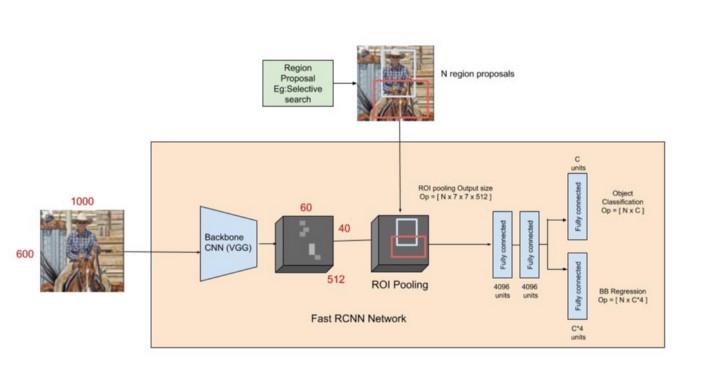
1)Describe the Quick R-CNN Architecture

.

Ans : Faster RCNN is an object detection architecture presented by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015, and is one of the famous object detection architectures that uses convolution neural networks like YOLO

The Fast R-CNN consists of a CNN (usually pre-trained on the ImageNet classification task) with its final pooling layer replaced by an “ROI pooling” layer and its final FC layer is replaced by two branches — a (K + 1) category softmax layer branch and a category-specific bounding box regression branch.



The entire image is fed into the backbone CNN and the features from the last convolution layer are obtained. Depending on the backbone CNN used, the output feature maps are much smaller than the original image size. This depends on the stride of the backbone CNN, which is usually 16 in the case of a VGG backbone.

Meanwhile, the object proposal windows are obtained from a region proposal algorithm like selective search[4]. As explained in Regions with CNNs, object proposals are rectangular regions on the image that signify the presence of an object.

The portion of the backbone feature map that belongs to this window is then fed into the ROI Pooling layer.

The ROI pooling layer is a special case of the spatial pyramid pooling (SPP) layer with just one pyramid level. The layer basically divides the features from the selected proposal windows (that come from the region proposal algorithm) into sub-windows of size h/H by w/W and performs a pooling operation in each of these sub-windows. This gives rise to fixed-size output features of size (H x W) irrespective of the input size. H and W are chosen such that the output is compatible with the network’s first fully-connected layer. The chosen values of H and W in the Fast R-CNN paper is 7. Like regular pooling, ROI pooling is carried out in every channel individually.

The output features from the ROI Pooling layer (N x 7 x 7 x 512 where N is the number of proposals) are then fed into the successive FC layers, and the softmax and BB-regression branches. The softmax classification branch produces probability values of each ROI belonging to K categories and one catch-all background category. The BB regression branch output is used to make the bounding boxes from the region proposal algorithm more precise.

2)Describe two fast R-CNN loss functions.

Ans : The classification branch of the softmax layer gives probabilities for every ROI over (K +1) categories p = p₀, … pₖ. The classification loss L𝒸ₗₛ(p,u) is given by -log(pᵤ) which is the log loss for the true class u.

Equation 1: smooth L1 loss for BB regression

Equation 2: Joint loss for multi-task training

The regression branch produces 4 bounding box regression offsets tᵏᵢ where i = x, y, w, and h. (x, y) stands for the top-left corner and w and h denote the height and width of the bounding box. The true bounding box regression targets for a class u are indicated by vᵢ where i = x, y, w, and h when u≥1. The case where u=0 is ignored because the background classes have no groundtruth boxes. The regression loss used is a smooth L1 loss given in equation 1. The joint multi-task loss for each ROI is given by the combination of the two losses as shown in equation 2. Notice that the Fast R-CNN here has a combined learning scheme that fine-tunes the backbone CNN, and classifies and regresses the bounding box.

3)Describe the disabilities of fast R-CNN.

Ans :This is the basic difference between the Fast R-CNN and Faster R-CNN. Faster R-CNN uses a region proposal method to create the sets of regions. Faster R-CNN possesses an extra CNN for gaining the regional proposal, which we call the regional proposal network.

