SSD: single shot detector

Feng Wang

AIRD, Coretronic Co.

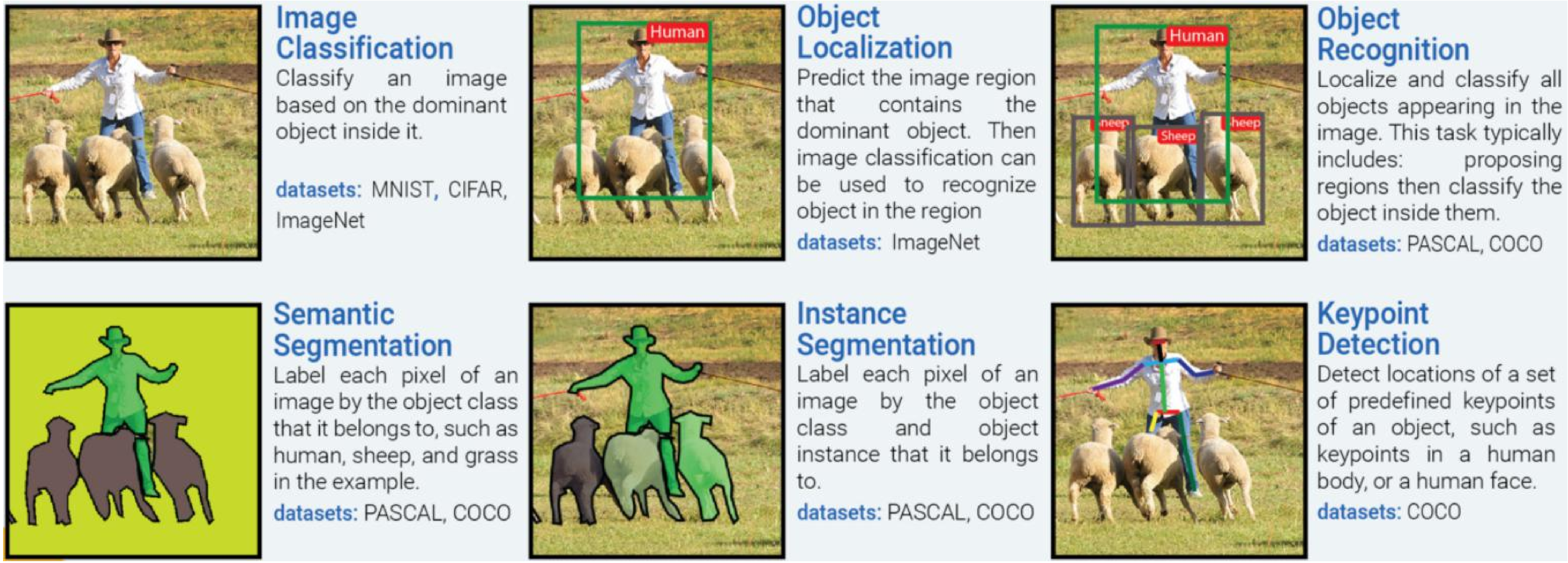
May 03, 2019

The slides and a list of references can be found from
https://github.com/fwcore/object-detection

Outlines

- **Review a few key concepts in object detection**
- **SSD (arXiv: 1512.02325)**
  - design
  - loss function
  - training

