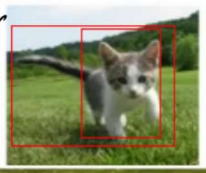
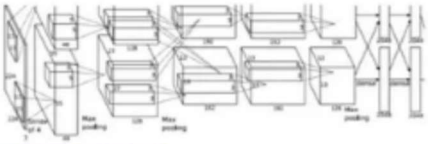
R-CNN, Fast R-CNN and Faster R-CNN

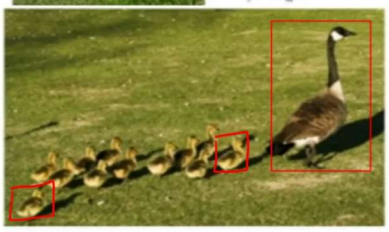
CNN → Classification

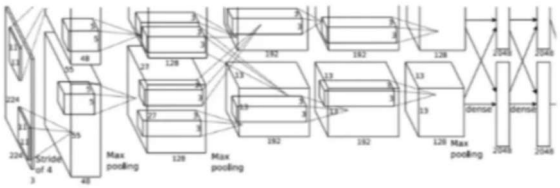




CAT: (x, y, w, h)

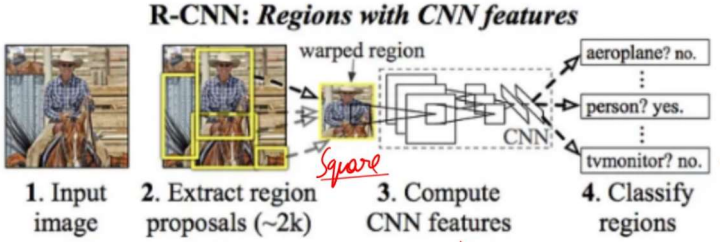
Major Reason for no CNN ↓
length of O/P layer is not constant





DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Naive Approach → Take different regions of interest from image, and use a CNN to classify the presence of the object within that region.

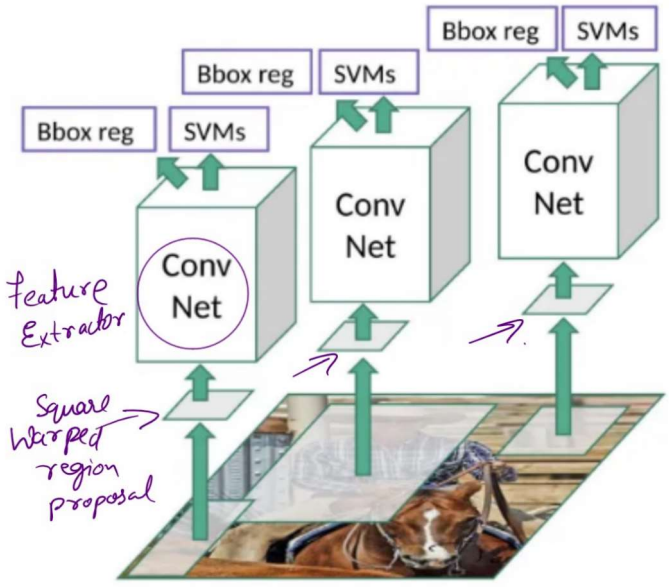


Oct 2014 → Ross Girshick
Jeff Donahue
Trevor Darrell
Jitendra Malik

4096 dim feat. vect. as O/P

→ Use selective search to extract just 2000 regions from the image.

2000 regions → Region Proposal → generated using → selective search



Selective Search
1.) Generate many candidate regions.
2.) Use greedy algo. to recursively combine similar regions into larger ones.
3.) Use the generated regions to produce the final candidate region proposals.

R-CNN.
↳ Image as input
↳ Generate 2000 candidate region proposals (R.P.)
↳ warp the the regioned proposal image → Square
↳ The square is given to a CNN
↓
produces output
↓
4096 dim. feature Vector
↓
SVM
↓
To classify whether the object is there or not.
+
four values which are offset values to increase the precision of bounding box.

Problems with R-CNN

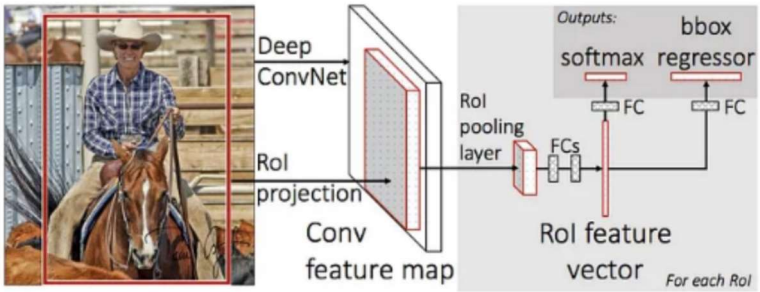
1.) Huge amount of time to train the network
2.) Can't be implemented in Real time (~50 to 60 sec)
↓
Detection on Test Img.
3.) Selective Search is a fixed algo. No learning is happening at this stage.
↳ This could lead to generating bad candidate R.P.

