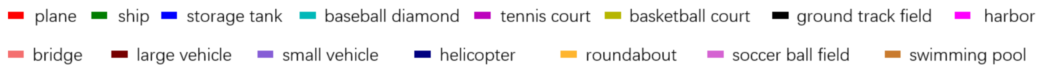
**Table 3.** Average running time of patch-based training and inference strategies.

| Method | 0_based(0)+0_scale (a) | 0_based(0)+1_scale (b) | 1000_based(0)+1_scale (c) |
|---|---|---|---|
| Average running time per image (second) | 0.3882 | 4.2434 | 3.8553 |

We note that the patch-based inference strategies (Table 3(b),(c)) consume more average running time than the original-image-based inference strategy (Table 3(a)), which is easy to understand because the patch-based inference strategy handles more images (patches). In addition, Table 3(c) takes less time than Table 3(b), which further demonstrates that the patch-based training strategy can more adequately extract the characteristics of the objects. The quantified PRCs over two ablation experiments are plotted in Figure 6.




(**a**) The PRC of Table 2(a) (**b**) The PRC of Table 2(b) (**c**) The PRC of Table 2(c)




(**d**) Legend

**Figure 6.** The PRCs of training and inference strategies.

### 5.1.2. Multi-Scale Training Data and Multi-Scale Sizes Used during Training Strategies

Multi-scale training data consist of the original images, patches that based on the original images, the enlarged images and the shrunken images. Multi-scale sizes used during training refers to that an image or patch will be resized to a random scale from specified range before being fed into the framework and each scale is the pixel size of an image or patch's shortest side. We performed two relevant ablation experiments to verify the significance of multi-scale training data and multi-scale sizes used during training. The results are shown in Table 4.

**Table 4.** The AP values of ablation experiments for multi-scale strategies.

| Method | 1000_based(1)+1_scale (a) | 800_based(4)+1_scale (b) | 900_based(4)+1_scale (c) | 1000_based(4)+1_scale (d) | (800,900,1000)_based(4)+1_scale (e) |
|---|---|---|---|---|---|
| plane | 0.8986 | 0.899 | 0.9 | 0.8983 | **0.9007** |
| ship | 0.8886 | 0.8854 | 0.8856 | 0.891 | **0.8919** |
| storage tank | 0.7808 | 0.7781 | 0.7805 | 0.7794 | **0.7817** |
| baseball diamond | 0.8112 | **0.8339** | 0.8199 | 0.8172 | 0.8257 |
| tennis court | 0.9078 | 0.908 | **0.9084** | 0.908 | 0.908 |
| basketball court | 0.6671 | 0.6914 | 0.6976 | **0.7275** | 0.7061 |
| ground track field | 0.7225 | 0.7789 | 0.7681 | **0.7966** | 0.7683 |
| harbor | 0.8894 | 0.8832 | 0.8853 | 0.8894 | **0.891** |
| bridge | 0.6326 | 0.6232 | 0.6306 | 0.6362 | **0.6444** |
| large vehicle | **0.764** | 0.7504 | 0.752 | 0.7636 | 0.7599 |
| small vehicle | 0.679 | 0.6298 | 0.6416 | 0.7182 | **0.7209** |
| helicopter | 0.654 | 0.6815 | 0.7226 | 0.7222 | **0.7385** |
| roundabout | 0.722 | 0.7232 | 0.7173 | 0.7254 | **0.7281** |
| soccer ball field | 0.6588 | 0.6338 | 0.6724 | 0.673 | **0.7122** |
| swimming pool | 0.6153 | 0.7049 | 0.7215 | 0.672 | **0.7253** |
| mAP | 0.7528 | 0.7603 | 0.7669 | 0.7745 | **0.7802** |

The training data used in the Table 4(a) are only from the original images while the training data used in the remaining groups include the original images, the patches from the original images, the enlarged images and the shrunken images. Table 4(b)–(d) resize the training data to 800 × 800, 900 × 900, 1000 × 1000 pixels respectively. Table 4(e) utilizes multiple sizes including (800, 900, 1000) pixels, and the training data will be resized to a randomly selected size before being fed into the network. Apart from this, all experiment settings and inference strategies are identical.

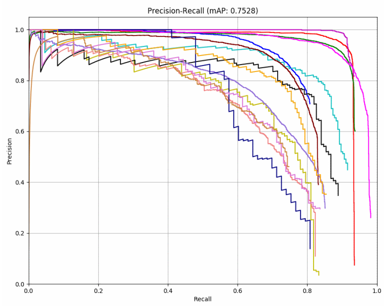
Combining Table 4(a) and Table 4(d), we can find that multi-scale training data can really improve the accuracy (0.7528 to 0.7745), especially for large-size categories such as basketball-court (0.6671 to 0.7275), ground-track-field (0.7225 to 0.7966) and sample-scarce category such as helicopter (0.654 to 0.7222). The accuracy of Table 4(e) is higher than Table 4(b)–(d), which indicates that multi-scale training sizes are helpful in improving the accuracy. Comparisons between Table 4(b)–(d) illustrate that the larger the training image size, the higher the detection average accuracy.

Table 5 shows computational efficiency of multi-scale strategies. Similarly, the comparison between Table 4(a) and Table 4(d) illustrates that multi-scale training data improve the framework performance to a certain extent, so it performs better in terms of computational efficiency. The comparisons between the last four groups reveal that multi-scale sizes used during training not only improve the detection performance but also improve the computational efficiency.

