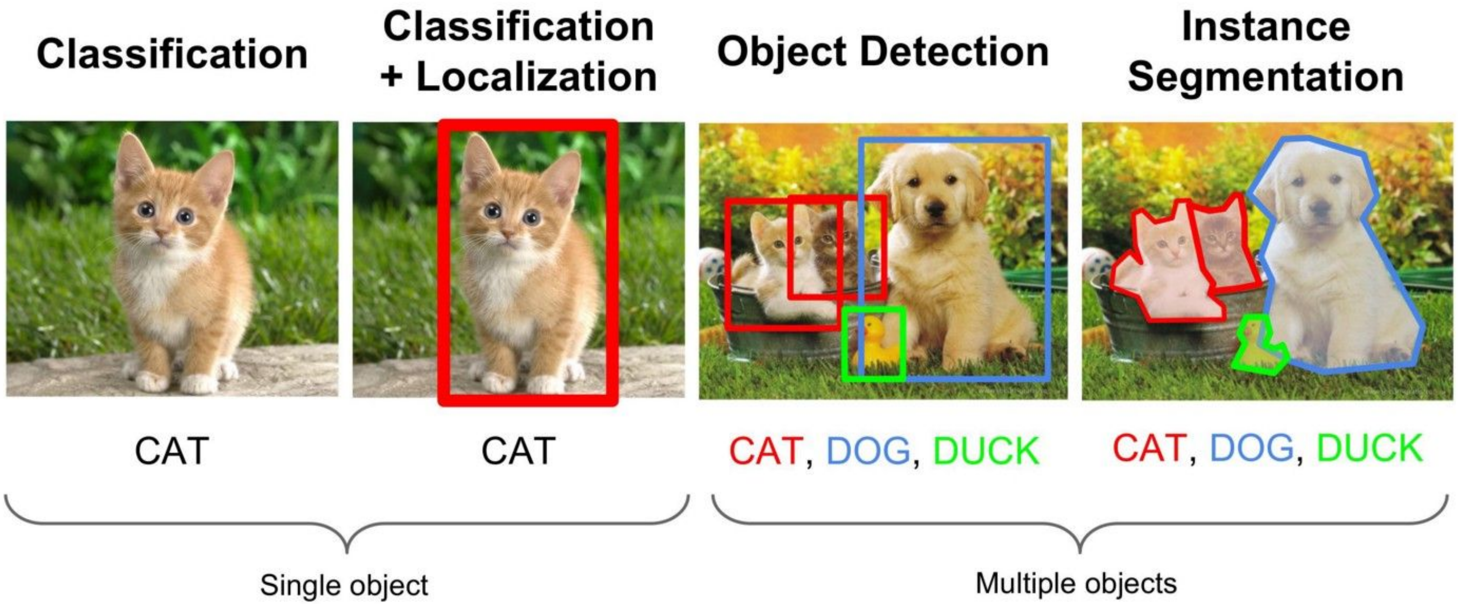
Deep Learning

Seminar 6

What are computer vision tasks?



Semantic segmentaion:

- Faster / easier than instance segm
- Allows "complete" explanation
- Merges instances
- Suitable for "stuff" and things

Next lecture

Object detection:

- Faster / easier than instance segm
- Distinguishes instances
- Inaccurate for some classes
- Incomplete
- Suitable for things

This time

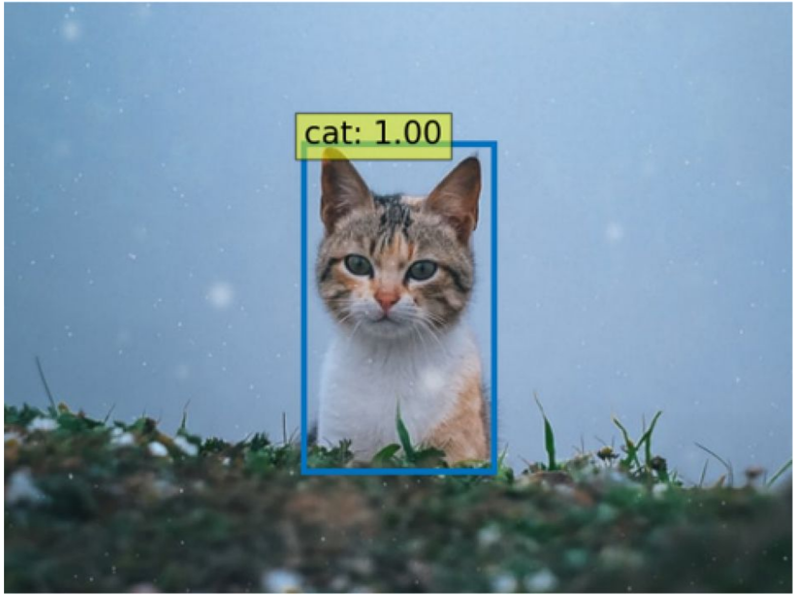
Instance / Panoptic segmentation:

