INCEPTION SINGLE SHOT MULTIBOX DETECTOR FOR OBJECT DETECTION

Chengcheng Ning, Huajun Zhou, Yan Song, Jinhui Tang*

ABSTRACT

In the current object detection field, one of the fastest algorithms is the Single Shot Multi-Box Detector (SSD), which uses a single convolutional neural network to detect the object in an image. Although SSD is fast, there is a big gap compared with the state-of-the-art on mAP. In this paper, we propose a method to improve SSD algorithm to increase its classification accuracy without affecting its speed. We adopt the Inception block to replace the extra layers in SSD, and call this method Inception SSD (I-SSD). The proposed network can catch more information without increasing the complexity. In addition, we use the batch-normalization (BN) and the residual structure in our I-SSD network architecture. Besides, we propose an improved non-maximum suppression method to overcome its deficiency on the expression ability of the model. The proposed I-SSD algorithm achieves 78.6% mAP on the Pascal VOC2007 test, which outperforms SSD algorithm while maintaining its time performance. We also construct an Outdoor Object Detection (OOD) dataset to testify the effectiveness of the proposed I-SSD on the platform of unmanned vehicles.

Index Terms— Object detection, Inception, SSD

1. INTRODUCTION

Object detection has been studied for many years. Since the R-CNN [1] was proposed in 2014, which is based on deep convolutional neural networks, object detection has developed greatly. Subsequently, lots of improved methods based on the R-CNN, such as Spp-net [2], fast R-CNN [3], faster R-CNN [4] and R-FCN [5], emerged in the object detection area. These methods achieved high accuracies, but their network structures are relatively complex.

In order to speed up, researchers proposed methods based on a single network which directly predict bounding boxes without region proposals. YOLO [8] is a regression based method which returns many object borders and give their recognition confidence directly. Because of its refined network structure, it could achieve real-time processing on GPUs for object detection. However, this method has the problem of inaccurate positioning. WeiLiu et al. [7] proposed the Single Shot Multi-Box Detector (SSD) based on the idea of the Faster R-CNNs Anchor method [4]. It is based on the traditional classification networks VGG [9] and introduces extra layers as feature extraction layers. The scale change of the extra layers is obvious, thus it is able to detect multi-scale objects. However, on the detection of small objects, its performance is not very satisfying.

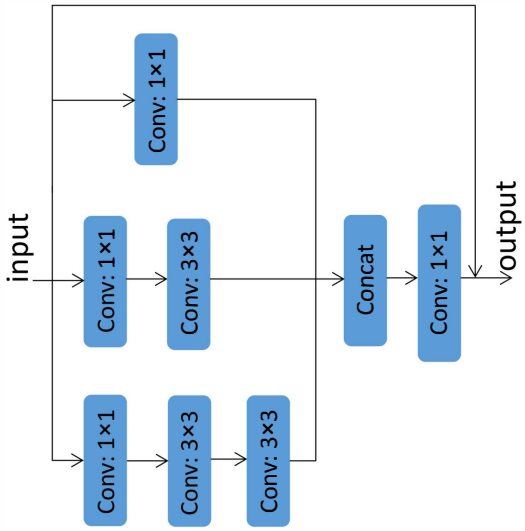


Fig. 1. The Inception building block [6].

To solve the above mentioned problems of SSD, we propose an improved SSD method, named I-SSD. Inspired by the GoogLeNets Inception block and the deep residual network, we redesign the network structure of SSD, which makes it more accurate at positioning. As known, when the network goes deeper, its ability of abstracting features becomes stronger. But it will also bring about some training problems, such as gradient disappearance and over-fitting. Considering the tradeoff between performance and speed, we introduce the Inception structure in extra layers after VGG16 by increasing the type of convolution kernels. As a consequence, the scope of the receptive field is expanded, which increases the sensitivity of the model to small objects without losing large objects. Fig. 1 shows the structure of the Inception building block. In addition, we proposes an improved non-maximum suppression method. We list the main contributions of this work as follows:

- We introduce Inception building blocks into SSD struc-

[1] Yan Song is the corresponding author (e-mail: songyan@njust.edu.cn).

