Faster R-CNN

Hongzhi Liu

July 27, 2018

Abstract

Recent advances in object detection are driven by the success of region proposal methods and region-based convolutional neural networks (R-CNNs). During preparation for URPC2018, I read a thesis written by Shaoqing Ren and Kaiming He who are from Microsoft Research. This paper introduces a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. For the very deep VGG-16 model, their detection system has a frame rate of 5fps on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP) and 2012 (70.4% mAP) using 300 proposals per image.

1. Overview of Faster R-CNN

Fast R-CNN, achieves near real-time rates using very deep networks, when ignoring the time spent on region proposals [2]. Now, proposals are the computational bottleneck in state-of-the-art detection systems. One may note that fast region-based CNNs take advantage of GPUs, while the region proposal methods used in research are implemented on the CPU, making such runtime comparisons inequitable. An obvious way to accelerate proposal computation is to re-implement it for the GPU. This may be an effective engineering solution, but re-implementation ignores the downstream detection network and therefore misses important opportunities for sharing computation.

In this paper, Shaoqing Ren and Kaiming He show that an algorithmic changecomputing proposals with a deep netleads to an elegant and effective solution, where proposal computation is nearly cost-free given the detection networks computation. To this end, they introduce novel Region Proposal Networks (RPNs) that share convolutional layers with state-of-the-art object detection networks [3, 2]. By sharing convolutions at test-time, the marginal cost for computing proposals is small.

They evaluate the method on the PASCAL VOC detection benchmarks [1], where RPNs with Fast R-CNNs produce detection accuracy better than the strong baseline of Selective Search with Fast R-CNNs. Meanwhile, their method waives nearly all computational burdens of SS at test timethe effective running time for proposals is just 10 milliseconds.

2. Region Proposal Networks

A Region Proposal Network (RPN) takes an image as input and outputs a set of rectangular object proposals, each with an objectness score. Shaoqing Ren and his team model this process with a fully convolutional network [4], which they describe in this section. Because their ultimate goal is to share computation with a Fast R-CNN object detection network [2], they assume that both nets share a common set of conv layers. In the experiments, they investigate the Zeiler and Fergus model (ZF) [8], which has 5 shareable conv layers and the Simonyan and Zisserman model (VGG)[7], which has 13 shareable conv layers.

To generate region proposals, Ren slides a small network over the conv feature map output by the last shared conv layer. This network is fully connected to an $n \times n$ spatial window of the input conv feature map. Each sliding window is mapped to a lower-dimensional vector. This vector is fed into two sibling fully-connected layersa box regression layer and a box classification layer. They use $n = 3$ in this paper, noting that the effective receptive field on the input image is large. This mini-network is illustrated at a single position in Fig. 1(left). Note that because the mini-network operates in a sliding-window fashion, the fully-connected layers are shared across all spatial locations. This architecture is naturally implemented with an $n \times n$ conv layer followed by two sibling 1×1 conv layers. ReLUs are applied to the output of the $n \times n$ conv layer [5].

2.1. Translation-Invariant Anchors

At each sliding-window location, Ren and his team simultaneously predict k region proposals, so the reg layer has 4k outputs encoding the coordinates of k boxes. The cls layer outputs 2k scores that estimate probability of object / not-object for each proposal. The k proposals are parameterized relative to k reference boxes, called anchors. Each

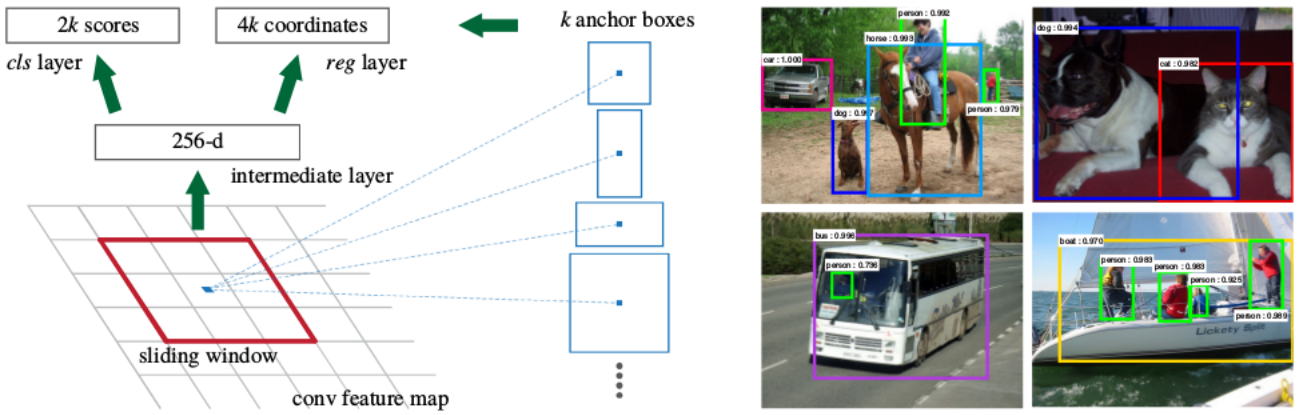

Figure 1. **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Their method detects objects in a wide range of scales and aspect ratios.

anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio. They use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position. For a conv feature map of a size $W \times H$, there are W Hk anchors in total. An important property of their approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors.