- Complete
- Distinguish instances
- Accurate
- Harder/slower

Next lecture

Object detection

Object detection



As data we have:

{

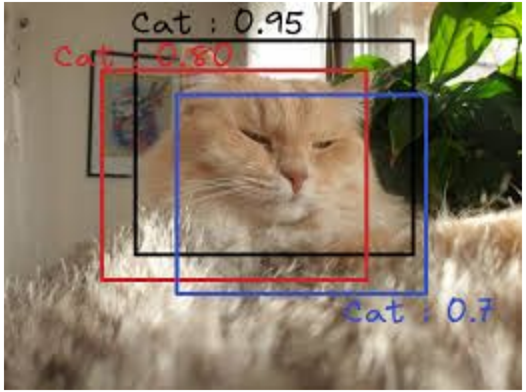
class: id
left up x: int
left up y: int
right bottom x: int
right bottom y: int

}

Our task is to predict both class and bounding box for image

Metrics

How to measure performance?



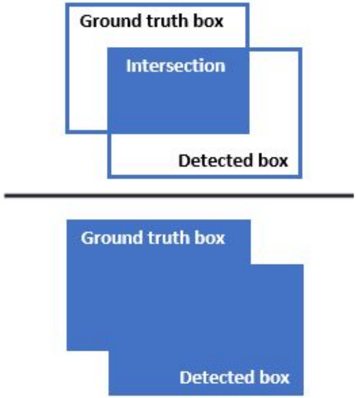
- IoU (Intersection over Union)
- mAP (mean average precision)

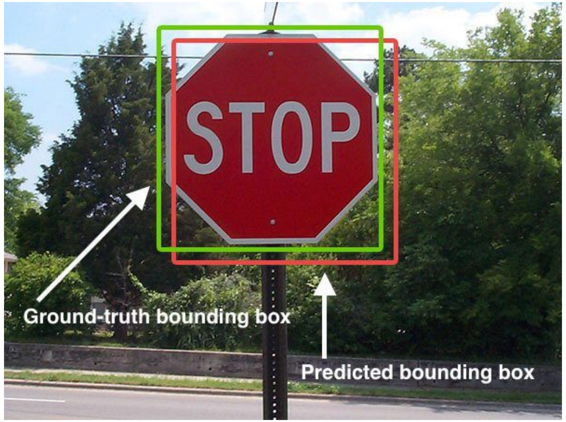
IoU (Intersection over Union)

Motivation:

- How does bounding boxes located? Good or bad?

