RESOLUTION-AWARE NETWORK WITH ATTENTION MECHANISMS FOR REMOTE SENSING OBJECT DETECTION

Zhuangzhuang Tian, Wei Wang*, Biao Tian, Ronghui Zhan, Jun Zhang

College of Electronic Science and Technology, National University of Defense Technology, Changsha, China - gkt_cn@outlook.com, wangwei_nudt@hotmail.com, tbncsz@126.com, (zhanrh, zhangjun)@nudt.edu.cn

KEY WORDS: Remote Sensing, Object Detection, Convolutional Neural Network, Backbone Network, Attention Mechanism

ABSTRACT:

Nowadays, deep-learning-based object detection methods are more and more broadly applied to the interpretation of optical remote sensing image. Although these methods can obtain promising results in general conditions, the designed networks usually ignore the characteristics of remote sensing images, such as large image resolution and uneven distribution of object location. In this paper, an effective detection method based on the convolutional neural network is proposed. First, in order to make the designed network more suitable for the image resolution, EfficientNet is incorporated into the detection framework as the backbone network. EfficientNet employs the compound scaling method to adjust the depth and width of the network, thereby meeting the needs of different resolutions of input images. Then, the attention mechanism is introduced into the proposed method to improve the extracted feature maps. The attention mechanism makes the network more focused on the object areas while reducing the influence of the background areas, so as to reduce the influence of uneven distribution. Comprehensive evaluations on a public object detection dataset demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

In the field of remote sensing image analysis, object detection is a fundamental and important task which has been widely used in traffic controlling, urban planning, environment monitoring, etc.. With the continuous development of remote sensing technology, rapidly increasing data makes manual detection unrealistic. Therefore, automatic object detection has drawn increasing research attention, and researchers have done a lot of work in this field (Cheng et al., 2016, Li et al., 2018, Ding et al., 2019, Li et al., 2019). With the advance of computer vision and machine learning, many machine-learning-based methods, such as saliency detection (Zhang, Zhang, 2017), have been utilized in object detection of remote sensing images and have shown excellent performance on some specific tasks. In recent years, much of the research has benefited from deep learning (Sun et al., 2016, Deng et al., 2017, Tang et al., 2017).

Different from the conventional methods, deep-learning-based object detection methods can automatically learn the effective features from the remote sensing images. These methods can be divided into one-stage and two-stage types according to the detection process. One-stage methods generate dense anchor boxes with multiple scales and aspect ratios at different positions of the images. Convolutional neural network (CNN) is then used for feature extraction, classification and localization. The common single-stage methods contain single shot multibox detection (SSD) (Liu et al., 2016), you only look once (YOLO) (Redmon et al., 2016, Redmon, Farhadi, 2017, Redmon, Farhadi, 2018), RetinaNet (Lin et al., 2017a), etc. There is only one step in the whole process, therefore this kind of methods is fast and efficient. However, in most remote sensing images, the scenes are quite large and the objects are sparely distributed. It makes the positive and negative samples of the anchor boxes extremely imbalance, which aggravates the detection performance. For this reason, the current object detection methods for remote sensing images are mostly two-stage methods such as region-based CNN (RCNN) (Girshick et al., 2014).

The RCNN-based methods include two stages. In the first stage, a series of class-agnostic proposal boxes are generated according to the feature maps extracted by the backbone network. The proposal generation algorithms include selective search algorithm (Uijlings et al., 2013) and region proposal network (RPN) (Ren et al., 2015). In the second stage, the feature maps corresponding to the proposal boxes are cropped and resized into the same size. The resized feature maps are then used to classify and further fine-tune the proposal boxes.

As mentioned above, the generation of proposal boxes, the classification and regression all rely on the extracted feature maps. Therefore, the ability to extract effective features plays a key role in deep-learning-based methods. Although various methods have achieved promising detection performances in remote sensing images, there are still some issues that need to be further studied.

(a) Backbone network of the deep-learning-based methods determines the extracted features. Actually, most backbone networks in the detection methods are fine-tuned from pre-trained classification models, such as VGGNet (Simonyan, Zisserman, 2014) and residual network (ResNet) (He et al., 2016). These networks are usually made up of some basic block, for example, the bottleneck block (He et al., 2016) in ResNet is composed of convolutional layers. The parameters and the number of these blocks determine the width and depth of the network. Although the current VGGNet and ResNet have a variety of forms, most of them do not take into account the consistency of width and depth. In addition, these forms usually ignore the impact of the resolution of input image. In the case of high-resolution remote sensing images, these backbone networks may limit the performance of object detection.

(b) Most remote sensing images contain large scenes and sparse objects. The current methods based on deep learning usually

*Corresponding author

use fully convolutional network to extract features. Therefore, the weight of the extracted features is the same for each position in the image. In this way, the network pays uniform attention to the objects and the background as well. However, for remote sensing images, there are many background areas in the scene. The features of the background areas are likely to interfere with object detection and may lead to false predictions.

