RCNN

CNN to classify the presence of the object within the Region. The problem with this approach is that the objects of interest might have different spatial locations within the image & diff aspect ratios Hence, you would have to select a huge No. of regions & this could computationally blow up. ∴ RCNN has been developed

In R-CNN proposed a method where we use selective search to extract just 2000 regions from the image & It called as Region proposals. Ie. instead of trying to classify a huge No. of regions, you can just work with 2000 regions., these 2000 region proposals are generated using selective Search Algorithm

Tensorflow Object Detection Classifier Training Steps:

1) Install Tensorflow - GPU
2) Set up object Detection directory structure & Anaconda Virtual Environment
3) Gather & label picture
4) Generate training data
5) Create label Map & configure training
6) Train Object detector
7) Export Inference Graph
8) Test it Out.!

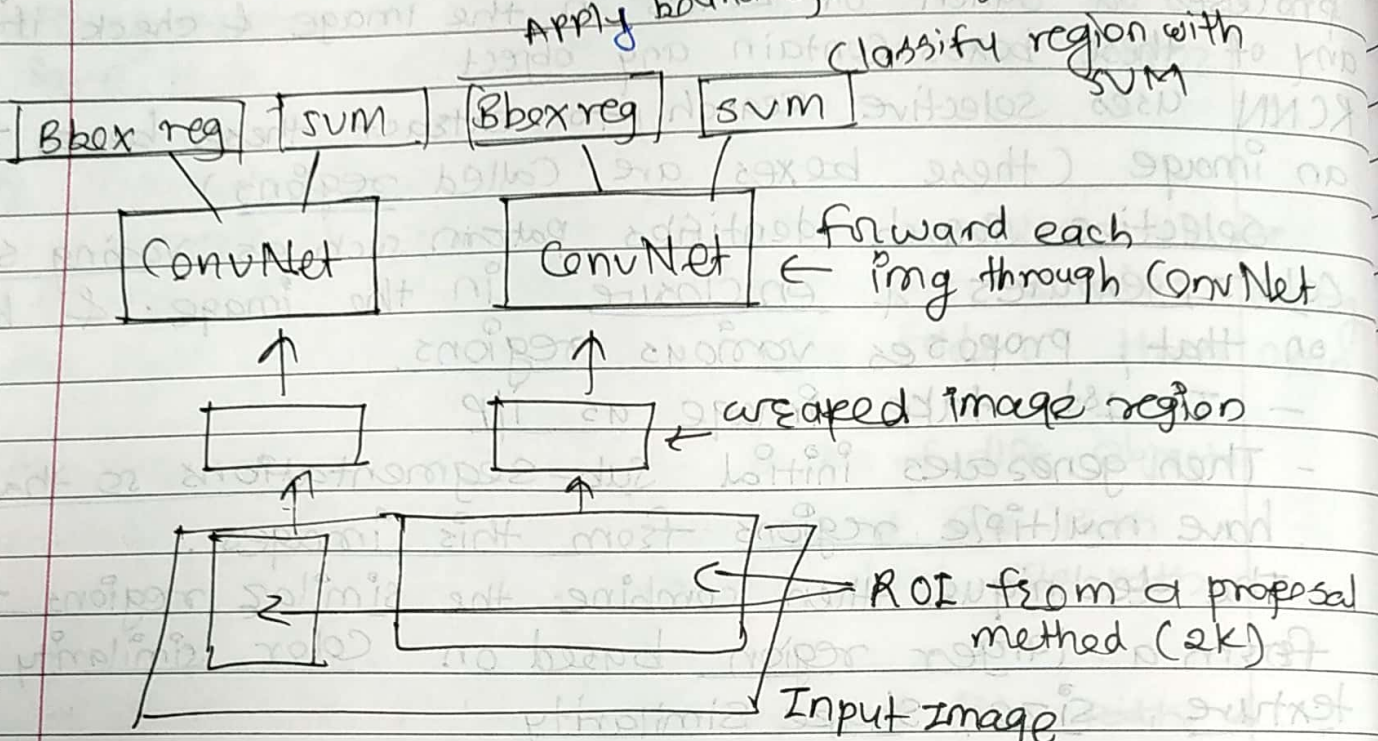
RCNN

Instead of working on a massive No. of regions, the RCNN algo proceses a bunch of boxes in the image & check if ~~the~~ any of these boxes contain any object.
RCNN uses selective search to extract these boxes from an image (these boxes are called regions)
Selective search identifies pattern such as varing scales, colors, textures & enclosure in the image. & based on that, proposes various regions.
- It 1st take image as i/p
- Then generates initial sub-segmentations so that we have multiple regions from this images.
- The technique then combine the similar regions to form a larger region based on color similarity, texture, size, shape similarity
finally these regions then produce the final object locations (ROI).

