CS-671: Deep Learning and its Applications

Lecture: 12

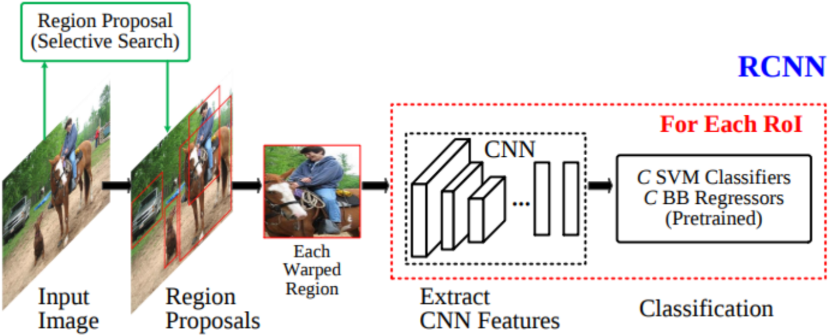
Single Shot Multi Box Detector (SSD)

Aditya Nigam, Assistant Professor
School of Computing and Electrical Engineering (SCEE)
Indian Institute of Technology, Mandi
http://faculty.iitmandi.ac.in/ãditya/ aditya@iitmandi.ac.in

Presentation for CS-671@IIT Mandi (26 March, 2019)

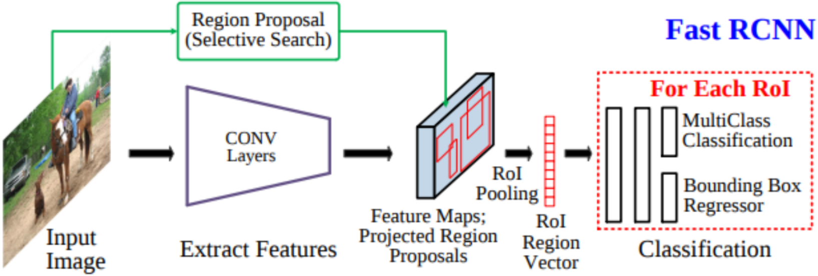
February - May, 2019

RCNN

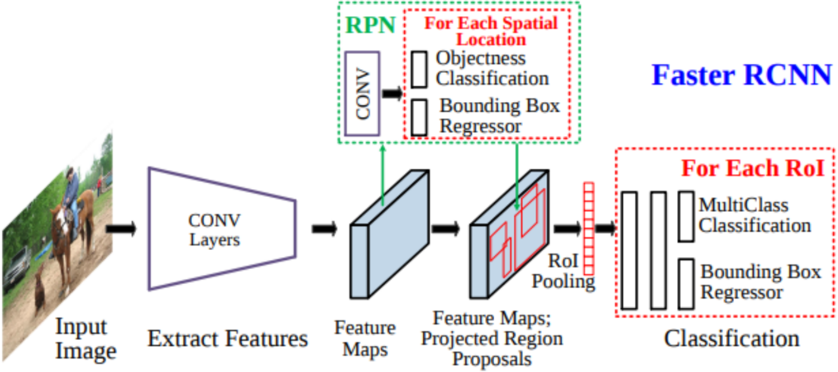


Region Proposal
(Selective Search)
RCNN
For Each RoI
CNN
...
C SVM Classifiers
C BB Regressors
(Pretrained)
Input
Image
Region
Proposals
Each
Warped
Region
Extract
CNN Features
Classification

Fast RCNN

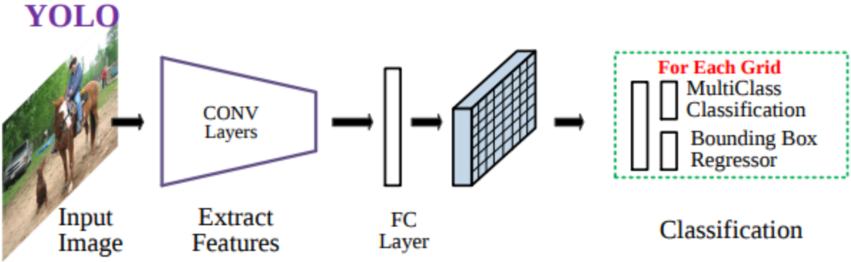

Region Proposal
(Selective Search)
Fast RCNN
CONV
Layers
For Each RoI
MultiClass
Classification
Bounding Box
Regressor
RoI
Pooling
Input
Image
Extract Features
Feature Maps;
Projected Region
Proposals
RoI
Region
Vector
Classification

