remote sensing

MDPI

Article

Extended Feature Pyramid Network with Adaptive Scale Training Strategy and Anchors for Object Detection in Aerial Images

Wei Guo, Weihong Li *, Weiguo Gong and Jinkai Cui

Key Lab of Optoelectronic Technology & Systems of Education Ministry, Chongqing University, Chongqing 400044, China; gwfemma@cqu.edu.cn (W.G.); wggong@cqu.edu.cn (W.G); jinkaicui@cqu.edu.cn (J.C.)

* Correspondence: weihongli@cqu.edu.cn; Tel.: +86-138-836-49662

Received: 15 December 2019; Accepted: 25 February 2020; Published: 1 March 2020

Abstract: Multi-scale object detection is a basic challenge in computer vision. Although many advanced methods based on convolutional neural networks have succeeded in natural images, the progress in aerial images has been relatively slow mainly due to the considerably huge scale variations of objects and many densely distributed small objects. In this paper, considering that the semantic information of the small objects may be weakened or even disappear in the deeper layers of neural network, we propose a new detection framework called Extended Feature Pyramid Network (EFPN) for strengthening the information extraction ability of the neural network. In the EFPN, we first design the multi-branched dilated bottleneck (MBDB) module in the lateral connections to capture much more semantic information. Then, we further devise an attention pathway for better locating the objects. Finally, an augmented bottom-up pathway is conducted for making shallow layer information easier to spread and further improving performance. Moreover, we present an adaptive scale training strategy to enable the network to better recognize multi-scale objects. Meanwhile, we present a novel clustering method to achieve adaptive anchors and make the neural network better learn data features. Experiments on the public aerial datasets indicate that the presented method obtain state-of-the-art performance.

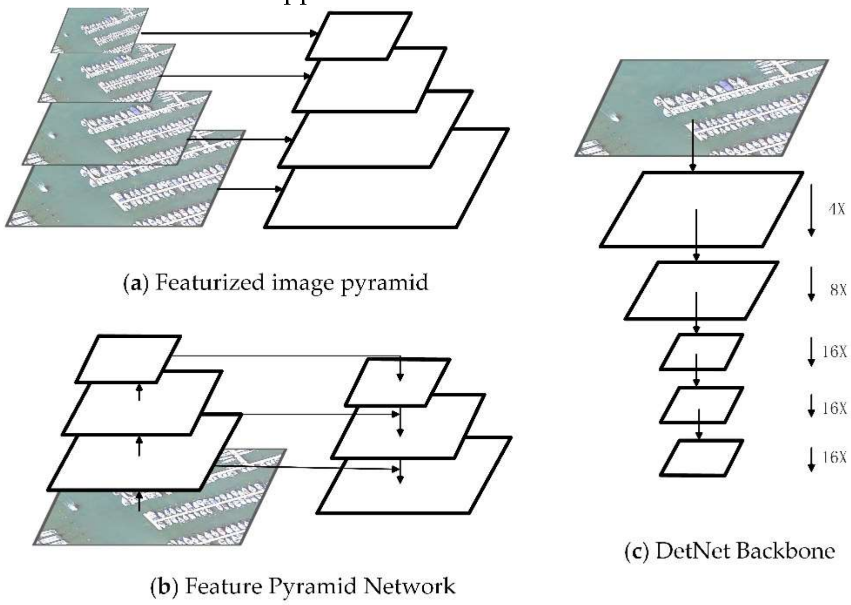
Keywords: aerial images; object detection; extended feature pyramid network (EFPN); adaptive scale training strategy; adaptive anchors

1. Introduction

With the rapid development of deep convolutional neural networks (CNNs) [1] in recent years, the conventional object detection methods [2,3] have made some remarkable achievements in natural images. However, due to the huge scale variations of the vast majority of objects and the compact distribution of many small objects in remote sensing images, it still remains a tremendous challenge for locating and predicting the target objects [4,5].

In order to detect objects at different scales, a basic method is to leverage a multi-scale featurized image pyramid (Figure 1a) [6], which is popular in both manual feature-based approaches [7,8] and deep CNNs-based approaches. Strong evidence [9,10] has shown that the current standard deep detectors can benefit from a multi-scale learning strategy. However, many object detectors based on deep learning have avoided this multi-scale image pyramid representation mainly because it requires a lot of calculations and memories.

Thus, Lin et al. [11] exploited the multi-scale pyramid structure in deep CNNs to construct the Feature Pyramid Network (FPN) with a small amount of additional cost. In the FPN (Figure 1b), it adopts a bottom-up pathway, a top-down pathway and lateral connections for constructing the high-level semantic information at each scale. This structure displays an obvious improvement as a common feature extractor in some practical applications. However, since large-scale objects are usually produced and predicted in the deeper convolution layers of the FPN, the boundaries of these objects might be too fuzzy to obtain accurate regression. Furthermore, the FPN usually predicts small-scale objects in the shallower layers with low semantic information which might not be enough to identify the class of the objects. The designer of the FPN has been aware of this problem and adopted a top-down structure with lateral connections to fuse shallow layers and high-level semantic information to relieve it. However, if the small-scale objects disappear in the deep convolution layers, the context information cues will disappear at the same time.


(**a**) Featurized image pyramid

(**b**) Feature Pyramid Network

(**c**) DetNet Backbone

Figure 1. (**a**) Featurized image pyramid: features are calculated independently on each image scale. (**b**) Feature Pyramid Network. (**c**) DetNet Backbone: due to the limitations of the graphic size, we do not display stage 1 (with stride 2) feature map.

Besides, Li et al. [12] presented a backbone network, called DetNet (Figure 1c), which is the first specifically dedicated to object detection. They pointed out that larger down-sampling factor can bring a larger effective receptive field, which is beneficial to image classification, but may damage the ability of the detector to locate. Therefore, DetNet includes the additional stages contrasted with the conventional backbone network only for classification, and retains high spatial resolution in deeper convolution layers. Due to the specifically designed backbone for object detection, DetNet is much more powerful, especially in finding the small objects and locating the boundaries of the large objects. However, it is useless for the location of small objects and has little contribution to find more ground-truth large objects.

In this work, for solving the above problems, we propose a new FPN-based structure called Extended Feature Pyramid Network (EFPN) which considers huge scale variations of object instances. In the proposed EFPN, we first design a low complexity multi-branched dilated bottleneck (MBDB) module for capturing much more semantic information. The dilated convolution [13,14] is usually blended in the convolution network model to expand the receptive field without increasing the computational complexity. Therefore, the designed MBDB module combines multiple branches with different dilated convolution layers, and is added at the lateral connections of FPN to achieve the feature maps with more details at all scales. Unfortunately, the dilated convolution is likely to cause the loss of some local information due to the increase of the receptive field, which is not beneficial for locating the objects. Recently, some researchers [15] discovered that attention mechanisms can not only focus on the object of interests, but also promote the representation of interests. Thus, we further design an attention pathway to better locate the objects and promote the accuracy of detection. Furthermore, an augmented bottom-up pathway is conducted in the designed

EFPN for making shallow layer information easier to spread and further promoting the performance of small object detection.

In addition, some aerial images are too large for training the CNNs. Therefore, reducing large images by resizing is a common process for saving computing and memory costs during training. However, the resizing process may result in small objects becoming smaller and more likely to be lost in the deeper layers. For solving this problem, the general solution is to simply cut large-scale images into small chunks [5,16]. However, when the cut images include relatively large objects, such as ground track field, these objects may be broken up into small pieces and make the network hard recognize. To better ease this problem, we present an adaptive scale training strategy to try to keep the large objects intact after cutting down in the remote sensing images by designing an adaptive adjustment rate for resizing the original images before dividing these images into smaller sub-images. That is to say, an image is likely to include some relatively large objects whose size may be larger than the sub-image size (we usually set to 800 or 1000 pixels). We can first multiply the original image size by the proposed adaptive adjustment rate to make the large objects with a proper size. Then, we divide the resized image into the smaller sub-images. By adaptively resizing before cutting down the original image, we can make the large objects more intact in the sub-images and promote the recognition ability of the neural network.

Moreover, our proposed EFPN detection framework is built on the faster region-based convolutional neural network (Faster R-CNN) [2] and FPN [11]. For the anchors which are the initialization of candidate boxes in the Faster R-CNN, their aspect ratios and scales are generally set artificially and empirically to several initial values for object detection. The region proposal network (RPN) is presented by Faster R-CNN to replace the original selective search algorithm (SS algorithm) [17] to optimize the generation method for the regions of interest (ROI) [18]. The ROIs are a set of class-independent candidate boxes that may include any objects. By sliding a tiny network on the convolution feature map, the RPN can output a suit of rectangular object proposals called anchors, and each anchor is accompanied by an aspect ratio and a scale. Unlike natural images, some objects in aerial images are of very different shapes and large aspect ratios, such as bridges and harbors. Improper prior scales and aspect ratios setting generally affect the accuracy of the detection positioning. Therefore, it may not be appropriate to directly use the prior scales and the aspect ratios of natural images for remote sensing image detection. For solving this problem, we analyze the training data and propose a special clustering method to obtain the appropriate aspect ratios and scales of anchors.

We did experiments on the public aerial datasets and the results indicate that the presented method obtain state-of-the-art performance. DOTA [19] is a large-scale dataset for object detection in aerial images and the DOTA-v1.5 is the latest version of DOTA-v1.0. NWPU VHR-10 dataset [20] is a publicly available 10-class geospatial object detection dataset. RSOD [21] is an open dataset for object detection in remote sensing images.

The main contributions of this work are summarized as follows:

1. We propose a new framework called Extended Feature Pyramid Network (EFPN) for object detection in aerial images.
2. In the EFPN, we first design the multi-branched dilated bottleneck (MBDB) module in the lateral connections to capture much more semantic information. Then, for better locating the objects, we further design an attention pathway in the deeper layer of EFPN. Finally, an augmented bottom-up pathway is conducted for making shallow layer information easier to spread and further improving performance.
3. We propose an adaptive scale training strategy to try to keep the large objects intact after cutting down in the aerial images and improve the recognition ability of the presented network. Meanwhile, we develop a new clustering method for getting adaptive anchors to replace the initial values which are set artificially.
4. The presented method obtains optimal performance in the challenging DOTA-v1.5 dataset [19], NWPU VHR-10 dataset [20] and RSOD dataset[21].

