remote sensing

MDPI

Article

EFM-Net: Feature Extraction and Filtration with Mask Improvement Network for Object Detection in Remote Sensing Images

Yu Wang, Yannan Jia and Lize Gu *

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; jiuge@bupt.edu.cn (Y.W.); jyn_1023@bupt.edu.cn (Y.J.)
* Correspondence: glzisc@bupt.edu.cn

Abstract: Object detection is an essential task in computer vision. Many methods have made significant progress in ordinary object detection. Due to the particularity of remote sensing images, the detection target is tiny, the background is messy, dense, and has mutual occlusion, which makes the general detection method challenging to apply to remote sensing images. For these problems, we propose a new detection framework feature extraction and filtration method with a mask improvement network (EFM-Net) to enhance object detection ability. In EFM-Net, we designed a multi-branched feature extraction (MBFE) module to better capture the information in the feature graph. In order to suppress the background interference, we designed a background filtering module based on attention mechanisms to enhance the attention of objects. Finally, we proposed a mask generate the boundary improvement method to make the network more robust to occlusion detection. We tested the DOTA v1.0, NWPU VHR-10, and UCAS-AOD datasets, and the experimental results show that our method has excellent effects.

Keywords: remote sensing; object detection; feature enhancement; attention mechanism; occlusion improvement

Citation: Wang, Y.; Jia, Y.; Gu, L. EFM-Net: Feature Extraction and Filtration with Mask Improvement Network for Object Detection in Remote Sensing Images. *Remote Sens.* **2021**, *13*, 4151. https://doi.org/10.3390/rs13204151

Academic Editor: Paolo Addesso

Received: 30 August 2021
Accepted: 13 October 2021
Published: 16 October 2021

1. Introduction

As a basic task in computer vision project, object detection has achieved good results in many mature general detection frameworks [1–6]. Some general benchmark datasets, such as COCO [7] and VOC2007 [8], have become the test standard for the general framework. However, due to the differences between remote sensing images and natural images, the general detection framework cannot be well applied to remote sensing object detection.

Aerial overhead and synthetic aperture radar (SAR) technologies are usually used to capture remote sensing images to obtain optical or hyperspectral images. Our research mainly focuses on optical images. Because of the particularity of remote sensing images, object detection based on this technology also has many difficulties.

Small object: Small object recognition is a difficult task in natural image detection, and remote sensing images contain a large number of small objects due to the long imaging distance, resulting in the unclear characteristics of these objects. At the same time, because the size of remote sensing images may be large, there may be a huge size difference between the large objects and small objects in the image. Drastic multi-scale changes also lead to the poor detection of small objects.

Background complex: There is little difference between the features of an object and the background of an image. The features of small objects are likely to be submerged by the complex background, resulting in false positives and missed detection.

Dense arrangement: Due to the uneven distribution of objects, there are dense objects in a specific area, and there may be mutual occlusion between objects.

Arbitrary direction: The results of natural object detection are as follows: horizontal bounding boxes (HBB). It is represented by four parameters (x, y, w, h), where x, y, w, h are

the center coordinates, height, and width of the bounding box. The remote sensing target can appear in the image in any direction, and the detection result is usually the oriented bounding boxes. It is represented by five parameters (x, y, w, h, θ), where θ is the angle of the bounding box.

In order to solve the problem of remote sensing object detection, the most basic solution is to train remote sensing datasets by using general objects detection frameworks, such as faster region convolutional neural networks (Faster R-CNN) [1] and RetinaNet [5]. However, the detection effect of these networks for small targets is not very good. The scheme, which is based on a multi-scale image pyramid, scales the image to different sizes for better multi-scale training, but it requires computation and memory. Lin et al. [4] proposed a feature pyramid network (FPN) for multi-scale detection, which significantly reduces cost. At the same time, the top-down structure is used to fuse shallow and deep features to improve the positioning ability of objects.

In addition to the general detection framework, the researcher proposes some network structures designed for rotating object detection. Ma et al. [9] proposed rotation region proposal networks (RRPN), an introduced rotation anchor in Faster R-CNN, to extract features. Jiang et al. [10] proposed a rotational region convolutional neural network (R2CNN) that considered other shapes of the region of interest (RoI). They used a new representation $(x1, x2, y1, y2, h)$ to represent the prediction box, where $x1, x2, y1, y2$ represents the coordinates of the long side, and h is the length of the short side. Yang et al. [11] proposed SCRDet, and the paper proposed a sampling fusion network (SF-Net), which effectively fused the feature images of different layers to detect small targets. At the same time, it proposed the multi-dimensional attention network (MDA-net) to reduce the noise of images by relying on the channel attention and spatial attention mechanism and solved the problem of discontinuous angle change in prediction. Both authors of F3-Net [12] and the extended feature pyramid network (EFPN) [13] propose a different feature extraction and filtering structure to enhance the ability of feature representation and to avoid noise interference.

Our paper proposes a new convolutional neural network called feature extraction and filtration with a mask improvement network (EFM-Net) to solve the above problem. EFM-Net takes the Faster R-CNN network as the main structure and adds the feature extraction module, the background filtering module, and the mask improvement module. The feature extraction module aims to better extract semantic information through the convolution branches of different shapes and to sample the target area more accurately. The module expands the perception area by mixing the convolution of different effects and is horizontally embedded into the FPN network. It improves the detail acquisition level of the FPN network. We construct a multi-dimensional attention network composed of coordinate attention and cross-channel attention mechanism to solve the background filtering problem. For dense arrangement and mutual occlusion between objects, we propose a mask generating and a boundary improvement method to enhance detection robustness.

We conducted experiments on public aviation remote sensing image datasets, and the experimental results show that our method has better detection accuracy than the original method. We used three datasets to verify our experimental results. DOTA [14–16] is a significant scale benchmark and provides challenges for object detection in aerial images. NWPU VHR-10 [17] is a 10-level remote sensing dataset for space object detection. UCAS AOD [18] is a dataset that only contains two kinds of targets: car and airplane.

Our main contributions are as follows:

1. We propose a new object detection framework for remote sensing image detection called EFM-Net.
2. In EFM-Net, we design a multi-branched feature extraction (MBFE) module embedded in an FPN to improve the feature capture ability. At the same time, in order to reduce the complex background information in remote sensing images, we design a background filtering module based on an attention mechanism to reduce the background interference.

3. We design an RoI region enhancement method with occlusion and key point enhancement to improve the occlusion detection accuracy.
4. This method achieved good results with the DOTA v1.0, NWPU VHR-10, UCAS AOD datasets.

2. Related Works

2.1. Multi-Scale Object Detectors

It is always challenging to detect targets of different sizes, especially the features of small targets, which are easily lost in the deep feature map. In recent years, CNN has become the most effective object detection algorithm. There are two kinds of object detection that are based on CNN: one-stage and two-stage: the one-stage algorithm predicts the result directly, and the two-stage algorithm will predict the candidate box first and will then predict the result of the candidate box. The one-stage algorithm includes the you only look once (YOLO) series [2,19,20], the single shot multi-box detector (SsD) [6], and RetinaNet [5], and the two-stage algorithm includes R-CNN [21], Fast R-CNN [22], and Faster R-CNN [1]. The FPN [4] relies on multi-scale prediction and the fusion of the top-down feature map and the original bottom-up feature map to solve small target feature loss. There are many papers on the improvement of the FPN network. PANet [23] increases the path of bottom-up secondary integration. The Bi-FPN [24] proposed by Google obtains the irregular optimal topology through a large number of calculations and searches. AugFPN [25] solves the problems of similar images not being able to be used in different scales and of the loss of M5 information when the ASFF [26] module is added in different FPN structures.

