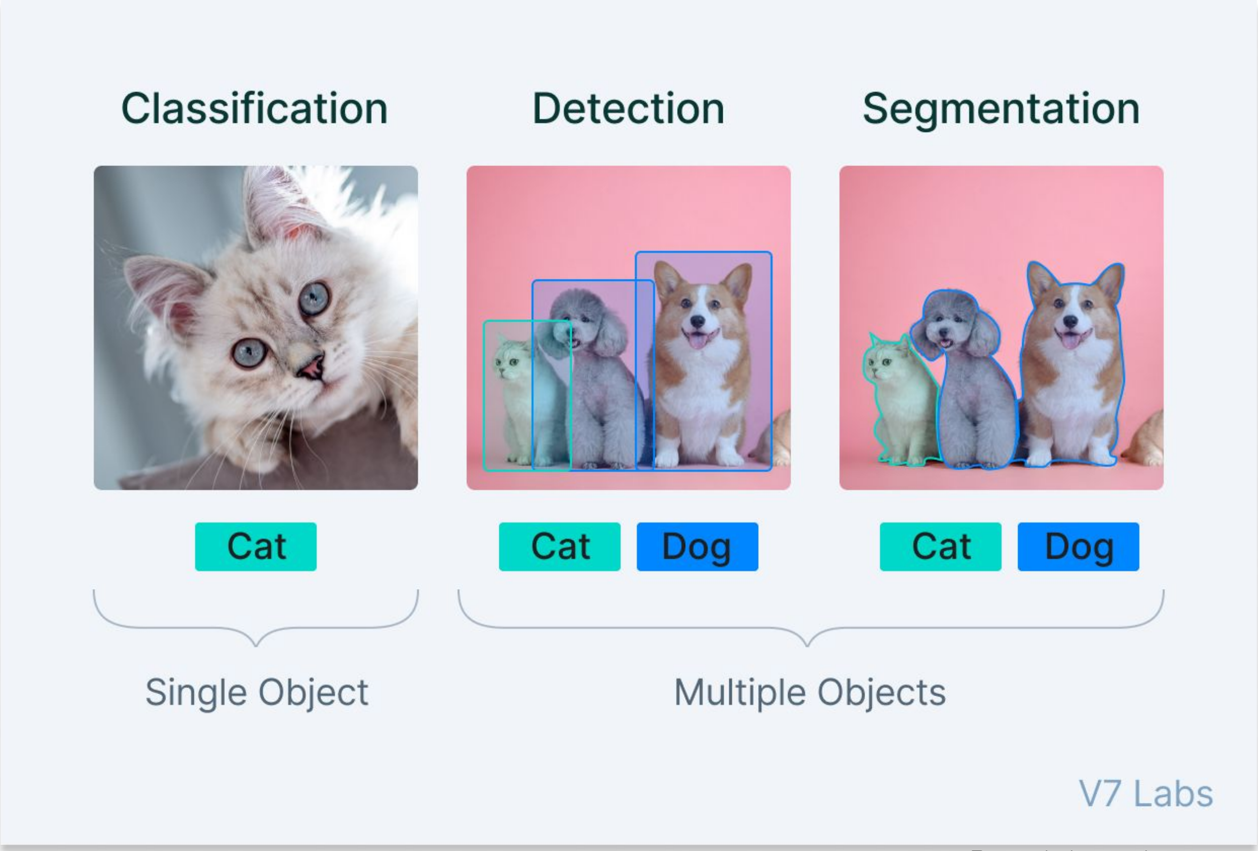
Object Detection

Dr. Konda Reddy Mopuri
Deep Learning for Computer Vision (DL4CV)
IIT Guwahati
Aug-Dec 2022

So far in Computer Vision

Dog: 0.1
Bird: 0.01
Car: 0.01
Cat: 0.8
Deer: 0.01
Truck: 0.01
....
....



No localization

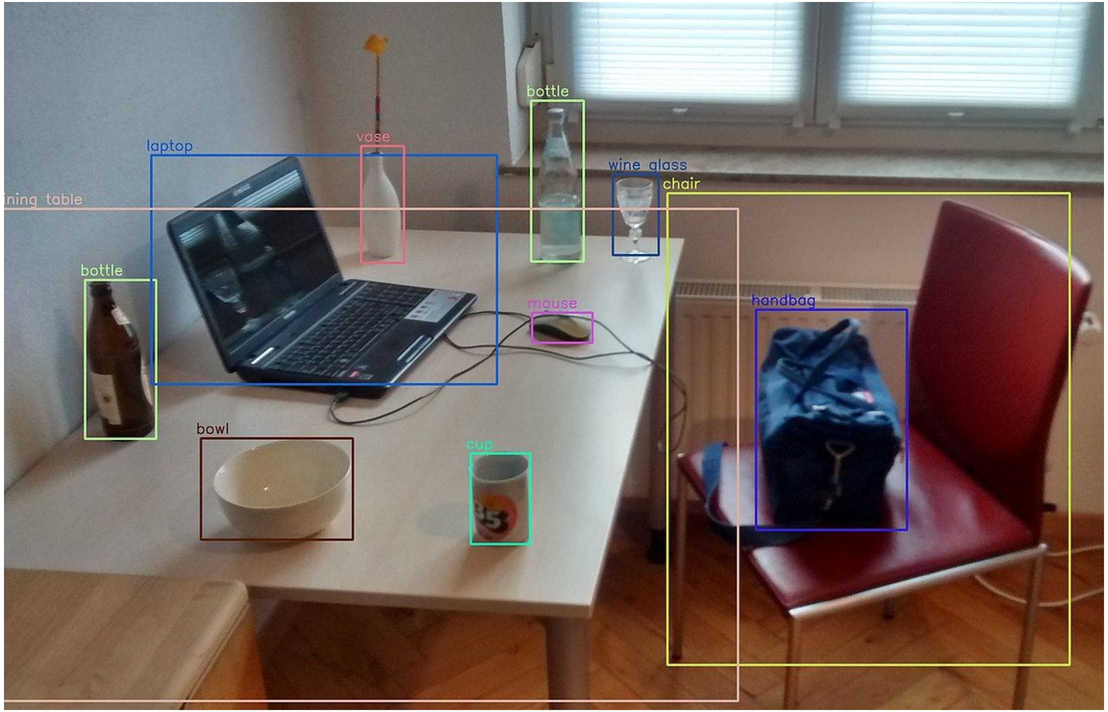
Bounding box localization

Pixel-level localization

This lecture: Object Detection

- Input: image
- Output: set of detected objects, for each
  - Class label: one from a predefined set of labels (similar to classification)
  - Bounding box: (x, y, w, h)

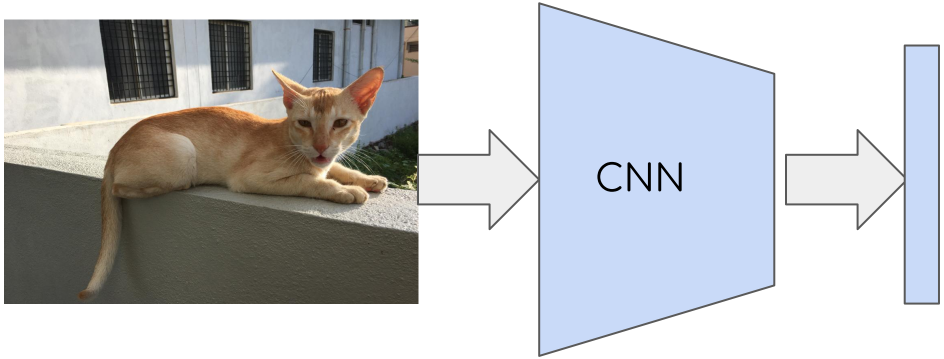
This lecture: Object Detection



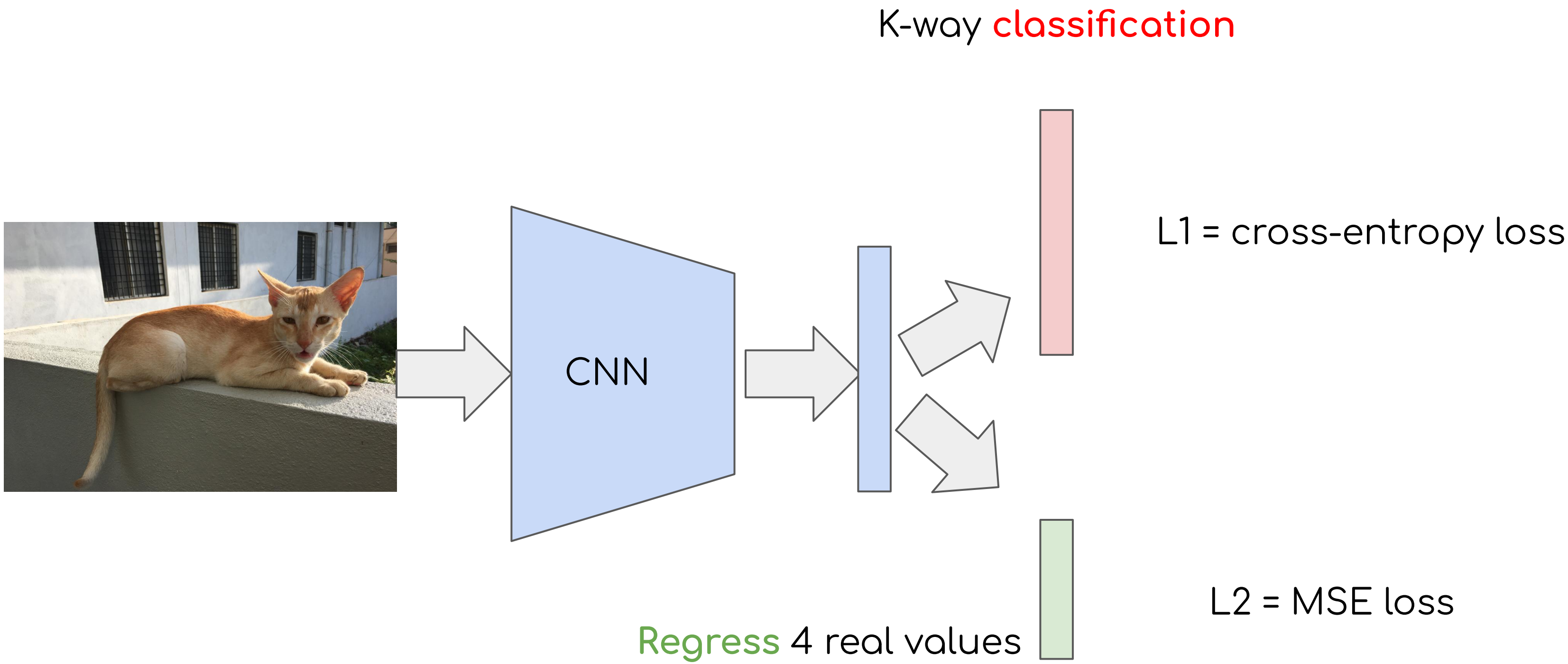
Object detection: Challenges

- Variable number of objects in each image
- For each object: two different kinds of predictions (label & coordinates)
- Typically works on high-res images

