Efficient Small Object Detection with an Improved Region Proposal Networks

DongWen Ma¹,², XiaoJun Wu¹,²,a and Honghong Yang¹,²

¹Key Laboratory of Modern Teaching Technology, Ministry of Education (Shaanxi Normal University), 710062 Xi’an, China
²School of Computer Science (Shaanxi Normal University), Xi’an, China

Abstract. Although the state-of-the-art object detection methods, which depend on region proposal algorithms to hypothesize object locations, have achieved high detection accuracy, it still struggles in small-size object detection. In this paper, we present a novel method with a multi-scale and multi-tasking region proposal method to effectively detect small object. In the proposed method, multi-scale features and high-level features are employed to locate object position and identify object category, respectively. The main contributions of the proposed approach are two-fold: (1) A simpler way is used to improve the accuracy performance of small object detection, instead of the complex image pyramids and the complex combination framework, and make the object detection task more flexible. (2) Based on multi-scale and multi-tasking approaches, object location information in low layers and object semantic information in deep layers are made fully advantage respectively. The experimental results on the PASCAL VOC dataset show that the proposed method achieves the state-of-the-art object detection accuracy.

1. Introduction
The task of object detection is to find the correct location of the objects and determine the corresponding object category of these objects. It is a challenging task in computer vision field and widely used in our daily lives. Recently, advances in object detection are driven by deep learning methods. Although the object detection method based on Convolutional Neural Network (CNN) achieves satisfied accuracy by using very deep network, it stills a difficult task for the small-size objects detection. Traditional object detection methods via merging multiple segments or scoring candidate windows that probably contain objects to generate region proposals. These methods unusually adopt cues such as super-pixels, edges [1, 2], saliency [3] and shapes [4] as features. Since the traditional methods always with poor accuracy, several methods based on CNN are proposed [5, 6] to improve the detection performance due to the deep learning show great success in object recognition [7]. The object detection algorithms based on deep learning can be roughly categorized into a couple classes, the regional proposal methods, such as the R-CNN [8], SPPNet [5], Fast R- CNN [6], Faster R-CNN [9], R-FCN [10], FPN [11], Mask R-CNN [12], and the regression methods such as You Only Look Once (YOLO) [13,14,15], Single Shot Detector (SSD) [16]. In [8], Ross B. Girshick et al. proposed an object detection network depend on regional proposal method with CNN features (R-CNN). In their work, they combine candidate regions with
convolutional neural networks. This is fulfilled by extracting $\sim 2k$ region proposals by Selective Search [17] and classifies them with a pre-trained CNN model (VGG-16).

Ross Girshick proposed Fast R-CNN to improve the performance of R-CNN in [6]. In their work, Fast R-CNN employs multiple regions of interest (ROIs) and bounding box regression to improve the speed of training and testing and also improve the detection accuracy. In ROIs, the max pooling is used to convert the features inside any valid region of interest into a small feature map. Moreover, the bounding box regression is used as the multi-tasking loss function in the convolutional neural network for training. Despite Fast R-CNN achieves superior performance in object detection, it still relies on Selective Search, which is the major drawback for Fast R-CNN since the process to generate the region proposal is very time consumes. It nearly cost half time of all training.

As the Faster R-CNN has a good detection performance, it became a fundamental algorithm in object detection field. Faster R-CNN introduced a Region Proposal Network (RPN) that utilizes convolutional neural networks to generate candidate regions and combines region proposal generation and detection into a unified network. Usually, RPN is employed to detect regions of interest in the convolutional neural network and many State-of-the-art regional proposed methods also followed this idea, such as literature [11] and [18]. However, the poor localization performance exists in the deep layer, making it still difficult to locate small-size objects. The deep layers of the network can find the object of interest with high recall but poor localization performance. While the low layers of the network can better locate the object but with a low recall. So finding features that includes both strong semantics information and high resolution information is the key to solving the problem of small object detection.

Intuitively, an effective way to solve above problem is combining the features from deep and low layers. Several algorithms are proposed to solve this issue such as [11] [18] and [19]. However, the existed methods tend to find a feature with rich semantic information and precise location information. It is a difficult task to build a feature with both those two kinds information at a same time.

