# Deep Position-Sensitive Network for Object Detection

Feng Xiao, Mengmeng Bai, Li Zhao and Defa Hu

***Abstract*—**Recently the deep network's ability of learning position-sensitive information is some insufficient in object detection. To improve the ability, we compare the performance of single position-sensitive score maps with various sizes and verify that different sizes sample the different granularities of position-sensitive. Based on the conclusion and the idea of pyramid structure pooling, we propose a deep position-sensitive network that aggregates different divisions of position-sensitive score maps. Our network extracts feature maps using a modified ResNet, and then using two fully convolutional layers to produce the pyramid structure of various sizes score maps. We candidate regions using the region proposal network (RPN), and compute the generated scores of each region of interest (ROI) using different sizes position-sensitive ROI pooling layers. In the end, we apply the softmax layer to generate the probability of every ROI. Our experimental results show that the proposed method can effectively enhance the capacity to learn the object's position-sensitive information. For the same experimental conditions, we train our network with various sizes assembly on PASCAL 2007+2012 dataset and test on the PSACAL 2007 testset. Most of results is better than the single size that is included in the assembly, but since the granularity of $\mathbf{5 \times 5}$ and $\mathbf{7 \times 7}$ size is too close, the performance is similar with single $\mathbf{7 \times 7}$ szie. The best result is 74.69% mAP with $\mathbf{3 \times 3}$ and $\mathbf{7 \times 7}$ size, which is better than the best single position-sensitive network $\mathbf{7 \times 7}$ size by 2.56%.

***Keywords*—**object detection, deep learning, position-sensitive, residual network

## I. INTRODUCTION

DEEP learning has grown rapidly and revolutionized computer vision over the last few years. In varied vision problems, deep learning has been shown to reach the state-of-the-art stage, including image classification [9], object segmentation [10], and object detection [11]. According to deep learning, there are two prevalent families of object detection: (i) classification methods, which are networks that hypothesize bounding boxes and proposal regions using selective search [2,3] or RPN [19], sample features or pixels with a convolutional subnetwork for every proposal region, and then classify the accurate objects' region and bounding boxes; and (ii) regression methods [8,12,14], which are networks that regard object detection as a regression problem and use a single convolutional network that is not independent of the region of interest for prediction. All of these object detection networks are based on high-quality image classification networks, such as Alexnet [9], VGG [18], GoogLeNet [13], and ResNet [21], which can sample more quality features than other artificial features, for example, SIFT [4], HOG [5], or Harr [6]. Because of the difference between image classification and object detection, image classification networks fail to consider the object's translation variance. To remedy this issue, the aforementioned networks' developers design various structures, including over Feat [8], YOLO [12], SSD [14], RPN, and the position-sensitive network [20], to reinforce the ability to learn the object's translation variance.

We follow the approach of classification and propose a deep position-sensitive network (see Fig. 1). Using an idea from pyramid structure pooling [1, 16], the fundamental improvement in the capacity of the learning object's translation variance results from aggregating the position-sensitive score maps [20] of different ROI output sizes. We compare the ROI output sizes from $2 \times 2$ to $3 \times 3$, which demonstrates that different output sizes have a disparate capacity to learn translation variance. Thus, we naturally assemble the various position-sensitive score maps to remedy their weaknesses. While this contribution may seem to be a small development, we substantially enhance the ability to learn position-sensitive information and improve the accuracy of object detection for PASCAL VOC 2007 testset from 72.13% mAP for single position-sensitive networks to 74.69% for our network.

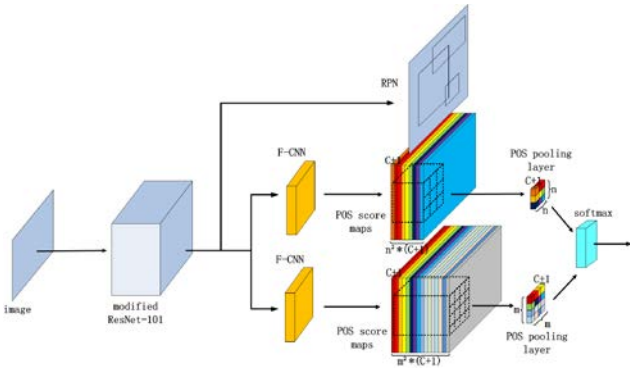



**Fig. 1.** Architecture: Our detection model has two fully convolutional layers and RPN followed by modified ResNet-101. Each fully

This work was supported by National Natural Science Foundation of China (Grant No. 61572392，61671362 and 61202464), Natural Science Foundation in Shaanxi Province of China (Grant No.2017JC2-08) and National Joint Engineering Laboratory Fund Project of New Network and Detection Control (Grant No. GSYSJ2016006).

