SSD: Single Shot MultiBox Detector

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C.Berg

[arXiv][demo][code] (Mar 2016)

Image Processing Group
Signal Theory and Communications Department
Universitat Politècnica de Catalunya. BARCELONATECH

Slides by Míriam Bellver
Computer Vision Reading Group, UPC
28th October, 2016

Outline

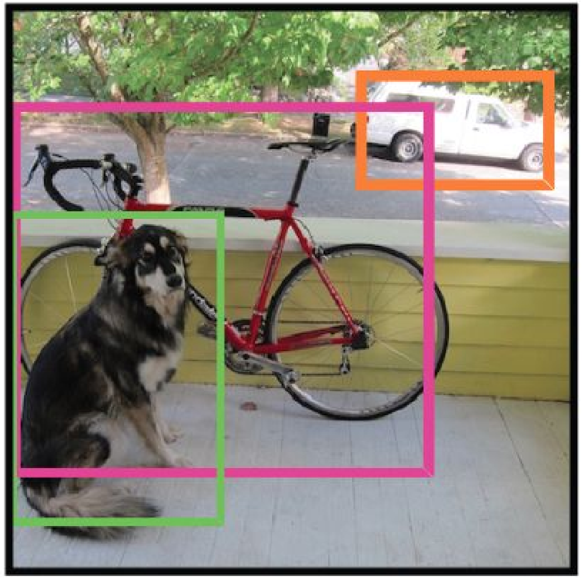
1.

Introduction

SSD: Single Shot MultiBox Detector

Introduction

Object detection



Introduction

Current object detection systems

Image pixels → Object proposals generation → resample pixels for each BBOX / resample features for each BBOX → high quality classifier

Introduction

Current object detection systems

Image pixels

Object proposals generation

resample pixels for each BBOX

resample features for each BBOX

high quality classifier

Computationally too intensive and too slow for real-time applications

Faster R-CNN 7 FPS

Introduction

Current object detection systems

Image pixels → Object proposals generation → ~~resample pixels for each BBOX~~ / ~~resample features for each BBOX~~ → high quality classifier

Computationally too intensive and too slow for real-time applications

Faster R-CNN 7 FPS

Introduction

SSD: First deep network based object detector that does **not resample pixels or features** for bounding box hypotheses and is **as accurate as approaches that do**.

Improvement in **speed vs accuracy trade-off**

Introduction

Method	*mAP*	**FPS**	**# Boxes**
Faster R-CNN [2](VGG16)	73.2	7	300
Faster R-CNN [2](ZF)	62.1	17	300
YOLO [5]	63.4	45	98
Fast YOLO [5]	52.7	155	98
SSD300	72.1	58	7308
SSD500	**75.1**	23	20097

Introduction

Contributions:

- ▷ A single-shot detector for multiple categories that is faster than state of the art single shot detectors (YOLO) and as accurate as Faster R-CNN
- ▷ Predicts category scores and boxes offset for a fixed set of default BBs **using small convolutional filters applied to feature maps**
- ▷ Predictions of different scales from feature maps of different scales, and separate predictions by aspect ratio
- ▷ End-to-end training and high accuracy, **improving speed vs accuracy trade-off**

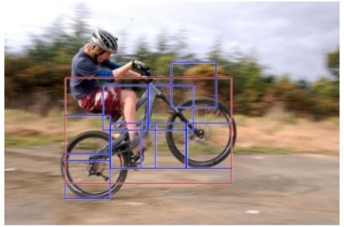
2.
Related Work

SSD: Single Shot MultiBox Detector

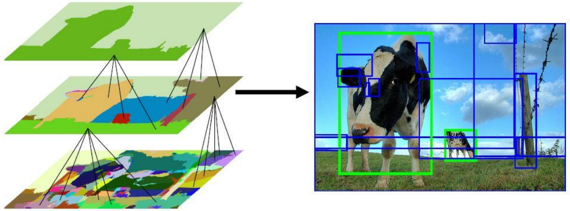
Object Detection prior to CNNs

Two different traditional approaches:

- Sliding Window: e.g. Deformable Part Model (*DPM*)



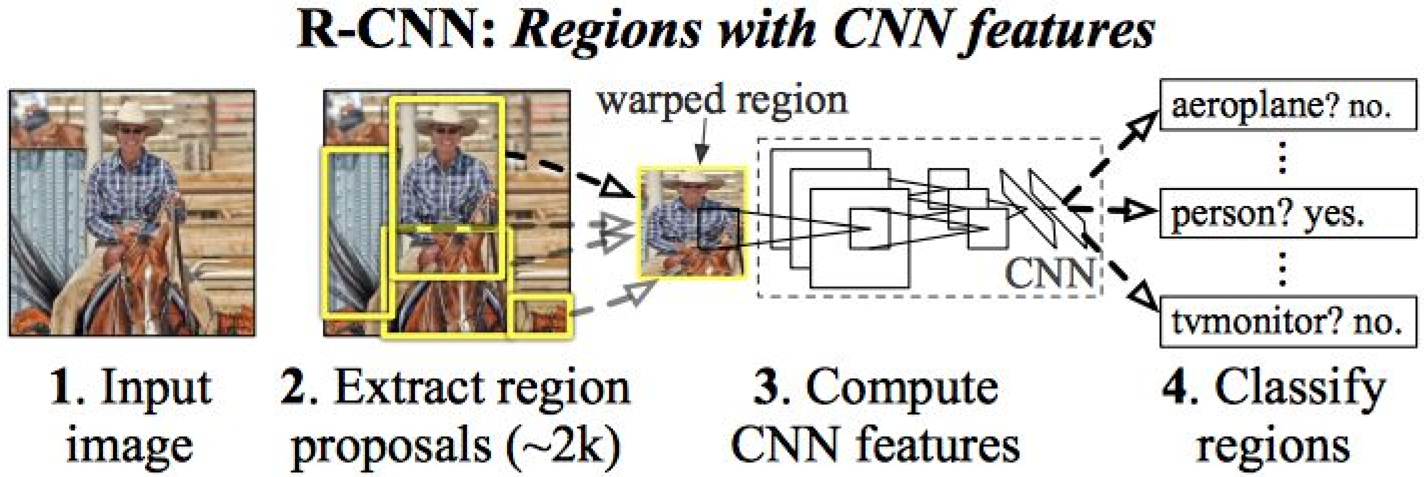
- Object proposals: e.g. Selective Search



Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008, June). A discriminatively trained, multiscale, deformable part model
Uijlings, J. R., van de Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition

Object detection with CNN's

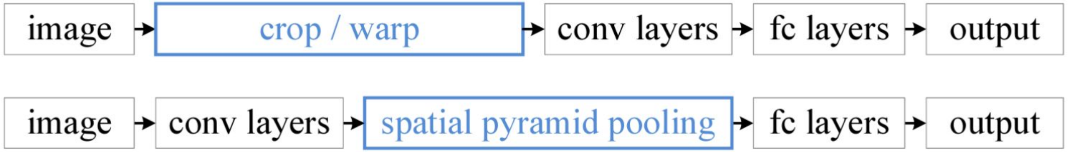
- R-CNN



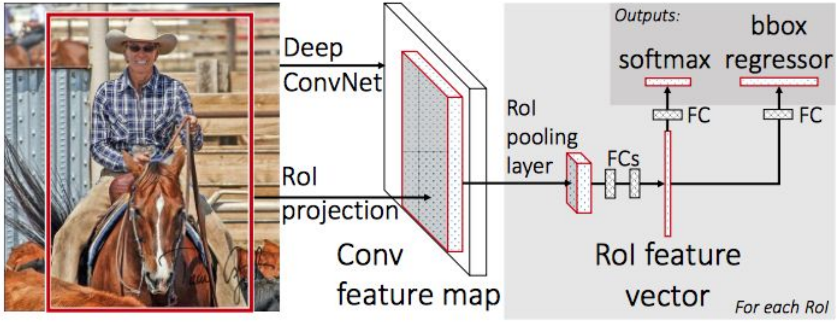
Girshick, R. (2015). Fast r-cnn

Leveraging the object proposals bottleneck

- SPP-net



- Fast R-CNN



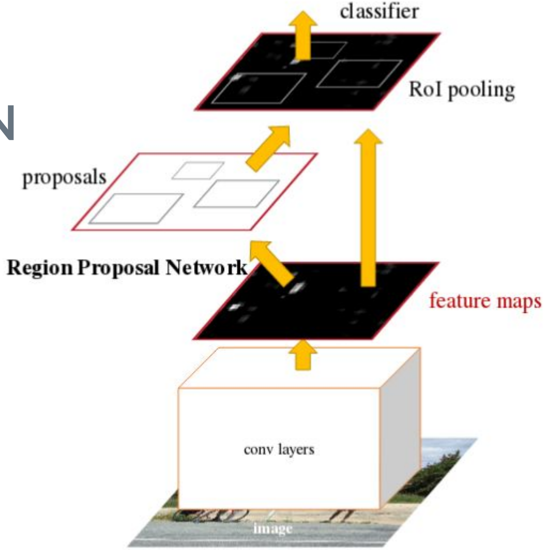
He, K., Zhang, X., Ren, S., & Sun, J. (2014, September). Spatial pyramid pooling in deep convolutional networks for visual recognition
Girshick, R. (2015). Fast r-cnn.

Improving quality of proposals using CNNs

Low-level features object proposals

Proposals generated directly from a DNN

E.g. : MultiBox, Faster R-CNN



Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks
Szegedy, C., Reed, S., Erhan, D., & Anguelov, D. (2014). Scalable, high-quality object detection.

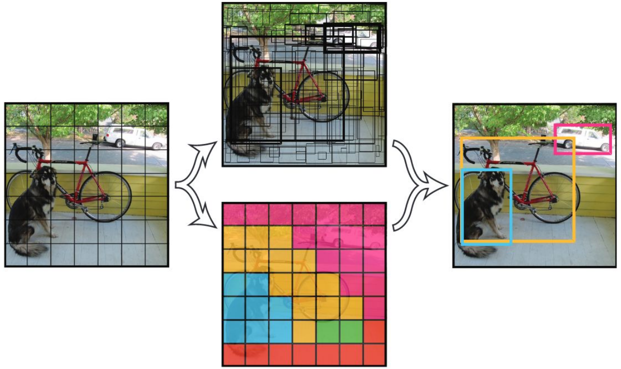
Single-shot detectors

Instead of having two networks

Region Proposals Network + Classifier Network

In Single-shot architectures, bounding boxes and confidences for multiple categories are predicted directly with a single network

e.g. : Overfeat, YOLO



Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection
Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks.

