2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Multi-scale Fusion with Context-aware Network for Object Detection

Hanyuan Wang, Jie Xu, Linke Li, Ye Tian, Du Xu, Shizhong Xu*
School of Information and Communication Engineering
University of Electronic Science and Technology of China
Chengdu, China
Email: {xuj, xudu}@uestc.edu.cn, {xsz.uestc}@gmail.com

***Abstract*—Almost all of the state-of-the-art object detectors employ convolutional neural network (CNN) to extract feature. However, how to fully utilize spatial information is a challenge. In this paper, we propose an effective framework for object detection. Our motivation is that multi-scale representation and context are extremely important for object detection. For multi-scale representation, our mothed combines hierarchical feature maps to a fusion map, which has abundant spatial information and high-level semantics. For context, we exploit spatial information by stacking multi-region feature maps. The network is learned end-to-end, by minimize an objective function. Our network achieves competitive results, 75.9% mAP on PASCAL VOC 2007, 72.0% mAP on PASCAL VOC 2012 and 23.2% mAP on MS COCO. The speed of the network is 10 fps. Our studies demonstrate that multi-scale representation and context can further improve performance of object detection.**

I. Introduction

Object detection is an important application direction in computer vision. There are three major key points to evaluate the performance of object detectors: detection accuracy, detection efficiency and localization accuracy. Based on the three major keys, the region-based methods [1] [2] [3] have achieved good results by improving detection accuracy. The region-free methods [4] [5] have the advantages of improve detection efficiency. Method [6] produced precise bounding box (bbox) to locate informative positive samples. These methods have promoted the development of object detection.

Among them, one of the most remarkable methods is R-CNN [1], which is the foundation of the region-based methods. R-CNN performs CNN-based detection with proposals generated algorithm, such as Selective Search [7] and Edge Boxes [8]. However, in terms of detection speed, these algorithms are still suffering from unnecessary computational cost. To improve the accuracy and efficiency of R-CNN, Fast R-CNN [2] has proposed ROI-pooling layer. ROI-pooling layer is the extension of Spatial Pyramid Pooling Network (SPP-Net) [9], which can generate a fixed-length representation regardless of image size. The step of region proposal generation consumes much time by Selective Search. Therefore, Faster R-CNN [3] proposed Region Proposal Network (RPN), a network to generate region proposals by sharing convolutional layer.

These methods have effectively improved the performance of object detectors. However, one of the issue is that the high-level semantic feature is too coarse to degrade the performance of localization, so that these methods are weak in detecting small objects. In fact, [2] [3] have tried to solve this problem by up-sampling the input image, but it increases computational costs. Another issue is that these works only use information in the object's region of interest (ROI), it is not enough to distinguish object categories. There is still a lot of information to mine, such as contextual information. Sometimes, a larger surrounding region may contain more important part of the ground truth boxes. In a word, these works does not consider multi-scale representation and contextual information, which are beneficial for localizing small objects and improving detection accuracy.

There are two motivations for our research. The first one is the difference between high-level feature and low-level feature. Due to the spatial sampling, the high-level feature are coarser but semantically stronger, the low-level feature has lower semantic but can locate small objects accurately. Therefore, we use multi-scale feature to capture fine-grained detail. Another motivation is that contextual information are important for accurate visual recognition. When a candidate region cover a part of the ground-truth boxes, regions surrounding the ROI are useful. Therefore, it is necessary to distinguish object categories from finding a contextual visual pattern.

In this paper, we aim to enhance object detection performance by expanding feature extraction methods. We propose a novel object detection framework, which exploits extra information to enrich feature map. Our main contributes are as follows:

- We proposed an effective framework to improve the performance of object detection, especially for small object.
- On the PASCAL VOC object detection challenges, we obtain satisfactory results: 75.9% mAP on VOC 2007, 72.0% mAP on VOC 2012 and 23.3% on MS COCO.
- We conduct a series of experiments and analyze separately. For example, we compare the effect of combining different layers, two normalized methods and diverse L2 normalization scale.
- Our network is efficient in time. With GeForce GTX 1080 Ti GPU, the total feed-forward speed is 10 fps.