Google proposed the inception module [27–30], the inception of the V1-proposed 1×1 convolution, compressed the number of channels, improved the computational efficiency, stacked convolution cores of different sizes, and improved the network depth. Inception V2 and V3 decompose the large-scale convolution kernel to reduce computation and to introduce the batch normalization (BN) layer [28]. Inception V4 introduces the residual neural network (ResNet) [31] into nodes, which improves network speed.

2.2. Deformable Convolutional Networks

The deformable convolutional kernel is a module that can be inserted into any neural network. It changes the fixed shape of the original convolution kernel. By learning the direction of the convolution kernel through the network, the convolution kernel can better extract the characteristics of different shapes of objects. The FPN relies on the up sampling of the deep network and the addition of adjacent shallow networks to enhance the feature map. However, after the up sampling and 1×1 convolution, the internal data structure and spatial level information may be lost, so it cannot ultimately obtain the target information. In order to solve this problem, researchers provide many effective structures. The dilated convolution [32] fills 0 in the convolution core. At the same time, equal distance sampling is conducted to expand the receptive field without increasing the amount of calculation. Deformable convolutional networks (DCN)v1 [33] achieve a more practical effect than dilated convolution does. By learning the offset of the convolution kernel in the X and Y directions, it can convolute the feature map with the convolution check of different shapes, which effectively improves the extraction of semantic information. DCN v2 [34] improves DCN v1 by adding a penalty parameter m, which means that when the region extracted by the convolution kernel exceeds the region of the target, the offset parameter can be punished. We rely on DCN v2 to extract the features of different scales of the FPN network in the multi-branched feature extraction structure to reduce the loss of FPN network features.

2.3. Attention Mechanism

The attention mechanism was first used in natural language processing, and later, there were some methods to apply attention mechanism to images. Its principle is to suppress

the unimportant part, highlight the critical part, namely weakening the background, highlighting the object, and making the neural network pay more attention to the objects. A squeeze and excitation network (SENet) [35] relies on learning the importance of different feature channels and then uses this importance to improve the significance of objects in the feature map. Efficient channel attention networks (ECANet) [36] replace the full connection layer in the sensor with the convolution layer, which avoids the channel being compressed and depends on adjacent correlation rather than global correlation. The convolutional block attention module (CBAM) [37] combines channel attention with spatial attention, and it has a better effect than the single-channel attention mechanism. Coordinate attention [38] decomposes attention into one-dimensional vectors along two directions to construct an attention mechanism. BorderDet [39] through experiments, it is found that there may be redundancy in extracting features from all object regions. Therefore, it can reduce the number of calculations that are necessary and can improve the accuracy by extracting the features of the boundary and the center of the object and by enhancing the features.

2.4. Occlusion Improvement

Object detection builds a network model by learning the image, which requires the network to have good robustness to deformation, occlusion, and other problems to be able to accurately detect objects in all situations. One method to solve this problem is to increase the distribution of occluded images by sampling them. The other method is to generate all possible occlusions and deformations and to then train the neural network. Wang et al. [40] proposed adversary Fast-RCNN, a network that introduces adversary learning into object detection and that generates occluded and deformed objects through the adversary network. The network includes an adversarial spatial transformer network (ASTN) structure and an adversarial spatial dropout network (ASDN) structure. The distributed network deformation learning and occlusion parameters make the network more robust.

3. Proposed Method

3.1. Overall Architecture

Figure 1 is the complete network structure of EFM-Net, which is based on Faster R-CNN [1] and FPN [4], and different aspects have been improved. EFM-Net is mainly composed of four parts:

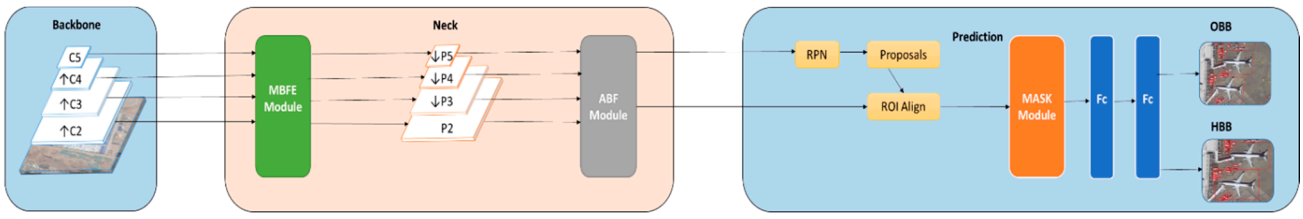

Figure 1. The framework of the proposed EFM-Net. EFM-Net consists of four main components: the backbone uses the ResNet and MBFE module to extract features, a background filtering module for background filtering, a mask improvement module for enhanced features, a detection module for object classification, and regression.

The backbone network uses ResNet [31]. In the horizontal connection of each layer of ResNet, we designed and added an MBFE module to obtain more semantic information through different branches and to reduce the loss caused by channels.

Then, we designed an attention module called the background filtering module in the horizontal connection to weaken the background information and to enhance the network's attention to objects.

After the RoI layer, we designed a mask improvement network and enhanced it by using the maximum value of the boundary as the eigenvalue so that the network can detect occluded objects more effectively.

Finally, the prediction branch was used to predict the boundary frame and object type. Because remote sensing object detection needs horizontal bounding boxes and oriented bounding boxes, the prediction branch was divided into two branches.

The specific process is as follows: ResNet generates C2, C3, C4, C5 feature maps of different sizes, extracts features through the MBFE module, and adds them to the feature maps of the same size obtained by up sampling. Then, ResNet generates feature maps of different sizes through the background filtering module based on the attention mechanism. Then, the region is extracted by the RPN network to obtain the region proposals. The extracted region is partially occluded and enhanced by the mask improvement network. Finally, the network can obtain the predicted results.

3.2. MBFE Module

We used the residual network as the backbone and the FPN as the neck to construct the neural network. Through convolution stride and the pooling of the residual network with level deepening, the spatial resolution of the feature map gradually decreases, while the number of channels of the feature map gradually increases. When the residual network transmits the feature map to the FPN, the spatial resolution of the feature map gradually decreases. Only a simple 1×1 convolution is used to reduce the channel of the feature map, which means that in order to reduce the parameters, the feature map may lose part of the feature information of the object. The size of the remote sensing target is generally small, so it contains insufficient features. If information only travels through a simple channel reduction, then the phenomenon of information loss is more serious. Therefore, it is very important to increase the method of context information acquisition. We improved the horizontal connection mode of the residual network and FPN and proposed an MBFE module to better capture semantic information. The details of the MBFE module are shown in Figure 2.

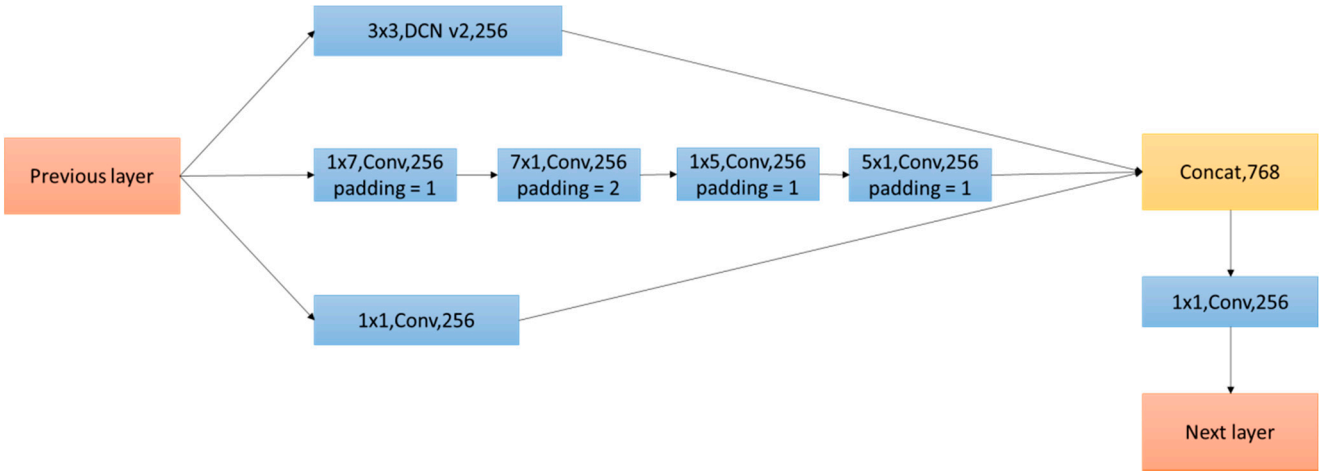

Figure 2. Multi-branched feature extraction (MBFE) module.

