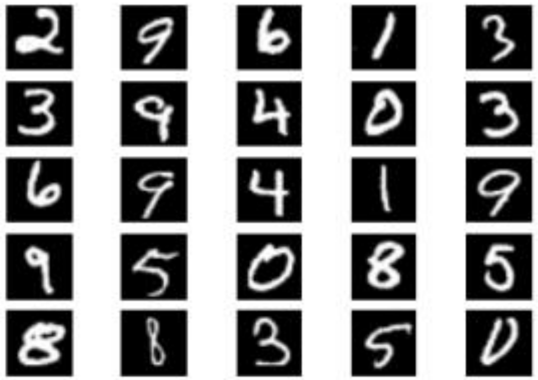
Object Detection Based on Deep Learning

Yurii Pashchenko

AI Ukraine 2016, Kharkiv, 2016

Image classification (mostly what you've seen)

- K classes
- Task: Assign the correct class label to the whole image

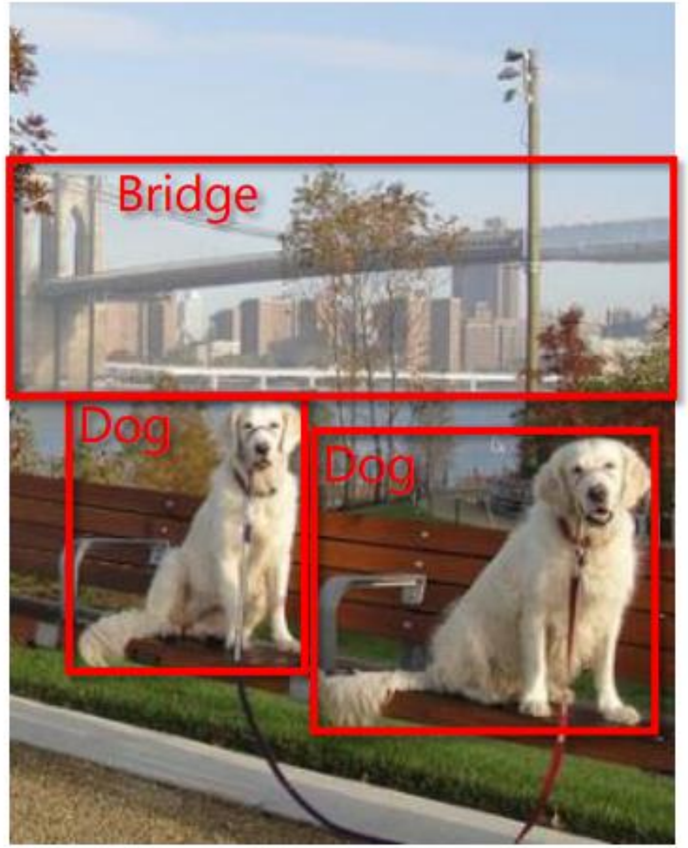


Digit classification (MNIST)

Object recognition (Caltech-101, ImageNet, etc.)

http://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-detection.pdf

Classification vs. Detection



http://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-detection.pdf

Benchmarks

- PASCAL VOC 2007/2012
- ILSVRC
- MS COCO

PASCAL VOC 2007/2012

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

20 classes:

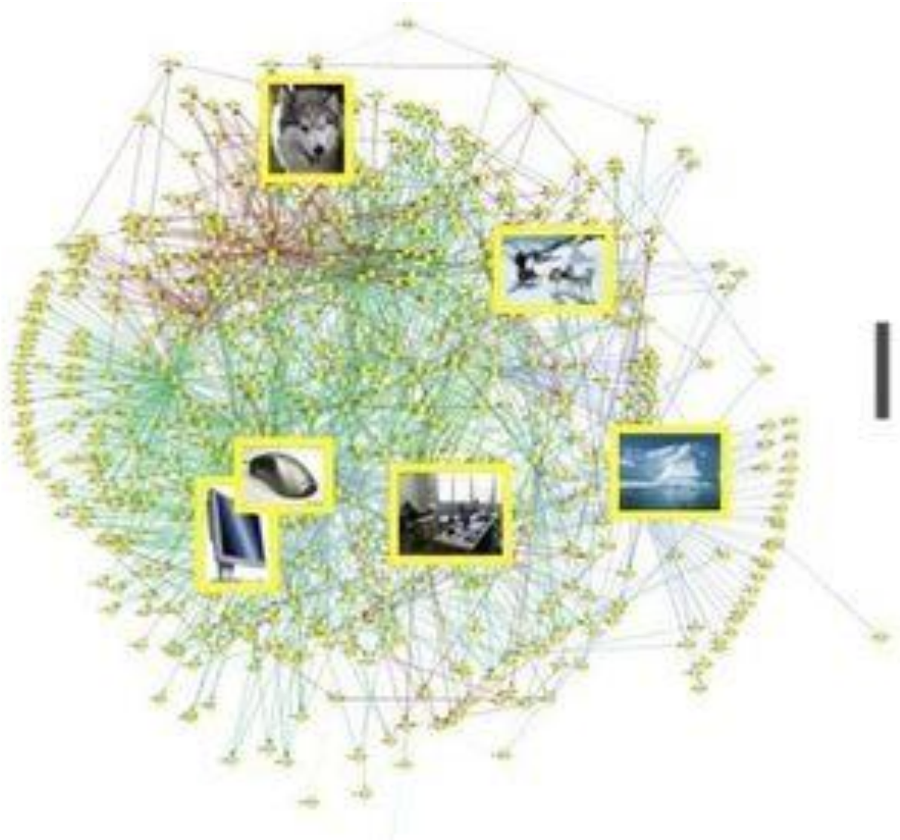
- *Person:* person
- *Animal:* bird, cat, cow, dog, horse, sheep
- *Vehicle:* aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor

