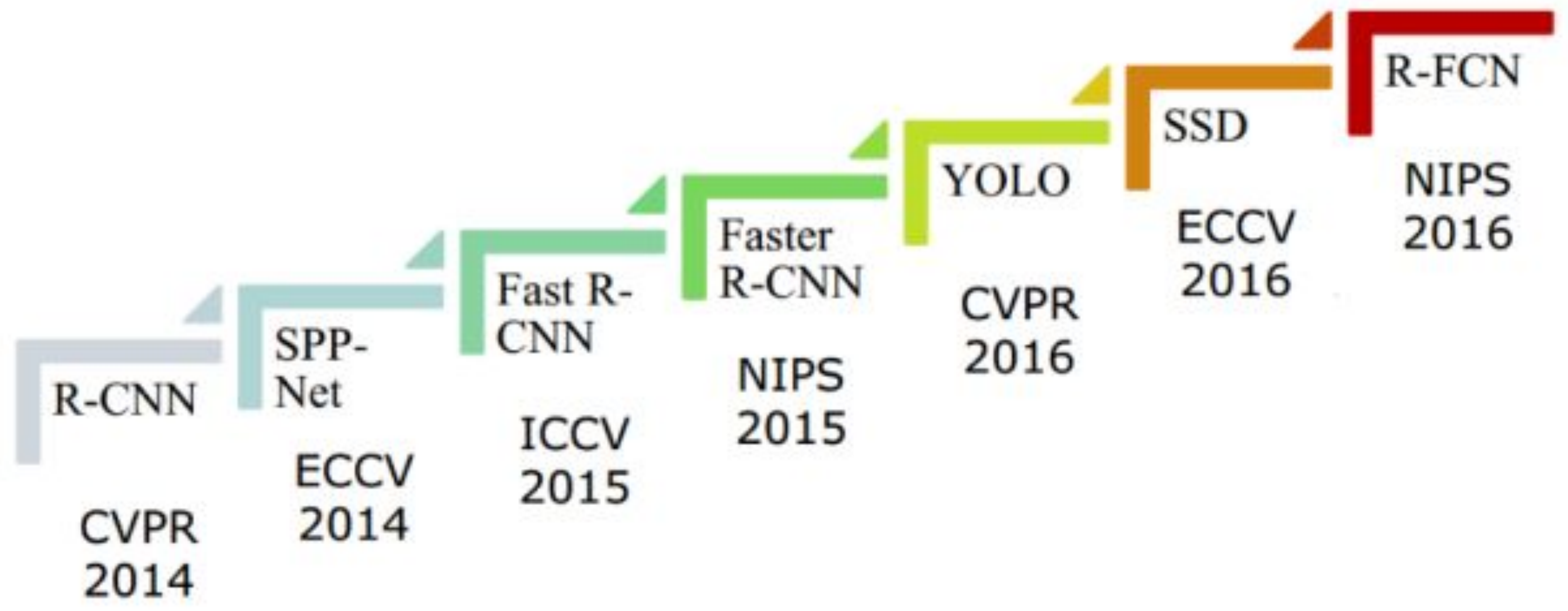
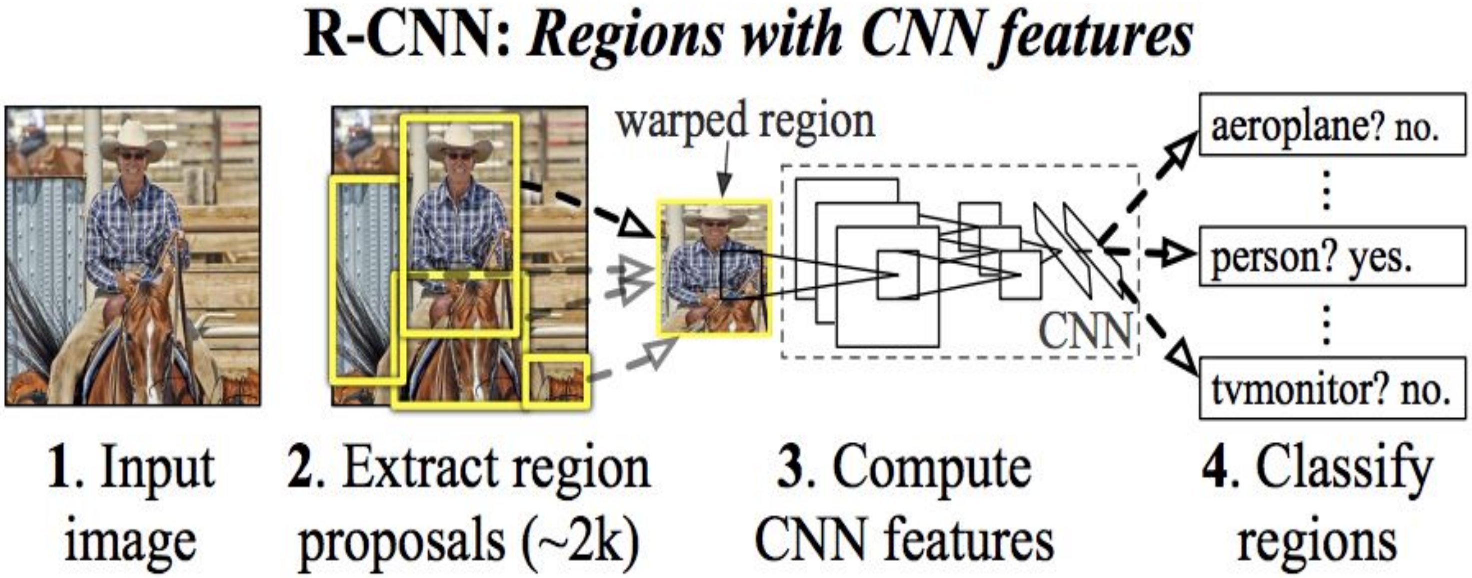
Faster-RCNN & SSD



RCNN

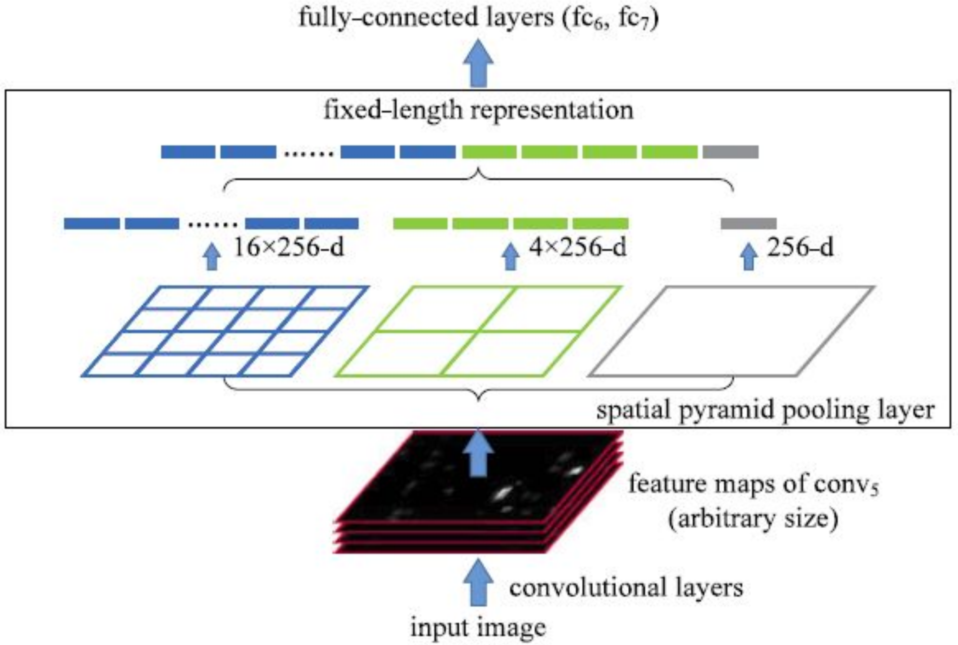


- CNN applied over each ROI detected from Selective search

SPP-Net

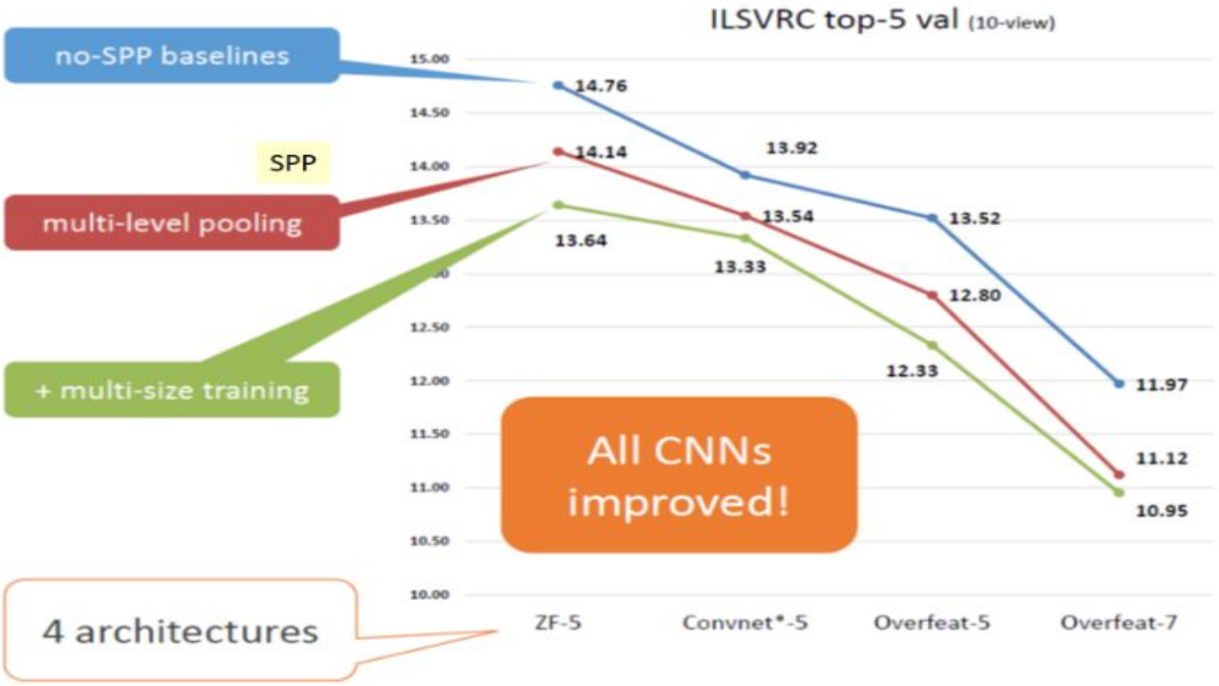
- Enables shared computation and makes RCNN faster
- Same features may belong to multiple selective search ROI
- Features with different size will generate different size of vector , SPP helps in making the size fixed to feed that into FC layer
- SPP divides each ROI in fixed no of bins and Max pool is performed over that
- As no of bins are fixed , so FC layer will get fixed size input vector
- Multiple pooling at different scales earlier only one max pooling was used
- **Drawback was network was only training the FC part not the SPP part**

SPP Layer



Three-Level Spatial Pyramid Pooling (SPP) in SPPNet with Pyramid {4×4, 2×2, 1×1}.