This work was supported by National Key Research and Development Program (Grant No. 2016YFB0800105), Sichuan Province Scientific and Technological Support Project (Grant Nos. 2016GZ0093, 2018GZ0255), the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2015J009).

978-1-5386-3787-6/18/$31.00 ©2018 IEEE

II. Related Work

A. Region-based object detectors.

In recent years, convolutional neural networks (CNNs) has made an enormous contribution in object detection. At present, CNN-based object detectors could be divided into two main categories: the region-based proposals detectors [1] [2] [3] [10] and the region-free proposals detectors [4] [11]. The region-based proposals detector R-CNN replaces the manual methods [12] [13] with CNN models in feature extraction stage. Fast R-CNN improves detection efficiency by ROI-pooling layer. RPN [3] generated more precise proposals than Selective Search. Our experiments are conducted with Faster R-CNN, which is an end-to-end region-based object detector.

B. Multi-scale feature representation

Recently, it is proved that multi-scale feature representation is essential in many significant works. Hypercolumn [14] use hypercolumns as pixel descriptors for object localization and segmentation. HyperNet [10] aggregates hierarchical feature maps and compresses them into a uniform space. ION [15] use ROI-Pooling layer [2] to extract fixed-size descriptors and concatenate them. PVANet [16] presents a lightweight network based on proper fusion of coarse-to-fine CNN features. Another combined strategy is combining features from multiple layers after prediction. For example, in order to perform detection at multiple scales, SSD [11] applies separate predictors to multiple feature maps. RON [17] predicts objects at different layers with reverse connection.

Different from these works, our work builds a deep fusion model that combine multi-scale feature by element-wise sum. More importantly, we add a new convolution layer 6 to exploit high feature. It is unnecessary to add a 1x1 convolution layer which brings computational cost.

C. Contextual information

In recent years, a lot of works have been made to improve the accuracy by contextual information. To gather contextual information, Parikh *et al.* [18] uses the relative location and relative scale. Cinbis *et al.* [19] uses set-based classification. Bell *et al.* [15] explores the use of spatial recurrent neural networks (RNNs). Cai *et al.* [20] embeds context from multiple regions. Chu *et al.* [21] considers the relationships among objects. Zeng *et al.* [22] proposes a bi-directional network structure to passes contextual visual messages in both directions. Mottaghi *et al.* [23] studies the role of context in existing state-of-the-art detection and segmentation approaches.

Similar to [20] [22], our method uses ROI-pooling to obtain contextual information with different support regions. However, our scheme is not simply to concatenate contextual feature maps along the channel direction, as in [15] [22]. Instead, our scheme is to sum each feature descriptor by element-wise.

III. Network Architecture

In this section, we introduction our framework and describe design details. We basically follow the Faster R-CNN architecture but has some modifications specialized. Our approach consists of three stages, as illustrated in Fig. 1. Firstly, our network takes an entire image into the convolutional layers, and produces hierarchical activation maps. Secondly, a fusion map is generated by combining layer 4, 5 and 6. Then, heaping up multi-region proposals to produce a feature descriptor. Finally, Two Fully-connected (Fc) layers are used to predict class probabilities (Softmax) and regress bounding box (Bbox reg).

A. Multi-scale Feature Fusion

In feature extraction network, hierarchical feature maps have different characteristics. This paper takes a unique way to use these feature maps. For example, feature map comes from layer 4 is more beneficial to small objects location and detection. On the contrary, feature map comes from layer 6 is better for detecting large objects because of the high-level semantic. Multi-scale representation is a perfect combination of the two advantages.