The output of MBFE is extracted by three branches. The first branch contains the parameters of DCN v2 [34],

$$y(p_0) = \sum_{p_n \in R} w(p_n) \times x(p_0 + p_n + \Delta p_n) * \Delta m_n \quad (1)$$

$$R = \{(-1,-1),(-1,-0),\cdots,(0,1),(1,1)\} \quad (2)$$

where Δp_n represents the diffusion position learned by each convolution grid, Δm_n represents the penalty coefficient learned by each convolution grid, and the convolution kernel variable is obtained by calculating the conventional convolution kernel with these parameters.

The second branch contains asymmetric convolution kernels of different sizes. Through these convolution kernels of different sizes, we increased the depth of the network and the ability to extract information from objects of different shapes. The large convolution kernels enhance the sensing area of each feature point, and the sensing field of each element is larger in order to achieve a better global information capture ability. At the same time, compared to the ordinary convolution kernel, the asymmetric convolution kernel can reduce the number of parameters without affecting the detection effect, and the third branch is a 1×1 convolution. We concatenate the three branches and then use a 1×1 convolution to reduce the dimension to produce a 256-dimensional feature map. Through experiments, we were able to verify different branches and different convolution sizes. We introduced the MBFE branch into the horizontal connection of each layer of the FPN and obtained better semantic information capture ability without introducing too many parameters.

3.3. Background Filtering Module

Because the remote sensing objects in remote sensing images are small and lack features, they are easily submerged by complex environments. At the same time, the arrangement between objects is very dense. Excessive noise will confuse the boundaries of different targets, causing false reports and false detection. In order to eliminate redundant information, we introduced a background filtering module into the neural network to weaken the background and the noise and to enhance the characteristics of the object. The specific structure of the background filtering module is shown in Figure 3.

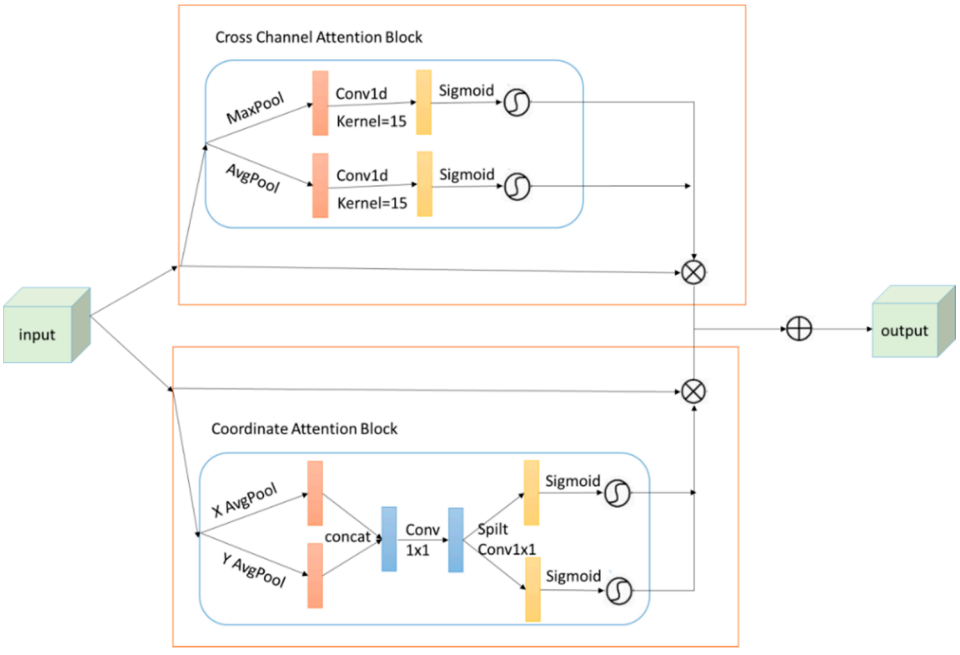

Figure 3. Background filtering module.

The background filtering module is composed of a cross-channel attention block (CCAB) and a coordinate attention block (CAB). The weights obtained by the two attention blocks are multiplied by the original feature graph. After adding the two feature graphs, a denoised network can be obtained. The complete calculation details of the background filtering module are as follows:

$$F_{out} = W_{cc}(F_{in}) + W_{ca}(F_{in}) \tag{3}$$

where F_{in} represents the input feature map obtained by the FPN network, F_{out} represents the result feature map obtained by the background filtering module, W_{cc} represents the weight obtained by CCAB, and W_{ca} represents the weight obtained by CAB. In CCAB, we

first used maximum pooling and average pooling to compress the input feature graph and then generated F_{cc}^{avg} and $F_{cc}^{\max}$. The calculation of F_{cc}^{avg} and $F_{cc}^{\max}$ is as follows:

$$F_{cc}^{avg} = \frac{1}{H \times W} \sum_{i=1}^{1} \sum_{j=1}^{1} x_{ij}^{k} \ (0 < l < C) \tag{4}$$

$$F_{cc}^{\max} = \max\left\{ x_{ij}^{k} \middle| 0 < i < H, 0 < j < W \right\} \ (0 < k < C) \tag{5}$$

where H, W, C represent the width, height, and channel number of the feature graph, and x^k represents the k-th channel. After transposing F_{cc}^{avg} and $F_{cc}^{\max}$, we used a one-dimensional convolution to realize the interaction between the adjacent channels, of the reduction of the SENet parameters [35] through the full connection layer to be avoided, and then the weight W_{cc} was generated by using a sigmoid function. The calculation of W_{cc} is as follows:

$$W_{cc} = \sigma_s(\sigma_R(\sigma_{BN}(f^{1\times 15}(F_{cc}^{\max})))) \otimes \sigma_s(\sigma_R(\sigma_{BN}(f^{1\times 15}(F_{cc}^{avg})))) \tag{6}$$

where σ_s is the sigmoid function, σ_{BN} represents the BN [28] layer, σ_R represents the ReLU function, $f^{1\times 15}$ represents the one-dimensional convolution kernel with a convolution size of 1×15, and finally, W_{cc} is obtained. The complete calculation process of CAB is as follows: First, we compressed the input characteristic graph into and F_{ca}^{h} by average pooling; the size of F_{ca}^{w} is $C \times 1 \times W$, and the size of F_{ca}^{wh} is $C \times H \times 1$. They can be calculated as follows:

$$F_{ca}^{h} = \frac{1}{W} \sum_{i=1}^{H} h_i^k \ (0 < k < C) \tag{7}$$

$$F_{ca}^{w} = \frac{1}{H} \sum_{j=1}^{\mathrm{W}} w_j^k \ (0 < k < C) \tag{8}$$

where h_i^k represents the i-th row of the k-th channel, and w_j^k represents the j-th column of the k-th channel. Then, we transposed F_{ca}^{h} and F_{ca}^{w} and concatenated them. After the 1×1 convolution, we reduced the number of channels to 32 and then activated them through ReLU to obtain F_{ca}^{wh}.

$$F_{ca}^{wh} = \sigma_R(\sigma_{BN}(f^{1\times 1}(F_{ca}^{h}; F_{ca}^{w}))) \tag{9}$$

W_{ca} is obtained by splitting F_{ca}^{wh} and the sigmoid function.

$$W_{ca} = \sigma_s(F_{ca}^{w\prime}) \otimes \sigma_s(F_{ca}^{h\prime}) \tag{10}$$

We can achieve the class activation mapping in Figure 4 using the Grad-CAM [41]. After adding the background filtering module, the ability of the network to focus on the target area is better than FPN.

