# MFRPN: Towards High-Quality Region Proposal Generation in Object Detection

Dingqian Zhang(✉), Hui Zhang, Wanling Zeng, Zhongxing Han, and Xiaohui Hu

Institute of Software Chinese Academy of Sciences, Beijing, China
dingqian2015@iscas.ac.cn

**Abstract.** Most state-of-the-art object detection networks need region proposals in their two-step framework. Popular region proposal networks can provide hundred proposals with acceptable accuracy. In this paper, we introduce a Multiple Filters Region Proposal Network (MFRPN) that can change its structure with dataset. We calculate the suitable sizes of filters and use multiple filters with appropriate reference boxes to make the regression of coordinates of proposals more accurate. To illustrate the proposed MFRPN, we adopt the framework of Faster R-CNN [1] and replace the RPN with the MFRPN. As a result, we get 0.98% improvement in mean AP on PASCAL VOC 2007 and 1.45% on PASCAL VOC 2012.

**Keywords:** Object detection · Multiple filters · Reference box
Region proposal

## 1 Introduction

Object detection is to detect specific objects in images. Generally, it consists two steps: finding where the objects are (proposals generation), then giving these objects category labels and confidence scores (objects classification). This two-step division matches to visual mechanism of human beings. We first give a scan of the whole image to get the region we really care about. Then we observe carefully for more details to identify what we look at. Although one-step object detection algorithms (e.g., YOLO [12] and SSD [10]) exist, their prediction accuracy is lower than two-step algorithms. In this paper, we focus on improving two-step object detection algorithms. Based on the difference of one-step and two-step algorithms, we can draw a conclusion that proposals are useful for object detection asking for high accuracy. Our algorithm Multiple Filters Region Proposal Network (MFRPN) settles down to generating high-quality proposals. As we will

D. Zhang—Student of ISCAS.

This work is supported by the Natural Science Foundation of China (U1435220) (61503365).

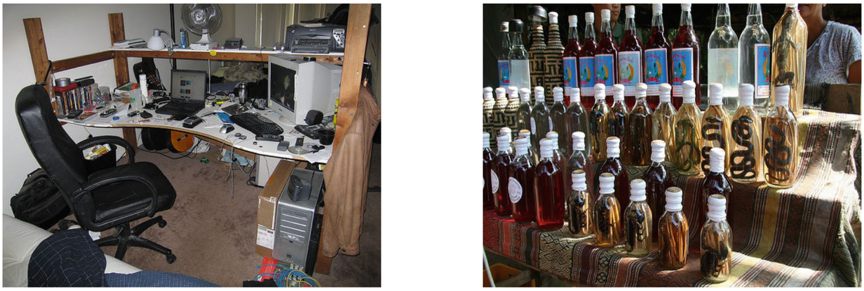
J. Yang et al. (Eds.): CCCV 2017, Part III, CCIS 773, pp. 61–72, 2017.
https://doi.org/10.1007/978-981-10-7305-2_6

show in following sections, detection network with MFRPN shows advantages in detection accuracy.

In early time, proposals are generated by some classical algorithms (e.g., Selective Search [16] and EdgeBox [17]) which provide about thousands of proposals per image to insure covering all the possible objects. However, these solutions are summaried by experts. They have their weaknesses in some situations. Thanks to the development of deep learning, the work of extracting features from pictures is done perfectly by Convolutional Neural Networks (CNN). CNN has rich representation capacity and powerful generalization ability and it can take advantage of computing ability of GPU. Therefore, we can get more accurate features of objects easily.