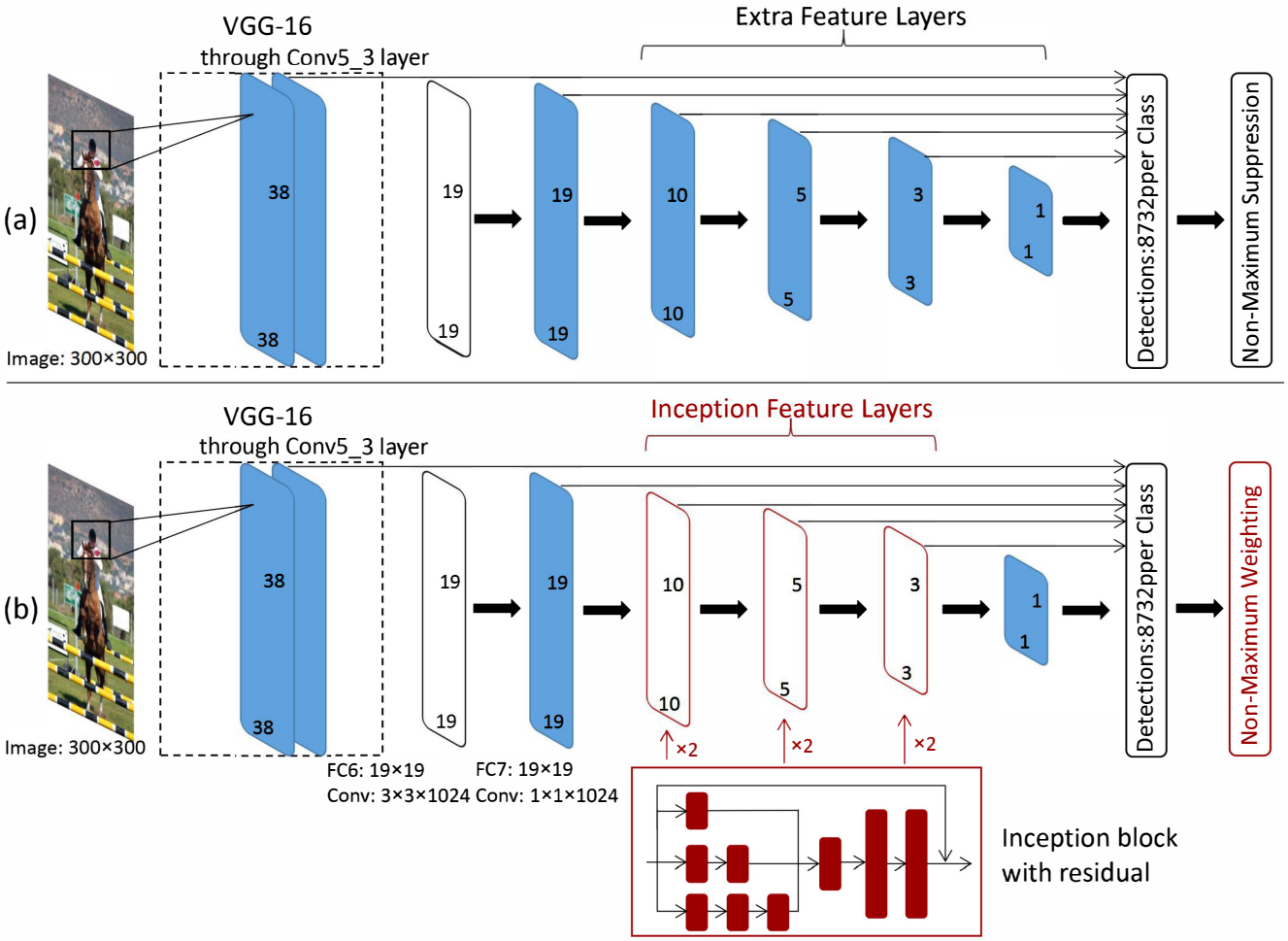


Fig. 2. (a) The architecture of SSD network [7]. (b) The architecture of I-SSD network.

ture to improve its performance by catching more information without increasing its complexity;

- We improve the non-maximum suppression by calculating the weighted average of the bounding boxes that are considered as the same object to generate the final output.

2. RELATED WORK

In the field of object detection, the deep neural networks have outperformed the traditional methods such as Selective Search [10]. The deep learning methods for object detection can be divided into two kinds, including the region-proposal based methods and the regression based methods. The first kind includes R-CNN [1], SPP-net [2], fast R-CNN [3], faster R-CNN [4], PVANET [6] and R-FCN [5], which generate object boxes in the first stage, and use deep neural networks for classification and location regression in the second stage. The early R-CNN [1], SPP-net [2], and fast R-CNN [3] generate region proposals by Selective Search method [10], which is a bottleneck of the whole algorithm. Subsequently, faster R-CNN [4] abandons Selective Search method [10], which uses Region Proposal Network (RPN) [4] instead to generate regional proposals. The PVANET [6] modifies VGG16 [9] with the Inception block on the faster R-CNN [4]. On the VOC2012 dataset of the PASCAL VOC Challenge, the R-FCN [1] ranked the first, which is based on the faster R-CNN [4] and the deep residual network [11]. Although the R-CNN based methods are currently the state-of-the-art, they cannot achieve real-time processing.