Single-shot detectors

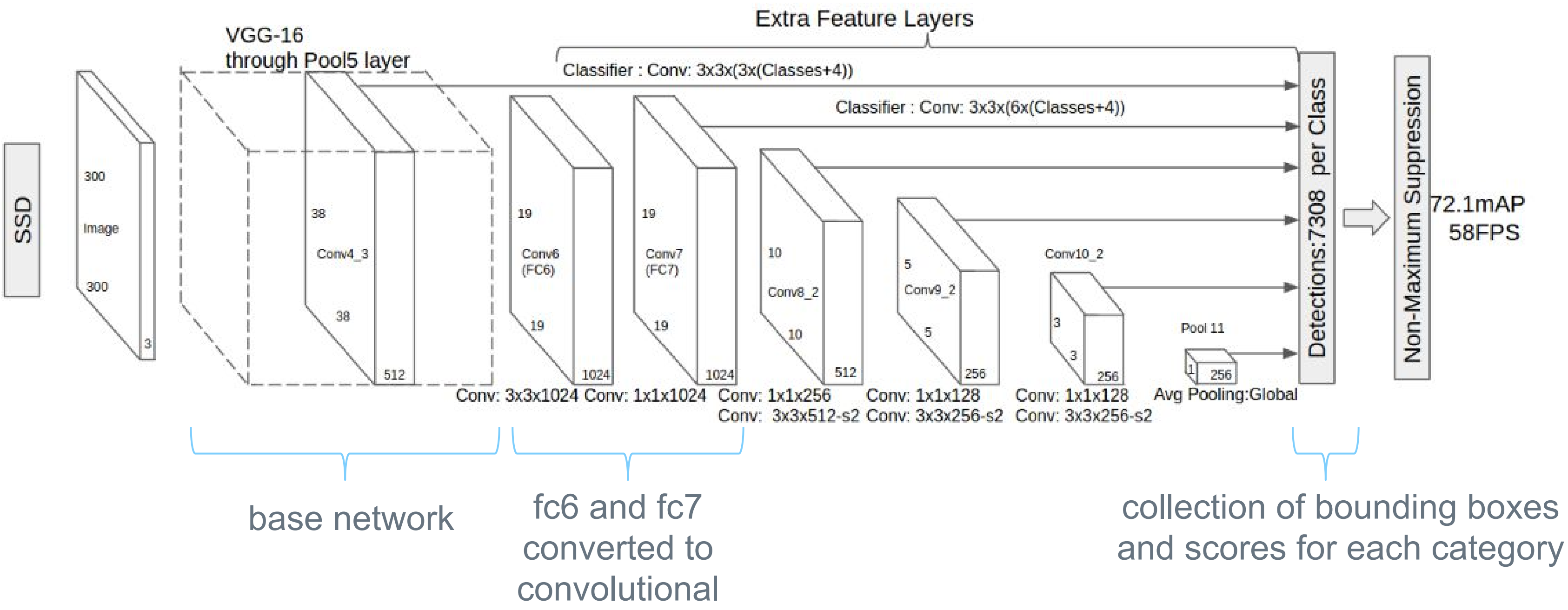
Main differences of SSD over YOLO and Overfeat:

- Small **conv. filters** to predict object **categories and offsets in BBs locations**, using separate predictors for different aspect ratios, and applying them on different feature maps to perform detection on multiple scales

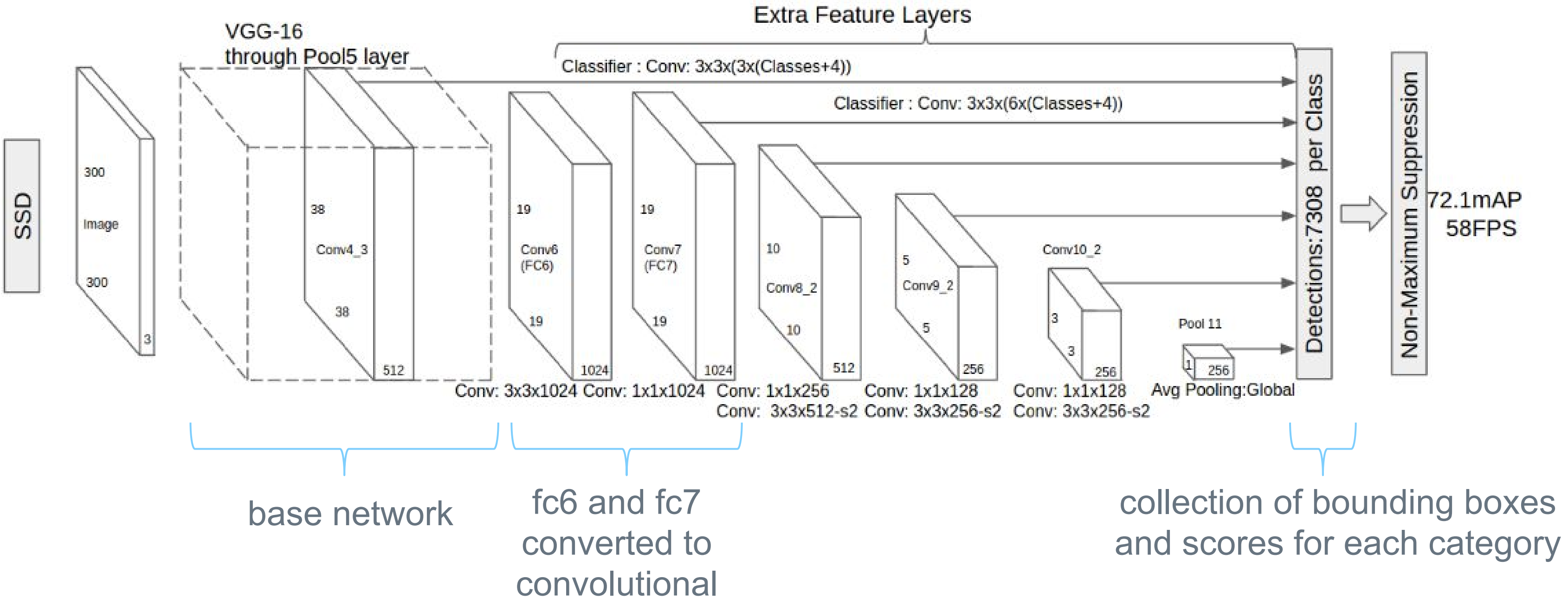
3.1
The Single Shot Detector (SSD)