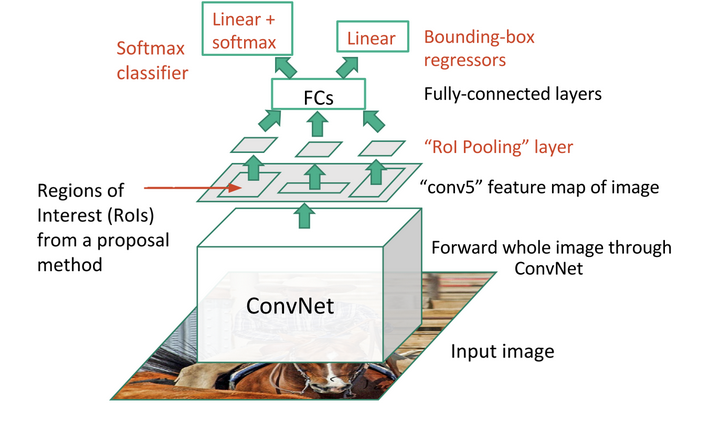
4)Describe how the area Proposal network works.

Ans :.This region proposal network takes convolution feature map that is generated by the backbone layer as input and outputs the anchors generated by sliding window convolution applied on the input feature map.

For each sliding window, the network generates the maximum number of k- anchor boxes. By the default the value of k=9 (3 scales of (128\*128, 256\*256 and 512\*512) and 3 aspect ratio of (1:1, 1:2 and 2:1)) for each of different sliding position in image. Therefore, for a convolution feature map of W \* H, we get N = W\* H\* k anchor boxes. These region proposals then passed into an intermediate layer of 3\*3 convolution and 1 padding and 256 (for ZF) or 512 (for VGG-16 ) output channels. The output generated from this layer is passed into two layers of 1\*1 convolution, the classification layer, and the regression layer. the regression layer has 4\*N (W \* H \* (4\*k)) output parameters (denoting the coordinates of bounding boxes) and the classification layer has 2\*N (W \* H \* (2\*k)) output parameters.

5)describe how the Roil pooling layer works.

Ans : ROI pooling solves the problem of fixed image size requirement for object detection network. ROI pooling produces the fixed-size feature maps from non-uniform inputs by doing max-pooling on the inputs. The number of output channels is equal to the number of input channels for this layer.



ROI pooling produces the fixed-size feature maps from non-uniform inputs by doing max-pooling on the inputs. The number of output channels is equal to the number of input channels for this layer. ROI pooling layer takes two inputs

1.A feature map obtained from a Convolutional Neural Network after multiple convolutions and pooling layers.

‘N’ proposals or Region of Interests from Region proposal network. Each proposal has five values, the first one indicating the index and the rest of the four are proposal coordinates. Generally, it represents the top-left and bottom-right corner of the proposal.

2.ROI pooling takes every ROI from the input and takes a section of input feature map which corresponds to that ROI and converts that feature-map section into a fixed dimension map. The output fixed dimension of the ROI pooling for every ROI neither depends on the input feature map nor on the proposal sizes, It solely depends on the layer parameters.

Layer Parameters: pooled\_width, pooled\_height, spatial scale.

Pooled\_width and pooled\_height are hyperparameters which can be decided based on the problem at hand. These indicate the number of grids the feature map corresponding to the proposal should be divided into. This will be the output dimension of this layer. Let us assume that W, H are the width and height of the proposal and P\_w,P\_h are pooled width and height. Then the ROI will be divided into P\_w\*P\_h blocks, each of dimensions (W/P\_w, H/P\_h).

bin\_size\_h = roi\_height/pooled\_height;

bin\_size\_w = roi\_width/pooled\_width

Spatial scale is a scaling parameter for resizing the proposal according to the feature map dimensions. Let's say in our network, the image size is 1056x640 and due to many convolution and pooling operations, the feature map size reduced to 66x40, which is being used by ROI pooling. Now the proposals are generated based on input image size, so we need to rescale the proposals to feature map size. In this case, we can divide all dimensions of proposal by 16 (1056/66=16 or 640/40=16). So the spatial scale will be 1/16 .

int roi\_start\_w = round(bottom\_rois[1] \* spatial\_scale\_);

int roi\_start\_h = round(bottom\_rois[2] \* spatial\_scale\_);

int roi\_end\_w = round(bottom\_rois[3] \* spatial\_scale\_);

int roi\_end\_h = round(bottom\_rois[4] \* spatial\_scale\_);

For every proposal in the input proposals, we take the corresponding feature map section and divide that section into W\*H blocks defined by layer parameters. After that take the maximum element of each block and copy to the output. So the output size will be P\_w\*P\_h for every ROI proposal and N\*P\_w\*P\_h for all N proposals which is a fixed dimension feature map irrespective of the various sizes of the input proposals.