Our design method is similar to PVANet and HyperNet, which combine coarse, high-level layer with fine, low-level layer. However, we combine layer 4, 5 and 6 to enhance semantic information. The convolution layer 6 is produced by pooling layer 5, which feature map size is 1/32 of the input size, both in width and height. The scale of the layer 4, 6 are twice and four times of the layer 5 respectively. To combine multi-level maps at the same resolution, we add a max pooling layer behind the layer 4, and a deconvolution layer is applied to layer 6 to achieve up-sampling. To improve the standardization of model, we normalize multiple maps using L2 normalize. Because different layers may have different amplitudes, L2 normalize is useful for feature combination. Then, we fuse each output by element-wise sum. By now, the fusion feature map size feed to ROI-pooling layer is 1/16 of the input size.

It is proved that pre-train a deep CNN model for initializing basic layers is effective. Our method starts from the pre-trained VGG-16 [24] network. In addition to matching the original shape, the shape of the final feature map must be $512 \times 7 \times 7$, which is the standard shape to feed into fully-connected layer (fc6). Therefore, the fusion feature map can feed into the ROI-pooling layer directly, so that it retains more details.

B. Context-aware Information

Context feature has been computed by a recurrent neural network in [15]. Zeng et al. [22] uses a gated bi-directional CNN to pass messages between features from different contextual regions. Inspired by [20], we focus on context from multiple regions. But we extract contextual information from another finer feature map (fusion map).

As is shown in Fig. 1, we heap up object region feature and context region feature after ROI-pooling. Firstly, the ROI-pooling layer is used to generate two fixed-length feature descriptor of size $7 \times 7 \times 512$. The object region (grey cube) and the context region (blue cube) are provided by the region generating layer (in section III-C). The context region is 1.5 times of the object region.

Different from [20], we concatenate each pooled feature descriptor by element-wise operations. It is no need to reduce the dimension with a cumbersome 1×1 convolution layer, thus the computational costs and redundant parameters have been reduced.

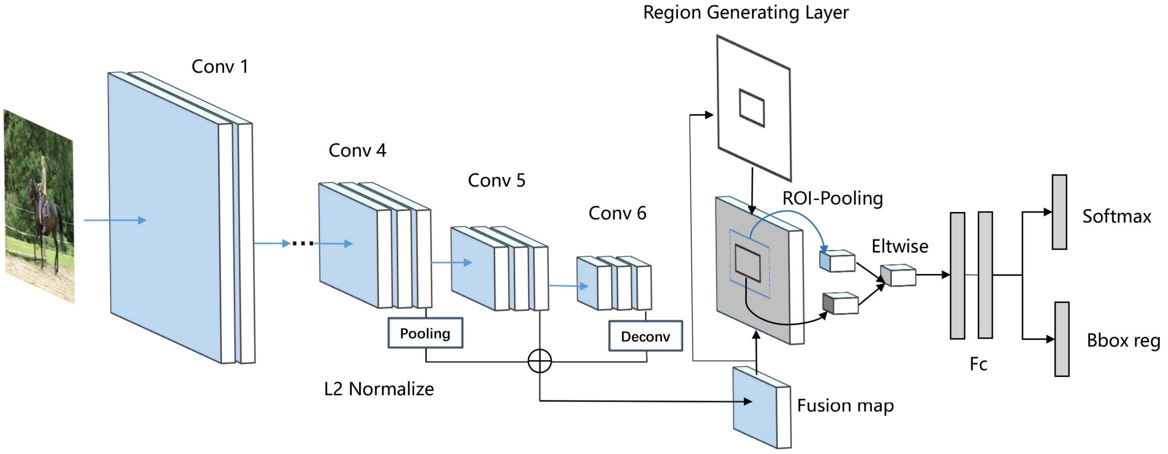

Fig. 1. Overview of our model (see Section III for details). A CNN is used to compute fusion map from the input image. Extracted fusion map is then fed to the ROI-pooling layer, which generates contextual feature descriptor. Two fully-connected (Fc) layers process each descriptor and produce two outputs: a class prediction ("Softmax"), and an adjustment to the bounding box ("Bbox reg").