2.2. A Loss Function for Learning Region Proposals

For training RPNs, Ren and his team assign a binary class label (of being an object or not) to each anchor. They assign a positive label to two kinds of anchors: (i) the anchor/anchors with the highest Intersection-over-Union (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. Note that a single ground-truth box may assign positive labels to multiple anchors. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective.

With these definitions, they minimize an objective function following the multi-task loss in Fast R-CNN [2]. Their loss function for an image is defined as Eq. 1:

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

Here, i is the index of an anchor in a mini-batch and p_i is the predicted probability of anchor i being an object. The ground-truth label p_i^* is 1 if the anchor is positive, and is 0 if the anchor is negative. t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box, and t_i^* is that of the ground-truth box associated with a positive anchor. The classification loss L_{cls} is log loss over two classes. For the regression loss, the team use $L_{reg}(t_i, t_i^*) = R(t_i, t_i^*)$ where R is the robust loss function defined in [2]. The term $p_i L_{reg}$ means the regression loss is activated only for positive anchors $p_i = 1$ and is disabled otherwise $p_i = 0$. The outputs of the cls and reg layers consist of p_i and t_i respectively. The two terms are normalized with N_{cls} and N_{reg}, and a balancing weight λ.

2.3. Optimization

The RPN, which is naturally implemented as a fully-convolutional network, can be trained end-to-end by back-propagation and stochastic gradient descent (SGD). Ren and his team follow the "image-centric" sampling strategy from [2] to train the network. Each mini-batch arises from a single image that contains many positive and negative anchors. It is possible to optimize for the loss functions of all anchors, but this will bias towards negative samples as they are dominate. Instead, they randomly sample 256 anchors in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:1. If there are fewer than 128 positive samples in an image, they pad the mini-batch with negative ones.

They randomly initialize all new layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. All other layers are initialized by pre-training a model for ImageNet classification [6], as is standard practice. The team tune all layers of the ZF net, and conv3_1 and up for the VGG net to conserve memory [2]. They use a learning rate of 0.001 for 60k mini-batches, and 0.0001 for the next 20k mini-batches on the PASCAL dataset. They also use a momentum of 0.9 and a weight decay of 0.0005.

3. Experiments

Ren and his team comprehensively evaluate their method on the PASCAL VOC 2007 detection benchmark [1]. This

Table 1. Detection results on **PASCAL VOC 2007 test set**(trained on VOC 2007 trainval). The detectors are Fast R-CNN with ZF, but using various proposal methods for training and testing.

train-time region proposals		test-time region proposals		
method	# boxes	method	# proposals	mAP(%)
SS	2K	SS	2K	58.7
EB	2K	EB	2K	58.6
RPN+ZF, shared	2K	RPN+ZF, shared	300	59.9

dataset consists of about 5k trainval images and 5k test images over 20 object categories. They also provide results in the PASCAL VOC 2012 benchmark for a few models. For the ImageNet pre-trained network, they use the fast version of ZF net [8] that has 5 conv layers and 3 fc layers, and the public VGG-16 model 5 that has 13 conv layers and 3 fc layers. They primarily evaluate detection mean Average Precision (mAP), because this is the actual metric for object detection.

Tab. 1 shows Fast R-CNN results when trained and tested using various region proposal methods. These results use the ZF net. For Selective Search, the team generate about 2k SS proposals by the fast mode. For EdgeBoxes, they generate the proposals by the default EB setting tuned for 0.7 IoU. SS has an mAP of 58.7% and EB has an mAP of 58.6%. RPN with Fast R-CNN achieves competitive results, with an mAP of 59.9% while using up to 300 proposals.

References

[1] M. Everingham, A. Zisserman, C. K. I. Williams, L. V. Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, and G. Dork. The PASCAL visual object classes challenge 2007 (VOC2007) results. In *ICML*, 2005. 1, 2

[2] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 2

[3] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE TPAMI*, 2015. 1

[4] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[5] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 1

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 2

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 1

[8] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1, 3