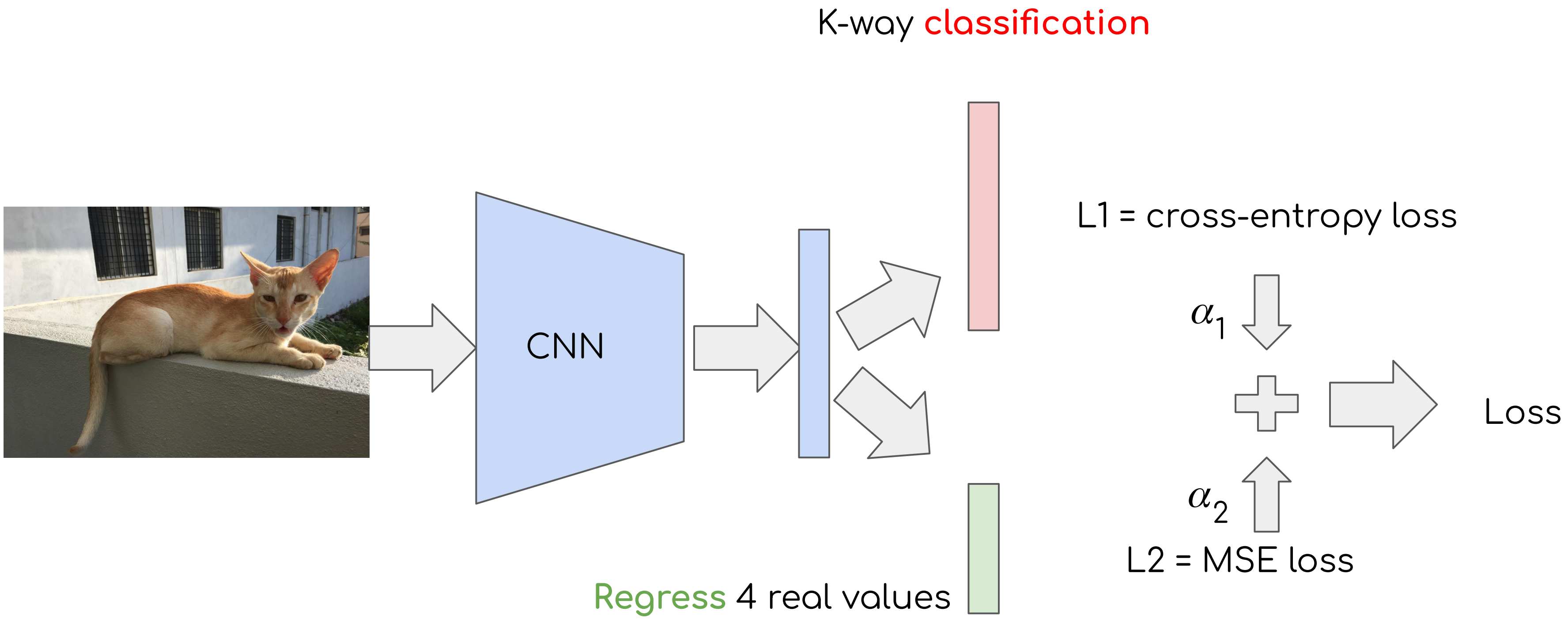
Detecting a single object



Detecting a single object



Detecting a single object



Detecting multiple objects

- Needs to be able to output variable number of predictions

Detecting multiple objects: Sliding window

- Probe different crops with a classification CNN

Cat:
Dog:
Person:
Car:
....
....
Background: ✓

Detecting multiple objects: Sliding window

- Probe different crops with a classification CNN

Cat:
Dog: ✓
Person:
Car:
....
....
Background:

Detecting multiple objects: Sliding window

- Probe different crops with a classification CNN

Cat:
Dog:
Person: ✓
Car:
....
....
Background:

Detecting multiple objects: Sliding window

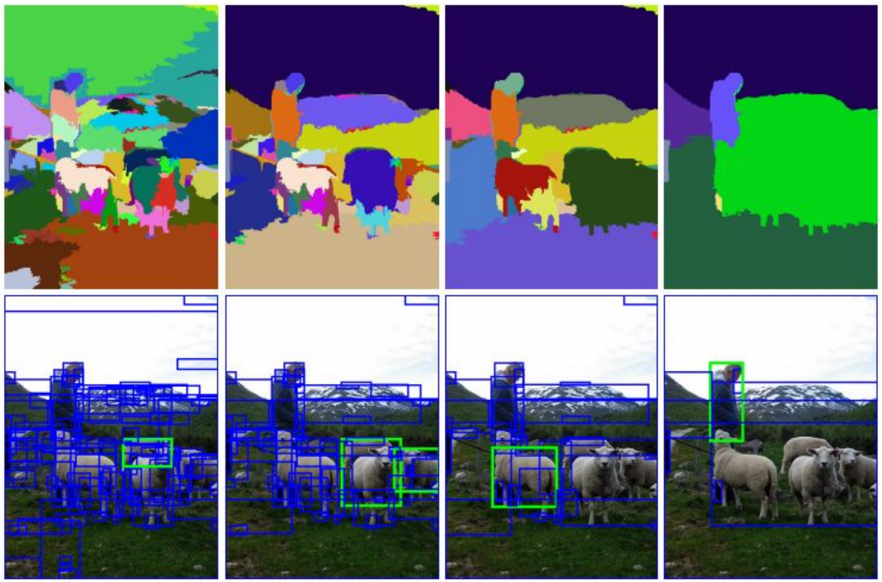
- Computationally very demanding
- Different sizes of possible boxes
- Total possible boxes: $O(W^2H^2)$
  - E.g. 800 X 600 image → 58M boxes!

Solution: Region Proposals

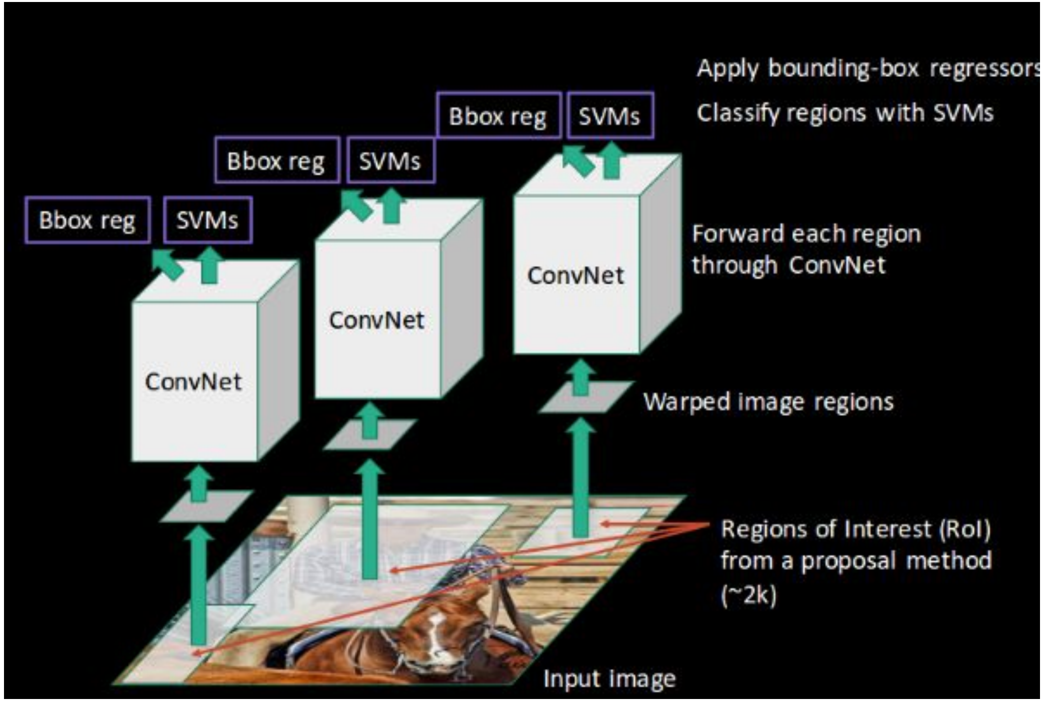
- Identify small set of potential boxes (that may contain the objects)
- Use low-level image processing cues (e.g. blob like regions)
- Faster processing (~1K/second on a cpu)

Solution: Region Proposals

- E.g. selective search by Uijlings et al. IJCV 2013



Region-based CNN (R-CNN) for detection



RCNN: Stage-1

- Feature extraction from the region proposals
- AlexNet is fine-tuned with N+1 neurons in the final layer
  - +ve ← GT of the BB with maximal IOU (>0.5) with the proposal
  - -ve ← background
- o/p: 4096D fine-tuned features for all the proposals

RCNN: Stage-2

- SVM for object classification
  - +ve: 4096D features of the GT BB
  - -ve: 4096D features of the proposals with <0.3 IOU
  - Rest of the proposals are ignored for training the SVM
- Set of +ve proposals for each class

RCNN: Stage-3

- BB regression: separate transformation for each class

$$\boxed{\begin{aligned} P^i &= (P^i_x, P^i_y, P^i_w, P^i_h) \\ G &= (G_x, G_y, G_w, G_h) \end{aligned}}$$

(1)

$$\boxed{\begin{aligned} t_x &= (G_x - P_x)/P_w \\ t_y &= (G_y - P_y)/P_h \\ t_w &= \log(G_w/P_w) \\ t_h &= \log(G_h/P_h). \end{aligned}}$$

(2)

$$\boxed{\begin{aligned} \hat{G}_x &= P_w d_x(P) + P_x \\ \hat{G}_y &= P_h d_y(P) + P_y \\ \hat{G}_w &= P_w \exp(d_w(P)) \\ \hat{G}_h &= P_h \exp(d_h(P)). \end{aligned}}$$

(3)

$$\boxed{d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P)}$$

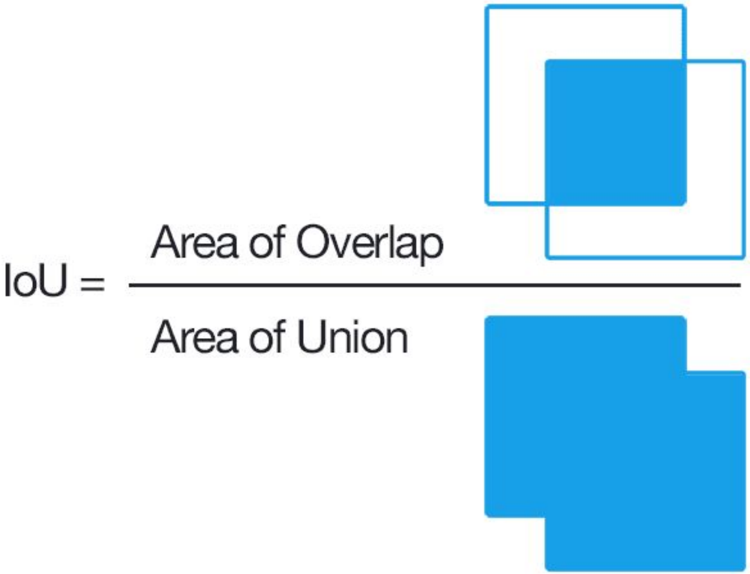
$$\boxed{\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2}$$