2. Related works

2.1. Multi-scale Object Detectors.

Object detection with various aspect ratios and scales is an extremely challenging problem in the domain of computer vision. In recent years, CNN has become one of the most effective techniques for object detection. Generally, these CNN-based methods are roughly summarized to two technology paths: one-stage detection methods and two-stage detection methods. Both of these two methods apply a variety of techniques to handle the scale variation problem in multi-category target detection tasks.

In general, the one-stage detection methods are more efficient, because they can classify the predefined anchors directly and further refine them without the proposal generation step. YOLO9000 [3] is a real-time object detection system that can detect over 9000 object categories, and it simply used multi-scale training by selecting new image size randomly per 10 batches to make the neural network model scale-invariant. The single-shot multibox detector (SSD) method [22] achieved multi-scale features by fusing different scale features from different layers without adding additional computation. RetinaNet [23] applied FPN as the backbone and used the focal loss to deal with the imperfection of one-stage object detection that the network suffers from an extreme class imbalance between foreground and background during training. RefineDet [24] selected four feature layers with different stride sizes to deal with objects of different scales.

Besides, the two-stage detection methods first generate a suit of region proposals and then refine them through CNNs. Thus, they usually have better positioning accuracy than the one-stage detection methods. Faster R-CNN [2] improved the Fast R-CNN [25] by developing the RPN to replace the original SS algorithm [17] to optimize the generation method for the ROIs [18]. R-FCN[26] presented a region-based full convolution network and designed the position-sensitive score maps for accurate and efficient object detection. The unified multi-scale CNN (MS-CNN) [27] detected multi-scale objects at multiple layers. Faster FPN [11] is one of the predominant detectors for different scale object detection, which further introduced a top-down structure to promote the semantic information of low-level features. Besides, the presented EFPN is inspired by this architecture.

2.2. Dilated Convolution.

Nowadays, due to its powerful feature extraction ability, deep CNN has obtained great success in the field of object detection. However, there are still some defects in the deep CNN, especially in the design of up-sampling and pooling layers. There are some key problems in the design of up-sampling and pooling layers. First of all, the up-sampling (e.g. bilinear interpolation) and pooling layers are deterministic, which means their parameters are unlearnable. Secondly, in the up-sampling and pooling process, the internal data structures and spatial hierarchy information may be lost and thus the information of small objects cannot be rebuilt.

To solve the above problems, researchers provided many effective structures and the dilated convolution [13] is one of the most excellent structures. In the dilated convolution, it injects a hole in the convolution kernel to enlarge the convolutional kernel with original weights, and the number of the injected holes is determined by one dilation parameter called dilation rate. The purpose of this structure is to provide the greater receptive field without pooling and with the same amount of calculation. The dilated convolution has the characteristics of retaining the internal data structures and avoiding the use of down-sampling. Therefore, using the combination of layers with different dilation rates can improve semantic information. Dilated convolution has been proverbially used in the field of semantic segmentation [28] to better combine local and global context information [14]. In the object detection field, DetNet [12] designed a detection backbone network by introducing dilated convolution in the deeper layers, hence it can hold the spatial resolution and expand the receptive field simultaneously. In this work, we utilize dilated convolution in the presented multi-branched dilated bottleneck (MBDB) module with different dilation rates to extract richer detail information.

2.3. K-means and Mini-batch k-means Algorithm [29].

In the k-means, the distance is used as an evaluation index of similarity, which indicates that the closer the two objects are, the greater the similarity. Considering that the cluster is comprised of close objects, thus the clustering algorithm takes the compact and independent cluster as its ultimate goal. Mini batch k-means algorithm is a variant of k-means algorithm, which utilizes a small-batch subset of data randomly selected to reduce computing time. Using the mini-batch k-means algorithm can greatly reduce the computation time and the results are usually similar to the standard k-means algorithm. The iterative steps of mini-batch k-means can be explained as follows: (1) randomly extract some samples from the dataset to form a small-batch subset and classify them to the nearest center of mass which means the clustering flat of the dataset; (2) update the center of mass.

The mini-batch m-means has a faster convergence speed than the k-means algorithm, while keeping nearly the same clustering effect. K-means clustering has been used in some detectors to obtain the initial size of candidate boxes of the interest area. In order to automatically find the good prior anchors, YOLO9000 [3] ran a k-means clustering directly on the bounding boxes of the training data. In this paper, we obtain the adaptive scales and aspect ratios of anchors by the optimized mini-batch k-means algorithm, which is liable to be realized and can effectively improve detection performance.

3. Proposed method

3.1. Extended Feature Pyramid Network

Figure 2 is the whole architecture of the presented EFPN, which is built on Faster FPN [11] and improves it from different aspects. First, we design the multi-branched dilated bottleneck (MBDB) module in the lateral connections to capture much more semantic information. Then, we further devise an attention pathway to better locate the objects. Finally, an augmented bottom-up pathway is conducted for making shallow layer information easier to spread and further improving performance. On the whole, the proposed EFPN consists of a bottom-up pathway, lateral connections, a top-down pathway, an attention pathway and an augmented bottom-up pathway. The details are described in the following subsections.

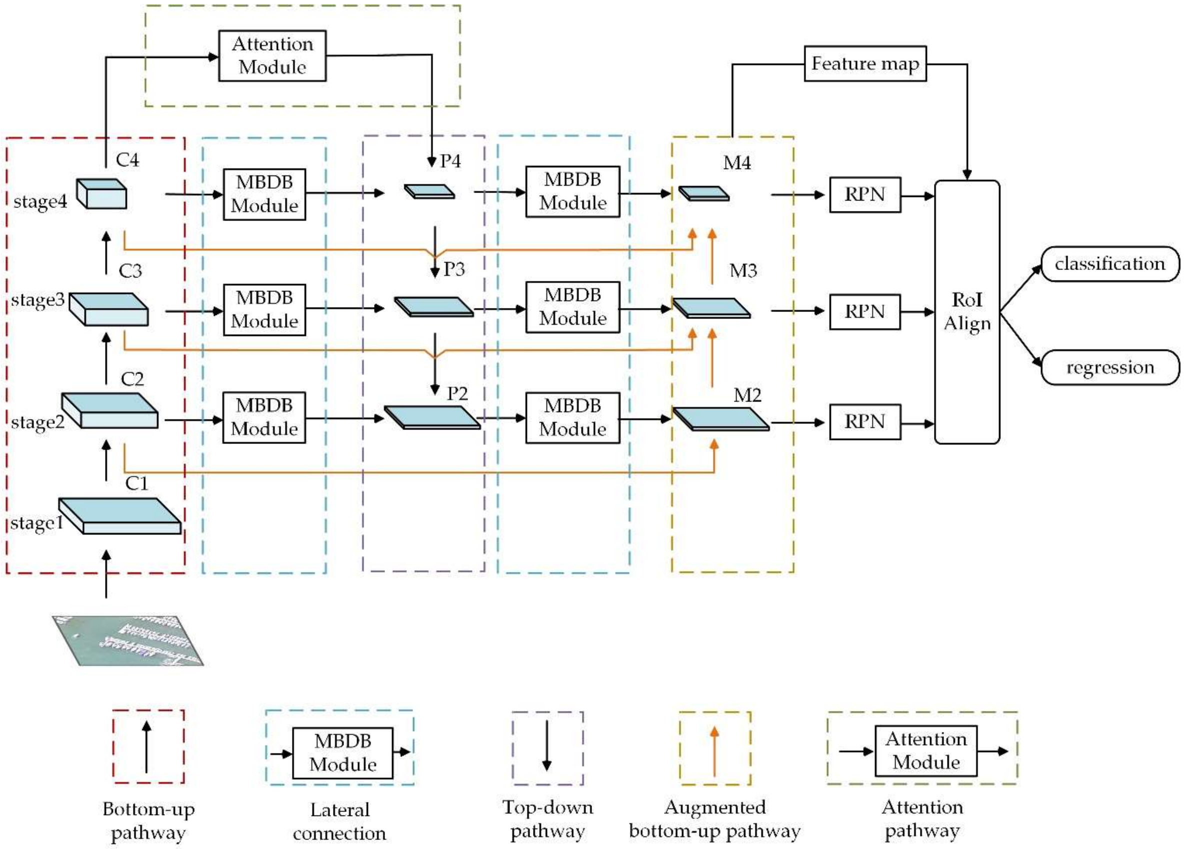


Figure 2. Architecture of the proposed Extended Feature Pyramid Network (EFPN). RPN: region Proposal Network.

Following the EFPN, the RPN is executed at each level of the EFPN output to produce the object proposals. Unlike previous methods using the RoI Pooling operator [2], we adopt the RoI Align operator proposed by Mask R-CNN [30] to extract RoI features. The final detection results are obtained by further precise location regression and fine classification.

3.1.1. Bottom-up Pathway

We adopt the aggregated residual transformations for deep neural networks (ResNeXt-101 32 × 8*d*)[31] as our backbone of the bottom-up pathway. Due to its superior performance in the field of image processing, it is widely used in many object detectors [4]. The backbone usually has many layers that generate feature maps with the same spatial size and we define these layers as stages. The ResNeXt-101 contains five stages. As can be seen from Figure 2, we only use the stage1, stage2, stage3, stage4 of the ResNeXt-101 in our backbone and keep these stages as the same as their original form. The outputs of the last residual block of each stage are expressed as *{C2, C3, C4}*, for which the strides are {4, 8, 16} pixels corresponding to the initial image. They will be extracted to construct the feature pyramid. We do not use stage1 in the pyramid, because it is memory-consuming. The reasons that we do not use the stage5 for the EFPN are as follows. For one thing, traditional backbone networks with the large down-sampling factor can bring a larger effective receptive field, which is beneficial to image classification, but may damage the ability of detector to locate. Thus, the stage5 with a scaling step of 32 is of little use in pinpointing larger objects and adding semantic information of the smaller objects that may have disappeared in this layer. For another, with the proposed MBDB module (described in the lateral connections bellow), we have enlarged the receptive field to get richer semantic information based on stage4. Thus, since the other stages are of little use to our neural network model, we can choose to discard them to save memory and computation.