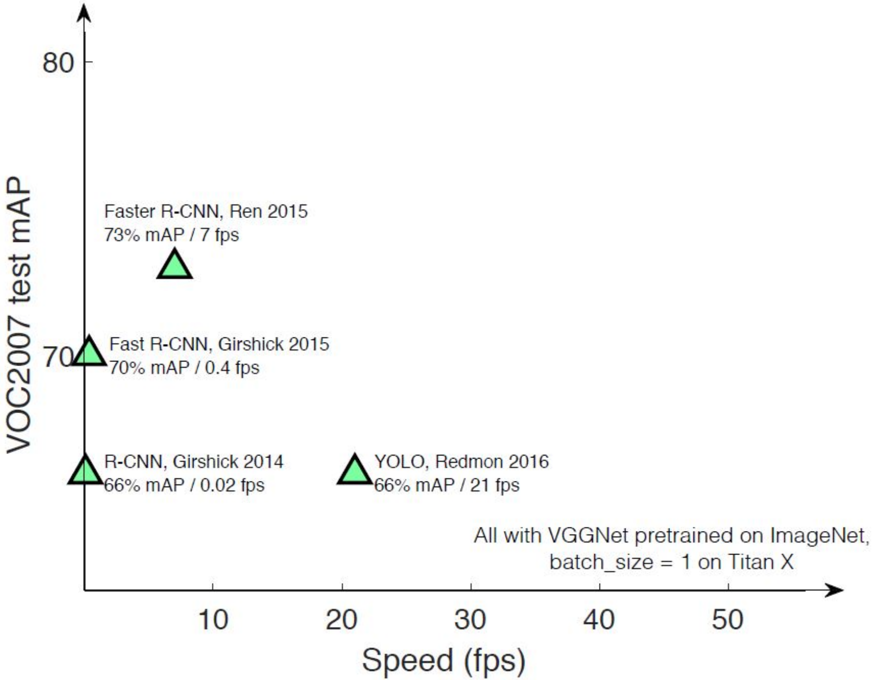
Faster RCNN



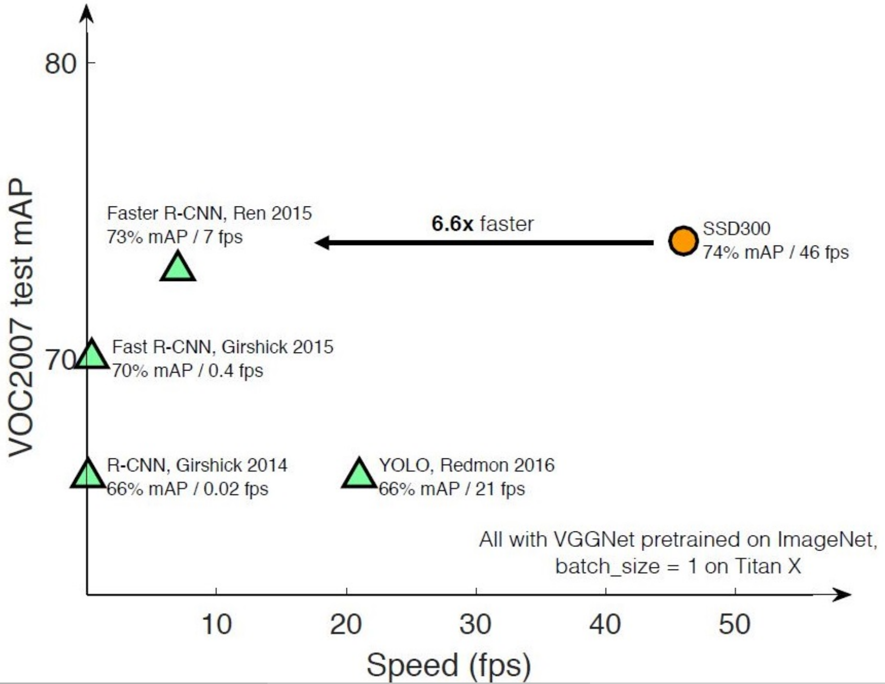
RPN
For Each Spatial Location
CONV
Objectness Classification
Bounding Box Regressor
Faster RCNN
For Each RoI
MultiClass Classification
Bounding Box Regressor
CONV Layers
RoI Pooling
Input Image
Extract Features
Feature Maps
Feature Maps; Projected Region Proposals
Classification