Common tasks on images

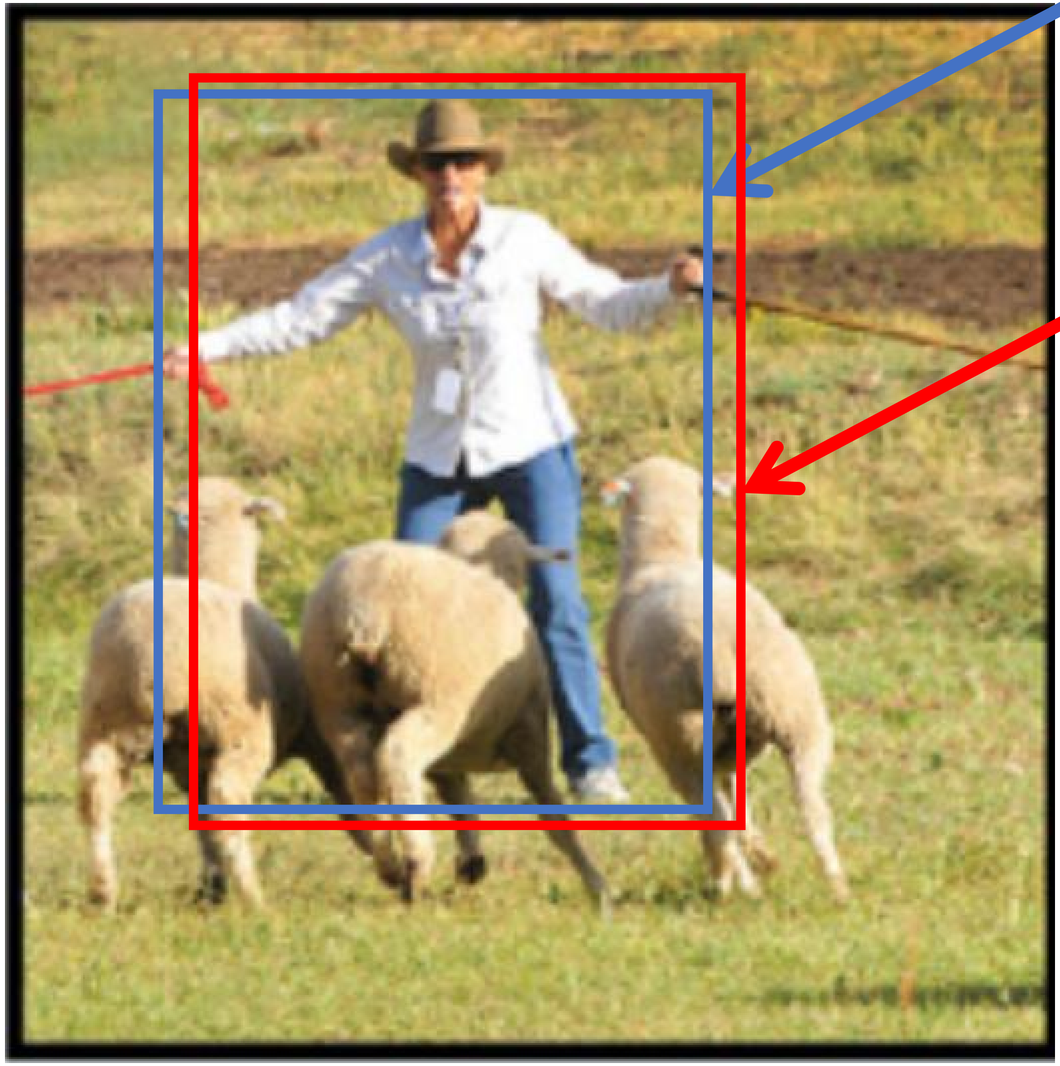


https://medium.com/@nikasa1889/the-modern-history-of-object-recognition-infographic-aea18517c318