The regression based methods include YOLO [8] and SSD [7], which only use a single network to generate bounding boxes and classifications simultaneously. These methods can achieve real-time processing on GPUs. YOLO [8] divides the input image into grids, and each mesh predicts the confidence and the locations of two object boxes. It is a limit to the performance when there is a lot of objects. SSD associates a set of default boxes with feature maps at the top of the network, which can identify objects with various scales and as-

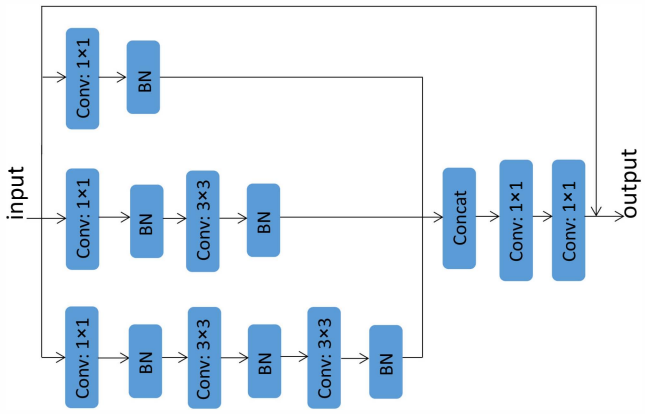


Fig. 3. The Inception building block with BN and residual structure.

pect ratios. It achieves multi-scale effects by adding several additional convolution layers. The feature maps of these additional convolution layers are very important which directly influences the final position and confidence of the bounding boxes. However, the size of the extra layers is relatively small which contain limited information of small objects. As SSD network goes deeper, this information gets less. In this paper, we use Inception building block and residual structure to maximize the retention of the information of small objects.

3. ALGORITHM

In this section, we will introduce the proposed method for objects detection in detail. In section 3.1, we mainly introduce how to construct I-SSD network structure. In section 3.2, we describe the implementation details of the improvement of the non-maximum suppression.

3.1. Inception SSD

Deep neural networks have a more complex model structure compared with traditional methods, which are trained on a large number of data to obtain relatively better performance. In general, the performance can be improved by increasing their depth and width. However, this will cause the parameter increase, which may lead to computation increase and overfitting.

The main idea of Inception is to use dense components to approximate the optimal local sparse structure [12]. The Inception structure not only utilizes the high performance of dense matrix, but also keeps the sparsity of the network. GoogLeNet Inception V1 [12] argued that the basic method to solve the two problems caused by the parameter increase is to convert the full connection layers or even half of the convolution layers to sparse links. Therefore, we believe that Inception block could catch more information without increasing networks' complexity.

SSD uses VGG network as its basis, and adds extra layers to capture objects, as shown in Fig. 2(a). The extra layers of SSD only have one type of convolution kernel, which is the 3×3 kernel. The feature maps of these extra layers will produce the objects' location offset and confidence by small convolution operations. In object detection, large convolution kernels tend to capture large objects, and small receptive fields can locate small objects. Hence, the top feature maps are likely to miss objects details. Thereby, we modify the last few layers of SSD by replacing them with Inception blocks, as shown in Fig. 2(b). Here, convolution kernels of different sizes are stacked, which have different receptive fields. Specifically, we stack 1×1 convolutional layer, 3×3 convolutional layer and 5×5 convolutional layer instead of the original 3×3 convolutional layer in the extra layers. 5×5 convolutional layer is replaced with a sequence of two 3×3 convolutional layers. In this way, we could retain more object details. We reduce the number of feature maps in each layer of the Inception block while keeping the sum of feature maps as same as that in the original extra layers. In order to reflect the significance of various scales of the receptive fields, we give different weight to the output of each type of convolution ($conv1 \times 1$, $conv3 \times 3$ and $conv5 \times 5$) by $w = \{1, 2, 1\}$. We set $w = 2$ for 3×3 convolutional layer and $w = 1$ for the others. In addition, different from [13], we use BN after every convolutional layer in the Inception block. We use two 1×1 convolutional layers on the top. Fig. 3 shows our modified Inception block.

