Scalable Object Detection Using Deep but Lightweight CNN with Features Fusion

Qiaosong Chen(✉), Shangsheng Feng, Pei Xu, Lexin Li, Ling Zheng, Jin Wang, and Xin Deng

Chongqing Key Laboratory of Computational Intelligence,
Chongqing University of Posts and Telecommunications, Chongqing, China
chenqs@cqupt.edu.cn

Abstract. Recently, deep Convolutional Neural Network (CNN) is becoming more and more popular in pattern recognition, and have achieved impressive performance in multi-category datasets. Most object detection system include three main parts, CNN features extraction, region proposal and ROI classification, just like Fast R-CNN and Faster R-CNN. In this paper, a deep but lightweight CNN with features fusion is presented, and our work is focused on the improvement of the features extraction part in Faster R-CNN framework. Inspired by recent technical innovation structures, such as Inception, HyperNet and multi-scale construction, the proposed network is able to result in lower computation consumption with considerable deep layers. Besides, the network is trained with the help of data augmentation, fine-tune and batch normalization. In order to apply scalable with features fusion, there are different sampling methods for different layers, and various size kernel to extract both global and local features. Then fuse these features together, which can deal with diverse size object. The experimental results shows that our method have achieved better performance than Faster R-CNN with VGG16 on VOC2007, VOC2012 and KITTI datasets while maintaining the original speed.

Keywords: Deep CNN · Features fusion · Multi-scale · Object detection

1 Introduction

Object detection and classification is a hot topic in the field of computer vision. Recently, object detection and classification have got widely applications in many aspects, such as intelligent transportation, video surveillance and robot environment awareness. As a core part of object detection and classification, deep learning has achieved great success in this area, but there are still some problems that make it become a challenging task, such as the complexity of image scene, the non-uniform of image shooting angle, object occlusion, and different postures of the same object or small size object.

For object detection and classification, the traditional machine learning method basically exists four stages: sliding window, features extraction, features selection and features classification. The heated research area are features extraction (How to enhance the ability of expression and anti-deformation ability), and features classification (How

Y. Zhao et al. (Eds.): ICIG 2017, Part I, LNCS 10666, pp. 374–385, 2017.
https://doi.org/10.1007/978-3-319-71607-7_33

to improve the accuracy and speed of the classifier). Researchers have proposed various of features and classifiers, there are some representative features (Haar [1], HOG [2], SIFT [3], SURF [4], etc.) and classifiers (Adaboost [5], SVM [6], DPM [7], etc.).

The traditional object detection method uses the characteristics of manual design, and the accuracy of traditional object detection can not meet the actual requirements even with the best non-linear classifier for feature classification. There are three shortcomings in the designing of characteristics: (a) Hand-crafted features are low-level features, which lack of expression of the object. (b) The separability of designed features is poor, which will result in a higher classification error rate. (c) It is difficult to choose a single feature applied to multi-category datasets.

In order to extract better features, Hinton presented Deep Learning [8] in 2006, the using of deep neural network from a large number of data can automatically learn high-level features. Compared with the hand-crafted features, the learning features of deep learning is richer, and the ability of expression is stronger. With continuous development in Deep Learning, the researchers have found that the accuracy based on CNN for objection detection can be greatly improved. Not only the convolution neural network can extract high-level features and improve the expression of features, but also combine feature extraction, feature selection and feature classification into the same model. In training by end-to-end, function optimization from the overall can enhance the separability of features. Especially in the past three years, Deep Learning has become more popular in the major pattern recognition competition, and achieved better and better performance, speed and accuracy have been greatly improved. This paper has three main contributions: (1) Proposed a deep but lightweight network model. (2) Adapted the multi-scale structure that can learn both global and local parts features, and then combine them to a new feature which has better ability to express. (3) The features fusion and multi-scale structure are added to the pre-trained VGG16 [9] model. The experimental results shows that the proposed method achieved better performance than original VGG16 model.