To address these problems, in this paper we propose a new object detection method for remote sensing images on the basis of RCNN. First, we use EfficientNet (Tan, Le, 2019) as the backbone network of the proposed framework. For large scenes and various object sizes, EfficientNet can balance the width, depth and input resolution through the compound scaling method. Therefore, it makes the network architecture more consistent with the input resolution. In addition, in view of the large scenes and the sparse objects, we try to make the network pay more attention to the object area rather than the background area. To this end, we apply the attention mechanism (Vaswani et al., 2017) into the proposed method, so as to adjust the weights of the extracted features. Thereby the proposed method can enhance the features of the object regions and decrease the influence of the background.

The remainder of this paper is organized as follows: Section 2 reviews the related works on backbone network and attention mechanism. Section 3 depicts the proposed method for object detection. In Section 4, the experiments are conducted to validate the effectiveness of the proposed method. Section 5 concludes and further discusses this paper.

2. RELATED WORK

In recent years, researchers have tried to improve the RCNN-based object detection methods from many aspects, such as loss function (Rezatofighi et al., 2019) and anchor generation (Wang et al., 2019). In this section, we mainly review the common backbone networks and attention mechanisms.

2.1 Backbone Network

The backbone network for object detection is mainly used for feature extraction. The current methods usually adopt classification networks as the backbone network. In general, these networks are composed of some basic blocks. The channel number of the block determines the width of the network, the number of the blocks and the stacking method determines the depth of the network. According to the experience, the researchers have designed different combinations of the blocks with diverse depths and widths, so as to meet the needs of various tasks. At present, the commonly used backbone networks include VGGNet, ResNet and DarkNet (Redmon et al., 2016).

VGGNet is proposed by Simonyan et al., and proves that the depth of network contributes to the classification accuracy. The basic block of VGGNet is a simple convolutional layer, and spatial pooling is carried out by a max-pooling layer which follows the basic block. In which, the sizes of convolution kernels are 3×3 and the max-pooling strides are 2×2. VGGNet is usually used in SSD and faster RCNN (Ren et al., 2015). ResNet is developed by Kaiming He et al., who proposes shortcut connection to make information propagation smooth and to ease the training procedure. The depth of ResNet is up to 152 layers which is far deeper than VGGNet, meanwhile ResNet has low complexity in computation. In detail, the basic block of ResNet is a bottleneck architecture, which contains a stack of 3 layers. These layers are 1×1, 3×3 and 1×1 convolution, where two 1×1 convolutions are used to reduce and restore the channels respectively, thus to decrease the channels of 3×3 convolution. In terms of object detection, ResNet is widely used in RCNN-based methods. DarkNet is proposed by Joseph Redmond and used in YOLO. The basic block of DarkNet is composed of 1×1 and 3×3 convolutional layers. In addition, DarkNet adopts the shortcut connection as well.

Although the backbone networks mentioned above have achieved good performance in object detection, they do not consider the influence of the input image resolution, i.e., the depth and width may be not consistent with the image resolution.

2.2 Attention Mechanism

Humans usually pay more attention to some specific parts of the visual scene according to their needs, while ignoring the other parts. The above phenomenon is often referred to the attention mechanism. For machine learning, by applying the attention mechanism, we can learn the importance of each element in the feature, and obtain its corresponding weight coefficient.

Inspired by the non-local mean method, non-local neural network (NLNet) (Wang et al., 2018) uses the non-local operation to model the pixel-level pairwise relations and computes the weight coefficient at a position as the weighted sum of the features at all positions. In NLNet, attention weight means the captured long-range dependency which is not constrained by the distance. However the calculation of the pixel-level pairwise relations needs abundant computational resources, and the resource requirement grows rapidly with the increase of image resolution. Then, the criss-cross network (CCNet) (Huang et al., 2018) is proposed to model the pixel-level pairwise relations in a resource-saving way. For each point, CCNet only computes the pixel-level pairwise relations based on its surrounding points on the criss-cross path. The relations between the point and points at the other positions can be obtained through the recurrent operation. Therefore, for each position, the pixel-level pairwise relations can be modeled by the criss-cross attention module. In order to further reduce the amount of computation, Cao et al. propose the global context network (GCNet) (Cao et al., 2019), which assumes that the attention weight is irrelevant to the position of the pixel to be calculated. Specifically, GCNet uses a convolutional layer and a softmax function to calculate the global attention weight. This method can not only reduce the amount of calculation, but also maintain the accuracy to a certain extent.

In addition to calculating the attention weight in the spatial dimension, the relation between different feature channels should also be explored, since the contents in different channels play different roles in the interpretation task. Squeeze-excitation network (SENet) (Hu et al., 2018) tries to apply the attention mechanism in the channel dimension, and allocates different weights to the channels. Specifically, SENet obtains the global distribution of channel-wise response through squeeze operation, and generates the attention weight value of each feature channel by excitation operation. The channel-level attention weights can be used to select and emphasize the informative feature channels and suppress useless ones.