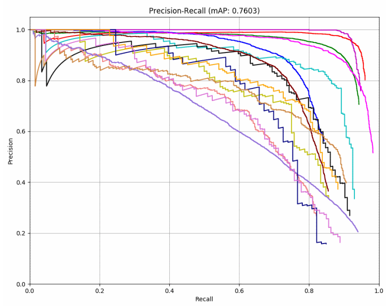
**Table 5.** Average running time of multi-scale strategies.

| Method | Average Running Time per Image (second) |
|---|---|
| 1000_based(1)+1_scale (a) | 3.8553 |
| 800_based(4)+1_scale (b) | 4.103 |
| 900_based(4)+1_scale (c) | 3.864 |
| 1000_based(4)+1_scale (d) | 3.818 |
| (800,900,1000)_based(4)+1_scale (e) | 3.7654 |

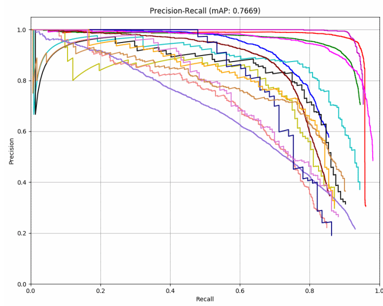
The quantified PRCs over multi-scale training data and multi-scale sizes used during training are plotted in Figure 7.



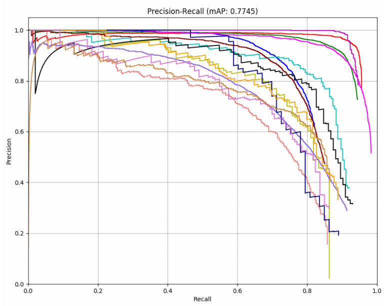

(**a**) The PRC of Table 4(a)



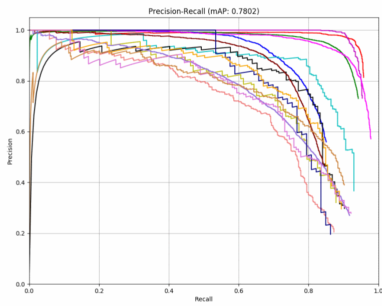

(**b**) The PRC of Table 4(b)



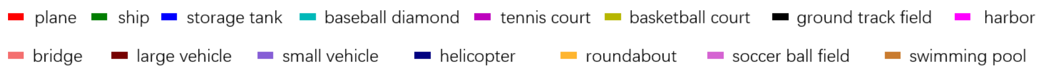

(**c**) he PRC of Table 4(c)




(**d**) The PRC of Table 4(d)




(**e**) The PRC of Table 4(e)




(**f**) Legend

**Figure 7.** The PRCs of multi-scale strategies.

### 5.1.3. Multi-Scale Inference and ACNMS Strategies

We performed multi-scale inference on the original images, the shrunken images, the horizontal rotation and vertical rotation images simultaneously. For small and dense objects mainly including ship, large vehicle and small vehicle, we appropriately increase the NMS threshold according to their CI. The common NMS threshold is 0.3 while the ACNMS threshold is 0.38. The results are shown in Table 6.

**Table 6.** The AP values of ablation experiments for multi-scale inference and ACNMS strategies.

| Method | (800,900,1000)_based(4)+1_scale (a) | (800,900,1000)_based(4)+1_scale$^+$ (b) | (800,900,1000)_based(4)+4_scales (c) | (800,900,1000)_based(4)+4_scales$^+$ (d) |
|---|---|---|---|---|
| plane | 0.9007 | 0.9007 | **0.901** | 0.9004 |
| ship | 0.8919 | 0.8949 | 0.893 | **0.895** |
| storage tank | 0.7817 | 0.7817 | 0.8037 | **0.8037** |
| baseball diamond | 0.8257 | 0.8257 | 0.8265 | **0.8294** |
| tennis court | 0.908 | 0.908 | **0.908** | 0.9079 |
| basketball court | 0.7061 | 0.7061 | 0.7192 | **0.7192** |
| ground track field | 0.7683 | 0.7683 | 0.7985 | **0.7985** |
| harbor | 0.891 | 0.891 | 0.8924 | **0.8924** |
| bridge | 0.6444 | 0.6444 | 0.6652 | **0.6653** |
| large vehicle | 0.7599 | 0.78 | 0.7654 | **0.8201** |
| small vehicle | **0.7209** | 0.7208 | 0.7192 | 0.7183 |
| helicopter | 0.7385 | 0.7385 | 0.7447 | **0.7447** |
| roundabout | 0.7281 | 0.7281 | 0.7553 | **0.7554** |
| soccer ball field | 0.7122 | 0.7122 | **0.7179** | 0.7179 |
| swimming pool | 0.7253 | **0.7253** | 0.7197 | 0.7231 |
| mAP | 0.7802 | 0.7817 | 0.7887 | **0.7927** |

We note that the top right corner "+" in Table 6(b),(d) indicate that we utilized ACNMS strategy. The two comparisons between Table 6(a) and Table 6(c), Table 6(b) and Table 6(d) illustrate the effectiveness of multi-scale inference strategy, which has improved detection performance both in large and small objects such as storage tank, ground track field and roundabout. The two comparisons between Table 6(a) and Table 6(b), Table 6(c) and Table 6(d) illustrate the effectiveness of ACNMS strategy. We slightly improved the NMS threshold of ship, large vehicle and small vehicle because their CIs are far greater than other's. Specifically, the AP values of ship increase by 0.003 and 0.002 respectively in two comparison experiments, the AP values of large vehicle increase by 0.002 and 0.0055 respectively while the AP values of small vehicle remain unchanged. The relevant comparisons illustrate that increasing NMS threshold according to the category intensity does improve the detection accuracy.