Feng Xiao is with the school of computer science and technology, Xi'an Technological University, Xi'an 710021, Xi'an, China (corresponding author; e-mail: xffriends@163.com).

Mengmeng Bai is with the school of computer science and technology, Xi'an Technological University, Xi'an 710021, Xi'an, China.

Third Author is with the school of computer science and technology, Xi'an Technological University, Xi'an 710021, Xi'an, China.

Defa Hu is with the School of Computer and Information Engineering, Hunan University of Commerce, Changsha 410205, Hunan, China

convolutional layer reduces the output dimension of ResNet-101 from 2048-d to 1024-d, and then combines with a different size of position-sensitive score map. For each of the $m \times m$ and $n \times n$ bins in an ROI, POS pooling layers compute its map score. At the end of the model, we use softmax to produce the probability of each proposal region by assembling the result of the POS pooling layers' votes.

## II. Deep Position-Sensitive Network

Our network still adopts the prevalent two-stage object detection strategy. Similar to Faster-RCNN or R-FCN, we apply an RPN for the region proposal; however, we use a deep position-sensitive network, which is deeper than R-FCN, for region classification: see Fig. 1. Given the ResNet released by the authors of [21], which demonstrates a high quality of image classification and better comparison with other networks, we use ResNet-101 to sample the pixels or features. To increase the depth of the position-sensitive network, we attach two randomly initialized fully convolutional layers, which need to modify the last convolutional block in ResNet-101: see Fig. 2. Following R-FCN, all learnable weight layers are convolution, and at the end of two convolutional layers, they produce two banks of m×m and n×n position-sensitive score maps for classification. Thus, each C object category, including the background, are m×m×(c+1)-channel and n×n×(c+1)-channel output layers deeper than the R-FCN. Each bank of m×m or n×n score maps corresponds to its spatial grid that describes relative positions. The two position-sensitive ROI pooling layers are applied to generate the corresponding scores for each ROI, and then we aggregate the two score map out of the m×m and n×n score maps as the last scores for each ROI. We introduce more details as follows:

### A. Modified ResNet

We remove two end layers, 1000-class fully connected (FC) layer and average pooling layer, at the end of ResNet, and then add two fully convolutional layers and RPN. The other layer weights of the modified ResNet are initialized by ResNet-101 pre-trained on ImageNet [7]. Each added fully convolutional layer's dimension is 1,024, which can link with position-sensitive networks and reduce dimensions. The RPN is same as the Faster-RCNN: see Fig. 2.

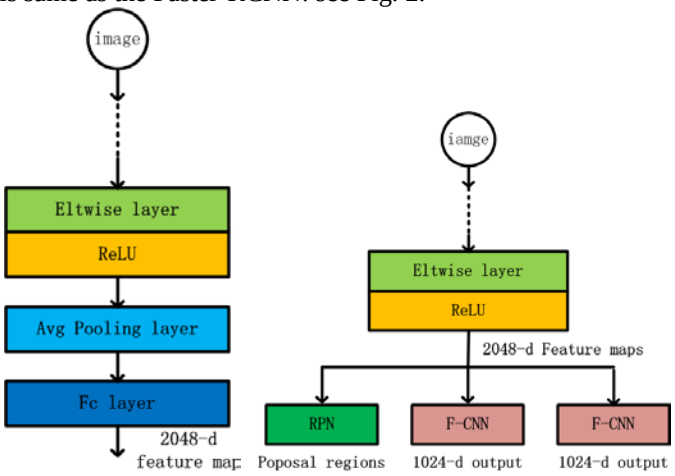
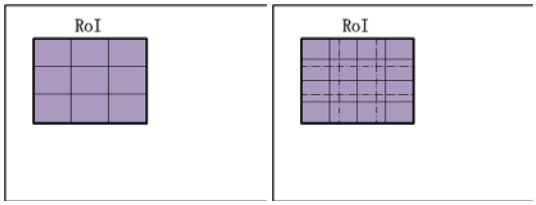