(4)

RCNN: Test time

- Collect proposals
- Extract CNN features after resizing them
- Run the classification and BB regression predictions
    - Use scores to select a subset from the proposals (e.g. top-k proposals per image)

Metric: Intersection over Union (IoU)



Source: pyimagesearch.com

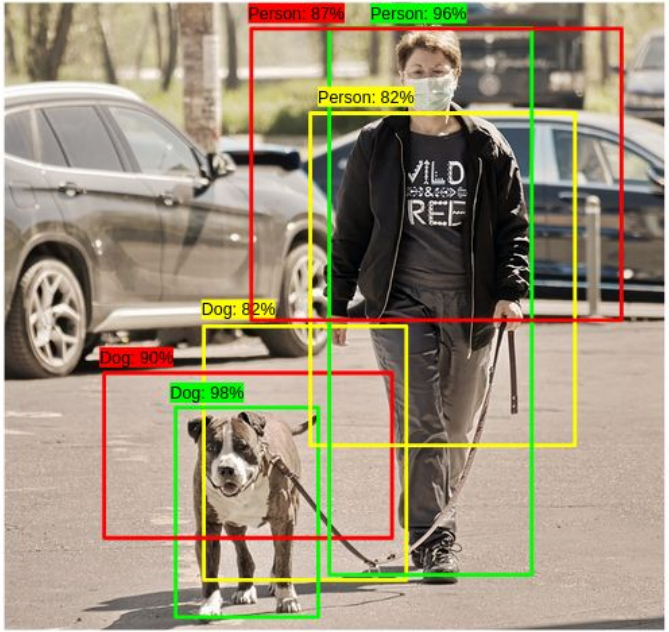
Metric: Intersection over Union (IoU)

Source: pyimagesearch.com

Duplicate predictions

https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

- Multiple overlapping boxes that are duplicate



Input: A list of Proposal boxes B, corresponding confidence scores S and overlap threshold N.

Output: A list of filtered proposals D.

Algorithm:

Select the proposal with highest confidence score, remove it from B and add it to the final proposal list D. (Initially D is empty).
Now compare this proposal with all the proposals — calculate the IOU (Intersection over Union) of this proposal with every other proposal. If the IOU is greater than the threshold N, remove that proposal from B.
Again take the proposal with the highest confidence from the remaining proposals in B and remove it from B and add it to D.
Once again calculate the IOU of this proposal with all the proposals in B and eliminate the boxes which have high IOU than threshold.
This process is repeated until there are no more proposals left in B.

Duplicate predictions

Post-processing: Non-Maximal Suppression (NMS)

1. Consider the next highest scoring BB
2. Remove all the lower-scoring BBs that have >0.7 IoI
3. Repeat

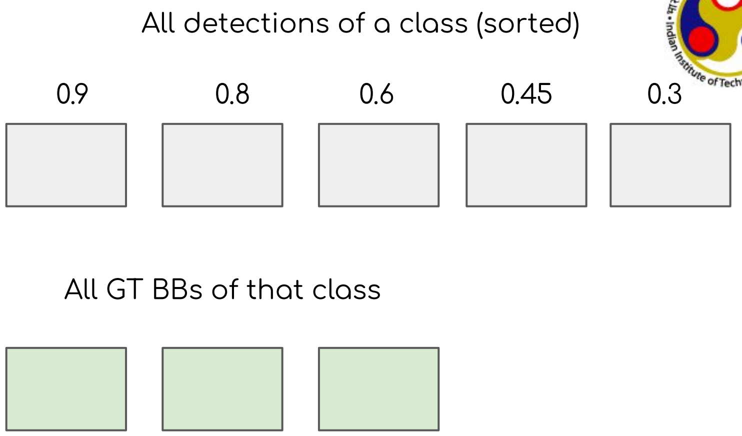
NMS may remove 'required' BBs in case of overlapping objects in the image

Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve

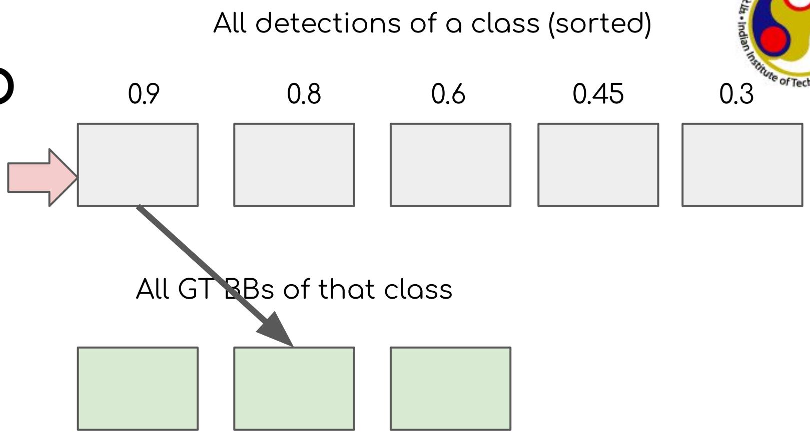
Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve



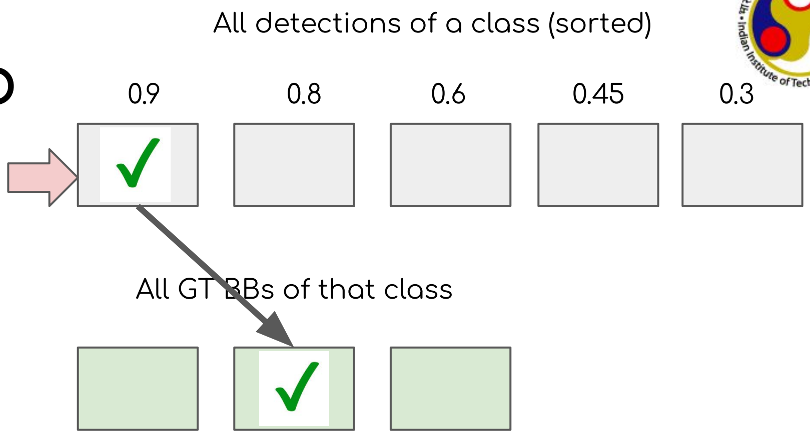
Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

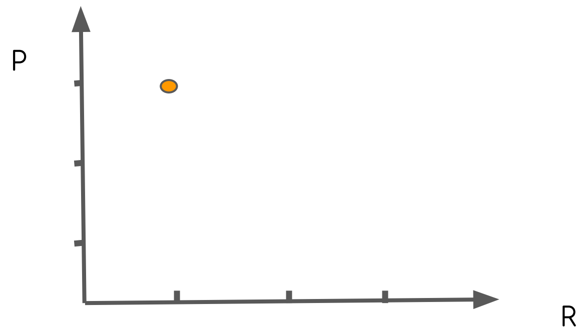


Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

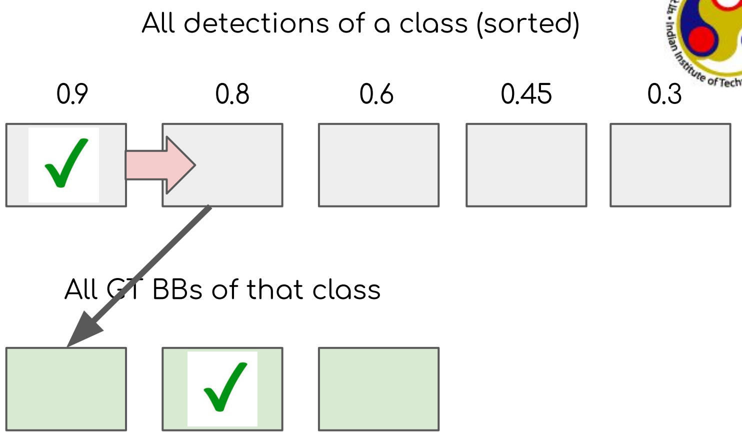


P = 1/1
R = ⅓ → (⅓, 1)



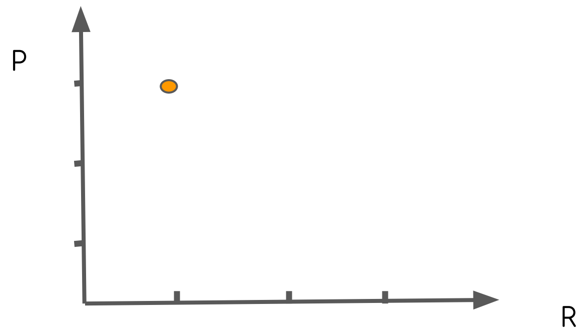
Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve



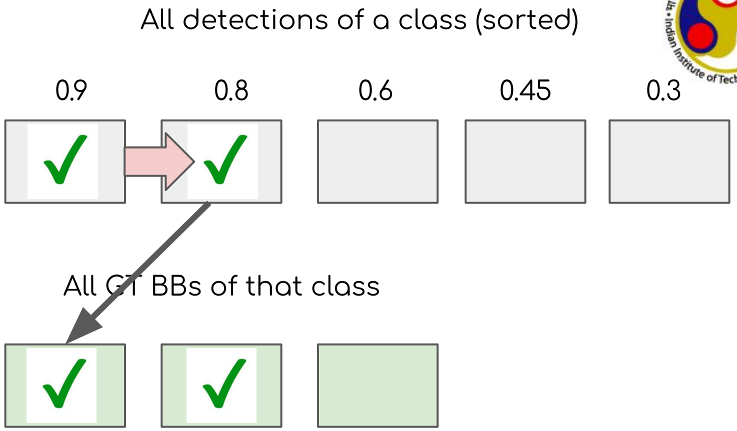
P = 1/1

R = ⅓ → (⅓, 1)