Table 7 shows computational efficiency of multi-scale inference and ACNMS strategies. We note that the average running time of multi-scale inference is about three times longer than that of single-scale inference because the number of image (patch) processed by multi-scale inference is about three times more than that of single-scale inference. In addition, using ACNMS strategy does not increase additional average running time.

**Table 7.** Average running time of multi-scale inference and ACNMS strategies.

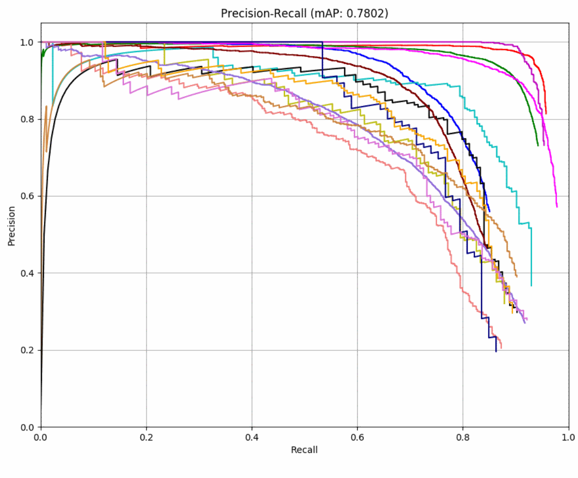
| Method | Average Running Time per Image (second) |
|---|---|
| (800,900,1000)_based(4)+1_scale (a) | 3.7654 |
| (800,900,1000)_based(4)+1_scale$^+$ (b) | 3.7237 |
| (800,900,1000)_based(4)+4_scales (c) | 12.5504 |
| (800,900,1000)_based(4)+4_scales$^+$ (d) | 12.7018 |

The quantified PRCs over multi-scale test and adaptive category NMS strategies are plotted in Figure 8.

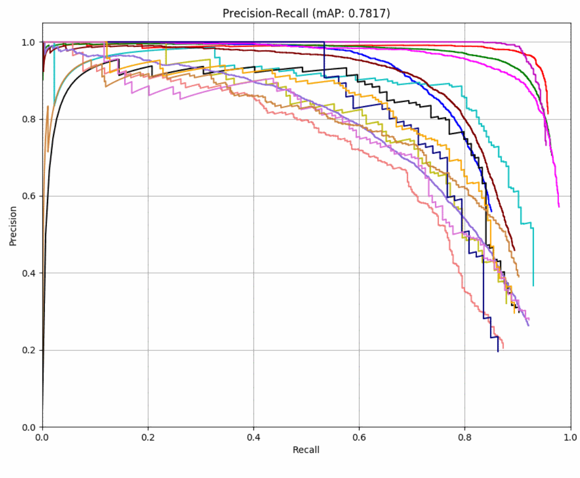
### *5.2. Comparison with Other Methods*

#### 5.2.1. Comparison with Other Methods on DOTA Validation Dataset

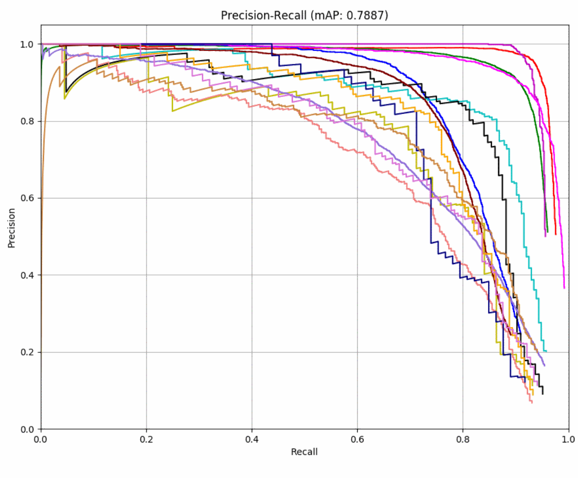
We compared our framework with other region-based object detection networks mainly including Faster R-CNN [37] and FPN [40] on DOTA validation dataset. The selected networks had the same experimental settings as ours, however, they did not adopt our multi-scale training, inference and ACNMS strategies. Table 8 shows the comparison of different networks on DOTA validation dataset.



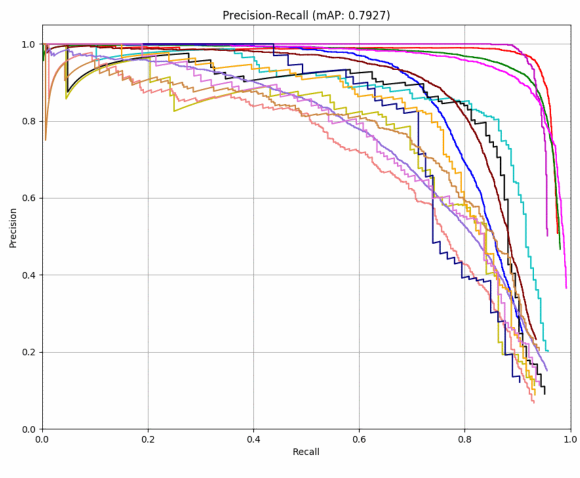

(**a**) The PRC of Table 6(a)



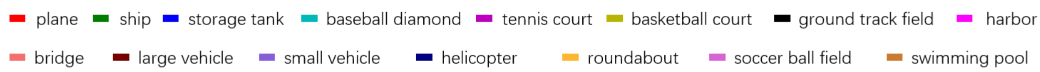

(**b**) The PRC of Table 6(b)




(**c**) The PRC of Table 6(c)




(**d**) The PRC of Table 6(d)