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} =$$

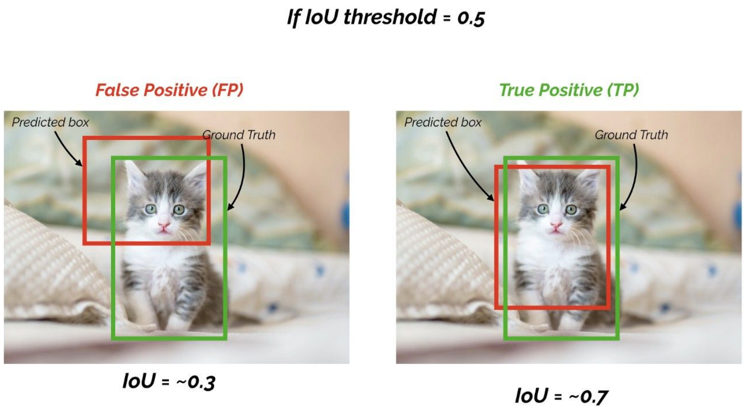


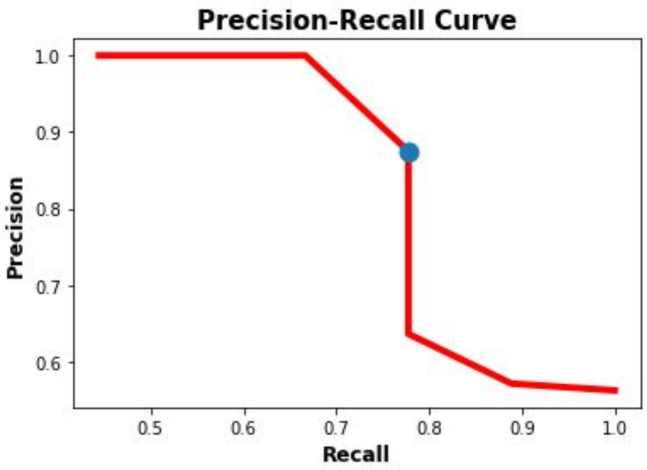


mAP (mean averaged precision)

Motivation:

- Different IoU threshold gives us different predictions

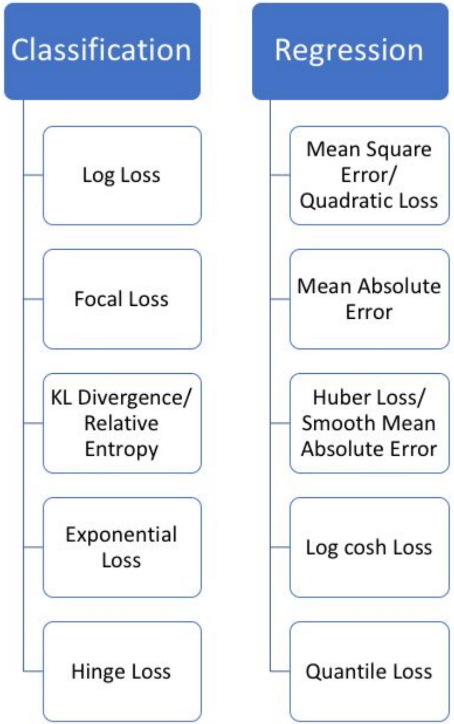




Loss function

What is the loss function for this task?

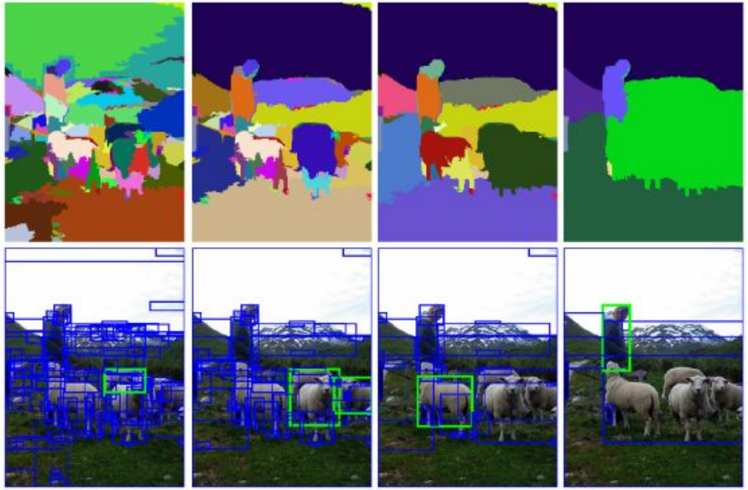
$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{reg}$$