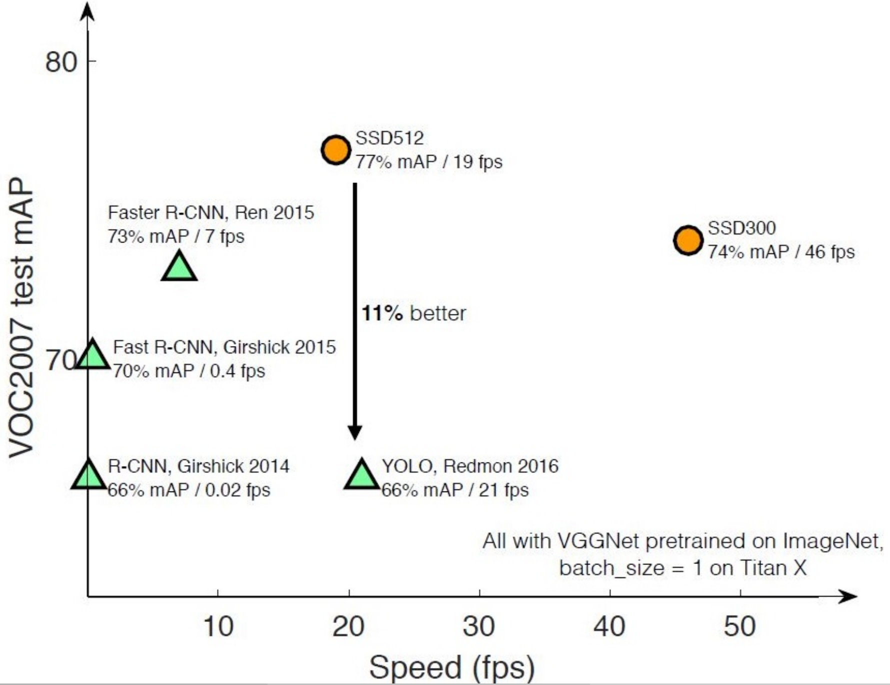
YOLO - You only Look Once

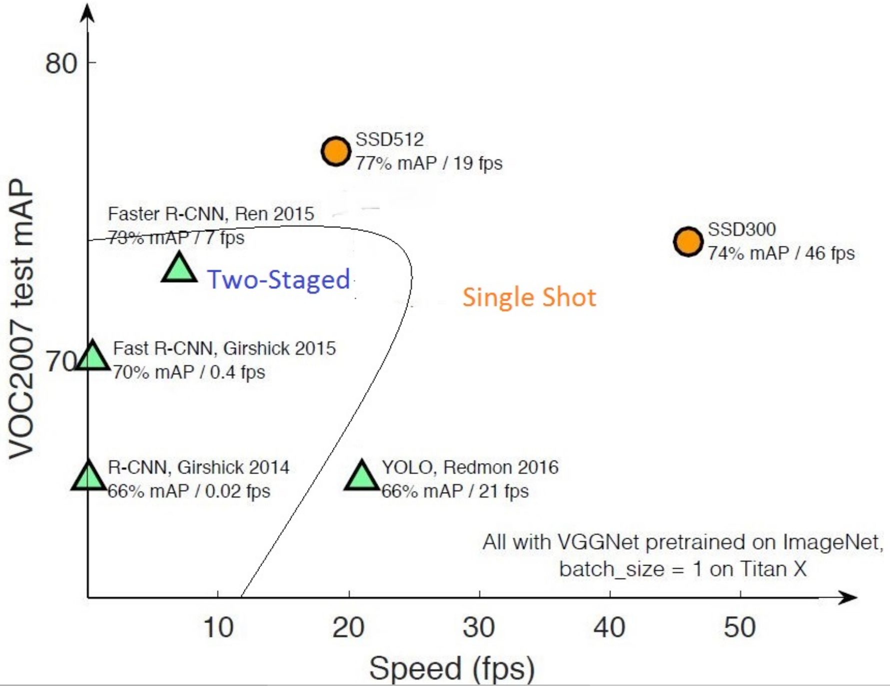




VOC2007 test mAP
80
70
Faster R-CNN, Ren 2015
73% mAP / 7 fps
Fast R-CNN, Girshick 2015
70% mAP / 0.4 fps
R-CNN, Girshick 2014
66% mAP / 0.02 fps
YOLO, Redmon 2016
66% mAP / 21 fps
All with VGGNet pretrained on ImageNet,
batch_size = 1 on Titan X
10
20
30
40
50
Speed (fps)


80
70
VOC2007 test mAP
Faster R-CNN, Ren 2015
73% mAP / 7 fps
6.6x faster
SSD300
74% mAP / 46 fps
Fast R-CNN, Girshick 2015
70% mAP / 0.4 fps
R-CNN, Girshick 2014
66% mAP / 0.02 fps
YOLO, Redmon 2016
66% mAP / 21 fps
All with VGGNet pretrained on ImageNet,
batch_size = 1 on Titan X
10
20
30
40
50
Speed (fps)


VOC2007 test mAP
80
70
SSD512
77% mAP / 19 fps
Faster R-CNN, Ren 2015
73% mAP / 7 fps
SSD300
74% mAP / 46 fps
11% better
Fast R-CNN, Girshick 2015
70% mAP / 0.4 fps
R-CNN, Girshick 2014
66% mAP / 0.02 fps
YOLO, Redmon 2016
66% mAP / 21 fps
All with VGGNet pretrained on ImageNet,
batch_size = 1 on Titan X
10
20
30
40
50
Speed (fps)