C. *Region Generating Layer*

Region generating layer takes a feature map as input, and outputs a set of region proposals. The input feature map is covered by a sliding-window. At each sliding-window location, region proposals are predicted by anchor boxes of multiple scales and aspect ratios. In this paper, we have found the Region Proposal Network (RPN) is too simple to generate various boxes. Thanks to this observation, we make the fusion map as the input of RPN, and set four scales {80,144,156,512} with three aspect ratios {1:1, 2:1, 1:2} to cover objects of different size.

After region generating layer, some region proposals are overlap with each other. To delete duplication, we adopt non-maximum suppression (NMS) to these region proposals. For a region proposal, NMS suppress others intersection-over-union (IoU) higher than the threshold, so that redundancy can be reduced. We set the overlap threshold to 0.7, and use the top 300 region proposals for detection.

D. *Joint Training*

In this paper, we adopt an end-to-end training to jointly optimize the models. The region proposal network and the detection network are merged into one network.

In each iteration of training, region generating layer generates a set of region proposals to predict classification score and regress box locations. This process is the pre-compute of forward propagation. For generation proposal region, we give positive label to a box which IoU higher than 0.7, or which has the highest IoU with a ground-truth box. On the contrary, negative label is given to the box which IoU lower than 0.3. In backward propagation, gradient signal is come from the loss of region proposal generation and detection. Therefore, we minimize the multi-task loss function defined in Faster R-CNN:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*), \quad (1)$$

where p_i is the predicted probability of anchor i is being an object. The value of p_i^* indicates the ground truth label of anchor i, $p_i^* = 1$ denote the anchor is positive and $p_i^* = 0$ denote the anchor is negative. So, $L_{cls}(p_i,p_i^*)$ is log loss function for classification, and $L_{reg}(t_i,t_i^*)$ is regression loss function as following:

$$L_{cls}(p_i,p_i^*) = -\log[p_i^* p_i + (1-p_i^*)(1-p_i)], \quad (2)$$

$$L_{reg}(t_i,t_i^*) = R(t_i - t_i^*), \quad (3)$$

here, R is the robust loss function smooth L1 [2]:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \quad (4)$$

Note that just the positive anchors ($p_i^* = 1$) being object has the regression loss. The outputs of the *cls* and *reg* layers consist of $\{p_i\}$ and $\{t_i\}$ respectively.

IV. Experiment

We perform experiments on the 20 category detection dataset: PASCAL VOC [25] and COCO[26]. For PASCAL VOC, all models are trained on the joint set of PASCAL VOC 2007 trainval set and PASCAL VOC 2012 trainval set ("07+12"). For MS COCO, we trained models on the train set. Average Precision (mAP) is a measure of object detection accuracy.

A. *Experimental setup*

All of our models are built on Fast R-CNN framework and VGG-16 architecture. Our network is implemented by Caffe [27]. In each training iteration, images are resizing such that the shorter edges is 608 pixels, and max size of the longest side is 1120 pixels. For parameter solver, we adopt Stochastic Gradient Descent (SGD) to minimize objective function. The initial learning rate is set to 0.001, and decreased by a factor of 10 times after every 50,000 iterations.

TABLE I. DETECTION RESULTS ON PASCAL VOC 2007 TEST SET. LEGEND: M: MULTI-SCALE FEATURE FUSION, C: CONTEXT-AWARE INFORMATION

Method	M	C	Train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN			07+12	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 R-CNN			07+12	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			07+12	74.3	75.5	80.2	72.3	**66.3**	47.6	83.0	84.2	86.1	54.7	78.3	**73.9**	84.5	85.3	**82.6**	76.2	48.6	73.9	**76.0**	83.4	74.0
ION			07+12	75.6	79.2	83.1	**77.6**	65.6	54.9	**85.4**	85.1	87.0	54.4	80.6	73.8	**85.3**	82.2	82.2	74.4	47.1	75.8	72.7	**84.2**	**80.4**
Ours	√		07+12	74.4	**79.9**	78.8	72.3	63.0	**61.0**	81.8	86.7	87.2	56.6	80.1	67.0	84.2	83.8	77.4	78.4	45.1	76.2	73.2	81.1	73.8
Ours		√	07+12	75.0	78.5	79.6	75.0	64.4	60.9	82.2	86.7	**88.0**	56.7	**83.9**	68.3	83.4	85.0	78.6	78.2	43.7	**78.2**	70.2	82.6	75.8
Ours	√	√	07+12	**75.9**	79.7	**83.2**	75.4	64.1	59.4	82.6	**87.5**	85.4	**58.7**	83.4	72.5	84.7	**86.4**	77.1	**78.7**	**49.2**	76.7	75.6	83.7	74.7