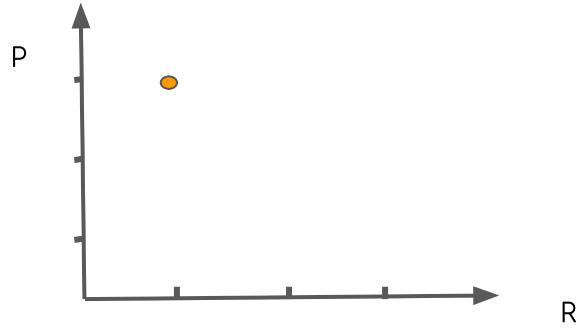


Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve

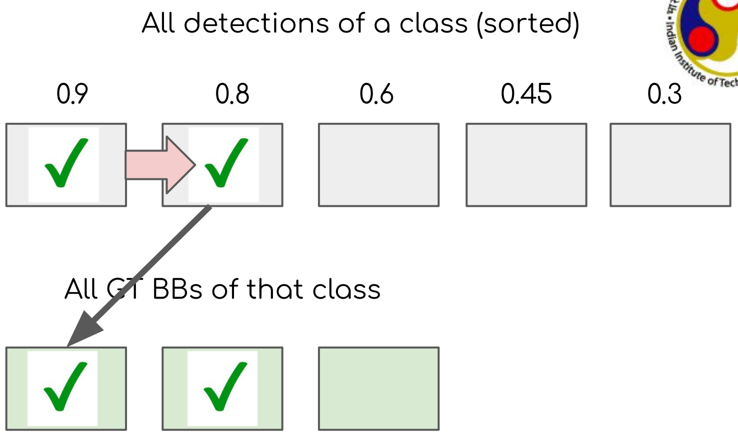


P = 1/1
R = ⅓ → (⅓, 1)

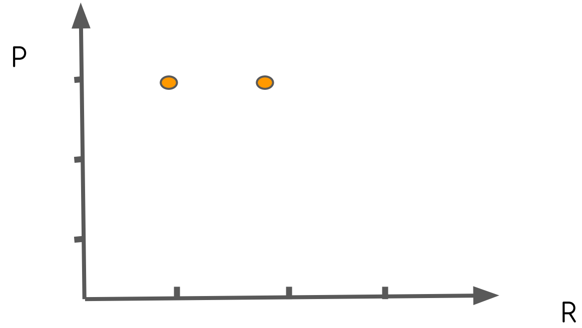


Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve



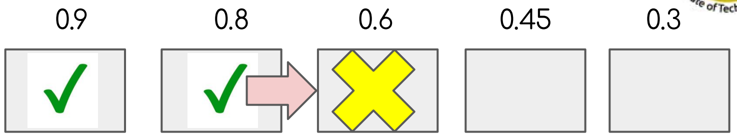
P = 2/2 = 1
R = ⅔ → (⅔, 1)



Performance metric: mAP

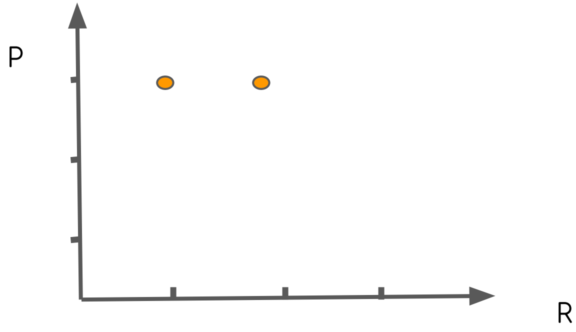
1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

All detections of a class (sorted)



All GT BBs of that class

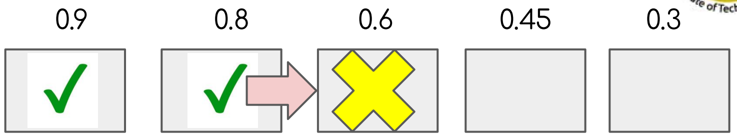
$P = 2/2 = 1$
$R = 2/3 \rightarrow (2/3, 1)$



Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

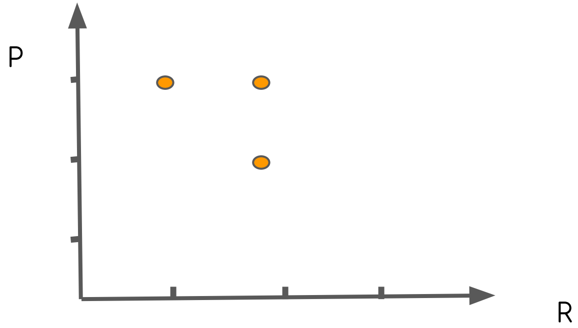
All detections of a class (sorted)



All GT BBs of that class

$P = 2/3 = \frac{2}{3}$

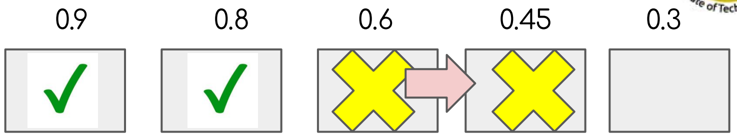
$R = \frac{2}{3} \rightarrow (\frac{2}{3}, \frac{2}{3})$



Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

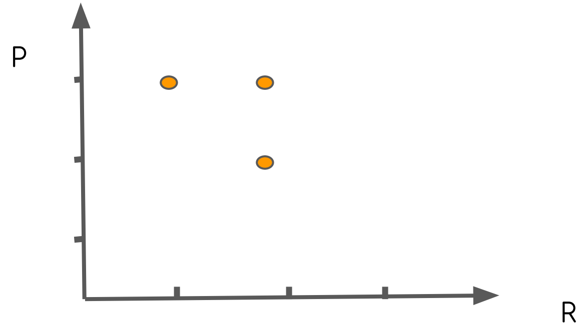
All detections of a class (sorted)



All GT BBs of that class

$P = 2/3 = \frac{2}{3}$

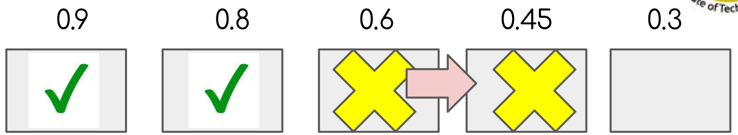
$R = \frac{2}{3} \rightarrow (\frac{2}{3}, \frac{2}{3})$



Performance metric: mAP

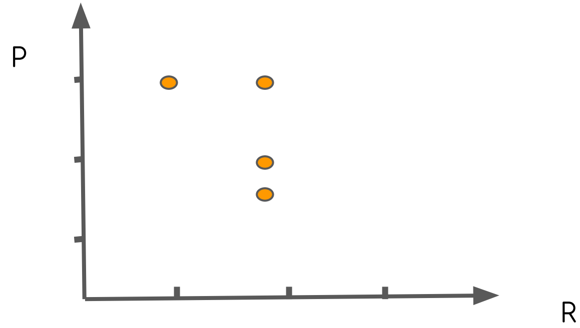
1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve

All detections of a class (sorted)



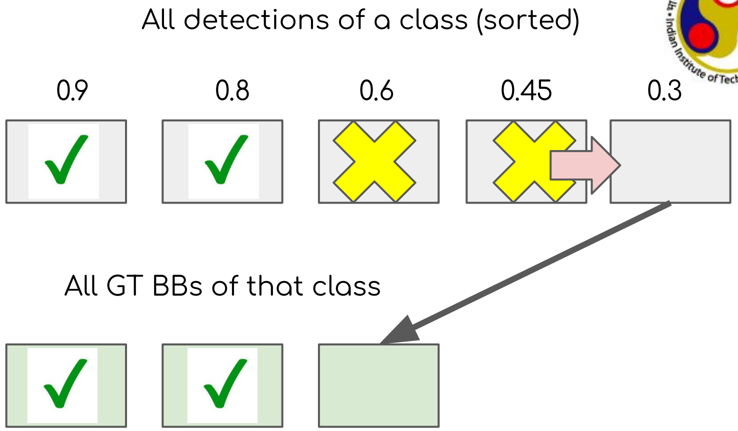
All GT BBs of that class

P = 2/4 = ½
R = ⅔ → (⅔, ½)

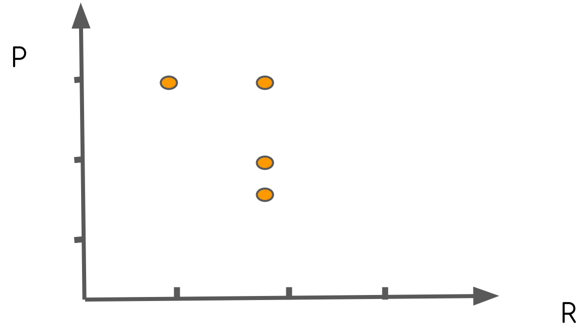


Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve

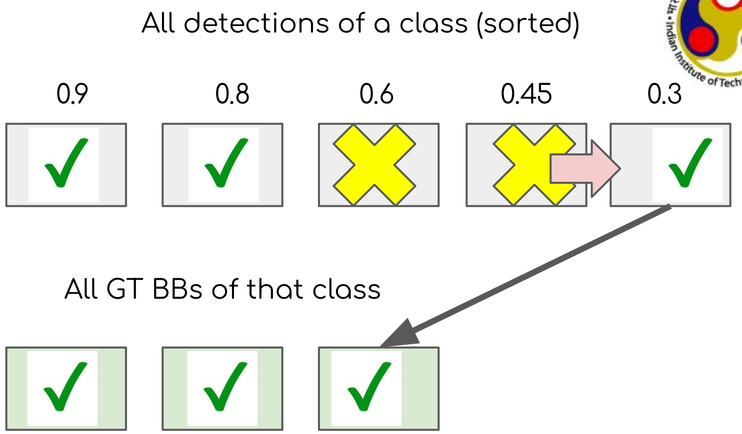


$P = 2/4 = \frac{1}{2}$
$R = \frac{2}{3} \rightarrow (\frac{2}{3}, \frac{1}{2})$



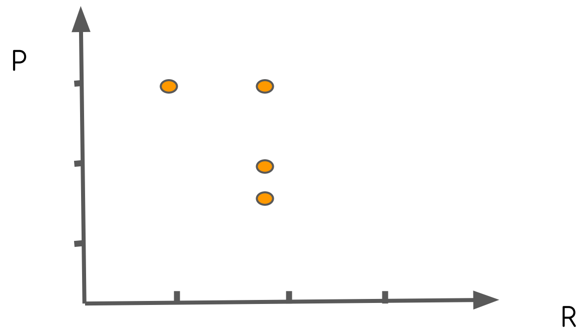
Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve



