Object Detection with R-FCN

Zhaoyan Lin
Smith College
Northampton, MA
zlin@smith.edu

Abstract

Region-based Fully Convolutional Network (R-FCN) [1] is a variation of the state-of-the-art region-based object detection networks Faster R-CNN[9]. With its fully convolutional architecture, R-FCN gets higher accuracy and efficiency than Faster R-CNN. I train R-FCN on the 20 classes of objects in PASCAL VOC dataset[2, 3] with a 50-layer Residual Network (ResNet) [6] as backbone on one GPU. It gets mAP of 0.752 using Selective Search (SS) region proposals, and mAP of 0.774 using Region Proposal Network (RPN).

1. Introduction

In last few years, there has been a great progress in the accuracy of object detection using deep neural networks. However, compared to image classification, object detection is obviously more challenging and the detection accuracy is much worse than classification accuracy.

A major difference between image classification and object detection lies in the translation variance of objects, which is a desirable feature for image classification, but not for object detection. For object detection, the precise location of each object matters. In order to get candidate bounding boxes, previous region-based detectors have used different methods, including Selective Search (SS) [5] and Region Proposal Network (RPN) [9].

In this paper, R-FCN is experimented on training and testing on PASCAL VOC datasets using both SS and RPN. They each result in 75.2% and 77.4% mAPs. The bounding boxes after different numbers of iterations are visualized for analysis.

2. Related Work

After AlexNet[7] showed high classification accuracy in ILSVRC 2012, researchers have seen the potential of using CNNs in object detection tasks and received significantly higher detection accuracy when region-based CNN (R-CNN) [5] first came out to close the gap between classification and detection. The main idea of it is to use a two-stage detection, which consists of two steps: (1) candidate region proposals of where are likely to have objects, (2) CNN for classification. R-CNN uses an external algorithm, Selective Search, to generate the bounding box proposals. Selective Search functions like a bottom-up segmentation that groups pixels together. Then the patches are extracted from the row image to forward through CNN, resulting in repeated CNN computation for overlapping proposals.

Based on R-CNN, Fast R-CNN [4] makes a change to train bounding box regression and final classification together. The training of these in R-CNN is separate. Fast R-CNN only passes the entire image through CNN once by extracting patches from the convolutional feature map.

Therefore, the training of Fast R-CNN gets much faster.

Faster R-CNN [9] further increases the sharing of convolutional computation. Now box proposals and classification share convolutional computation. It uses Region Proposal Network for region proposals, a small network integrated with the CNN network.

Fast R-CNN [4] and Faster R-CNN [9] save computation by sharing repeated convolutional computation for object classification and region proposals, respectively. Given the idea that more sharing improves detection, R-FCN [1] gets rid of the last few fully connected layers, making it fully convolutional. Therefore, it lowers the computation for each region proposal and saves more time.

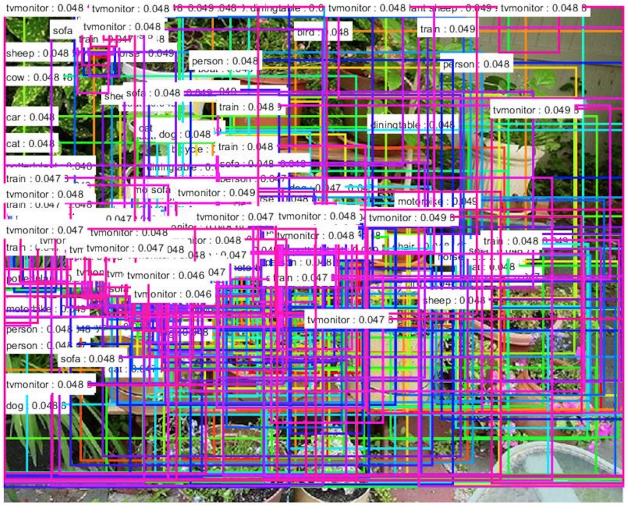

Figure 1. Example of Bounding Box Proposals at Original State

3. Preparation

3.1. Setup

The experiments require the use of GPU because of the high computing needed for a deep network. Therefore, the experiments are done on a lab computer with Windows 10 system and one Nvidia GTX 1080 GPU. Other necessary installations include CUDA 9.1, cuDNN v4, Visual Studio 2017 Community v15.4.5, Matlab R2018a and Caffe package compiled under Windows. All the mentioned versions have been tested out to finally realize the implementation. The Matlab code is published by the auther [1].