On the basis of RCNN, EfficentNet is used as the backbone network of the proposed framework in this paper. The compound scaling method in EfficientNet can solve the inconsistency of

depth and width in network design in terms of different resolutions. In addition, in order to reduce the influence of the background, we also use the attention mechanism to make the network pay more attention to the object areas.

3. THE PROPOSED METHOD

3.1 EfficientNet

As described in Section 2, the backbone network plays a key role in feature extraction, and the current backbone networks for object detection are usually transformed from the ones designed for classification tasks through fine-tuning, such as ResNet and VGGNet. Although these backbone networks have achieved good results, the width, depth and basic block of the backbone network are usually designed based on experience. Compared with the natural optical images, the scene of the remote sensing images is much larger, and the sizes of the objects have a wider range. These characteristics increase the difficulty of network design. In addition, the backbone network should fit the input data with different image resolutions.

In (Tan, Le, 2019), Tan et al. find that the classification accuracy can be improved by scaling up any dimension of the width, depth or resolution of the network, but these dimensions are not independent. It is important to balance all dimensions during scaling up. Their EfficientNet uses the compound scaling method to uniformly scale the width, depth and resolution in a principled way. The EfficientNet takes the resolution as one of the adjustable dimensions, which needs to be balanced with the other two dimensions, namely width and depth.

Inspired by EfficientNet, we argue that with fixed input resolution, a better dimension balance should be achieved by adjusting width and depth, thus improving the effectiveness of extracted features. Empirically, as resolution increases, we indeed need to expand the receptive fields and extract fine-grained features by increasing the depth and width of the network.Therefore, EfficientNet is used as the backbone network in this paper. It is notable that, because the resolution of the input image is fixed, the compound scaling method only adjusts the depth and width in our detection framework. Specifically, we first need a baseline backbone network, the width and depth are then uniformly scaled by compound coefficient. The scaled network is more suitable for the image resolution and is used as the final backbone network. Due to the excellent performance in (Tan, Le, 2019), we still adopt baseline network EfficientNet-B0 obtained by network architecture search in EfficientNet. Due to the different tasks, we remove the last layer used for classification in the original EfficientNet-B0.

EfficientNet-B0 contains multiple stages and every stage consists of the several mobile inverted bottleneck convolution (MBConv) blocks. Compared with the bottleneck block in ResNet, MBConv is more effective in feature extraction. In detail, MBConv block first uses the depth-wise convolution (Howard et al., 2017) instead of the standard convolution to reduce the amount of computation. Different from the standard convolution, a depth-wise convolution kernel is only responsible for one input channel, namely each channel is convoluted by one convolution kernel. Therefore, the number of the input channels is the same as that of output channels, and it affects the extracted features. To increase the output channels and the diversity of the features, the inverted residual block (Sandler et al., 2018) is adopted in MBConv, and it uses a convolutional layer to expand the channel dimension before the depth-wise convolution. In addition, squeeze-and-excitation is introduced into MBConv, and it is placed after the depth-wise convolution to build the dependencies between different channels. Furthermore, the shortcuts between the bottlenecks are used to improve the gradient propagation between layers. The structure of MBConv is shown in Figure 1, where Conv means convolutional layer, BatchNorm is the batch normalization (Ioffe, Szegedy, 2015), Swish is a kind of activate function (Ramachandran et al., 2017), and FC denotes the fully connection layer. The details of the baseline network Efficient-B0 are listed in Table 1.

Stage	Block	Kernel Size	Stride	Expand Ratio	Channel Number	Block Number
1	Conv	3 × 3	2	1	32	1
2	MBConv	3 × 3	1	1	16	1
3	MBConv	3 × 3	2	6	24	2
4	MBConv	5 × 5	2	6	40	2
5	MBConv	3 × 3	2	6	80	3
6	MBConv	5 × 5	1	6	112	3
7	MBConv	5 × 5	2	6	192	4
8	MBConv	3 × 3	1	6	320	1

Table 1. The details of baseline backbone network Efficient-B0.

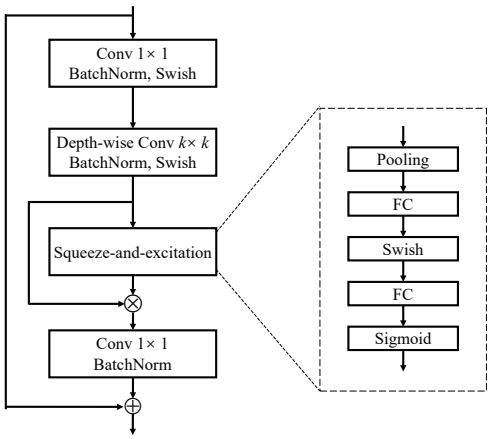


Figure 1. The structure of mobile inverted bottleneck convolution.

