

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

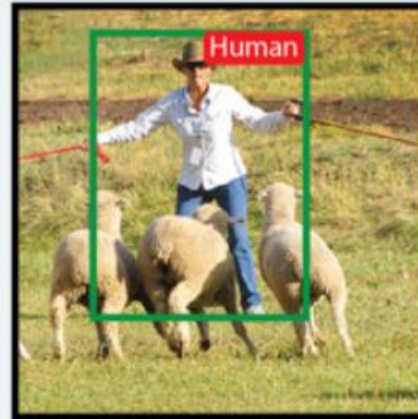
Common tasks on images



Image Classification

Classify an image based on the dominant object inside it.

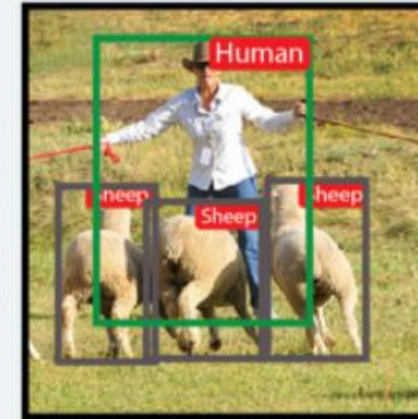
datasets: MNIST, CIFAR, ImageNet



Object Localization

Predict the image region that contains the dominant object. Then image classification can be used to recognize object in the region

datasets: ImageNet



Object Recognition

Localize and classify all objects appearing in the image. This task typically includes: proposing regions then classify the object inside them.

datasets: PASCAL, COCO



Semantic Segmentation

Label each pixel of an image by the object class that it belongs to, such as human, sheep, and grass in the example.

datasets: PASCAL, COCO



Instance Segmentation

Label each pixel of an image by the object class and object instance that it belongs to.