TABLE II. DETECTION RESULTS ON PASCAL VOC 2012 TEST SET. LEGEND: M: MULTI-SCALE FEATURE FUSION, C: CONTEXT-AWARE INFORMATION

Method	M	C	Train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN			07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN			07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
SSD300			07++12	**72.4**	**85.6**	80.1	70.5	**57.6**	46.2	**79.4**	76.1	89.2	**53.0**	77.0	60.8	87.0	**83.1**	82.3	79.4	45.9	**75.9**	**69.5**	**81.9**	67.5
HyperNet			07++12	71.4	84.2	78.5	73.6	55.6	**53.7**	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	**83.3**	81.8	**48.6**	73.5	59.4	79.9	65.7
OHEM			07++12	71.9	83.0	**81.3**	72.5	55.6	49.0	78.9	74.7	**89.5**	52.3	75.0	**61.0**	87.9	80.9	82.4	76.3	47.1	72.5	67.3	80.6	**71.2**
Ours	√		07+12	70.4	84.7	79.0	72.7	52.4	52.1	76.2	78.1	87.2	47.3	74.6	51.1	86.6	80.0	81.6	82.1	47.3	74.8	57.0	78.9	64.9
Ours		√	07+12	71.9	84.0	80.6	**74.9**	56.2	52.6	78.4	78.3	88.6	50.4	**79.3**	54.3	**88.3**	82.4	81.5	82.3	46.2	74.0	57.9	80.6	66.9
Ours	√	√	07+12	**72.0**	**85.6**	79.3	72.6	56.9	52.9	79.2	78.1	87.6	51.6	77.1	57.3	87.1	81.1	82.3	**82.5**	**48.6**	74.8	59.1	80.9	65.0

We set weight decay to 0.0005 and momentum to 0.9, so that the learning rate is 0.001 for the first 50k mini-batches and 0.0001 for the next 20k. 128 ROIs were sampled from an image in each mini-batch. The weights of all new layers were initialized with "Xavier", so that it can automatically determine the scale of initialization based on the number of input neurons after concatenation.

All models are based on the same VGG-16 architecture pre-trained on the ImageNet classification task, and then fine-tuned on the detection dataset.

B. Result in PASCAL VOC

As shown in Table I, for the PASCAL VOC2007 detection task, our method achieves a mAP of 75.9%. Our method is 5.9% higher than Fast R-CNN, 2.7% higher than Faster R-CNN, 1.6% higher than SSD300 and 0.3% higher than ION. Unlike original detector, we get the final bounding box localization by weighted voting from boxes which IoU more than 0.5. The performance increases by 0.5 points when bounding box voting [28] is added.

We also evaluate our models on PASCAL VOC 2012 test set by submitting the results to the public evaluation server.[1] We use the same setting with VOC 2007 test set. In Table II, we compare our models against some state-of-the-art networks. Our mothed obtains a mAP of 72.0%, which is the most accurate for many categories.

C. Result in MS COCO

We test our model on MS COCO test-dev 2017 and get report from the public evaluation server.[2] In Table III, our model achieves 34.3% on test-dev score, which outperforms the baseline Faster R-CNN. We observe that the precision of ours model with 0.5:0.95 IOU is lower than ION, SSD300 and DSSD321, but the result for small area is comparable. Therefore, our method is more effective for small object detection. Note that DSSD321 base on the Residual-101 network, but our network base on VGG16 architecture.