(**e**) Legend

**Figure 8.** The PRCs of multi-scale inference and ACNMS strategies.

We note that Faster R-CNN, FPN and Table 8(c) performed training and inference on the original images instead of patches. The proposed framework has an overwhelming advantage in mAP and AP values of each category. The mAP of Table 8(c) is 0.1712 higher than that of Faster R-CNN and 0.066 higher than that of FPN, which illustrate the superiority of the proposed network. The mAP of Table 8(d) is 0.4163 higher than that of Faster R-CNN, 0.3111 higher than that of FPN and 0.2451 higher than that of Table 8(c) , which illustrate the great superiority of the proposed network and the multi-scale training, inference and ACNMS strategies. The framework has great advantage in detecting small and dense objects such as ship, large vehicle, small vehicle and storage tank. The detection accuracy of sample-scarce objects such as helicopter and roundabout have also been greatly improved, which further confirms that the proposed framework has outstanding performance in detecting both small dense objects and large-scale objects.

**Table 8.** The AP values of ablation experiments with other frameworks on DOTA validation dataset.

| Method | Faster R-CNN (a) | FPN (b) | 0_based(0)+0_scale (c) | (800,900,1000)_based(4)+1_scale (d) |
|---|---|---|---|---|
| plane | 0.4263 | 0.5404 | 0.7078 | **0.9007** |
| ship | 0.0909 | 0.3545 | 0.6023 | **0.8919** |
| storage tank | 0.1907 | 0.2656 | 0.4213 | **0.7817** |
| baseball diamond | 0.4852 | 0.6605 | 0.6478 | **0.8257** |
| tennis court | 0.8141 | 0.8179 | 0.888 | **0.908** |
| basketball court | 0.3612 | 0.4363 | 0.4822 | **0.7061** |
| ground track field | 0.385 | 0.464 | 0.4304 | **0.7683** |
| harbor | 0.5793 | 0.7114 | 0.8391 | **0.891** |
| bridge | 0.1972 | 0.377 | 0.2973 | **0.6444** |
| large vehicle | 0.4911 | 0.6115 | 0.675 | **0.7599** |
| small vehicle | 0.2852 | 0.4004 | 0.5571 | **0.7209** |
| helicopter | 0.3077 | 0.2727 | 0.3309 | **0.7385** |
| roundabout | 0.2312 | 0.3313 | 0.2943 | **0.7281** |
| soccer ball field | 0.3785 | 0.4072 | 0.4059 | **0.7122** |
| swimming pool | 0.2356 | 0.3862 | 0.4472 | **0.7253** |
| mAP | 0.3639 | 0.4691 | 0.5351 | **0.7802** |

The computational efficiency of different frameworks on DOTA validation dataset are shown in Table 9. There is no doubt that the first three groups consume less time than the last group because they performed training and inference on the original images instead of numerous patches. Besides, the proposed DM-FPN (Table 9(c)) can achieve higher object detection accuracy while maintain the same level of computational efficiency.

**Table 9.** Average running time of different frameworks on DOTA validation dataset.

| Method | Faster R-CNN (a) | FPN (b) | 0_based(0)+0_scale (c) | (800,900,1000)_based(4)+1_scale(d) |
|---|---|---|---|---|
| Average running time per image (second) | 0.3268 | 0.2895 | 0.3882 | 3.7654 |

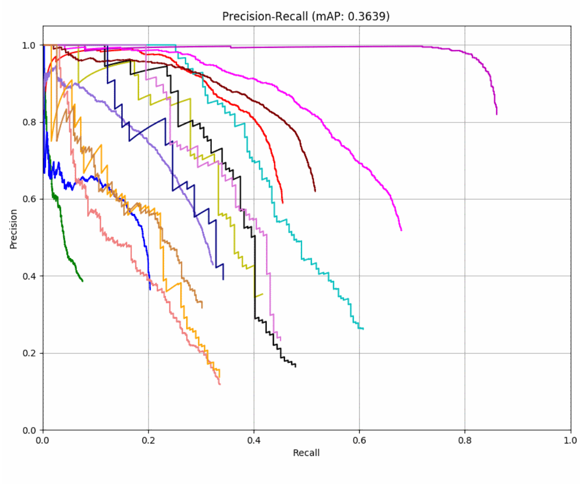
The quantified PRCs over different frameworks on DOTA validation dataset are plotted in Figure 9. We also visualized some detection results as shown in Figure 10.

### 5.2.2. Comparison with Other Frameworks on DOTA Testing Dataset

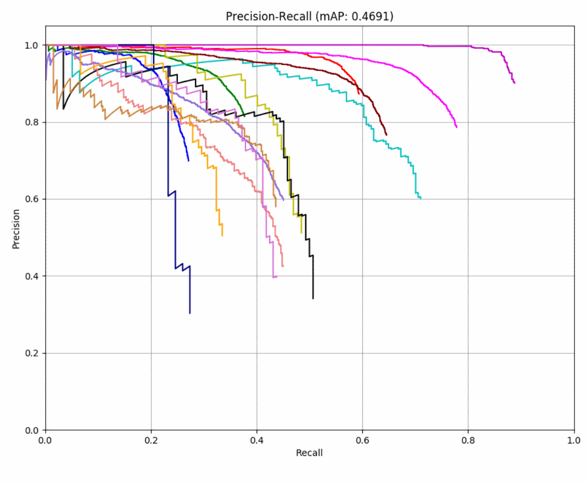
We submitted the inference results based on the testing dataset to DOTA Evaluation Server (http://captain.whu.edu.cn/DOTAweb/results.html) to verify the effectiveness of the proposed framework. Table 10 shows several current top rankings and our DM-FPN achieves the state-of-the-art performance (Our result is named of "CVEO" in Task 2, which achieves the best mAP of 0.793.). Specifically, DM-FPN achieves higher AP on 11 categories, especially in ship, small vehicle, large vehicle and swimming pool, which demonstrates that DM-FPN performs better on small and dense objects. In addition, some large-scale objects such as harbor and ground track field also achieve higher AP than the other frameworks, which further demonstrates that our proposed framework can achieve better results both in small dense objects and large-scale objects. The detection results on DOTA testing dataset are shown in Figure 11.