In our paper, we propose a multi-scale and multi-tasking method for object detection. Firstly, the image is fed a pretrained convolutional neural network to obtain the feature maps of each layer. Then achieve the deep feature map through the last convolutional layer and obtain the multi-scale features via fusing feature maps from different convolutional layer. Moreover, we use slight region proposal generative network, which includes two branches–box-regression (\textit{reg}) and box-classification (\textit{cls}), to estimate existence of an object and generate the proposals, respectively. The box-classification is used to estimate the existence of an object by calculating the probability of it belonging to the foreground or background. This is based on that whether the current pixel is in the ground-truth box or not. The box-regression is used to predict the approximate position of each object instance. For each image, the neural networks generate 2000 proposals through non-maximum suppression (NMS) [8]. Finally, these proposals are classified and adjusted via the detection module. The pipeline of this work is shown in Figure.1.

The main contributions of this paper are as follows.

1. A more concise and effective feature fusion method with less parameters and simpler architecture is proposed.

2. We use deep convolutional layers for classification and multi-scale features from different layers for positioning since the low-level features are more suitable for the region of interest (RoI), and high-level semantic features have a great effect on classification. Our experimental results show that the proposed method has a good detection performance, especially for small objects detection.

2. Related Works

The goal of object detection is to localize and recognize object instances in the image with a bounding box [20,21,22]. The traditional approaches generate regional proposals based on ranking the object scores of a bunch of candidate windows. With the emergence and development of deep learning, the neural networks are used to generate regional proposals. Compared with the traditional methods, the
neural networks based on region proposal methods are less computational complexity and less time-consuming.

S. Bell et al. [19] proposed inside-Outside Net, an object detector that exploits information both inside and outside of the interest region. The inside information is a feature that uses a skip pooling to connect feature maps from different convolutional layers, while the outside information is contextual information from the outside of interest region, which is extracted by using spatial recurrent neural networks.

In [18], T. Kong et al. proposed a network based on Hyper Feature. The Hyper Feature is a hierarchical feature that combines the fine information and coarse information from deep and shallow level, respectively. However, it makes the features more abundant.

In [11], T.-Y. Lin et al. proposed a clever feature pyramid method that uses top-down structure and lateral connections to build high-level semantic feature maps from all scales. In their work, the top-down structure is combined by high-level features and low level features. The lateral connection method combines the upsampled result with the same size feature map from the bottom up. However, these methods tend to find features with rich semantic information and precise location information. As known, high semantic information and high-resolution information are inherently contradictory. Therefore, building features with both types of information is a difficult task.

Figure 1. Our Detection Architecture.

3. Multi-scale and Multi-Tasking Implementation for Object Detection

3.1. Multi-Tasking

The proposed multi-tasking approach in our work is consisted by two main parts: (1) multi scale features from different layers are merged together as input of the regression tasks. (2) The features from the deep layer are adopted to represent the high-level semantic for classification.

Different from the region proposal network based on the single scale feature, the region proposal network in our work is build based on the features from different layers. However, keep the same dimension of the features is important for object detection since the classification and regression task of object detection are performed on a same object instance. This means that we need to keep the feature map of positioning and the corresponding feature map for classification in same dimension. It indicates that a candidate box only correspond to one category.

As shown in Figure 1, the proposed method is implemented with two 3×3 convolution layers, followed by two sibling 1×1 convolution layers (reg and cls, respectively). It consists by two parallel steps running through the region proposal network. However, it is not appropriate to use the same multi-scale features in regression and classification tasks. The first reason is that for a candidate region, the multi-scale features generated from different layers may contain noise, which will affect the classification performance. The second one is that adding more location information to category
detection also affects the classification performance. Thus, we can add the semantic information on the original feature maps to improve localization performance.

3.2. Multi-scale Features
There are several ways to combine the multi-scale maps together with subsampling and pooling operations in CNN. In [18], a multi-scale scheme called Hyper Feature is proposed (HyperNet). For different scales of different layers, a max pooling layer is added on the lower level for subsampling. For higher layers, a deconvolutional operation is added to carry out upsampling. Then, the Hyper Feature is generated by performing local response normalization (LRN) [20] on multiple feature maps and concatenates them to one single output cube. By this method, the author claims that more semantic information is achieved. In [11], the proposed FPN uses the idea of a feature pyramid network, which is modified the original network directly. Specifically, each feature map is merged with the adjacent higher-resolution feature maps (which undergo a 1×1 convolutional layer to reduce channel dimensions) by element-wise addition. By this way, the feature map (for prediction) of each layer can combine the resolution information of different layers and the semantic information of different features map. Then, the combined feature maps of different resolutions are used to detect objects with corresponding resolutions.