Steps

1) pre-trained CNN Network, we train last layer of the Network based on the No. of classes that need to be detected
2) get ROI of each image using selective search (2K)
3) Reshape these region so that they can match CNN input ~~sh~~ size.
4) after, train SVM to classify objects & background for each class → train on binary SVM for each classes
5) finally, we train a linear regression model to generate tighter bounding boxes for each identified ~~put~~ image in the image

CNN are used to extract each regions & SVM used for class classification.



problem with RCNN –

RCNN model is extremly slow because

- Extracting 2000 regions for each image based on selective search
- Extracting features using CNN for every img region suppose we have N images, No. of features → N*2000

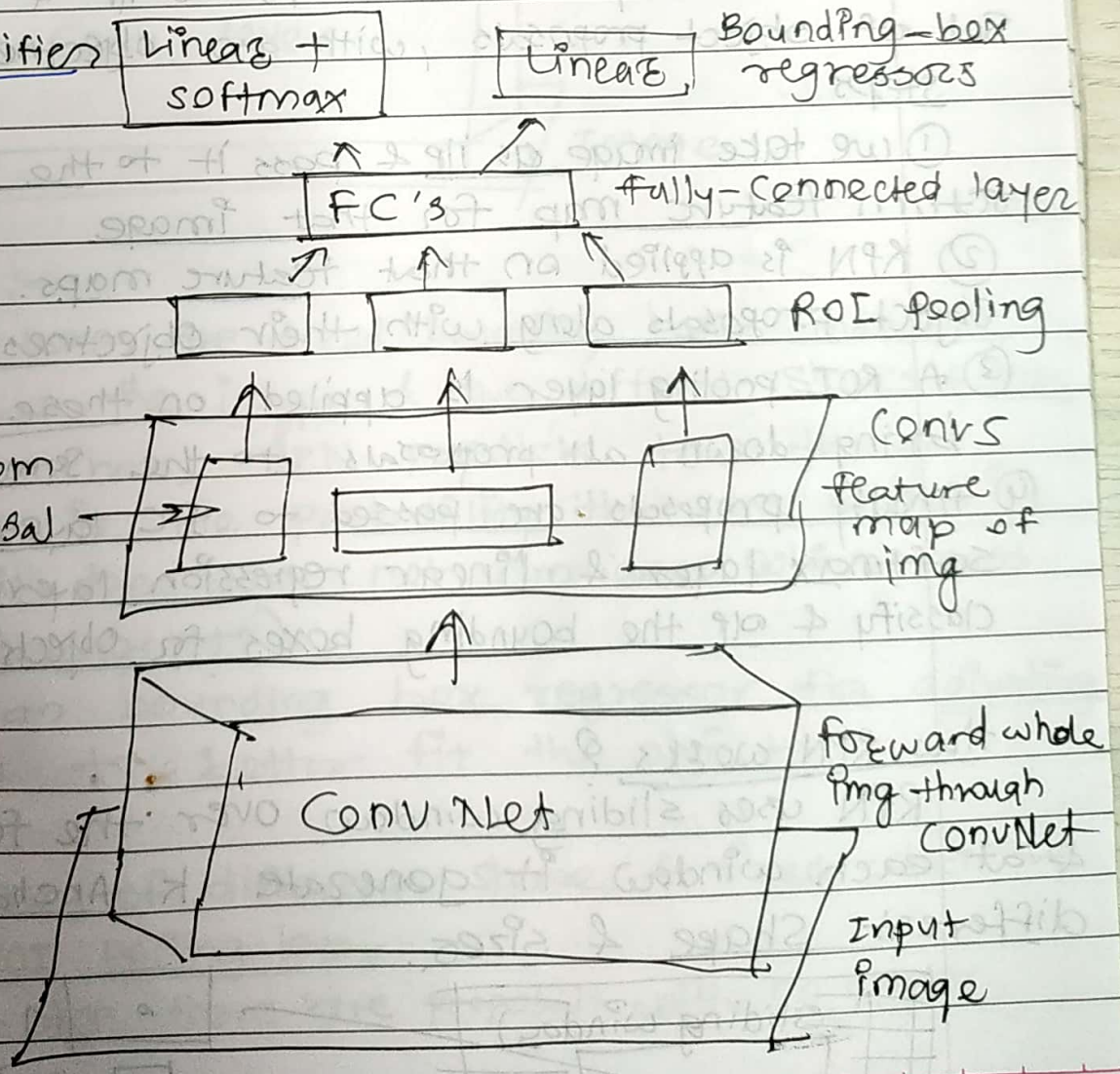
So RCNN makes very slow.

fast RCNN –

In fast RCNN, we feed the input image to the CNN, which turn generates the ~~CNN~~ convolutional feature Maps. Using these maps the regions of proposals are extracted, we then use a ROI pooling layer to reshape all proposed regions into a fixed size.

