**FASTER RCNN MODEL (Architecture/ Overview Document)**

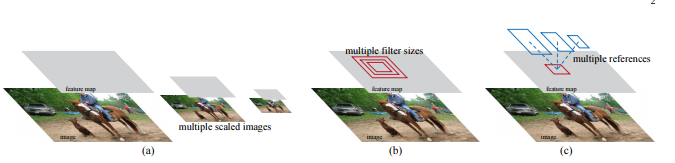
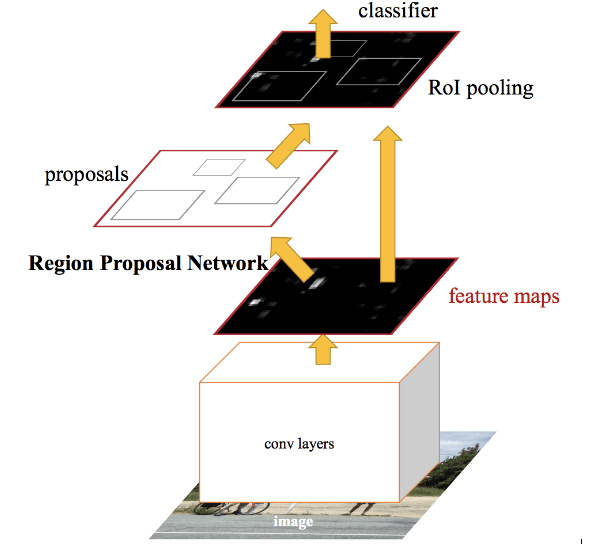
****

Fig1

****

**Fig2**

**MODEL INTRODUCTION:**

Our object detection system, called ***Faster R-CNN***, is composed of two modules. The first module is a deep fully convolution network that proposes regions, and the second module is the Fast R-CNN detector,that uses the proposed regions. The entire system is a single, unified network for object detection.

The following are the designs and properties of the network for region proposal followed by the development of algorithms for training both modules with features shared.

Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.

**MODEL STRUCTURE**

1. Region Proposal Networks:

* A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score. We model this process with a fully convolution network.
* Our ultimate goal is to share computation with a Fast R-CNN object detection network , we assume that both nets share a common set of convolution layers. Experiments investigate the Zeiler and Fergus model (ZF), which has 5 shareable convolutional layers and the Simonyan and Zisserman model (VGG-16), which has 13 shareable convolutional layers.
* To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes as input an n × n spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature (256-d for ZF and 512-d for VGG, with ReLU.
* This feature is fed into two sibling fullyconnected layers—a box-regression layer (reg) and a box-classification layer (cls).
* This architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively).
* ANCHORS: At each sliding-window location, we simultaneously predict multiple region proposals, where the number of maximum possible proposals for each location is denoted as k. So the reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal . The k proposals are parameterized relative to k reference boxes, which we call.
* For simplicity we implement the cls layer as a two-class softmax layer. Alternatively, one may use logistic regression to produce k scores, example: 4 car : 1.000 dog : 0.997 person : 0.992 person : 0.979 horse : 0.993
* ***Comparion between SSD Multibox and Faster RCNN***:
* An important property of our approach is that it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location.
* As a comparison, the MultiBox method uses k-means to generate 800 anchors, which are not translation invariant. So MultiBox does not guarantee that the same proposal is generated if an object is translated.
* If considering the feature projection layers, Faster Rcnn proposal layers still have an order of magnitude fewer parameters than MultiBox . We expect this method to have less risk of overfitting on small datasets.

* **Training RPN’s**: The RPN can be trained end-to-end by back propagation and stochastic gradient descent (SGD). We follow the “image-centric” sampling strategy to train this network. Each mini-batch arises from a single image that contains many positive and negative example 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, we 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, we pad the mini-batch with negative ones.
* Represented RPNs for efficient and accurate region proposal generation. By sharing convolutional features with the down-stream detection network, the region proposal step is nearly cost-free. This method enables a unified, deep-learning-based object detection system to run at near real-time frame rates. The learned RPN also improves region proposal quality and thus the overall object detection accuracy.

**MODEL OVERVIEW:**

* Take an input image and pass it to the ConvNet which returns feature maps for the image
* Apply Region Proposal Network (RPN) on these feature maps and get object proposals
* Apply ROI pooling layer to bring down all the proposals to the same size
* Finally, pass these proposals to a fully connected layer in order to classify any predict the bounding boxes for the image
* Faster RCNN replaces the selective search method with region proposal network (RPN) which makes the algorithm much faster.
* The approximate prediction time for each image is around 0.2 sec
* Faster RCNN takes the feature maps from CNN and passes them on to the Region Proposal Network. RPN uses a sliding window over these feature maps, and at each window, it generates **k** Anchor boxes of different shapes and sizes
* Anchor boxes are fixed sized boundary boxes that are placed throughout the image and have different shapes and sizes. For each anchor, RPN predicts two things.
* The first is the probability that an anchor is an object (it does not consider which class the object belongs to)
* Second is the bounding box regressor for adjusting the anchors to better fit the object
* We now have bounding boxes of different shapes and sizes which are passed on to the RoI pooling layer. Now it might be possible that after the RPN step, there are proposals with no classes assigned to them. We can take each proposal and crop it so that each proposal contains an object
* The feature maps are passed to a fully connected layer which has a softmax and a linear regression layer. It finally classifies the object and predicts the bounding boxes for the identified objects.