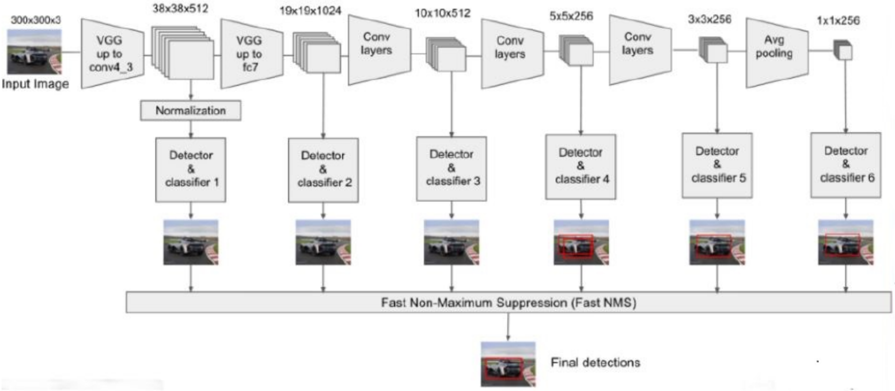


80
70
VOC2007 test mAP
SSD512
77% mAP / 19 fps
Faster R-CNN, Ren 2015
73% mAP / 7 fps
Two-Staged
Single Shot
SSD300
74% mAP / 46 fps
Fast R-CNN, Girshick 2015
70% mAP / 0.4 fps
R-CNN, Girshick 2014
66% mAP / 0.02 fps
YOLO, Redmon 2016
66% mAP / 21 fps
All with VGGNet pretrained on ImageNet,
batch_size = 1 on Titan X
10
20
30
40
50
Speed (fps)

Architecture

- Base network of VGG-16.
- Auxiliary structure for detection.

Architecture

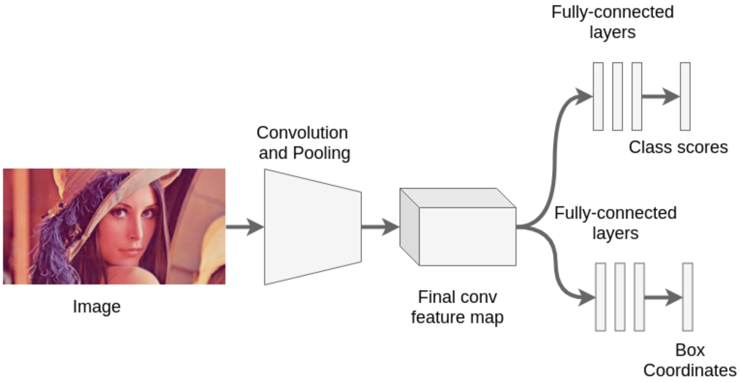


Architecture

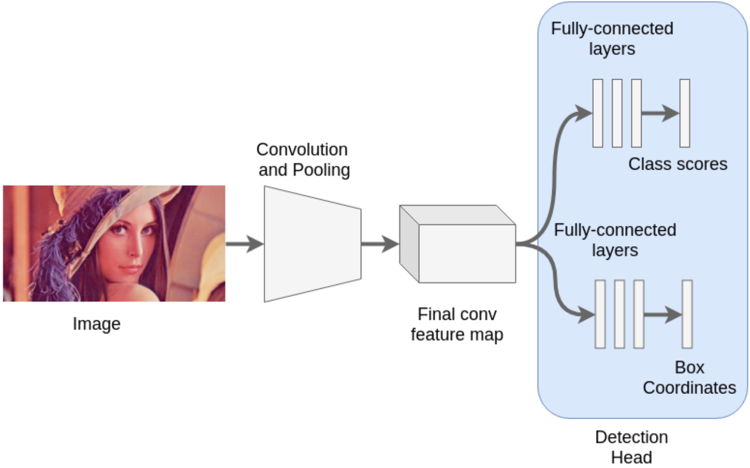
- Convolutional layers in Auxiliary network are 1x1 convolution with stride 2.
- They create feature maps with decreasing sizes.
- These varying sizes feature maps are used for scale variance of objects.
- Detector and classifier will be applied on each feature map.
- Let a feature map be of size $mxnxp$

The detector will be a convolutional layer with filter of $3x3xp$.