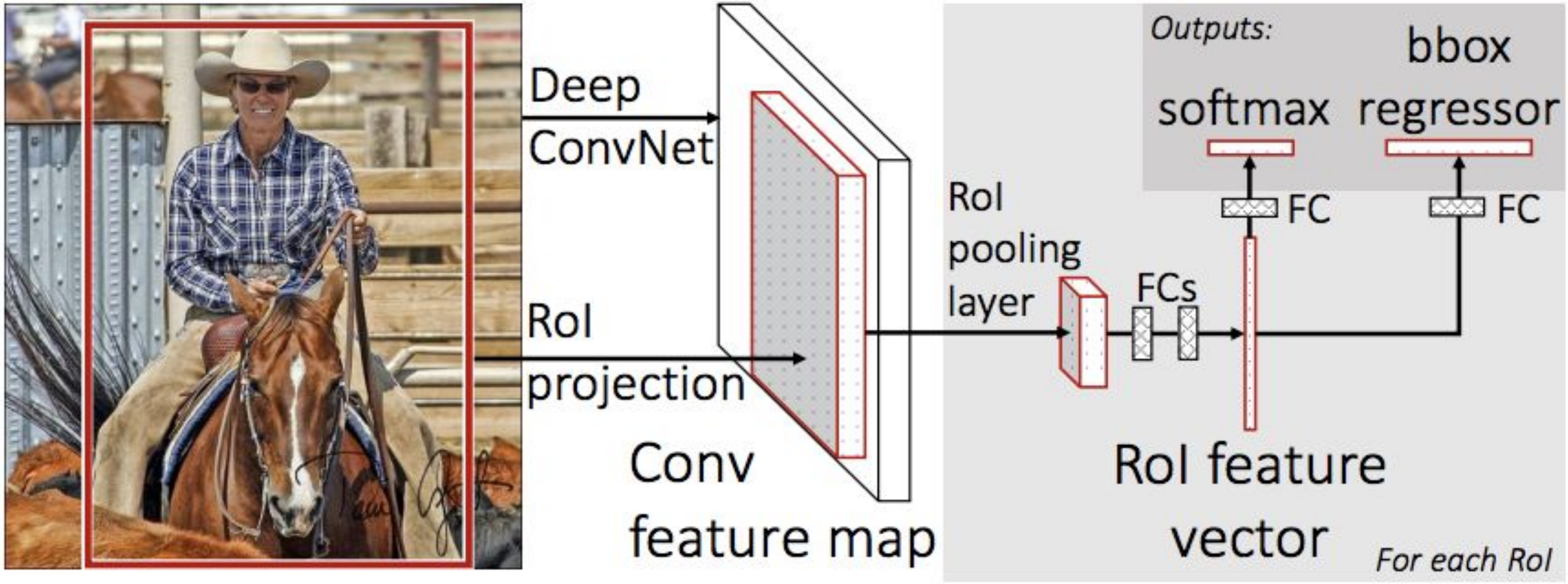
SPP Layer pros



Top-5 Error Rates for SPP and Multi-Size Training

4-level SPPNet is used here with the pyramid {6×6, 3×3, 2×2, 1×1}.

Fast-RCNN



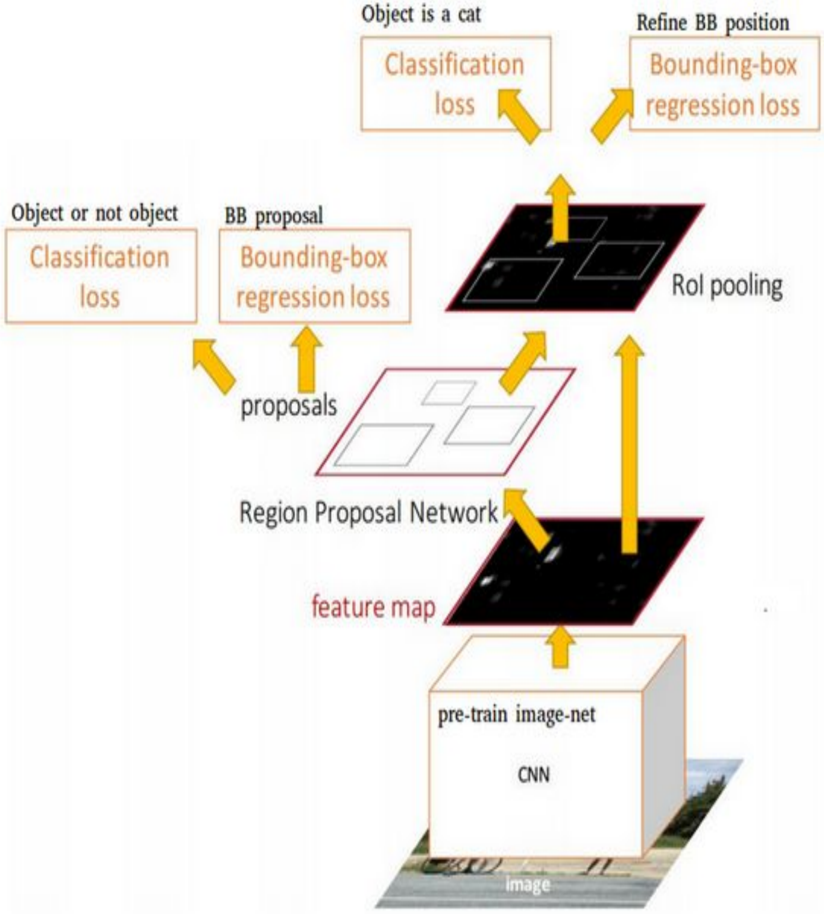
- CNN on full image =>selective Search on full image =>ROI from CNN feature Map =>ROI pooling layer to reduce size =>input to FC layer=>softmax /regressor

Fast RCNN Features

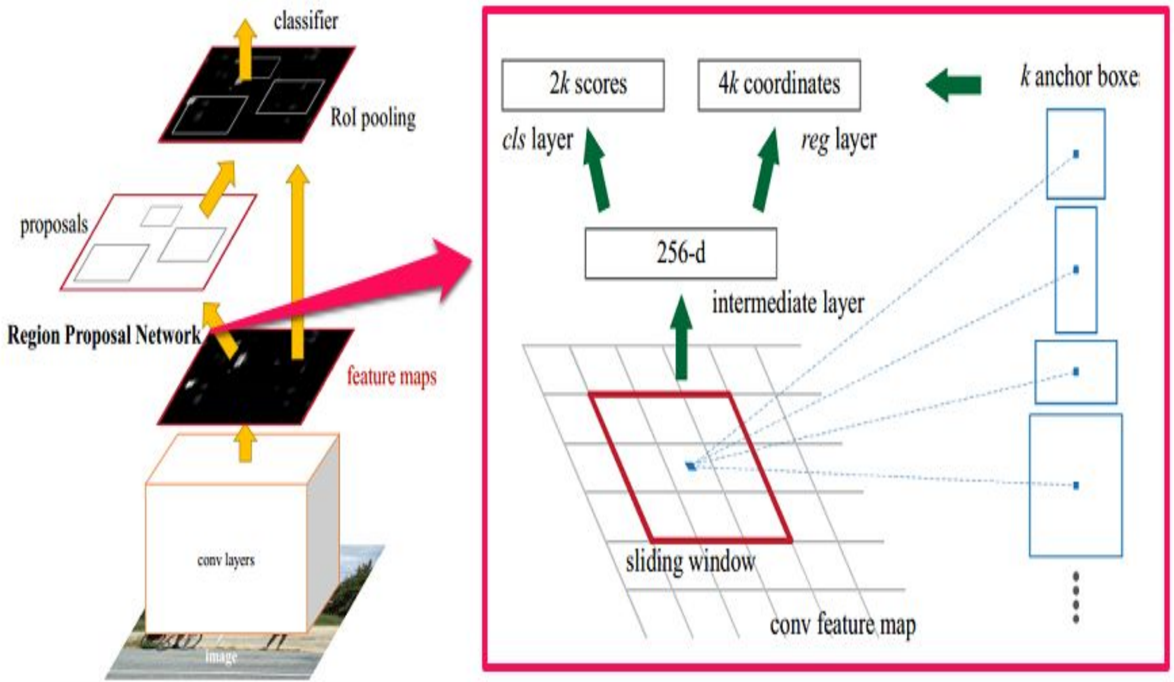
- Uses concept from SPP and added SPP layer
- Made network end to end trainable
- Added Regression and Classification training both in the network
- These changes helped in fast training and achieving better accuracy

Faster-RCNN

- Combination of RPN and Fast-RCNN
- Input entire image 3*3 slides to generate feature map and input to FC
- Multiple regions are predicted by FC layer at max k(anchor boxes)
- Output of regression layer (4k) and classification layer (2k)
- These detected Anchor boxes and feature map is input to Fast-RCNN model
- RPN uses pretrained model of imagenet classification
- Generated anchor boxes are used to train Fast-RCNN