**Fig. 2.** Modified ResNet: At the end of ResNet-101, our model removes its average pooling layer and fully connected layer, which makes the ResNet-101 output dimension 2048-d. Following the Faster-RCNN, our model combines the RPN with the end layer of the modified ResNet-101, and then we also add two 1×1 fully convolutional layers behind this layer, which not only reduces the dimension but also links different position-sensitive networks.

### B. Deep position-sensitive network

The position-sensitive network contains position-sensitive score maps and ROI pooling layers. In the R-FCN, each ROI rectangle is only divided into k×k bins using a regular grid: see Fig. 3(a). This single size cannot encode all granularity position-sensitive information. To improve encoding, we divide each ROI rectangle into m×m and n×n bins, which results in a division similar to the pyramid structure: see Fig. 3(b).




( a ) Single score map size ( b ) Different size assembly

**Fig. 3.** Assembly of various sizes: (a) Single size division of ROI, whereby each ROI is divided into k×k (this figure shows 3×3). (b) Our model's division, in which we divide each ROI using different sizes (this figure shows 3×3 and 4×4), which can make the position-sensitive network learn a different granularity of position-sensitive information.

The size of each $3 \times 3$ or $4 \times 4$ bin is approximately $\frac{w}{m} \times \frac{m}{h}$ or $\frac{w}{n} \times \frac{n}{h}$. Following the R-FCN, the ROI pooling layer only pools the $(i, j)$-th $(0 \le i, j \le l-1)$ score map:

$$r_c(i,j|\theta) = \sum_{(x,y)\in bin\,(i,j)} Z_{i,j,C}(x + x_0, y + y_0|\theta)/l \quad (1)$$

where $Z_{i,j,C}$ is one score map out of the $g^2(C+1)$ score maps, $l$ is the number of pixels contained in each bin, $(x_0, y_0)$ indicates the top-left corner of an ROI, and $\theta$ is the weight of the network that we need to learn. To improve the pyramid structure, for the $m \times m$ size, the $(i,j)$-th bin spans $\lfloor i\frac{w}{m} \rfloor \le x < \lceil (i+1)\frac{w}{m} \rceil$ and $\lfloor j\frac{h}{m} \rfloor \le y < \lceil (j+1)\frac{h}{m} \rceil$, and for the $n \times n$ size, it spans $\lfloor i\frac{w}{n} \rfloor \le x < \lceil (i+1)\frac{w}{n} \rceil$ and $\lfloor j\frac{h}{n} \rfloor \le y < \lceil (j+1)\frac{h}{n} \rceil$. Then each position-sensitive score votes by averaging the scores in the ROI, and produces two $(C+1)$-dimensional vectors: $r_C^m(\theta) = \sum_{i,j} r_C(i,j|\theta)$ and $r_C^n(\theta) = \sum_{i,j} r_C(i,j|\theta)$. We compute the softmax responses across categories:

$$s_c(\theta) = e^{r_C^m(\theta)+r_C^n(\theta)} / \sum_{C'=0}^{C} (e^{r_e^m(\theta)} + e^{r_e^n(\theta)}) \quad (2)$$

### C. Training

For the unmodified layers of ResNet-101, we directly initialize weights using the ResNet-101 pre-trained on the ImageNet 1000-class competition dataset [7]. For each size of score map, we adopt an independent training approach, and then integrate the two trained score maps. During one training, our loss function is

$$L(s, t_{x,y,w,h}) = L_{cls}(s_{C^*}) + [C^* > 0]L_{re}(t, t^*) \quad (3)$$

where $L_{cls}$ is the cross-entropy loss used for classification, $L_{re}$ is the bounding box regressing loss, $C^*$ represents the ground-truth label of the ROI, $t^*$ is the object's ground-truth