As shown in Figure 2, the combination method used in this paper is different from the above two. In our work, we add max pooling layer to the low-level features for subsampling, and change the channel size to 512 by a 1x1 convolution operation, then we perform element-wise addition for the high-level features. In terms of connection, contrast to [18], this uses a direct connection method to handle the features of different layers. We use element-wise addition to features. The proposed approach requires fewer parameters. As shown in Table 1, assuming the size of input image is 1000×600, the channel of multi-scale features in our work will be 512, while the channel of HyperNet will be 768.

| Table 1. HyperNet vs. Our multi-scale features (Assume that the size of the input image is 1000 600) |
|-----------------|-----------------|-----------------|
| feature map     | Scale           | Channel         |
| HyperNet [18]   | 215×150         | 768             |
| Ours            | 62×32           | 512             |

3.3. End-to-end Training
As known, YOLO is a typical end-to-end training method. In the traditional method, CNN is only used to classify. The end-to-end method can achieve the object location and identification at the same time via the CNN. In our work, we use the end-to-end training method, the input is the original image and the output is the object instances. Compared with HyperNet, our training process does not require
separate training of object proposal and detection networks, and many fine-tuning steps for region proposals. Hence, in our work, we only need to concern the input and output. We minimize the multi-tasking loss function:

\[
L(k, k^*, t, t^*) = L_{\text{cls}}(k, k^*) + L_{\text{reg}}(t, t^*)
\]  

(1)

The Eq.1 is consists by \( L_{\text{cls}} \) and \( L_{\text{reg}} \). Where \( L_{\text{cls}} \) denotes the classification loss, which is log loss over two classes (object and non object). \( k^* \) and \( k \) are the true and predicted label respectively. The second task \( L_{\text{reg}} \) is the loss that learns the weight of the network by reducing the difference between the ground truth box and the predicted anchor box. The \( t \) is ground truth box and vector \( t^* \) is the predicted anchor box. In our work, all the parameters for \( L_{\text{cls}} \) are obtained by single scale features, while the parameters for \( L_{\text{reg}} \) are obtained by the multi-scale feature.

In addition, in the proposed object detection method, we have made some minor changes for training. Firstly, we use the `crop_and_resize` [23] operator to crop and resize feature maps, which can make the feature information more complete. Secondly, we cancel the step of deleting small proposal (at the original height or width less than 16 pixels), since it would reduce the detection performance, especially for small objects. Thirdly, we exclude ground-truth bounding boxes in the RoIs for training, since they are not accessible for testing and they bias the input distribution for region classification.

4. Experiments

4.1. PASCAL VOC Results

For the performance evaluation, we experiment the proposed method on two challenging publicly available datasets: PASCAL VOC dataset and Coco dataset. The PASCAL VOC dataset is consisted by about 5k trainval images and 5k test images over 20 object categories. The proposed end-to-end training is performed on an NVIDIA Tesla K40 GPU and the public VGG-16 model. In training stage, a mini-batch, involves 1 image and 256 anchors per image, is displayed on the TensorBoard as shown in Figure 3. While the learning rate for training is starts with 0.001 and is reduced after 50k iterations. It can be seen that the loss is basically stable after 120k iterations. For evaluation, we use the standard evaluation metric mean Average Precision (mAP) for quantitatively evaluating the detection performance.

![Figure 3. The results of our training step displayed on the TensorBoard (with IoU =0.7).](image)

We compare the proposed approach with Fast R-CNN and Faster R-CNN for generic object detection on PASCAL VOC 2007. Table 2 shows the detection results with IoU=0.7. As shown in Table 2, the average precision of the proposed method is 70.9\%, which is higher than the Faster R-CNN and the Fast R-CNN. The precision for Faster R-CNN and the Fast R-CNN are 69.9\% and 66.9\%, respectively.
The precision of bottle detection for Faster R-CNN, HyperNet and the proposed method are 54.9%, 62.4% and 64.9%, respectively. Therefore, above experimental results demonstrate that the multi-scale and multi-tasking object detection method proposed in our work has a good detection performance, proved the effectiveness of our detection method.