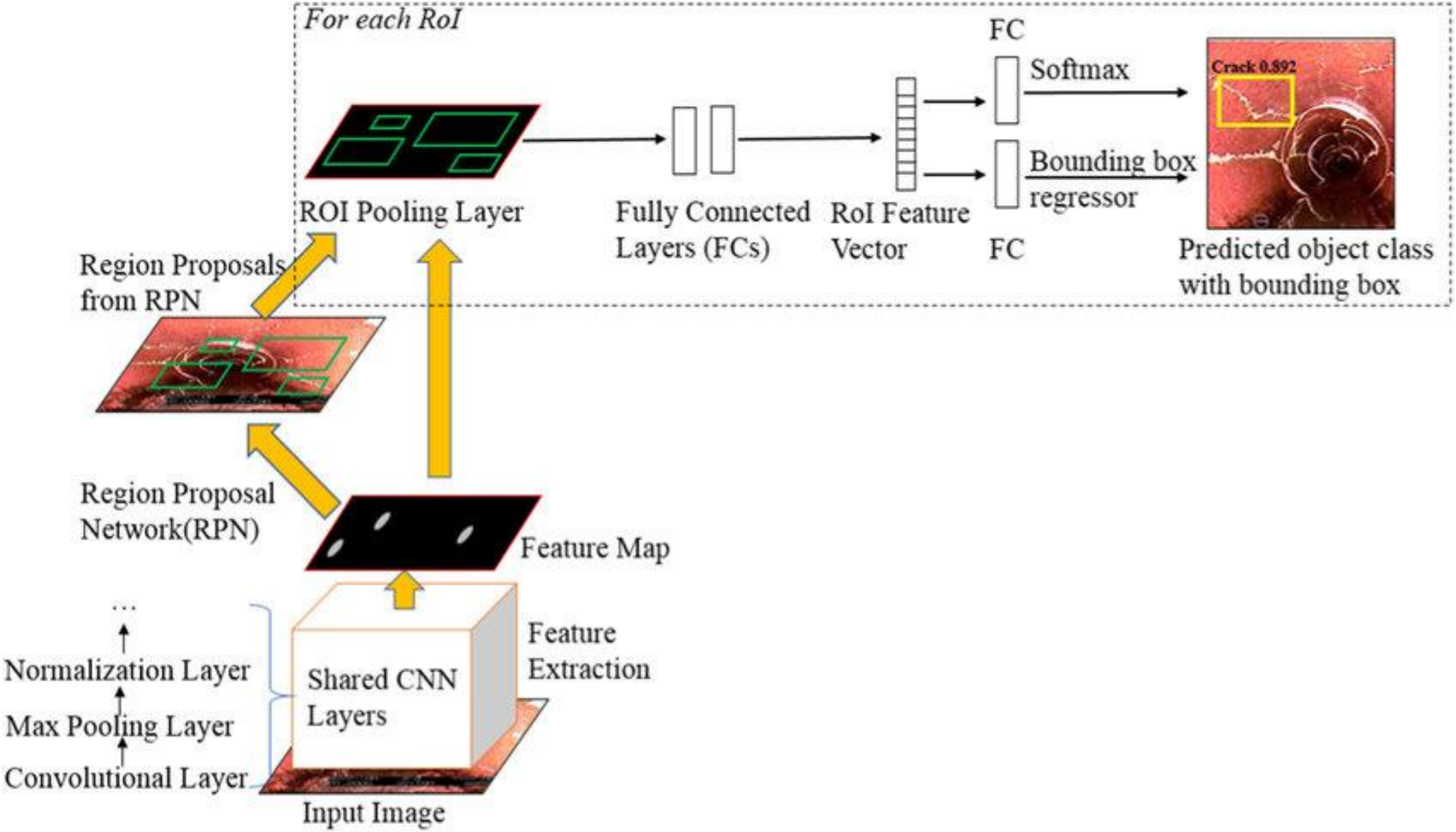


Faster-RCNN

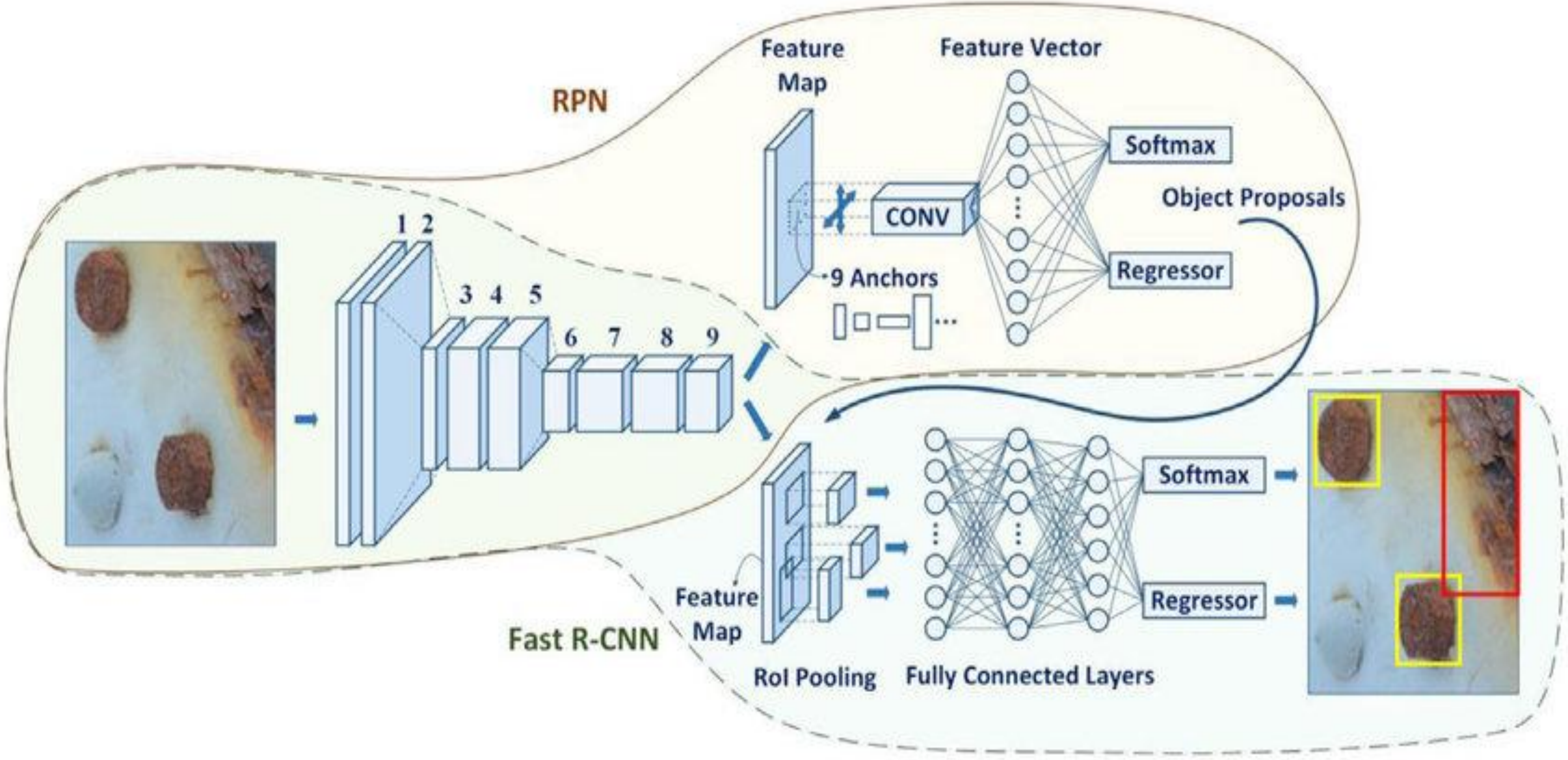


- Faster-RCNN uses 9 (k , anchok boxes 3: different scale , 3 : different ratio)
- Major feature was selective search was replaced with RPN

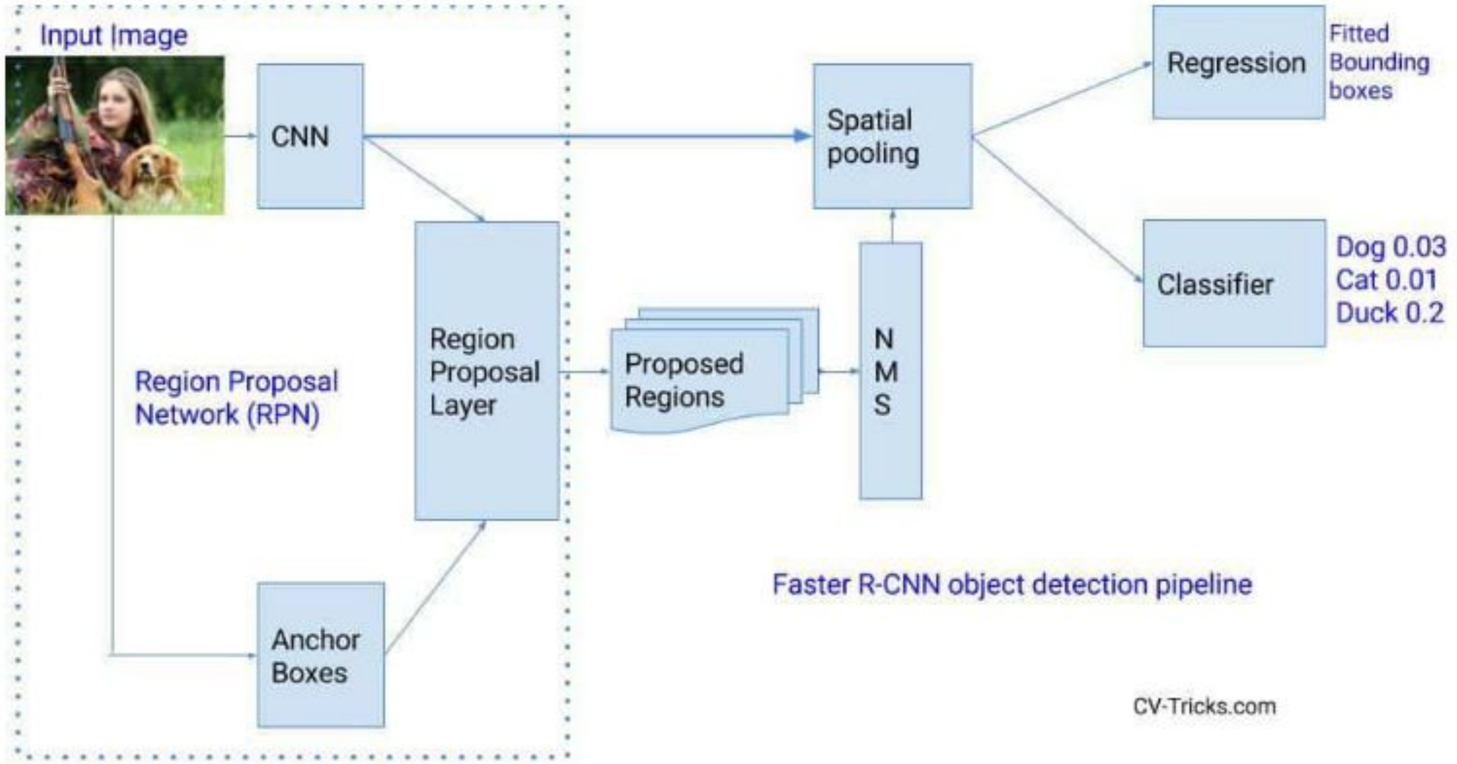
Faster-RCNN


For each RoI
FC
Softmax
Crack 0.892
Bounding box
regressor
ROI Pooling Layer
Fully Connected
Layers (FCs)
RoI Feature
Vector
FC
Predicted object class
with bounding box
Region Proposals
from RPN
Region Proposal
Network(RPN)
Feature Map
...
Normalization Layer
Max Pooling Layer
Convolutional Layer
Shared CNN
Layers
Feature
Extraction
Input Image