3.2. Dateset

I use datasets PASCAL VOC 2007 and 2012 [2, 3], which is a visual object challenge widely used for testing detectors' accuracy. PASCAL VOC 2007 [2] has 5,011 images with 12,608 objects in them, while PASCAL 2012 [3] has 11,540 images with 27,450 objects in them. Both datasets have images of 20 classes: aeroplanes, bicycles, birds, boats, bottles, buses, cars, cats, chairs, cows, dining tables, dogs, horses, motorbikes, people, potted plants, sheep, sofas, trains, tv/monitors.

[1] https://github.com/daijifeng001/r-fcn

3.3. Model Details

The backbone used is the state-of-the-art deep network Residual Network. Here I use a pretrained 50-layer ResNet [6] trained on ImageNet. ImageNet is complicated enough and generally performs well to be used in transfer learning.

The region proposals are also pretrained with Selective Search and Region Proposal Network. For each image, there are 300 predicted bounding boxes. An example of the original state of the bounding boxes is shown in Figure 1.

4. Experiments

4.1. Experiments on PASCAL VOC

I perform two complete experiments on PASCAL VOC 2007 and 2012 datasets [2, 3]. The training uses both datasets and testing uses only VOC 2007. The maximum iteration number in both experiments is set as 110,000.

The first experiment uses region proposals generated by SS. The training takes 15 hours on one GPU and results in mAP of 75.2%. Average testing time per image is 0.177s. The average preci-

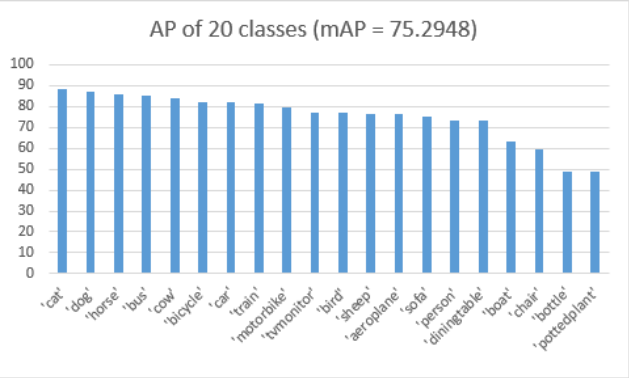

Figure 2. APs of 20 classes using SS

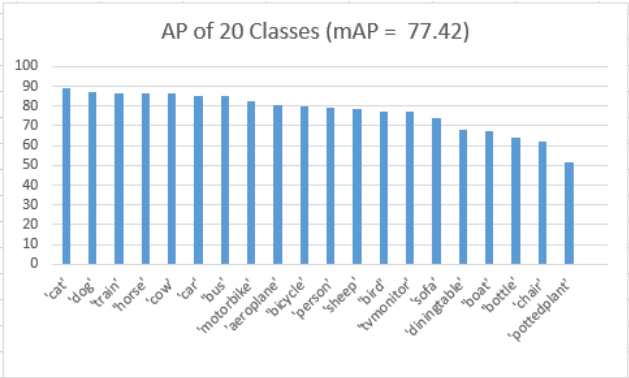

Figure 3. APs of 20 classes using RPN

sions for each of the 20 categories are ranked from highest to lowest in Figure 2 below. Cats ranked first, while potted plants appear at last.

The second experiment uses region proposals pretrained by RPN. The training is done in 12 hours and mAP on testing is 77.4%. Average testing time per image is 0.120s. As in Figure 3 , the highest AP for detection happens on cats and potted plants have the lowest AP.

In both cases, animals like cats, dogs and horses are detected relatively accurately, with mAP approaching 90%. Potted plants ranked last in both: only around half of them are detected. Other furniture like chairs, sofas and diningtables is also among the ones with the lowest accuracy.

Besides, there are two unfinished training. The first one end when the storage runs out on the computer. During training, it produces around 100GB cache because all the feature maps need to be stored on disk. For this experiment, the accuracy at the last iteration (96,000th) is 89.2% on training. The other one is intended to test the GPU, which ends in 20,000 iterations in two hours and reaches mAP of 65.6%.

4.2. Comparison between SS and RPN

As expected, using RPN proposals results in two percent higher mAP than SS. Also, training with RPN proposals is three hours shorter than with SS on one GPU. It aligns with why Faster R-CNN [9] starts to use RPN over SS for region proposals: RPN saves time by sharing convolutional computation.

4.3. Visualization of Bounding Boxes

I visualize (Figure 4, 5) the bounding boxes with category-wise scores after every 10,000 iterations. The threshold score is 0.6. The two training processes using SS and RPN are visualized on the image of several potted plants on a shelf. Potted plants get the worst detection results for both experiments. Therefore, even at the end of training, the bounding boxes are generally misplaced and the scores for each are not very high. The images show the progress of bounding boxes refining their locations and sizes. Probably because potted plants are placed together too closely, it is hard to figure out the bound between two plants, so in some conditions, the boxes bound more than one plants together. Furthermore, the color of the wood shelf might also be quite confusing.