D. Analysis For Multi-scale Feature Fusion

An important characteristic of our model is that it fuses coarse-to-fine information. So we conduct a series of experiments on the multi-scale feature fusion, and analyze how feature fusion affects the final performance. All results in this section are tested on PASCAL VOC 2007 test set.

Our model obtains significant gains in performance, but average precision alone could not explain our thoughts clearly. Hence, we compare the recall performance against Faster R-CNN on PASCAL VOC 2007 test set. We add online hard example mining (OHEM) [29] to multi-scale feature fusion model, and evaluate recall with different overlap thresholds *Ot*. As shown in Fig. 2, when *Ot* is 0.5 and 0.6, our model obtains significant gains in precision as recall increases. The reason is that our model can produce distinctive feature map, which feed into detection network. However the two curves have crossed at *Ot* is 0.7, our model also perform better than Faster R-CNN at higher *Ot*. Therefore, we can conclude that the multi-scale fusion feature works better than a single layer.

TABLE III. DETECTION RESULTS ON MS COCO TEST-DEV SET

Method	Train data	Avg. Precision, IoU:			Avg. Precision, Area:		
		0.5:0.95	0.5	0.75	S	M	L
Fast R-CNN	train	19.3	39.3	-	-	-	-
Faster R-CNN	trainval	21.9	42.7	-	-	-	-
ION	train	23.6	43.2	23.6	6.4	24.1	38.3
SSD300	trainval35k	25.1	43.1	25.8	6.6	25.9	41.4
DSSD321	trainval35k	28.0	46.1	29.2	7.4	28.1	47.6
Ours	trainval	23.2	44.9	21.7	9.3	26.7	32.3

[1] Anonymous URL: http://host.robots.ox.ac.uk:8080/eval/view_submission/14973/

[2] Anonymous URL: https://competitions.codalab.org/my/competition/submission/458443/stdout.txt

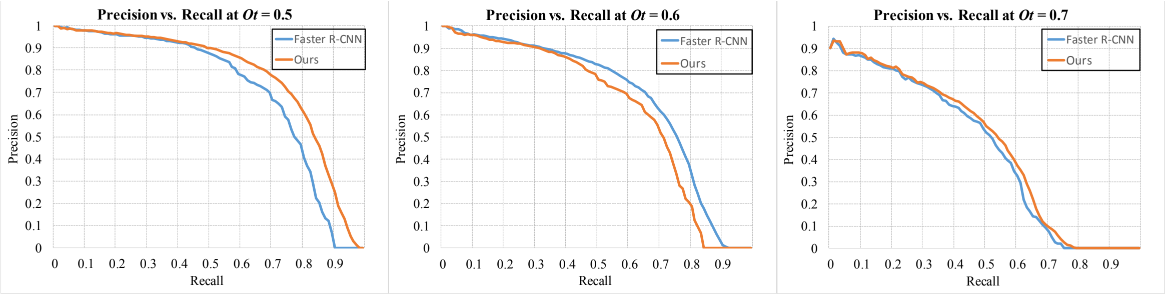

Fig. 2. Precision versus recall for multi-scale feature fusion at multiple overlap thresholds *Ot*.

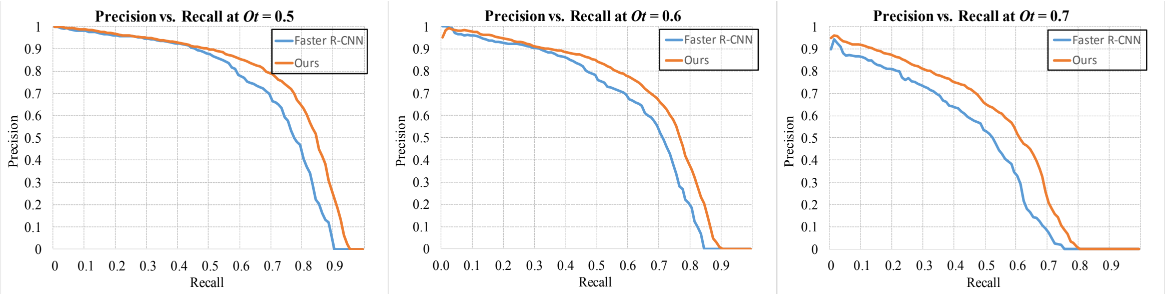

Fig. 3. Precision versus recall for context-aware at multiple overlap thresholds *Ot*.