Nowadays, most state-of-the-art object detection networks use the Region Proposal Network (RPN) [14] to generate proposals for object detection (e.g., Faster R-CNN [14] and R-FCN [3]). RPN has several convolutional layers, one regression layer and one classification layer. To effectively generate different scales or aspect ratios proposals, RPN uses multiple references for every predicted boxes regression work (typically, 9 different reference boxes). But in this method, many hyper-parameters are set directly, such as, the sizes of reference boxes. The construction of a nice network model should be data-driven. If parameters of a model are changed with dataset, we believe it has universality and stability. The original RPN is not suitable for images which include small (Fig. 1a) or dense (Fig. 1b) objects. Because proposals of these images need more accurate predicted coordinates and a confined network cannot deal with it well. We also notice that the sizes of receptive fields are important to generate proposals. Sizes of receptive fields should be close to the sizes of reference boxes.

If receptive fields are too small, it may cause under-fitting problem. Because the information is too little to make correct decision. Oppositely, If receptive fields are too large, it may cause over-fitting problem. Because much information is redundant. For example, a network may misunderstand that chairs must be put next to a table. Our MFRPN has the ability to deal with details and all multiple filters (kernel sizes are different) on the last layer of CNN has suitable reference boxes. We believe MFRPN can sovle the problems above.



(a) (b)

**Fig. 1.** Example images with complex contents: (a) full of small objects, (b) dense similar objects.

Figure 2 shows our idea clearly. The size of receptive field determine how much information a RPN can get in one time regression. If receptive fields are too large or too small, the features extracted by a RPN will be more or less then the object itself. So we choose suitable sizes of reference boxes in our object detection network. Since different categories of objects have their own standard sizes. We should use different filters for different objects. And training networks with multiple filters simultaneously can make regression smoother. In conclusion, our main contributions are:

1. We emphasis appropriate mount of information is quite important for regression. The information here is a part of an image in receptive field. So the size of the filter on the last convolution layer should be changed with dataset and match the object's size.
2. We summarize the phenomenon that appropriate sizes of reference boxes are important for boxes regression. It means the sizes of reference boxes should match their filters receptive fields rather than the object itself.
3. We design multiple filters region proposal network. For small and dense object detection, we can use multiple filters with similar sizes. For detecting object with various sizes, we can use multiple filters with various sizes.

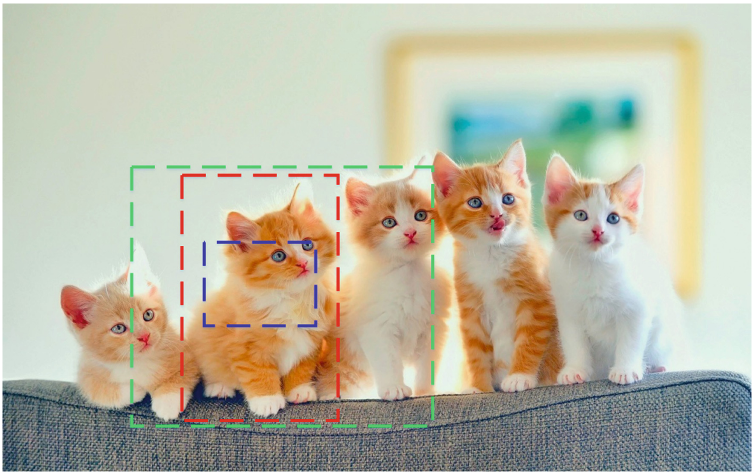


**Fig. 2.** Our idea: Small receptive field (blue) cannot get the whole features of object. Large receptive filed (green) will bring wrong features. Appropriate receptive filed (red) make regression more accurate. (Color figure online)

## 2 Related Work

There are diverse methods for generating object proposals. The most widely used unsupervised method is Selective Search [16]. It is a clustering method. Other popular clustering method are EdgeBox [17] and MCG [1]. The advantage of clustering methods is they can provide proposals of multiple scales and sizes spontaneously. BING [2], MultiBox [4] are typical in supervised methods. BING uses the binary feature which can be promoted computing speed by registers. MultiBox uses CNN to generate proposals. But recently proposed promising solution, Region Proposal Network (RPN) [14], has more excellent performance. It uses multi-task loss function to combine proposals generation and coordinates regression together in one step. RPN can reduce the number of proposals to less than 300 with higher recall rate. In our experiments, we uses RPN as the baseline algorithm.