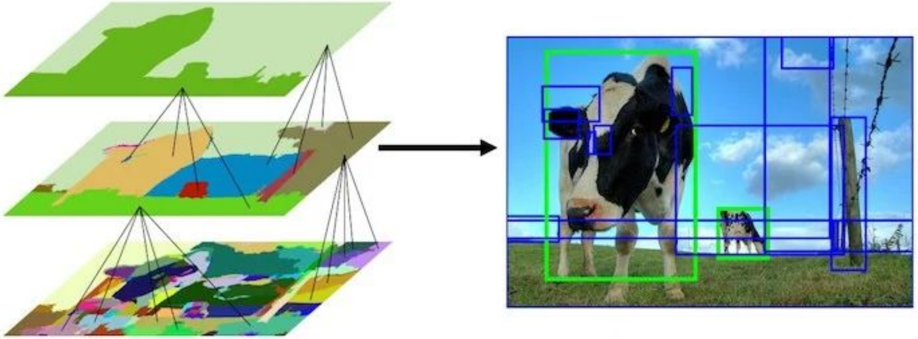


Object detection

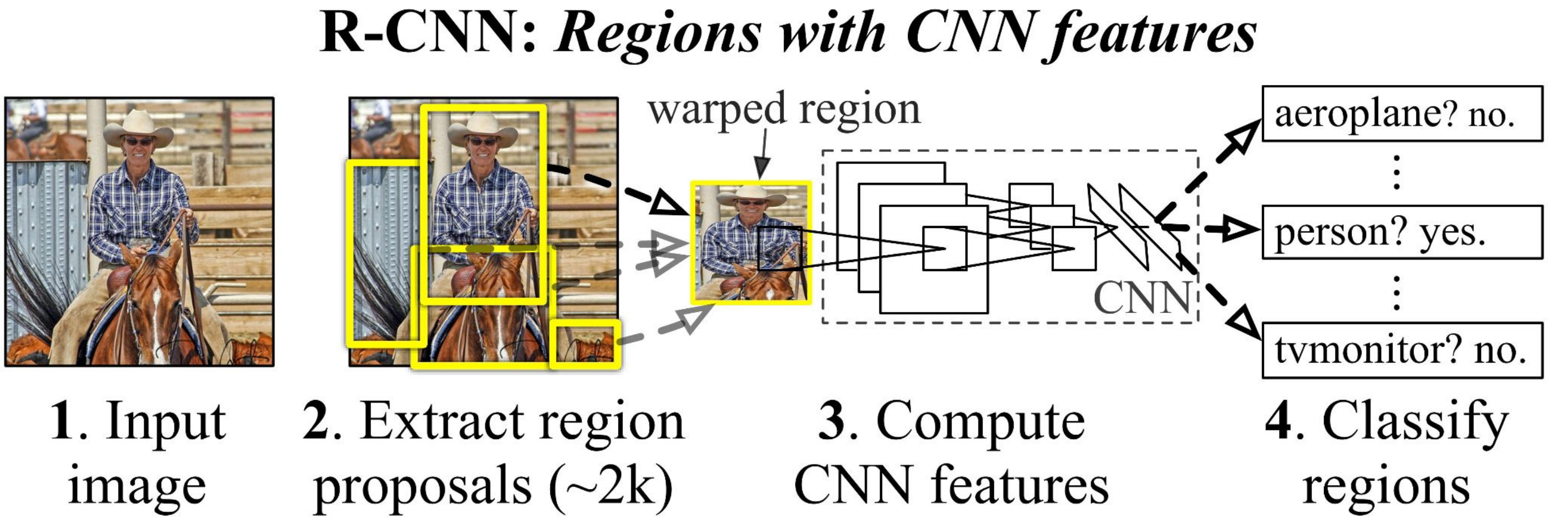
How do you approach the object detection task?

Standard method of boxes generation





R-CNN


R-CNN: *Regions with CNN features*
warped region
CNN
aeroplane? no.
person? yes.
tvmonitor? no.
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Non-maximum suppression (NMS)

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)

    $B_{nms} \leftarrow \emptyset$ Initialize empty set

    for $b_i \in B$ **do** => Iterate over all the boxes

        $discard \leftarrow \text{False}$ Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded

        for $b_j \in B$ **do** Start another loop to compare with b(i)

            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then** If both boxes having same IOU

                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**

                    $discard \leftarrow \text{True}$ Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.