After obtaining the baseline network, we need to uniformly adjust the depth and width of the network by using compound scaling method. The depth of the network is related to the number of stages and the number of basic blocks per stage, and the width is related to the number of channels per basic blocks. In EfficientNet, the baseline network EfficientNet-B0 fixes the number of stages. Therefore, we scale up the number of basic blocks and their channels to adjust the depth and width. Assuming the compound coefficient is represented by ϕ, the numbers of basic blocks and channels in the i-th stage are denoted by r_i' and c_i', respectively. The compound scaling method can be expressed as:

$$r_i' = \mathrm{ceil}(\mathrm{r_i} \cdot \alpha^{\phi}), \mathrm{c_i'} = \mathrm{ceil}(\mathrm{c_i} \cdot \beta^{\phi}), \quad (1)$$

where r_i and c_i denote the basic block number and the channel number of the i-th stage in EfficientNet-B0. α and β are

the corresponding hyper-parameters which are set optimally according to the neural architecture search. ceil function can obtain the next highest integer value by rounding up value. By adjusting the compound coefficient ϕ, we can control the numbers of basic blocks and the channels in each stage of the final EfficientNet. Take the fifth stage of the network as an example, there are 3 MBConvs and 80 channels in original EfficientNet-B0. Referring to (Tan, Le, 2019), α and β are set as 1.2 and 1.1, respectively. According to Equation 1, there should be 5 MBConvs and 97 channels in the fifth stage of the EfficientNet when $\phi = 2$, namely $r'_5 = 5$ and $c'_5 = 97$.

3.2 Attention Mechanism

In general, the scene of the remote sensing images is large, while the object areas occupy only a small part of the whole scene. A large amount of information in the background area can easily interfere with object detection. Therefore, it would be beneficial if we suppress the feature of the background areas and make the network more focused on the feature of the object areas. To achieve this end, we apply the attention mechanism into the network.

It is known that the calculation of the weights in the attention mechanism mainly involves two contents: query and key. In the field of computer vision, visual content such as pixels or regions of interest (RoIs) can be considered as queries and keys in the attention mechanism. Xizhou Zhu et al.(Zhu et al., 2019) summarizes the attention mechanism into a general formula:

$$y_q = \sum_{m=1}^{M} W_m[\sum_{k\in\Omega_q} A_m(q,k,z_q,x_k) \odot W'_m x_k], \qquad (2)$$

where z_q denotes the content of the q-th query element, x_k denotes the content of the k-th key element. Ω_q is the key region for the query. $A_m(q,k,z_q,x_k)$ is the attention weight in the m-th attention sub-network, and W_m and W'_m are the learnable weights. M is the number of the attention sub-networks. y_q is the output of the attention mechanism. $\odot$ means element-wise multiplication.

How to get the attention weight by using the query and key in the image is one of the issues we need to consider. As mentioned in Section 2, some methods including NLNet, aggregate the features of all points by building query-specific attention map. These methods can capture the long-range dependency in the image, and have achieved good performance. However, they need to set different weights for different query locations, taking up a lot of computation resources, especially for high-resolution images. In addition, (Cao et al., 2019) find through experiments that although NLNet calculates independent attention weight for every query position, the attention weights of different query locations tend to be the same after training. Namely, the attention weights are independent of the query locations. Thereby, in this paper, we apply the attention method which is independent of the position of query, and is only related to the position of key. Referring to (Zhu et al., 2019), the attention weight can be modeled as:

$$A_m(q,k,z_q,x_k) = \mathrm{softmax}(u_m^{\mathrm{T}} V_m^C x_k), \qquad (3)$$

where V_m^C denotes the learnable embedding matrix for the key content, and u_m is a learnable vector.

According to Equation 2 and Equation 3, the final formula of attention weights is:

$$y_q = \sum_{m=1}^{M} W_m[\sum_{k\in\Omega_q} \mathrm{softmax}(u_m^{\mathrm{T}} V_m^C x_k) \odot W'_m x_k]. \qquad (4)$$

Based on Equation 4, we can get the flowchart of attention weight calculation, as shown in Figure 2. Specifically, for the m-th attention sub-network, the input feature maps x_k go through a convolutional layer, generating the feature maps $\hat{x}_k^m$. $\hat{x}_k^m$ is reshaped and then multiplied with the learnable vector u_m. In this way, the feature maps are weighted and summed up to get $\hat{x'}_k^m$. Then the softmax function is applied to $\hat{x'}_k^m$ to normalize the attention weights and get the final attention weights $\hat{x''}_k^m$. In the other pathway, the input feature maps x_k go through another convolutional layer and the feature maps $\tilde{x}_k^m$ can be obtained. According to the learned attention map $\hat{x''}_k^m$, the feature maps $\tilde{x}_k^m$ are element-wise weighted and summed up to obtain the channel-level dependencies. Subsequently, these dependencies are broadcast in each spatial dimension to match the input features. Finally, the input feature maps and the broadcast channel-level dependencies are element-wise added to generate the enhanced feature maps.