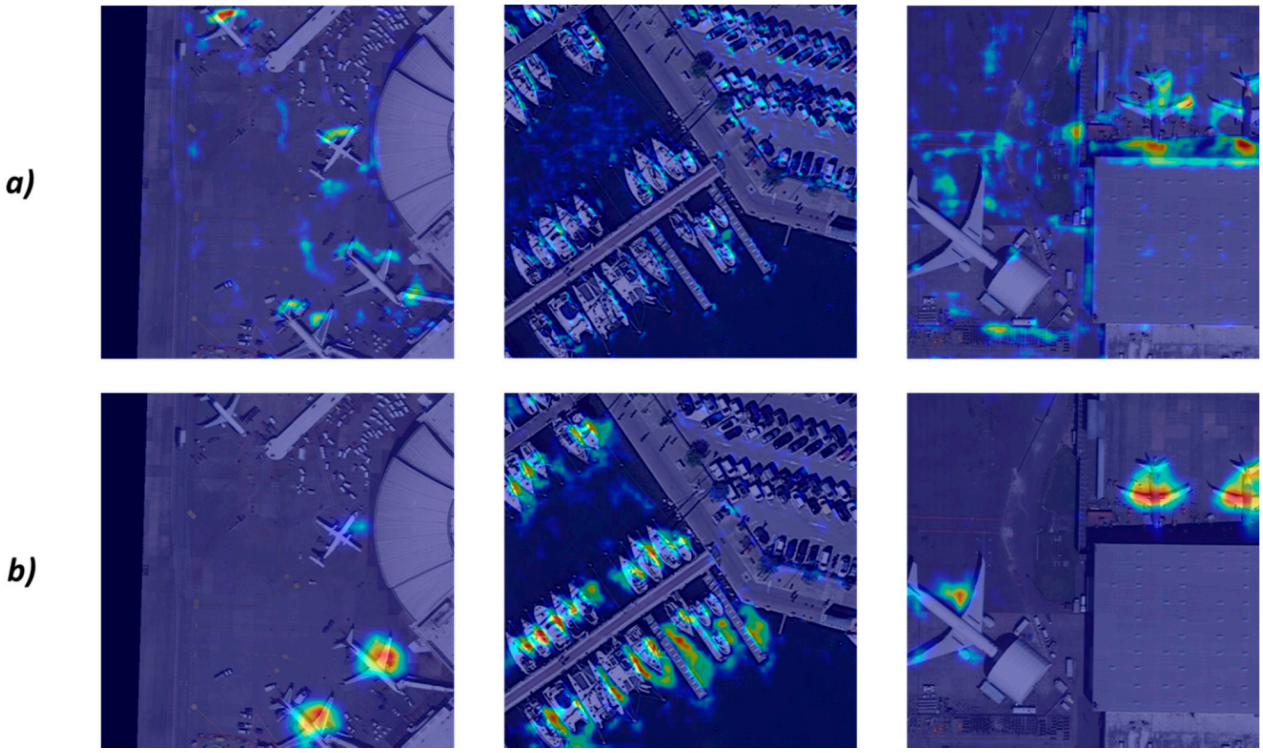

Figure 4. Comparison of class activation mapping. (**a**) Input class activation mapping. (**b**) Class activation mapping after background filtering module.

Figure 5 shows that the complex background affects the detection results and that the background filtering module improves this problem.

3.4. Mask Improvement Module

Because the remote sensing target may be occluded and small, the features are not rich, and they are easily blurred by the complex background. In order to solve the above problems, we generated the occluded network by way of anti-sampling to improve the robustness of the network to object detection. At the same time, to avoid the necessary features being blurred, we also enhanced the features through the boundary of the target. Figure 6 shows the effect that we envisioned.

We proposed a mask generating module and a boundary improvement module to enhance the training effect. Its function was to generate mask feature maps for some regions and to transform important boundary information into enhanced information. First, the RoI align layer divided the object feature map into 7×7 regions, and then the ROI region performed boundary enhancement and feature masking through two branches.

The mask improvement module was used to generate the occlusion matrix. A total of seven occlusion positions were obtained through convolution and global average pooling The seven regions were set to 0, and other regions remained unchanged. The generated mask matrix was multiplied by the original feature image.

Figure 5. Comparison of detection result. (**a**) Faster R-CNN; (**b**) Faster R-CNN + background filtering module.

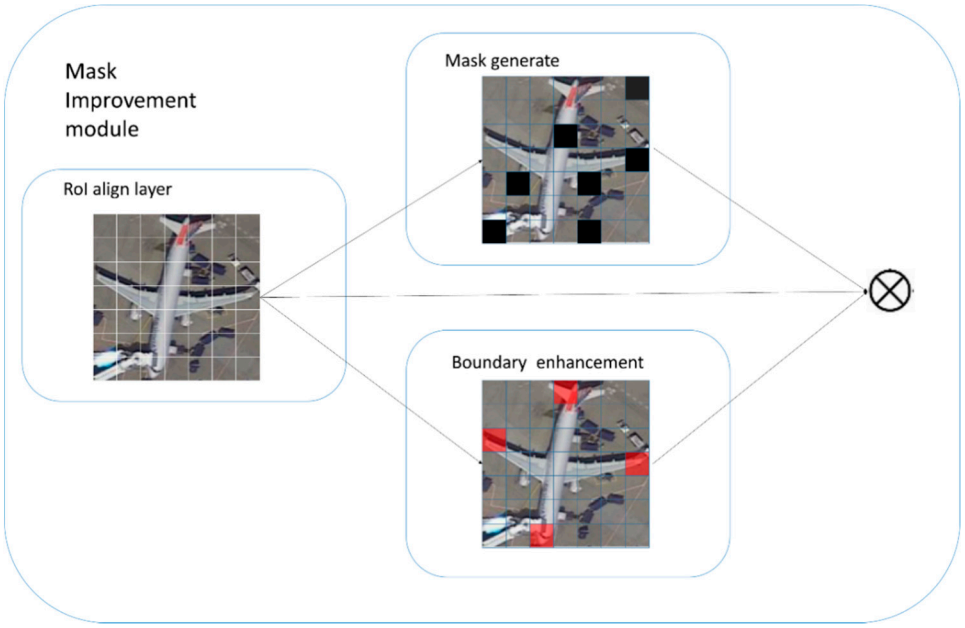

Figure 6. Mask improvement module takes image patches with the features extracted as input using the ROI align layer then blocks the position generated by the mask and enhances the image using boundary features.

