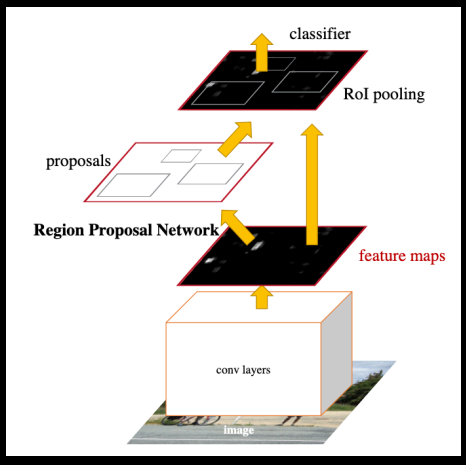
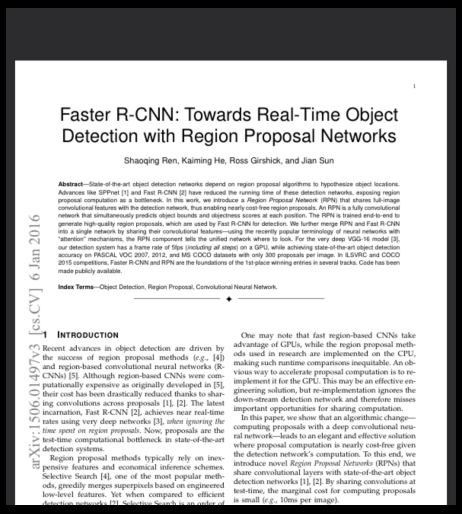
Faster R-CNN (Region-based Convolutional Neural Network) is an advanced object detection model that improves upon previous architectures like Fast R-CNN and R-CNN by introducing a Region Proposal Network (RPN). It is widely used for tasks such as object detection, image segmentation, and instance segmentation.





Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

Index Terms—Object Detection, Region Proposal, Convolutional Neural Network.

1 Introduction

Recent advances in object detection are driven by the success of region proposal methods (*e.g.*, [4]) and region-based convolutional neural networks (R-CNNs) [5]. Although region-based CNNs were computationally expensive as originally developed in [5], their cost has been drastically reduced thanks to sharing convolutions across proposals [1], [2]. The latest incarnation, Fast R-CNN [2], achieves near real-time rates using very deep networks [3], *when ignoring the time spent on region proposals*. Now, proposals are the test-time computational bottleneck in state-of-the-art detection systems.

Region proposal methods typically rely on inexpensive features and economical inference schemes. Selective Search [4], one of the most popular methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks [2], Selective Search is an order of

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 down-stream detection network and therefore misses important opportunities for sharing computation.

In this paper, we show that an algorithmic change—computing proposals with a deep convolutional neural network—leads to an elegant and effective solution where proposal computation is nearly cost-free given the detection network's computation. To this end, we introduce novel *Region Proposal Networks* (RPNs) that share convolutional layers with state-of-the-art object detection networks [1], [2]. By sharing convolutions at test-time, the marginal cost for computing proposals is small (*e.g.*, 10ms per image).

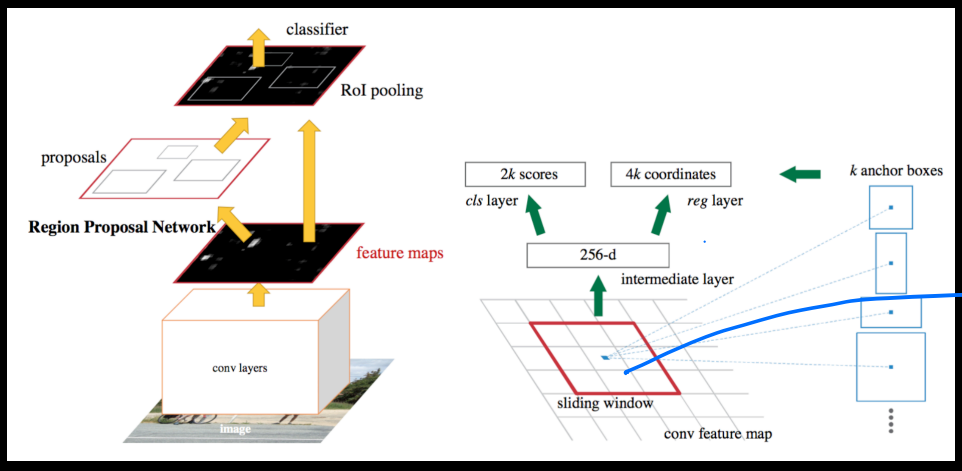
Architecture

1. Convolution Backbone
2. Region Proposal Network
   Fully Conv Network ⟶ generate region proposals

Use Anchor Boxes to detect at multiple scale

3. ROI Pooling ⟶ Resize Region Proposals
4. FC layers ⟶ Classification
   ⟶ Regression

RPN Mechanism



predict region proposal

1. Sliding Window
   → predict region proposal

2. 256 d Intermediate

3. Classification & Regression layer
   - Classification → 2K scores
     - K = 9
     - no of anchors
       - Contains Object / Foreground
       - Background
   - Regression → 4K
     - Refine the bbox coordinates
     - Adjust anchor boxes to fit objects

Anchor Box ⟶ Sliding Window

Scale

Aspect Ratio

3 Scales and 3 Aspect Ratios

Convolutional Feature

W x H (~2400), NHK

Advantages

1) High detection accuracy
2) End to End Trainable
3) Efficient RPN
4) Handle multiple scales

Disadvantages

1) Computationally Expensive

2) Real Time X

3) Sensitive to small object