Besides, the visualization of the RPN version shows that the bounding boxes with scores higher than 0.6 appear faster than the SS version. More plants are detected in a shorter time when using RPN proposals and the final result is also better.

4.4. Some Results

Some of the detection results are shown in Figure 6 here. The scores for cats and dogs are pretty high. Even cats in unusual poses can be detected with lower scores. Potted plants prove to be a re-

Figure 4. Visualization of Bounding Boxes in Training with SS

ally hard task for the detector. Some results also indicate that the detection accuracy tends to be higher when the background is clearly different from the objects. In contrast, when the color or texture of the background is similar to the objects, the detector is not very confident. Similar issues also hold true in image classification cases. Sometimes the overlapping of multiple objects will make bounding boxes larger than they should be. The problem probably happens in the process of bounding box regression.

5. Conclusion and Future Work

Object detection using R-FCN with ResNet-50 gets mAP above 75%. With more layers of ResNet, the result could be better, but the running time would be longer at the same time. RPN proposals improve the performance and efficiency. However, the improvement is not very remarkable. More research on an even better region proposal method might benefit object detection.

The idea of that more sharing of convolution results in better performance begins from Fast R-CNN [4]. However, it seems to be based on experiments without actual theories now. With a little more sharing of convolutional layers, R-FCN [1] does not show a significant improvement in the accuracy on the previous Faster R-CNN [9]. It admittedly saves some computing time for training. However, the overall performance is only slightly better. Although object detection accuracy is now higher, more changes probably could be made on aspects other than the sharing of convolution.