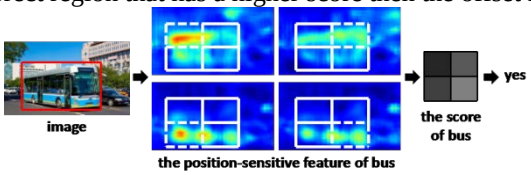
box, and t is the bounding box from the RPN. Because the proposal region from the RPN contains many negative samples, we use online hard example mining (OHEM) [22], which can reduce the influence of imbalanced data in the network. OHEM is the approach used to select the top k-most loss proposal regions from the all proposal regions updated by the weight of the network.

## III. Experiment

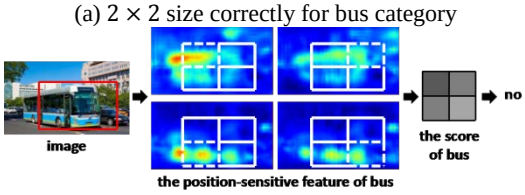
To precisely verify the improvement in the ability to learn position-sensitive information, we conducted a series of comparison experiments with various size score maps and their combinations. To demonstrate the accuracy of our network, we trained on the PASCAL VOC 2007+2012 trainval and evaluated on the VOC 2007 test dataset [23], and then compared our network with other networks using region proposals.

### *A. Comparison with various size score maps*

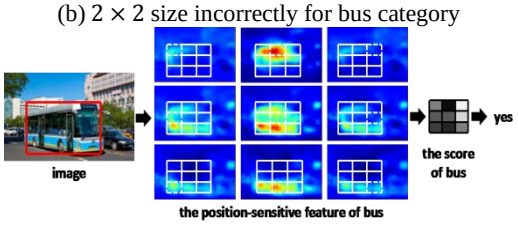
The single position-sensitive score maps sample the different granularities of position-sensitive information with different sizes. To verify this, we trained the 2×2 and 3×3 sizes on VOC 2007+2012 with 50,000 iterations. For the 2×2 size, the region proposal from the RPN was divided into four grids that represented the probability of four object's regions. In Fig. 4, we visualize the single position-sensitive network with 2×2 and 3×3 sizes to compute the probability for the bus. Figure 4 shows the correct region that has a higher score then the offset region.



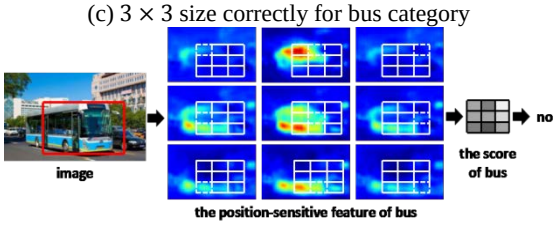

(a) 2 × 2 size correctly for bus category




(b) 2 × 2 size incorrectly for bus category




(c) 3 × 3 size correctly for bus category

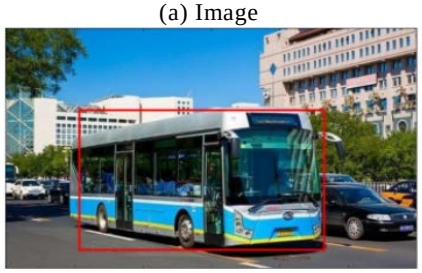



(d) 3 × 3 size incorrectly for bus category

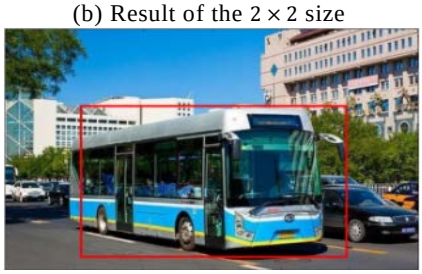
**Fig. 4.** Visualization for single 2×2 and 3×3 size detection. Each k×k size outputs k×k position-sensitive feature maps. Each feature map represents the score of one grid's position-sensitive information: see the dotted borders. Then sum all the grid's score, and the highest score is the correct bus. We use the gray level to visualize the probability of the bus category.