3.1.2. Lateral Connections with MBDB Module

Considering huge scale variations of aerial object instances and single receptive fields may not effectively learn all situations. The different dilation rates can obtain different scale receptive fields without pooling and with the same amount of calculation. Thus, for the lateral connections, the proposed EFPN employs the low complexity multi-branched dilated bottleneck (MBDB) module to capture much more semantic information. The details of the MBDB module are shown in Figure 3. It can be seen that the MBDB module combines the multiple branches with the dilated convolution layers of different dilation rates to achieve the feature maps with more details at all scales. The MBDB first reduces channel dimensions by a 1 × 1 convolution layer. Then the outputs are divided equally among the three branches, and each branch is a 3 × 3 dilated convolution layer with the different dilation rates that are 3, 2 and 1 respectively. Finally, we append a 1 × 1 convolution on the incorporated feature maps of three branches for producing the final feature map with 256 dimensions. We have experimented with more dilated convolution layers and observed marginally better results. Thus, in order to achieve an approximate optimal effect without introducing too many parameters, we choose to introduce this MBDB module. At each level of the feature pyramid, the presented EFPN can hold the feature map with high spatial resolution and meanwhile retain the large receptive field due to the added MBDB module, thus it has better semantic information capture capability.

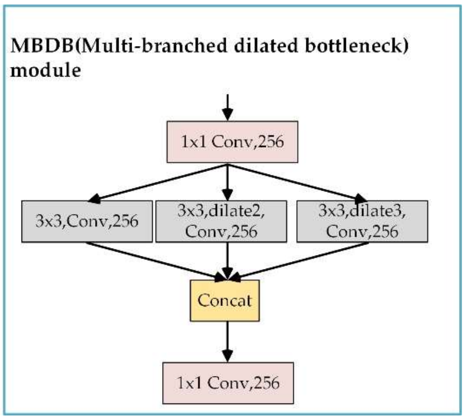


Figure 3. Multi-branched dilated bottleneck (MBDB) module.

3.1.3. Top-down Pathway

For the top-down pathway, we first simply attach an MBDB module on *C4* to generate a coarsest resolution map *P4*. Then, factor 2 is used to conduct spatial resolution up-sampling on the produced feature map *P4*. Finally, we merge the up-sampling map with its corresponding bottom-up map that has attached an MBDB module as the lateral connection. Note that we use nearest neighbor up-sampling for simplicity and element-wise addition for merging here. Repeating this process until the finest resolution map is produced. The final set of feature map can be signed as *{P2, P3, P4}*, which with the same spatial sizes corresponding to *{C2, C3, C4}*, respectively.

3.1.4. Attention Pathway

The designed MBDB module in the lateral connections can capture much more details information. Unfortunately, some local information may be lost by the dilated convolution, which is not beneficial for locating the objects. Thus, to better locate the objects and promote the accuracy of detection, we further design an attention pathway with the attention module. Since the designed attention module can further refine the feature map, the network performs better and has better robustness to noise input. In addition, beyond the previous works [32,33], the proposed attention module can better fuse the global and the detail information at the pixel level through the novel concatenation. The details of the proposed attention module are illustrated in Figure 4.

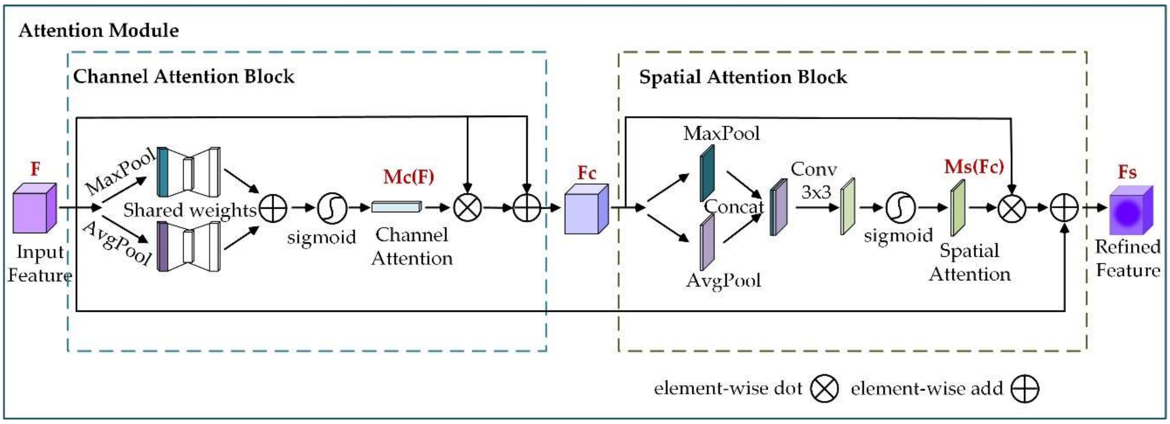


Figure 4. The whole structure of the designed attention module.

The attention module mainly consists of two parts: channel attention block (CAB) and spatial attention block (SAB). The designed details of each block are shown in Figure 4 and the whole attention process of CAB is calculated as follows:

$$F_C = \sigma(M_C(F) \otimes F) \oplus F, \tag{1}$$

where σ is Rectified Liner Unit (ReLU) [34] activation function; $\otimes$ is element-wise dot product; $\oplus$ is element-wise addition; $M_C(F) \in \mathrm{R}^{C\times 1\times 1}$ is the channel attention weight; $F \in \mathrm{R}^{C\times H\times W}$ represents the input feature map and F_C represents the output feature map of the CAB. Concretely, in the CAB, the spatial dimension of the input feature is first compressed by max-pooling and average-pooling simultaneously. Then the generated max-pooling features $F^C_{max} \in \mathrm{R}^{C\times 1\times 1}$ and average-pooling features $F^C_{avg} \in \mathrm{R}^{C\times 1\times 1}$ are followed by two weights-shared full connection layers. The size of the hidden activation layer is set to $\mathrm{R}^{C/r\times 1\times 1}$ for reducing parameter overhead and a ReLU is followed by it. The reduction ratio (r_0) is set as 16. The F^C_{avg} and F^C_{max} can be computed as the following:

$$F^C_{avg}(u^l) = \frac{1}{H\times W}\sum_{\mathrm{i}=1}^{H}\sum_{j=1}^{W} u^l_{ij}\ , 0 < l < C \tag{2}$$

$$F^C_{max}(u^l) = \max\{u^l_{ij} | 0 < i < H, 0 < j < W\}, 0 < l < C \tag{3}$$

where H, W, C and u^l are the height, width, channel and l-th element of feature maps, respectively. The output feature vectors of weight-shared full connection layers are merged via element-wise addition. Finally, the merged vector passes through a sigmoid function for producing our channel attention weight $M_C(F)$, which can be summarized as:

$$M_C(F) = \sigma\left(W_1(\sigma_0\left(W_0(F^C_{avg})\right)) + W_1(\sigma_0(W_0(F^C_{max})))\right), \tag{4}$$

where σ denotes the sigmoid function; σ_0 is ReLU activation function; $W_0 \in \mathrm{R}^{C/r\times C}$ and $W_1 \in \mathrm{R}^{C\times C/r}$ are shared for both max-pooling and average-pooling inputs.

In addition, the whole attention process of SAB is calculated as follows:

$$F_S = \sigma(M_S(F_C) \otimes F_C) \oplus F, \tag{5}$$

where $M_S(F_C) \in \mathrm{R}^{C\times 1\times 1}$ is the spatial attention weight; F_S represents the output feature map of the SAB and it is also the final refined output. Firstly, in the SAB, the feature map $F^S_{avg} \in \mathrm{R}^{1\times H\times W}$ and $F^S_{max} \in \mathrm{R}^{1\times H\times W}$ are produced by the max-pooling and average-pooling processes along the channel axis, respectively. They are calculated by:

$$F^S_{avg}(u^l) = \frac{1}{C}\sum_{l=1}^{C} u^l_{ij}, 0 < i < H, 0 < j < W \tag{6}$$

$$F^S_{max}(u^l) = \max\{u^l_{ij} | 0 < l < C\}, 0 < i < H, 0 < j < W. \tag{7}$$

Then, these two produced maps are concatenated and a convolution layer is applied to reduce the dimension. Finally, a sigmoid function is added to generate the spatial attention weight $M_S(F_C) \in \mathrm{R}^{1\times H\times W}$. In short, the spatial attention weight is computed as:

$$M_s(F_C) = \sigma\left(f^{3\times3}\left([F^s_{avg}; F^s_{max}]\right)\right), \tag{8}$$

where σ represents the sigmoid function; $f^{3\times3}$ is the convolution operation with a filter size of 3×3.

3.1.5. Augmented Bottom-up Pathway

Generally, low-level features are advantageous to access accurate localization information. However, there is a long path passing through even about 100 layers from shallow-level to high-level features in bottom-up pathway of the backbone. Thus, for reducing the loss of information transmission and strengthening the precise position signals existing in the shallow layers, an augmented bottom-up pathway that consists of several layers is adopted in the proposed framework.

Figure 2 shows the designed augmented bottom-up pathway, which is used to produce the new feature map M_{i+1} through a higher resolution feature map M_i, a coarser map P_{i+1} and C_{i+1}. Noting that M_2 is produced only by P_2 and C_2, and the feature maps used in this structure are always with 256-channels. The details are as follows. Firstly, the 3×3 convolution layer with stride 2 is used for reducing the spatial dimension of each feature map M_i and meanwhile getting a down-sampling map. Each corresponding convolutional layer follows a ReLU. Then the generated down-sampling map from M_i, the feature map P_{i+1} which undergoes an MBDB module and C_{i+1} which undergoes a 1×1 convolution layer is added to produce the informative M_{i+1}. Repeating this process until reaching *M4*. Finally, in regards to reducing the aliasing effect, the 3×3 convolution layer is applied on each incorporated map for producing the final feature map *{M2, M3, M4}*.