Model

The Single Shot Detector (SSD)



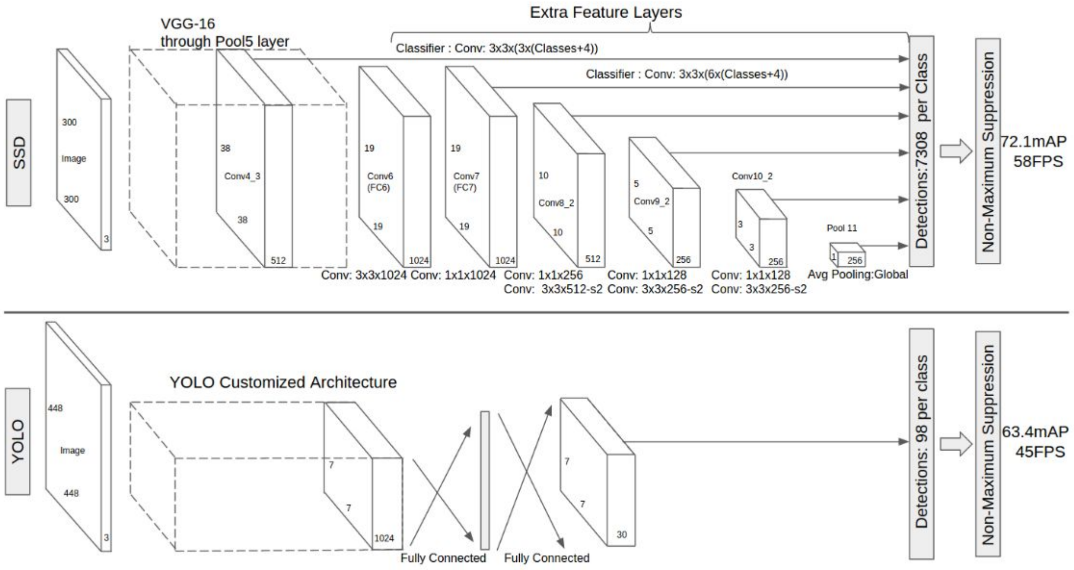
The Single Shot Detector (SSD)



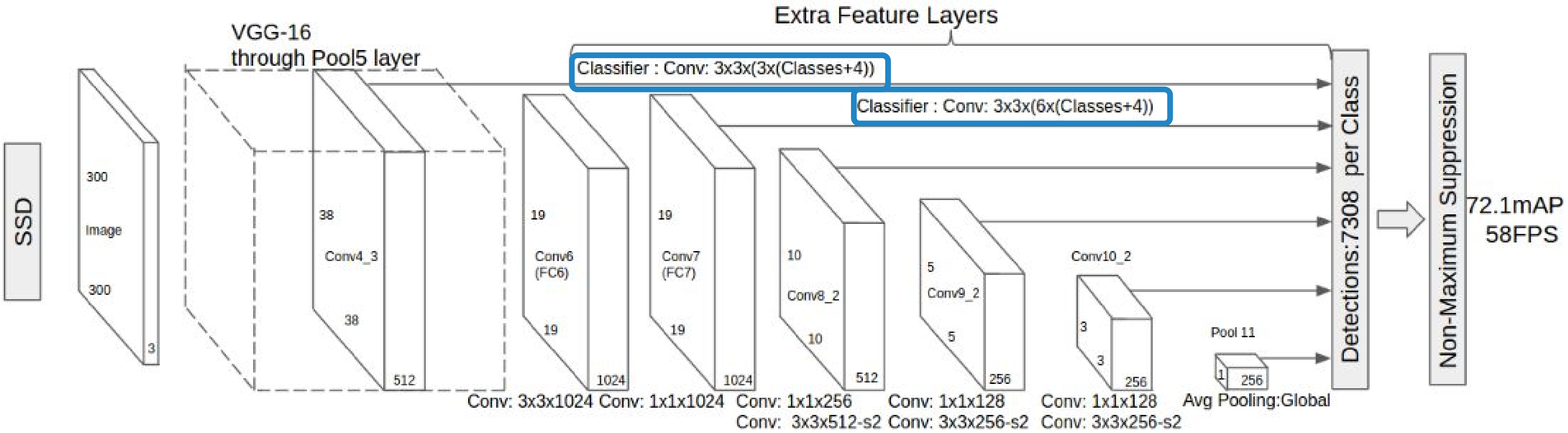
Multi-scale feature maps for detection: observe how conv feature maps decrease in size and allow predictions at multiple scales

The Single Shot Detector (SSD)

Comparison to YOLO



The Single Shot Detector (SSD)



Convolutional predictors for detection: We apply on top of each conv feature map a set of filters that predict detections for different aspect ratios and class categories

The Single Shot Detector (SSD)

What is a detection ?

Described by **four parameters** (center bounding box x and y, width and height)

Class category

For all categories we need for a detection a total of #classes + 4 values

The Single Shot Detector (SSD)

Detector for SSD:

Each detector will output a single value, so we need **(*classes* + 4) detectors for a detection**

The Single Shot Detector (SSD)

Detector for SSD:

Each detector will output a single value, so we need **(*classes* + 4) detectors for a detection**

BUT there are different types of detections!