Bounding box proposal

Region of interest, region proposal, box proposal

Ground truth



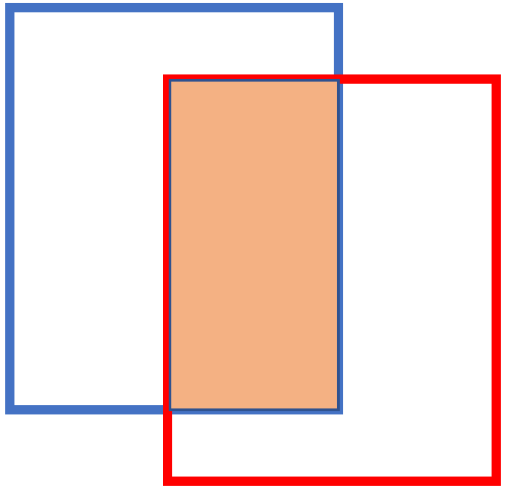
Proposed bounding box

<u>5 parameters</u>

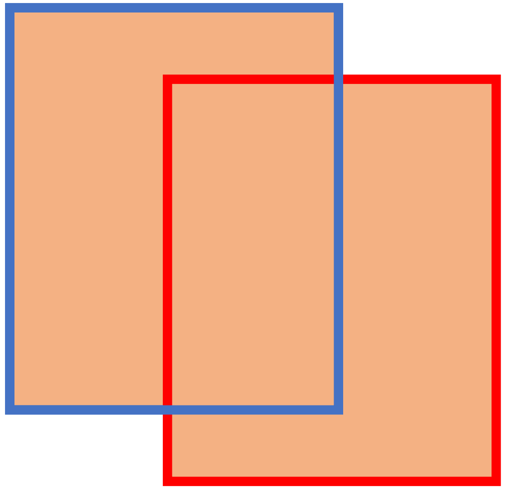
- w, h
- x, y
- ~~confidence score: how likely it contains an object & accuracy of the box~~

SSD predicts 4 parameters + class score for each box

How good: Intersection over Union (IOU)

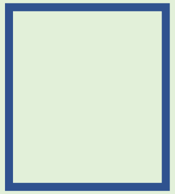
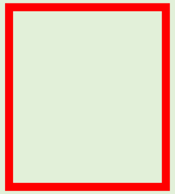


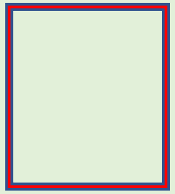
$$\text{IOU} = \frac{\textbf{Overlap Area}}{\textbf{Union Area}}$$



Jaccard overlap

Examples

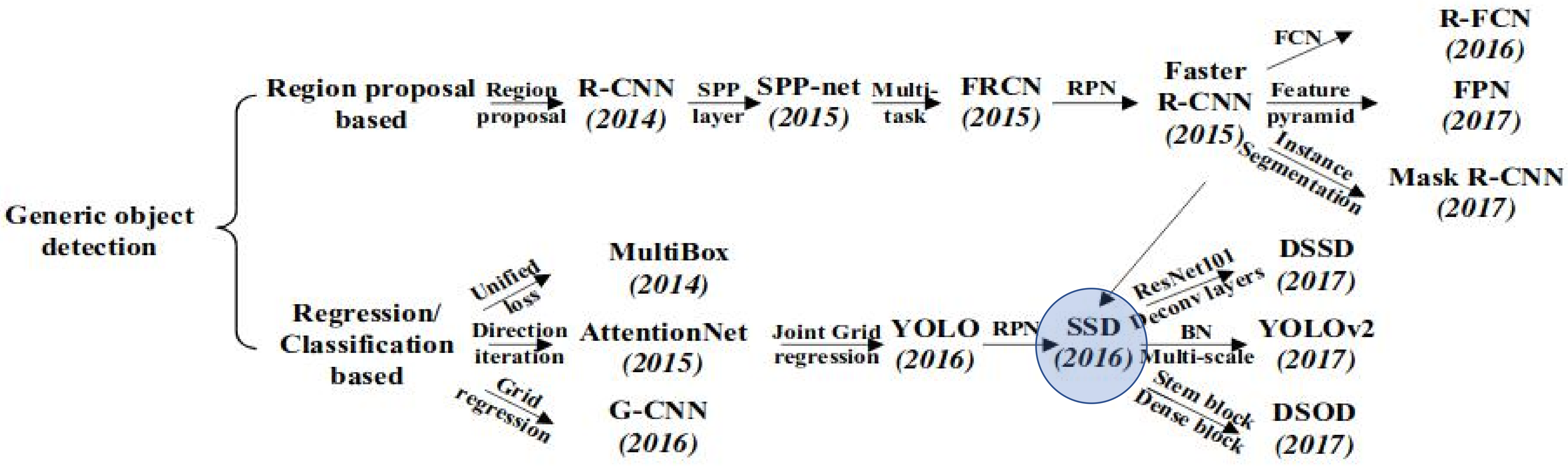
0:  

1: 

Outlines

- Review a few key concepts in object detection
- **SSD** **(arXiv: 1512.02325)** *focusing on the difference with YOLO*
  - design
  - loss function
  - training

SSD's relation with other detectors



SSD: single shot detector

Single shot
(Look once)