① Take image as input passed to convNet which generate ROI
② A ROI pooling layer is applied on all of these regions to reshape them as per input of the ConvNet.
③ Then each region is passed to fully connected Nlw.
④ softmax layer is used on top of fully connected Nlw to output classes. along with softmax, linear regression layer is also used parallely to output bounding box cordinates for predicted classes.

Instead of using three layes fast RCNN uses a single layer which extract features from the regions, divide them into diff classes, return bounding boxes.



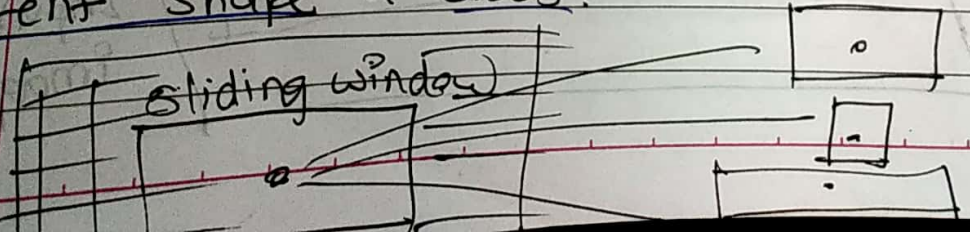
Fast RCNN resolves two major issues of RCNN,
- passing one instead of 2000 regions / image to convNet.
- instead of 3 diff layers use one layer for all.

Problems of fast RCNN
- It also uses selective search as a proposal method to find the ROI, which is slow & time consuming process on real time dataset.

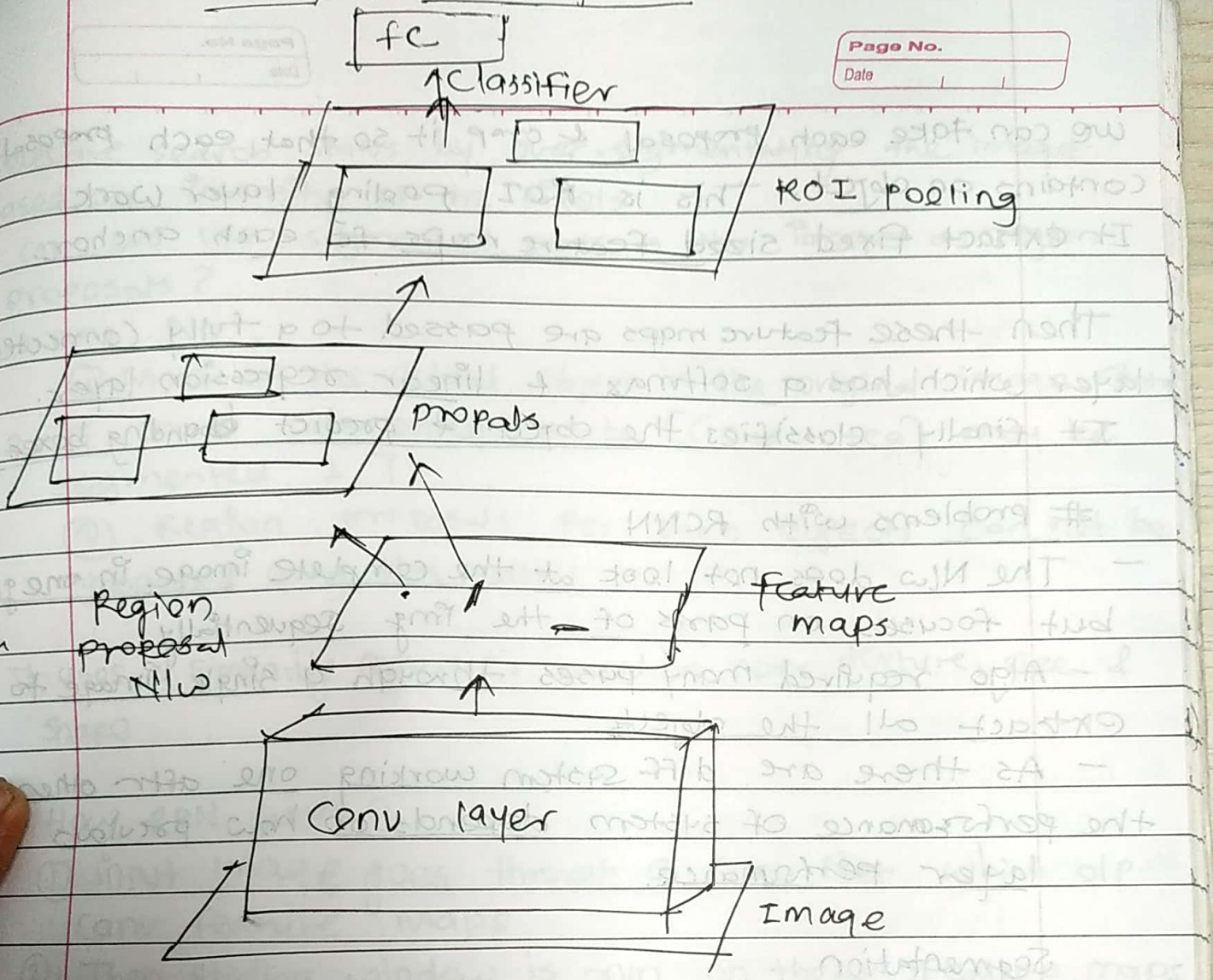
faster RCNN -
faster RCNN is modified version of fast RCNN. diff betn them fast RCNN uses selective search for generating ROI, while faster RCNN uses "Region Proposal Network" → RPN. RPN takes image feature maps as i/p & generates a set of object proposals, with an objectness score as o/p