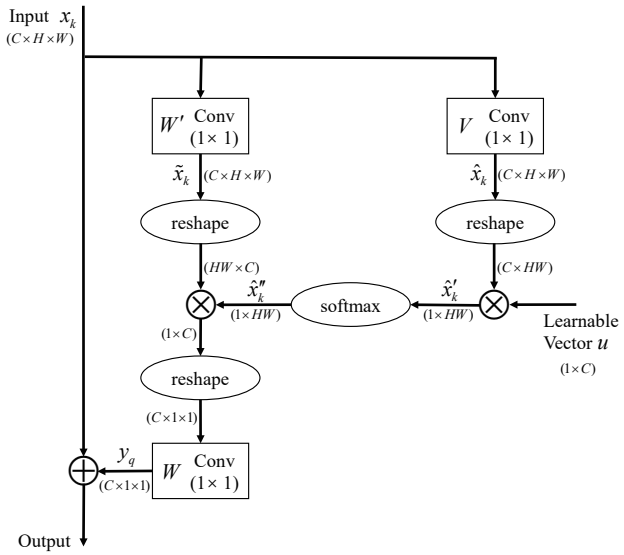


Figure 2. The flowchart of attention weight calculation in the proposed method. The numbers in parentheses denote the shape of the feature map.

4. EXPERIMENTS

In order to verify the effectiveness of the proposed method, we conduct several experiments on the dataset for object detection in aerial images (DOTA) (Xia et al., 2018). All the experiments are conducted on a computer with a central processing unit (CPU) of Intel 6700K, a graphics processing unit (GPU) of NVIDIA GTX 1080Ti, and random access memory (RAM) of 32 GB. All the experiments are implemented based on the open source code base mmdetection (Chen et al., 2019).

4.1 Data Preparation

DOTA is a large object detection dataset, which contains 2,806 images. The original sizes of the images in DOTA range from

about 800×800 pixels to 4000×4000 pixels. DOTA dataset contains 15 categories: 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). There are 188,282 annotated objects in the dataset, and the object sizes range from 10 to 1500 pixels. DOTA dataset includes three subsets: training set, validation set and testing set. The training set contains 1411 images, the validation set contains 458 images and the testing set contains 937 images. Because the testing set does not have the corresponding annotations, the detection performance on the validation set is used for evaluation.

In the experiments, we resize the training set to two scales, 1.0 and 0.4. Then the images are cropped and resized into small images of 1024×1024 in step of 824. As with (Ding et al., 2019), we rotate some samples of the minority categories to augment the dataset and reduce the data imbalance between different categories. The rotation angle is randomly selected in [0, 90, 180, 270]. After all this preparation, a total of 19,757 training samples are obtained. In the testing phase, we also crop the testing image to small images of 1024×1024 in step of 824. These small images are detected separately and the detection results are merged into one which is consistent with the original large image.

4.2 Evaluation Metric

In order to quantitatively evaluate the performance of the proposed method, the commonly used metric, i.e. average precision (AP) is adopted.

The detection results can be divided into three types: true positive (TP), false positive (FP) and false negative (FN) according to the intersection over union (IoU) between the predicted box and the corresponding ground truth box. When the IoU exceeds a threshold, the predicted box can be regarded as TP, otherwise it is considered as FP. FN means the missing object. The precision $\mathcal{P}$ measures the proportion of TP in all detection results and the recall $\mathcal{R}$ measures the proportion of correctly identified positives in all positives. The precision $\mathcal{P}$ and recall $\mathcal{R}$ are defined as:

$$\begin{aligned} \mathcal{P} &= \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})}, \\ \mathcal{R} &= \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}. \end{aligned} \tag{5}$$

AP computes the average value of $\mathcal{P}$ over the interval from $\mathcal{R} = 0$ to $\mathcal{R} = 1$. Due to that the objects have multiple categories, the mean average precision (mAP) is also computed to evaluate the overall performance on the multi-class dataset.

4.3 Experimental Configuration

In the experiments, we adopt EfficientNet as the backbone network. Referring to (Tan, Le, 2019), the hyper-parameters α and β are set as 1.2 and 1.1, respectively. Moreover, the compound coefficient ϕ is set as 2 in our experiments. The overall framework is based on FPN (Lin et al., 2017b), which uses the bottom-up pathway to calculate the feature hierarchy composed of feature maps of multiple scales. The top-down pathway is used to obtain high-resolution feature maps with stronger semantic information. The feature maps of the same size, which are from the bottom-up and top-down pathways, are merged by lateral connection. The merged feature maps are finally used to obtain the final feature maps by convolution. The final feature maps in our experiments have 4 scales, and their strides are $2^2, 2^3, 2^4, 2^5$, respectively. These feature maps are enhanced by attention mechanism and used in subsequent steps. There are 8 attention sub-networks in the experiments. It is worth noting that the backbone network has 8 stages in total, we use the output of the 3rd, 4th, 6th and 8th stages of the backbone network as the input of FPN. The schematic of the detection framework is shown in Figure 3.