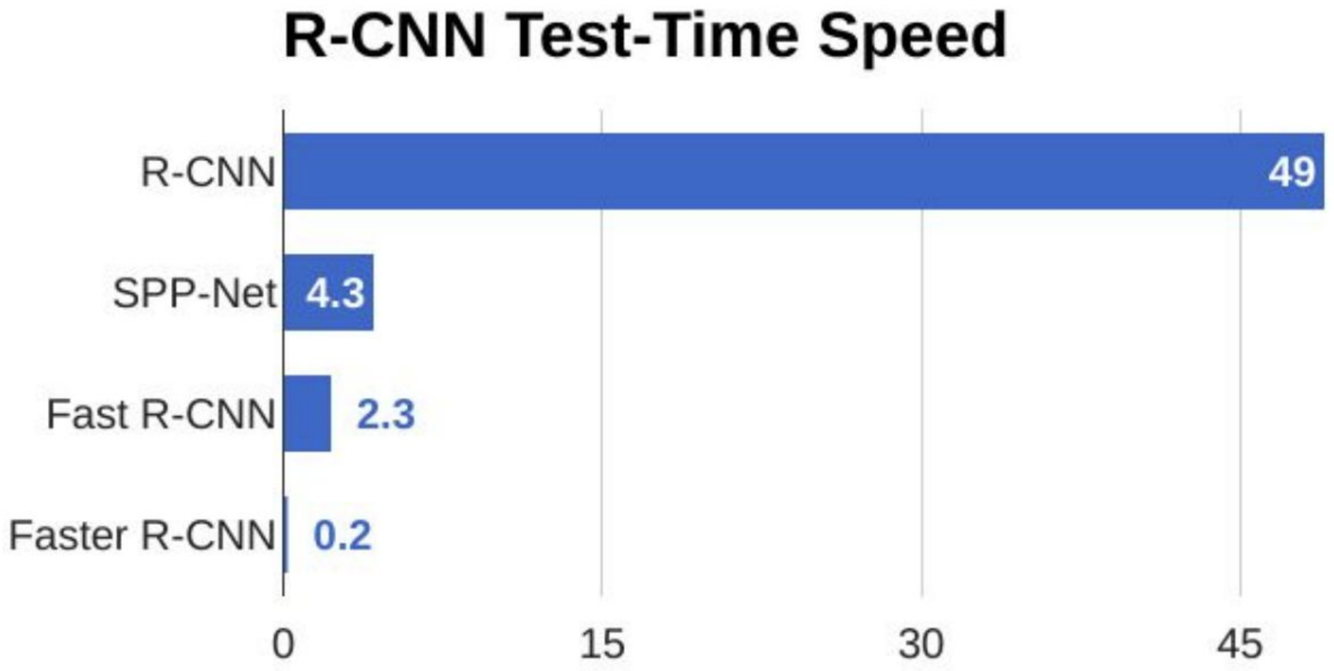
Faster-RCNN


RPN
Feature Map
Feature Vector
Softmax
CONV
Object Proposals
9 Anchors
Regressor
1 2
3 4 5
6 7 8 9
Feature Map
Softmax
Regressor
Fast R-CNN
RoI Pooling
Fully Connected Layers

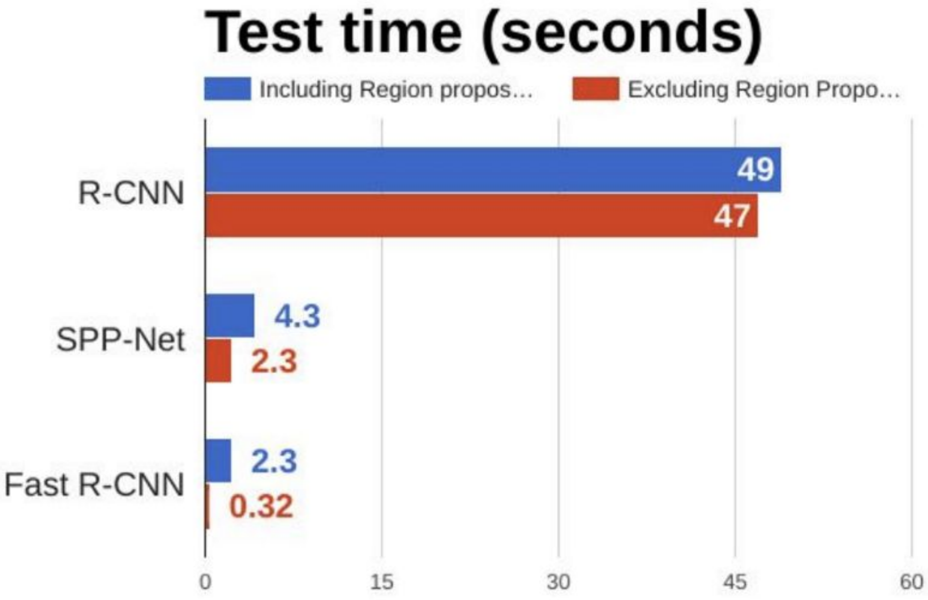
Faster RCNN Overview



Performance



Performance



R-FCN

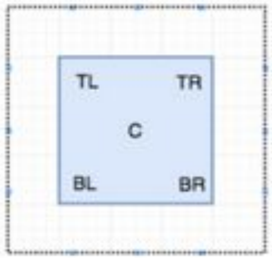
- Region based fully convolutional network
- In Faster-RCNN , FC layer does not share common features among ROI due to which it takes time
- In RFCN FC layer is removed ,positive Score maps are used before ROI pooling
- Average voting after ROI pooling , and No learnable parameter which makes it faster
-

R-FCN : Position Sensitive Score map and ROI pooling

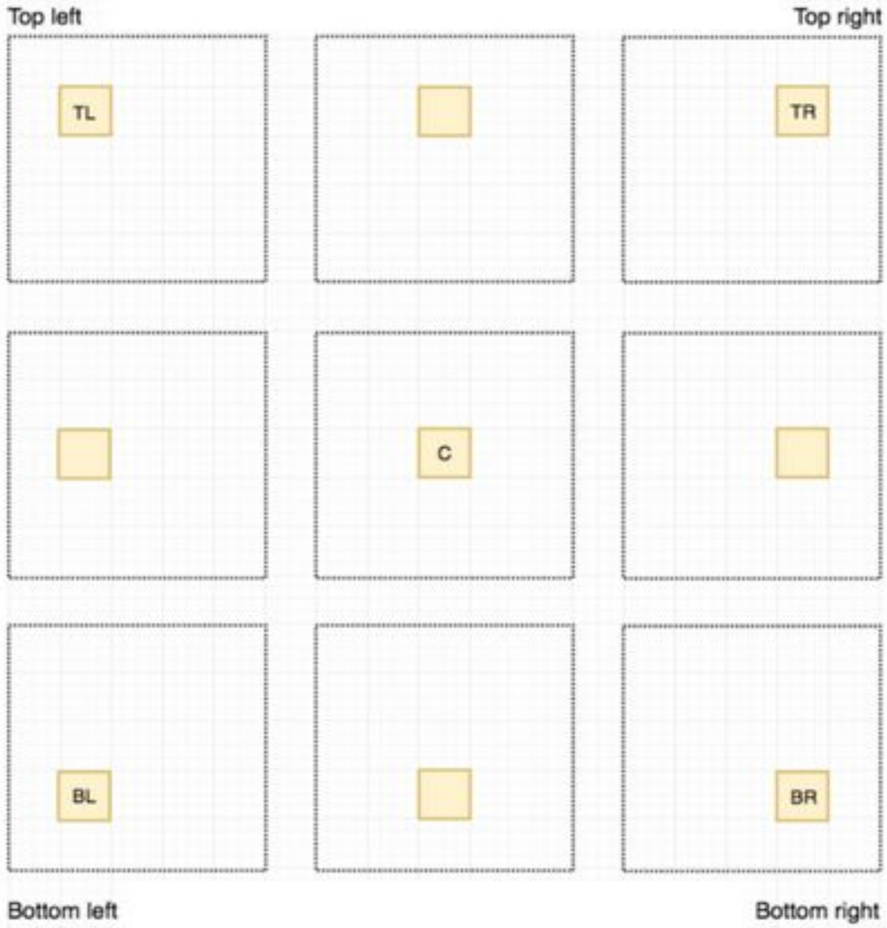
- C classes , 1 background , (C+1) total classes
- Before **Positive sensitive score map** , lots of convolutional operations are applied on the feature map
- For each class k*k convolution operations are applied
- k*k feature map denotes (TL , TC ... BL , BR) points of object
- So total , k*k*(C+1) operations are applied to generate **positive sensitive score map ,** this is for classification
- To perform bounding box regression , another filter 4*k*k is used and position based pooling is applied to generate bounding box