steps
① we take image as i/p & pass it to the convNet which return feature map for that image
② RPN is applied on that feature maps. this return object proposals along with their objectness score
③ A ROI pooling layer is applied on these proposals to bring down all proposals to the same size.
④ Finally proposals are passed to FC layer which has softmax layer & linear regression layer at its top, to classify & o/p the bounding boxes for objects.

How RPN works ?
RPN uses sliding window over the feature maps, at each window it generate k Anchor boxes of different Shape & sizes.



Softmax + linear Regression



Anchor boxes :-

Anchor boxes are fixed bounding boxes that are placed throughout the image & have different shapes & sizes for each anchor, RPN predicts two things:

- The first is the probability that an anchor is an object (it does does not consider which class the object belongs to).
- Second is an bounding box regressor for adjusting the anchors to better fit the object

Then bounding boxes of different shape & sizes which are passed to the ROI pooling layer

After RPN step, there are proposals with no classes assigned to them..

we can take each proposal & crop it so that each proposal contains an object. This is RoI pooling layer work. It extract fixed sized feature maps for each anchor.

Then these feature maps are passed to a fully connected layer which has a softmax & linear regression layer. It finally classifies the object & predict bounding boxes.

problems with RCNN

- The N/w does not look at the complete image in one go, but focuses on parts of the img sequentially
  - Algo required many pases through a single image to extract all the objects
- As there are diff system working one aftr other, the performance of system depends on how previous layer performance.

Segmentation

we group adjacent regions which are similar to each other based on some criteria such as color, texture etc.

region praposal N/w

It works by grouping pixels into a smaller No of segments, An important method of RPN is to having high recall. value

Selective Search

It is a region proposal algorithm used in object detection. It is desined to be fast with high recall. It is based on computing hierarchical grouping of similar regions based on color, texture, size & shape compatibility

selective search starts by over-segmentating the image based on intensity of the pixels

can we use segmented parts in this image as region proposals?

→ ans No.,

① Most of the actual object in the original image contains 2 or more segmented parts ((cup - tea) may be segmented]

② Region proposals for such objects can not be generated using this method

It uses 4 similarity measures based on color, texture, size & shape

How RPN work?

① input image goes through a Conv Net which o/p conv feature maps

② Then sliding window is run on these feature maps, size of sliding window is nxn (3x3 Ex.) for each sliding window, a set of 9 anchors are generated which all have the same center (x_a, y_a) with diff aspect ratio, scale

for each anchor box how much this overlap with ground truth bounding boxes

③ finally feature map are extracted & thous feature maps are fed to smaller N/w which has two task, classification & regression.

o/p of regression → points of bounding box

o/p of classifier → probability indicates the whether predicted box contains an object(1) or background(0)

0 - for No Object.