Scaled\_Proposals = Proposals \* spatial\_scale

for every ROI in Scaled\_Proposals:

fmap\_subset = feature\_map[ROI] (Feature\_map for that ROI)

Divide fmap\_subset into P\_wxP\_h blocks (ex: 6\*6 blocks)

Take the maximum element of each block and copy to output block

The main advantage of ROI pooling is that we can use the same feature map for all the proposals which enables us to pass the entire image to the CNN instead of passing all proposals individually.

6)what are fully convolutional networks and how do they works?(FCNs)

Ans : fully convolution network (FCN) is a neural network that only performs convolution and subsampling or upsampling operations.an FCN is a CNN without fully connected layers.

Fully Convolutional Networks,or FCNs are an architecture used mainly for semantic segmentation. They employ solely locally connected layers, such as convolution, pooling and upsampling. Avoiding the use of dense layers means less parameters.

7)what are the anchor boxes and how do you use them ?

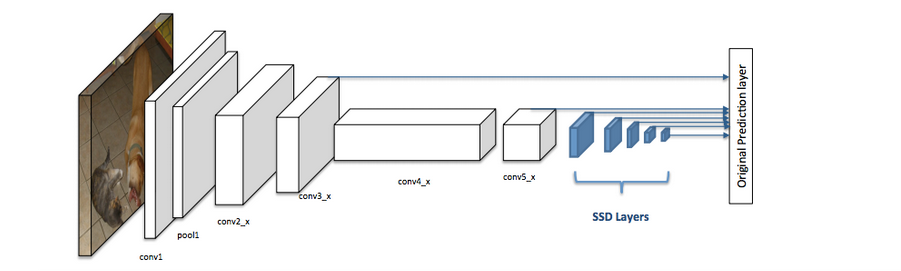
Ans : Anchor boxes are a set of predefined bounding boxes of a certain height and width. These boxes are defined to capture the scale and aspect ratio of specific object classes you want to detect and are typically chosen based on object sizes in your training datasets.

The network predicts the probability and other attributes, such as background, intersection over union (IoU) and offsets for every tiled anchor box. The predictions are used to refine each individual anchor box. You can define several anchor boxes, each for a different object size. Anchor boxes are fixed initial boundary box guesses.The network does not directly predict bounding boxes, but rather predicts the probabilities and refinements that correspond to the tiled anchor boxes. The network returns a unique set of predictions for every anchor box defined. The final feature map represents object detections for each class. The use of anchor boxes enables a network to detect multiple objects, objects of different scales, and overlapping objects.

8)Describe the single shot Detectors architecture.(SSD)

Ans : SSD has two components: a backbone model and SSD head. Backbone model usually is a pre-trained image classification network as a feature extractor. This is typically a network like ResNet trained on ImageNet from which the final fully connected classification layer has been removed.

thus left with a deep neural network that is able to extract semantic meaning from the input image while preserving the spatial structure of the image albeit at a lower resolution. For ResNet34, the backbone results in a 256 7x7 feature maps for an input image. The SSD head is just one or more convolutional layers added to this backbone and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layers activations.the first few layers (white boxes) are the backbone, the last few layers (blue boxes) represent the SSD head.



9)how does the SSD network predict?

Ans :SSD uses a matching phase while training, to match the appropriate anchor box with the bounding boxes of each ground truth object within an image. Essentially, the anchor box with the highest degree of overlap with an object is responsible for predicting that object’s class and its location. This property is used for training the network and for predicting the detected objects and their locations once the network has been trained. In practice, each anchor box is specified by an aspect ratio and a zoom level.

10)explain multi scale detectors.

Ans :The multi scale detectors consists of a proposal sub-network and a detection sub-network. In the proposal sub-network, detection is performed at multiple output layers, so that receptive fields match objects of different scales. These complementary scale-specific detectors are combined to produce a strong multi-scale object detector.Most filters are applied to an image at a fixed scale, while image features occur at all scales. ImagePyramid creates different resolutions of an image. Results of filtering all levels of image pyramid can be combined to create a multiscale feature detection.

11)what are dilated(or atrous) convolutions ?

Ans : The idea of Dilated Convolution is come from the wavelet decomposition. It is also called “atrous convolution”, “algorithme à trous” and “hole algorithm”.



The left one is the standard convolution. The right one is the dilated convolution. We can see that at the summation, it is s+lt=p that we will skip some points during convolution.

When l=1, it is standard convolution.

When l>1, it is dilated convolution.

Dilated convolution is basically a convolution with a wider kernel created by regularly inserting spaces between the kernel elements. we present a new version of the dilated convolution in which the spacings are made learnable via backpropagation through an interpolation technique.