The Single Shot Detector (SSD)

Different “classes” of detections

aspect ratio 2:1
for cats

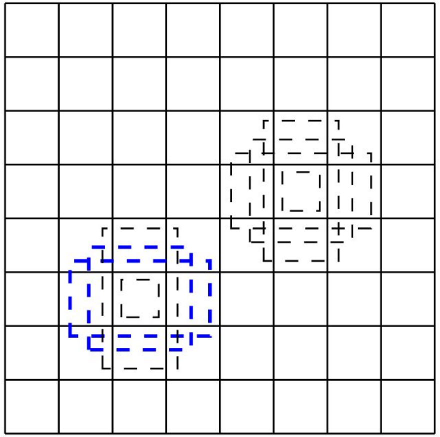
aspect ratio 1:2
for cats

aspect ratio 1:1
for cats

The Single Shot Detector (SSD)

Default boxes and aspect ratios

Similar to the *anchors* of Faster R-CNN, with the difference that SSD applies them on several feature maps of different resolutions



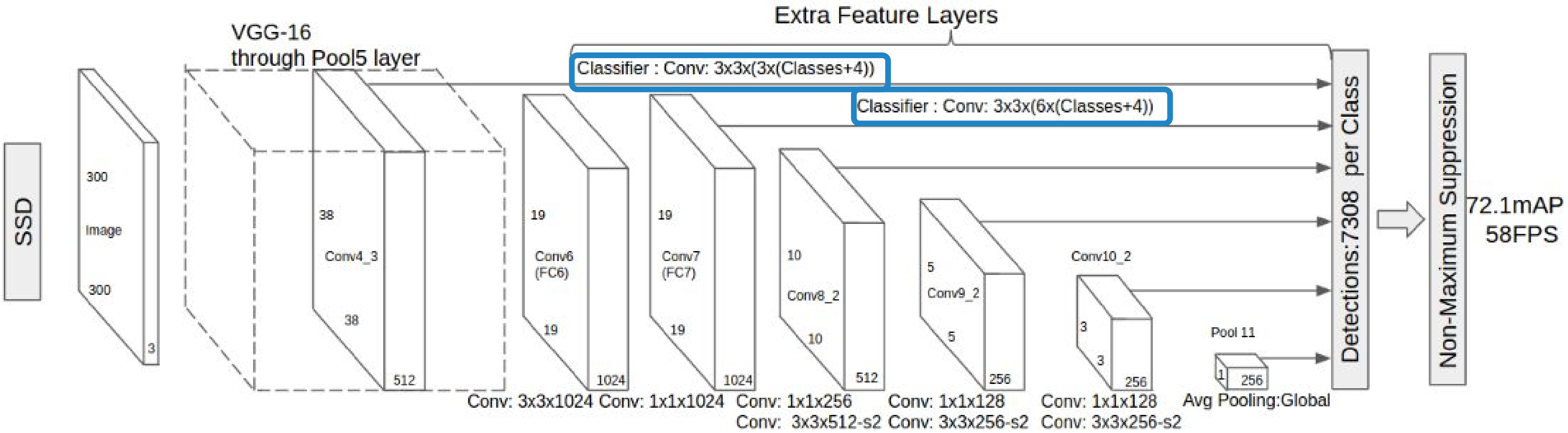
The Single Shot Detector (SSD)

Detector for SSD:

Each detector will output a single value, so we need
(*classes* + 4) detectors for a detection

as we have **#default boxes**, we need
(classes + 4) x #default boxes detectors

The Single Shot Detector (SSD)

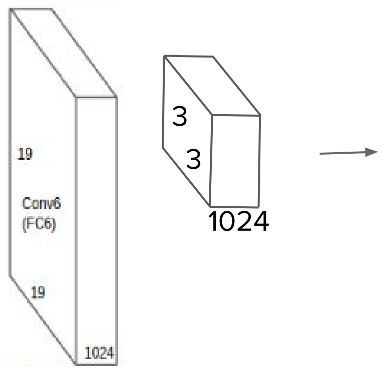


Convolutional predictors for detection: We apply on top of each conv feature map a set of filters that predict detections for different aspect ratios and class categories

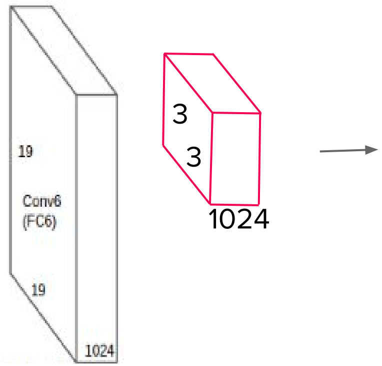
The Single Shot Detector (SSD)

For each feature layer of *m* x *n* with *p* channels we apply **kernels of 3 x 3 x *p*** to produce either a **score for a category**, or a **shape offset** relative to a default bounding box coordinates

Example for conv6:



shape offset xo to default box of aspect ratio: 1 for category 1



score for category 1 for the default box of aspect ratio: 1

up to ***classes+4*** filters for each default box considered at that conv feature map

The Single Shot Detector (SSD)

For each feature layer of *m* x *n* with *p* channels we apply kernels of 3 x 3 x *p* to produce either a **score for a category**, or a **shape offset** relative to a default bounding box coordinates

So, for each conv layer considered, there are

(*classes* + 4) x *default boxes* x *m* x *n* outputs

3.2

The Single Shot Detector (SSD)

Training

The Single Shot Detector (SSD)

SSD requires that ground-truth data is **assigned** to specific outputs in the fixed set of detector outputs

The Single Shot Detector (SSD)

Matching strategy:

For each ground truth box we have to select from **all the default boxes** the ones that best fit in terms of **location**, **aspect ratio** and **scale**.