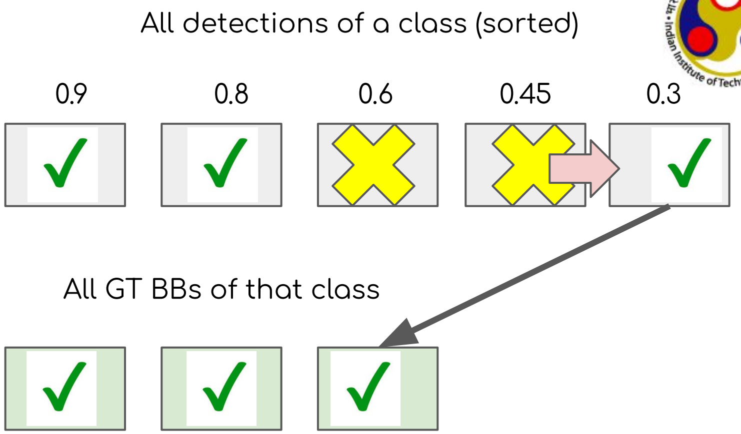
$P = 2/4 = \frac{1}{2}$

$R = \frac{2}{3} \rightarrow (\frac{2}{3}, \frac{1}{2})$



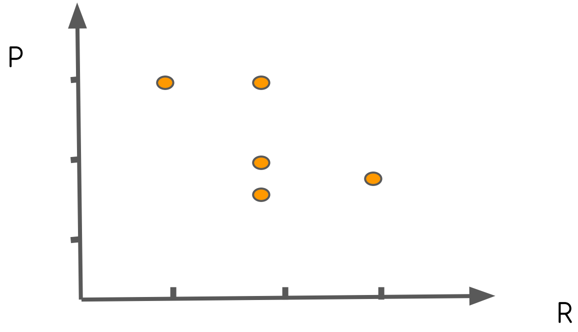
Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve



$P = 3/5$

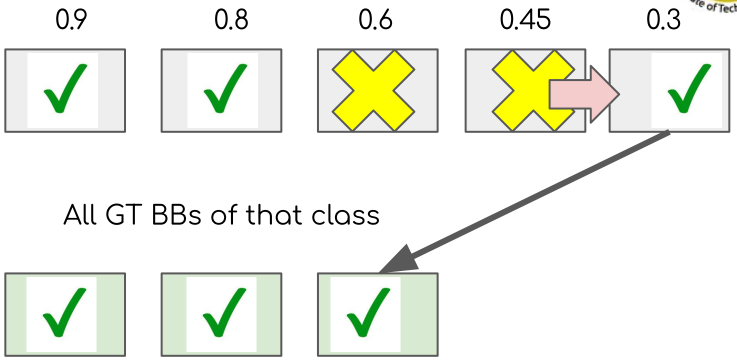
$R = 3/3 \rightarrow (1, 3/5)$



Performance metric: mAP

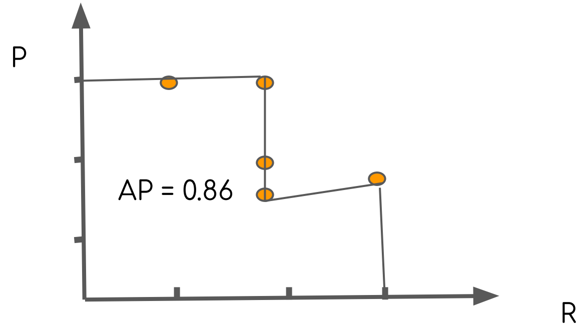
1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

All detections of a class (sorted)



$P = 3/5$

$R = 3/3 \rightarrow (1, 3/5)$



Performance metric: mAP

1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
    a. For each predicted detection
        i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
        ii. Otherwise, False Positive (FP)
        iii. Plot the corresponding point on the PR curve
    b. AP = Area under the Precision and Recall curve

Mean average precision (**mAP@0.5**) = Average AP across all the object categories

Performance metric: mAP

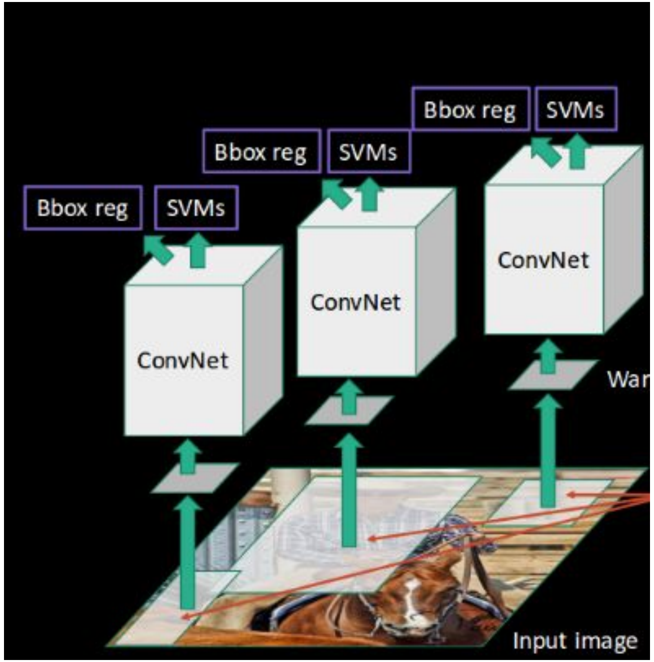
1. Run the detector + NMS
2. Sort the predicted detections in the decreasing order of confidence
3. For each category, compute the avg. precision (AP)
   a. For each predicted detection
      i. If it matches with a GT BB (with IoU>0.5) → True Positive (TP)
      ii. Otherwise, False Positive (FP)
      iii. Plot the corresponding point on the PR curve
   b. AP = Area under the Precision and Recall curve

mAP@0.5
mAP@0.55
mAP@0.6
....
......
mAP@0.9
mAP@0.95

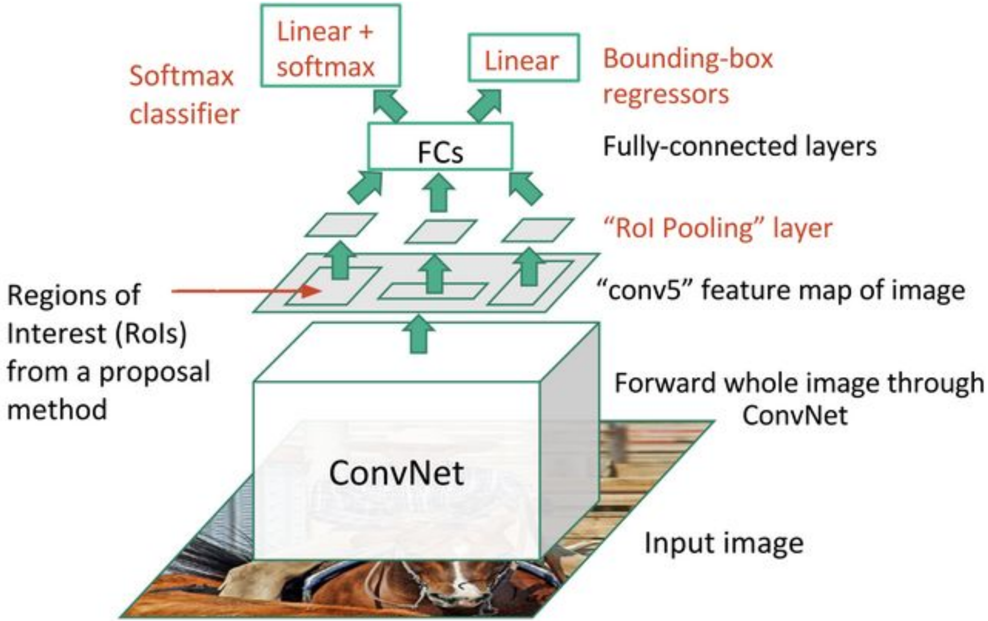
mAP

RCNN: drawbacks

- Very slow! (~2K proposals per image)
- → 2K forwardpasses of CNN
- Solution: Run CNN and then warp → Fast RCNN



Fast R-CNN

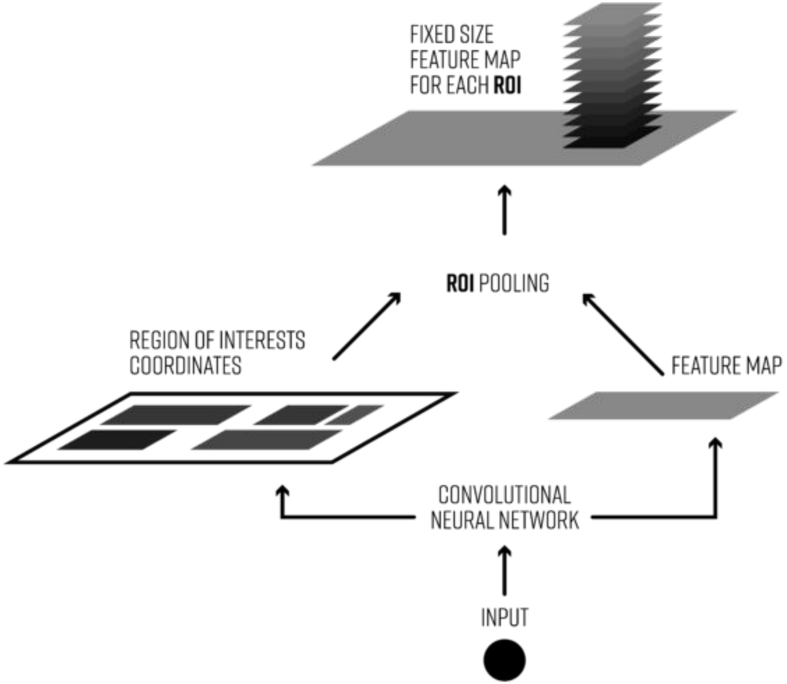


Linear +
softmax
Softmax
classifier
Linear
Bounding-box
regressors
FCs
Fully-connected layers
"RoI Pooling" layer
"conv5" feature map of image
Regions of
Interest (RoIs)
from a proposal
method
Forward whole image through
ConvNet
ConvNet
Input image

Region of Interest (RoI) pooling

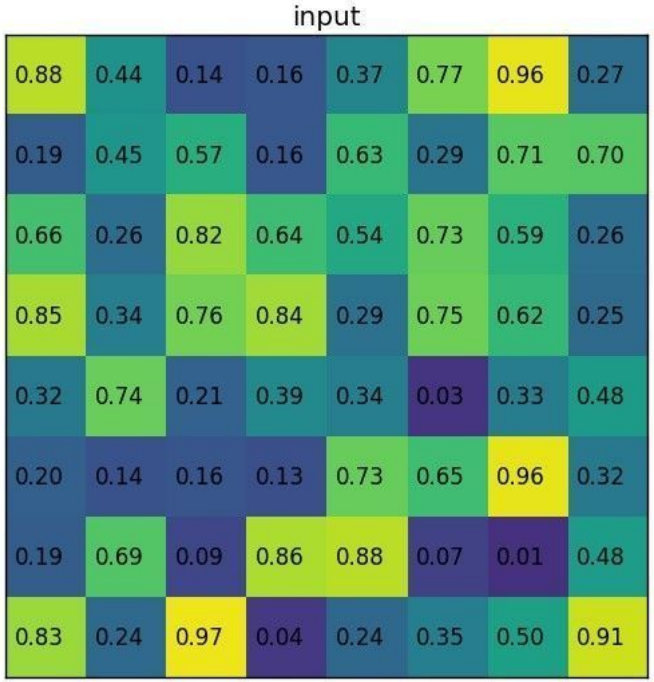
- Produces fixed-size feature maps from non-uniform input via max-pooling
- Per-region CNN (light weight) takes over