The 3×3 size is the same as the 2×2 size, but we found that different sizes detected different regions for one image (Fig. 6). In the analysis of the score maps, we note that for the 2×2 size, the network only learns the object's essential position-sensitive information, and if the size is 5×5 or 7×7, the network learns more detailed information. Because the detailed information can increase accuracy, it also makes detection more easily influenced by noise. Figure 5(b) shows the detection of the main body of the bus, but loses the rear-view mirror, whereas Figures 5 (c) and (d)'s results appear offset, even though they detect the rear-view mirror well.

(a) Image



(b) Result of the 2 × 2 size

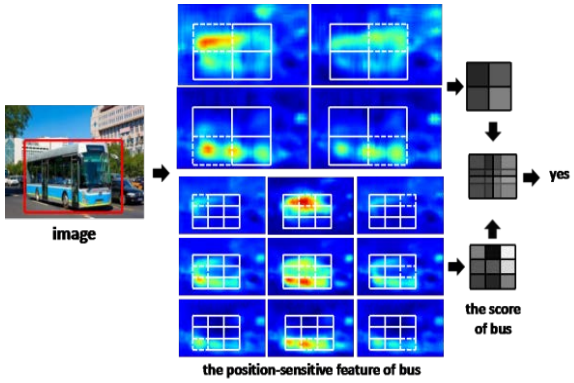


(c) Result of the 5 × 5 size

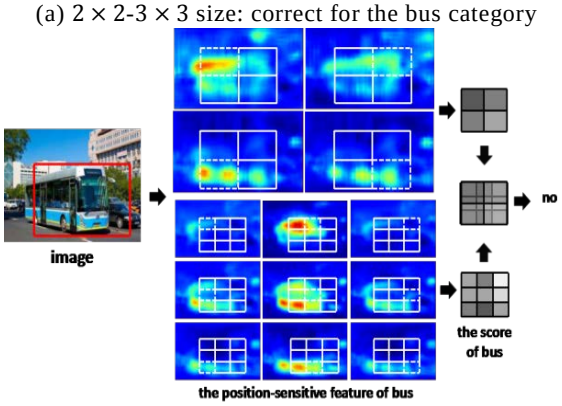
(d) Result of 7 × 7 size

**Fig. 5.** Comparison of single score maps' results using various sizes: (a) Image including the bus. (b)(c)(d) Results of using single-size score maps, whose sizes are 2×2, 5×5, and 7×7, respectively. Using the 2×2 size, the single-size model can only detect the backbone of the bus, but using 5×5 or 7×7, the bounding box is offset, even though it contains the rear-view mirror.

To remedy these weaknesses, we assembled different size score maps to compute the probability category of the proposal region, Fig. 6, which demonstrates the main idea of our network. Thus, our network not only learns the backbone position-sensitive information but also samples the detailed information.




(a) 2 × 2-3 × 3 size: correct for the bus category

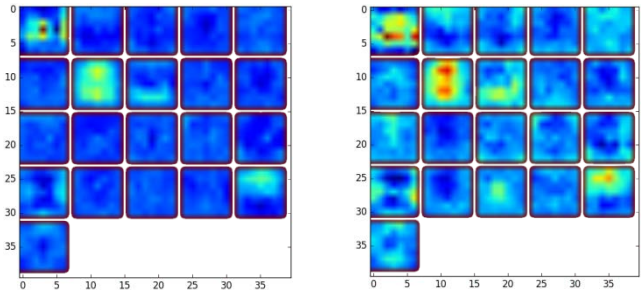



(b) 2 × 2-3 × 3 size: incorrect for the bus category

**Fig. 6.** Visualization for single 2×2 and 3×3 size detection

### B. *Comparison with POS pooling layers*

( a ) Image



( b ) Features of the single 7 × 7 size score

( c ) Features of aggregating the size score

**Fig. 7.** Visualization to compare the features of single and aggregating sizes.

Figure 7(a) shows the image, including the bus and car, Fig. 7(b) shows the features of the position-sensitive pooling layers sampled by R-FCN[20]( 7×7 size, ResNet-101), and Fig. 7(c) shows the result for our network (3×3 and 7×7 size, ResNet-101). In these feature maps, the top-left map is the background, and the other 20 maps are the features of the 20-class PASCAL VOC dataset. From left to right and top to bottom, the seventh map is the bus category and the car is behind the bus. Comparing the features of Figs. 7(b) and (c), since we adopted a multiple granularity score map, our network obtains more position-sensitive information and results of object detection (Fig. 8).