3.2. Adaptive Scale Training Strategy and Anchors

3.2.1. Adaptive Scale Training Strategy

In remote sensing images, when these are many large images, we may split them into small chips to alleviate the computational and memory cost. Generally, a constant sub-size of sub-images and a constant overlap (*G*) are set to divide the image from left to right and top to bottom of the original image into smaller sub-images. An example of the split process is depicted in Figure 5a. From Figure 5a, the green box denotes one object in the image; *h* and *w* denote the height and width of this object, respectively. When the height and width of the object are less than the sub-size (such as 800 or 1000 pixels), the object can be divided into up to four parts and each part exists in a different sub-image (denoted by the pink, blue, yellow and red dotted boxes respectively). We can call these parts as sub-objects and two adjacent parts intersect each other. The overlap value of two sub-objects is the same as that of the sub-images and denoted by *G*.

In this work, we propose an adaptive scale training strategy by designing an adaptive adjustment rate to resize the original images before dividing these images into smaller sub-images to keep the large object intact after cutting down and reduce the number of difficult samples for large targets. The definition and relative quantity of difficult samples have a great effect on the performance of the neural network model. When a certain category contains many difficult samples in the training data, the neural network model is difficult to learn the characteristics of this category and accurately identify it. Besides, we can use the ratio of the sub-object region to the original object region to define whether one object is a difficult sample or not. If the ratio of the sub-object region to the original object region is lower than a threshold N_t (we generally set it to 0.7), the sub-object in the sub-image is difficult to be detected and we can call it a difficult sample, whereas we call it a simple sample.

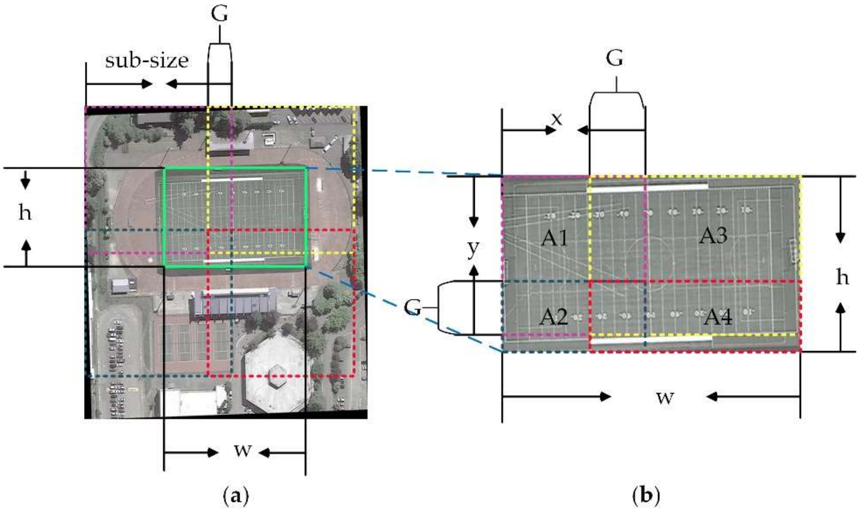


Figure 5. Calculation for adaptive adjustment rate. (**a**) An example of the splitting process. (**b**) A larger version of one object of (**a**).

The object denoted by the green box in Figure 5a is zoomed in to Figure 5b. Figure 5b indicates that the four sub-objects can be represented by the pink box *A1*, the blue box *A2*, the yellow box *A3* and the red box *A4*, respectively. In addition, x and y denote the width and height of the up-left sub-object respectively. So, the areas of these four sub-objects can be expressed by the following formulas:

$$A1 = x \times y \tag{9}$$

$$A2 = (h - y + G) \times x \tag{10}$$

$$A3 = (w - x + G) \times y \tag{11}$$

$$A4 = (h - y + G) \times (w - x + G). \tag{12}$$

If the maximum ratio of these four sub-objects to the original object size is larger than the threshold N_t mentioned above, the object is easy to be detected and it is called as a simple sample. Thus, the condition that this object is a simple sample can be expressed by the following inequality:

$$ratio_{max} = \frac{max(A1, A2, A3, A4)}{w \times h} \geq N_t. \tag{13}$$

The areas of these four sub-objects $(A1, A2, A3, A4)$ are arbitrarily distributed, thus the $ratio_{max}$ is uncertain. Nevertheless, in all cases, when the minimum of $ratio_{max}$ is larger than the threshold N_t, all of the $ratio_{max}$ is larger than the threshold N_t. From the mathematics, the minimum of $ratio_{max}$ is expressed as the following:

$$\min(ratio_{max}) = \left(\frac{w}{2} + \frac{G}{2}\right) \times (\frac{h}{2} + \frac{G}{2})/(w \times h). \tag{14}$$

Thus, the formula (13) can be expressed as below:

$$\left(\frac{w}{2} + \frac{G}{2}\right) \times \left(\frac{h}{2} + \frac{G}{2}\right)/(w \times h) \geq N_t. \tag{15}$$

Therefore, for one image, we know the width and height of the object, and we can set the proper value of overlap G and threshold N_t, then the adaptive adjustment rate (r) can be calculated by the equation:

$$\frac{\left(\frac{r \times w}{2} + \frac{G}{2}\right) \times \left(\frac{r \times h}{2} + \frac{G}{2}\right)}{\mathrm{r \times w \times r \times h}} \geq N_t. \tag{16}$$

When the Equation (16) takes the equal sign and $N_t = 0.7$, the adaptive adjustment rate (r) can be expressed as:

$$\mathrm{r} = \frac{10G(w+h) + G\sqrt{100(w+h)^2 + 720wh}}{36wh}. \quad (17)$$

From the mathematical reasoning, this adaptive adjustment rate is also applicable to other cases not discussed here. The adaptive adjustment rate (r) is a preset variable before image cutting, which is determined by the width (*w*), height (*h*), overlap (*G*) and threshold N_t. The threshold of sub-object to the original object ratio (N_t) is generally set to 0.7 to judge whether one object is a difficult sample or not. In general, when N_t is given empirically, the adaptive adjustment rate (*r*) can be derived according to the width (*w*), height (*h*) and overlap (*G*). Using the proposed adaptive adjustment rate to resize the original image before dividing the images into smaller sub-images can ensure most of the sub-objects are the simple samples and improve the recognition ability of the neural network.

3.2.2. Adaptive Anchors

The aspect ratio represents the rate of the width of the anchor to its height. When an anchor is square, the scale is the side of this anchor. In practical applications, we may detect some objects with special shapes, such as bridges and harbors in remote sensing images. At this time, the general initialization of the anchor box size will have an impact on the accuracy of the final training model and we need to generate the corresponding anchor box size according to our own data instead of the default value.

For the aspect ratios of anchors, when the amount of data is very huge in some remote sensing image detection tasks, it is a huge time drain and might not be necessary if we directly apply the k-means algorithm for obtaining the appropriate aspect ratios of anchors by clustering the aspect ratios of training data. To solve this problem, we substitute mini-batch k-means algorithm for the k-means algorithm to reduce the calculation time. However, there are still the main two problems to cluster an exact result. (1) Because the mini-batch k-means algorithm is sensitive to the selection of initial centroids, the results of each training for clustering may not be the same and precise enough; (2) the number of cluster centers should be specified in advance and the different numbers of cluster center points will have very different results. The number of artificially assigned cluster centers, which also is the number of the adaptive aspect ratios of anchors at this time, may not be the best one for the optimal results.

To address these problems, we first randomly initialize the cluster centroids for many times and calculate the values of the loss function every time. The value of the loss function represents the average Euclidean distance between every data sample and its corresponding closest cluster center. Then, the cluster centroids corresponding to the minimum result of the loss function are taken as the clustering results. The values of cluster centroids are the adaptive aspect ratios of anchors. The loss function is expressed as follows:

$$\mathrm{J}\left(c^{(1)}, \dots, c^{(m)}, u_1, \dots, u_K\right) = \frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - u_{c^i} \|^2 \,. \quad (18)$$

where *m* represents the number of the data; $x^{(i)}$ is one sample and u_{c^i} is the corresponding closest cluster center of this sample; *K* represents the number of cluster centers and it is a pre-defined hyper-parameter that we can set it from two to eight. Finally, the Elbow Method [35] is used to determine the appropriate number of cluster centers *K* which also is the number of the adaptive aspect ratios of anchors here, and it is a great tradeoff between high recall and complexity of the neural network model.

For the scales of anchors, we use the mini-batch k-means algorithm with the Intersection of Union (IoU) distance [3] instead of the aforementioned Euclidean distance to obtain the adaptive settings. In [3], it expresses that if we utilize standard k-means with Euclidean distance directly, larger boxes bring forth more error than smaller boxes, but the IoU distance is independent of the box size. The IoU distance is denoted as:

$$\mathrm{D}(\mathrm{box}, \mathrm{centroid}) = 1 - \mathrm{IoU}(\mathrm{box}, \mathrm{centroid}). \tag{19}$$

We ran mini-batch k-means for various values of *K*. For one set *K*, as the same to get the adaptive aspect ratios of anchors, we also randomly initialize the cluster centroids for many times and calculate the value of average IoU distance every time. Then, we select the cluster centroids corresponding to the minimum of average IoU distance. With the various *K* and the corresponding minimum of average IoU distance, Elbow Method [35] is also used to determine the appropriate *K*. At this time, the cluster centroids represent the width and height of the prior anchors, and the scales of anchors can be acquired by calculating the side of the square which has the same area as the produced prior anchors.

Furthermore, the mini-batch k-means algorithm is not only simple and liable to implement, but also can greatly promote the detection performance. Hence, it can be further explored and applied to other parameters of the neural network model in future work.