The RPN in Faster RCNN adopts 3 aspect ratios (1:1, 1:2, 2:1) and 3 scales ($128^2$, $256^2$, $512^2$) anchor boxes as reference boxes basing on the Simonyan and Zisserman model (VGG-16) [15]. Different aspect ratios anchor boxes are convenient to generate different shape proposals. Different scales anchor boxes are used for generating multiple scales proposal more accurately. The size of 3 scales are designed elaborately which are smaller than the receptive field, nearly equal to the receptive field and larger than the receptive field. But their information are all come from the same piece of feature map. That is to say if the predicted field matches the size of receptive field, the predicted result will be more reliable. So we claim that reference boxes should not much larger than receptive field of filter. And we should pay more attention to details in receptive field.

YOLO 9000 [13] clusters images by size and changes the reference boxes' sizes with dataset. But it doesn't change the filters' sizes. We believe that changing dataset should bring changes in filters' sizes. And changing filters should bring changes in reference boxes, too. Appropriate filters are more important than reference boxes.

As for multiple filters train simultaneously, SPP [6] and Grid Loss [11] does the similar but different job. SPP proposes a spatial pyramid pooling layer which uses multiple kernel sizes to get a fixed-length representation. Grid Loss proposes a novel loss layer for CNN which minimizes error rate on both sub-blocks and the whole feature map. These algorithms focus on use the relationship of part and whole to detect objects. Because we expect to focus the whole object itself, we use kernels of multiple sizes separately.

In recent years, many novel methods have been proposed to improve prediction accuracy given by RPN. HyperNet [8] and FPN [9] are typical two of these methods. They try to use both high level and low level feature together to predict proposals. But we put attention to improve accuracy of coordinates regression. MFRPN is a new path to get better proposals and can cooperate with their networks.

## 3 Our Approach

In this section, we introduce Multiple Filter Region Proposal Network (MFRPN). We will explain how we design our network, how it works, and how to use MFRPN in an object detection network in detail.

### 3.1 Confirming Sizes of Filters

In this step, we will cluster all bounding boxes in our training set. And try to find appropriate filters which receptive fields are the nearest top K to clustering result. Figure 3 shows the relationship between reference box and bounding box. In this paper, we use k-means clustering method to solve this problem. There are three steps:

1. Run k-means clustering on dataset to divide sizes of objects into K categories.
2. According to the used network, calculate the corresponding relationship between filters and receptive fields.
3. Confirm sizes of filters on the last convolution layer which receptive fields are the nearest top K to bounding boxes clustering result.

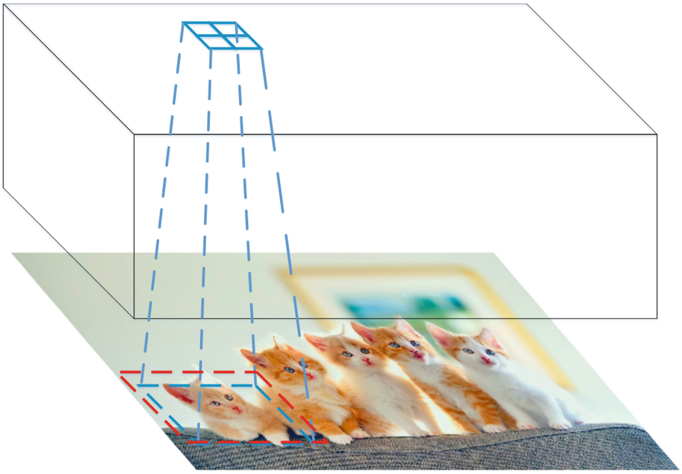


**Fig. 3.** The relationship between receptive field (blue) and bounding box (red) in our network. We try to use filters with appropriate receptive fields. (Color figure online)