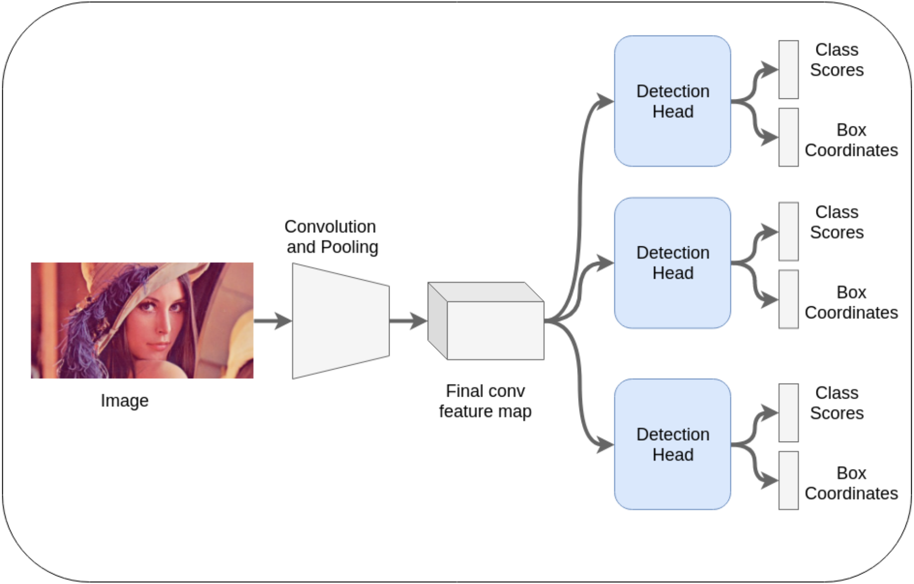
Understanding Detector and Classifier



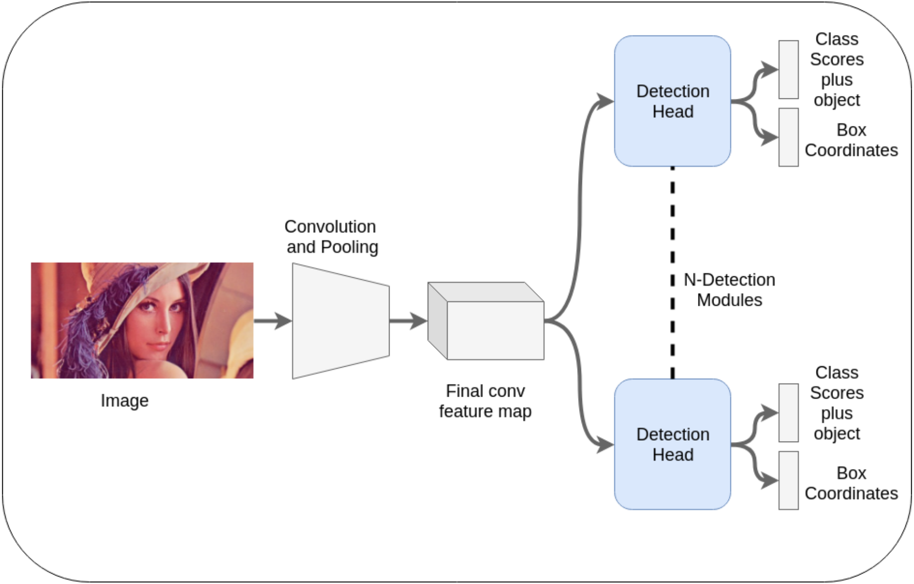
Understanding Detector and Classifier



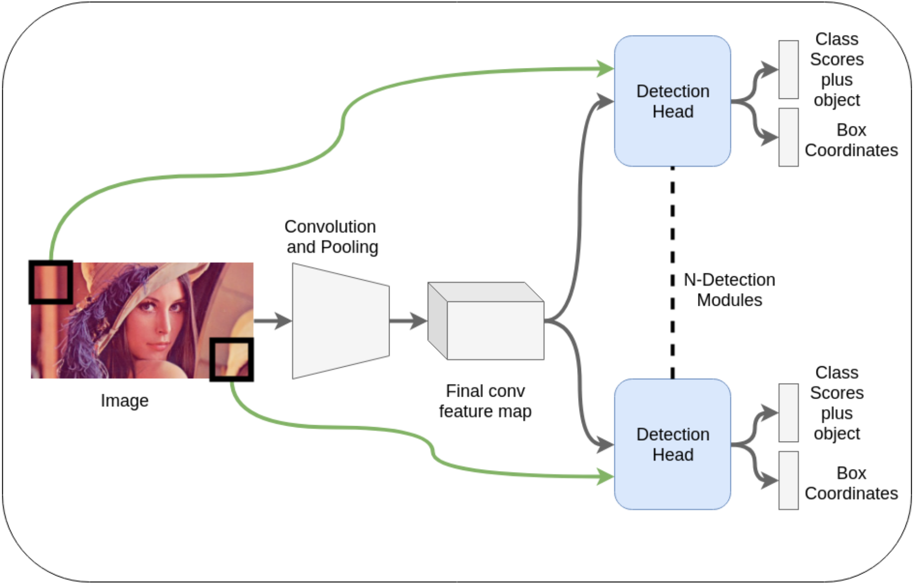
Understanding Detector and Classifier



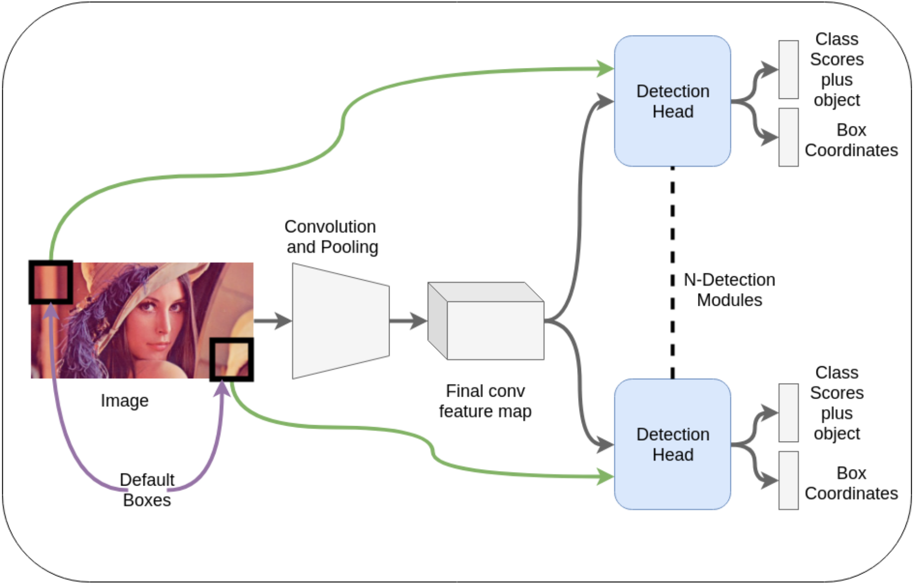
Understanding Detector and Classifier



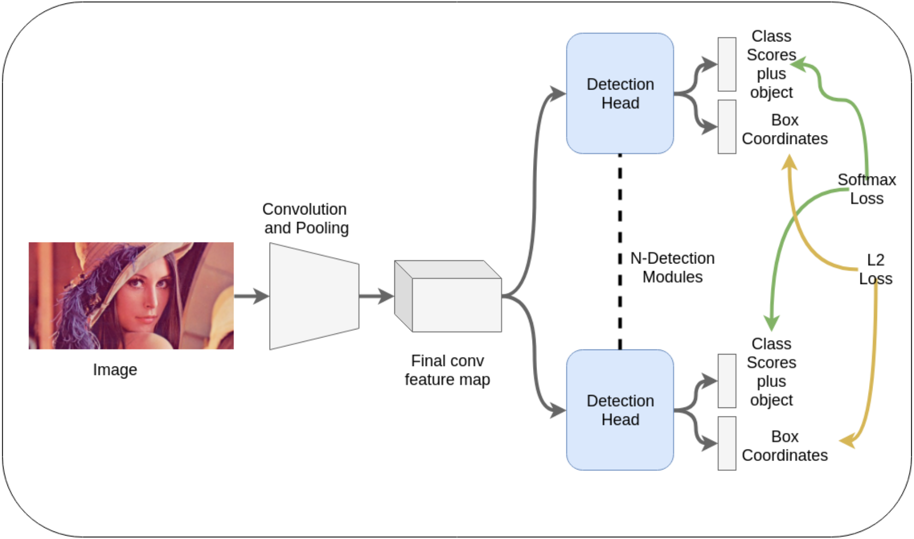