### 4.2. Analysis for Detection Performance with Different Layers

Due to the deep layer contains higher-level information than low layer, many methods adopt the last CNN layer for region proposal generation or detection [9, 6]. However, recent research [19, 18, 11] shows that combine the multi-scale information can achieve better performance than a single layer.

### Table 4. Detection performance with different layer combination strategies. All networks are trained with the same configuration.

| Layers | Detection mAP |
|--------|---------------|
| 3      | 68.4%         |
| 5      | 73.2%         |
| 3+5    | 76.6%         |
| 3+4+5  | 76.0%         |
| 1+3+5  | 75.6%         |

In the experiment, we trained several different models based on VGG-16. Then, we fuse different layers, such as layer 3 and 5, layer 1, 3 and 5. In this section, all models are trained with the same configuration. The experimental results are shown in Table 4. The combining layer 3 and 5 get the best detection mAP, which proves that the detection performance can be improved by fuse different layers. In addition, since the first layer and final layer are far each other, the information from those two layers is very different. Hence, the merged feature, achieved by element-wise operation the
features of layer 1 and layer 5, has lot of redundant information. Furthermore, as shown in Figure 4, we explore the impact of mAP with different number of proposals for different layers. This result indicates that the multi-scale combination works better than single layer, both for proposal and detection.

**Figure 4.** The mAP versus number of proposals for different layer combinations (with IoU =0.7).

### 4.3. **COCO Results**

In this section, we present more experimental results on the Microsoft COCO object detection dataset [24], which contains involves 80 object categories. We use the COCO trainval35k split (union of 80k images from train and a random 35k subset of images from the 40k image val split). Then, we evaluate the mAP averaged for IoU $\in [0.5:0.05:0.95]$ (COCO’s standard metric, simply denoted as mAP@[.5, .95]) and mAP@0.5 (PASCAL VOC’s metric).

The model in our work is trained on a GPU, the learning rate starts with 0.001 and is reduced after 350k iterations. Training finishes at 900k iterations. We fix one image per batch, using `crop_and_resize` pooling, and keep the small region proposals, which gives better average recall than original method. The rest of the implement details are the same as on PASCAL VOC.

| Method       | data       | Avg. Precision, IoU: 0.50:0.95 | Avg. Precision, Area: S | Avg. Recall, #Dets: 1 | Avg. Recall, Area: S M L |
|--------------|------------|-------------------------------|--------------------------|------------------------|--------------------------|
| Faster R-CNN[9] | trainval   | 21.9                          | 42.7                     | -                      | -                        |
| ION[19]      | train      | 23.6                          | 43.2                     | 23.6                   | 38.3                     |
| R-FCN[10]    | trainval   | 29.9                          | 51.9                     | 10.8                   | 32.8                     |
| SSD300*[16]  | trainval35k| 25.1                          | 43.1                     | 25.8                   | 66.6                     |
| SSD512       | trainval35k| 26.8                          | 46.5                     | 27.8                   | 9.0                      |
| YOLOv2[14]   | trainval35k| 21.6                          | 44.0                     | 19.2                   | 5.0                      |
| Ours         | trainval35k| 25.3                          | 44.5                     | 25.9                   | 10.5                     |

We evaluate the proposed approach on the COCO dataset and compare test-dev results with the state-of-the-art methods, including both one-stage and two-stage models. The experiment results are shown in Table 5. Compared with the two-stage methods, our approach has higher precision performance and as shown in Table 5, our multi-scale and multi-tasking method achieves better performance than one-stage methods, especially in the detection of small-size objects. For size S objects, our approach achieves a 2.9 point AP (Average Precision) gap (10.5 vs. 9.0) and 2.9 point AR (Average Recall) gap (16.9 vs. 14.0) with the SSD512 [16]. The good performance of the proposed method is achieved owning to the proposed method has a good multi-scale feature representation, and the multi-tasking implementation can gives consistent boost on small-size objects. Figure 5 shows some results of the proposed method and the original Faster R-CNN on the PASCAL VOC 2007 dataset.
Figure 5. Some results compare the original Faster R-CNN

5. Conclusions
In this paper, we have proposed a clean and simple framework for object detection, which is which is use the information from different convolutional layers. Compared with the original Faster R-CNN and the state-art-of-the method, our method shows obvious improvement on the object detection, especially in small-size object detection. In our future work, we will consider to accelerate the training speed of the proposed method and apply it to video object detection.