RPN is used to generate proposal boxes, where the sizes of anchor boxes are $\{2, 4, 8, 16, 32\}$, and the aspect ratios are $\{0.5, 1, 2\}$. In the training phase, the cross entropy function is used as the loss function of classification, and the smooth L1 loss is used for regression. In the testing phase, the cropped small images are overlapping and the generated proposal boxes are usually redundant. Therefore, the non-maximum suppression (NMS) algorithm is used to filter the similar predicted boxes. The IoU threshold used to determine the correct boxes is set as 0.5.

4.4 Experimental Results and Comparisons

In these experiments, several classic and state-of-the-art methods including one-stage and two-stage ones are used for comparison. These methods contain Faster RCNN, FPN, GCNet, SSD, RetinaNet, and their configurations are introduced as follows:

(1) Faster RCNN. Faster RCNN uses ResNet-50 as the backbone network and uses RPN to generate proposal boxes. The stride of the feature maps generated by backbone network is 16. The loss functions are the same as those in the proposed method.

(2) FPN. FPN also uses ResNet-50 as the backbone network. As the foundation of the proposed method, FPN is the baseline for comparison. Therefore, except for the backbone network and attention mechanism, all the settings of FPN are the same as the proposed method.

(3) GCNet. As mentioned in Section 2, GCNet is an effective detection method which is combined with the attention map. On the basis of FPN, GCNet adds the attention map to enhance the extracted feature maps. Apart from this, GCNet has the same framework and parameter settings as FPN.

(4) SSD. SSD is a one-stage method, which extracts the feature maps of different scales from the images for object detection. There are multiple anchor boxes of different scales and aspect ratios in each sampled position. The features of the anchor boxes are classified and regressed by CNN. In this experiment, SSD uses VGG-16 (Simonyan, Zisserman, 2014) as the backbone network. The anchor boxes adopt 7 scales according to the sizes of the feature maps, and 5 aspect ratios. The cross entropy function and smooth L1 function are also used as the loss functions.

(5) RetinaNet. RetinaNet is a one-stage method based on FPN, and the backbone network is ResNet-50. Backbone network attaches two sub-networks, which are used to classify and regress anchor boxes, respectively. The anchor scales are $\{2^0, 2^{1/3}, 2^{2/3}\}$ and the aspect ratios are $\{0.5, 1.0, 2.0\}$. The main contribution of RetinaNet is that the focal loss is proposed to ease the imbalance between positive and negative samples.

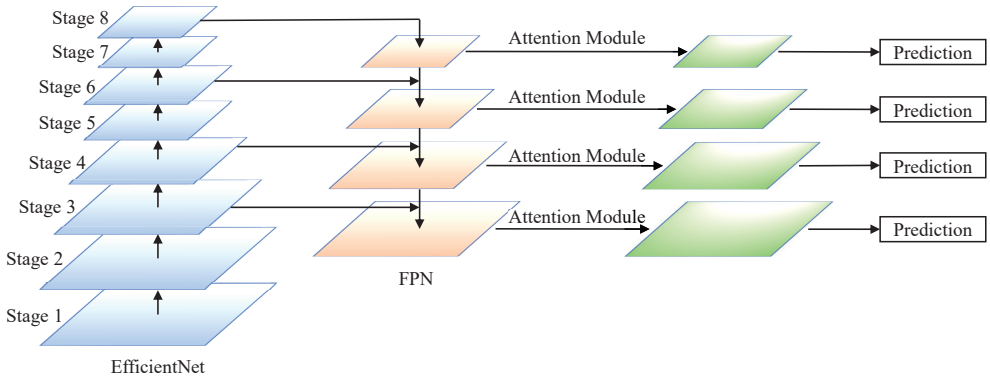


Figure 3. The network structure of the proposed method.

Table 2 lists the APs of all the methods on different categories. The full names of the category acronyms are described in Section 4.1. Nothing that, Faster RCNN, FPN, GCNet and the proposed method are two-stage methods, while SSD and RetinaNet are single-stage methods. As we can see from Table 2, two-stage methods perform obviously better than single-stage methods. In all two-stage methods, the proposed method achieves mAP of 75.0% and outperforms the comparison methods. Compared with the baseline FPN, the proposed method contributes to an improvement of 2.2%.

We also conduct the ablation experiments on the DOTA dataset. In detail, we apply EfficientNet and attention mechanism, respectively, to compare their respective impact on the detection performance. In order to compare these improvements fairly, all the other settings are the same as the baseline (FPN). In addition, we adopt different compound coefficient ϕ in the experiments to compare the impact of different widths and depths. The results are shown in Table 3.

