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

In this paper authors 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. R-CNN is slow because it applies layers of CNN on each box proposal to extract the image feature to classify objects in the box. If there are 2000 box proposal the CNN should be applied 2000 times to extract the features and this makes the overall process slow. To solve this problem, Fast R-CNN model makes a different approach. Instead of applying convolutional in each object proposal it applies conv layer once on an image and extracts features but the bottleneck is in the region proposal stage.Authors of this paper solves this problem with FasterR-CNN.They merge RPN and Fast R-CNN into a single network by sharing their convolutional features.

Model Architecture and Working Process :

The model is the enhanced version of Fast R-CNN model with Region Proposal Network. It consists of two modules.

1. Region Proposal Network (RPN) for generating region proposals
2. Fast R-CNN for detecting objects in the proposed region.

The RPN module is responsible for generating region proposals. It applies the concept of attention mechanism in neural networks. RPNs are designed to efficiently predict region proposals with a wide range of scales and aspect ratios. In contrast to prevalent methods that use pyramids of images or pyramids of filters. RPN guides the Fast R-CNN module to where to look for objects in the image. To unify RPNs with Fast R-CNN object detection networks,authors propose a training scheme that alternates between fine-tuning for the region proposal task and then fine-tuning for object detection, while keeping the proposals fixed. This scheme converges quickly and produces a unified network with convolutional features that are shared between both tasks. Translation-Invariant Anchors is an important property of this approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. The translation-invariant property also reduces the model size. MultiBox has a (4 + 1) × 800-dimensional fully-connected output layer, whereas this method has a (4 + 2) × 9-dimensional convolutional output layer in the case of k = 9 anchors. As a result, output layer has 2.8 × 104 parameters (512 × (4 + 2) × 9 for VGG-16), two orders of magnitude fewer than MultiBox's output layer that has 6.1 × 106 parameters (1536 × (4 + 1) × 800 for GoogleNet in MultiBox.

Images fed into the convolutional layer and outputs a feature map and to make region proposals a small layer of convolution is applied on this extracted feature map. This layer of convolution generates box proposals. 3x3 window traverses through the feature map and at each position it

outputs a box and class information. At each output point it does not produce only one box but K number of boxes. The K classes output the probability of the box containing an object or not from the relative coordinate position of anchor boxes. In this paper authors define the value of K=9 to make anchor boxes .For anchors, authors use 3 scales with box areas of 128x128 , 256x256 , and 512x512 pixels, and 3 aspect ratios of 1:1, 1:2, and 2:1.

Training:

In this paper authors mentioned 3 different ways to train both the RPN and Fast R-CNN while sharing the convolutional layers.

1. Alternative Training
2. Approximate Joint Training
3. Non Approximate Joint Training

The training of the Faster R-CNN model is divided into four steps. First the RPN is trained on the object detection dataset to make region proposals. In the second step, they train a separate detection network by Fast R-CNN using the proposals generated by the step-1 RPN. This detection network is also initialized by the ImageNet-pre-trained model. At this point the two networks do not share convolutional layers. In the third step, authors use the detector network to initialize RPN training but they fix the shared convolutional layers and only fine-tune the layers unique to RPN. Now the two networks share convolutional layers. Finally, keeping the shared convolutional layers fixed, authors fine-tune the unique layers of Fast R-CNN.
An anchor is considered to be a “positive” if it has highest IOU with a ground truth box or an anchor box that has IOU higher than 0.7 with any ground truth box. If an anchor has an IOU ratio less than 0.3 then it is considered as “negative”.

Results :

In all the experiments on the PASCAL datasets, Fast R-CNN was chosen as a detector. The use of the RPN+ZF backbone as just a proposal network matched the performance of using “Selective Search” (SS) as a region proposal algorithm. Detection with a VGG RPN takes 198ms compared to the 1.8 seconds of Selective Search. Faster R-CNN has a frame rate of 5fps 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. SS has an mAP of 58.7% and EB has an mAP of 58.6% under the Fast R-CNN framework. RPN with Fast R-CNN achieves competitive results, with an mAP of 59.9% while using up to 300 proposals

Contributions :

1. Proposing region proposal network

2. This paper introduced the concept of anchor boxes rather than using pyramid images or pyramids of filters.
3. The convolutional computations are shared across the RPN and the Fast R-CNN. This reduces the computational time.

Limitations :

The RPN is trained where all anchors in the mini-batch, of size 256, are extracted from a single image. Because all samples from a single image may be correlated, the network may take a lot of time until reaching convergence.