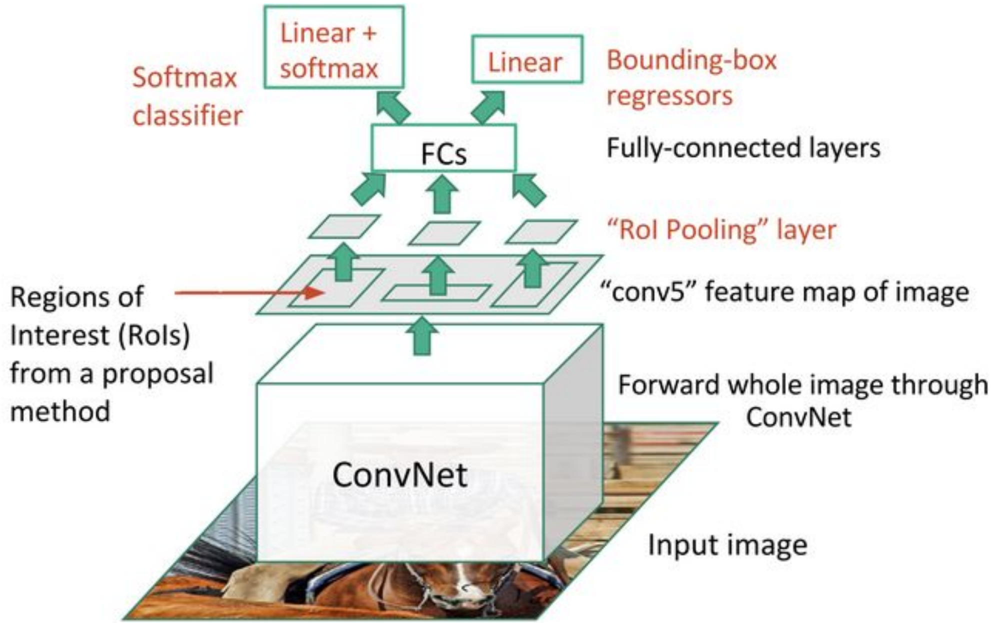
        if not $discard$ **then** Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.

            $B_{nms} \leftarrow B_{nms} \cup b_i$

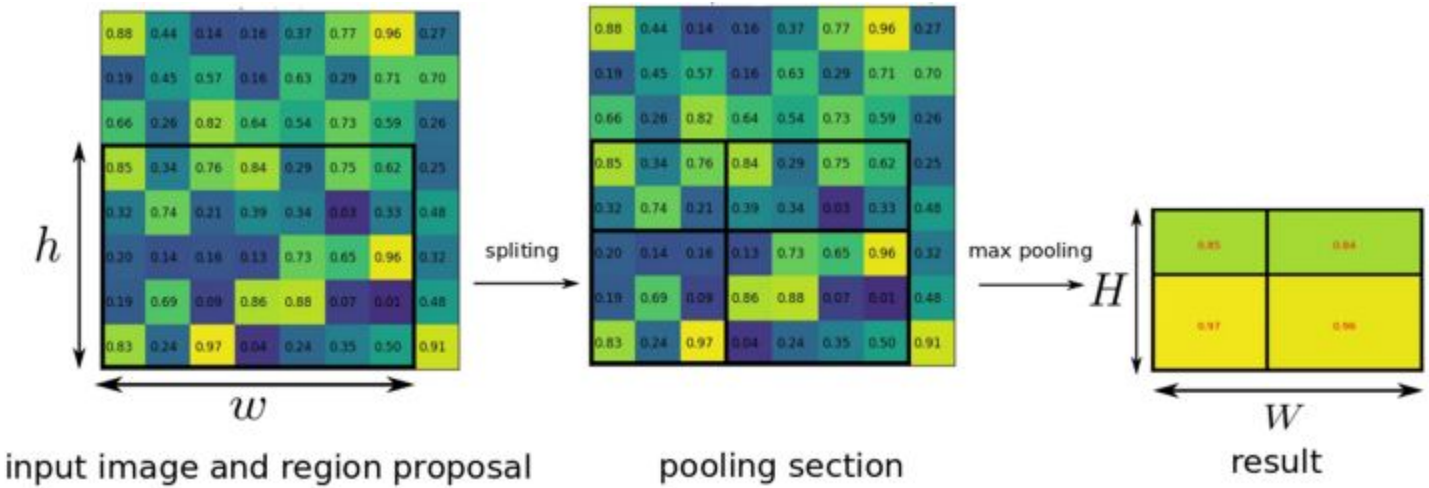
    return B_{nms} Do the same procedure for remaining boxes and return the final list

Non-maximum suppression (NMS)