### 3.2 Multiple Filters Region Proposal Networks

After calculating filters' sizes, We will build our region proposal network. The MFRPN is several convolutional layers (Fig. 4). We use $K$ small networks to slide over the feature map output by the last convolutional layer [14]. Note that the $K$ here is equal to K in k-means clustering. These small networks take $n_1^2 \sim n_k^2$ part of the last feature map. That means the kernel sizes of the networks' filters are $n_1 \sim n_k$. Although the sizes of filters are different, we equal the numbers of outputs of filters. In this paper, the number of outputs is 512. So every filter generates 512-d lower-dimensional features. Every independent convolutional layer is followed by two $1 \times 1$ convolutional layers. One for classification, one for coordinates regression.

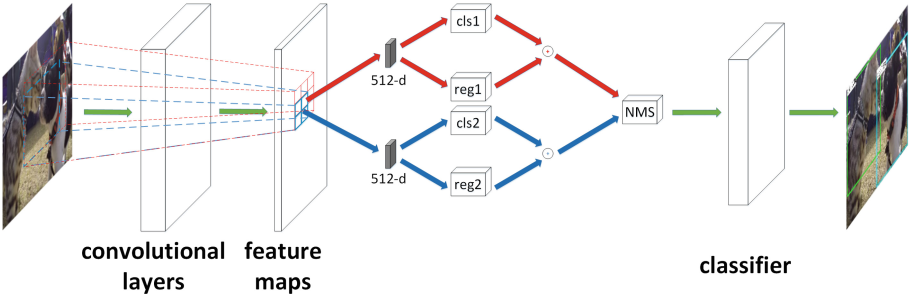


**Fig. 4.** MFRPN object detection architecture. This architecture can be extended to include more than two filters to generate high-quality proposals. Our experiments just use 2 filters for demonstration. Different filters provide different proposals which are merged by NMS.

In the adopted network (VGG-16), the receptive fields of filters is showed in Table 1. We will choose the appropriate filter for each category clustered in Sect. 3.1.

**Table 1.** The size of receptive fields of filters in VGG-16.

| Size of filter | 1 × 1 | 2 × 2 | 3 × 3 | 4 × 4 | 5 × 5 | 6 × 6 |
|---|---|---|---|---|---|---|
| Size of receptive field | $196^2$ | $212^2$ | $228^2$ | $244^2$ | $260^2$ | $276^2$ |

### 3.3 Extensible Loss Function

For training MFRPNs, we use multi-task loss function. And all filters contribute to the loss. The loss function for an image is defined as:

$$L(p_{ki}, t_{ki}) = \frac{1}{N_{cls}} \sum_k \sum_i L_{cls}(p_{ki}, p_i^*) + \frac{1}{N_{reg}} \sum_k \lambda_k \sum_i p_i^* L_{reg}(t_{ki}, t_i^*). \quad (1)$$

In this equation, $i$ is the index of a reference box in a mini-batch, $k$ is the index of multiple filters. $p_{ki}$ is the probability of reference box being an object predicted by filter $k$. But the ground-truth label $p_i^*$ is independent of filters. If reference box is positive, $p_i^*$ is 1. And if reference box is negative, $p_i^*$ is 0. In second part of the equation, $t_{ki}$ ($t_i^*$) is 4-d feature stands for coordinates of the center of a predicted box (reference box), height and width normalized in method proposed in [5]. $L_{cls}$ is log loss, $L_{reg}$ is robust loss [5], defined as:

$$L_{reg}(t_{ki}, t_i^*) = \sum_{i \in x,y,w,h} smooth_{L_1}(t_{ki} - t_i^*). \quad (2)$$

in which

$$smooth_{L_1} = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise. \end{cases} \quad (3)$$

The two part loss functions are normalized by $N_{cls}$ and $N_{reg}$. In this paper $N_{cls}$ (=256) and $N_{reg}$ (≈2400) are balanced by $\lambda_k$, and $\lambda_k$ are all 10 in this paper. As we can see, the closer to ground truth, the more accurate regression is. Because the loss is quadratic, when the difference value is in $(-1, 1)$. But MFRPN uses multiple filters to train convolutional layers together. In this case, there are more difference values in $(-1, 1)$. So our algorithm makes the regression smoother.