Do Convolution Layer 6 Help? An important property of our method is fusing high-level features from higher convolution layer 6. Therefore, our detector can obtain better semantics in deep CNN models. However, does the convolution layer 6 really help? We did some experiments to discuss the issue. First of all, we train an object detector with single layer features (layer 5), as Faster R-CNN. Secondly, we fuse features come from layer 3, 4 and 5. Finally, we take another fusion strategy: combining layer 4, 5 and 6. Note that feature maps are normalized to the same resolution. The comparison results curve as shown in Fig. 4. As shown in Table IV, single layer 5 achieves a recall of 80% requires 1000 proposals; 143 proposals needed by combining layer 3,4,5; only 43 proposals needed by combining layer 4,5,6. Therefore, conclusion can be summarized: The new convolution layer 6 performs satisfactorily in feature fusion, which has better semantic than convolution layer 5.

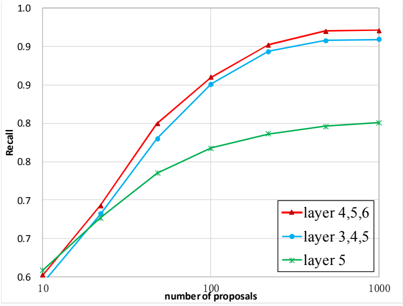

Fig. 4. Recall versus number of proposals for different layer fusion on the PASCAL VOC 2007 test set (with IoU = 0.5).

The role of L2 Normalize. Normalize is proved to be essential prior to fusing different layers in ParseNet [30]. As shown in Table V, we compare two normalized methods: L2 normalize and Local response normalization (LRN) [31]. LRN response normalization by a form of lateral inhibition, but it just normalizes locally. It is obvious that L2 normalize is more effective. After adding contextual information, we further compare the performance with different L2 normalization scales (Table VI), and obtain better performance at scale is 15 (mAP=75.9%).

TABLE IV. REGION PROPOSAL NUMBER NEEDED FOR DIFFERENT RECALL RATE (WITH IOU = 0.5)

Recall	Layer 5	Layer 3,4,5	Layer 4,5,6
60%	10	9	8
70%	36	25	22
80%	1000	59	43

TABLE V. DETECTION PERFORMANCE WITH DIFFERENT LAYER COMBINATION STRATEGIES AND NORMALIZATION METHODS

ROI-Pooling from：				Normalization Method	
L3	*L4*	*L5*	*L6*	*LRN*	*L2 Norm*
		√		70.0	
√	√	√		70.0	73.1
	√	√	√	67.3	**74.4**

TABLE VI. SCALE TO NORMALIZING FEATURE AMPLITUDE. METRIC: PASCAL VOC07 TEST MAP

L2 Normalization Scale	mAP
10	73.7
15	**75.9**
20	74.8

E. Analysis For Context-aware

As shown in Fig. 3, when adding contextual information, our model largely outperforms Faster R-CNN at the three different thresholds *Ot*, especially at *Ot* is 0.7. From the results, we have summarized three keys: (1) our model improves the performance of detection by embedding contextual information; (2)

comparing with the baseline, our mothed performs better at higher *Ot*; (3) the element-wise sum operation is useful to add contextual information.

F. The Relationship between Multi-scale Feature and Context

The multi-scale fusion feature and context are complementary in theory. The fusion feature not only has strong response, but also effective at improving detection for small object. Sometimes, candidate box may not be sufficient to distinguish object categories. Therefore, context can avoid the situation of misclassification for right candidate box. The fusion feature guarantees the depth of feature cube, and the context guarantees the width of feature cube.