The structure of the boundary enhancement module is shown in Figure 7, and the calculation process is as follows:

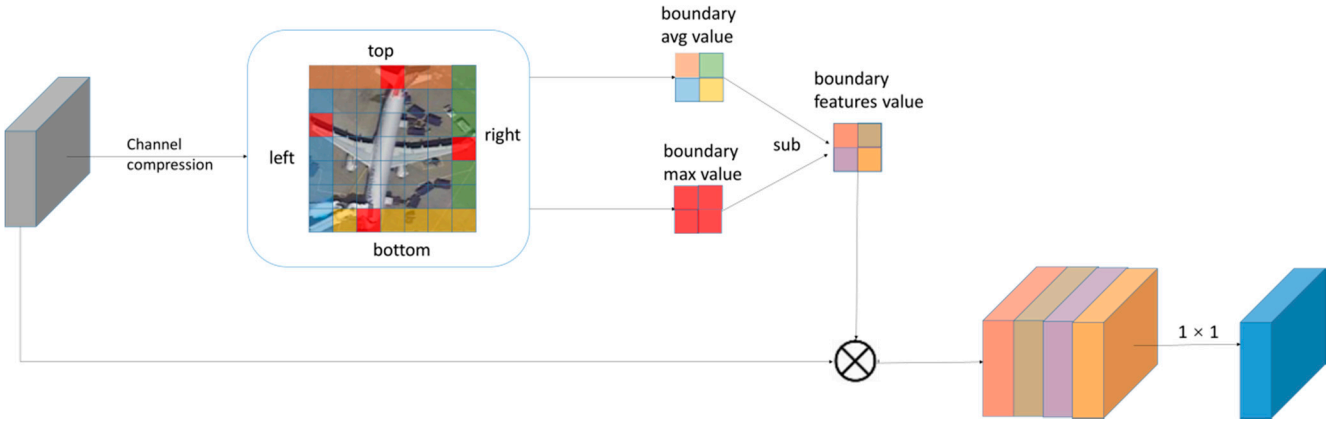

Figure 7. Boundary enhancement module.

Sum the values of all channels to make the number of channels become 1.

Divide the boundary into four regions: top, bottom, left, and right to obtain the maximum pooling value and the average pooling value of each region.

The average pooled value is subtracted from the maximum pooled value and uses the sigmoid function in the result.

The four boundary eigenvalues are multiplied by the original feature map and concat them.

1×1 convolution is performed to restore the original number of channels.

3.5. Loss Function

The loss function is similar to the Faster R-CNN [1], but our prediction includes detectors of HBB and OBB and HBB and OBB using the multitasking loss function.

$$L = \frac{\lambda_1}{N_{reg}} \sum_n p_n^* L_{reg}(t_n, t_n^*) + \frac{\lambda_2}{N_{cls}} \sum_n L_{cls}(p_n, p_n^*) \tag{11}$$

where L represents the loss function of HBB and OBB; N_{reg} represents the number of selected anchors set to 2000; N_{cls} represents the batch size set to 512; n represents the index of the bounding box; p_n represents the confidence of object prediction; p_n^* is a binary value when the anchor is positive; p_n^* is 1, otherwise it is 0; t_n represents the coordinate vector of object prediction; t_n^* represents the coordinate vector of the ground truth box. L_{cls} is the cross entropy function, and L_{reg} is the smoothL1 function. The regression box is defined as follows:

$$\begin{aligned}
&t_x = (x - x_a)/w_a, t_y = (y - y_a)/h_a, \\
&t_w = \log(w/w_a), t_h = \log(h/h_a), \\
&t_\theta = \theta - \theta_a, \\
&t'_x = (x' - x_a)/w_a, t'_y = (y' - y_a)/h_a, \\
&t'_w = \log(w'/w_a), t'_h = \log(h'/h_a), \\
&t'_\theta = \theta' - \theta_a,
\end{aligned} \tag{12}$$

where x, y, w, h represent the center coordinates, height, and width of the object; θ represents the angle of the object; x and x' represents the prediction box of the HBB and OBB; and x_a represents the anchor box (y, w, h are the same).

4. Experiments

4.1. Datasets and Evaluation Criteria

Our network was tested on the following datasets:

DOTA: DOTA v1.0 [14] contains 2806 aerial images that range in size from 800 × 800 to 4000 × 4000. There are 188,282 instances in 15 categories. The 15 categories are 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), and helicopter (HC). The DOTA v1.0 dataset is OBB, and the labeling method uses the vertex coordinate labeling method. First, the head of the object is selected, and then it is labeled point by point in the clockwise direction. Finally, eight coordinates of four vertices are obtained. The proportion of the training set, validation set, and test set is 1/2, 1/6, and 1/3, and the test set is not disclosed.

NWPU VHR-10: The NWPU VHR-10 [17] contains more than 600 labeled aerial images that are about 1000 × 1000 in size, which are divided into 10 categories, including aircraft, baseball field, basketball court, bridge, port, ground track and field, ship, storage tank, tennis court, and vehicle. The NWPU VHR-10 dataset is HBB.

UCAS-AOD: The UCAS-AOD [18] contains 1510 aerial images that are about 1000 × 1000 in size and only include two kinds of objects: car and airplane. The UCAS-AOD dataset is OBB.

We divided all of the data sets except for the DOTA dataset into 60%, 20%, and 20% according to training set, verification set, and test set, respectively. We used the mAP and the P–R curve to evaluate the accuracy of all of the methods. The P–R curve can be defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{13}$$

$$recall = \frac{TP}{TP + FN} \tag{14}$$

where precision represents the detection accuracy; recall represents the detection completion rate; and TP, FP, and FN represent the number of true positions, false positions, and false negatives. The curve with recall is labeled as abscissa, precision as the ordinate is the P–R curve, and AP is the area under the P–R curve. The mAP is the average AP of each object.

4.2. Training Data and Settings

We cut the DOTA v1.0 dataset into a square image with 800 × 800 pixels with an overlap of 200 pixels. Additionally, we then completed the ablation experiment on this dataset. For the last portion of the ablation experiment and for other datasets, we used a data enhancement method, and set the data to 1000 × 1000 resolution images, set the overlapping pixels to 500, set the probability to 0.5 random flips, and rotated the images of the training set by 90 degrees, 180 degrees, and 270 degrees to increase the diversity and richness of the datasets.

Our code was built with mm detection [42] and was tested on a server with a NVIDIA Geforce GTX 1080ti. The optimizer optimizes with SGD with a weight attenuation of 0.0001 and a momentum of 0.9. In each mini-batch, only one picture is set. The network learned 94 K iterations on the datasets, and the learning rate was 0.01. In the first 500 iterations, the learning rate increased from 0.004 to 0.01 due to preheating and decreased to 0.001 and 0.0001 when the iterations reached 62 k and 86 K, respectively. We demonstrated the effectiveness of our method through ablation experiments and compared it with other state-of-the-art methods through comparative experiments.

4.3. Ablation Experiment

In order to compare the effectiveness of each part of EFM-Net, we completed experiments on the DOTA v1.0 dataset. We chose Faster R-CNN [1,16] as the baseline and added the MBFE module, background filtering module, and mask improvement module to it to compare the performance changes. We also used the data enhancement method to

expand the data and to replace the backbones to improve the feature extraction ability of the network. The experimental result is shown in Table 1.

Table 1. Ablation experiment of the components on the DOTA v1.0 dataset.

Method	Backbone Network	MBFE	Background Filtering	Mask Improvement	Data Augmentation	mAP (%) OBB	mAP (%) HBB	Execution Times (Tasks/s)
Faster R-CNN [1,16]	ResNet-50	-	-	-	-	70.06	71.09	7.7
our	ResNet-50	√	-	-	-	71.77	72.37	6.8
our	ResNet-50	-	√	-	-	71.92	72.63	7.7
our	ResNet-50	-	-	√	-	71.89	72.64	6.4
our	ResNet-50	√	√	√	-	74.32	74.43	5.1
our	ResNet-101	√	√	√	-	75.48	76.27	4.8
our	ResNet-101	√	√	√	√	76.22	77.30	4.8

Our baseline was a Faster R-CNN using ResNet-50 as the backbone network, and FPN was also used as the neck. Besides the common HBB, the prediction head added OBB as an extension. On our machine, the detection result of baseline was 71.09%, and OBB was 70.06%. As shown in Table 1, the MBFE module reached 71.77% on the OBB task and 72.37% on the HBB task, leading to a gain of 1.71% and 1.28%. The background filtering module reached 71.92% on the OBB task and 72.63% on the HBB task, leading to a gain of 1.86% and 1.54%. The mask improvement module reached 71.89% on the OBB task and 72.64% on the HBB task, leading to a gain of 1.83% and 1.55%. When three improved modules were added, the HBB and OBB of EFM net obtained 74.32% and 74.43% detection accuracy. Compared to the original baseline, there were 4.26% and 3.34% rates of improvement, respectively. The experimental results show that our improvements have a good effect on improving network detection accuracy.