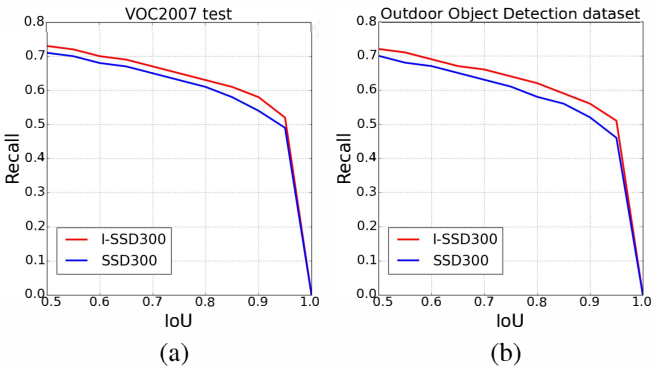
We replace $conv6$, $conv7$ and $conv8$ with the Inception block. Specifically, each of them is replaced by three convolutional towers. We use $conv4\_3$, $fc7$, $conv6\_2\_res$, $conv7\_2\_res$, $conv8\_2\_res$ and $conv9\_2$ as the feature extract layer to detect objects. However, as the network goes deeper, it becomes more difficult to get converged. In order to overcome this problem, we introduce the residual architecture in the extra layers. We concatenate the inputs and outputs of $conv6\_2\_res$ as the inputs of $conv7\_1$.

3.2. Bounding box selection method

SSD method is based on a feed-forward convolutional network, followed by a non-maximum suppression step to produce final detections [7]. A group of bounding boxes (b-boxes) is identified as the same object when the IoU value is higher than a threshold. Let B indicate the b-boxes set, and C_i indicate the confidence of the i^{th} b-box, the non-maximum suppression can be expressed as follows:

$$box = B_{\underset{i}{argmax} C_i} \tag{1}$$

where box represents the b-box with the highest confidence. These boxes are chosen as the final output. However, it is possible that the non-maximum detection results contain the maximum value of the features, so it is inappropriate to directly ignore the detection results of the non-maximum.



(a) (b)

Fig. 4. Recall vs. IoU overlap ratio on the VOC2007 and our OOD dataset.

Suppose that all the b-boxes are from the same object, we make the best use of the object information by considering the non-maximum results rather than simply keep the b-boxes with the highest confidence. So we propose a new b-box selection method called the Non-Maximum Weighting (NMW) by:

$$box = \frac{\sum_{i=1}^{n} w_i \times B_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

$$w_i = C_i \times iou\left(B_i, B_{\underset{i}{argmax} C_i}\right) \tag{3}$$

where n is the number of the b-boxes, and $iou\left(B_i, B_{\underset{i}{argmax} C_i}\right)$ is the IoU of the i^{th} b-box and the b-box with the max confidence, w is the weight for each b-box.

4. EXPERIMENT

We carry out experiments on two datasets, including the union of PASCAL VOC2007 and VOC2012 as well as the Outdoor Object Detection (OOD) dataset that we collect. We use the deep learning library Caffe to build the network and train the model on NVIDIA Tesla K20c GPU. As for the network initialization, we use the pre-trained parameters on the ILSVRC CLS-LOC dataset [14] for VGG16 and "xavier" method [15] for the Inception building block. In all experiments, We use SGD with initial learning rate set to 0.001, momentum set to 0.9, weight decay set to 0.0005, and batch size set to 32. We set the input resolution as 300×300.

4.1. PASCAL VOC dataset

The PASCAL VOC dataset contains 16551 trainval images (PASCAL VOC2007 trainval, VOC2012 trainval and test) and 4952 test images (VOC2007 test) over 20 categories [4]. To demonstrate the effectiveness of the proposed method, we use the same training strategy as SSD algorithm [7]. We first train the model with 10^{-3} learning rate for 80k iterations, and then continue training for 20k iterations with 10^{-4} and 20k iterations with 10^{-5}.