4. Dataset and Experimental Settings

To verify the effectiveness of the presented method, we execute comparative experiments on public aerial DOTA-v1.5 datasets [19], NWPU VHR-10 dataset [20] and RSOD dataset [21]. The dataset description, implementation details and evaluation criteria will be discussed in this section.

4.1. Dataset Description

4.1.1. DOTA-v1.5 Dataset

The DOTA-v1.5 is the latest version of DOTA-v1.0. They all have identical aerial images, but DOTA-v1.5 modifies and updates the annotations of the object. In DOTA-v1.0, many small object instances of about 10 pixels or less were omitted, and additional annotations were made in DOTA-v1.5. The DOTA-v1.5 dataset collates a total of 2806 aerial images and includes 400,000 annotated object instances in 16 categories, containing plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), helicopter (HC) and container crane (CC). Furthermore, all of the instances are in two forms of annotation: the oriented bounding boxes (OBB) and the horizontal bounding boxes (HBB). For ensuring that the distribution of training data and test data is roughly matched, 1/2 of the dataset is randomly selected as a training set, 1/6 as a verification set and 1/3 as a test set. Besides, the analysis result of the training part of the DOTA-v1.5 and DOTA-v1.0 datasets is shown in Table 1. It should be noted that the number of images in this table is not the actual number of the images of the training data, because one image may contain more than one category. As shown in Table 1, the main object instances added are small vehicles and large vehicles. In addition, the category of a container crane is added in the DOTA-v1.5. Therefore, it requires the neural network model to have more powerful capability for the detection of small objects and stronger robustness for the detection of multi-scale objects.

4.1.2. NWPU VHR-10 Dataset

In the NWPU VHR-10 dataset [20], it consists of 800 images (about 1000×1000) with 650 positive objects and 150 negative objects. It includes ten categories, which are Airplane, Ship (SH), Storage tank (ST), Baseball diamond (BD), Tennis court (TC), Basketball court (BC), Ground track field (GTF), Harbor (HR), Bridge (BR) and Vehicle. In this paper, the dataset is randomly split into a training set, a verification set, and a test set according to the proportion of 20%, 20% and 60%.

4.1.3. RSOD Dataset

The RSOD dataset [21] has 936 annotated images, including 4993 aircrafts, 191 playgrounds, 180 overpasses and 1586 oil tanks. In this paper, the dataset is randomly split into a training set, a verification set and a test set according to the ratio of 25%, 25% and 50%.

Table 1. Analysis results of the training part of the DOTA-v1.5 and DOTA-v1.0 dataset. The number of images in this table is not the actual number of images, because one image may contain more than one category.

Object Classes	**Number of Images**		**Number of Instances**	
	DOTA-v1.5	**DOTA-v1.0**	**DOTA-v1.5**	**DOTA-v1.0**
Plane	198	197	8072	8055
Baseball-diamond	to 121	122	412	415
Bridge	213	210	2075	2047
Ground-track-field	180	177	331	325
Small-vehicle	904	486	126,686	26,126
Large-vehicle	545	380	22,400	16,969
Ship	435	326	32,973	28,068
Tennis-court	308	302	24,38	2367
Basketball-court	116	111	529	515
Storage-tank	202	161	5346	5029
Soccer-ball-field	130	136	338	326
Roundabout	182	170	437	399
Harbor	340	339	6016	5983
Swimming-pool	226	144	2181	1736
Helicopter	32	30	635	630
Container-crane	7	-	142	-

4.2. Implementation Details

The experiments are performed on the Detectron [36] platform. The proposed detector was trained on an Nvidia GTX 1080Ti GPU with 11 GB of RAM and optimized with synchronous stochastic gradient descent (SGD) by setting the 0.0001 for weight decay and 0.9 for momentum. In each mini-batch, there is only one image. For the DOTA-v1.5 dataset, we first chipped the images into 1024 × 1024 sub-images and set the overlap value to 200 due to the high resolution of these images. Then for those cut sub-images of DOTA-v1.5 and the other two datasets with relatively not high resolution of the image (about 1000×1000), the short edge was resized to 800 pixels and the long edge was limited to 1000 pixels. Finally, the proposed network was learned a total number of 180k iterations on the DOTA-v1.5 dataset. Before 140k iterations, the learning rate was 0.001 and it was reduced by a factor of 10 every next 20k iterations. Besides, for the other two datasets, we also chose this learning policy of step with decay but only a total number of 45k iterations. Before 30k iterations, the learning rate was 0.001 and it was reduced by a factor of 10 for the next 15k iterations. All the experiments were initialized with common objects in context (COCO) [37] pre-trained weights. For data augmentation, we did not execute data augmentation processing except random flipping images during training. As for ROI generation, we first picked up 10,000 proposals with the highest scores and then got 2000 ROIs at most by the NMS procedure. Furthermore, the effective Group normalization (GN) [38] and ROI-Align [30] techniques were used in the proposed EFPN.

4.3. Evaluation Criteria

In the experiments, we utilized the precision-recall curve (PRC) and the average precision (AP) as the evaluation criteria. The PRC depicts the correlation between the precision value and the recall rate which can be formulated as follows:

$$\text{Precision} = TP/(TP + FP) \tag{20}$$

$$\text{Recall} = TP/(TP + FN), \quad (21)$$

where *TP*, *FP* and *FN* are the number of true positives, false positives and false negatives, respectively. In general, if the particular detector can maintain high precision with the increase of the recall rate, it is considered to be excellent in performance.

The region area under the PRC is AP, which is the average precision of all recall values from 0 to 1. The mean average precision (mAP) denotes the average precision value for all categories. Note that the higher the value of AP, the better the performance of the detector. In addition, we evaluate the detections of small, medium and large objects with different scales which range from 1 pixel to 50 pixels, 50 pixels to 300 pixels and over 300 pixels, respectively. By calculating the average of the AP values of different scales in each category, the mean average precision of each scale is acquired, and the AP of the corresponding small, medium and large objects scales are represented by AP_S, AP_M, AP_L respectively.

5. Results

5.1. Ablation Experiments

5.1.1. Ablation for EFPN

To verify the effectiveness of each part of the presented EFPN, we compare the performance changes when separately adding the MBDB module, the attention pathway (AP) and the augmented bottom-up pathway (ABUP) to the baseline FPN on the DOTA-v1.5 validation set. The first to the fourth row of Table 2 demonstrate the comparison results. The combination strategies are FPN with the multi-branched dilated bottleneck module (FPN+MBDB), FPN with the attention pathway (FPN+AP), FPN with the augmented bottom-up pathway (FPN+ABUP). Compared with FPN, all combinations yield better results, increasing mAP by 2.46%, 2.87% and, 2.18% respectively. The EFPN achieves the best result with mAP value of 74.67%. In remote sensing images, there are objects with vastly different scales and the scale AP of some categories is small, so the average scale AP (AP_S, AP_M, AP_L) for all categories is generally smaller than mAP. As shown in Table 2, for the AP_S of each combination (FPN+MBDB, FPN+AP, FPN+ABUP), they all have a certain boost relative to FPN. In addition, we also compare the parameters (Params), computational cost (FLOPs) and average run time per image of the baseline FPN and the proposed EFPN as well as each combination (FPN+MBDB, FPN+AP, FPN+ABUP). Table 3 shows the detailed comparison results. As described in section 3.1.1, we discard the stage5 of the backbone for the EFPN to save memory because it is of little use to our neural network model. Thus, the EFPN and all combinations (FPN+MBDB, FPN+AP, FPN+ABUP) contain fewer parameters than the traditional FPN. Due to some extra operation, the floating-point operations (FLOPs) still increase for the final EFPN, but the average run time per image has only a small fluctuation.

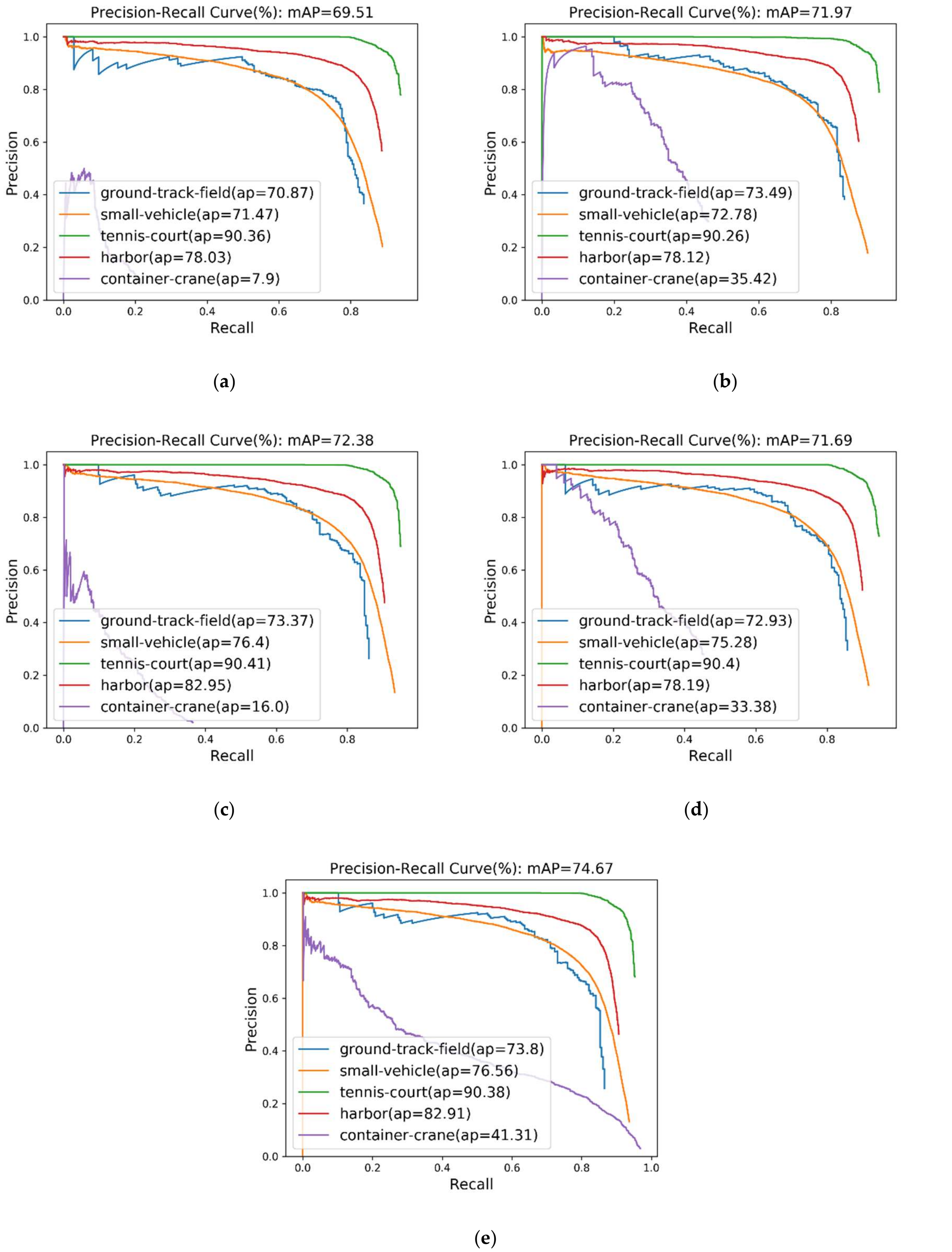
Table 2. The comparison results for the FPN and EFPN on the DOTA-v1.5 validation dataset. FPN: Feature Pyramid Network. FPN+MBDB: FPN with the multi-branched dilated bottleneck module which is described in section 3.1.2. FPN+AP: FPN with the attention pathway. FPN+ABUP: FPN with the augmented bottom-up pathway. EFPN: Extended Feature Pyramid Network.

