**SPRINT-6 (Faster-RCNN Model Architecture)**

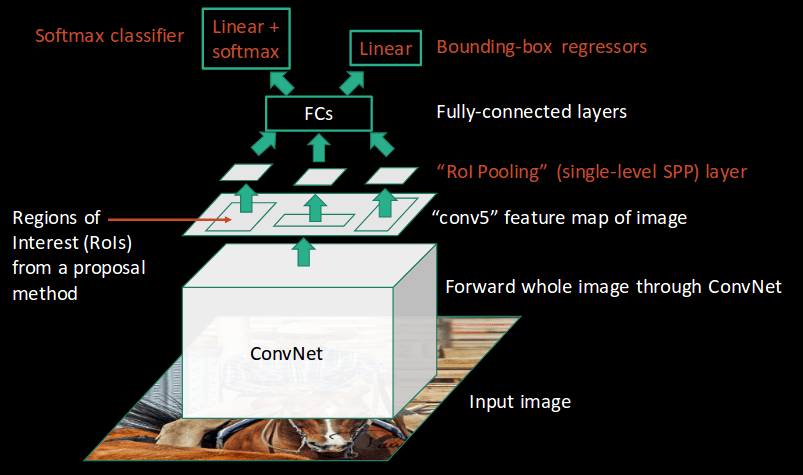
4.1. Intuition of Faster RCNN

Faster RCNN is the modified version of Fast RCNN. The major difference between them is that Fast RCNN uses selective search for generating Regions of Interest, while Faster RCNN uses “Region Proposal Network”, aka RPN. RPN takes image feature maps as an input and generates a set of object proposals, each with an objectness score as output.

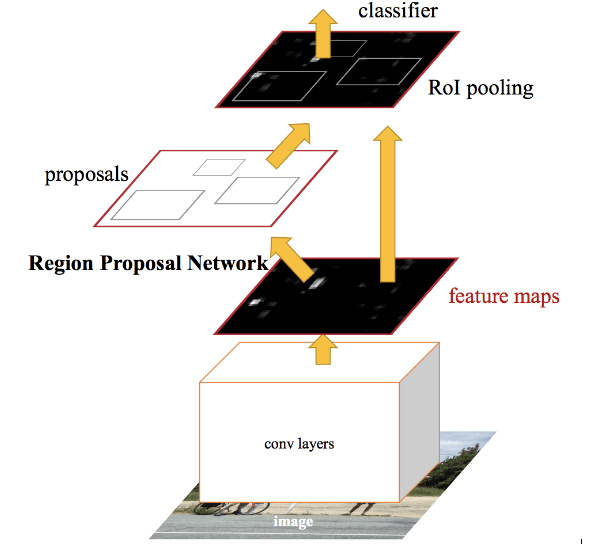
The  below steps are typically followed in a Faster RCNN approach:

1. We take an image as input and pass it to the ConvNet which returns the feature map for that image.
2. Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score.
3. A RoI pooling layer is applied on these proposals to bring down all the proposals to the same size.
4. Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

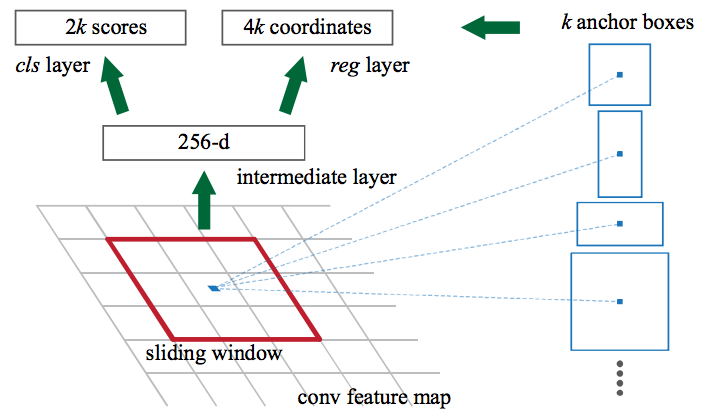
**FAST\_RCNN ARCHITECTURE**



**FASTER\_RCNN\_ARCHITECTURE**



To begin with, 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. This is what the RoI pooling layer does.

Then these 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.

**Comparison Of CNN ALGORITHMS :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Features** | **Prediction time / image** | **Limitations** |
| CNN | Divides the image into multiple regions and then classify each region into various classes. | – | Needs a lot of regions to predict accurately and hence high computation time. |
| RCNN | Uses selective search to generate regions. Extracts around 2000 regions from each image. | 40-50 seconds | High computation time as each region is passed to the CNN separately also it uses three different model for making predictions. |
| Fast RCNN | Each image is passed only once to the CNN and feature maps are extracted. Selective search is used on these maps to generate predictions. Combines all the three models used in RCNN together. | 2 seconds | Selective search is slow and hence computation time is still high. |
| Faster RCNN | Replaces the selective search method with region proposal network which made the algorithm much faster. | 0.2 seconds | Object proposal takes time and as there are different systems working one after the other, the performance of systems depends on how the previous system has performed. |