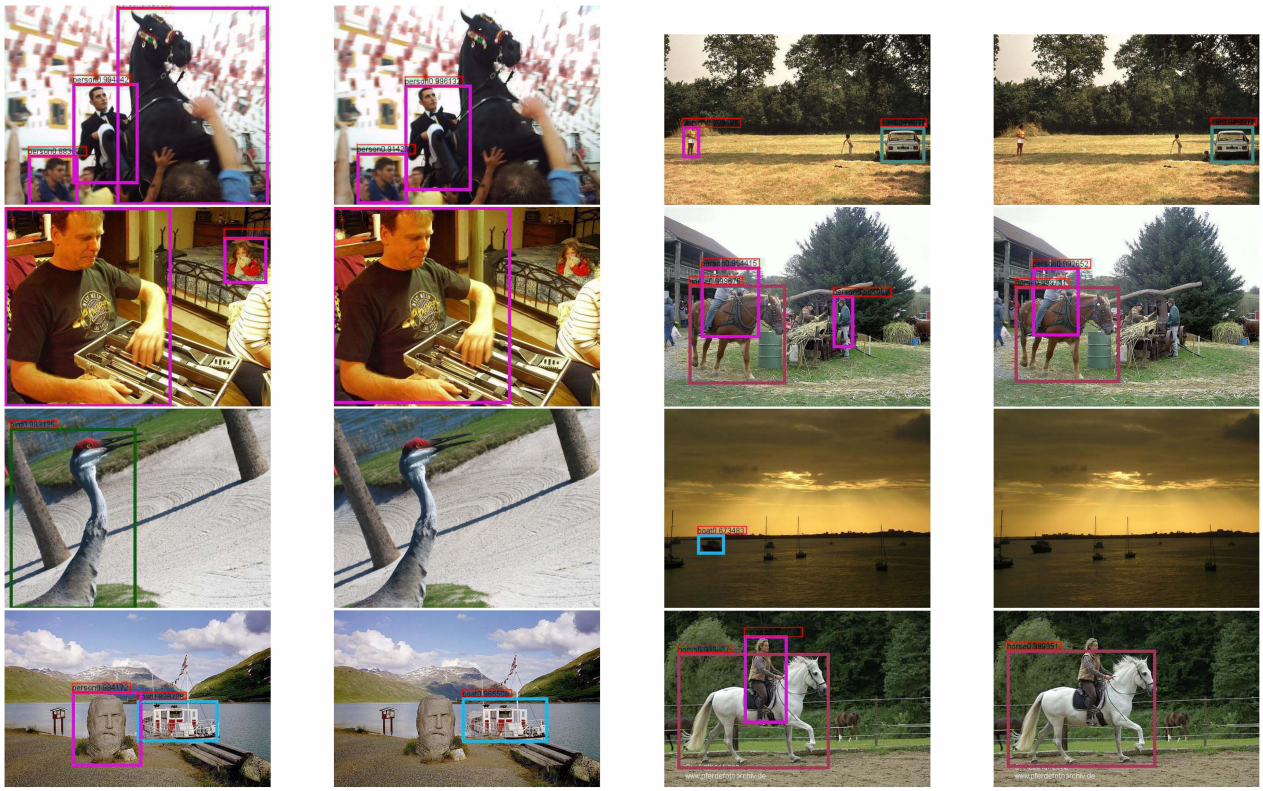
Table 1 shows the mAPs of our model compared with SSD and other object detection methods on the PASCAL VOC2007 test. Fast R-CNN uses Selective Search to produce proposals, and takes the entire image and the proposals as the input of VGG16 network for recognition and regression. Faster R-CNN uses a more efficiency method to produce proposals called RPN network. Both the Fast and Faster R-CNN use input images whose resolution is $\sim 1000 \times 600$. While I-SSD and SSD use a single network with the input size 300×300.

Compared with these methods, I-SSD achieves the highest ranking for eighteen classes. In particular, for the classes of $bottle$, $chair$, cow, $table$, $mbike$, $sofa$ and $train$, the performances are improved significantly. We list the recall on VOC2007 test in Fig. 4(a). When the IoU is 0.5, the recall of I-SSD is around 72.8% while that of SSD is 71.0%. As IoU increases, the recall of I-SSD keeps higher than that of SSD. Fig. 5 shows several samples of detection results of I-SSD and SSD on VOC2007 test. It demonstrates that our method has a stronger ability to detect and recognize objects of small scale. On the other hand, the testing speed of our model is still fast, which is 16 FPS on Tesla k20c GPU compared to 17 FPS of SSD. We list the time performance in Table 2.

4.2. Outdoor Object Detection dataset

The OOD dataset is constructed for our unmanned vehicles project. The unmanned vehicles are supposed to have the ability to automatically avoid trees, cars, stones, stairs and people on the road when it is running outdoors. We collect images from three sources, which are real-world photos, the PASCAL VOC datasets and the Internet pictures. There are about 12338 images in total, containing 5 class (tree, people, car, stone and stair). We choose about one-tenth (1284) images as the test set.