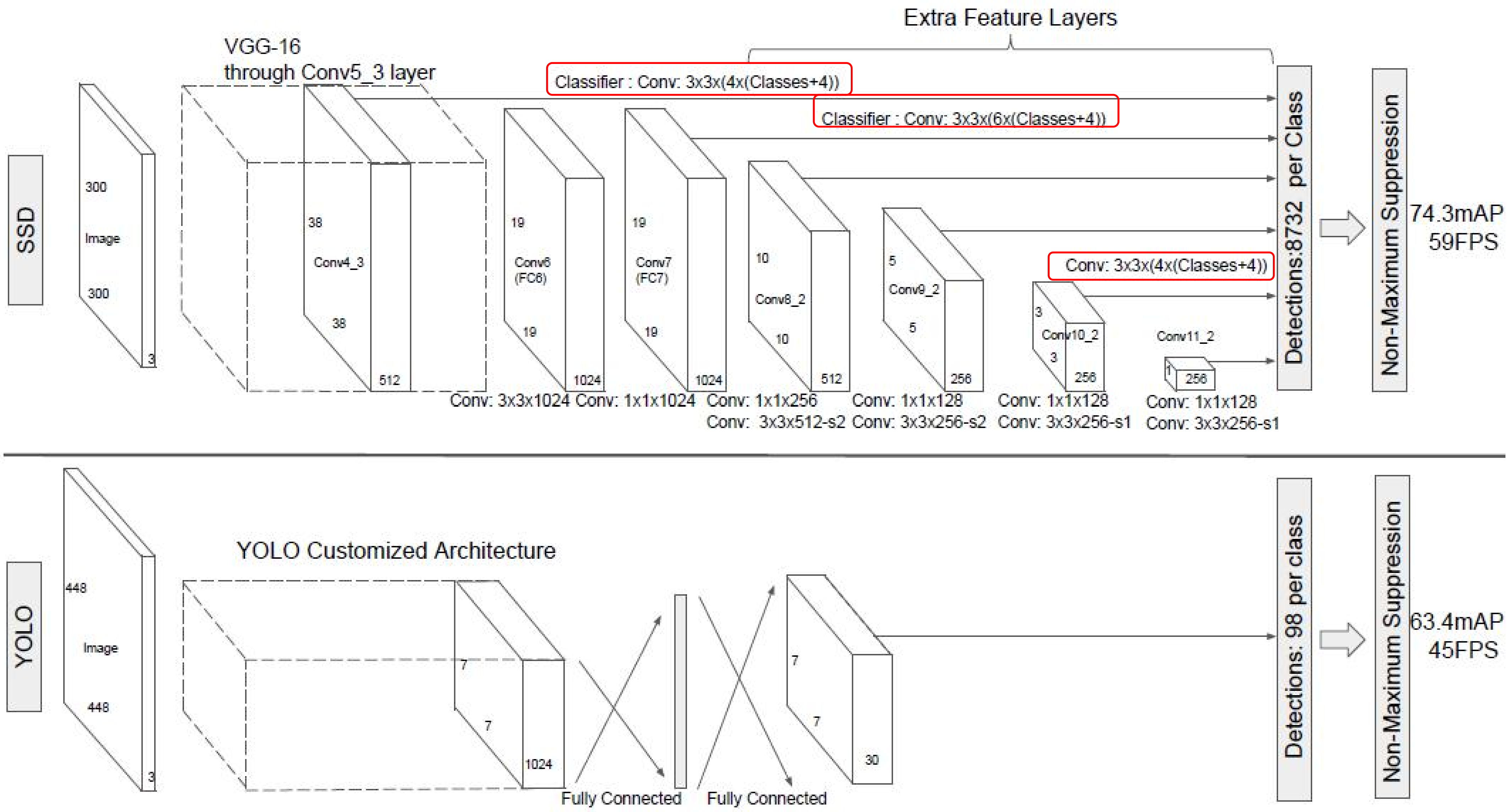
Results

- x, y, w, h
- ~~confidence score: contain an object & box accuracy~~
- class score: belong to a class