The rest of this paper is organized as follows. In Sect. 2, we review some related works. Section 3 introduces details of the designed network model, and Sect. 4 is presentation of the experimental results and evaluation. Finally, we conclude our work and arrange the future work in Sect. 5.

2 Related Work

Object detection can be divided into two categories, one is the early traditional machine learning methods, the other is the rise of Deep Learning in recent years. In this section, we generalize the development of these two methods.

2.1 Traditional Machine Learning

In 2004, Viola and Jones [1] proposed a new feature named Haar-like with cascade Adaboost classifier for face detection, it shows a great speed advantage compared with other methods at the same period. Therefore it also attracted many researchers in the feature design, cascade structure, boosting algorithm three aspects of in-depth research at the same time. Next year, Dalal and Triggs [2] proposed a local image texture called Histograms of Oriented Gradient (HOG), and combined it with Support Vector Machine (SVM) for pedestrian detection. With the development of HOG, Deformable Parts Model (DPM) [7] appeared, and also won the championship for three consecutive years at The Pascal Visual Object Classes (VOC) Challenge. Due to the fact that DPM considers well for local and global relationships, it has got higher detection accuracy and better performance. Although the above methods have achieved great performance, their development are limited by the limitations of hand-crafted design features and redundant time caused by sliding window.

2.2 Current Deep Learning

In 1998, Lécun et al. [10] proposed famous LeNet-5 model. It includes convolution layer, Relu layer, polling layer and the final innerproduct layer, and these layers have been still used, the network is also considered to be the first true sense of the convolution neural network. In 2012, Krizhevsky et al. [11] proposed AlexNet model, and have got lower ten percentage points than the previous year champion in ImageNet Large Scale Visual Recognition Competition (ILSVRC). This year is called the turning point of Deep Learning, marking the Deep Learning to take off. With the development of Deep Learning, some famous model like ZF [12], VGG [9], GoogleNet [13], and ResNet [14] are proposed.

In past three years, Deep Learning has got rapid development. Li et al. [15] proposed a kind of cascade convolution neural network named Cascade CNN. It contains six independent networks, three for the classification of the network, the other three for the bounding box regression. Cascading ideas can combine weak classifiers for higher accuracy, but the 6 networks of this paper are separated and can not be trained by end-to-end. So Qin et al. [16] proposed a joint training cascade convolution neural network for face detection, it has maintained the advantages of cascade and trained by end-to-end. In [17], Can and Fan proposed a multi-scale network named MS-CNN, it can detect different size objects at the same time. GoogleNet [13] uses Inception structure to make the network deeper, and the training parameters less. Ren et al. [18] proposed a network based on region proposal network (RPN) called Faster R-CNN, it decomposes the object detection problem into two subproblems. Firstly, the RPN network generates proposal bounding boxes, and uses these bounding boxes as input to the R-CNN. Because the RPN and R-CNN networks share the convolution feature, so the detection time is reduced and the detection accuracy is higher. Although RPN can reduce the detection time, the time is still too long. Aiming at this problem, YOLO [19] is an approach proposed by Redmon and Divvala. It removes the RPN network, can further reduce the detection time, but reduce the accuracy a little. On the basis of Faster R-CNN and YOLO framework, many classical methods are proposed by related researchers, such as FCN [20], PVANET [21], SSD [22] and YOLO9000 [23]. It is worth mentioning that our work is also based on the Faster R-CNN framework.

3 The Proposed Scalable Object Detection Method

In this section, we present the details of the proposed Scalable object detection method. Firstly, we describe the overall framework, next elaborate the feature fusion part of the pre-training model, and then expound the multi-scale structure. Finally, we present the training details.