- We select the default box with **best jaccard overlap**. Then every box has at least 1 correspondence.
- Default boxes with a jaccard overlap **higher than 0.5** are also selected

The Single Shot Detector (SSD)

Training objective:

Similar to MultiBox but handles multiple categories.

$$L(x,c,l,g) = \frac{1}{N}(\underbrace{L_{conf}(x,c)}_{\substack{\text{confidence loss} \\ \text{softmax loss}}} + \alpha \underbrace{L_{loc}(x,l,g)}_{\substack{\text{localization loss} \\ \text{Smooth L1 loss}}})$$

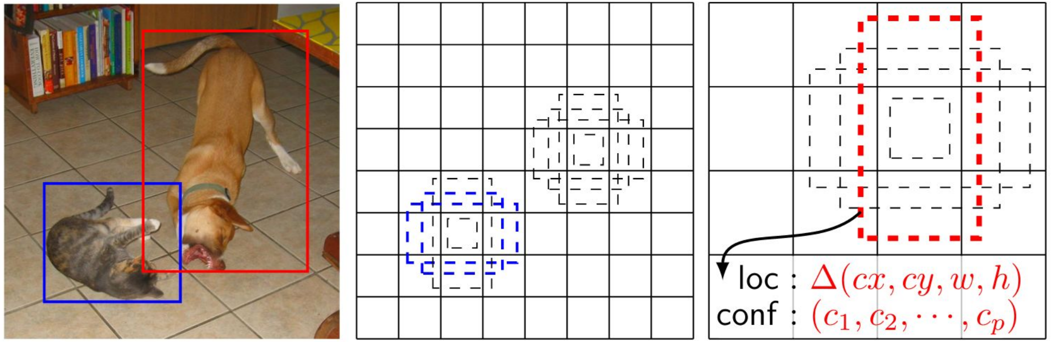
is 1 by
cross-validation

N: number of default matched BBs
x: is 1 if the default box is matched to a determined ground truth box, and 0 otherwise
l: predicted bb parameters
g: ground truth bb parameters
c: class

The Single Shot Detector (SSD)

Choosing scales and aspect ratios for default boxes:

- Feature maps from different layers are used to handle scale variance
- Specific feature map locations learn to be responsive to specific areas of the image and particular scales of objects



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

The Single Shot Detector (SSD)

Choosing scales and aspect ratios for default boxes:

- If *m* feature maps are used for prediction, the **scale** of the default boxes for each feature map is:

$$s_k = s_{\text{min}} + \frac{s_{\text{max}} - s_{\text{min}}}{m-1}(k-1), \quad k \in [1, m]$$

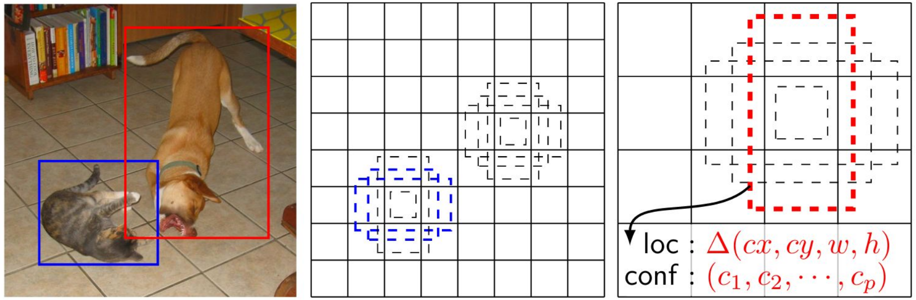
$s_{\text{min}} =$ 0.1

$s_{\text{max}} =$ 0.95

The Single Shot Detector (SSD)

Choosing scales and aspect ratios for default boxes:

▷ At each scale, different **aspect ratios** are considered:



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

$$a_r \in \{1, 2, 3, \tfrac{1}{2}, \tfrac{1}{3}\}$$

$$\boxed{(w_k^a} = s_k\sqrt{a_r})$$

$$\boxed{(h_k^a} = s_k/\sqrt{a_r})$$

width and height of default bbox

The Single Shot Detector (SSD)

Hard negative mining:

Significant imbalance between **positive** and **negative** training examples

- ▷ Use negative samples with **higher confidence score**
- ▷ Then the ratio of positive-negative samples is **3:1**

The Single Shot Detector (SSD)

Data augmentation:

Each training sample is randomly sampled by one of the following options:

- Use the original image
- Sample a path with a minimum jaccard overlap with objects
- Randomly sample a path

4.

Experimental Results

SSD: Single Shot MultiBox Detector

Experimental Results

Base network:

- VGG16 (with fc6 and fc7 converted to conv layers and pool5 from 2x2 to 3x3 using *atrous* algorithm, removed fc8 and dropout)
- It is fine-tuned using SGD
- Training and testing code is built on caffe

Database:

- Training: VOC2007 trainval and VOC2012 trainval (16551 images)
- Testing: VOC2007 test (4952 images)