Table 3 shows that both EfficientNet and attention mechanism can improve the detection performance. Specifically, individual EfficientNet can contribute to an improvement of 1.0%, and attention mechanism can increase the detection result by 1.4%. As for the compound coefficient, we find that the network can obtain the best result when the compound coefficient is set as 2.

Several samples of the detection results obtained by the proposed methods are shown in Figure 4. The green, yellow and red rectangles denote true positive, false negative and false positive, respectively.

5. CONCLUSION

In this paper, we propose an effective object detection method for optical remote sensing images. Instead of the conventional backbone networks such as ResNet, we introduce EfficientNet into the proposed method. The compound scaling method of EfficientNet can adjust the width and depth of the network more consistently, thereby solving the problem of network design under different image resolutions, especially high-resolution. Moreover, we apply the attention mechanism in the detection framework. Attention mechanism makes the network pay more attention to the important areas such as the object areas, and suppress the background areas. It can effectively reduce the influence of background noise on object detection. The quantitative comparison results on the public DOTA dataset demonstrate the superiority of the proposed method over several state-of-the-art methods. In the future, more efforts will be made to further investigate the attention mechanism in deep learning.

REFERENCES

Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H., 2019. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. http://arxiv.org/abs/1904.11492.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., Lin, D., 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. http://arxiv.org/abs/1906.07155.

Cheng, G., Zhou, P., Han, J., 2016. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12), 7405-7415.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Zou, H., 2017. Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8), 3652-3664.

Ding, J., Xue, N., Long, Y., Xia, G.-S., Lu, Q., 2019. Learning roi transformer for oriented object detection in aerial images. *2019 IEEE Conference on Computer Vision and Pattern Recognition*.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

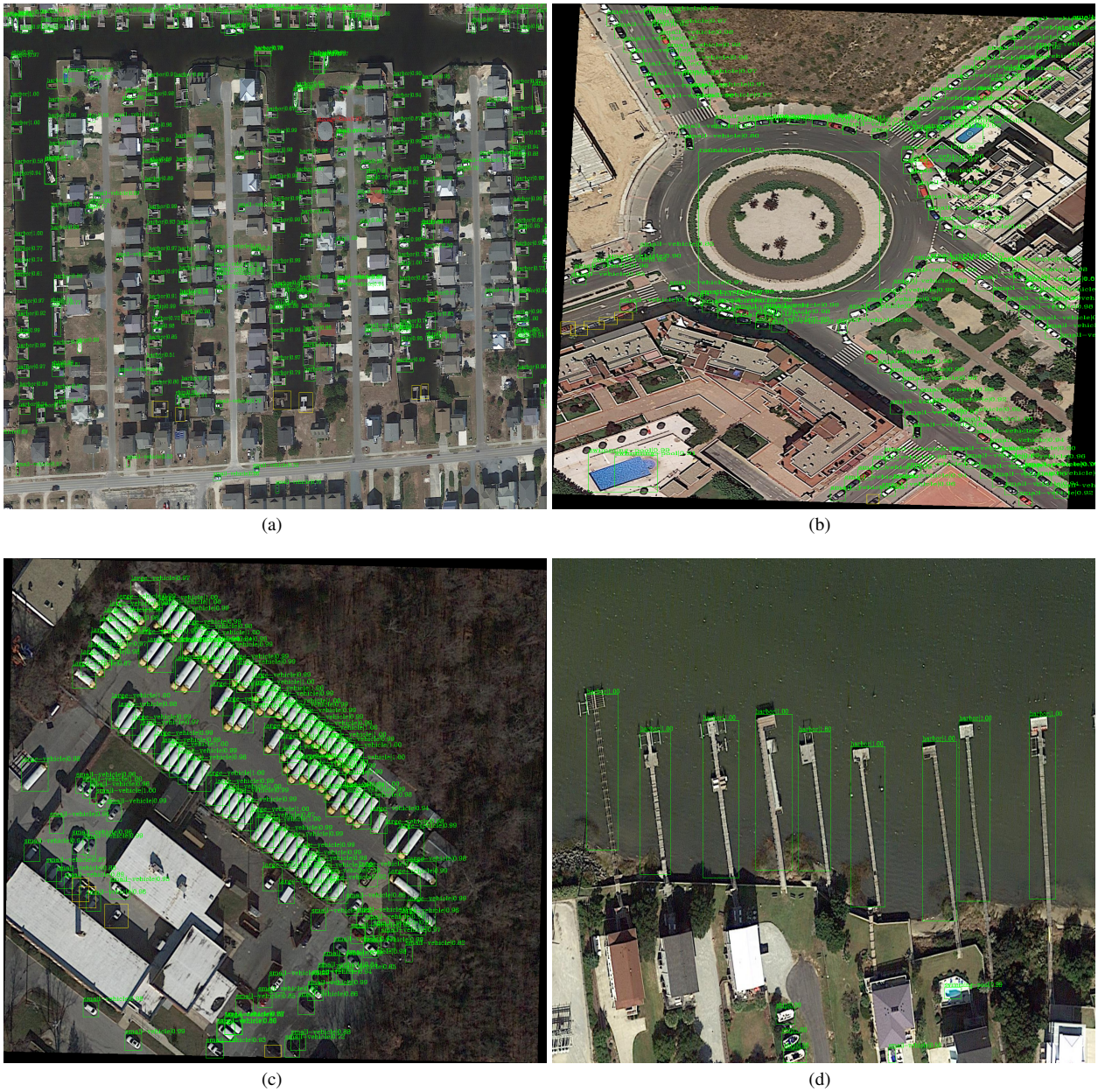
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. http://arxiv.org/abs/1704.04861.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
Faster RCNN	94.7	70.8	40.3	61.2	57.8	71.9	68.0	94.8	64.0	84.1	70.8	72.5	75.2	**78.0**	52.3	70.4
FPN	95.3	73.3	47.6	**69.3**	61.6	77.7	68.9	94.0	59.1	85.4	74.2	72.3	77.7	76.0	59.7	72.8
GCNet	95.9	73.0	48.1	70.5	59.6	**78.8**	69.1	94.1	64.1	87.0	74.2	72.6	**79.1**	74.7	64.4	73.7
SSD	86.8	58.7	21.9	41.4	42.7	57.7	71.3	91.0	44.7	69.7	56.1	39.0	54.6	67.0	18.0	54.7
RetinaNet	93.1	69.2	37.3	53.2	55.5	74.0	**73.5**	93.6	54.4	77.3	72.4	63.8	77.4	70.7	22.5	65.9
Proposed Method	**96.2**	**74.0**	**48.8**	68.9	**63.1**	78.3	70.0	**94.9**	**69.0**	**88.9**	**79.0**	**73.4**	77.6	76.9	**65.7**	**75.0**