use CNN

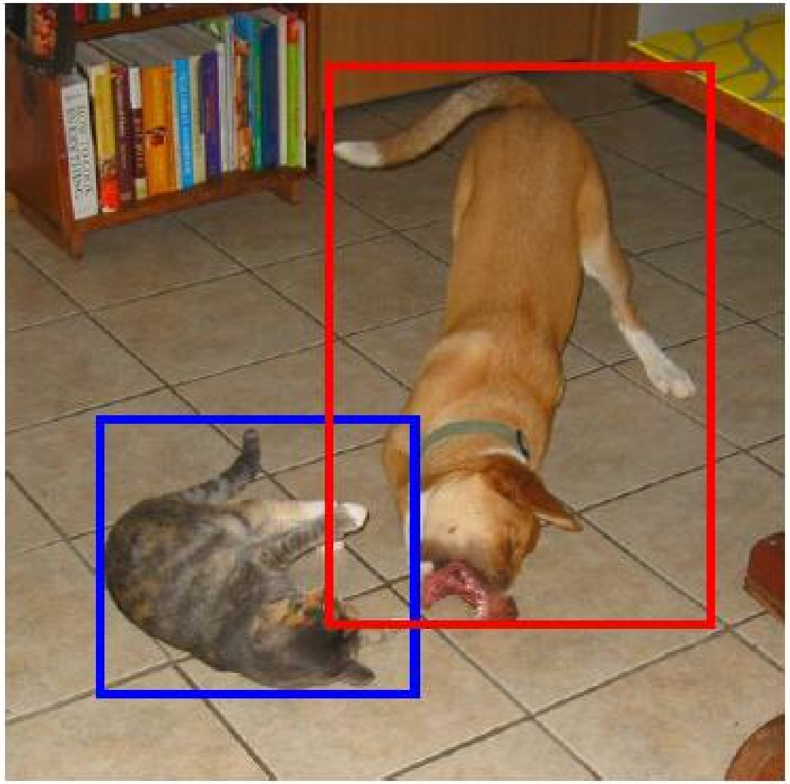
use regression with convolution

CNN backbone (base network)

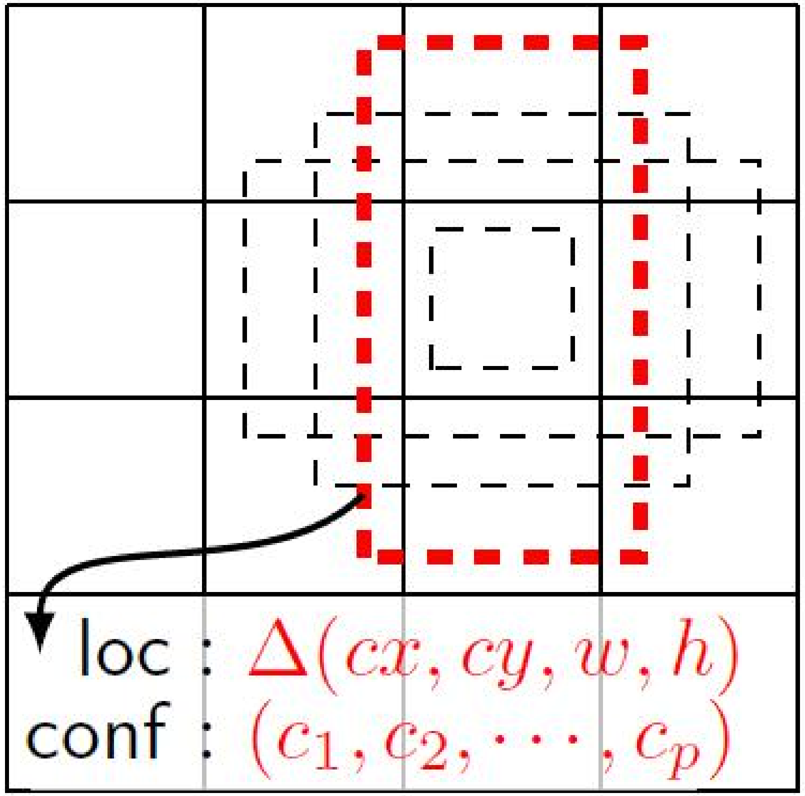


Bounding box

- YOLO uses grid on raw images.
- SSD takes advantage of the feature map,
  which is a "grid" coarse-grained from the raw images.