(a) Single size for the bus

(b) Aggregating sizes for the bus

(c) Single size for the car

(d) Aggregating sizes for the car

**Fig. 8.** Visualization to compare the results of single and aggregating sizes

### C. *Comparison with other networks*

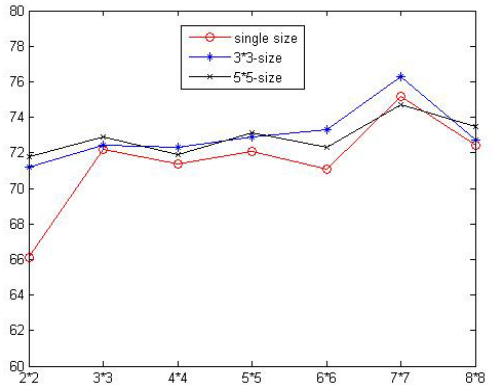
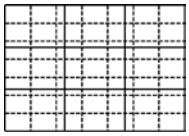



**Fig. 9.** Comparison of various sizes of POS score maps

To further demonstrate the effectiveness of our network, we conducted a series of experiments, which included various single sizes and different composite models. All these networks used ResNet-101, the GPU was GTX 1080-8G; CPU was Xeon-E5, CUDA-8.0, cuDNN-V5; the dataset was VOC 2007+2012; we tested on VOC 2007; and trained these networks with 100,000 iterations. In Fig. 9, a single

position-sensitive score map shows the accuracy of various sizes that modified the R-FCN from 2×2 to 8×8. Among these single sizes, the best is 7×7 and the worst is 2×2. The curve in Fig. 9 shows the rule that the accuracy increases as the size aggregates at the beginning, but decreases when the size continues to aggregate. The 3×3-size curve represents the result of the 3×3 size aggregated with other sizes from 2×2 to 8×8. Clearly, the 3×3+2×2 size is the worst, but better than the single 2×2 size, and the 3×3+7×7 size is the best for all types of assembly. The 5×5 size curve has a similar trend to the 3×3 size curve, but the 5×5+7×7 size is worse than the single 7×7 size.