Understanding Detector and Classifier



Understanding Detector and Classifier

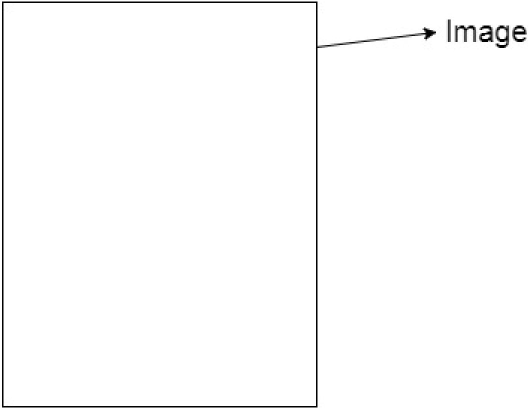


Understanding Detector and Classifier



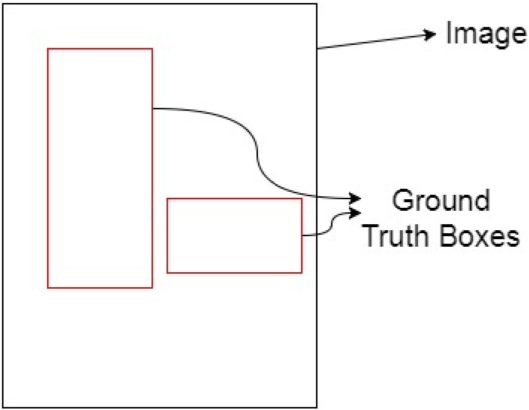
Training

- SSD input is a image having ground truth boxes.



Training

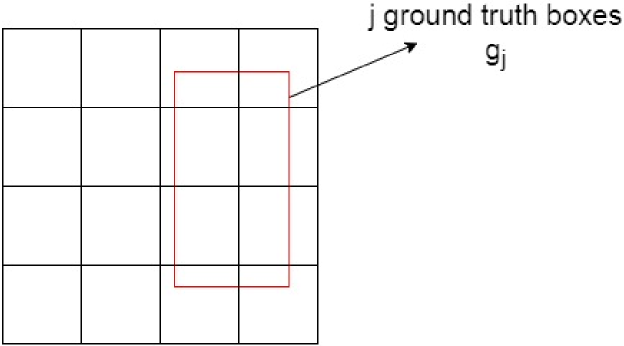
- SSD input is a image having ground truth boxes.