## 6. Discussion

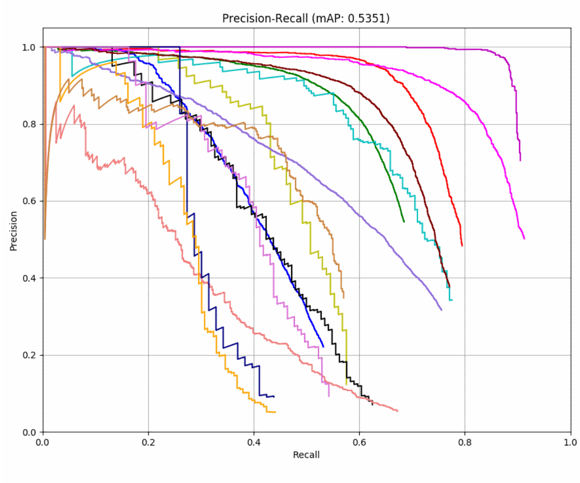
We adopted DOTA dataset to train, verify and test the proposed DM-FPN, which achieved considerable results in the object detection of very-high-resolution optical remote sensing images with RGB three channels. DOTA is the largest dataset for object detection in aerial images, which contains numerous very-high-resolution remote sensing images and 15 common categories. The spatial resolution of the training dataset ranges [0.1, 5] meters, our framework achieves a better performance within this range. The differential spatial resolutions allow the detector to be more adaptive and robust for varieties of objects of the same category. In order to show the overall detection effect, we performed inferences on full images and the results are shown in Figure 12.



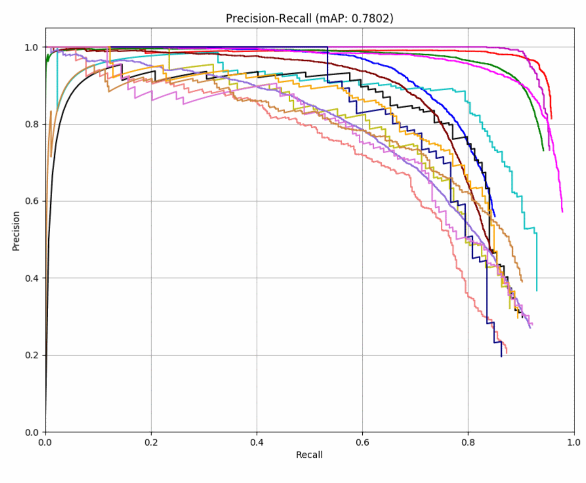

(**a**) The PRC of Table 8(a)



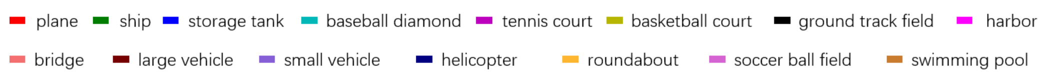

(**b**) The PRC of Table 8(b)




(**c**) The PRC of Table 8(c)




(**d**) The PRC of Table 8(d)




(**e**) Legend

**Figure 9.** The PRCs of different frameworks on DOTA validation dataset.

**Table 10.** The AP values of ablation experiments with other frameworks on DOTA testing dataset.

| Method | changzhonghan | R2CNN_FPN_Tensorflow | FPN with Hobot-SNIPER | Improving Faster RCNN | Ours |
|---|---|---|---|---|---|
| plane | 0.901 | **0.902** | 0.882 | 0.898 | 0.887 |
| ship | 0.851 | 0.781 | 0.839 | 0.851 | **0.873** |
| storage tank | 0.828 | 0.864 | 0.838 | 0.843 | **0.871** |
| baseball diamond | 0.819 | 0.819 | 0.797 | 0.824 | **0.851** |
| tennis court | 0.908 | **0.909** | 0.904 | 0.909 | 0.908 |
| basketball court | 0.836 | 0.824 | 0.803 | 0.797 | **0.848** |
| ground track field | 0.706 | 0.733 | 0.746 | 0.738 | **0.789** |
| harbor | 0.79 | 0.758 | 0.788 | 0.676 | **0.833** |
| bridge | 0.588 | 0.553 | 0.51 | 0.517 | **0.621** |
| large vehicle | 0.82 | 0.776 | 0.767 | 0.733 | **0.833** |
| small vehicle | 0.698 | 0.721 | 0.665 | 0.645 | **0.782** |
| helicopter | **0.646** | 0.638 | 0.601 | 0.499 | 0.64 |
| roundabout | 0.624 | 0.634 | 0.648 | 0.596 | **0.693** |
| soccer ball field | 0.584 | 0.645 | 0.627 | 0.549 | **0.683** |
| swimming pool | **0.8** | 0.782 | 0.753 | 0.737 | 0.782 |
| mAP | 0.759 | 0.754 | 0.738 | 0.73 | **0.793** |