Acknowledgments
This work was partially supported by the National Key Research and Development Program of China under Grant 2017YFB1402102, the National Natural Science Foundation of China under Grant 11872036, Grant 11502133, Grant 11772178, Grant 11372167, the 111 Project under Grant B18032, and the Fundamental Research Funds for the Central Universities under Grant GK201703082, Grant GK201801004.
References

[1] Zitnick, C. Lawrence, and P. Dollár. *Edge Boxes: Locating Object Proposals from Edges.* Computer Vision – ECCV 2014. Springer International Publishing, (2014):391-405.

[2] Cheng, Ming Ming , et al. "BING: Binarized Normed Gradients for Objectness Estimation at 300fps." Computer Vision & Pattern Recognition IEEE,(2014).

[3] Alexe, B, T. Deselaers, and V. Ferrari. "What is an object?" Computer Vision and Pattern Recognition IEEE, (2010):73-80.

[4] Lee, Tom , S. Fidler , and S. Dickinson . "Learning to Combine Mid-Level Cues for Object Proposal Generation." IEEE International Conference on Computer Vision IEEE, (2016).

[5] Zhang, Yuting, et al. "Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, (2015):249-258.

[6] Girshick, Ross. "Fast R-CNN." Computer Science (2015).

[7] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks." International Conference on Neural Information Processing Systems Curran Associates Inc. (2012):1097-1105.

[8] Girshick, Ross, et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, (2014):580-587.

[9] Ren, S., et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis & Machine Intelligence 39.6(2017):1137-1149.

[10] Dai, Jifeng, et al. "R-FCN: Object Detection via Region-based Fully Convolutional Networks." (2016).

[11] Lin, Tsung Yi, et al. "Feature Pyramid Networks for Object Detection." IEEE Conference on Computer Vision and Pattern Recognition IEEE, (2017):936-944.

[12] He, Kaiming, et al. "Mask R-CNN." IEEE International Conference on Computer Vision IEEE Computer Society, (2017):2980-2988.

[13] Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." Computer Vision and Pattern Recognition IEEE, (2016):779-788.

[14] Redmon, Joseph, and A. Farhadi. "YOLO9000: Better, Faster, Stronger." IEEE Conference on Computer Vision and Pattern Recognition IEEE, (2017):6517-6525.

[15] Redmon, Joseph, and A. Farhadi . "YOLOv3: An Incremental Improvement." (2018).

[16] Liu, Wei, et al. SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016. Springer International Publishing, (2016):21-37.

[17] Van, de Sande, Koen E. A, et al. "Segmentation as selective search for object recognition. " 104.2(2011):1879-1886.

[18] Kong, Tao, et al. "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection." Computer Vision and Pattern Recognition IEEE, (2016):845-853.

[19] Bell, Sean, et al. "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks." IEEE Conference on Computer Vision and Pattern Recognition IEEE, (2016):2874-2883.

[20] Everingham, Mark, et al. "The Pascal, Visual Object Classes Challenge: A Retrospective." International Journal of Computer Vision111.1(2015):98-136.

[21] Russakovsky, Olga, et al. "ImageNet Large Scale Visual Recognition Challenge." International Journal of Computer Vision 115.3(2015):211-252.

[22] Yang H, Qu S, Chen C, et al. Multiple objects tracking with improved sparse representation and rank based dynamic estimation[J]. IEEE Access, (2018), PP(6):42264 – 42278.

[23] Chen, Xinlei, and A. Gupta. "An Implementation of Faster RCNN with Study for Region Sampling." (2017).

[24] Lin, Tsung Yi, et al. "Microsoft COCO: Common Objects in Context." 8693(2014):740-755.