### 3.4 Training Object Detection Network with MFRPN

For training MFRPN, we follow the training method in [14]. But we use different labels and box-targets for different filters. In each mini-batch, we use 256 different reference boxes for each filter.

In this paper, we use the Fast R-CNN [5] as our classification network. For traning object detection network with MFRPN, we use 4-Step Alternating Training [14] algorithm.

## 4 Experiments

To compare with Faster R-CNN we replace the RPN of Faster R-CNN with our MFRPN. It is worth mentioning that our convolution layers can be replace by other CNNs like ResNet [7] and our classification network can be replaced, too. We use PASCAL VOC 2007 and 2012 dataset in training and testing phase.

### 4.1 Experiments on Choosing Filters

We choose k-means as our clustering method. The distance between too objects is defined as:

$$distance(a, b) = -IOU(a, b) \tag{4}$$

In this step, the position of object is useless. So we move all the bounding boxes to the top left corner and then calculate the distance. Table 2 shows some clustering results on PASCAL VOC 2007 and PASCAL VOC 2012 dataset.

**Table 2.** Part of clustering results on PASCAL VOC 2007 training set. We change the number of clustering categories for each experiment.

| | K = 2 | K = 3 |
|---|---|---|
| VOC 2007 | 70 × 93 263 × 253 | 54 × 74 147 × 183 340 × 287 |
| VOC 2012 | 71 × 90 277 × 262 | 53 × 67 150 × 188 352 × 294 |

We will use $K = 2$ in this paper. Combined with Table 1 can be seen, the appropriate sizes of filters are $1 \times 1$ and $5 \times 5$ for PASCAL VOC 2007, $1 \times 1$ and $6 \times 6$ for PASCAL VOC 2012.

### 4.2 Experiments on Changing Reference Boxes

We claim that reference box must match the receptive field. We use the Faster R-CNN [14] as the basic experimental method. Faster R-CNN use the RPN to generate proposals. On the last convolutional layer, the size of filter is 3. From the Table 1 we can find the receptive field of this filter is 218. Faster R-CNN uses multiple references. The sizes of the references are $128^2, 256^2, 512^2$. As we can see, $512^2$ is much larger than the filter's receptive field, so we change the references to $64^2, 128^2, 256^2$. The results are showed in Table 3.

**Table 3.** Detection results on PASCAL VOC 2007. We use different sizes of reference boxes, noting that the receptive fields are all close to 256.

| | Size of filter | Reference boxes | Mean AP |
|---|---|---|---|
| 1 | 3 | $128^2, 256^2, 512^2$ | 69.94 |
| 2 | 3 | $64^2, 128^2, 256^2$ | 70.62 |

From these results, we find the original network gets the lowest mean AP. Because the original reference boxes are much larger than the receptive field. It means, in a mini-batch, there are too many unknown factors to predict. The reference boxes experimental results meet our idea that appropriate sizes of reference boxes are good for generating high-quality proposals.

In our next experiments, we chose $1 \times 1, 5 \times 5$ and $6 \times 6$ filters to extract feature vectors. So we choose $64^2, 128^2, 256^2$ as the sizes of our reference boxes.

### 4.3 Experiments on Using Multiple Filters

We believe multiple filters can improve detection accuracy. Since images of PASCAL VOC are small. So our filters are similar sizes. Our theory is dividing receptive field more accurately is necessary. So in our experiments, multiple filters are smaller than single filter. We use multiple filters on the last convolutional layer of RPN and keep other parameters the same as Faster R-CNN. The results are showed in Table 4. When using multiple filters, the mean AP increases by over 0.15%. Although the improvement is little, it proves that multiple filters have a beneficial effect on generating high-quality proposals.