Method	FPN	MBDB	AP	ABUP	AP_S	AP_M	AP_L	mAP(%)
FPN[11]	√	-	-	-	52.91	62.38	38.41	69.51
FPN+MBDB	√	√	-	-	64.72	65.21	46.31	71.97
FPN+AP	√	-	√	-	60.93	64.62	48.97	72.38
FPN+ABUP	√	-	-	√	63.32	63.78	42.89	71.69
EFPN	√	√	√	√	66.11	70.16	50.38	**74.67**

The precision-recall curves of the 5 object classes in Table 2 are shown in Figure 6. These 5 categories are selected from all 16 categories and they have a visible distinction in performances between network architectures. The recall rate assesses the ability to detect more objects, but the precision measures the ratio of correct objects to all detected objects. Therefore, as the curve decreases sharply, the higher the recall rate, the better the detection effect of the class. Due to the high similarity between the object and the background, and the lack of training samples, the container cranes are poorly recognized in the FPN and the proposed method can promote the detection effect to a certain extent. For each combination (FPN+MBDB, FPN+AP, FPN+ABUP), the small objects (such as small vehicles) and the large objects (such as ground track field) both get better detection results. From Figure 6e, for the presented EFPN, the recall curves of most object classes begin to decline sharply when the recall value exceeds 0.8. It is because the proposed EFPN has stranger semantic information extraction ability and nice detection performance.

Table 3. The detailed comparison at each operation stage of EFPN.

Method	**Params**	**FLOPs**	**Average Run Time (s)**
FPN[11]	8.98×10^7	2.34×10^{11}	0.20 s
FPN+MBDB	6.64×10^7	3.24×10^{11}	0.21s
FPN+AP	6.03×10^7	2.16×10^{11}	0.20s
FPN+ABUP	6.59×10^7	3.19×10^{11}	0.21s
EFPN	7.25×10^7	4.27×10^{11}	**0.23s**



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

(**c**) (**d**)

(**e**)

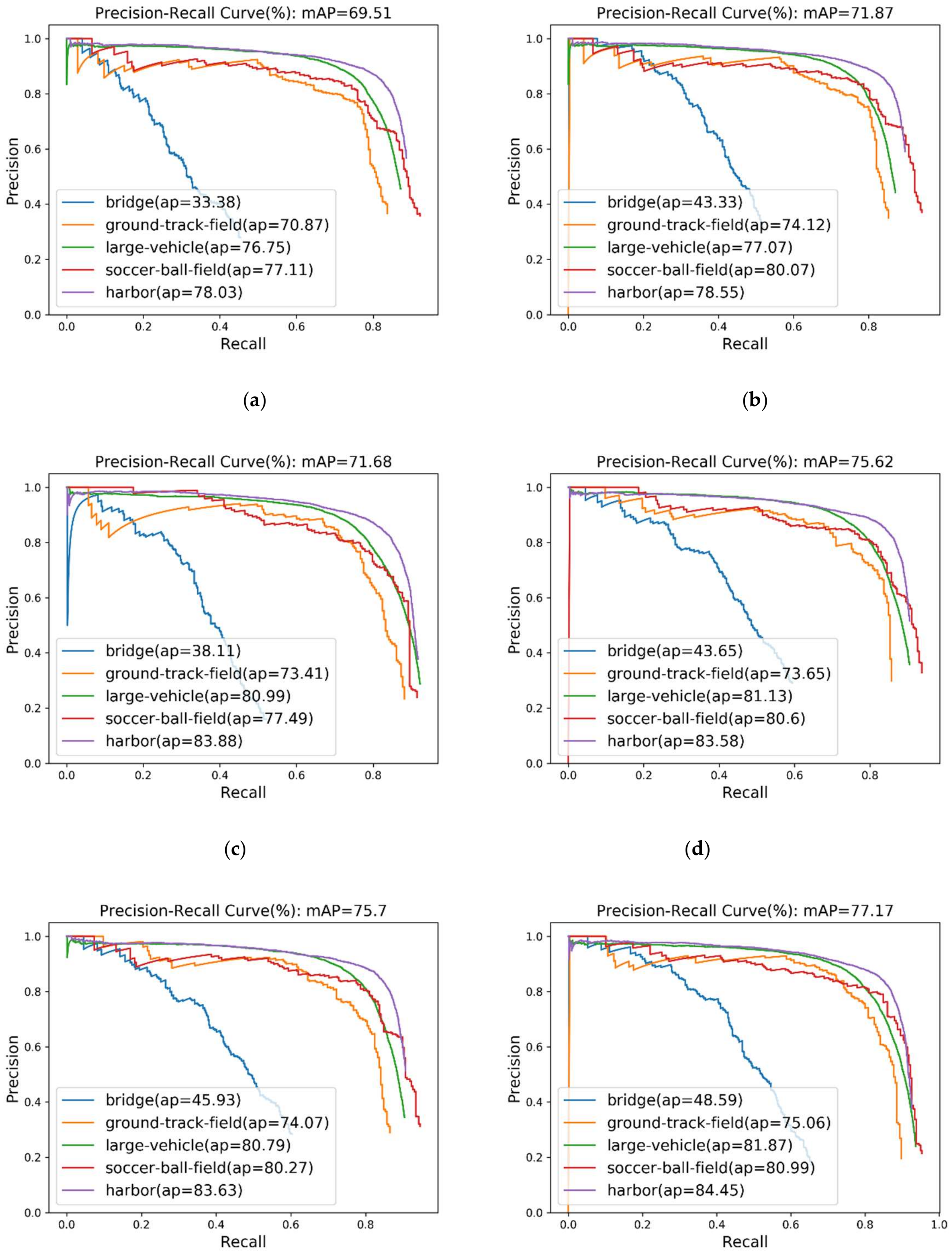
Figure 6. The precision-recall curves of Table 2: (**a**) FPN: Feature Pyramid Network. (**b**) FPN + MBDB: FPN with the multi-branched dilated bottleneck module. (**c**) FPN + AP: FPN with the attention pathway. (**d**) FPN + ABUP: FPN with the augmented bottom-up pathway. (**e**) EFPN: Extended Feature Pyramid Network.

5.1.2. Ablation for the Adaptive Scale Training Strategy and Anchors

For assessing the validity of the adaptive scale training strategy and anchors which are described in section 3.2, we compare the baseline FPN with FPN+AS and FPN+AA on the DOTA-v1.5 validation set, where the AS and AA are used to shortly represent the adaptive scale training strategy and adaptive anchors, respectively. In addition, we also tested the three combination methods with EFPN, which were EFPN+AA, EFPN+AS and EFPN++ (EFPN+AA+AS). The comparison results are exhibited in Table 4. It should be noted from Table 4 that after the proposed methods were added, the detection results can be improved with varying degrees. Compared with FPN, the incorporated FPN+AS and FPN+AA increase the total mAP by 2.36% and 2.17%, respectively. In the combination works, the EFPN++ achieves the highest mAP value 77.17% compared with EFPN+AS and EFPN+AA, increasing the mAP by 1.55% and 1.47%, respectively. In addition, compared with FPN and EFPN, the AP_L of FPN+AS and EFPN+AS both have improved because the proposed adaptive scale training strategy can promote the detection of the large object. The precision-recall curves of Table 4 over the 5 classes are shown in Figure 7. From Figure 7a and Figure 7b, we can also see that the proposed FPN+AS can improve the detection of large objects, such as ground track field and soccer ball field. Moreover, Figure 7a and Figure 7c show that the proposed FPN+AA can promote the detection of objects with special shapes, such as bridge and harbor.

Table 4. The results of comparison for using the adaptive scale training strategy and anchors on the DOTA-v1.5 validation dataset. EFPN: Extended Feature Pyramid Network. AS: adaptive scale training strategy. AA: adaptive anchors. EFPN++: EFPN with the adaptive scale training strategy and the adaptive anchors.