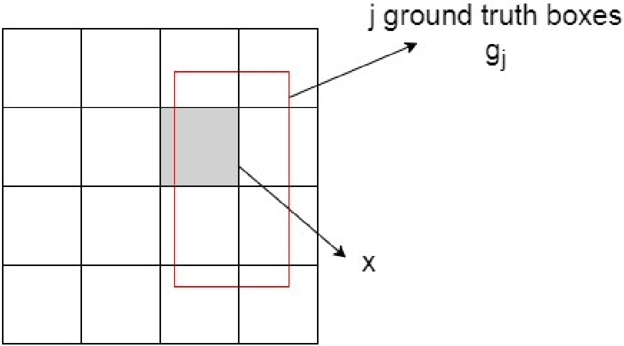
Training

- For a particular feature map from auxiliary network.

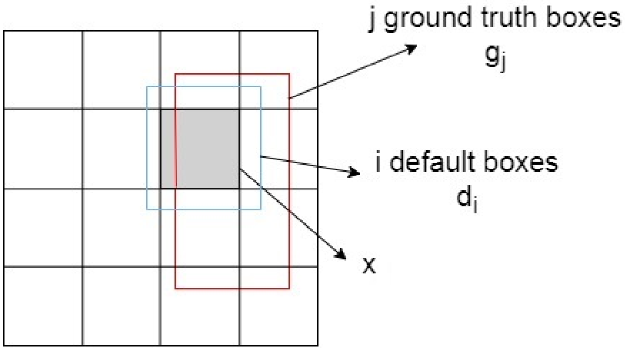
Training



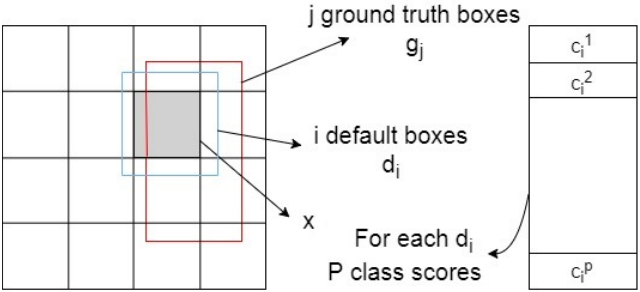
Training



Training



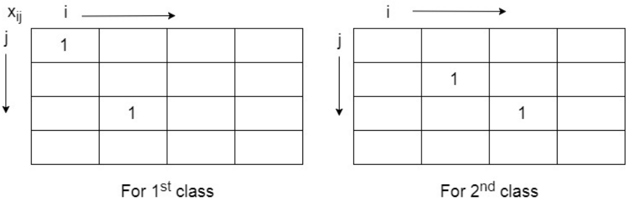
Training



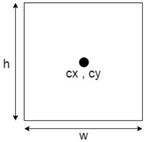
j ground truth boxes
g_j
i default boxes
d_i
x
For each d_i
P class scores
c_i^1
c_i^2
c_i^p

Training

- There will be huge number of bounding boxes.
- We handle them by matching.
- The i^{th} default box is matched to j^{th} ground truth box using Jaccard Index that is IOU (Intersection over Union).
  If $IOU > 0.5, x_{ij} = 1$
  Else, $x_{ij} = 0$



Training



- Corresponding every default box d_i, we calculate a predicted box l_i having 4 parameters cx, cy, w and h
  cx =Centre x coordinate
  cy =Centre y coordinate
  w =width
  h =height
- Every box also contains class scores.
  Let there be p class scores
  Total number of parameters per box$= p + 4$
- Let a feature map be mxn size.
  Total number of parameters $= (p + 4).m.n.\#$default boxes

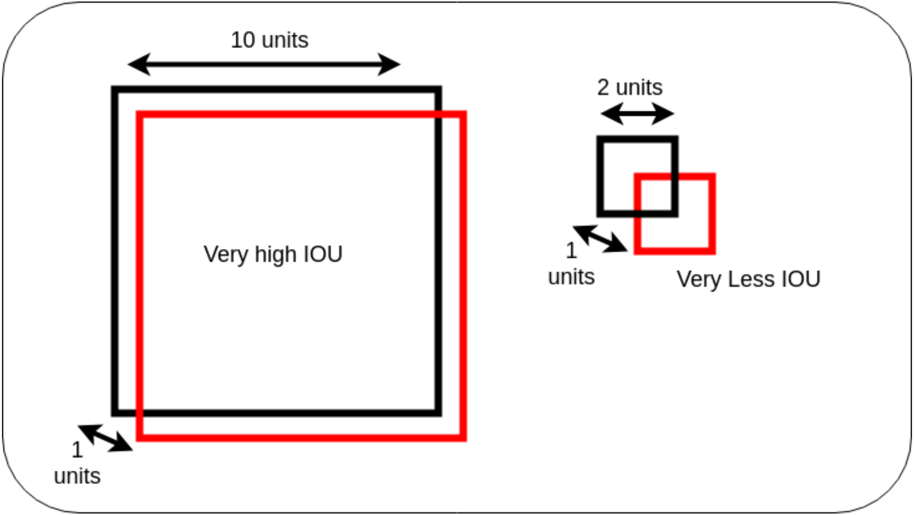
Loss Function

- Let N be the number of total boxes with Jaccard Index > 0.5.
- We have 2 losses
  - Location Loss
  - Confidence Loss
- $L(x, c, l, g) = \frac{1}{N}[L_{conf}(x, c) + \alpha L_{loc}(x, l, g)]$
  - $N =$ number of matched boxes
  - $x =$ pixel under consideration
  - $c =$ class scores
  - $l =$ predicted boxes
  - $g =$ Ground truth boxes