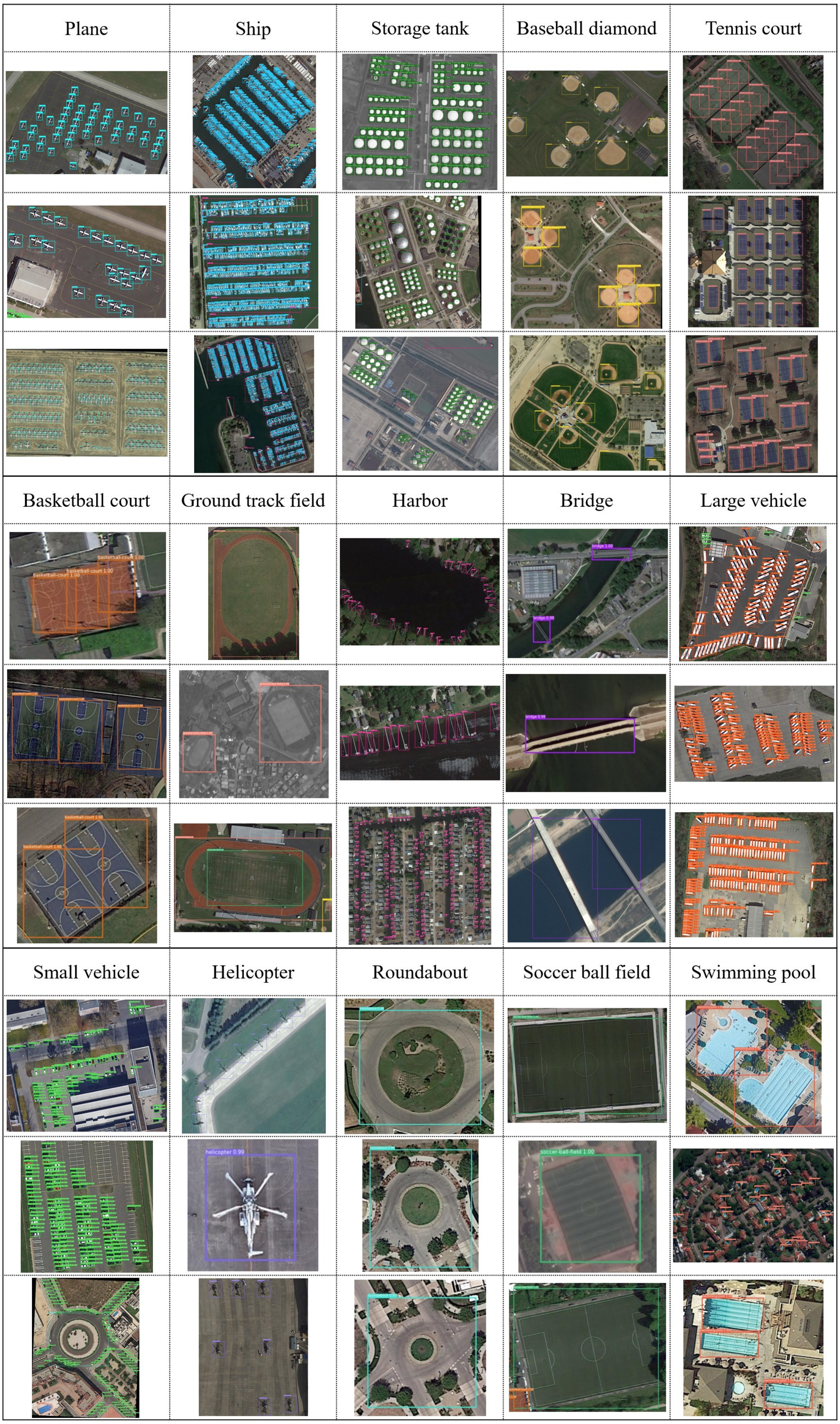



**Figure 10.** Detection results on DOTA validation dataset.

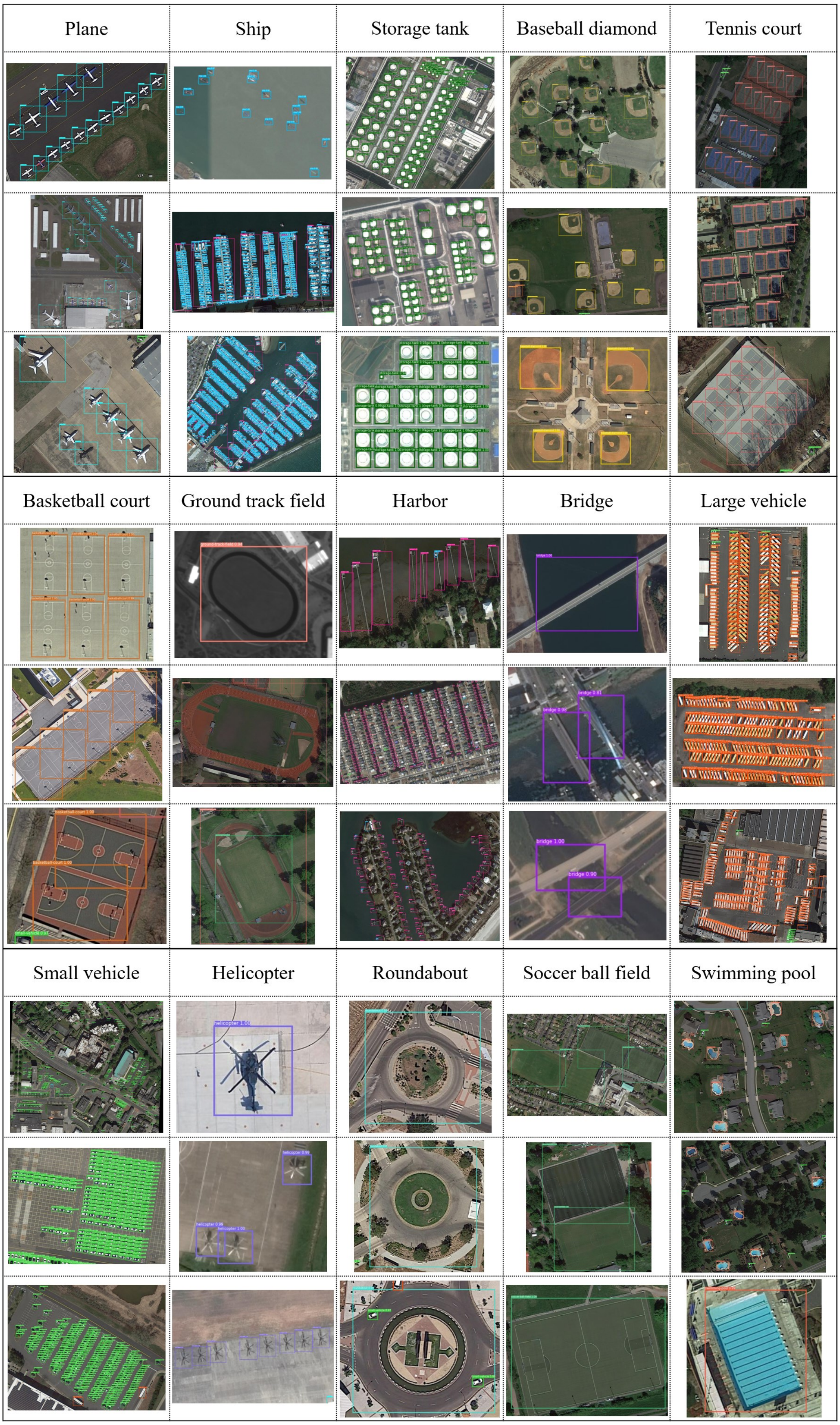



**Figure 11.** Detection results on DOTA testing dataset.

Large vehicle, small vehicle

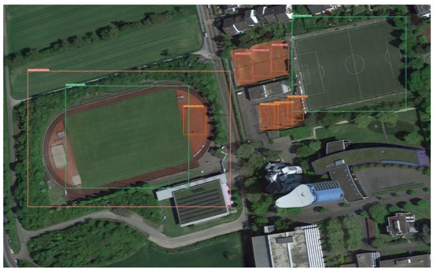
Ship, harbor, small vehicle, swimming pool, roundabout

Swimming pool, roundabout, small vehicle

Small vehicle, storage tank, baseball diamond, tennis court, basketball court, swimming pool

Large vehicle, small vehicle, tennis court, basketball court, swimming pool, soccer ball field

Harbor, ship



Ground track field, soccer ball field, basketball court, tennis court

Plane, small vehicle, large vehicle

**Figure 12.** Detection results on full images of DOTA.

The trained network performs better in detecting the existing 15 categories. However, the detection effects are not satisfactory in detecting the categories or scenes that did not appear in the training dataset, e.g., plane or helicopter over snow. It is also a common problem of all deep learning frameworks. If training samples are provided, the detection can still be performed hopefully.

## 7. Conclusions

In this paper, an effective region-based object detection framework named DM-FPN was proposed to solve small and dense object detection problem in VHR remote sensing imagery. DM-FPN makes full use of coarse-resolution, semantically strong features and high-resolution, semantically weak features simultaneously. We also proposed multi-scale training, inference and ACNMS strategies to solve the problem of the overlarge remote sensing images, the complex image backgrounds and the uneven size and quantity distribution of training samples.

Our framework was experimented on DOTA dataset. The internal ablation experiments (the same framework but different strategies) demonstrate the effectiveness of our proposed strategies while the external ablation experiments (different frameworks) demonstrate the effectiveness of our framework. In addition, we also submitted the inference results based on the testing dataset to DOTA Evaluation Server and DM-FPN achieves the state-of-the-art performance, especially in detecting small and dense objects.