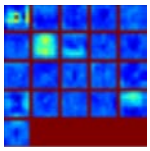
(a)5 × 5+7 × 7



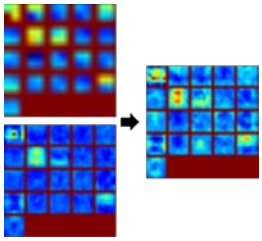
(b) 3 × 3+7 × 7

**Fig. 10.** Visualization for different composite models

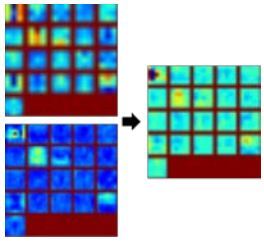
(a)Image



(b)7 × 7 features



(c)3 × 3+7 × 7 features



(d)5 × 5+7 × 7 features

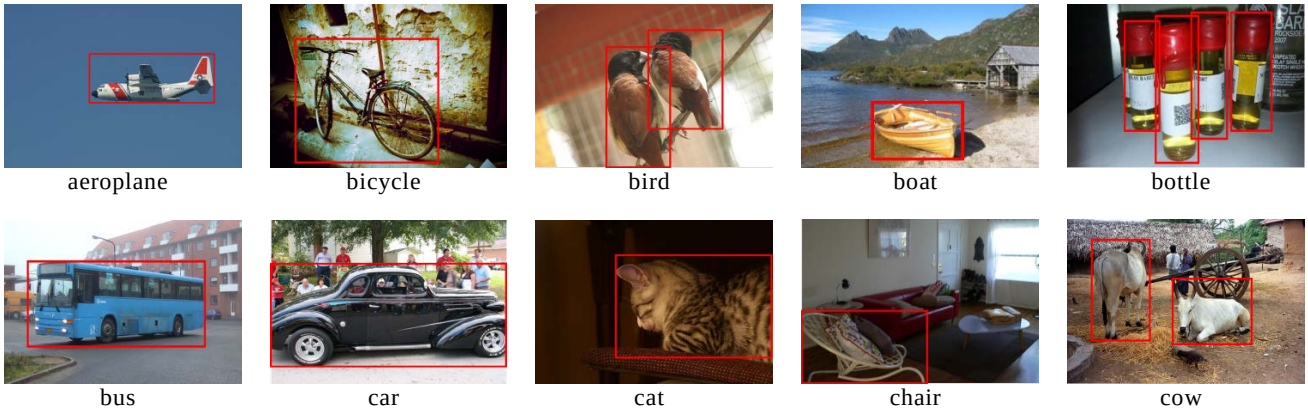
**Fig. 11.** Visualization to compare various size features

To further examine our model, we consider the detailed results of the comparison with other RPN (Faster-RCNN and R-FCN). Using the same condition that we mentioned previously, we train these networks on VOC 2007+2012 with 100,000 iterations and evaluate them on the VOC 2007 test set. The three networks use the same RPN for the region proposal and then classify it. For a fair comparison, we combine the Faster-RCNN and R-FCN with ResNet-101, and for R-FCN, we adopt the best single score map size, which is 7×7. Aggregated with 3×3 and 7×7, the mAP increases by 2.56% to 74.69%: see Table 1 for more details and Fig. 12 for the detection results.

**Table 1** Model experiment on PASCAL VOC

| Algorithm | VOC07+12 | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow | dining table |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Faster-rcnn | 0.7079 | 0.7255 | 0.7730 | 0.7369 | 0.6032 | 0.5398 | 0.7691 | 0.8277 | 0.8432 | 0.5145 | 0.7932 | 0.5802 |
| R-FCN | 0.7213 | 0.7511 | 0.7873 | 0.7341 | 0.6139 | 0.5283 | 0.7737 | 0.8410 | 0.8491 | 0.5464 | 0.7841 | 0.6308 |
| Ours | **0.7469** | **0.7882** | **0.7970** | **0.7643** | **0.6161** | **0.6035** | **0.8200** | **0.8554** | **0.8810** | **0.5888** | **0.8196** | **0.6684** |

| Algorithm | dog | horse | motorbike | person | pottedplant | sheep | sofa | train | tvmonitor |
|---|---|---|---|---|---|---|---|---|---|
| Faster-rcnn | **0.8556** | 0.8185 | 0.7539 | 0.7742 | 0.4292 | 0.6681 | 0.6874 | 0.8181 | 0.6458 |
| R-FCN | 0.8089 | 0.8306 | 0.7663 | 0.7860 | **0.4732** | **0.7503** | 0.6843 | 0.7882 | 0.6986 |
| Ours | 0.8385 | **0.8420** | **0.7925** | **0.7925** | 0.4434 | 0.7473 | **0.7429** | **0.8204** | **0.7169** |



aeroplane bicycle bird boat bottle

bus car cat chair cow

**Fig. 12** Example of our model's qualitative results

## IV. Conclusion

We introduced a deep position-sensitive network for object detection, and analyzed how our model enhanced the ability to obtain position-sensitive information. Using a series experiments, we found that different score map sizes encoded different granularity information of the position-sensitive network, which directly influenced the accuracy of detection. In the same experimental condition, our model's pyramidal structure of a position-sensitive network, better than the single size network, increases the accuracy of object detection. In the

future, we will continue improving our network, for instance aggregating more various sizes score maps.

## REFERENCES


[1] Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov, "Scalable Object Detection Using Deep Neural Networks", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2155-2162.

[2] Uijlings, J. R., et al. "Selective Search for Object Recognition." *International Journal of Computer Vision*, vol.104,no.2, pp.154-171,2013.

[3] Theeuwes J. "Stimulus-driven capture and attentional set: selective search for color and visual abrupt onsets". *Journal of Experimental Psychology Human Perception & Performance*, vol.20, no.4, pp.799-806, 1994.