(a) Image with GT boxes



(c) 4×4 feature map

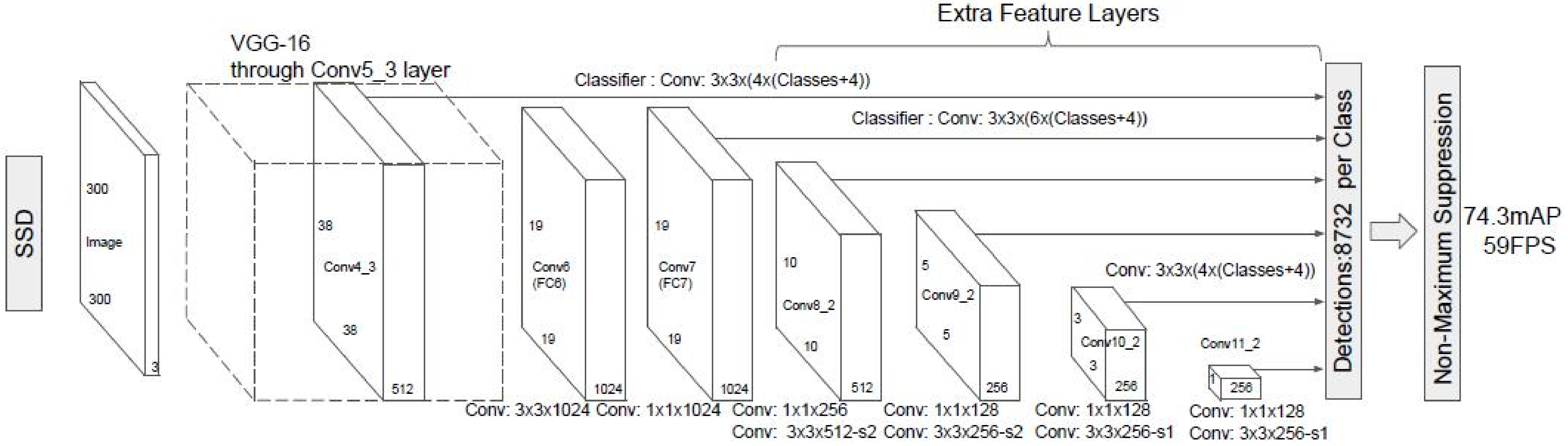
<u>Parameters for one box</u>

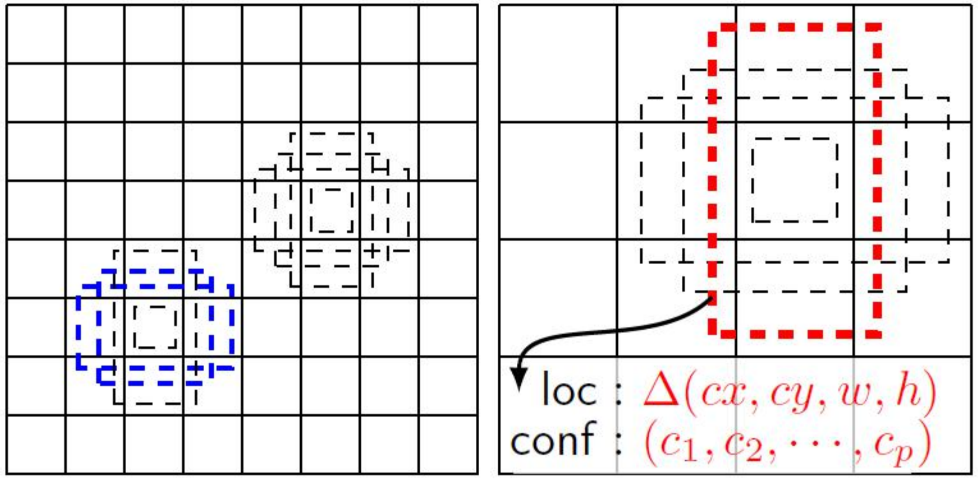
- x, y, w, h
- class score:
  belong to a class

<u>Number of boxes in feature map (FM)</u>

(4+class)*(FM size)