R-FCN : PSSM

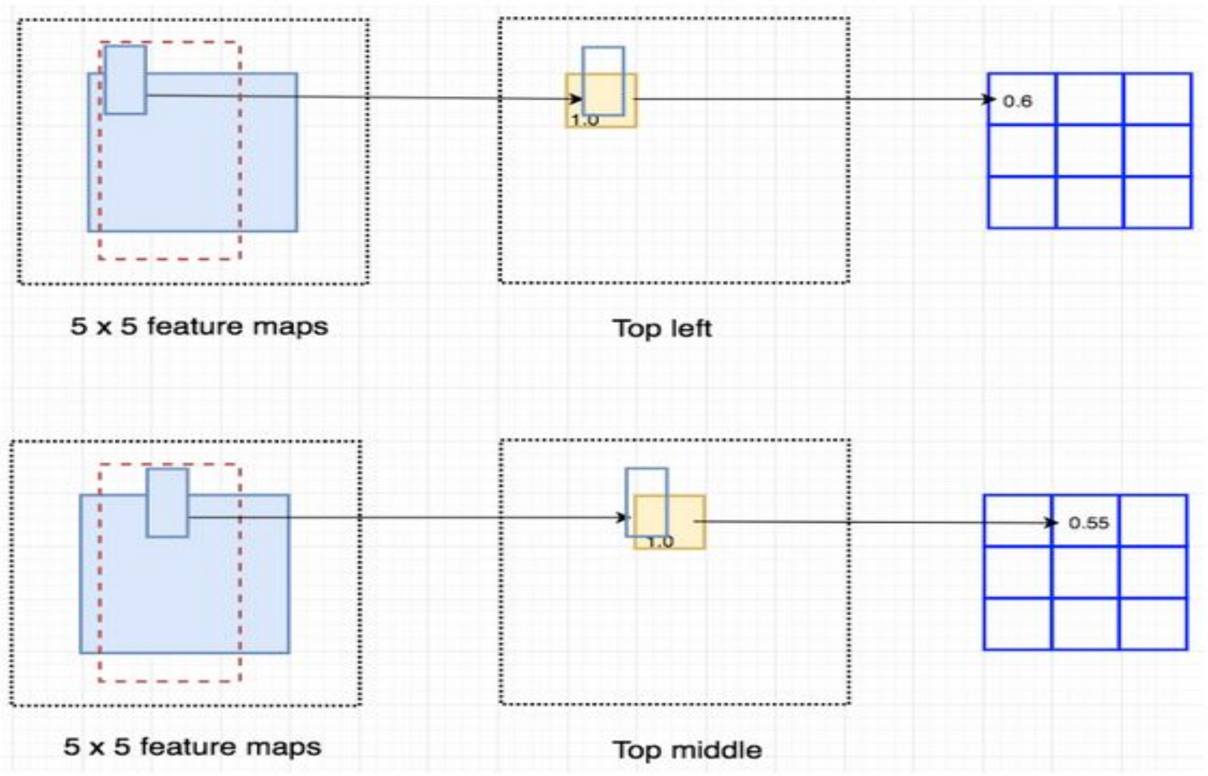
- 5*5 feature map divided into 3*3 regions
- These 9 regions are called PSSM , because these ,maps scores a subregion of object



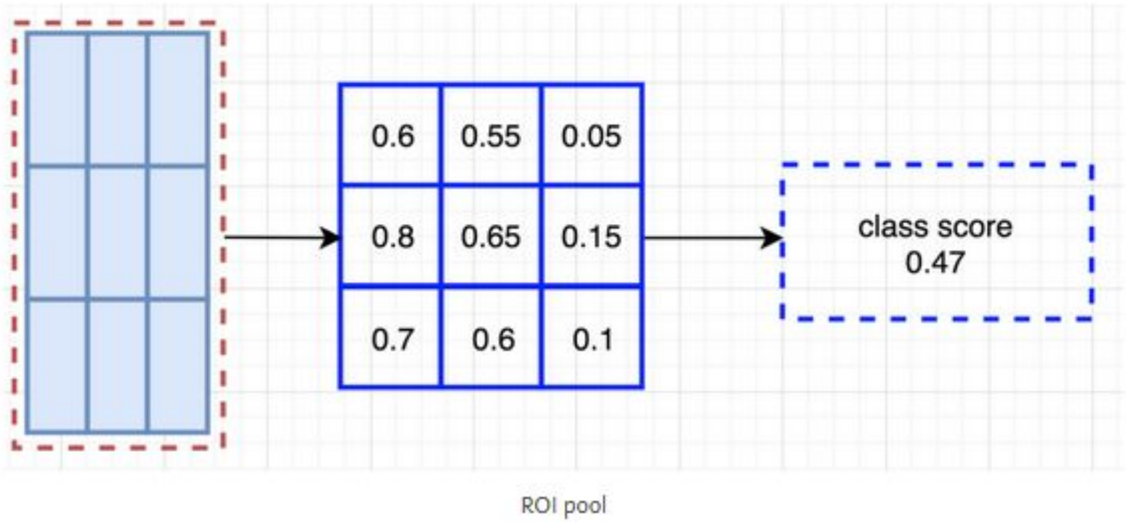
5 x 5 feature maps



RFCN : PSSM , for each feature map

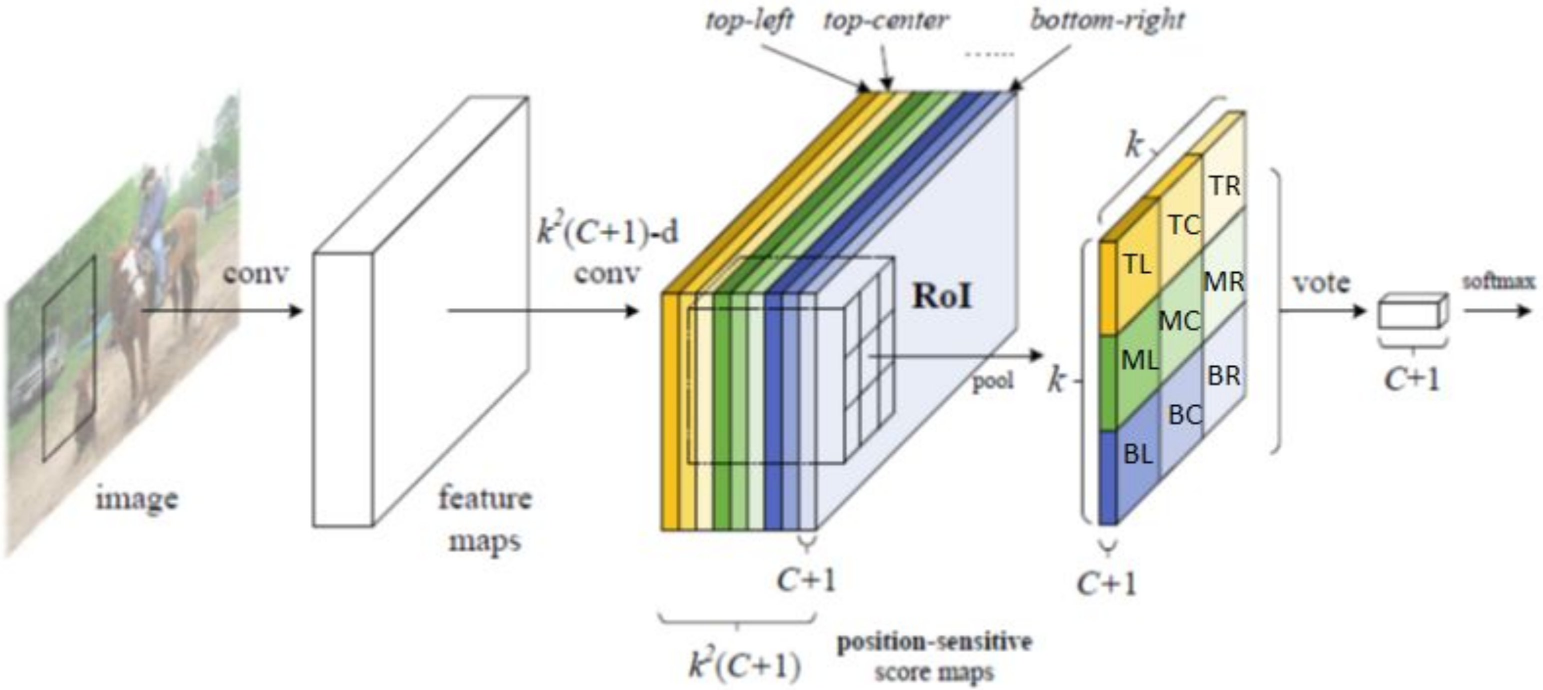


RFCN : PSSM and Average ROI pooling



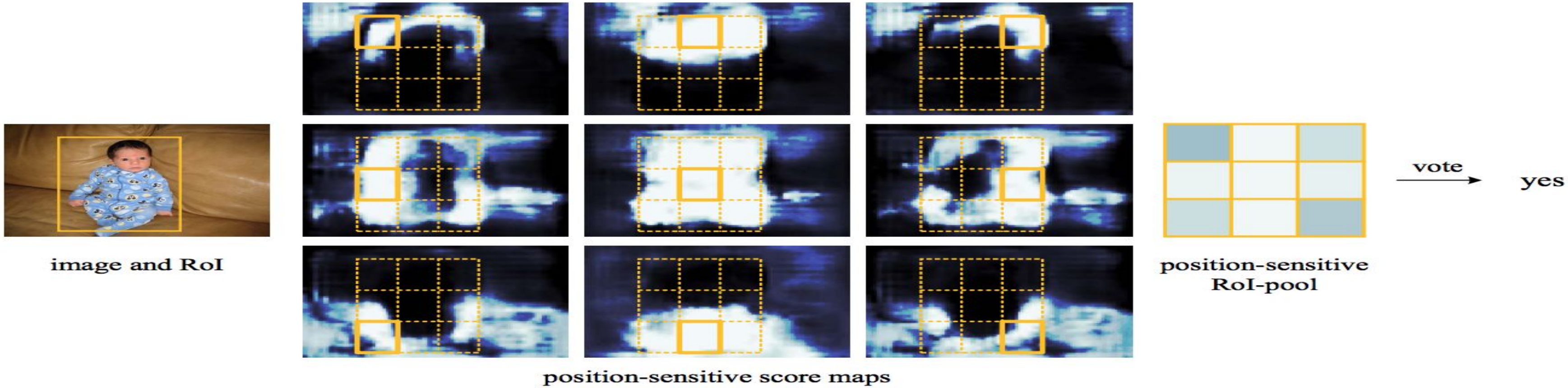
- Each class will have its own 3*3 maps , so total 3*3*(C+1) feature maps