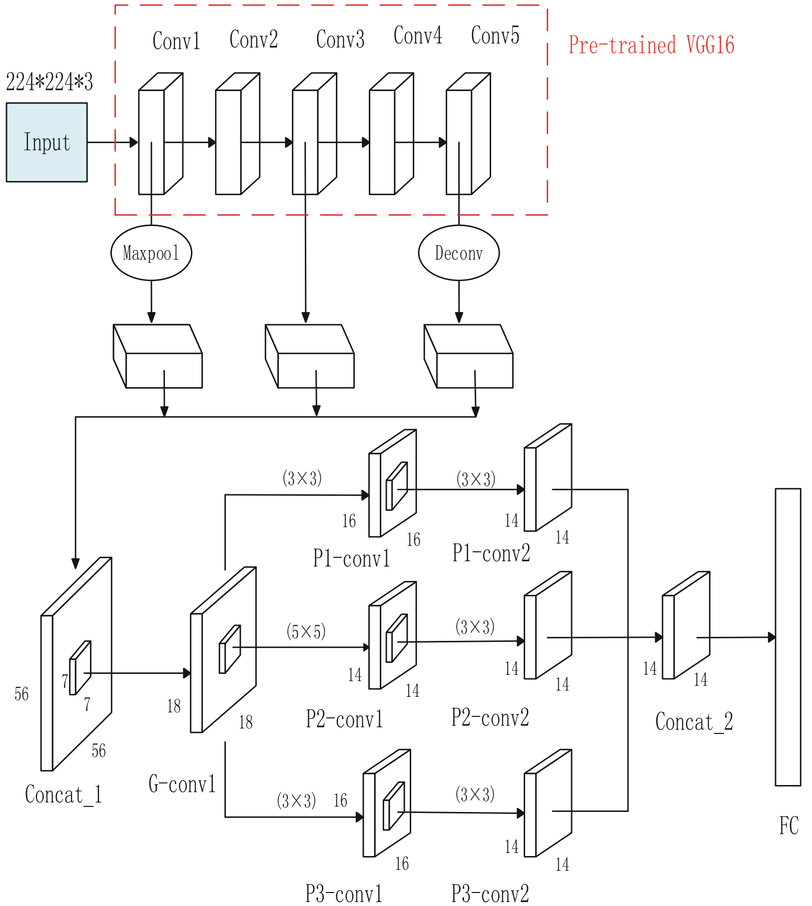


Fig. 1. Scalable object detection architecture. The network takes an input image of size 224 × 224, (1) combine the downsampling of Conv1, Conv3 and upsampling of Conv5 feature maps of pre-trained VGG16 model to carry out Concat_1, (2) behind the Concat_1, there is a global convolution name G-Conv1, (3) and then divided into three equal local parts named as Pi-Conv1 (i = 1, 2, 3), finally combine the Pi-Conv2 (i = 1, 2, 3) to get the Concat_2

3.1 The Overall Framework

The proposed scalable object detection architecture is showed in Fig. 1, and the details of the parameters of the network are given in Table 1. Initially, a 224 × 224 image is forwarded through the convolutional layers of pre-trained VGG16, and the features maps are produced. We aggregate hierarchical feature maps and then compress them into a uniform space, namely Concat_1. There is a global convolution with the kernel size of 7 × 7 to get global features, and a cascaded multi-scale structure consists of three parts for extracting local features, we combine the three local part feature maps to get the layer Concat_2. Finally, the innerproduct layer outputs detection classification results. Besides, each convolution layer is followed by a normalizing layer using local response normalization (LRN) and RELU layer.

