Lab 3: Faster R-CNN pre-report

CHI YEONG HEO[1],
[1]School of Computer Science and Engineering, Pusan National University, Busan 46241 Republic of Korea

1. Introduction

This report provides a summary of a paper in the field of computer vision: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[5].

2. Overview of Faster R-CNN

Faster R-CNN is a object detection model, which is built upon previous objects detection models like R-CNN and Fast R-CNN. Unlike its predecessors, Faster R-CNN introduces a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. By merging the RPN with the Fast R-CNN detector into a single, unified network, Faster R-CNN achieves enhanced efficiency at region proposal process.

3. Limitations of Prior Methods

Previous object detection systems like R-CNN[3] and Fast R-CNN[2] relied on external object proposal methods such as Selective Search (SS) [8] and EdgeBoxes (EB) [9] to generate region proposals. These methods, based on grouping **super-pixels** or **sliding windows**, required substantial computational resources, which limited the speed of the overall detection.

Methods like OverFeat[7] and MultiBox[1] sought to address some of the limitations by predicting bounding boxes directly. However, these models either focused on single-object localization or used class-agnostic boxes as proposals without effectively sharing features between the proposal and detection stages.

4. Region Proposal Networks

The Region Proposal Network (RPN) accepts an image of arbitrary size and outputs a set of region proposals, each assigned an objectness score. This process is implemented using a fully convolutional network (FCN)[4], enabling the RPN and Fast R-CNN detector to share a common set of convolutional layers to produce feature maps.

To generate region proposals, a small network slides over the convolutional feature map produced by the last shared convolutional layer. At each spatial position, this sliding window captures an $n \times n$ region from the convolutional feature map, which is mapped to a lower-dimensional feature vector. This feature vector is then input into two sibling fully connected layers—a box-regression layer (*reg*) and a box-classification layer (*cls*). This architecture is implemented by an $n \times n$ convolutional layer followed by two sibling 1×1 convolutional layers for the *reg* and *cls* layers, respectively.

4.1. Anchors

At each position, the RPN generates multiple region proposals, with the maximum number of proposals per location denoted by k. The *reg* layer produces $4k$ outputs that encode the coordinates of k bounding boxes, while the *cls* layer outputs $2k$ scores that estimate the likelihood of each proposal containing an object. These proposals are parameterized relative to k reference boxes, referred to as *anchors*. Each anchor is centered at the current sliding window position and is associated with a predefined scale and aspect ratio. For a convolutional feature map of size $W \times H$, there are WHk anchors in total.

Translation-Invariant Anchors: RPN can predict the proposal even if an object is translated. This property also reduces the model size. MultiBox has a $(4+1) \times 800$-dimensional fully-connected output layer, whereas Faster R-CNN has a $(4+2) \times 9$-dimensional convolutional output layer($k = 9$).

Multi-Scale Anchors as Regression References: Faster R-CNN utilizes a **pyramid of anchors** method, which is more cost-efficient than image/feature pyramids or sliding windows of multiple scales. It only relies on feature maps of a single scale, and uses filters of a single size.

4.2. Loss Function

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

Here, i indexes an anchor in a mini-batch, with p_i representing the predicted probability of anchor i being an object. The ground-truth label p_i^* is 1 for positive anchors and 0 for negative ones. t_i and t_i^* denote the predicted and ground-truth bounding box coordinates for positive anchors, respectively. The classification loss L_{cls} is a binary log loss, while the regression loss $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ measures the bounding box error where R is a smooth L_1 loss function[2]. L_{reg} is applied only to positive anchors, as indicated by the term $p_i^* L_{reg}$. The losses are normalized by N_{cls} and N_{reg} and balanced by λ. By default, $\lambda = 10$ to weight the classification and regression terms roughly equally.

5. Sharing Features for RPN and Fast R-CNN

4-Step Alternating Training. The shared convolutional layers are initialized by a model pre-trained for ImageNet classification[6]. All new layers are randomly initialized with $\mathcal{N}(0, 0.01^2)$. 1. The RPN is trained end-to-end by stochastic gradient descent (SGD). 256 anchors are sampled to compute the loss of a mini-batch, which ratio of positive and negative is 1:1. 2. The detector network is trained using the proposals from step-1 RPN without sharing convolutional layers. 3. The RPN is trained using detector network while only fine-tuning the RPN layers. Now the two networks share convolutional layers. 4. Only the detector network is fine-tuned.

6. Comparison

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	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**

Table 2: **Timing** (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. "Region-wise" includes NMS, pooling, fully-connected, and softmax layers.

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

As illustrated in Table 1, Faster R-CNN's Region Proposal Network (RPN) achieves a mean Average Precision (mAP) of 59.9%, surpassing Selective Search (SS) and EdgeBoxes (EB) despite utilizing only up to 300 proposals. This efficiency is attributable to the RPN's shared convolutional features, which reduce redundant computations.

Timing results in Table 2 further underscore the efficiency of Faster R-CNN. With the RPN and Fast R-CNN combination, computation time is significantly reduced compared to SS-based methods. Faster R-CNN achieves up to 5 frames per second (fps) with VGG, and up to 17 fps with ZF, due to minimized region proposal time and shared feature maps, which substantially lower the computational cost of region-wise processing.

[1] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[2] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[7] P Sermanet. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[8] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013.

[9] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 391–405. Springer, 2014.