R-FCN : Positive Sensitive Score map and ROI pooling

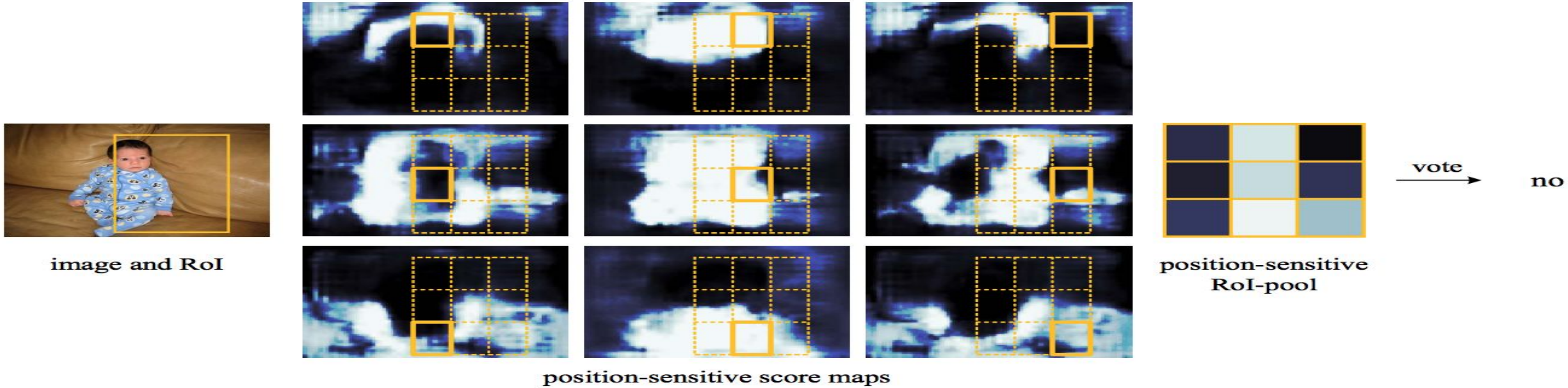


- Feature detected=>PSSM => Average Pooling => Softmax
- FC layer removed

RFCN : Example



Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.



Visualization when an RoI does not correctly overlap the object.

RFCN : Bounding Box

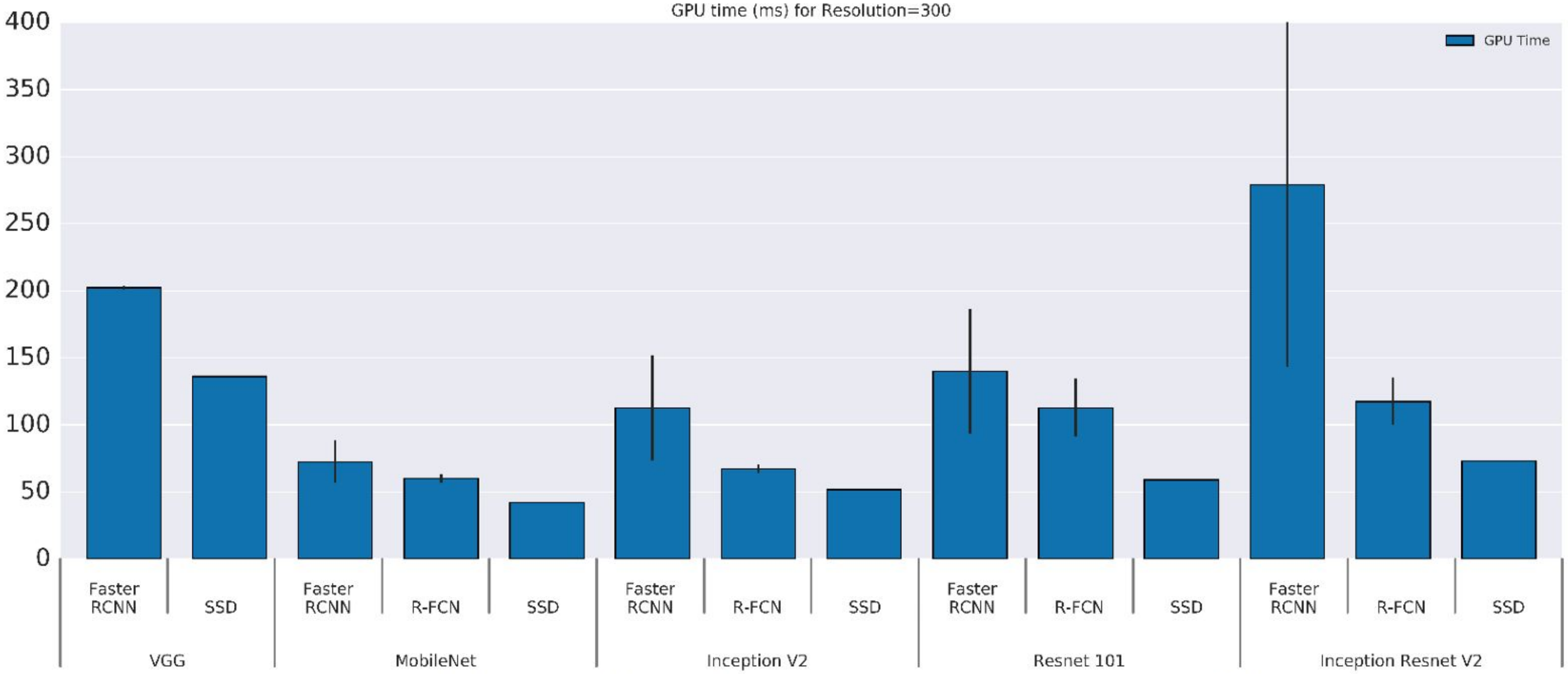
- To perform bounding box regression , another filter 4*k*k is used and position based pooling is applied to generate bounding box

RFCN : Performance

method	RoI output size ($k \times k$)	mAP on VOC 07 (%)
naïve Faster R-CNN	1×1	61.7
	7×7	68.9
class-specific RPN	-	67.6
R-FCN (w/o position-sensitivity)	1×1	*fail*
R-FCN	3×3	75.5
	7×7	**76.6**

- mAP increases with increase in size of k and at 7*7 size R-FCN is better than Faster-RCNN

RFCN : Performance



GPU time (milliseconds) for each model, for image resolution of 300.

Faster-RCNN vs R-FCN pcode

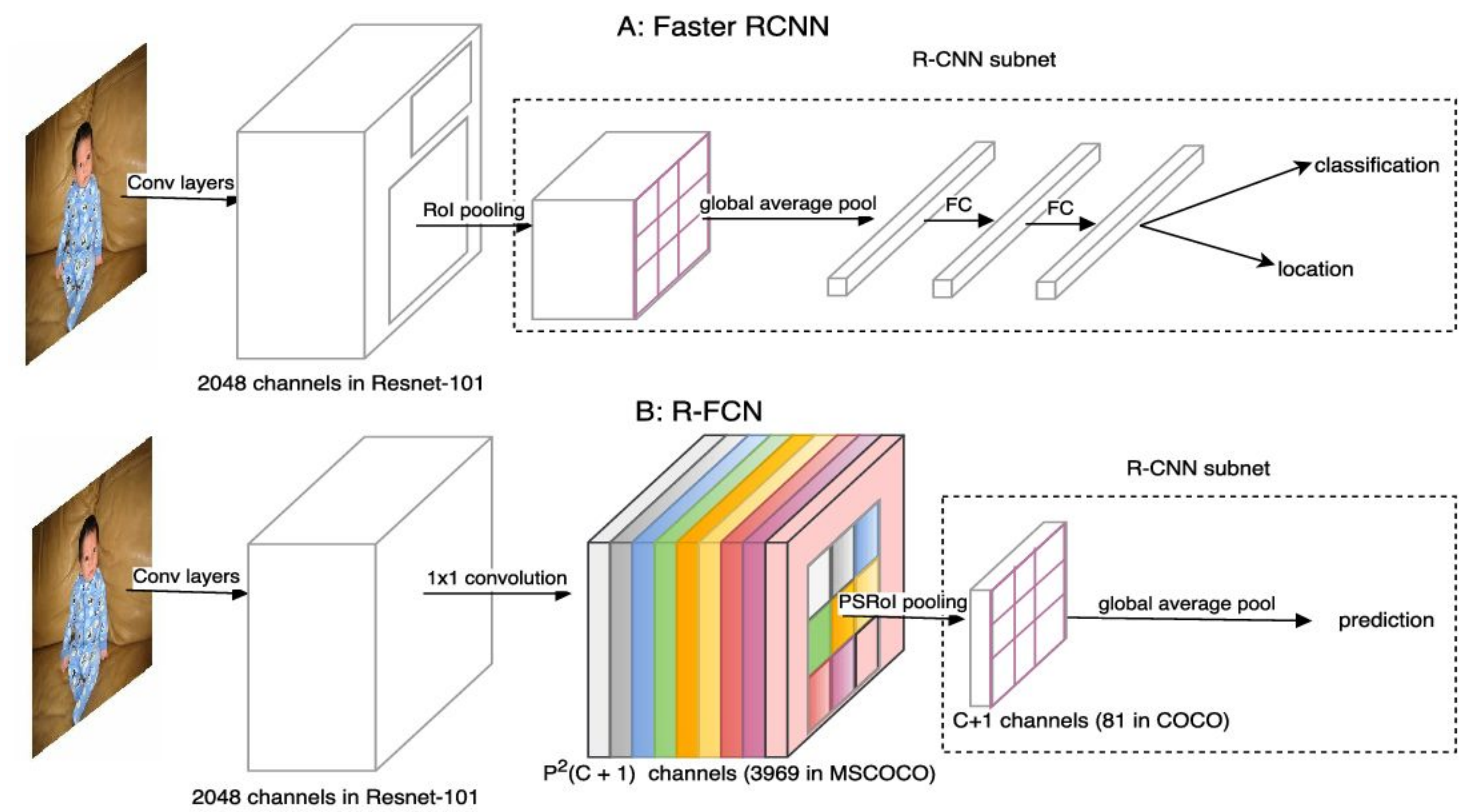
```
feature_maps = process(image)
ROIs = region_proposal(feature_maps)
for ROI in ROIs
    patch = roi_pooling(feature_maps, ROI)
    class_scores, box = detector(patch)
    class_probabilities = softmax(class_scores)
```

- RPN used and then feature extracted using feature map are given to ROI Pooling
- Followed by FC and classifier/regressor