**Table 4.** Detection results on PASCAL VOC 2007. We use multiple filters. In this experiment, sizes of filters are 2 and 3.

| | Size of filter | Reference box | Mean AP |
|---|---|---|---|
| 1 | 3 | $128^2, 256^2, 512^2$ | 69.94 |
| 2 | 2 & 3 | $128^2, 256^2, 512^2$ | 70.02 |

### 4.4 Object Detection Networks with MFRPNs

So far, we have proved most of our ideas are useful for improving proposals' quality. We train object detection network (Faster R-CNN) with MFRPN. Since our dataset is PASCAL VOC, we set the sizes of multiple filters are 1 and 5 for 2007 dataset. And the sizes are changed to 1 and 6, when use 2012 training set. And the reference boxes are $64^2, 128^2, 256^2$. If the images are bigger, we will change the number and sizes of filters and reference boxes as well. We keep other parameters the same as Faster R-CNN. But the mean AP is 69.7698%, lower than we expect.

Because we get more proposals and most of these have more accurate vertex coordinates, mixing some proposals together is important in our experiments. NMS is a typical solution to proposals fusion. In Faster R-CNN, NMS reduces the number of proposals to 300 before they are sent to classification network in the final testing phase. But in our experiments, MFRPN needs more. Further, several experiments have been performed to find the appropriate number of proposals that should be kept and the results are shown in Table 5. In order to indicate our improvement is not the result of more number of proposals. We give classification network more proposals from original RPN, too. After these experiments, we keep 1300 proposals left after NMS and finish training the object detection network with MFRPN. The detection results are showed in Table 6. The mean AP of proposed MFRPN is higher than our baseline. To show the stability of our method, we keep the parameters same as what we set on PASCAL VOC 2007 and change the dataset to PASCAL VOC 2012. The detection results are showed in Table 7. Our method increases mean AP to 68.44%. The increment proves our method is stable. Figure 5 shows some results on the VOC test-dev set.

**Table 5.** Detection results on PASCAL VOC 2007. We use MFRPN or original RPN to generate proposals and detectors of Faster R-CNN to classify objects. In these experiments, numbers of proposals kept after NMS are different.

| Number of proposals | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 | 1300 | 1400 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mean AP (MFRPN) (%) | 69.77 | 69.94 | 70.23 | 70.32 | 70.58 | 70.61 | 70.63 | 70.75 | 70.84 | 70.91 | 70.91 | 70.78 |
| Mean AP (RPN) (%) | 69.94 | 70.03 | 70.04 | 70.00 | 69.98 | 70.01 | 70.02 | 70.02 | 70.02 | 70.02 | 70.02 | 70.02 |

**Table 6.** Results on PASCAL VOC 2007 test set (trained on VOC 2007 trainval) with detectors of Faster R-CNN and VGG-16. The proposals are generated by different methods and MFRPN provide 1300 proposals for the detector.

| | mean AP | areo | bike | bird | boat | bottle | bus | car | cat |
|---|---|---|---|---|---|---|---|---|---|
| RPN | 69.94 | 68.55 | 78.20 | 67.28 | 57.66 | 51.23 | 79.57 | **79.36** | **85.15** |
| MFRPN | **70.92** | **71.66** | **80.06** | **69.55** | **60.63** | **52.46** | **81.21** | 78.96 | 84.69 |

| | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RPN | 49.44 | 75.49 | 64.60 | 82.53 | 83.16 | **77.64** | 76.20 | 36.83 | **72.95** | **67.30** | **78.03** | **67.87** |
| MFRPN | **51.30** | **78.87** | **66.26** | **83.55** | **84.99** | 75.54 | **76.84** | **40.07** | 71.87 | 66.03 | 77.46 | 66.46 |