Fully Convolutional Networks (FCN) [8] are

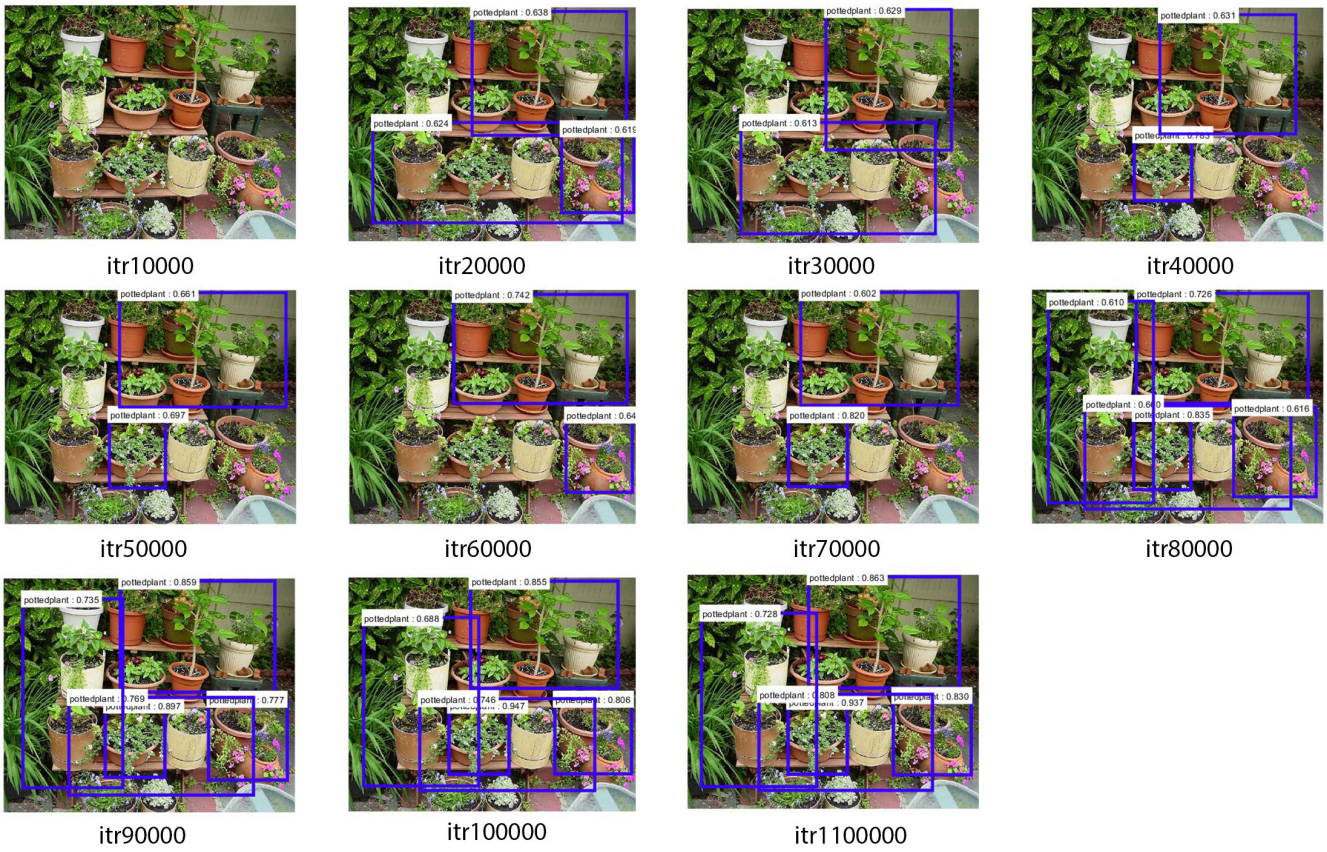

Figure 5. Visualization of Bounding Boxes in Training with RPN

first used to solve semantic segmentation problems. It might be an interesting idea to find how segmentation plays in detection problems.

Regard of future work, I can implement the model trained by VOC 2007 and 2012 on other datasets to see it could work generally. Furthermore, I can train R-FCN on larger dataset like ImageNet with 200 classes of objects. It could also be interesting to implement it on datasets with a specific focus, like KITTI dataset that consists mainly of photos taken on the road. Since object detection is a key requirement for driverless cars, KITTI should be a good choice to test the method on.

Besides, other than using ResNet as the backbone, I can also try using other networks resulting in high accuracy in image classification, such as GoogLeNet [11] and VGGNet [10]. It would be meaning to observe how the structure of the network affects the performance.

Current object detectors still have much for improvement. The field of object detection is still out there for us to explore.

References

[1] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

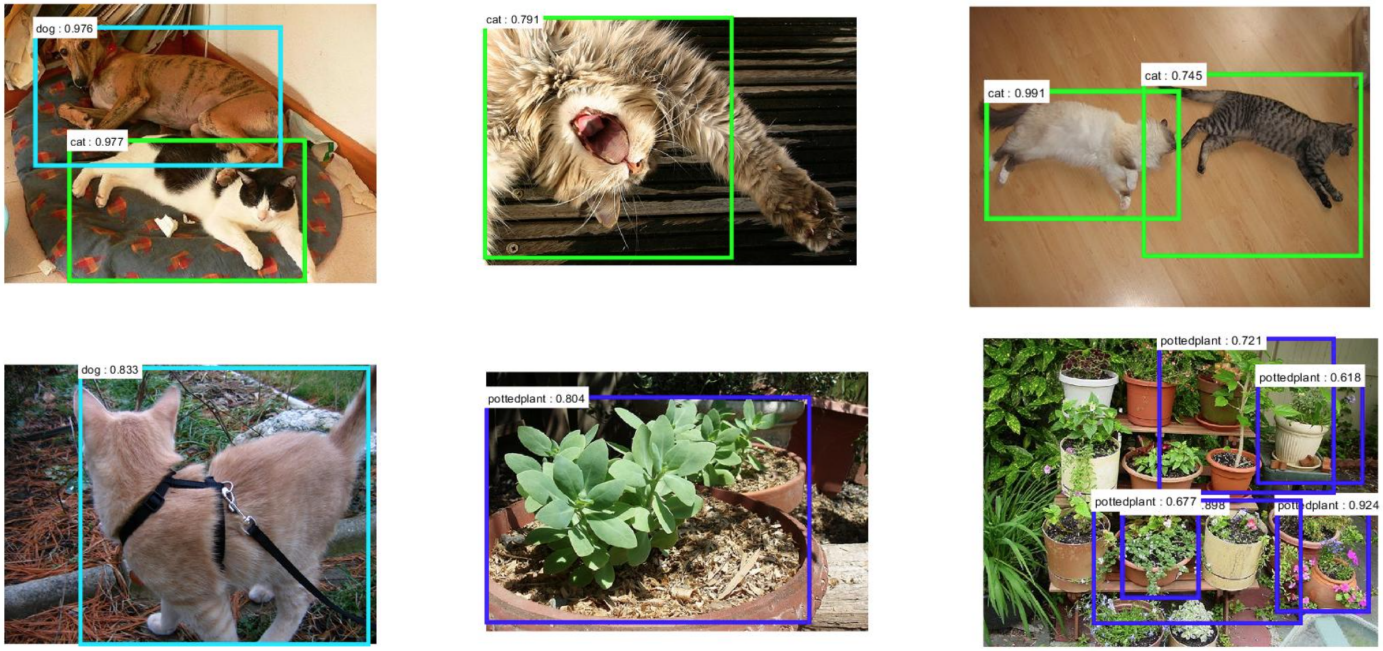

Figure 6. Some Testing Images

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[4] R. Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *Cvpr*, 2015.