In the future, we will improve our framework's performance in terms of detection speed and accuracy, thus constructing a faster and more accurate network for very-high-resolution remote sensing imagery object detection. At the same time, based on the work of this paper, we will expand our framework to the research of arbitrary-oriented bounding box object detection.

**Author Contributions:** X.Z. guided the algorithm design. K.Z. and G.C. designed the whole framework and experiments. K.Z. wrote the paper. G.C., X.T., L.Z. help organize the paper and performed the experimental analysis. F.D., P.L. help write python scripts of our framework. Y.G. contributed to the discussion of the design. K.Z. drafted the manuscript, which was revised by all authors. All authors read and approved the submitted manuscript.

**Funding:** This research was funded in part by LIESMARS Special Research Funding and the Fundamental Research Funds for the Central Universities.

**Acknowledgments:** The authors would like to thank Prof. Gui-Song Xia from State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University for providing the awesome remote sensing scene classification dataset DOTA. The authors would also like to thank the developers in the Caffe2 and Detectron developer communities for their open source deep learning frameworks.

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

## References

1. Chen, Z.; Zhang, T.; Ouyang, C. End-to-end airplane detection using transfer learning in remote sensing images. *Remote Sens.* **2018**, *10*, 139. [CrossRef]
2. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
3. Guo, W.; Yang, W.; Zhang, H.; Hua, G. Geospatial object detection in high resolution satellite images based on multi-scale convolutional neural network. *Remote Sens.* **2018**, *10*, 131. [CrossRef]
4. Chen, S.; Zhan, R.; Zhang, J. Geospatial object detection in remote sensing imagery based on multiscale single-shot detector with activated semantics. *Remote Sens.* **2018**, *10*, 820. [CrossRef]
5. 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]
6. Stankov, K. 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]
7. Lin, Y.; He, H.; Yin, Z.; Chen, F. Rotation-Invariant Object Detection in Remote Sensing Images Based on Radial-Gradient Angle. *IEEE Geosci. Remote Sens. Lett.* **2014**, *12*, 746–750.