Region of Interest (RoI) pooling

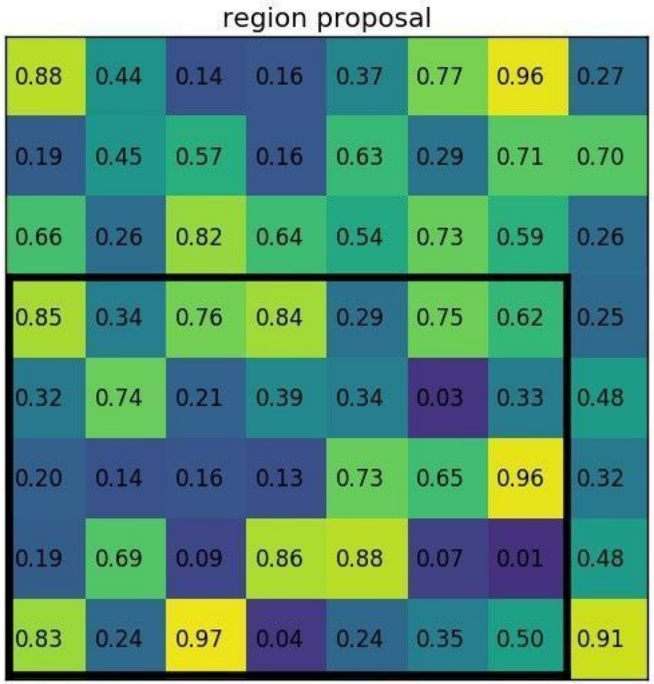


ROI pooling

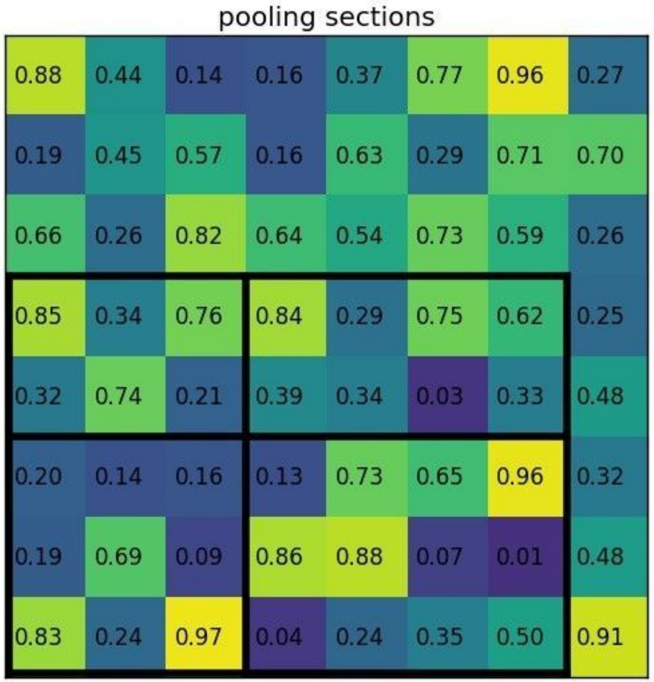
Pooled_width and pooled_height are hyperparameters which can be decided based on the problem at hand. These indicate the number of grids the feature map corresponding to the proposal should be divided into



ROI pooling


region proposal
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91

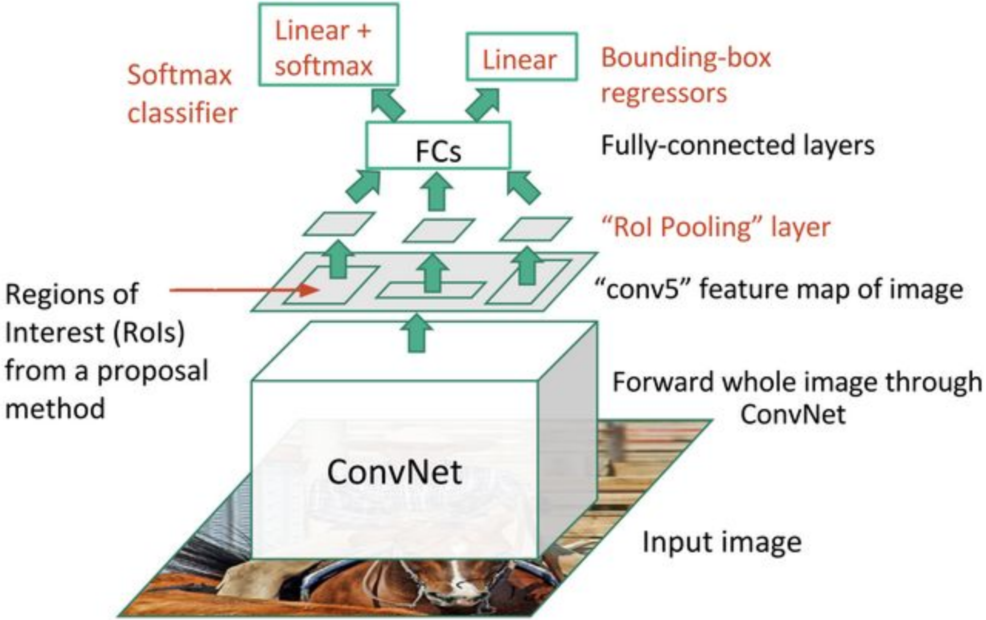
ROI pooling


pooling sections
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91

ROI pooling

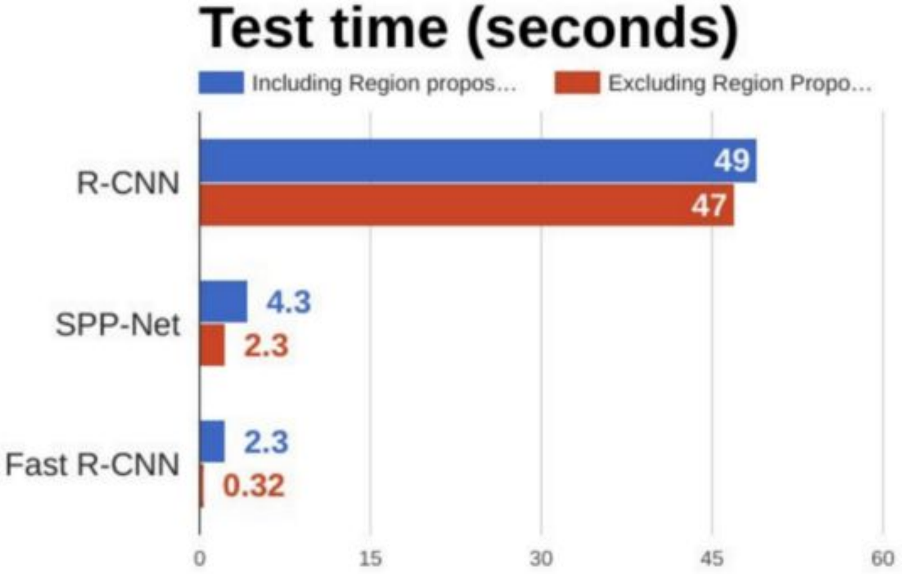
0.85	0.84
0.97	0.96

Fast R-CNN



https://towardsdatascience.com/region-of-interest-pooling-f7c637f409af

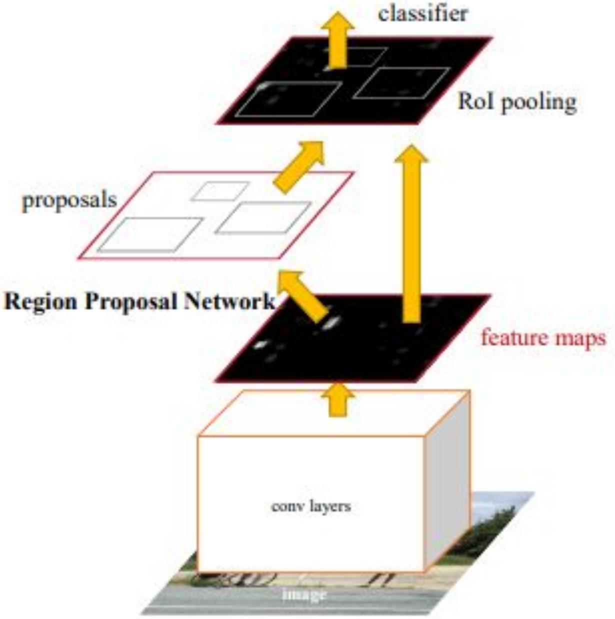
Slow vs fast R-CNN



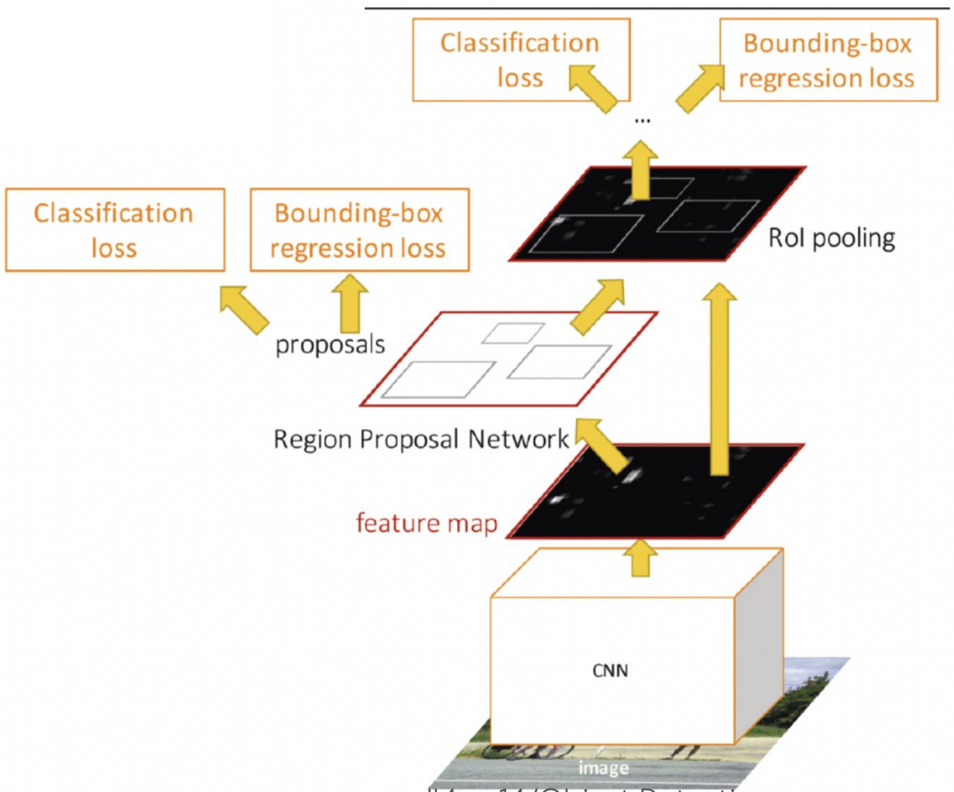
Fast R-CNN

- Most of the time is consumed by selective search
- → get the proposals also from the CNN backbone
- → Insert a region proposal network (RPN) → Faster R-CNN