Loss Function

- Calculate Smooth $L1$ loss between each parameter of Predicted box l_i and Ground Truth box g_j.
  $(l_i^{cx} - g_j^{cx}) SmoothL1$ $\quad$ $(l_i^{cy} - g_j^{cy}) SmoothL1$
  $(l_i^{w} - g_j^{w}) SmoothL1$ $\quad$ $(l_i^{h} - g_j^{h}) SmoothL1$
- Multiply each with $x_{ij} = 0, 1$ and add all.
- Repeat above steps $\forall i \in Pos$

Loss Function



Loss Function

- Normalization: First we will normalize the box parameters.

$$\hat{g}_j^{cx} = \frac{(g_j^{cx} - d_i^{cx})}{d_i^w} \qquad \hat{g}_j^{cy} = \frac{(g_j^{cy} - d_i^{cy})}{d_i^h}$$

$$\hat{g}_j^{w} = log(\frac{g_j^w}{d_i^w}) \qquad \hat{g}_j^{h} = log(\frac{g_j^h}{d_i^h})$$

  - d =Default boxes
  - cx, cy =Centre of boxes
  - w, h =Width and height of boxes
- Similarly we will normalize the parameters of predicted box $\hat{l}_i^{cx}, \hat{l}_i^{cy}, \hat{l}_i^{w}, \hat{l}_i^{h}$

Loss Function

- Calculate Smooth $L1$ loss between each parameter of Normalised Predicted box $\hat{l}_i$ and Normalised Ground Truth box $\hat{g}_j$.
  $(\hat{l}_i^{cx} - \hat{g}_j^{cx}) SmoothL1$ $(\hat{l}_i^{cy} - \hat{g}_j^{cy}) SmoothL1$
  $(\hat{l}_i^{w} - \hat{g}_j^{w}) SmoothL1$ $(\hat{l}_i^{h} - \hat{g}_j^{h}) SmoothL1$
- Multiply each with $x_{ij} = 0, 1$ and add all.
- Repeat above steps $\forall i \in Pos$

Loss Function

- Confidence loss: For each box i, we have p confidence scores c_i^p, where,
  $c_i^1 =$ Confidence of class 1
  $c_i^2 =$ Confidence of class 2
  $c_i^p =$ Confidence of class p

- Softmax loss over

$$c_i^p : \hat{c}_i^p = \frac{e^{(c_i^p)}}{\sum_p e^{(c_i^p)}}$$

Loss Function

- We have to maximize confidence of matched predictions (Pos).
- At same time minimize the confidence of remaining predictions (Neg).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^{p}) - \sum_{i \in Neg} log(\hat{c}_i^{0})$$

Choosing Scales

- Let there be m feature maps. $m = 6$ in paper.
- k be the map we want to find the scale of box in $k \in [1, m]$.
- Let S_k be scale at k^{th} map
- S_{min} =Minimum Scale$= 0.2$ S_{max} =Maximum Scale$= 0.9$

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m - 1}(k - 1)$$

Aspect Ratio

- For k^{th} scale, we have,
  $w_k^1, w_k^2, ..., w_k^a$ widths $h_k^1, h_k^2, ..., h_k^a$ heights
- Choose a value of a_r such that $a_r \in [1, 2, 3, \frac{1}{2}, \frac{1}{3}]$

$$h_k^a = \frac{S_k}{\sqrt{a_r}} \qquad w_k^a = S_k\sqrt{a_r}$$

- Let $a_r = 1$

$$h_k^a = \frac{S_k}{\sqrt{1}} = S_k \qquad w_k^a = S_k\sqrt{1} = S_k$$

$$Aspect\ Ratio = 1$$

- Let $a_r = 2$

$$h_k^a = \frac{S_k}{\sqrt{2}} \qquad w_k^a = S_k\sqrt{2}$$

$$Aspect\ Ratio = 2:1$$

Number of Default Boxes

- For a given scale we can choose 5 different aspect ratios.
- For aspect ratio $= 1$, we add another box having $S'_k = \sqrt{S_k S_{k+1}}$
- Hence, we have 6 Default Boxes per feature map location.

Hard Negative Mining

- Number of negative samples will be much greater than positive samples.
- Sort the negative samples using confidence score for each default box.
- Pick the top ones to keep the ratio of negative to positive to atmost 3:1

Non Maximum Suppression

- Sort all boxes of a class using confidence scores.
- Calculate Jaccard Index of first box with every other box.
  - If overlap > 0.45, remove the other box.
  - Otherwise keep the other box.
- Repeat the above process for each box in sorted order.

Results on VOC

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Thank You.
Any Questions.