datasets: PASCAL, COCO



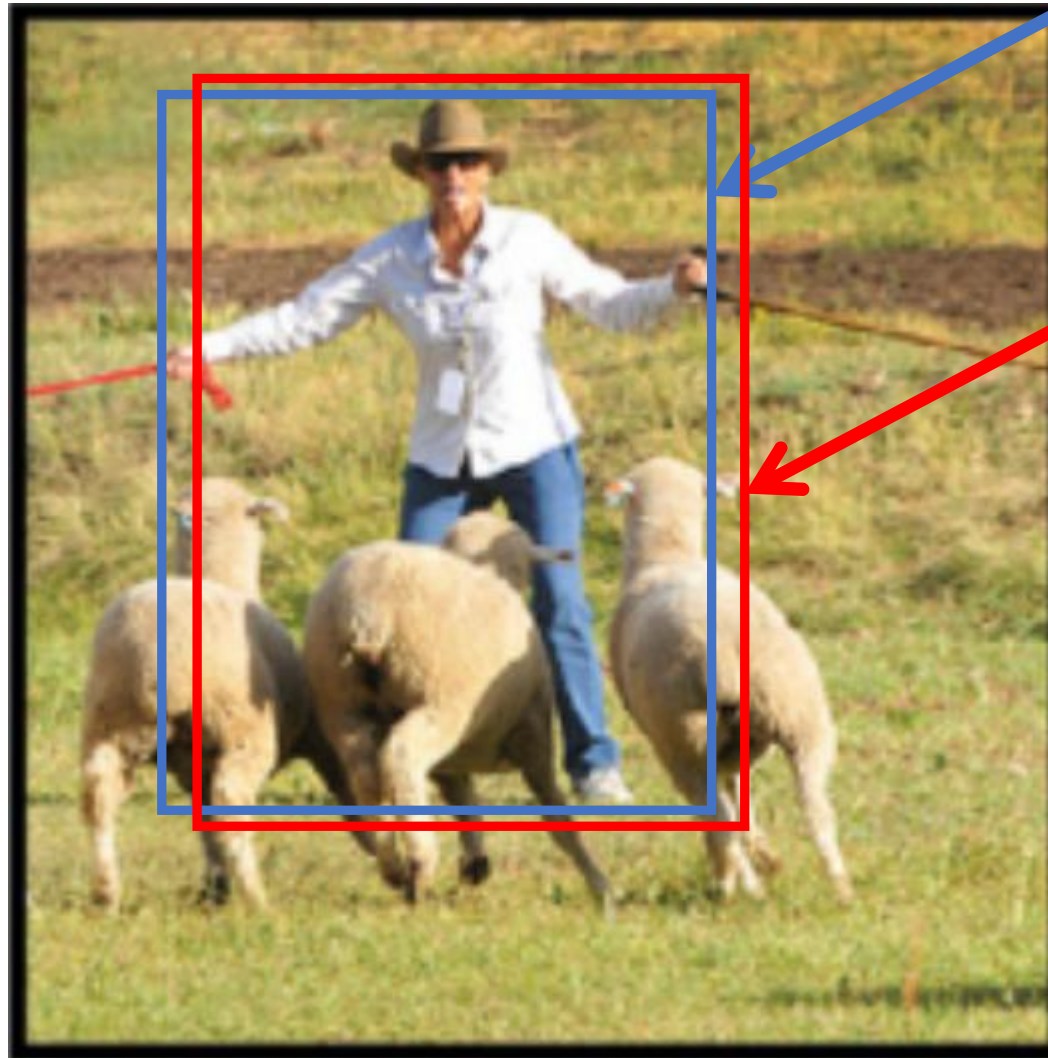
Keypoint Detection

Detect locations of a set of predefined keypoints of an object, such as keypoints in a human body, or a human face.

datasets: COCO

Bounding box proposal

Region of interest, region proposal, box proposal



Ground truth

Proposed bounding box

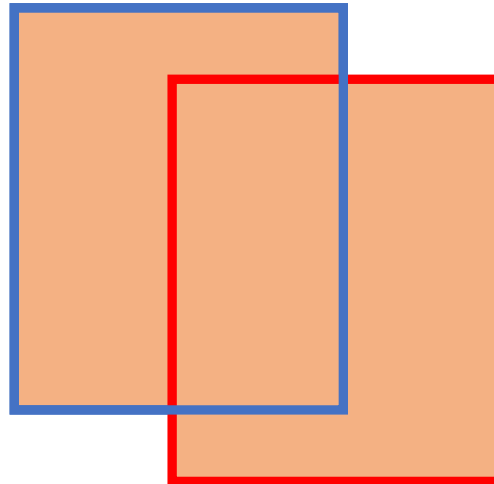
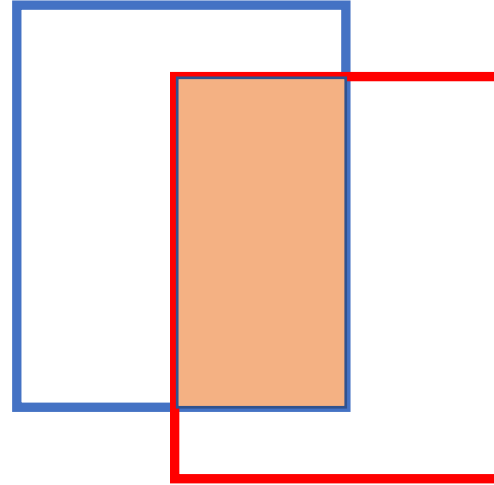
5 parameters

- 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)

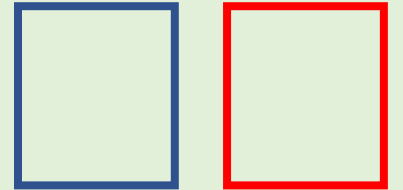
$$\text{IOU} = \frac{\text{Overlap Area}}{\text{Union Area}}$$



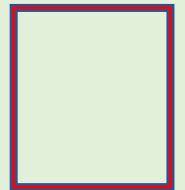
Jaccard overlap

Examples

0:



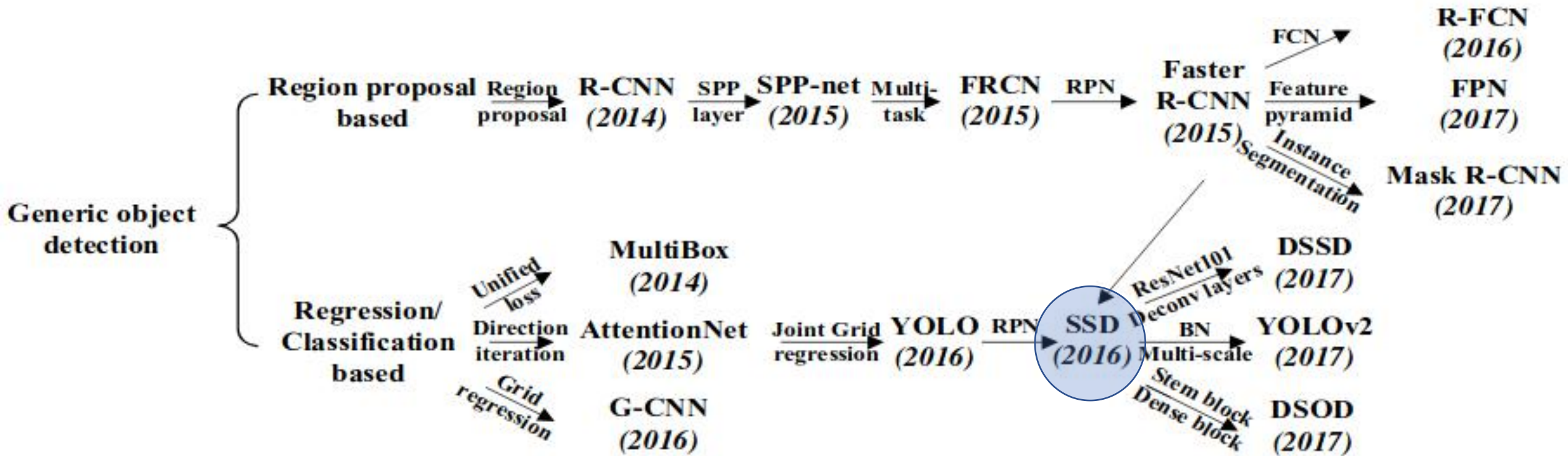
1:



Outlines

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

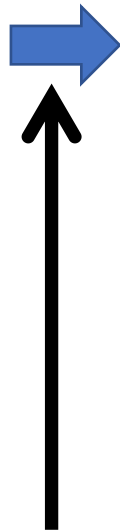
SSD's relation with other detectors



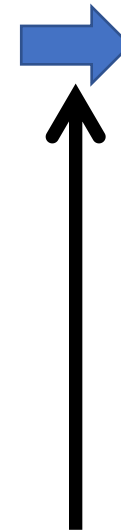
SSD: single shot detector



use CNN



Single shot
(Look once)

