**DOCUMENTATION ON IMPLEMENTATION OF FASTER RCNN:**

***Step by step implementation:***

1. Features Extraction from the image.
2. Creating anchor targets.
3. Locations and objectness score prediction from the RPN network.
4. Taking the top N locations and their objectness scores aka proposal layer
5. Passing these top N locations through Fast R-CNN network and generating locations and cls predictions for each location is suggested in 4.
6. generating proposal targets for each location suggested in 4
7. Using 2 and 3 to calculate rpn\_cls\_loss and rpn\_reg\_loss.
8. using 5 and 6 to calculate roi\_cls\_loss and roi\_reg\_loss.

**1. FEATURE EXTRACTION:**

The EfficientNet network is used as a feature extraction module here.

* Read a batch of training images along with their bounding boxes and lables.

# Resize the input images to (h=800, w=800)

# Use EfficientNet to extract features from input images

# Input images (batch\_size, H=800, W=800, d=3), Features: (batch\_size, H= 25, W=25, d=512)

* Pass the image through the layers

2.**ANCHOR BOXES GENERATION:**

1. Generate Anchor at a feature map location
2. Generate Anchor at all the feature map location.
3. Assign the labels and location of objects (with respect to the anchor) to each and every anchor.
4. Generate Anchor at a feature map location

* Here we use anchor\_scales of 4,8,16, ratio of 0.5, 1, 2 and sub sampling of 32 (Since we have pooled our image from 800 px to 25px).
* At each pixel location on the feature map, We need to generate 9 anchor boxes (number of anchor\_scales and number of ratios) and each anchor box will have ‘y1’, ‘x1’, ‘y2’, ‘x2’. So at each location anchor will have a shape of (9, 4).
* Find coordinates of the 625 center points to generate anchor boxes and display them
* for each of the 625 anchors, generate 9 anchor boxes and display on the image

**3. VALID ANCHOR BOX AND GROUND TRUTH BOXES**

* Calculate valid anchor boxes with (y1, x1)>0 and (y2, x2)<=800 i.e inside the boyundary of image.
* Calculate iou of the valid anchor boxes and check which anchor box has max iou with the ground truth bbox

**4. MINI BATCH TRAINING**

* 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.
* we need to randomly sample n\_pos samples from the positive labels and ignore (-1) the remaining ones.
* we will assign anchor locs to all the valid anchor boxes irrespective of its label, later when we are calculating the losses, we can remove them with simple filters.
* The final two matrices are
* anchor\_locations [N, 4] — [5625, 4]

anchor\_labels [N,] — [5625]These are used as targets to the RPN network.

**5. REGION PROPOSAL NETWORK**

* To generate region proposals, we slide a small network over the convolutional feature map output that we obtained in the feature extraction module.
* Each sliding window is mapped to a lower-dimensional feature [320 features]. This feature is fed into two sibling fully connected layers: box regression layer and box classification layer
* Now the outputs we got in the feature extraction state should be sent to this network to predict locations of objects with repect to the anchor and the objectness score assoiciated with it.
* Here, pred\_cls\_scores and pred\_anchor\_locs are the output the RPN network and the losses to updates the weights and pred\_cls\_scores and objectness\_scores are used as inputs to the proposal layer, which generate a set of proposal which are further used by RoI network

**6. GENERATING PROPOSALS TO FEED FASTER RCNN NETWORK**

* Parameters used:

n\_train\_pre\_nms — number of bboxes before nms during training

n\_train\_post\_nms — number of bboxes after nms during training

n\_test\_pre\_nms — number of bboxes before nms during testing

n\_test\_post\_nms — number of bboxes after nms during testing

min\_size — minimum height of the object required to create a proposal.

* convert the loc predictions from the rpn network to bbox [y1, x1, y2, x2] format.
* clip the predicted boxes to the image
* Remove predicted boxes with either height or width < threshold (min\_size).
* Sort all (proposal, score) pairs by score from highest to lowest.
* Take top pre\_nms\_topN (e.g. 12000 while training and 300 while testing).
* Apply nms threshold > 0.7
* Take top pos\_nms\_topN (e.g. 2000 while training and 300 while testing)

# Proposal targets:

The Fast R-CNN network takes the region proposals (obtained from proposal layer in previous section), ground truth boxes and their respective labels as inputs. It will take the following parameters

* n\_sample: Number of samples to sample from roi, The default value is 128.
* pos\_ratio: the number of positive examples out of the n\_samples. The default values is 0.25.
* pos\_iou\_thesh: The minimum overlap of region proposal with any groundtruth object to consider it as positive label.
* [neg\_iou\_threshold\_lo, neg\_iou\_threshold\_hi] : [0.0, 0.5], The overlap value bounding required to consider a region proposal as negitive [background object].

# NMS 2000 ROIs AND 128 ROI SAMPLES:

* Find the iou of each ground truth object with the region proposals

# *Find out which ground truth has high IoU for each region proposal, Also find the maximum IoU*

# Display ROI samples with positive

# Display ROI samples with negative

# 128 ROI SAMPLES FEATURES, MAX POOLING SAME SIZE, H=7, W=7 (ROI POOLING):

# Visualize the first 5 ROI's feature map (for each feature map, only show the 1st channel of d=320)

* calculate rpn\_cls\_loss and rpn\_reg\_loss.
* calculate roi\_cls\_loss and roi\_reg\_loss.