Faster R-CNN

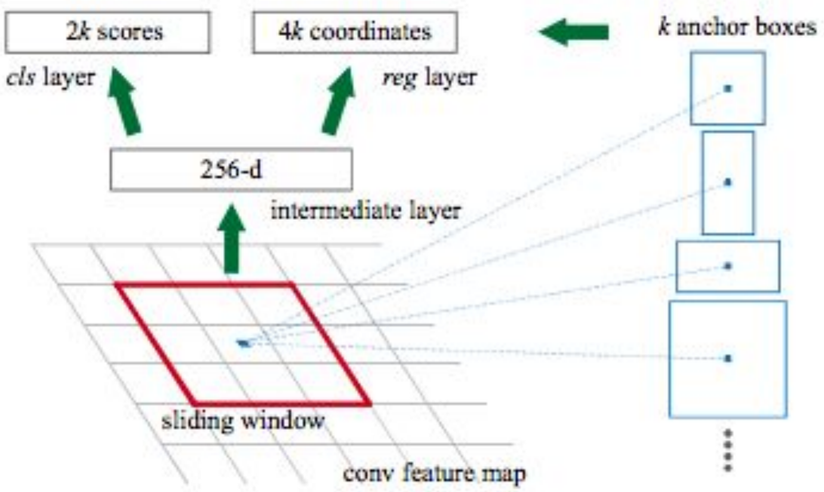

classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image

Faster R-CNN



Classification loss
Bounding-box regression loss
...
RoI pooling
Classification loss
Bounding-box regression loss
proposals
Region Proposal Network
feature map
CNN
image

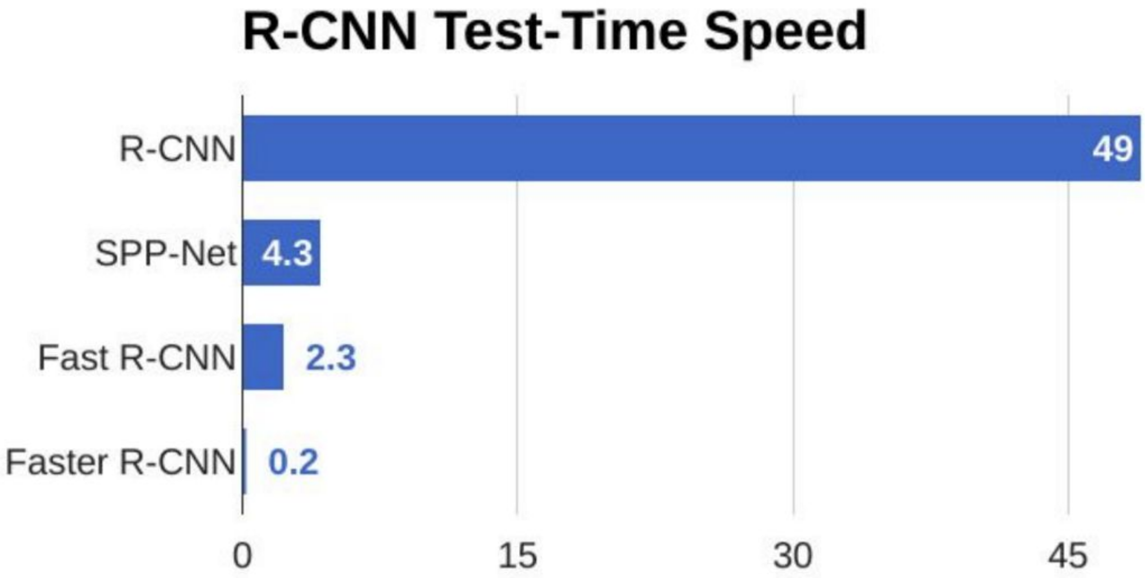
RPN


2k scores
4k coordinates
k anchor boxes
cls layer
reg layer
256-d
intermediate layer
sliding window
conv feature map

Faster R-CNN (trains with 4 losses)

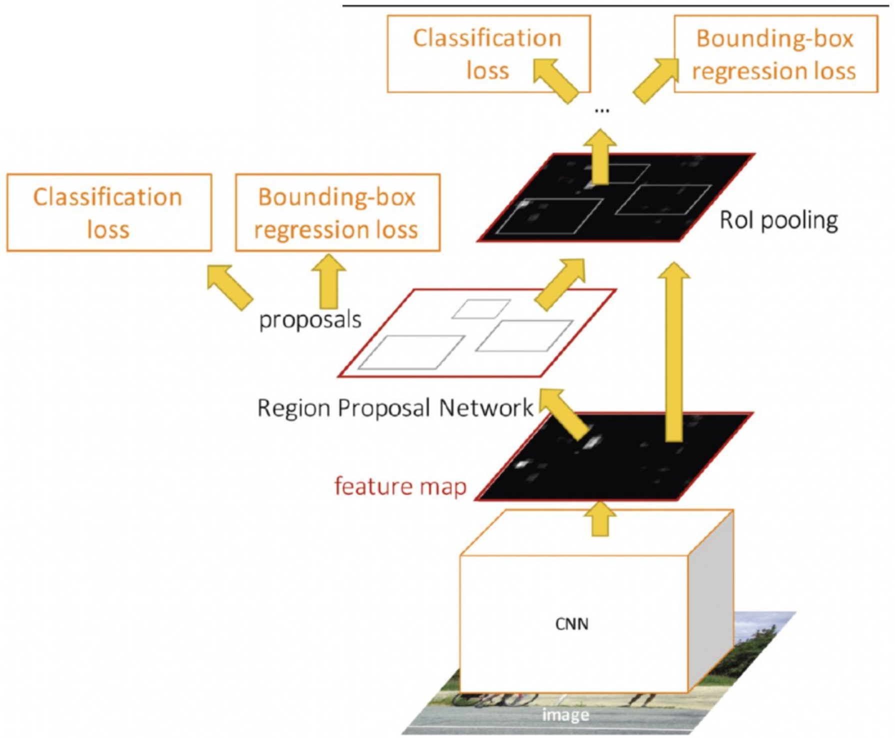
1. RPN classification: anchor box is object vs BG
2. RPN regression: predict the transformation to proposal box from anchor box
3. Object classification: classify the proposal as BG vs object class
4. Object Regression: predict the transformation from proposal box to the object box

Faster R-CNN


R-CNN Test-Time Speed
R-CNN
SPP-Net
Fast R-CNN
Faster R-CNN
49
4.3
2.3
0.2
0
15
30
45

Faster R-CNN

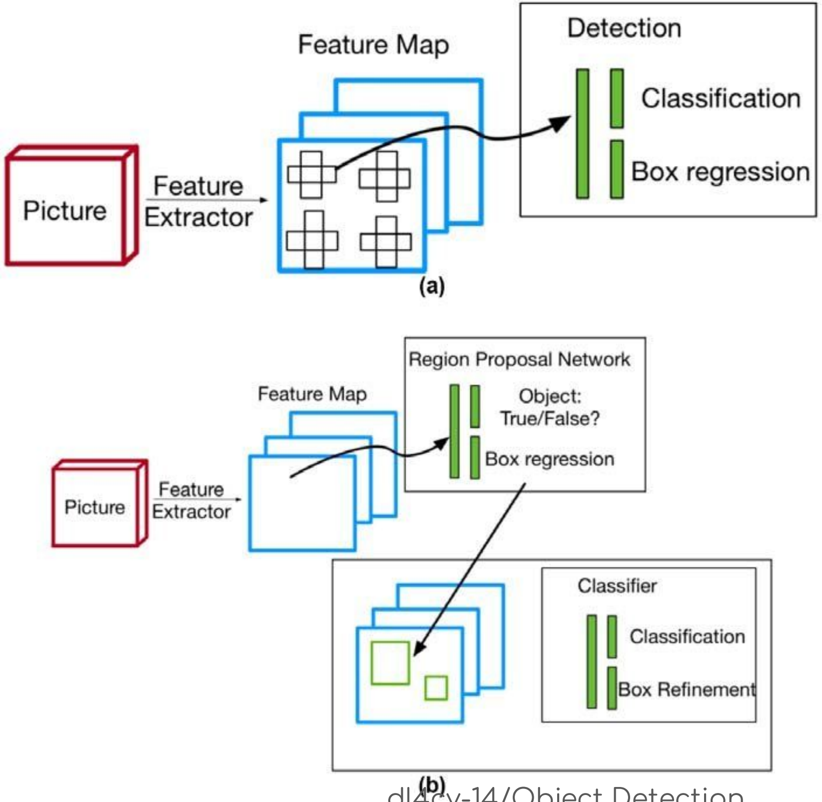
- Two stage approach
  - 1: once; backbone and RPN
  - 2: per proposal; crop features, predict class and coordinates



Single Stage Detectors (SSD)

- Do we need two stages? → Single Stage Detectors (SSD)
- RPN does all the job
  - Proposals
  - Classification (C+1 way)

Single Stage Detectors (SSD)