Table 1. Detail parameters of the network

Name	Type	Kernel size	Stride/pad	Output
Conv1_1	Convolution	3 × 3	1/1	224 × 224 × 64
Conv1_2	Convolution	3 × 3	1/1	224 × 224 × 64
Pool1	Maxpool	2 × 2	2/0	112 × 112 × 64
Conv2_1	Convolution	3 × 3	1/1	112 × 112 × 128
Conv2_2	Convolution	3 × 3	1/1	112 × 112 × 128
Pool2	Maxpool	2 × 2	2/0	56 × 56 × 256
Conv3_1	Convolution	3 × 3	1/1	56 × 56 × 256
Conv3_2	Convolution	3 × 3	1/1	56 × 56 × 256
Conv3_3	Convolution	3 × 3	1/1	56 × 56 × 256
Pool3	Maxpool	2 × 2	2/0	28 × 28 × 512
Conv4_1	Convolution	3 × 3	1/1	28 × 28 × 512
Conv4_2	Convolution	3 × 3	1/1	28 × 28 × 512
Conv4_3	Convolution	3 × 3	1/1	28 × 28 × 512
Pool4	Maxpool	2 × 2	2/0	14 × 14 × 512
Conv5_1	Convolution	3 × 3	1/1	14 × 14 × 512
Conv5_2	Convolution	3 × 3	1/1	14 × 14 × 512
Conv5_3	Convolution	3 × 3	1/1	14 × 14 × 512
Down	Maxpool	4 × 4	4/0	56 × 56 × 128
Up	Deconvolution	4 × 4	4/0	56 × 56 × 128
Concat_1	Concat			56 × 56 × 512
G-Conv1	Convolution	7 × 7	3/1	18 × 18 × 512
P1-Conv1	Convolution	3 × 3	1/0	16 × 16 × 128
P1-Conv2	Convolution	3 × 3	1/0	14 × 14 × 128
P2-Conv1	Convolution	5 × 5	1/0	14 × 14 × 256
P2-Conv2	Convolution	3 × 3	1/1	14 × 14 × 256
P3-Conv1	Convolution	3 × 3	1/0	16 × 16 × 128
P3-Conv2	Convolution	3 × 3	1/0	14 × 14 × 128
Concat_2	Concat			14 × 14 × 512

3.2 The Features Fusion Structure

We initialize the parameters of Conv1 to Conv5 layers according to the pre-trained model VGG16. Because of subsampling and pooling operations, these feature maps are not in the same dimension. In order to combine different levels of feature maps, we have different sampling methods for different layers. A max pooling layer is added on Conv1 to get its downsampling, a deconvolution layer is added on Conv5 to carry out its upsampling. It makes them and Conv3 into a unified space, and finally combines them to generate Concat_1. But why is Conv1, Conv3 and Conv5, because their characteristics are the largest different. If the feature difference is not big, the effect of fusion will be reduced.

The lower feature maps are the details of the information, it is conducive for bounding box regression. And the higher feature maps are semantic information, which is good for classification. When we combine these two type features together, we can get better performance. The experimental results will be a good proof, so it is effective.

3.3 Multi-scale Structure

There is a global convolution on Concat_1 layer named G-Conv1 with the kernel size of 7×7, because different sizes of the convolution of the kernel field is not the same, the characteristics of the extraction is also not the same. The kernel of size 7×7 can extract global features, and it is divided into three equal local convolution parts. According to the Inception structure, the network has kernel with size 5×5 and 3×3, each different parts is designed to learn different local features. While getting the local feature maps Pi-Conv2 ($i = 1, 2, 3$), we combine the three part feature maps to get the concatenation layer Concat_2. So we can obtain both global and local features at the same time.

3.4 Training Details

- **Data augmentation:** Data augmentation is an indispensable technique in Deep Learning, it can manually increase the training data, and effectively inhibit the over-fitting. To apply data augmentation, we resize the shorter side to 600, and do the same as the short side of the scale operation on long side. Then we randomly crop a small patch 224×224 around objects from the whole image, and each sample is horizontally flipped.
- **Faster R-CNN:** Faster R-CNN combines the region proposal network and the detection network into a unified network, including two independent networks, one is RPN, the other one is R-CNN. RPN is used to predict the region proposal of input image with the three scales (128, 256, 512) and three kinds of aspect ratio (1:1, 1:2, 2:1), the mechanism of mapping is called anchor, each convolution produces 9 anchor. IOU (Intersection-over-union) of these achor and ground-truth is less than 0.3 as negative (background) and greater than 0.7 as positive (foreground). If it does not belong to the above, the proposal bounding box will be lost. The remaining bounding

boxes are used as input to the R-CNN, and the two networks share the convolution feature.