**Table 7.** Results on PASCAL VOC 2012 test set (trained on VOC 2012 trainval) with detectors of Faster R-CNN and VGG-16. The proposals are generated by different methods. To show the stability of our network, we keep 1300 proposals after NMS in MFRPN experiment, too.

| | mean AP | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RPN | 66.99 | 82.33 | 76.43 | 71.02 | 48.37 | 45.20 | 72.08 | 72.27 | 87.25 | 42.18 | **73.72** | 50.03 | 86.76 | 78.68 | 78.36 | 77.35 | 34.50 | **70.11** | **57.08** | 77.14 | 58.93 |
| MFRPN | **68.44** | **83.92** | **78.17** | **71.17** | **51.69** | **46.80** | **77.24** | **72.63** | **88.14** | **43.47** | 73.04 | **51.81** | **86.96** | **80.24** | **81.54** | **77.86** | **36.02** | 69.01 | 56.64 | **81.35** | **61.19** |

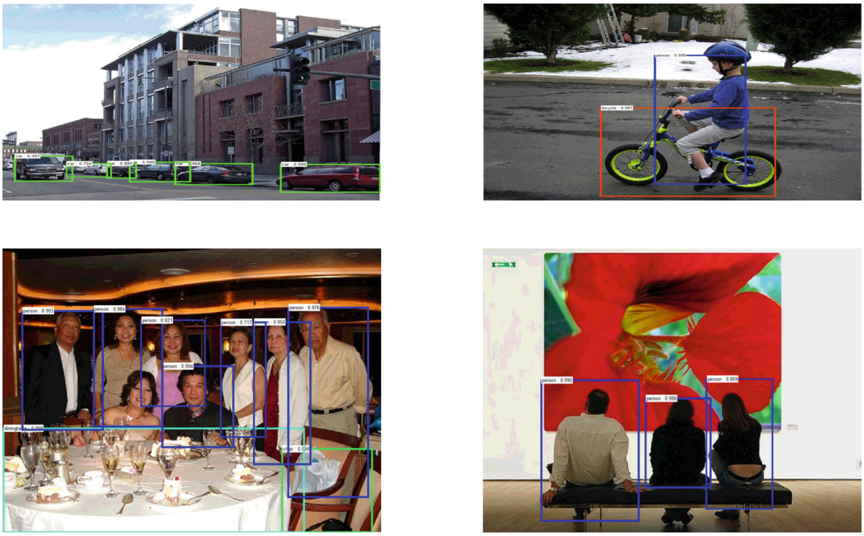


**Fig. 5.** Examples of our detection results.

We also compare the running time between baseline and our method. Because we use multi-task loss to implement our method, the training time is almost the same as the baseline. But during test phase, our method spend more time. In our experiment (NVIDIA TITAN X (Pascal)), the average test time is 0.350 s. Although the time is higher than the baseline which is 0.236 s, the speed is acceptable.

## 5 Conclusion

In this paper, we propose the Multiple Filters Region Proposal Network (MFRPN) for generating high-quality region proposals. The proposed MFRPN can change its structure with dataset. According to the general classification of objects' sizes, MFRPN can choose nice filters to cover objects automatically.

Then MFRPN adopts appropriate reference boxes and multiple filters to get more accurate proposals.

It can cooperate with most two-step object detection networks. And it is compatible with other improved methods of RPN. In conclusion, our method improves state-of-the-art object detection not only accuracy but also stability.

## References

1. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 328–335 (2014)
2. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: binarized normed gradients for objectness estimation at 300fps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3286–3293 (2014)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
4. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147–2154 (2014)
5. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_23
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Kong, T., Yao, A., Chen, Y., Sun, F.: HyperNet: towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 845–853 (2016)
9. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. arXiv preprint arXiv:1612.03144 (2016)
10. 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
11. Opitz, M., Waltner, G., Poier, G., Possegger, H., Bischof, H.: Grid loss: detecting occluded faces. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 386–402. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_24
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
13. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)

15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
17. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 391–405. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_26