8. Li, Y.; Zhang, Y.; Xin, H.; Hu, Z.; Ma, J. Large-Scale Remote Sensing Image Retrieval by Deep Hashing Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 950–965. [CrossRef]
9. Baltsavias, E.P. Object extraction and revision by image analysis using existing geodata and knowledge: Current status and steps towards operational systems. *ISPRS J. Photogramm. Remote Sens.* **2004**, *58*, 129–151. doi:10.1016/j.ISPRSjprs.2003.09.002. [CrossRef]
10. Leninisha, S.; Vani, K. Water flow based geometric active deformable model for road network. *ISPRS J. Photogramm. Remote Sens.* **2015**, *102*, 140–147. [CrossRef]
11. 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]
12. Blaschke, T. Object based image analysis: A new paradigm in remote sensing? In Proceedings of the 2013 American Society for Photogrammetry and Remote Sensing Conference, Baltimore, MD, USA, 26–28 March 2013.
13. Li, Y.; Wang, S.; Tian, Q.; Ding, X. Feature representation for statistical-learning-based object detection. *Pattern Recognit.* **2015**, *48*, 3542–3559. [CrossRef]
14. Li, X.; Cheng, X.; Chen, W.; Gang, C.; Liu, S. Identification of Forested Landslides Using LiDar Data, Object-based Image Analysis, and Machine Learning Algorithms. *Remote Sens.* **2015**, *7*, 9705–9726. [CrossRef]
15. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2005, San Diego, CA, USA, 21–23 September 2005; Volume 1, pp. 886–893. [CrossRef]
16. Fei-Fei, L.; Perona, P. A Bayesian hierarchical model for learning natural scene categories. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2005, San Diego, CA, USA, 21–23 September 2005; Volume 2, pp. 524–531. [CrossRef]
17. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [CrossRef]
18. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
19. Freund, Y. Boosting a Weak Learning Algorithm by Majority. *Inf. Comput.* **1995**, *121*, 256–285. [CrossRef]
20. Lafferty, J.; Mccallum, A.; Pereira, F.C.N.; Fper, F.P. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. in Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
21. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
22. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; Chen, T. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
23. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
24. Zhang, X.; Chen, G.; Wang, W.; Wang, Q.; Dai, F. Object-Based Land-Cover Supervised Classification for Very-High-Resolution UAV Images Using Stacked Denoising Autoencoders. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3373–3385. [CrossRef]
25. Ma, L.; Li, M.; Ma, X.; Cheng, L.; Du, P.; Liu, Y. A review of supervised object-based land-cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 277–293. doi:10.1016/J.Isprsjprs.2017.06.001. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop, Santiago, Chile, 7–13 December 2015; pp. 1026–1034. [CrossRef]
27. Fu, T.; Ma, L.; Li, M.; Johnson, B. Using convolutional neural network to identify irregular segmentation objects from very high-resolution remote sensing imagery. *J. Appl. Remote Sens.* **2018**, *12*, 1. [CrossRef]
28. 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]
29. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2337–2348. [CrossRef]
30. Cheng, G.; Zhou, P.; Han, J. RIFD-CNN: Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection. In Proceedings of the 2016 IEEE CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 2884–2893. [CrossRef]

31. Everingham, M.; Van~Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *IJCV* **2010**, *88*, 303–338. [CrossRef]
32. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV* **2015**, *115*, 211–252. [CrossRef]
33. 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**, *9*, 666. [CrossRef]
34. Tao, K.; Sun, F.; Yao, A.; Liu, H.; Ming, L.; Chen, Y. RON: Reverse Connection with Objectness Prior Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
35. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 23–28 June 2014.
36. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *IJCV* **2013**, *104*, 154–171. [CrossRef]
37. 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]
38. 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]
39. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
40. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 936–944. [CrossRef]
41. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR2014), Banff, AB, Canada, 14–16 April 2014.
42. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
44. Cheng, G.; Han, J.; Zhou, P.; Guo, L. 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]
45. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
46. Xu, Y.; Zhu, M.; Li, S.; Feng, H.; Ma, S.; Che, J. End-to-End Airport Detection in Remote Sensing Images Combining Cascade Region Proposal Networks and Multi-Threshold Detection Networks. *Remote Sens.* **2018**, *10*, 1516. [CrossRef]
47. Cai, B.; Jiang, Z.; Zhang, H.; Zhao, D.; Yao, Y. Airport Detection Using End-to-End Convolutional Neural Network with Hard Example Mining. *Remote Sens.* **2017**, *9*, 1198. [CrossRef]
48. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, 18–22 June 2018.
49. Zitnick, C.L.; Dollár, P. Edge Boxes: Locating Object Proposals from Edges. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014.
50. Cheng, M.M.; Zhang, Z.; Lin, W.Y.; Torr, P.H.S. {BING}: Binarized Normed Gradients for Objectness Estimation at 300fps. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.
51. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *39*, 640–651.

52. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
53. Cai, Z.; Fan, Q.; Feris, R.; Vasconcelos, N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
54. Honari, S.; Yosinski, J.; Vincent, P.; Pal, C. Recombinator Networks: Learning Coarse-to-Fine Feature Aggregation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
55. Ghiasi, G.; C. Fowlkes, C. Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 519–534. [CrossRef]
56. Pinheiro, P.O.; Lin, T.Y.; Collobert, R.; Dollár, P. Learning to Refine Object Segments. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016
57. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, 1. [CrossRef]
58. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]