V. Conclusion

The paper introduces a deep framework that exploits multi-scale feature and contextual information for object detection. For multi-scale feature, we add a convolution layer 6 to extract high-level features and fuse multi-scale feature maps after normalized and scaled. For contextual information, we combine multi-region descriptors after RoI-pooling layer. The network is trained end-to-end by optimizing a multi-task loss. To justify our design, we made a series of experiments and analyzed separately. For instance, the choice of layers combined, normalized method and L2 normalization scale. In addition, we analyzed the relationship between multi-scale representation and context based on experimental results. Our framework achieves the state-of-the-art results on the PASCAL VOC and MS COCO dataset, and makes an improvement in object detection performance, especially for small objects.

References

[1] R.Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,”in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580-587.

[2] R. Girshick, “Fast r-cnn,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, 2015, pp. 1440-1448.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2016.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Conference on ComputerVision and Pattern Recognition*, 2016.

[5] M. Najibi, M. Rastegari, and L. S. Davis, “G-cnn: an iterativegrid based object detector,”in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2369-2377.

[6] S. Gidaris and N. Komodakis, “Locnet: Improving localization accuracy for object detection,” in *Proceedingsof the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 789-798.

[7] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, 2013, pp. 154-171.

[8] Zitnick, C. Lawrence, and P. Dollár, “Edge Boxes: Locating Object Proposals from Edges,”in *European Conference on Computer Vision*, 2014, pp. 391-405.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,”in *European Conference on Computer Vision*, 2014, pp. 346-361.

[10] T. Kong, A. Yao, Y. Chen, and F. Sun, “Hypernet: Towards accurate region proposal generation and joint object detection,”in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 845-853.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C. Berg,“Ssd: Single shot multibox detector,” in *European Conference on Computer Vision*, 2016, pp. 21-37.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.

[13] Lowe, D. G, “Distinctive Image Features from Scale-Invariant Keypoints,” in *International Journal of Computer Vision*, 2004, pp. 91-110.

[14] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, “Hypercolumns for object segmentation and fine-grained localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447-456.

[15] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[16] K. He, S. Hong, B. Roh, Y. Cheon, M. Park,"PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection." *arXiv preprint arXiv:1608.08021*,2016

[17] T. Kong, F. Sun, A. Yao, H. Liuand M. Lu,“Ron: Reverseconnection with objectness prior networks for object detection,”in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] D. Parikh, CL. Zitnick and T. Chen,“Exploring tiny images: the roles of appearance and contextual information for machine and human object recognition,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2012, pp. 1978-1991.

[19] RG. Cinbis and S. Sclaroff, “Contextual object detection using set-based classification,” in *European Conference on Computer Vision*, 2012, pp. 43-57.

[20] Z. Cai, Q. Fan, R. S. Feris, et al. “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection,” in *European Conference on Computer Vision* ,2016, pp.354-370.

[21] W. Chu, D. Cai,“Deep Feature Based Contextual Model for Object Detection,” Neurocomputing, 2016.

[22] X. Zeng, W. Ouyang, B. Yang, J. Yan, X. Wang, “Gated Bi-directional CNN for Object Detection,” in *European Conference on Computer Vision*, 2016, pp. 354-369.

[23] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segmentation in the wild, ” in *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition*,2014, pp. 891-898.

[24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[25] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. “The pascal visual object classes challenge: A retrospective.*” International Journal of Computer Vision,*2015, pp .98–136.

[26] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, “Microsoft COCO: common objects in context,” in *European Conference on Computer Vision*, 2014, pp. 740-755.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093,* 2014.

[28] S. Gidaris, N. Komodakis. “Object detection via a multi-region & semantic segmentation-aware CNN model,” in *Proceedings of the International Conference on Computer Vision* ,2015, pp. 1134-1142.

[29] A. Shrivastava, A. Gupta, R. Girshick. “Training region-based object detectors with online hard example mining,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.,*2016, pp . 761-769.

[30] W. Liu, A. Rabinovich, and A. C. Berg. “ParseNet: Looking wider to see better. arXiv e-prints,” Computer Science, 2015.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012. pp. 1097-1105.