Method	**AS**	**AA**	**AP_S**	**AP_M**	**AP_L**	**mAP(%)**
FPN[11]	-	-	52.91	62.38	38.41	69.51
FPN+AS	√	-	65.93	64.87	49.40	71.87
FPN+AA	-	√	63.31	66.49	46.42	71.68
EFPN	-	-	66.11	70.16	50.38	**74.67**
EFPN+AS	√	-	67.29	71.17	52.33	75.62
EFPN+AA	-	√	67.79	70.89	50.83	75.70
EFPN++	√	√	70.15	72.82	53.86	**77.17**



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

(**c**) (**d**)

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

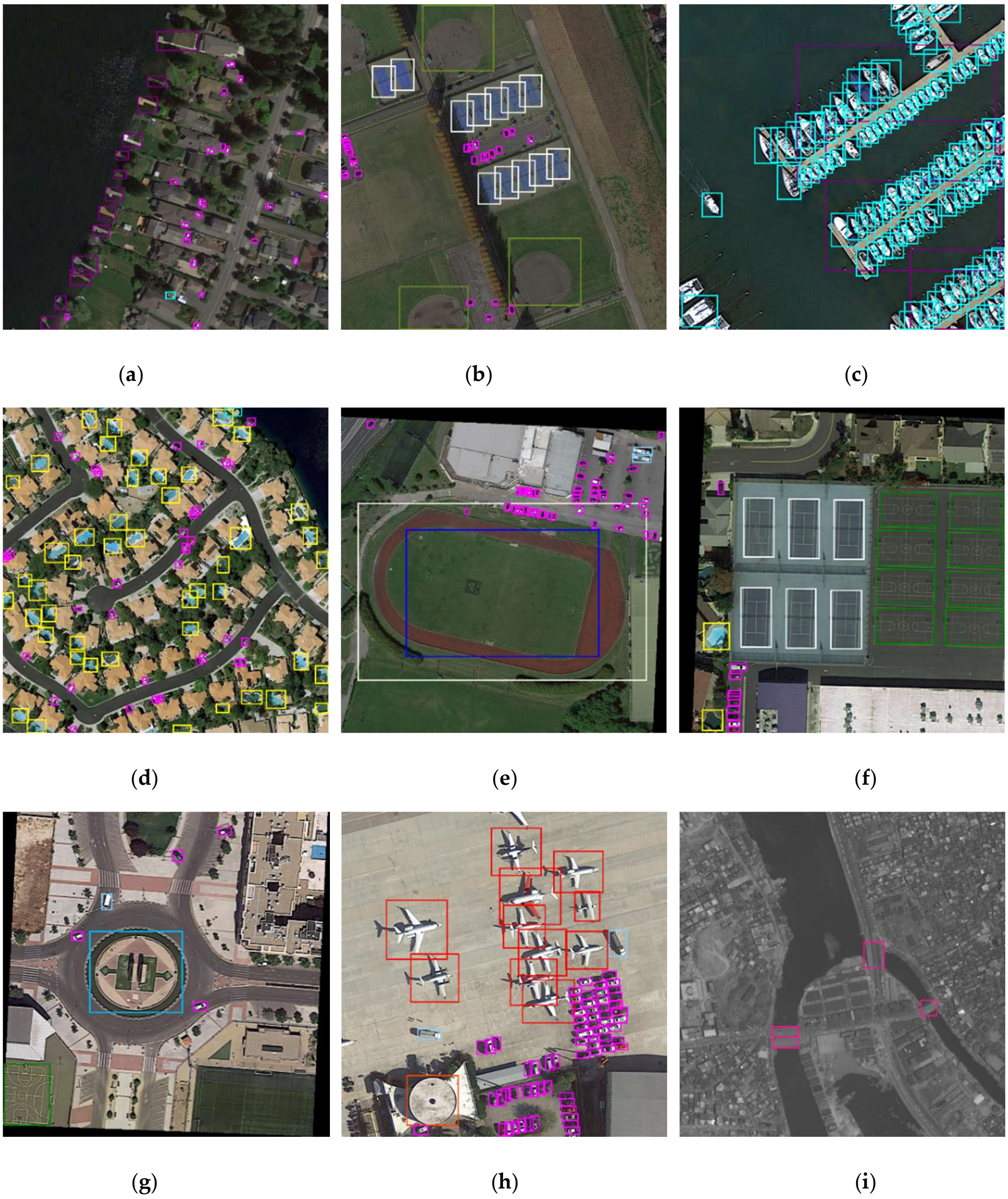
Figure 7. The precision-recall curves of Table 4: (**a**) FPN. (**b**) FPN + AS: FPN with the adaptive scale training strategy. (**c**) FPN + AA: FPN with the adaptive anchors. (**d**) EFPN + AS: EFPN with the adaptive scale training strategy. (**e**) EFPN + AA: EFPN with the adaptive anchors. (**f**) EFPN++: FPN with the adaptive scale training strategy and adaptive anchors.

AS and AA are used to shortly represent the adaptive scale training strategy and adaptive anchors

5.2. Comparison with the-State-of-the-Art Methods

5.2.1. Results on DOTA-v1.5 Dataset

To compare with the existing advanced methods, we reimplement the RetinaNet [23], Faster RCNN[2] and FPN[11] on DOTA-v1.5 Dataset. All of them are applicable to multi-category object detection. For ensuring the accuracy and fairness of experimental results, all experimental data and parameter settings are strictly consistent. Table 5 displays the comparison results which are obtained by submitting the predictions of the test set images to the official DOTA-v1.5 evaluation server. From Table 5, our method achieve the best performance, which greatly exceeds the RetinaNet, Faster R-CNN, FPN by 18.33%, 13.78%, 7.97% at mAP, respectively. For the small objects, such as small vehicles, our method remarkably outperforms the FPN by 16.49% at mAP due to its stronger ability of information extraction. The detection results of EFPN++ for each class on the DOTA-v1.5 test dataset are shown in Figure 8.



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

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

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

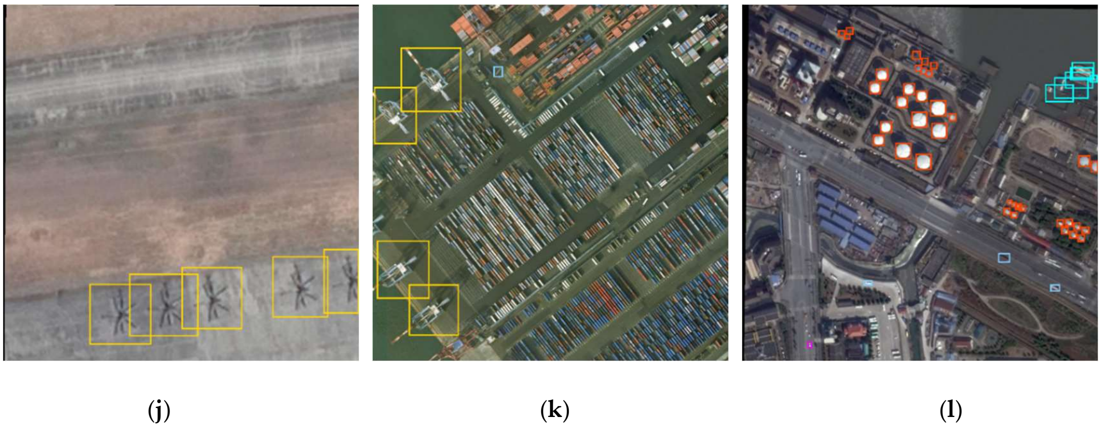


Figure 8. Visualization detection results for 16-classes on the DOTA-v1.5 test dataset. Main display categories: (**a**) harbor; (**b**) baseball diamond and tennis court; (**c**) ship and harbor; (**d**) swimming pool and small vehicle; (**e**) ground track field and soccer ball field; (**f**) tennis court and basketball court; (**g**) roundabout; (**h**) plane and lLarge vehicle; (**i**) bridge; (**j**) helicopter; (**k**) container-crane; (**l**) storage tank and ship.

5.2.2. Results on NWPU VHR-10 Dataset

We further compare the related advanced methods with the presented method on the NWPU VHR-10 dataset [20] and the comparison results as displayed in Table 6. From Table 6, it is remarkably shown that the presented method obtains better performance, which greatly exceeds Faster R-CNN [2] by 13.2% at mAP. In addition, our method outperforms the two advanced methods R-FCN [26] and Deformable R-FCN [39], increasing the mAP by 9.8% and 7.5%, respectively.

5.2.3. Results on RSOD Dataset

We also certify the effectiveness of the presented method with existing methods on the RSOD dataset [21] and Table 7 shows the comparison results. Tayara et al. [40] proposed a uniform one-stage model for object detection in aerial images and produced relatively competitive results with the mAP value of 94.19%. Overall, Table 7 shows that the proposed method is able to get better performance than the state-of-the-art methods.

Table 5. Comparison results on the DOTA-v1.5 test set. The abbreviation names of categories follow [19] and are described in detail in Section 4.1.1. The bold numbers represent the best detection results.

Method	**RetinaNet[23]**	**Faster RCNN[2]**	**FPN[11]**	**Ours**
PL	69.45	71.31	77.49	86.75
BD	75.91	75.47	79.93	84.17
BR	42.45	48.8	55.12	61.28
GTF	65.24	63.78	68.03	77.45
SV	34.36	51.52	61.17	77.66
LV	64.4	66.2	69.98	78.23
SH	69.26	78.19	84.87	88.35
TC	88.1	90.51	89.94	90.85
BC	68.48	68.76	75.93	83.67
ST	52.08	61.84	79.22	83.64
SBF	47.96	51.45	60.83	63.42
RA	63.84	69.63	71.24	76.70
HA	70.36	74.88	75.21	80.98
SP	65.41	66.83	74.51	80.99
HC	43.86	50.91	57.97	77.58
CC	25.06	28.93	30.49	47.74
mAP(%)	59.14	63.69	69.5	**77.47**

Table 6. Comparison results on the NWPU VHR-10 dataset. The abbreviation names of the category are described in detail in section 4.1.1 and the bold numbers represent the best detection results.