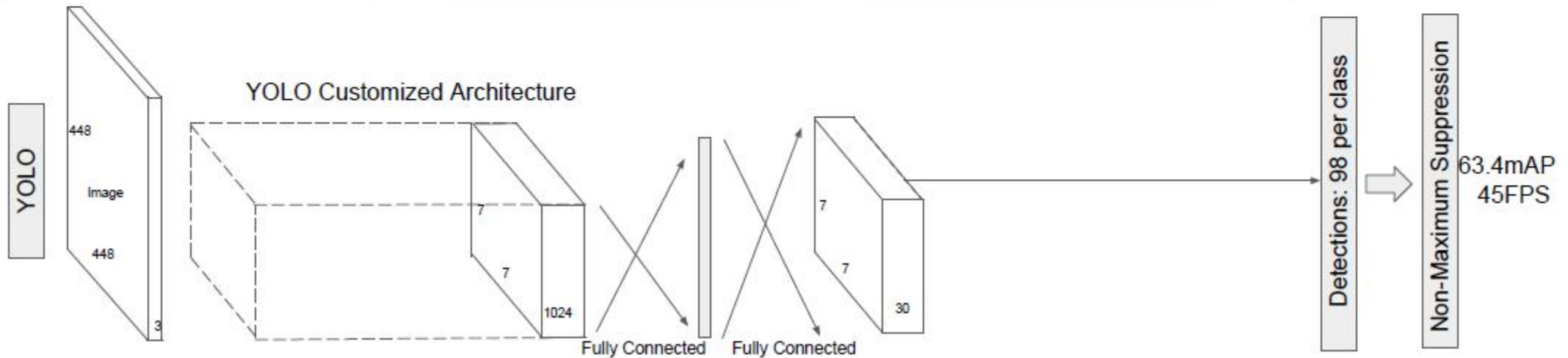
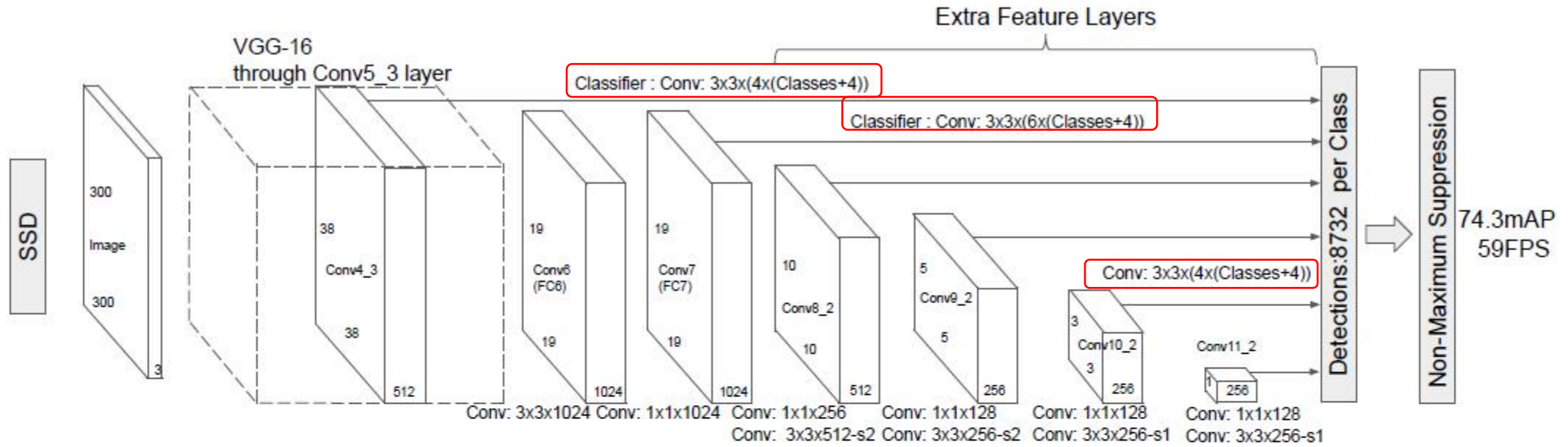