The backbone also plays an essential role in the network structure. Deeper networks can better extract feature information. In order to verify the generalization ability of our network, we replaced the backbones with ResNet101. Table 1 shows that the deeper network improved our detection performance. The deeper network reached 75.48% on the OBB task and 76.27% on the HBB task. In addition to using the original DOTA v1.0 dataset, we also extended the data content. After using the extended training set and test set, our accuracy rates reached 76.22% and 77.30%. The comparison of partial test results of EFM-Net and Faster R-CNN is shown in Figure 8.

4.4. Results of DOTA Dataset

We compared EFM-Net with other network structures, including some one-stage networks and two-stage networks, such as SsD [6], Yolov2 [20], and SCRDet [11]. Tables 2 and 3, respectively, list the HBB and OBB detection results of EFM-Net and other networks on the DOTAv1.0 dataset. The short names for the categories are PL-Plane, BD-Baseball diamond, BR-Bridge, GTF-Ground track field, SV-Small vehicle, LV-Large vehicle, SH-Ship, TC-Tennis court, BC-Basketball court, ST-Storage tank, SBF-Soccer-ball field, RA-Roundabout, HA-Harbor, SP-Swimming pool, and HC-Helicopter. In the OBB task, our EFM-Net had a better detection accuracy than the other network structures, achieving an accuracy of 76.22%. Compared with the other methods, we made improvements in many categories: for BD, the accuracy was 85.64% versus 83.62%; for TC, the accuracy was 90.88% versus 90.85%; for BC, the accuracy was 87.97% versus 87.94%; for RA, the accuracy was 67.39% versus 66.68%; and for HC, the accuracy was 66.25% versus 65.55%.

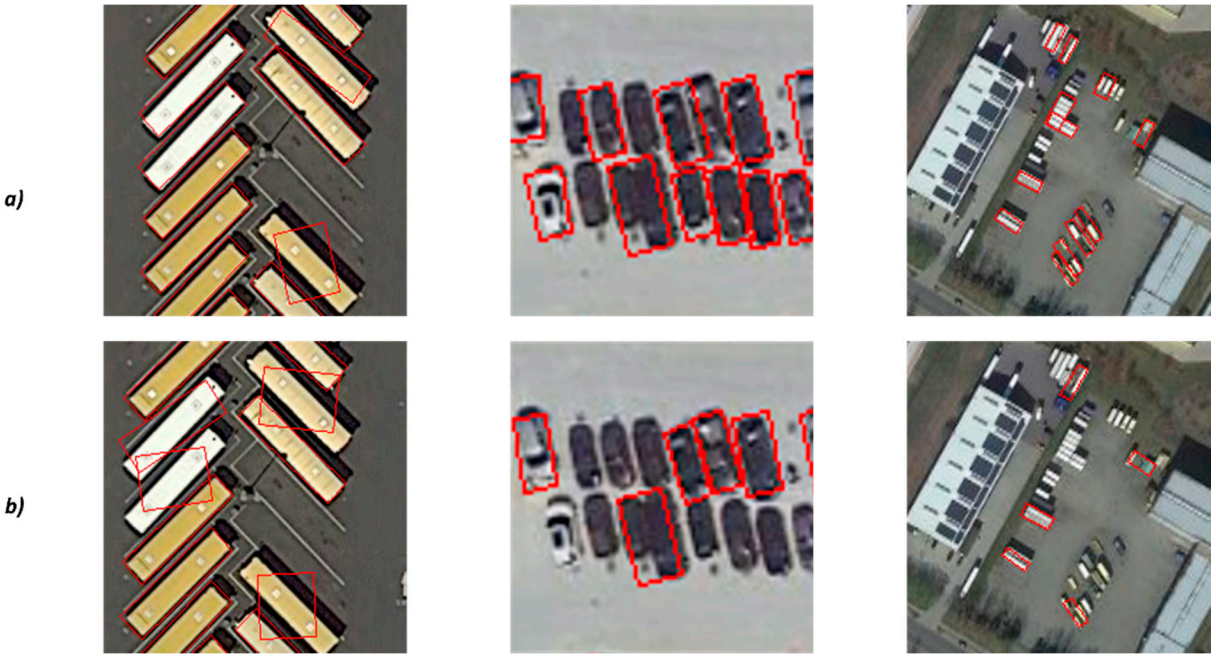

Figure 8. Comparison of partial test results of EFM-Net and Faster R-CNN on the OBB task. (**a**): EFM-Net and (**b**): Faster R-CNN.

Table 2. Performance evaluation of the OBB task on the DOTA dataset.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP (%)
One-stage methods																
SSD [6]	39.83	9.09	0.64	13.18	0.26	0.39	1.11	16.24	27.57	9.23	27.16	9.09	3.03	1.05	1.01	10.59
Yolov2 [20]	39.57	20.29	36.58	23.42	8.85	2.09	4.82	44.34	38.25	34.65	16.02	37.62	47.23	25.19	7.45	21.39
R4Det [43]	89.49	81.17	50.53	66.10	70.92	78.66	78.21	90.81	85.26	84.23	61.81	63.77	68.16	69.83	67.17	73.74
Two-stage methods																
FR-O [1,16]	84.76	77.46	47.17	63.49	75.29	74.69	85.71	90.52	81.04	79.83	48.03	61.68	62.76	63.79	54.70	70.06
ICN [44]	81.36	74.30	47.70	70.32	64.89	67.82	69.98	90.76	79.06	78.20	53.64	62.90	67.02	64.17	50.23	68.16
SCRDet [11]	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	87.94	86.86	65.02	66.68	66.25	68.24	65.21	72.61
APE [45]	89.96	83.62	53.42	76.03	74.01	77.16	79.45	90.83	87.15	84.51	67.72	60.33	74.61	71.84	65.55	75.75
F3-Net [12]	88.89	78.48	54.62	74.43	72.80	77.52	87.54	90.78	87.64	85.63	63.80	64.53	78.06	72.36	63.19	76.02
EFM-Net(our)	89.53	85.64	51.82	75.11	75.26	75.46	86.36	90.88	87.97	82.62	64.89	67.39	75.47	68.73	66.25	76.22

In the HBB task, our EFM-Net had better detection accuracy than the other network structures, achieving an accuracy rate of 77.30%. Compared to the other methods, we made improvements in many aspects: for BD, the accuracy was 86.34% versus 82.52%; for BR, the accuracy was 57.71% versus 56.20%; for SV, the accuracy rate was 80.18% versus 78.57%; for BC, the accuracy rate was 87.94% versus 87.64%; for ST, the accuracy rate was 87.38% versus 86.39%; for SBF, the accuracy rate was 65.00% versus 64.53%; and for RA, the accuracy rate was 69.34% versus 63.93%.

EFM-Net has tremendous advantages in terms of tiny targets because we rely on the extraction of context information and the filtering of the background information to make small-target localization more accurate. Our network can deal with the occlusion problem where HBB is more accurately likely to exist. The examples of detection results for both the OBB and the HBB tasks on the DOTA test set is shown in Figure 9.