- **Fine-Tune:** The pre-trained VGG16 is used to initialize the parameters of Conv1 to Conv5 layers, and the learning rate is set to 0. So we can reduce a lot of training parameters. The rest of the convolution layers are initialized with Xaiver, and set the bias terms to 0. The last innerproduct layer layers is randomly initialized with Gaussian distributions with the standard deviations of 0.01 and 0.001, and also set the bias terms to 0.
- **SGD parameters:** We set global learning rate 0.001. The RPN and RCNN both have 40000 iterations, after 30000 iterations, we lower the learning rate to 0.0001 to train more iterations. Following standard practice, we use a momentum term with weight 0.9 and weight decay factor of 0.0005.

4 Experiments and Evaluation

In our experiments, The proposed method is evaluated on VOC2007, VOC2012 and KITTI datasets. The PASCAL Visual Object Classes Challenge is well known in the field of pattern recognition competitions, the VOC dataset has also become a standard dataset for object detection and classification, so it is shown that the VOC dataset can well explain the advantage and disadvantage of our method. Compared with the VOC dataset, KITTI has more small objects, occlusion situation is serious and the shooting angle is different. Experimental results also have proved that the performance on VOC is better, some detection examples of different datasets are showed in Fig. 2. Our experimental environment is NVIDA GTX1070 with Caffe, because of the limitation of experimental environment, all our experimental results are lower than original paper. But it does not affect the comparison results, it can still explain the results.

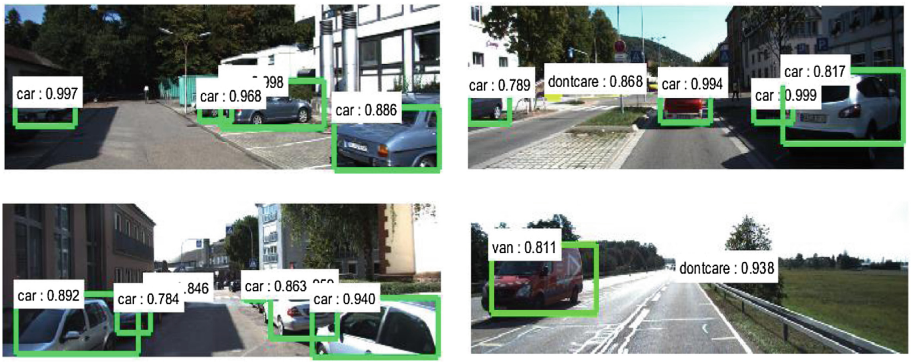
4.1 Datasets

- **VOC2007:** VOC2007 is a dataset containing 20 categories. Images are from our daily life scenes; Image size is around 500×375. It includes a total of 9963 images, 5011 training images and 4952 test pictures, 24640 annotated objects.
- **VOC2012:** Compared with VOC2007, occlusion flag is added to annotations and action classification is presented, the number of images increased to 11530, including 27450 annotated objects.
- **KITTI:** KITTI is a vehicle pedestrian dataset containing a total of 9 categories, including 7481 training images and 7518 test images, image size is around 1250×375.

4.2 VOC2007 and VOC2012 Results

Loss, accuracy and precision are three important indicators in the field of object detection and classification. The loss value can reflect whether the training situation of the model is stable, accuracy reflect the ability to judge the whole of the model, include both