Method	YOLOv2[3]	SSD[22]	Faster R-CNN[2]	R-FCN[26]	Deformable R-FCN[39]	Ours
Airplane	90.16	92.3	94.7	95.9	95.9	90.7
SH	82.22	82.42	79.8	83.4	83.8	89.2
ST	20.72	52.42	55.5	65	66.8	74.5
BD	94.39	97.62	92.2	94.6	95.3	87.5
TC	44.75	60.16	57.4	69.3	73.6	89.2
BC	65.74	61.84	69.1	73.9	76.8	90.8
GTF	99.85	98.67	99.5	97.4	98.1	99.3
HA	66.45	75.68	72.9	77.5	77.9	88.2
BR	66.45	72.27	62.9	47.8	57.8	77.7
Vehicle	41.82	53.82	58	71.3	72.8	86.7
mAP(%)	67.96	74.72	74.2	77.6	79.9	**87.4**

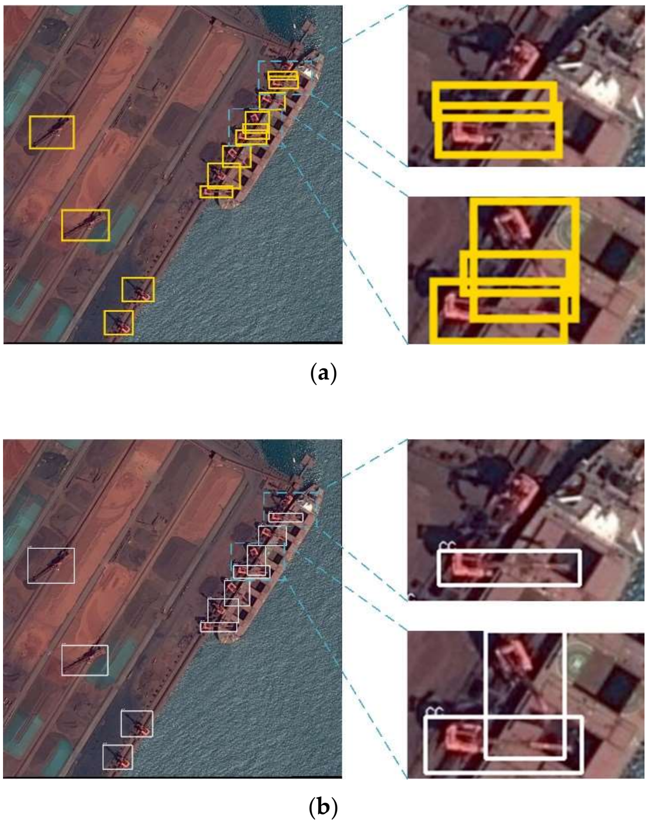
Table 7. Comparison results on the RSOD dataset. The bold numbers represent the best detection results.

Method	YOLOv2 [3]	SSD [22]	Faster R-CNN[2]	R-FCN[26]	Deformable R-FCN[39]	Tayara et al. [40]	Ours
Aircraft	64.8	72.5	76.6	84.3	84.1	86.25	96.3
Oil tank	93.77	92.83	95	95.7	96.8	95.98	96.9
Overpass	90.85	91.43	68	74.9	82.4	94.67	89.1
Playground	99.98	97.71	96	98	97.9	99.87	98.2
mAP(%)	87.35	88.62	83.9	88.2	90.3	94.19	**95.1**

6. Discussion

Extensive experimental results demonstrate that the presented method has achieved excellent detection performance in the multiple remote sensing datasets. The advantages of the proposed EFPN are illustrated as follows: (1) In the remote sensing images, numerous small-scale objects may be around or below 10 pixels, and when they are missing in the deep layers, the context cues will disappear simultaneously. Therefore, simply using the traditional feature pyramid structure can no longer improve performance in this case. Like the Feature Pyramid Network, the proposed EFPN also predicts small objects in the shallower layers. The difference is that through the proposed MBDB module and the added augmented bottom-up pathway, EFPN has better performance for detecting the small objects due to the stronger semantic information extraction capability; (2) Since large-scale objects are usually produced and predicted in deeper layers, the boundaries of these objects might be too fuzzy to obtain an accurate regression. However, the proposed EFPN can retain a high spatial resolution and have a larger receptive field in deeper layers, so it is more powerful in finding more ground-truth large objects and locating the boundary of the objects.

Although the effect is obvious, the small objects which are particularly similar to the background are poorly recognized, such as container cranes. In addition, we can see from Table 1 that the sample number of the container cranes is very small. Figure 9 compares one detection result with its ground truth for the container crane. We can well see that there are some false alarms due to the high similarity between the object and the background, and the lack of training samples. In the future work, we will consider optimizing our network with stronger feature extraction ability, and adopting a better sample balance and data amplification strategy to further promote the detection performance, especially for the small-scale objects under complex background.



(**a**)

(**b**)

Figure 9. Visual contrast: (**a**) detection result. (**b**) ground truth.

7. Conclusions

In this paper, considering the huge scale variations of the object instances in remote sensing images, we proposed an Extended Feature Pyramid Network (EFPN) which has stronger semantic information capture ability to detect multi-scale targets especially dense small targets. Through reasonable design, the proposed EFPN has fewer parameters than the original faster FPN, and can achieve better detection effect. The ablation study demonstrated the performance improvement of each component of the overall architecture. When preprocessing images and objects to be of appropriate size for training, we also proposed an adaptive scale training strategy for making the neural network better learn the features of different scale objects. In addition, due to the huge differences of the object shapes in remote sensing images, we presented a novel clustering method to obtain the adaptive scales and aspect ratios of anchors and ulteriorly improve the detection performance. Extensive experiments were performed on the open-source DOTA-v1.5 dataset, NWPU VHR-10 dataset and RSOD dataset, and the results indicate that the presented method outperformed the state-of-the-art methods on the mAP both for small objects and large objects.

Author Contributions: Conceptualization, W.G.; Methodology, W.G.; Software, W.G.; Validation, W.G. and J.K.C.; Formal Analysis, W.G. and J.K.C.; Investigation, W.G.; Writing-original draft, W.G. and J.K.C.; Writing-Review & Editing, W.W.G. and W.H.L.; Supervision, W.W.G. and W.H.L.; Project Administration, W.W.G. and W.H.L.; Funding Acquisition, W.W.G.

Funding: This work is supported by the Municipal Science and Technology Project of CQMMC, China (2017030502); National Science and Technology Key Program of China (2013GS500303); the Key Projects of Science and Technology Agency of Guangxi province, China (Guike AA 17129002).

Acknowledgments: We are very grateful for the dataset provided by the relevant scholars. We also thank the peer reviewers for valuable suggestions and comments.

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
2. 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.
3. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 7263–7271.
4. Chen, C.; Gong, W.; Chen, Y. Object Detection in Remote Sensing Images Based on a Scene-Contextual Feature Pyramid Network. *Remote Sens.* **2019**, *11*, 339.
5. Qiu, H.; Li, H.; Wu, Q. A2RMNet: Adaptively aspect ratio multi-scale network for object detection in remote sensing images. *Remote Sens.* **2019**, *11*, 1594.
6. Adelson, E.H.; Anderson, C.H.; Bergen, J.R. Pyramid methods in image processing. *RCA Eng.* **1984**, *29*, 33–41.
7. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005.
8. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
9. Huang, J.; Rathod, V.; Sun, C. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 7310–7311.
10. Liu, S.; Qi, L.; Qin, H. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
11. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. *arXiv* **2017**, arXiv:1612.03144.
12. Li, Z.; Peng, C.; Yu, G. Detnet: A backbone network for object detection. *arXiv* **2018**, arXiv:1804.06215.
13. Yu, F.; Koltun, V.; Funkhouser, T. Dilated residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 472–480.
14. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848.
15. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December *2017*; pp. 5998–6008.
16. Yan, J.; Wang, H.; Yan, M. IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sens.* **2019**, *11*, 286.
17. Uijlings, J.R.R.; Sande, K.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158.
19. Xia, G.S.; Bai, X.; Ding, J. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. *arXiv* **2017**, arXiv:1711.10398.
20. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Geosci. Remote Sens.* **2016**, *54*, 7405–7415.
21. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Geosci. Remote Sens.* **2017**, *55*, 2486–2498.
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Fu, C.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
23. Lin, T.Y.; Goyal, P.; Girshick, R. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October, 2017; pp. 2980–2988.

24. Zhang, S.; Wen, L.; Bian, X. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4203–4212.
25. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
26. Dai, J.F.; Li, Y.; He, K.M.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1606.06409.
27. Cai, Z.; Fan, Q.; Feris, R.S. A unified multi-scale deep convolutional neural network for fast object detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 354–370.
28. Zhang, Y.; Gong, W.; Sun, J. Web-Net: A Novel Nest Networks with Ultra-Hierarchical Sampling for Building Extraction from Aerial Imageries. *Remote Sens.* **2019**, *11*, 1897.
29. Feizollah, A.; Anuar, N.B.; Salleh, R. Comparative study of k-means and mini batch k-means clustering algorithms in android malware detection using network traffic analysis. In Proceedings of the 2014 International Symposium on Biometrics and Security Technologies (ISBAST), Kuala Lumpur, Malaysia, 26–27 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 193–197.
30. He, K.; Gkioxari, G.; Dollar, P. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *99*, 1, doi:10.1109/TPAMI.2018.2844175.
31. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv* **2017**, arXiv:1611.05431.
32. Wang, F.; Jiang, M.; Qian, C. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 3156–3164.
33. Woo, S.; Park, J.; Lee, J.Y. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
34. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
35. Syakur, M.; Khotimah, B.; Rochman, E.; Satoto, B. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *336*, 012017.
36. Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollar, P.; He, K. Detectron. Available online: https://github.com/facebookresearch/detectron (accessed on 22 January 2018).
37. Lin, T.Y.; Maire, M.; Belongie, S. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zürich, Switzerland, 6–12 September 2014; pp. 740–755.
38. Wu, Y.X.; He, K.M. Group Normalization. *arXiv* **2018**, arXiv:1803.08494.
39. Dai, J.; Qi, H.; Xiong, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
40. Tayara, H.; Chong, K.T. Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. *Sensors* **2018**, *18*, 3341.