Table 3. Performance evaluation of the HBB task on the DOTA dataset.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP (%)
One-stage methods																
SSD [6]	57.85	32.79	16.14	18.67	0.05	36.93	24.74	81.16	25.10	47.47	11.22	31.53	14.12	9.09	14.12	29.86
Yolov2 [20]	76.90	33.87	22.73	34.88	38.73	32.02	52.37	61.65	48.54	33.91	29.27	36.83	36.44	38.26	11.61	39.20
Two-stage methods																
FR-H [1,16]	88.95	82.52	51.06	62.01	78.57	71.37	85.14	88.99	82.99	84.72	41.19	62.39	72.69	70.30	43.43	71.09
ICN [44]	89.97	77.71	53.38	73.26	73.46	65.02	78.22	90.79	79.05	84.81	57.20	62.11	73.45	70.22	58.08	72.45
SCRDet [11]	90.18	81.88	55.30	73.29	72.09	77.65	78.06	90.91	82.44	86.39	64.53	63.45	75.77	78.21	60.11	75.35
F3-Net [12]	88.91	78.50	56.20	74.43	73.00	77.53	87.72	90.78	87.64	85.71	64.27	63.93	78.70	74.00	65.85	76.48
EFM-Net (our)	89.97	86.34	57.71	73.52	80.78	71.70	86.40	90.88	87.94	87.38	65.00	69.34	77.77	72.87	61.86	77.30

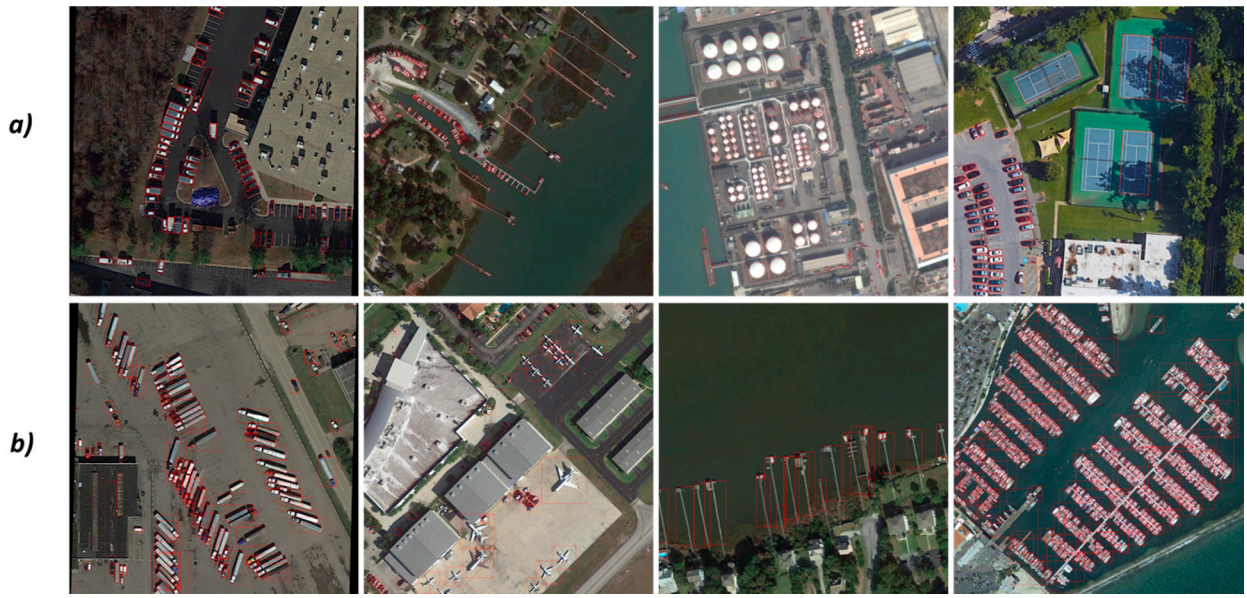

Figure 9. Examples of our detection results for both the OBB and the HBB tasks on the DOTA test set. (**a**): OBB task and (**b**): HBB task.

4.5. Results of NWPU VHR-10 and UCAS AOD Datasets

We also verified the EFM-Net detection results on other datasets. Table 4 shows the detection results of our network on the NWPU VHR-10 dataset [17]. The short names for the categories are PL-Plane, SH-Ship, ST-Storage tank, BD-Baseball diamond, TC-Tennis court, GTF-Ground track field, HB-Harbor, BR-bridge, and VE-vehicle. The detection accuracy reached 92.10%, and the accuracy of PL, TC, GTF was over 99.5%. Our feature extraction and filtering module solved severe background noise in an area, and the environment easily submerged object features. The Figure 10 shows the P-R curves of the NWPU VHR-10 dataset.

Table 4. Performance evaluation of the HBB task on the NWPU VHR-10 dataset.

Method	PL	SH	ST	BD	TC	BC	GTF	HB	BR	VE	mAP (%)
FMSSD [46]	99.70	90.80	90.60	92.90	90.30	80.10	90.80	80.30	68.50	87.10	87.10
F3-Net [12]	99.31	92.62	92.89	97.14	91.38	86.16	98.00	90.30	82.18	88.90	91.89
EFM-Net(our)	99.70	86.70	92.50	97.40	99.70	96.30	99.60	76.10	82.40	90.20	92.10

Table 5 shows the detection results of EFM-Net on the UCAS-AOD dataset, and the mAP reached 96.60% and 97.50%, respectively, on the OBB and HBB tasks. The experiment

shows that our method is also better than other methods when analyzing this dataset. The Figure 11 shows the P-R curves of the UCAS-AOD dataset.

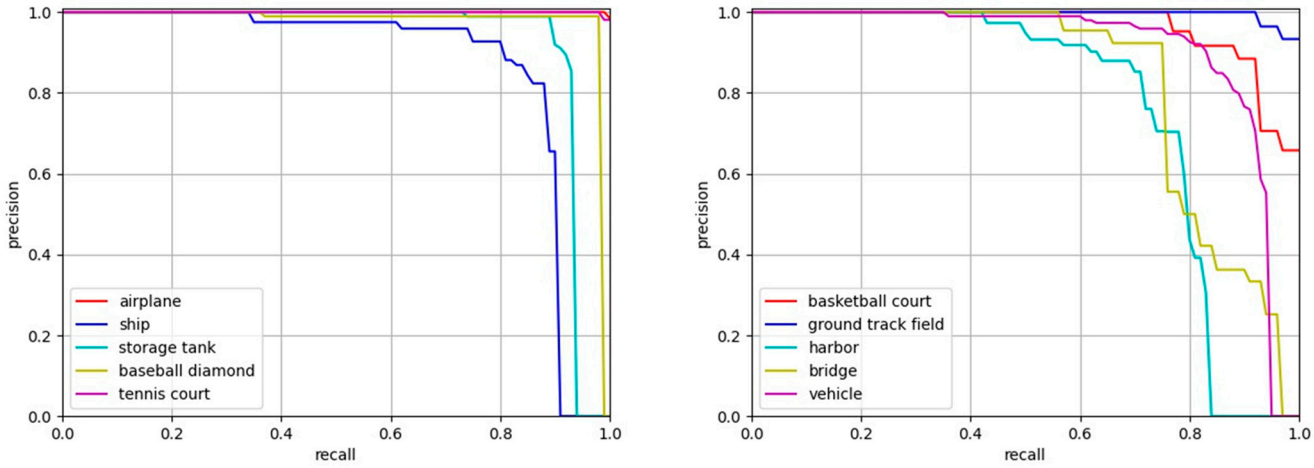

Figure 10. P–R curves of different objects on the NWPU VHR-10 dataset with the threshold set to 0.5.