Multiscale bounding box



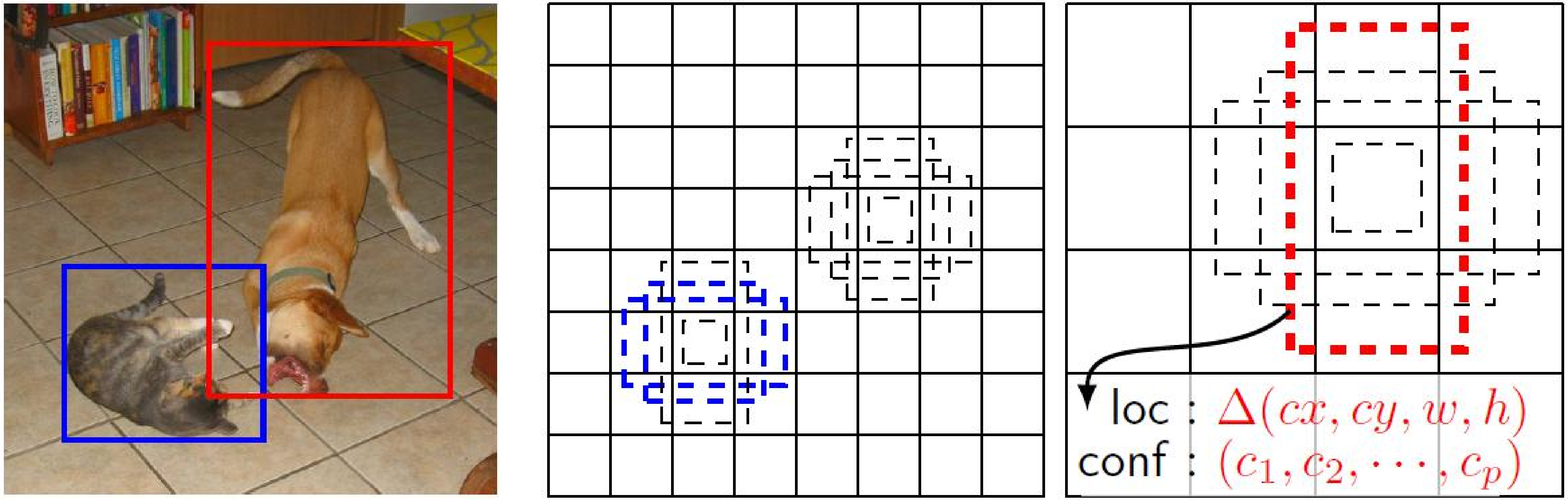


(b) 8×8 feature map (c) 4×4 feature map

Each layer is targeted to the given scale

- $s_k = s_{\text{min}} + \frac{s_{\text{max}} - s_{\text{min}}}{m-1}(k-1), \quad k \in [1, m]$
- $w_k^a = s_k\sqrt{a_r} \qquad h_k^a = s_k/\sqrt{a_r}$
- a_r: aspect ratio of default box
- default box center = grid center
- box is learnable

Rules for multiscale bounding box



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

training
➢ IoU > 0.5 with ground truth
➢ many boxes -> one ground truth

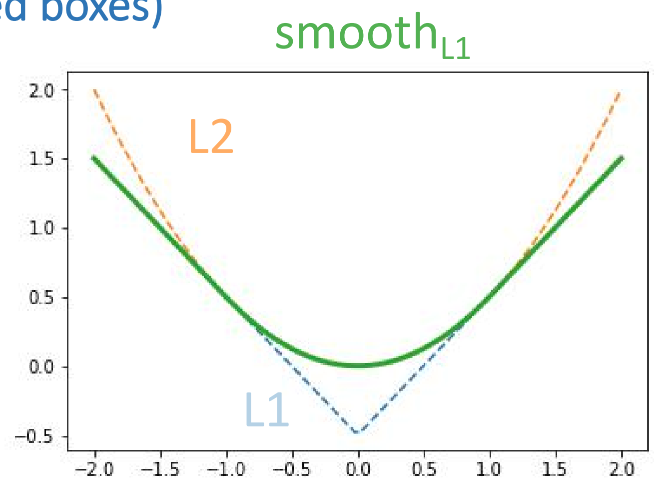
inference
1. filter out confidence < 0.01
2. non-maximum suppression