Fast R-CNN

↳Given by the authors of R-CNN

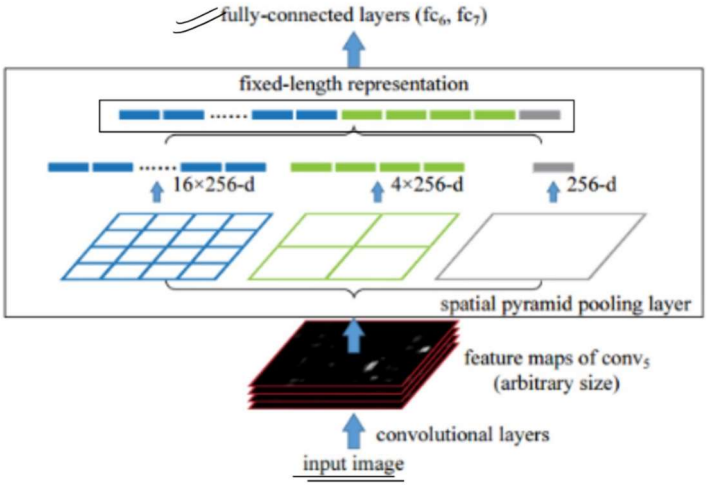
→ Instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map.

→ From the convolutional feature map, we identify the region of proposals and warp them into squares. and by using ROI pooling layer we reshape them into a fixed size, so it can be given into a fully connected layer.



→ From the ROI feature vector, we use a softmax layer to predict the class of proposed region. and also the offset values of the bounding box.

(Test time → 2 sec/ Image)

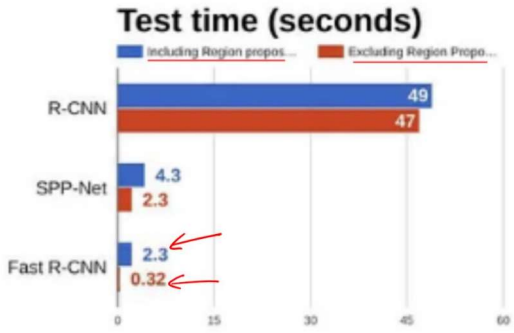


→ ROI pooling layer ≈ max pooling layer.

☆ Pool size is dependent on input size.

☆ The reason fast R-CNN is faster than R-CNN is because, you don't have to feed 2000 region proposals to the CNN every time.

Instead, the convolution operation is done once / image and a feature map is generated from it.



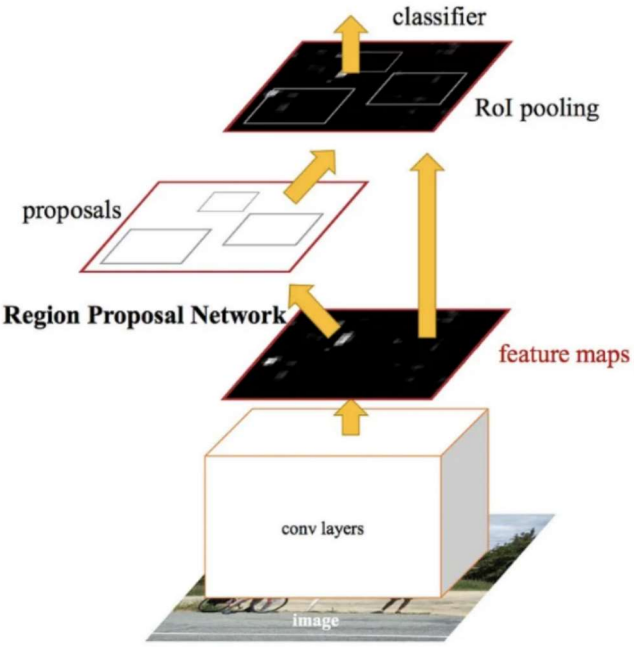
Problem: Region proposals become bottlenecks in fast R-CNN

⇓

affects the performace.

Faster R-CNN

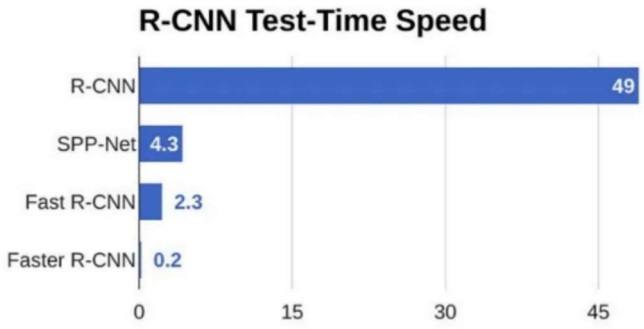
→ Jan 2016
- → Shaoqing Ren
- → Kaiming He
- → Ross Girshick
- → Jian Sun.



→ Eliminated Selective Search Algo.

→ Network learns the region proposals.

Input Image
↓
CNN
↓
Convolutional feature Map
↓
Separate network is used to predict the region proposals
↓
The predicted region proposals are then reshaped using RoI pooling layer. (x)
↓
Then (x) is used to classify the image within the proposed region & predict the offset values for the bounding boxes.



Faster-R-CNN can also be used for real-time object Detection