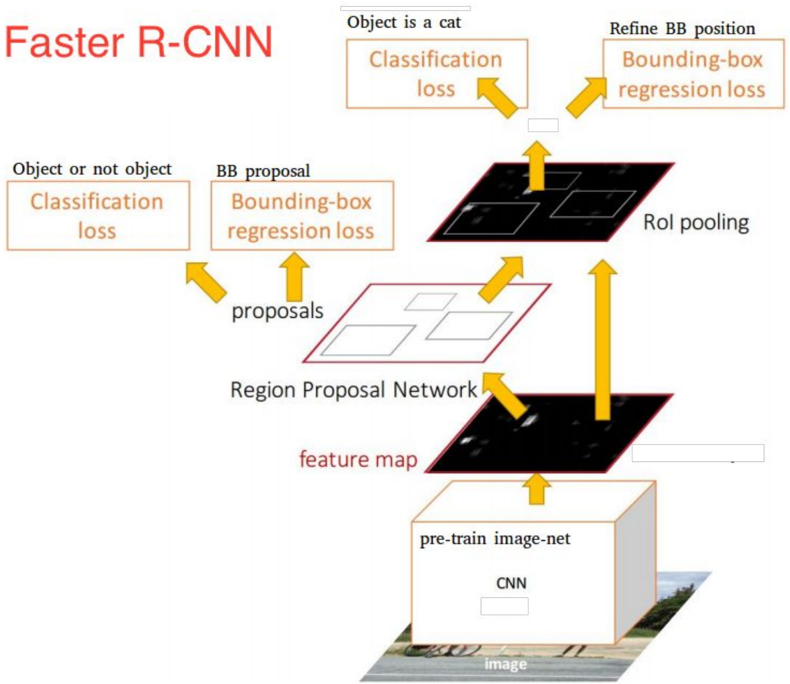
Fast R-CNN


Linear +
softmax
Linear
Softmax
classifier
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

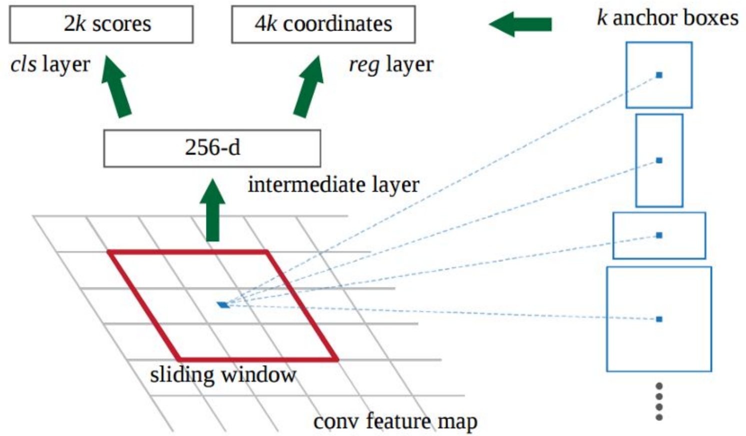
RoI pooling


h
w
spliting
max pooling
H
W
input image and region proposal
pooling section
result