[4] Ng P C, Henikoff S. "SIFT: predicting amino acid changes that affect protein function". *Nucleic Acids Research*, vol.31, no.13, pp.3812-3814, 2003.

[5] Dalal, N., and B. Triggs. "Histograms of oriented gradients for human detection", *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on IEEE*, vol.1, pp.886-893, 2005.

[6] Papageorgiou C P, Oren M, Poggio T. "A General Framework for Object DetectionInternational", *Conference on Computer Vision. IEEE Xplore*, vol.1, pp.555-562, 1998.

[7] Deng, Jia, et al. "ImageNet: A large-scale hierarchical image database", *Computer Vision and Pattern Recognition*, 2009. IEEE Conference on IEEE, 248-255, 2009.

[8] Sermanet P, Eigen D, Zhang X, et al. "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks". *Eprint Arxiv*, 2013.

[9] Krizhevsky A, Sutskever I, Hinton G E. "ImageNet classification with deep convolutional neural networks". *Advances in Neural Information Processing Systems*, vol.25, no.2,pp.2097-1105, 2012.

[10] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C].*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015: 3431-3440.

[11] Redmon J, Divvala S, Girshick R, et al. "You Only Look Once: Unified, Real-Time Object Detection". *Computer Science*, pp.779-788,2016.

[12] Szegedy C, Liu W, Jia Y, et al. "Going deeper with convolutions", *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp.1-9.

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., & Fu, C. Y., et al. "SSD: Single Shot MultiBox Detector", *European Conference on Computer Vision. Springer*, Cham, 2016, pp.21-37.

[14] Girshick R, Donahue J, Darrell T, et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". *Computer Science*, pp.580-587, 2014.

[15] He K, Zhang X, Ren S, et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.37, no.9, pp.1904-16, 2015.

[16] Girshick R. "Fast R-CNN", *IEEE International Conference on Computer Vision IEEE*, pp.1440-1448, 2015.

[17] Simonyan K, Zisserman A. "Very Deep Convolutional Networks for Large-Scale Image Recognition". *arXiv preprint arXiv*:1409.1556, 2014.

[18] Ren S, He K, Girshick R, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.". *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.39, no.6, pp.1137, 2016.

[19] Dai J, Li Y, He K, et al. "R-FCN: Object Detection via Region-based Fully Convolutional Networks". *arXiv preprint arXiv*:1605.06409, 2016.

[20] He K, Zhang X, Ren S, et al. "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society*, 2016, pp.770-778.

[21] Shrivastava A, Gupta A, Girshick R. "Training Region-Based Object Detectors with Online Hard Example Mining", *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2016, pp.761-769.

[22] Everingham, Mark, et al. "The Pascal Visual Object Classes (VOC) Challenge", *International Journal of Computer Vision*, vol.88, no.2, pp.303-338, 2016.



**Feng Xiao** was received his B.Sc. and M.Sc. degrees in computer science from Xi'an Technological University in 2000 and 2003, respectively. After the M.Sc. degree, he joined the faculty member of Xi'an Technological University. Since 2006 he worked toward his Ph.D. degree at Northwest University and received his Ph. D. degree in computer science from Northwest University in 2002. His research interests include digital image processing, pattern recognition, image retrieval machine learning

**Mengmeng Bai** received his B.Sc. degrees in computer science from Hebei University of Science and Technology in 2014. Now he is a graduate student in Xi'an Technological University and His research directions is pattern identification

**Li Zhao** received his B.Sc. degrees in computer science from Xi'an Technological University in 1994, and his M.Sc. degrees in computer science from Nanjing University of Science and Technology. Now she is a professor in Xi'an Technological University and her directions is artificial intellegence and data mining.

**Defa Hu** received the PhD degree in computer science and technology from Hunan University of China. Currently, he is a researcher (assistant professor) at Hunan University of Commerce, China. His major research interests include information security and image processing. He is invited as a reviewer by the editors of some international journals, such as Multimedia Tools and Applications, International Journal of Information and Computer Security, etc. He has published many papers in related journals.