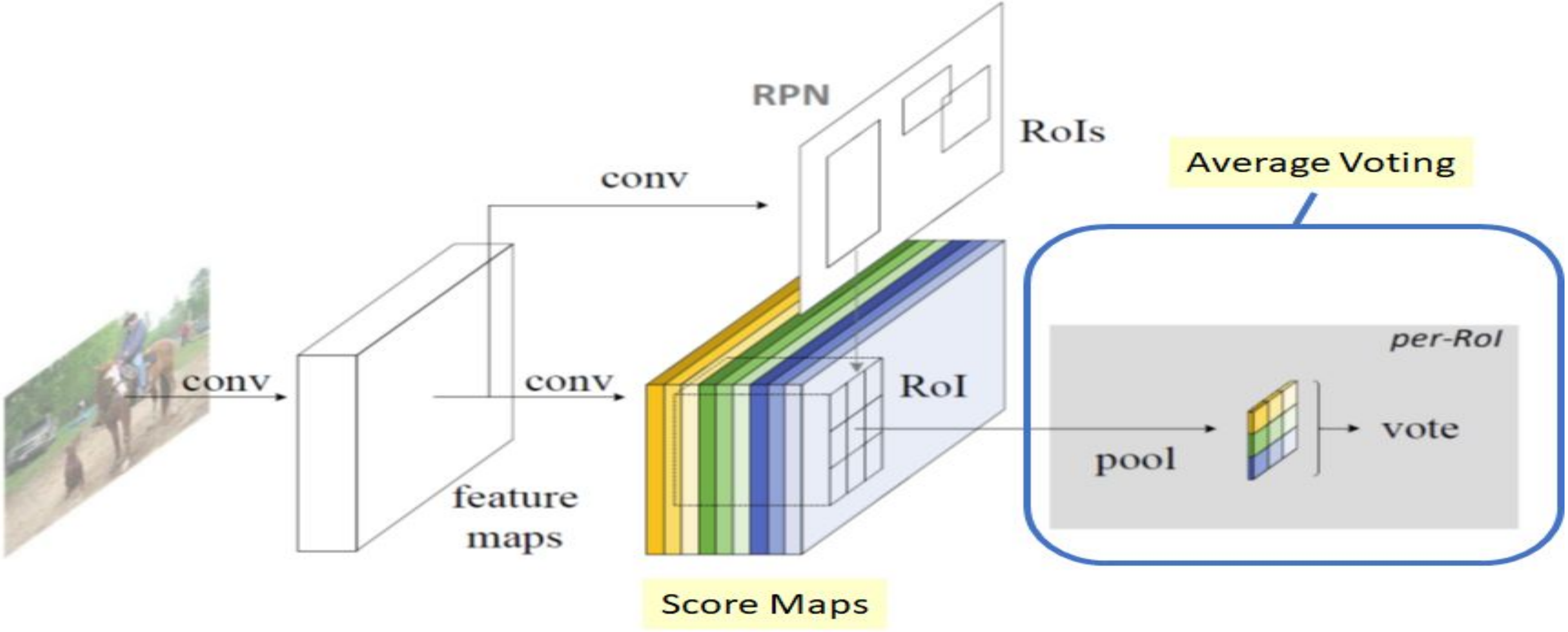
```
feature_maps = process(image)
ROIs = region_proposal(feature_maps)
score_maps = compute_score_map(feature_maps)
for ROI in ROIs
    V = region_roi_pool(score_maps, ROI)
    class_scores, box = average(V)
    class_probabilities = softmax(class_scores)
```

- Score map is calculated for each feature map and provides as input to ROI pooling layer
- FC layer removed and makes it faster

Faster-RCNN / RFCN



R-FCN Network



Why SSD

- Single Shot because it takes one single pass throughout the image to detect multiple objects in image along with their boundaries
- While in case of RCNN ,Faster-RCNN Selective search or RPN network was used, which make them double pass
- Selective search and RPN are reasons because of which Region based networks are slow
-

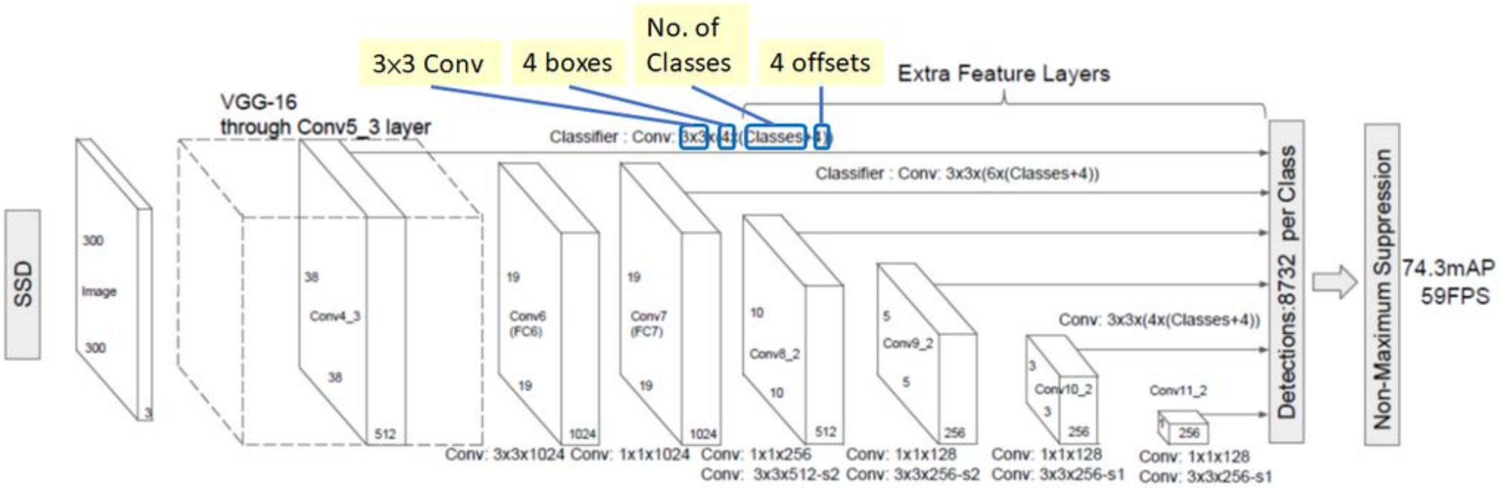
SSD : Single Shot Multibox detector

Single Shot : Localization and Classification both happens at the same time

Multibox : Multiple anchor boxes are used to predict the bounding box of the object

Detector : final output of network is a detected object along with its boundary and confidence

SSD Architecture



- VGG16 was used as base network to extract features and discarded fully connected layers
- Auxiliary conv layers are used in place of FC layers , to extract features at multiple scale

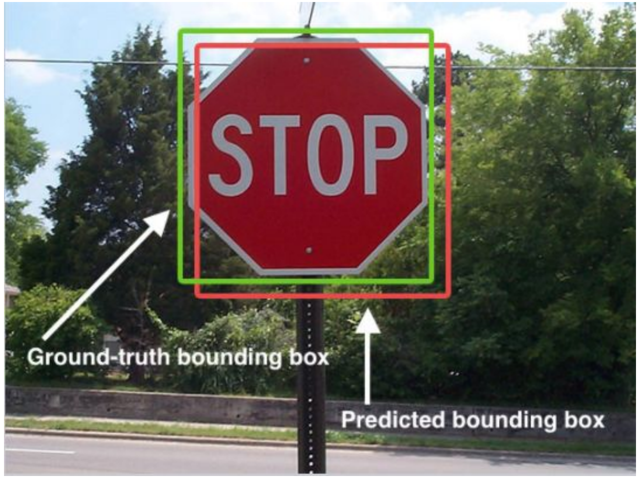
SSD : Bounding Box count

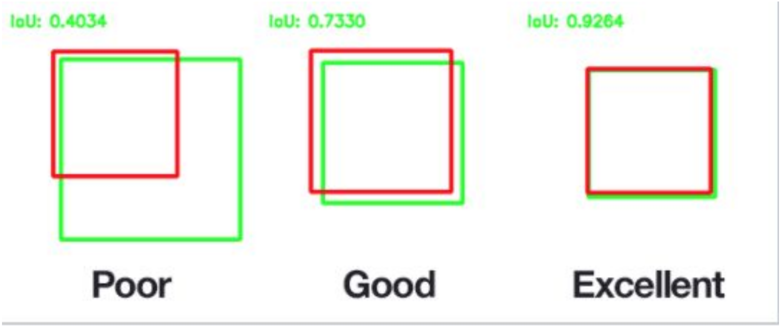
- At Conv4_3 layer : 38*38*4 = 5776 and similarly for others
- At Conv10_2 layer : 3*3*2 = 18
- At last layer conv_11 = 1*1*4 =4
- So total , SSD predicts 8732 bounding boxes

Anchor Box / Prior / default box

- Fixed size bounding boxes that closely matches the GT box
- Selected in such a way that IOU > 0.5
- Prior gives a good starting point for training instead of doing heavy operations like Selective search / RPN
- Prior applied over each cell and selected having IOU > 0.5
- Original arch contains 11 prior per feature map(8*8,6*6,4*4,3*3,2*2) and 1 for 1*1 feature map
-

SSD : Prios/Anchor Box





- Prior > 0.5 good for training

SSD Multibox LOSS

- Confidence Loss : To measure how much confident network is for presence of object
- Location Loss : To measure how far network predicted the box from GT

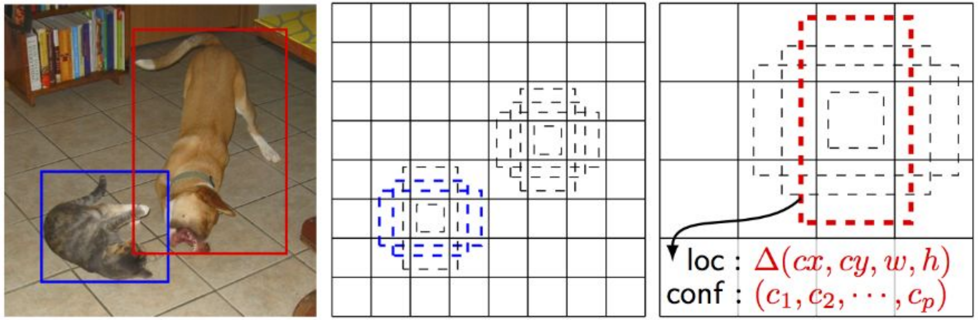
Multibox_Loss = Confidence Loss + alpha*(location Loss)