In this experiment, we first train the model with 10^{-3} learning rate for 80k iterations, and then continue training for 20k iterations with 10^{-4}. Table 3 lists the results, from which we can see that our I-SSD outperforms SSD on every category. Fig. 6 shows several detection examples on this dataset. Fig. 4(b) shows the chart of recall v.s. IoU compared with SSD on OOD dataset. The recall of I-SSD is around 71.8% when the IoU is 0.5. I-SSD keeps higher than SSD when IoU increases. I-SSD achieves more correct localizations because it uses more types of convolution kernels, which contain more object informations. Still, I-SSD runs in real-time on this dataset, and the time performance results are showed in Table 2.



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

Fig. 5. Examples of the detection results of I-SSD and SSD on PASCAL VOC2007 test. Different colors of boxes represent different categories. We set the confidence threshold to 0.6. Column (a) shows the results of I-SDD and column (b) shows the results of SSD.

Fig. 6. Samples of OOD dataset.

Table 2. Time performances of different methods. (test on Tesla K20c GPU.)

Method	dataset	FPS	batch size	# Boxes
Faster R-CNN	PASCAL VOC	4	1	~6000
SSD300	PASCAL VOC	17	1	8732
SSD300	PASCAL VOC	17	8	8732
SSD300	OOD	16	1	8732
SSD300	OOD	17	8	8732
I-SSD300+NMW	PASCAL VOC	15	1	8732
I-SSD300+NMW	PASCAL VOC	16	8	8732
I-SSD300+NMW	OOD	16	1	8732
I-SSD300+NMW	OOD	16	8	8732

5. CONCLUSIONS

This paper mainly introduces an improved algorithm named I-SSD. We use the Inception building block in the extra layers and add BN as well as the residual structure. At the same time, we modify the output layer of the network by considering the non-maximum. We validate the effectiveness of our I-SSD on two datasets, including PASCAL VOC2007 and OOD dataset. Compared with SSD algorithm, our improved algorithm achieves a higher mAP, while maintaining a fairly fast speed.

Table 1. Detection results of PASCAL VOC2007 test.

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [3]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [4]	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	77.2	81.3	**85.3**	76.6	**70.9**	50.0	84.3	85.5	88.1	59.0	79.8	76.0	86.1	87.3	84.2	79.4	51.9	77.7	77.7	87.7	75.3
I-SSD300	78.2	80.9	83.7	77.3	70.6	51.8	**86.7**	86.5	87.9	**62.7**	82.1	77.0	86.3	**88.8**	85.6	79.2	52.7	77.5	80.7	88.2	**78.0**
I-SSD300+NMW	**78.6**	**82.4**	84.3	**78.1**	70.6	**52.8**	85.7	**86.8**	**88.3**	62.4	**82.7**	**78.0**	**86.7**	88.3	**86.0**	**79.9**	**53.4**	**78.5**	**80.9**	**88.5**	77.8

Table 3. The performance on OOD dataset.

Method	mAP	person	car	tree	stone	stair
Faster R-CNN	66.9	63.1	69.0	54.5	70.4	77.5
SSD	78.9	77.5	79.4	67.8	83.2	86.7
I-SSD300	79.5	78.4	81.1	67.9	82.6	87.7
I-SSD300+NMW	**79.9**	**78.7**	**81.4**	**68.0**	**83.6**	**87.8**

6. ACKNOWLEDGEMENT

This work was supported in part by the 973 Program under Grant 2014CB347600; In part by the National Nature Science Foundation of China under Grants 61672285. Natural Science Foundation of Jiangsu Province under Grant BK20140058.

7. REFERENCES

[1] R Girshick, J Donahue, T Darrell, and J Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," pp. 580–587, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[3] Ross Girshick, "Fast r-cnn," *Computer Science*, 2015.

[4] S. Ren, K. He, R Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks.," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1–1, 2016.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," pp. 379–387, 2016.

[6] Kye Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park, "Pvanet: Deep but lightweight neural networks for real-time object detection," *arXiv preprint arXiv:1608.08021*, 2016.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg, "Ssd: Single shot multibox detector," pp. 21–37, 2016.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," *Computer Science*, pp. 779–788, 2016.

[9] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2015.

[10] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," pp. 770–778, 2016.

[12] C Szegedy, Wei Liu, Yangqing Jia, and P Sermanet, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[13] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.