Faster R-CNN

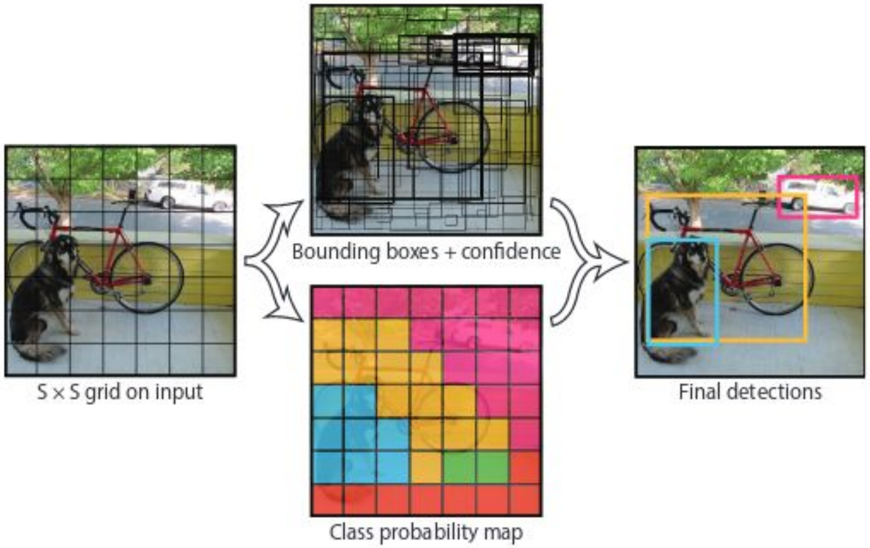


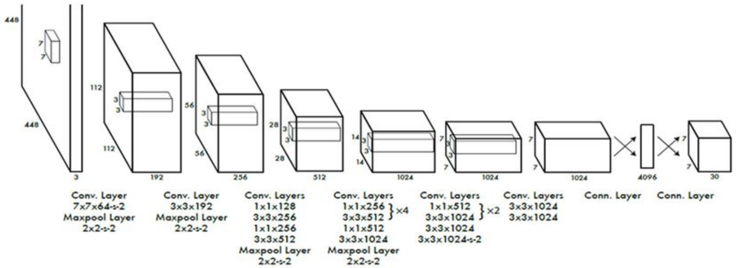
Faster R-CNN
Object is a cat
Classification loss
Refine BB position
Bounding-box regression loss
Object or not object
Classification loss
BB proposal
Bounding-box regression loss
RoI pooling
proposals
Region Proposal Network
feature map
pre-train image-net
CNN
image



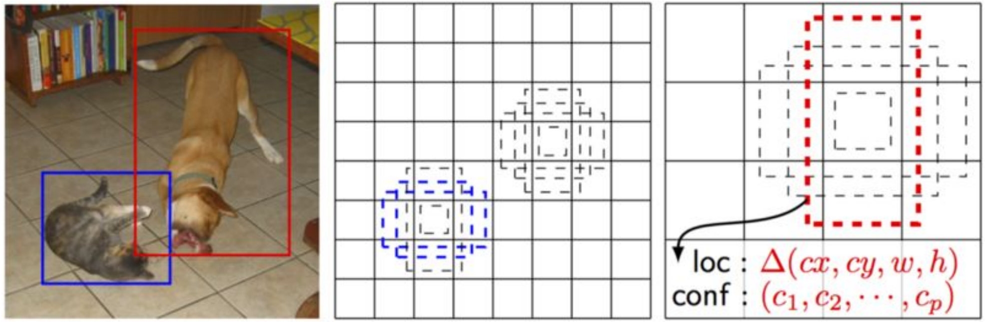
2k scores
4k coordinates
k anchor boxes
cls layer
reg layer
256-d
intermediate layer
sliding window
conv feature map

YOLO

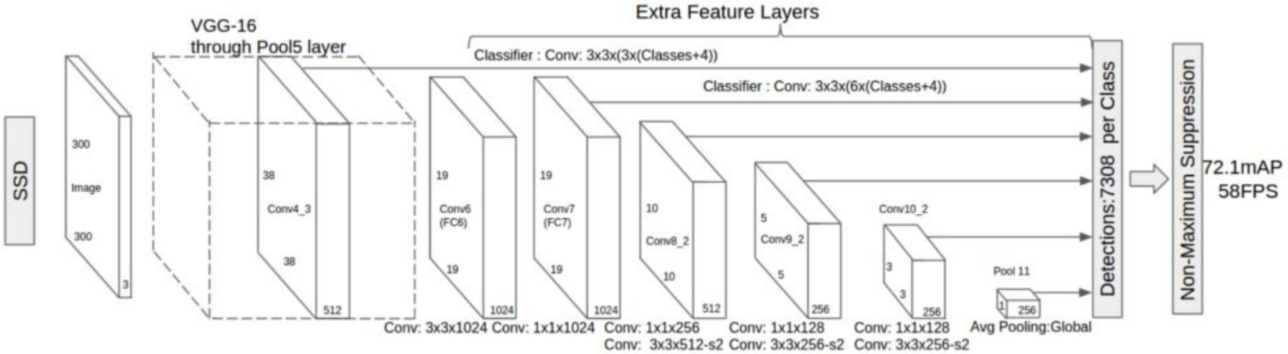

S × S grid on input
Bounding boxes + confidence
Class probability map
Final detections


Conv. Layer
7x7x64-s-2
Maxpool Layer
2x2-s-2
Conv. Layer
3x3x192
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x128
3x3x256
1x1x256
3x3x512
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x256
3x3x512 ×4
1x1x512
3x3x1024
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x512
3x3x1024 ×2
3x3x1024
3x3x1024-s-2
Conv. Layers
3x3x1024
3x3x1024
Conn. Layer
Conn. Layer

SSD



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



Recap

- Object detection
- RCNN
- Fast RCNN
- Faster RCNN
- YOLO
- SSD