(a) VOC2007+VOC2012



(b) KITTI

Fig. 2. Results on different datasets. (a) VOC2007 + VOC2012, (b) KITTI Datasets

positive and negative samples. And precision only reacts to the ability of the model to judge the positive samples. The Eqs. (1) and (2) are mathematical expressions of accuracy and precision. TP and FP respectively mean True Positive and False Positive, TN and FN respectively mean True Negative and False Negative. We use these three indicators to evaluate the experiment, and do it also in KITTI.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Figure 3 shows the comparisons of loss and accuracy, from this picture we can see that the loss of our method is lower while accuracy is higher. Besides, when we only add the features fusion structure (Faster Rcnn + Fusion) or multi-scale structure (Faster Rcnn + MS), the loss also lower than original Faster R-CNN with VGG16, and the accuracy is higher. It indicates that the features fusion and multi-scale structure which we add are valid. Table 2 shows our results compared with other methods in average precision (AP) and Frames Per Second (FPS) values. Because our work is based on the Faster R-CNN framework, the FPS of our method is lowest with 5, but we enhance the mean AP (mAP). Compared with other methods, our mAP is higher than YOLO but little lower than SSD500. And for single classes, our AP value is higher or lower. The reason is that different network structures perform differently for different object scene, such as different object size and pose. We combine different levels of feature maps, and use different convolution kernels in the multi-scale structure, so we have got a higher mAP on the whole. However, there is no single-scale features targeted for individual special classes, AP value may be lower. In general speaking, we have achieved better performance than original VGG16 model, and keep the speed at the same time.

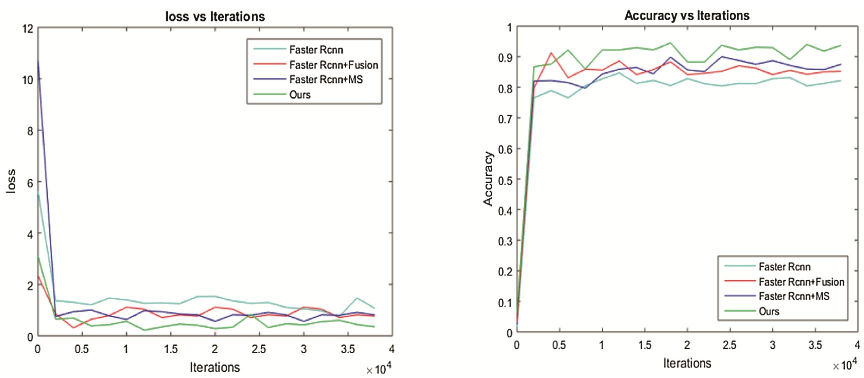


Fig. 3. Loss and accuracy on VOC2007 + VOC2012

4.3 KITTI Results

Just as we can see in the Fig. 4, we have got lower loss and higher accuracy the same as VOC2007 and VOC2012. It indicates that the feature fusion and multi-scale structure which we add is valid once again. Table 3 shows the results of mAP and FPS, we have got a higher mAP than Faster R-CNN and YOLO, but little lower than SSD500. Compared with the results on VOC2007 and voc2012, all the mAP are lower and FPS are identical. The reason is that there is more small size object in KITTI dataset, and occlusion situation is serious.

Table 2. Results on VOC2007 + VOC2012 (with IOU = 0.7)

Method	Faster R-CNN	YOLO	SSD500	HyperNet	Proposed
Tv	52.7	45.3	68.3	61.6	63.5
Bird	67.5	52.7	71.5	49.5	85.3
Boat	46.1	33.9	54.6	46.3	42.5
Bottle	42.2	19.4	47.2	48.8	52.4
Bus	66.7	62.6	77.4	72.2	61.2
Table	49.1	42.8	54.8	51.3	44.5
Cat	73.6	71.5	85.2	64.4	83.1
Chair	43.9	35.9	52.0	32.7	58.3
Cow	68.2	54.4	75.4	60.6	64.9
Car	65.4	51.2	77.1	66.9	78.7
Dog	74.3	71.3	83.7	59.1	87.6
Horse	73.7	66.1	80.3	63.8	85.8
Aero	74.9	71.8	84.6	61.8	70.2
Plant	36.5	24.4	44.9	26.7	49.1
Person	67.4	58.2	80.6	55.9	79.4
Sheep	62.3	46.7	72.7	62.4	76.8
Sofa	54.4	48.5	61.5	57.1	69.3
Train	73.6	67.1	82.9	62.2	68.9
mbike	70.8	64.6	81.2	54.6	84.4
Bike	71.3	62.3	78.1	62.9	81.6
mAP	61.7	52.5	70.7	56.0	69.4
FPS	5	32	14	5	5