Table 2. The mean average precisions of different methods.



(a) (b) (c) (d)

Figure 4. The samples of detection results of the proposed methods.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W., 2018. CCNet: Criss-Cross Attention for Semantic Segmentation. http://arxiv.org/abs/1811.11721.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. http://arxiv.org/abs/1502.03167.

Method	Backbone Network	Attention Mechanism	Compound Coefficient	mAP
1	ResNet-50	×	-	72.8
2	ResNet-50	√	-	74.2
3	EfficientNet	×	0	71.5
4	EfficientNet	×	1	73.4
5	EfficientNet	×	2	73.8
Proposed Method	EfficientNet	√	2	75.0

Table 3. The mean average precisions of ablation experiments.

Li, Q., Mou, L., Liu, Q., Wang, Y., Zhu, X. X., 2018. HSF-Net: Multiscale Deep Feature Embedding for Ship Detection in Optical Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 56(12), 7147-7161.

Li, S., Xu, Y., Zhu, M., Ma, S., Tang, H., 2019. Remote Sensing Airport Detection Based on End-to-End Deep Transferable Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 16(10), 1640-1644.

Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2017a. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision*, 2999–3007.

Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017b. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C., 2016. Ssd: Single shot multibox detector. B. Leibe, J. Matas, N. Sebe, M. Welling (eds), *2016 European Conference on Computer Vision*, Springer International Publishing, Cham, 21–37.

Ramachandran, P., Zoph, B., Le, Q. V., 2017. Searching for Activation Functions. http://arxiv.org/abs/1710.05941v2.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition*.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition*.

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. http://arxiv.org/abs/1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (eds), *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., 91–99.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE Conference on Computer Vision and Pattern Recognition*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE Conference on Computer Vision and Pattern Recognition*.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556.

Sun, H., Liu, S., Zhou, S., Zou, H., 2016. Unsupervised Cross-View Semantic Transfer for Remote Sensing Image Classification. *IEEE Geoscience and Remote Sensing Letters*, 13(1), 13-17.

Tan, M., Le, Q. V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. http://arxiv.org/abs/1905.11946. ICML 2019.

Tang, T., Zhou, S., Deng, Z., Zou, H., Lei, L., 2017. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. *Sensors*, 17(2), 336.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., Smeulders, A. W. M., 2013. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2), 154–171. https://doi.org/10.1007/s11263-013-0620-5.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., Polosukhin, I., 2017. Attention is all you need. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds), *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 5998–6008.

Wang, J., Chen, K., Yang, S., Loy, C. C., Lin, D., 2019. Region proposal by guided anchoring. *2019 IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Xia, G., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. Dota: A large-scale dataset for object detection in aerial images. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 3974–3983.

Zhang, L., Zhang, Y., 2017. Airport Detection and Aircraft Recognition Based on Two-Layer Saliency Model in High Spatial Resolution Remote-Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4), 1511-1524.

Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J., 2019. An Empirical Study of Spatial Attention Mechanisms in Deep Networks. http://arxiv.org/abs/1904.05873.