Results

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

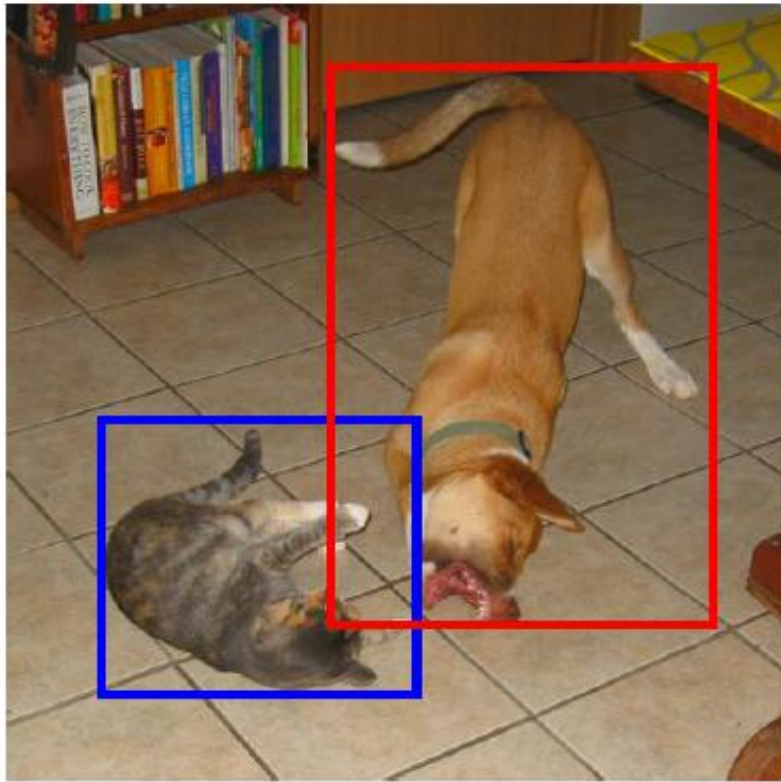
*use regression
with convolution*

CNN backbone (base network)