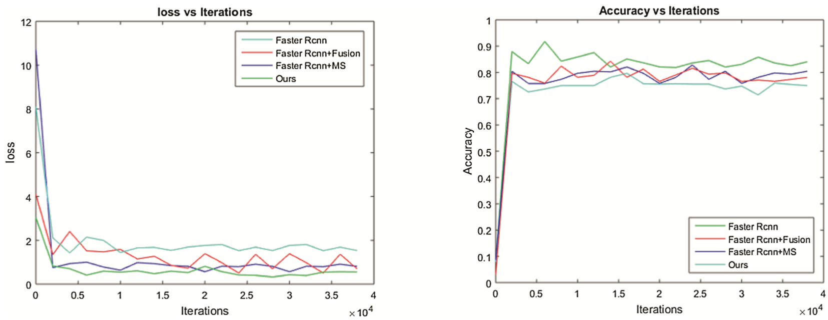


Fig. 4. Loss and accuracy on KITTI

Table 3. Results on KITTI (with IOU = 0.7)

Method	Faster R-CNN	YOLO	SSD500	HyperNet	Proposed
Tram	49.1	42.6	57.3	45.5	58.9
Car	69.4	60.2	74.5	64.7	78.3
Person_sitting	45.2	37.9	56.1	42.3	41.8
Pedestrian	63.5	55.8	70.2	60.2	60.6
Truck	57.4	49.3	61.6	54.3	66.7
Cyclist	64.9	57.2	73.3	62.1	62.3
Dontcare	56.3	48.4	64.1	51.9	63.8
Misc	38.5	34.6	43.8	39.5	51.2
Van	45.8	36.5	49.7	41.6	54.1
mAP	54.5	47.0	61.2	51.3	59.7
FPS	5	32	14	5	5

5 Conclusion

In this paper, we proposed a unified multi-scale network with features fusion, through combining different levels of feature maps, we can obtain advantage of both high and low-level maps, multi-scale structure can detect object of different sizes. Experimental results show that we have got a higher mAP as a whole on the VOC2007, VOC2012 and KITTI datasets, and maintained the original speed. We also analyzed the experimental results, compared with other mainstream methods, it illustrates the advantages and disadvantages of our approach. In the future work, our main focus is how to further improve the detection speed and achieve real-time performance, it is better to enhance the mAP at the same time.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (No. 61403054), and the Fundamental and Frontier Research Project of Chongqing under Grant No. cstc2014jcyjA40001.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. 511–518. IEEE Xplore (2001)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp. 886–893. IEEE (2005)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
4. Li, J., Zhang, Y.: Learning surf cascade for fast and accurate object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3468–3475 (2013)

5. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59119-2_166
6. Vapnik, V.: The Nature of Statistical Learning Theory, pp. 988–999. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-3264-1
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1627–1645 (2010)
8. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Computer Science (2014)
10. Lécun, Y., Bottou, L., Bengio, Y.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105. Curran Associates Inc. (2012)
12. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
13. Szegedy, C., Liu, W., Jia, Y.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, pp. 1–9 (2014)
14. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
15. Li, H., Lin, Z., Shen, X.: A convolutional neural network cascade for face detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5325–5334 (2015)
16. Qin, H., Yan, J., Li, X.: Joint training of cascaded CNN for face detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3456–3465. IEEE Computer Society (2016)
17. Cai, Z., Fan, Q., Feris, R.S.: A unified multi-scale deep convolutional neural network for fast object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
18. Ren, S., He, K., Girshick, R.: Faster R-CNN: towards real-time object detection with region proposal networks. In: IEEE Transactions on Pattern Analysis & Machine Intelligence, p. 1 (2015)
19. Redmon, J., Divvala, S., Girshick, R.: You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, pp. 779–788 (2015)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
21. Kim, K.H., Hong, S., Roh, B.: PVANET: deep but lightweight neural networks for real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
23. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)