Table 5. Performance evaluation of the HBB task on the UCAS-AOD dataset.

Task	Method	Plane	Car	mAP (%)
OBB	F3-Net [12]	93.92	98.14	96.03
	EFM-Net(our)	95.10	97.80	96.45
HBB	F3-Net [12]	95.68	98.12	96.90
	EFM-Net(our)	98.20	96.80	97.50

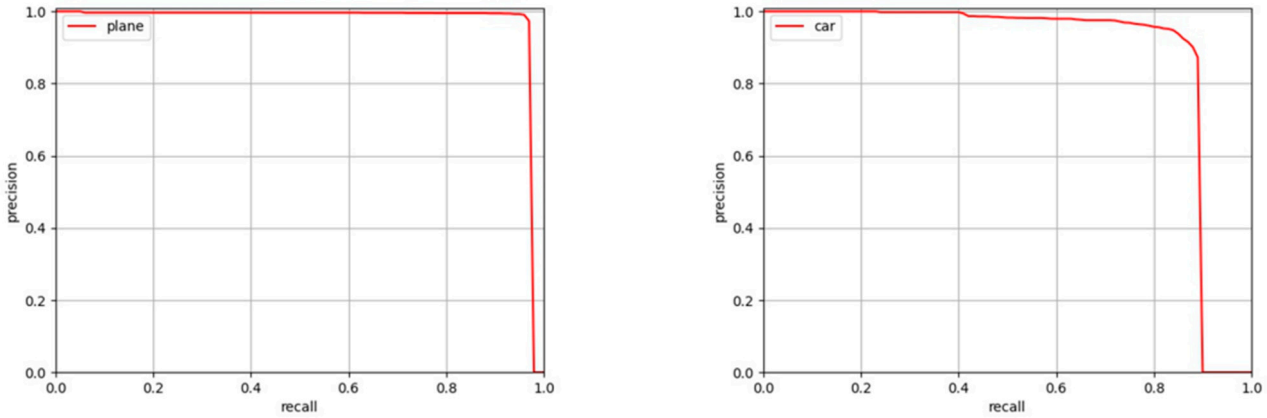

Figure 11. HBB P–R curves of different objects on the UCAS-AOD dataset with the threshold set to 0.5.

5. Discussion

Overall, our study established a new object detection model in the field of remote sensing observation. The advantages of the proposed EFM-Net are illustrated as follows: (1) A large number of small-sized targets may appear in remote sensing images, and their size may be less than 10 pixels. In the convolution process from the backbone to the FPN, the lost feature layer is likely to directly lose the features of the small targets. Therefore, it is problematic to simply use the traditional feature pyramid structure. The MBFE module can better extract features and can avoid feature loss in the process of transformation. (2) Because the background in remote sensing image is complex and because object features may be submerged by the environment, the background filtering module uses the attention mechanism to increase the weight of the objects in the network so that the network can

better focus on the object rather than the environment. (3) The mask improvement module improves the robustness of occluded object detection by a mask generating module and a boundary improvement module. Our research improves the detection accuracy of object detection in remote sensing images.

On the other hand, we chose Faster R-CNN [1] as the baseline model. Compared to one-stage network models such as the Yolo series [2,3,19,20], EFM-Net requires larger video memory. Additionally, while achieving higher detection accuracy, the detection speed also decreases. In future work, we will consider optimizing our network structure to improve the detection speed and to deal with more complex detection content and smaller targets.

6. Conclusions

In this paper, we proposed an object detection network structure, EFM-Net, for remote sensing image detection difficulties. By adding the MBFE module, the network can better capture the context information, especially in when a target is too small or when it has insufficient features. Our network can better reduce the channel and reduce the impact on small targets. During background processing, our background filtering module can better reduce the impact of complex environments on remote sensing targets. We also used a mask improvement module to reduce the influence of occlusion between objects to improve detection robustness. We conducted experiments on the open-source datasets DOTA v1.0, NWPU VHR-10, and UCAS-AOD and also verified the effectiveness of our improvements through ablation experiments. Our experiments show that our method is better than existing methods.

Author Contributions: All authors contributed to this manuscript: Conceptualization, Y.W.; methodology, Y.W.; data curation, Y.W.; supervision L.G.; validation, L.G. and Y.J.; resources, L.G.; writing—original draft preparation, Y.W.; writing—review and editing, L.G. and Y.J.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://captain-whu.github.io/DOTA/tasks.html, https://hyper.ai/datasets/5419, https://hyper.ai/datasets/5422 (accessed on 11 October 2021).

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

References

1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
2. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
3. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
4. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
5. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
6. 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 4th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
7. 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.
8. Everingham, M.; Van Gool, L.; Williams CK, I.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
9. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]

10. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2 cnn: Rotational region cnn for arbitrarily-oriented scene text detection. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 29 November 2018; pp. 3610–3615.
11. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. SCRDet: Towards More Robust Detection for Small, Cluttered and Rotated Objects. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 8232–8241.
12. Ye, X.; Xiong, F.; Lu, J.; Zhou, J.; Qian, Y. F3-Net: Feature fusion and filtration network for object detection in optical remote sensing images. *Remote Sens.* **2020**, *12*, 4027. [CrossRef]
13. Guo, W.; Li, W.; Gong, W.; Cui, J. Extended feature pyramid network with adaptive scale training strategy and anchors for object detection in aerial images. *Remote Sens.* **2020**, *12*, 784. [CrossRef]
14. Xia, G.; 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, UT, USA, 18–23 June 2018; pp. 3974–3983.
15. Ding, J.; Xue, N.; Long, Y.; Xia, G.; Lu, Q. Learning RoI Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
16. Ding, J.; Xue, N.; Xia, G.S.; Bai, X.; Yang, W.; Yang, M.Y.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; et al. Object detection in aerial images: A large-scale benchmark and challenges. *arXiv* **2021**, arXiv:2102.12219.
17. 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]
18. Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3735–3739.
19. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
21. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
22. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
23. 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.
24. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
25. Guo, C.; Fan, B.; Zhang, Q.; Xiang, X.; Pan, C. Augfpn: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 12595–12604.
26. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
28. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 448–456.
29. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Wojna, Z.; Shlens, J. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016.
30. Szegedy, C.; Sergey, I.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 23 February 2016.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
33. 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.
34. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 9308–9316.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 7132–7141.

36. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 11531–11539.
37. 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.
38. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), online, 19–25 June 2021; pp. 13713–13722.
39. Qiu, H.; Ma, Y.; Li, Z.; Liu, S.; Sun, J. Borderdet: Border feature for dense object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2020; pp. 549–564.
40. Wang, X.; Shrivastava, A.; Gupta, A. A-fast-rcnn: Hard positive generation via adversary for object detection. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2606–2615.
41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.
42. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.
43. Sun, P.; Zheng, Y.; Zhou, Z.; Xu, W.; Ren, Q. R4 Det: Refined single-stage detector with feature recursion and refinement for rotating object detection in aerial images. *Image Vis. Comput.* **2020**, *103*, 104036. [CrossRef]
44. Azimi, S.M.; Vig, E.; Bahmanyar, R.; Korner, M.; Reinartz, P. Towards Multi-class Object Detection in Unconstrained Remote Sensing Imagery. In Proceedings of the ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 4–6 December 2018; pp. 150–165.
45. Zhu, Y.; Du, J.; Wu, X. Adaptive period embedding for representing oriented objects in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7247–7257. [CrossRef]
46. Wang, P.; Sun, X.; Diao, W.; Fu, K. FMSSD: Feature-Merged Single-Shot Detection for Multiscale Objects in Large-Scale Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3377–3390. [CrossRef]