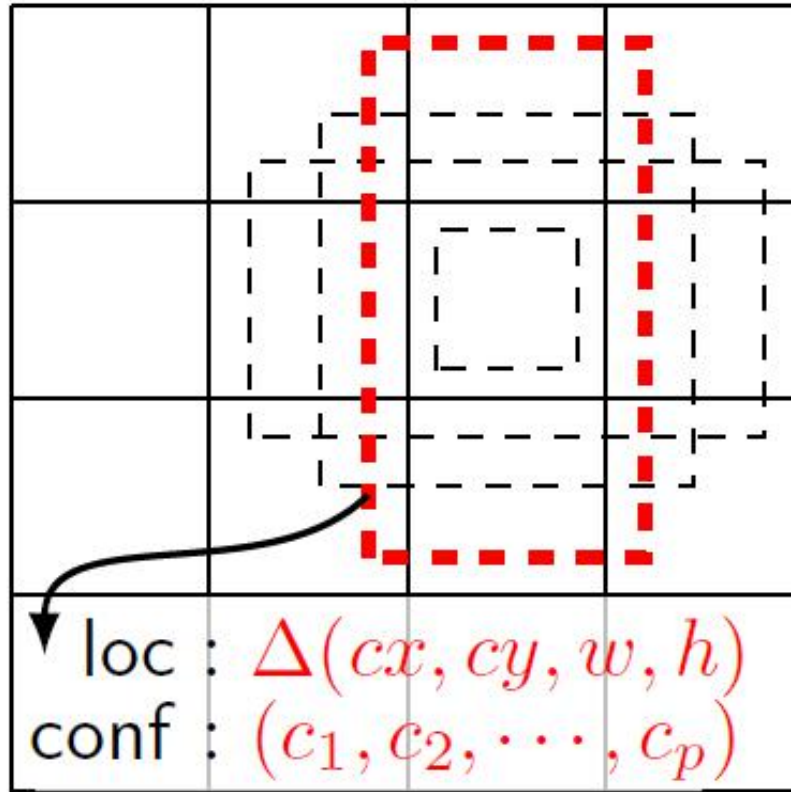


Bounding box

- 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

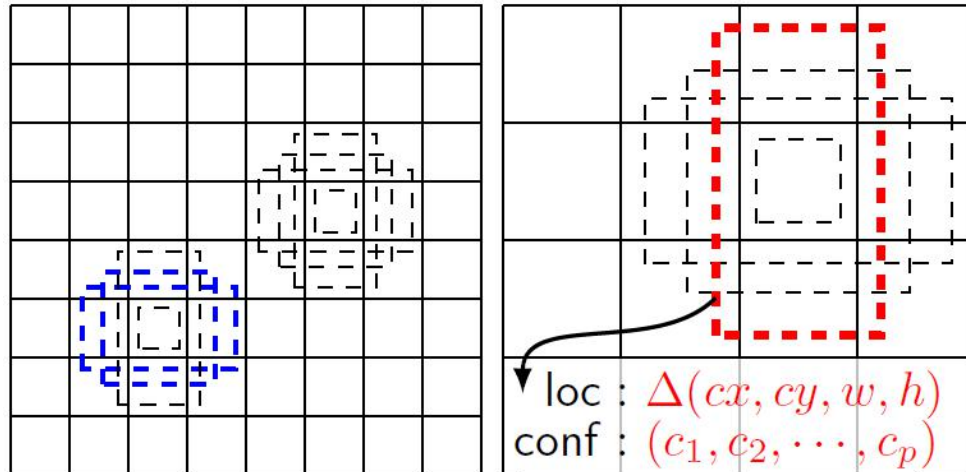
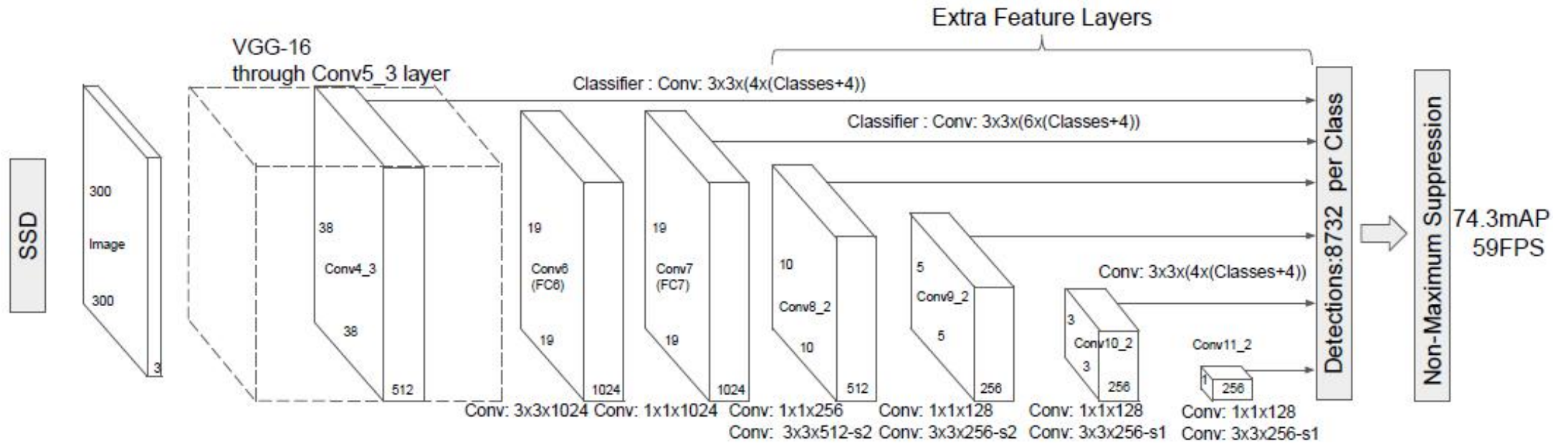
Parameters for one box

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

Number of boxes in feature map (FM)