Object detection: Impact of Deep learning

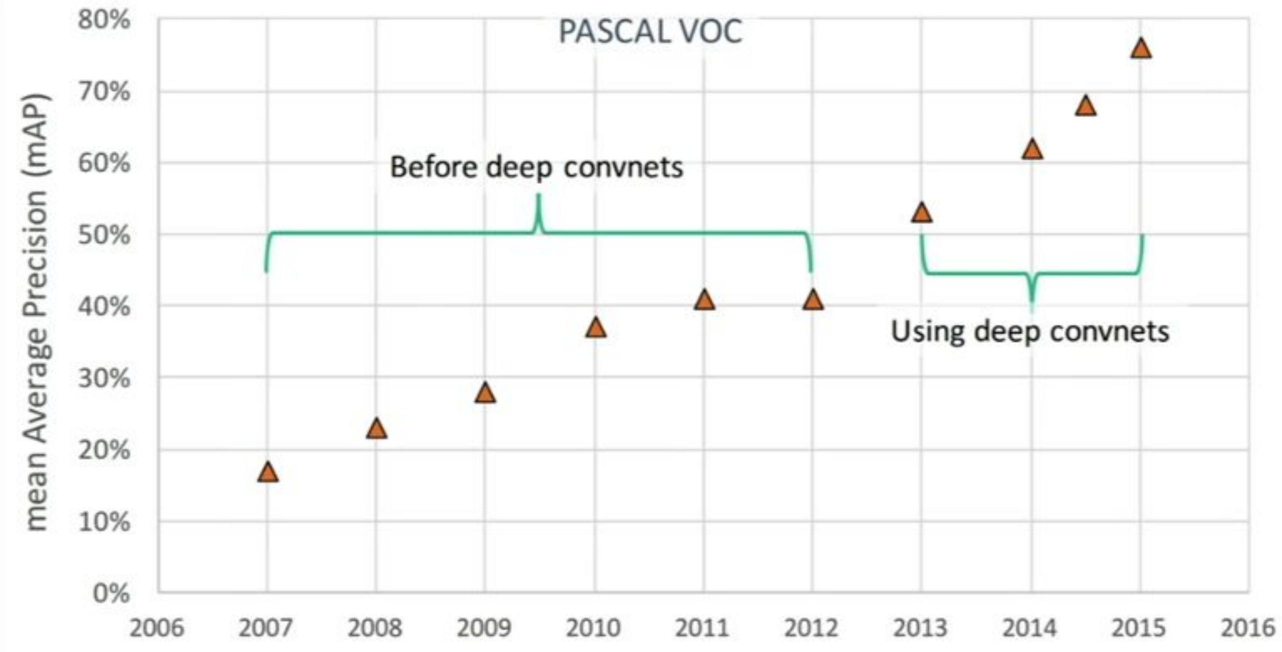


Figure Credits: Ross Girshick

Detection without the anchor boxes

- CornerNet (Law et al. ECCV 2018) poses the problem as predicting the possibility of each pixel to be a top left and bottom right corner for each category

Detection without the anchor boxes

- $C \times H \times W$ heatmap for upper left corners
- $C \times H \times W$ heatmap for bottom right corners
- 2 times $D \times H \times W$ Embeddings prediction for matching the corners

Detection without the anchor boxes

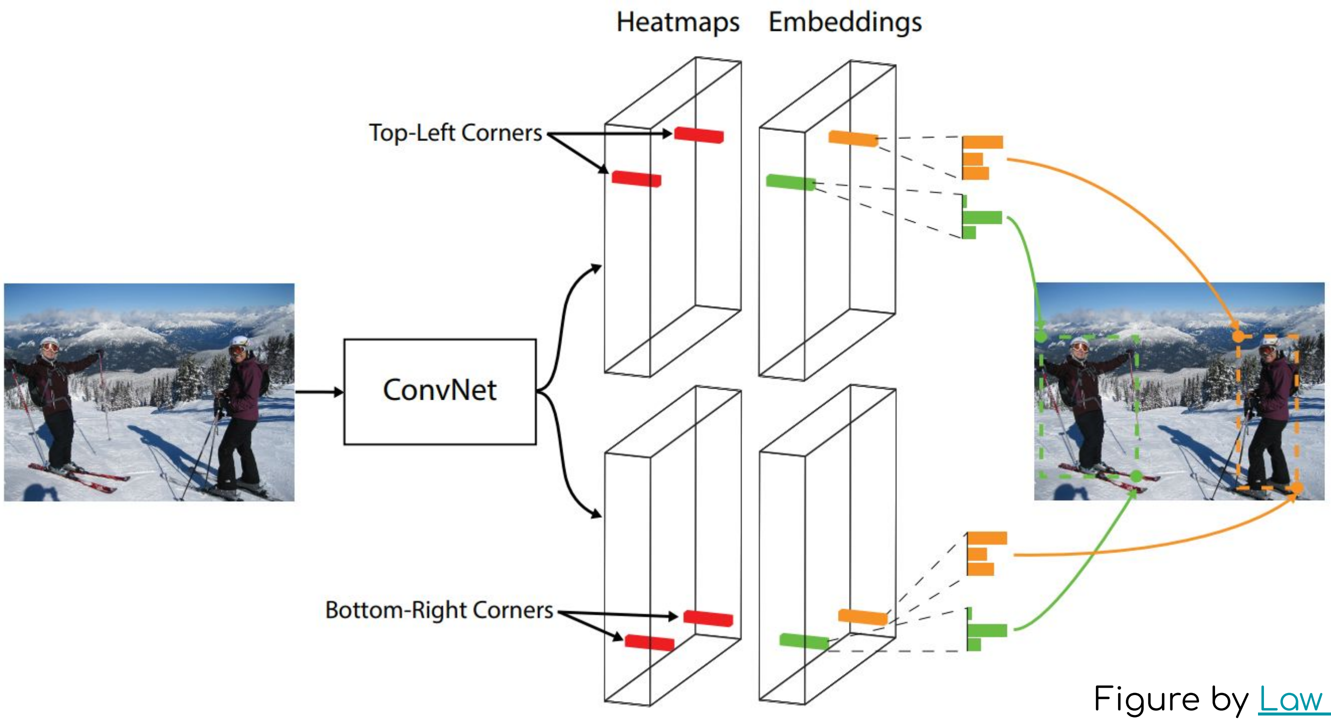


Figure by Law et al. ECCV 2018

Detection without the anchor boxes

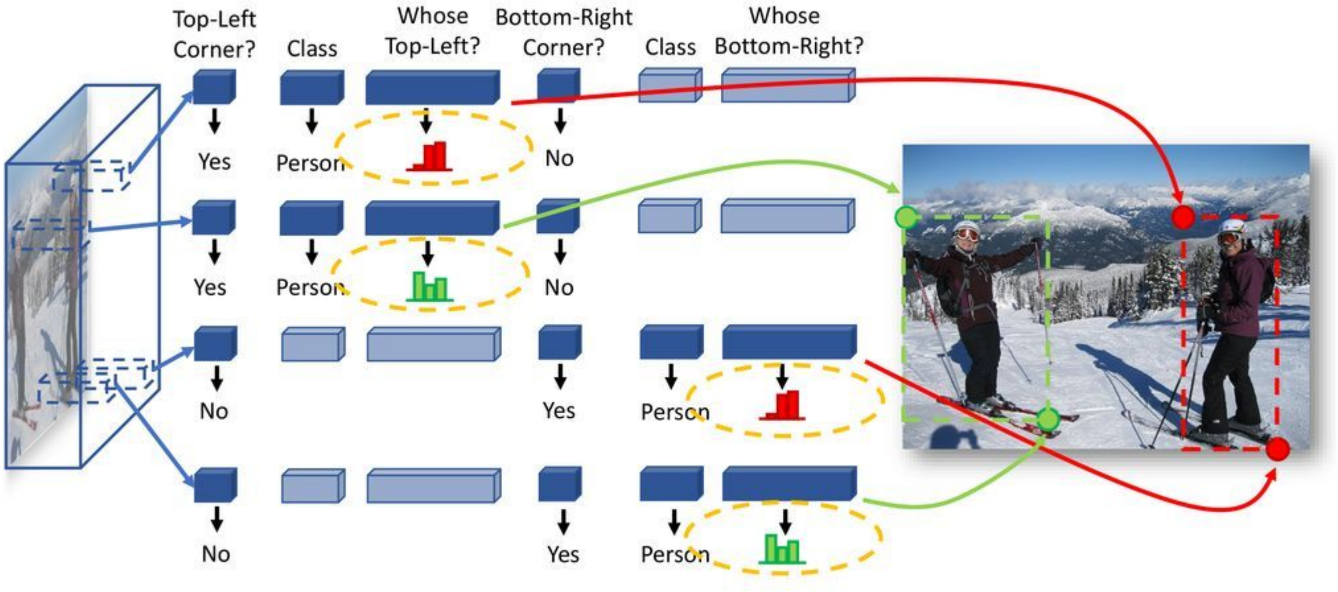
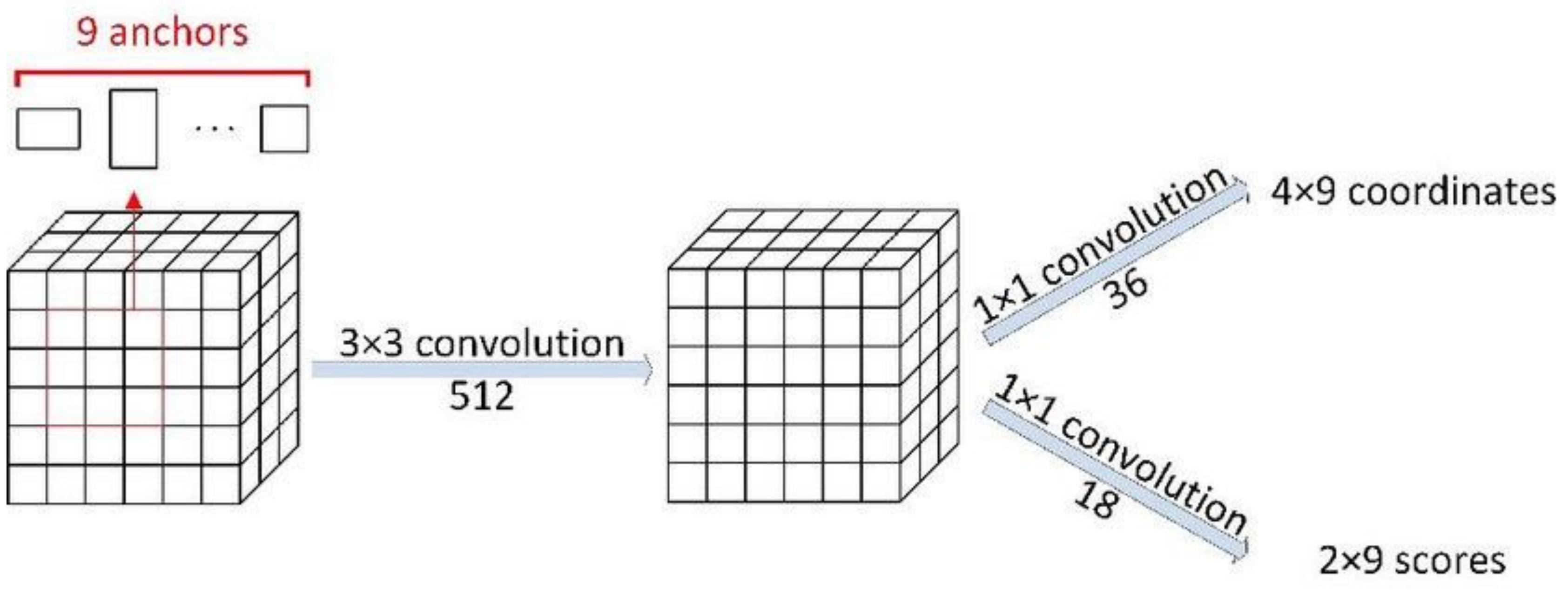


Figure by Law et al. ECCV 2018

Appendix

RPN



9 anchors
3×3 convolution
512
1×1 convolution
36
4×9 coordinates
1×1 convolution
18
2×9 scores