Experimental Results

Mean Average Precision for PASCAL '07

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

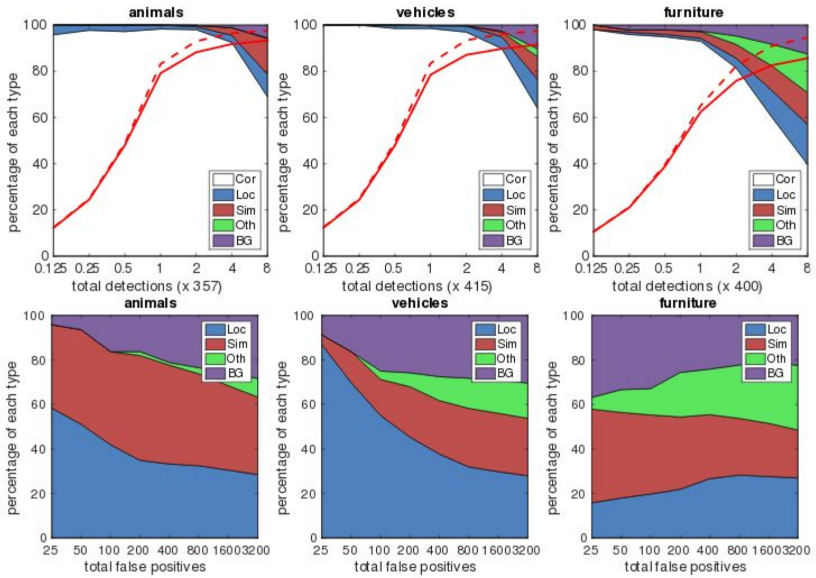
Experimental Results

Model analysis

	SSD300					
more data augmentation?		✔	✔	✔	✔	✔
use conv4_3?	✔		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔	✔			✔	✔
use atrous?	✔	✔	✔	✔		✔
VOC2007 test mAP	65.4	68.1	69.2	71.2	71.4	72.1

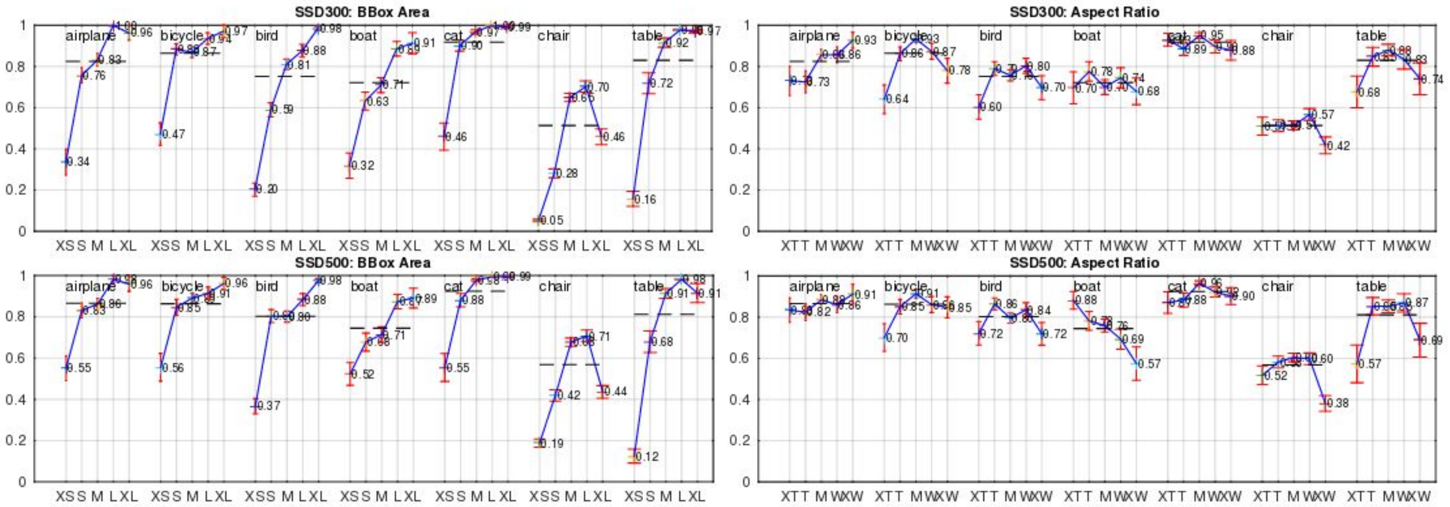
Experimental Results

Visualization for performance



Experimental Results

Sensitivity to different object characteristics with PASCAL 2007 :



Experimental Results

Database:

- Training: VOC2007 trainval and test and VOC2012 trainval (21503 images)
- Testing: VOC2012 test (10991 images)

Mean Average Precision for PASCAL '12

Method	*mAP*	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	**89.3**	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster [2]	70.4	**84.9**	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	**77.1**	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
YOLO [5]	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	70.3	84.2	76.3	69.6	53.2	40.8	78.5	73.6	88.0	50.5	73.5	**61.7**	85.8	80.6	81.2	77.5	44.3	73.2	**66.7**	81.1	65.8
SSD500	**73.1**	**84.9**	**82.6**	**74.4**	**55.8**	**50.0**	**80.3**	**78.9**	88.8	**53.7**	76.8	59.4	**87.6**	**83.7**	**82.6**	**81.4**	**47.2**	**75.5**	65.6	**84.3**	**68.1**

Experimental Results

MS COCO Database:

- A total of 300k images

Test-dev results:

Method	data	Average Precision		
		0.5	0.75	0.5:0.95
Fast R-CNN [6]	train	35.9	-	19.7
Faster R-CNN [2]	train	42.1	-	21.5
Faster R-CNN [2]	trainval	42.7	-	21.9
ION [21]	train	42.0	23.0	23.0
SSD300	trainval35k	38.0	20.5	20.8
SSD500	trainval35k	43.7	24.7	24.4