$$(4+\text{class}) * (\text{FM size})$$

Multiscale bounding box

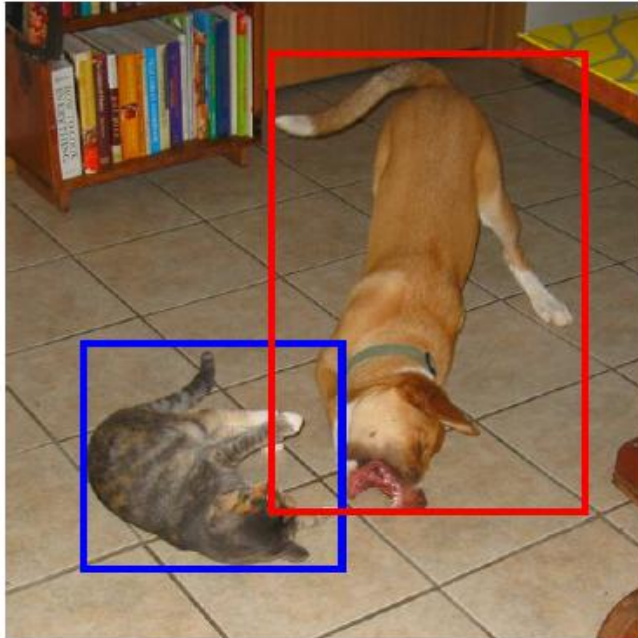


(b) 8×8 feature map (c) 4×4 feature map

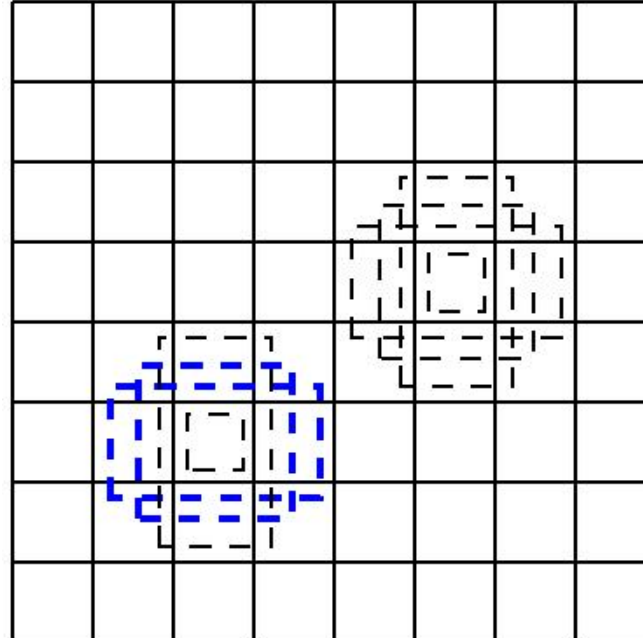
Each layer is targeted to the given scale

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

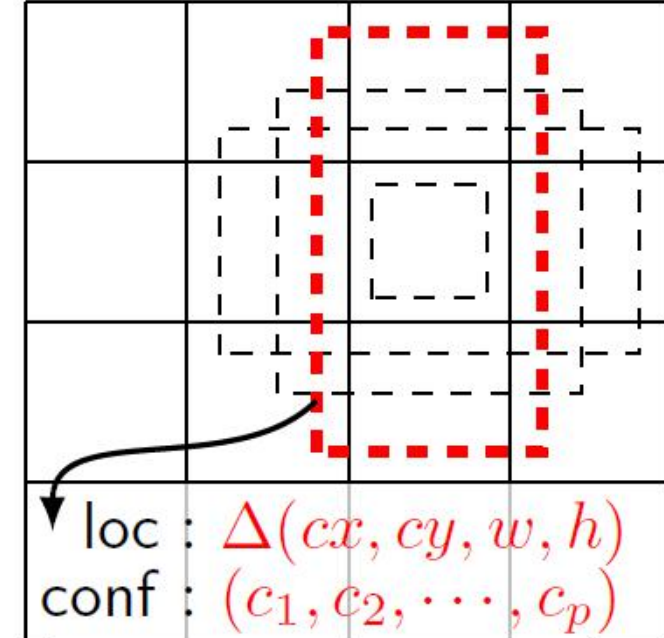
Rules for multiscale bounding box



(a) Image with GT boxes



(b) 8×8 feature map



(c) 4×4 feature map

training

- $\text{IoU} > 0.5$ with ground truth
- many boxes \rightarrow one ground truth

inference

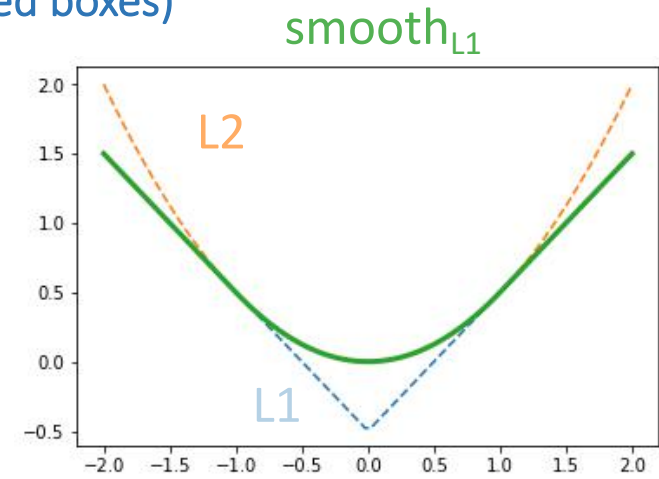
1. filter out confidence < 0.01
2. non-maximum suppression

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

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

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

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



+ α x
=1

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

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

hard negative
mining

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

Training

➤ Mutli-resolution input is not necessary.