Alpha : Helps in balancing the location loss

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

SSD : Fixed Priors count

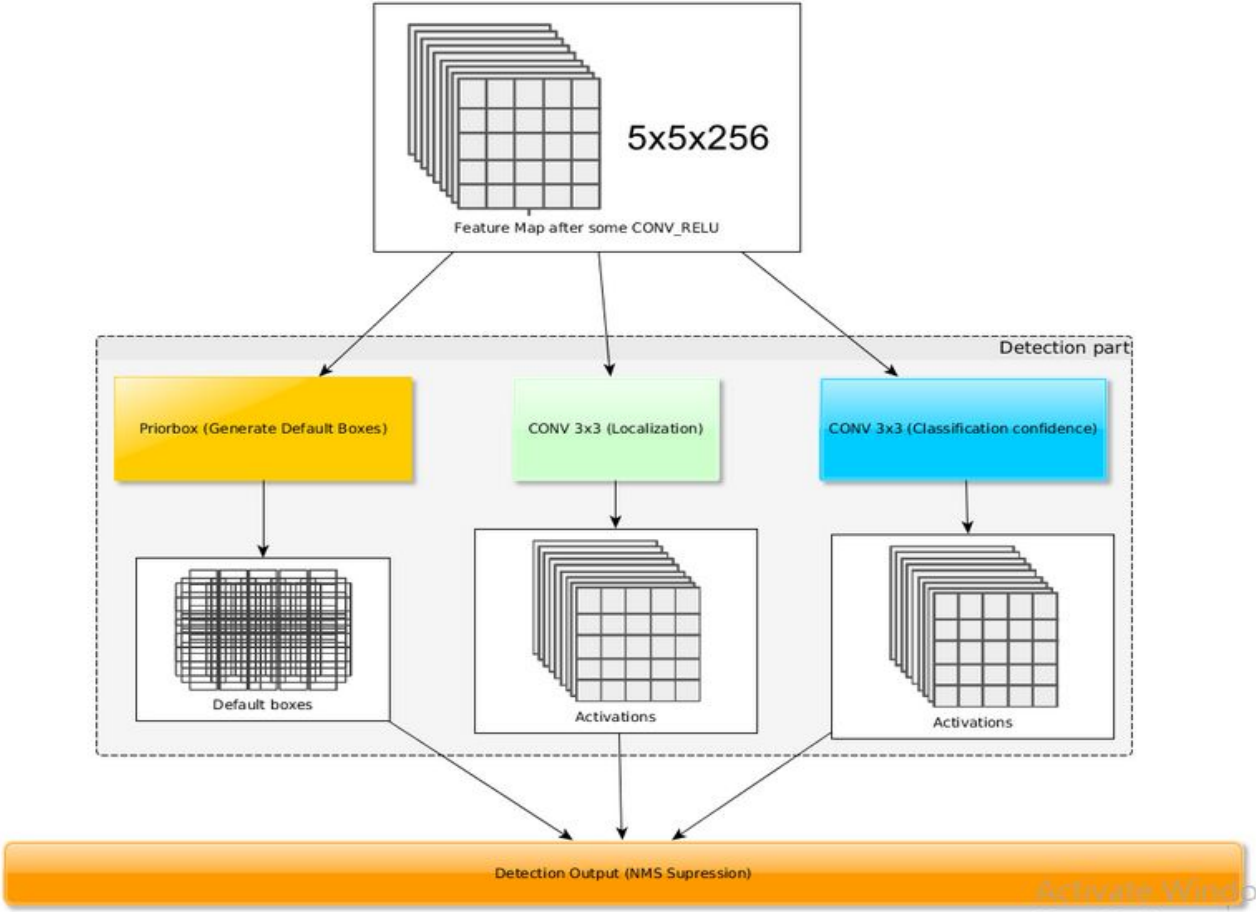
- Assuming we have **b** priors
- Feature map size is m*n
- C , number of classes to compute
- Feature map size f = m*n
- So total no of calculation : f*b*(4+c) per feature map



(a) Image with GT boxes (b) 8 × 8 feature map (c) 4 × 4 feature map

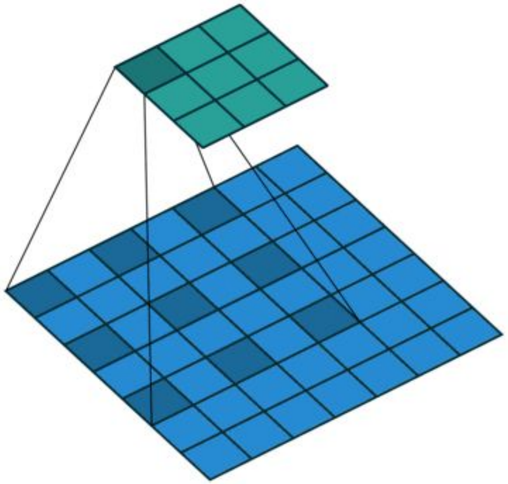
SSD default boxes at 8x8 and 4x4 feature maps

SSD : Detection Node



Key Features :

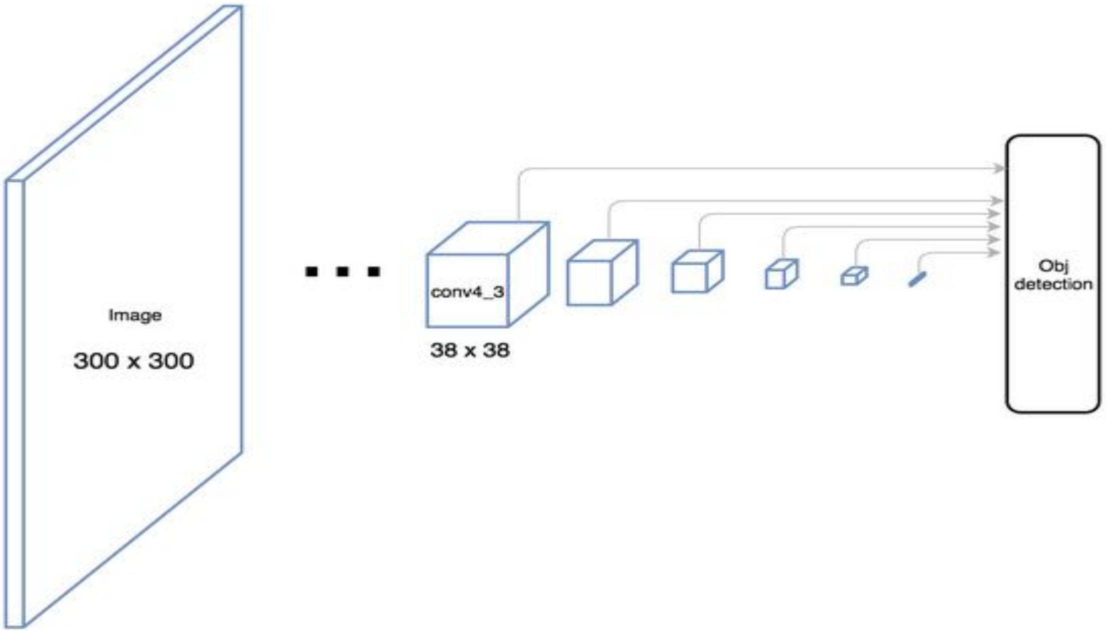
- **Hard Negative Mining :** sort negative examples on the basis of confidence loss and consider top ones only such that ratio remains 3 : 1
- **Data Augmentation :** Scales / Flipped / Contrast etc .
- **Dilated Conv :** At layer C6 and C7 dilated conv is used as feature map size is very large
- **NMS**
- **No FC Layer**
- **One pass only**



SSD Comparison chart

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

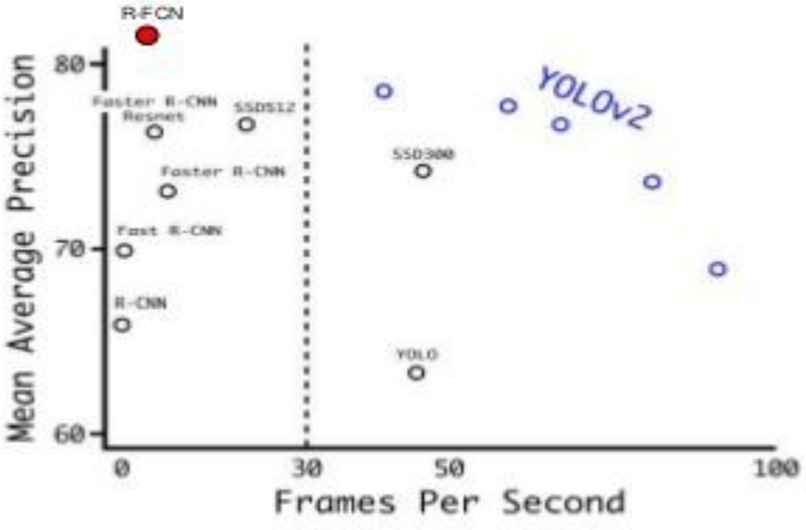
SSD Cons



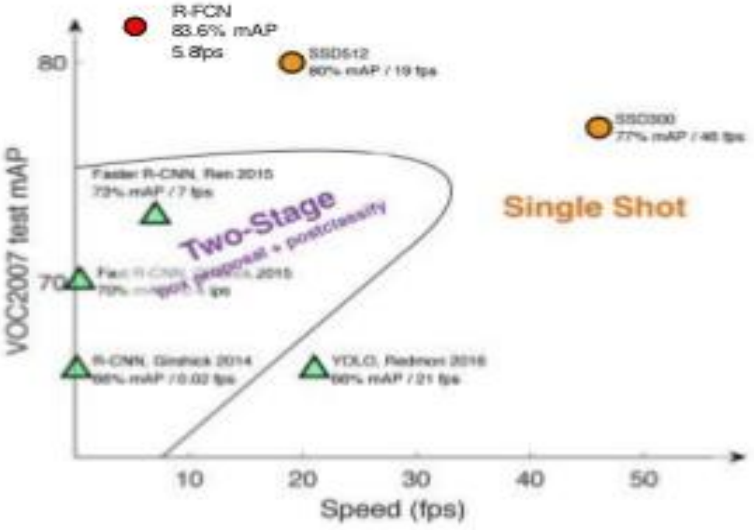
- Higher resolution Feature maps are responsible for small objects
- Conv4_3 has size 38*38 which is very small , due to which it does not perform good on small objects

Object Detection Network Benchmark

Comparison



From YOLOv2



From SSD