Loss function (before rescaled by the number of matched boxes)

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\,\mathrm{smooth_{L1}}(l_i^m - \hat{g}_j^m)$$

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

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



+ α x

=1

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

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

hard negative mining

1. sort by c^0
2. add top ones to loss function
3. neg : pos <= 3:1

Training

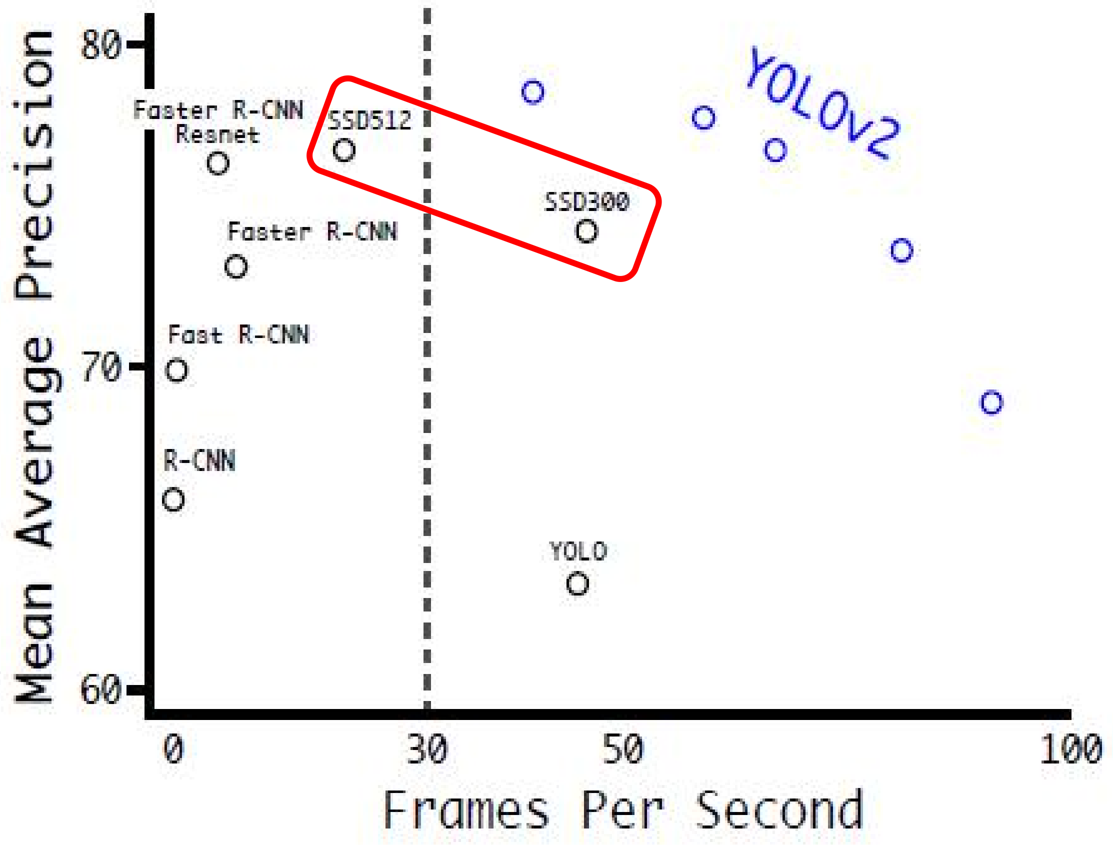
- Mutli-resolution input is not necessary.

- Low-res (300x300, 512x512) is enough.

  Faster R-CNN: 1000x600

- Data augmentation is very useful.
  - zoom in: image patches with size [0.1, 1] and aspect ratio [0.5, 2]
  - zoom out: randomly place the image on a canvas of 16x of the original image size filled with mean values before random crop. <u>*(It increases small object accuracy. Maybe we can replace it by randomly cropping a high-res image.)*</u>
  - horizontal flip
  - photo-metric distortion

Performance



From YOLOv2 paper
arXiv: 1612.08242

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

From SSD paper
arXiv: 1512.02325