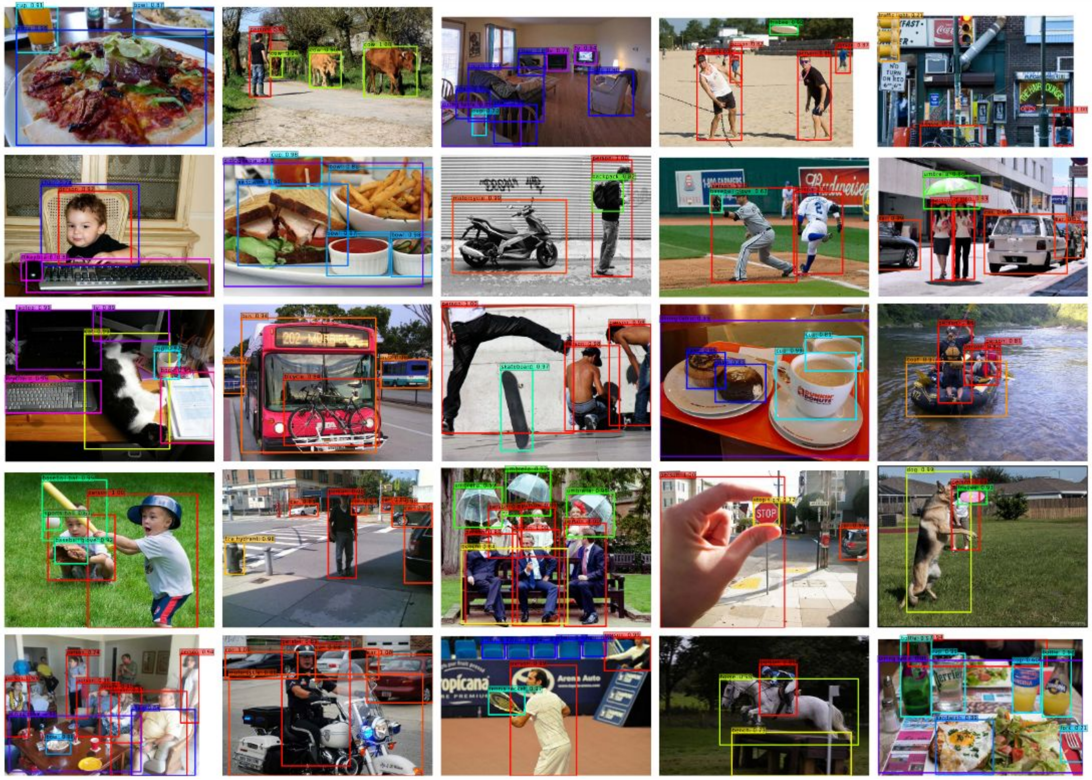
Experimental Results

Inference time :

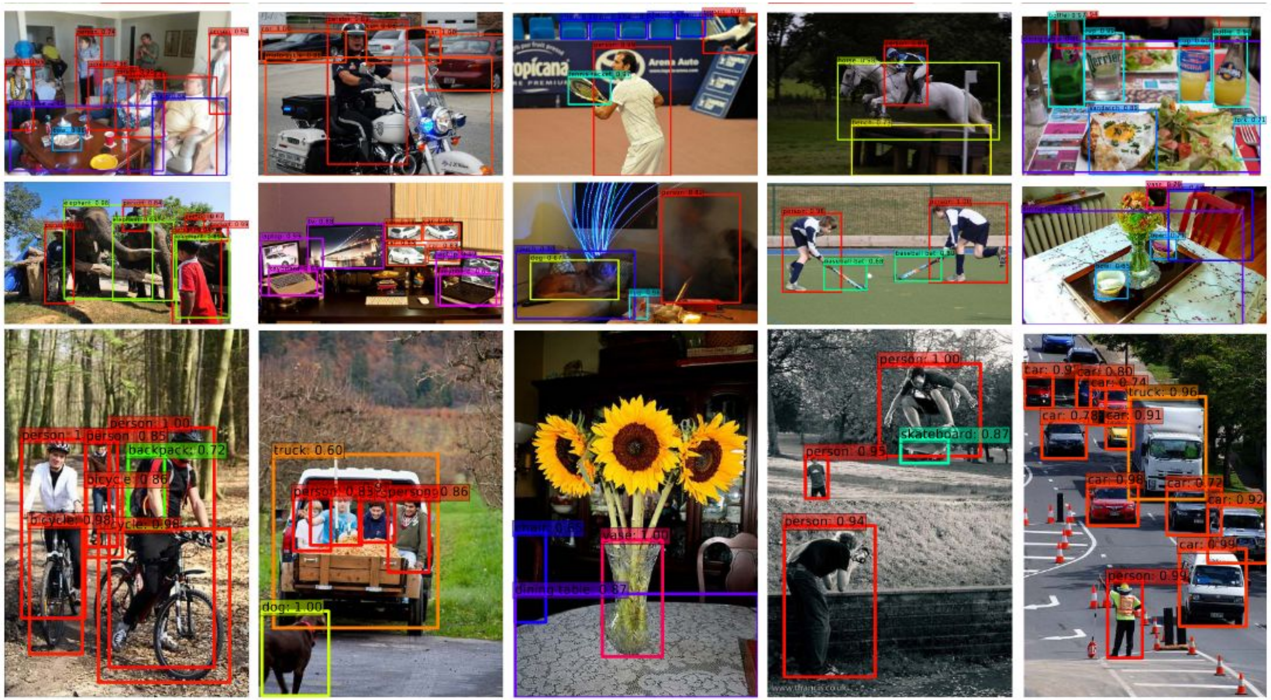
Non-maximum suppression has to be efficient because of the large number of boxes generated

Method	*mAP*	**FPS**	**# Boxes**
Faster R-CNN [2](VGG16)	73.2	7	300
Faster R-CNN [2](ZF)	62.1	17	300
YOLO [5]	63.4	45	98
Fast YOLO [5]	52.7	155	98
SSD300	72.1	58	7308
SSD500	**75.1**	23	20097

Visualizations



Visualizations



5.

Conclusions

SSD: Single Shot MultiBox Detector

Conclusions

- **Single-shot object** detector for multiple categories
- One key feature is to use **multiple convolutional maps** to deal with different scales
- **More default bounding boxes**, the better results obtained
- Comparable accuracy to state-of-the-art object detectors, but **much faster**
- Future direction: use RNNs to detect and track objects in video

Thank you for your attention! Questions?