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

Faster R-CNN: 1000x600

➤ Data augmentation is very useful.

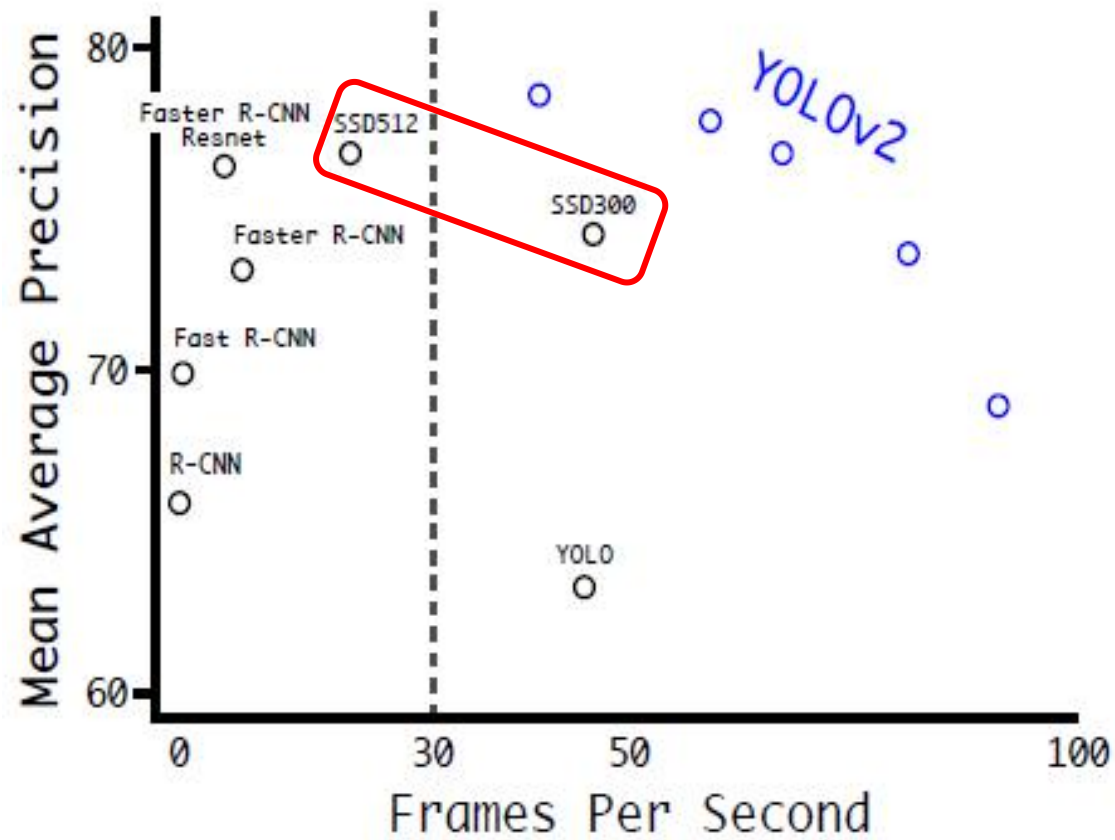
➤ zoom in: image patches with size $[0.1, 1]$ and aspect ratio $[0.5, 2]$

➤ zoom out: randomly place the image on a canvas of 16x of the original image size filled with mean values before random crop. (It increases small object accuracy. Maybe we can replace it by randomly cropping a high-res image.)

➤ horizontal flip

➤ photo-metric distortion

Performance



From YOLOv2 paper
[arXiv: 1612.08242](https://arxiv.org/abs/1612.08242)

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 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](https://arxiv.org/abs/1512.02325)