Train/val size:

- VOC 2007 has 9,963 images containing 24,640 annotated objects.
- VOC 2012 has 11,530 images containing 27,450

http://host.robots.ox.ac.uk/pascal/VOC/

ILSVRC (DET)



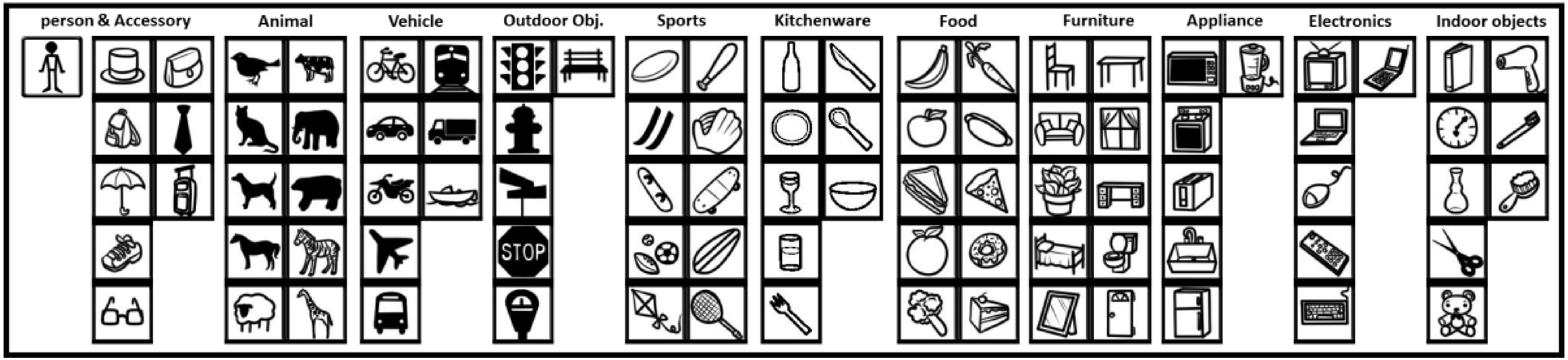
200 object classes 527,982 images

http://image-net.org/

MS COCO

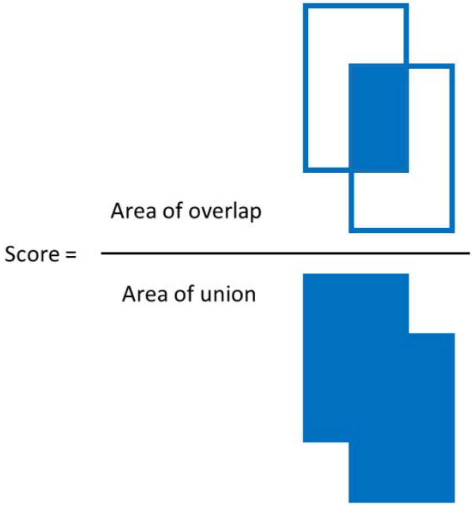
- 91 categories (80 available)
- 123,287 images, 886,284 instances

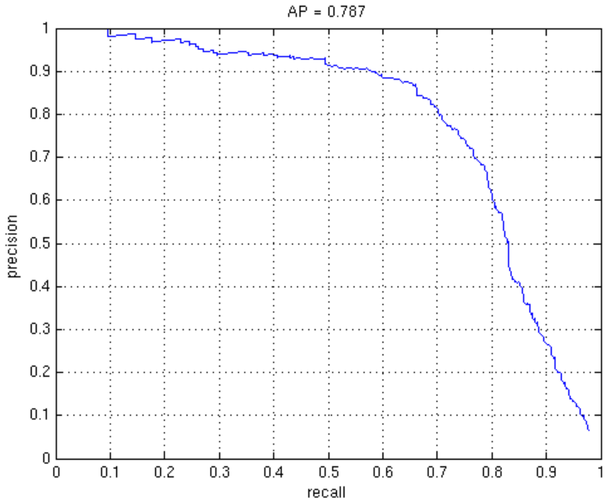
COCO
Common Objects in Context



http://*mscoco.org/*

Evaluation





- We use a metric called “mean average precision” (mAP)
- Compute average precision (AP) separately for each class, then average over classes A detection is a true positive if it has IoU with a ground-truth box greater than some threshold (usually 0.5) (mAP@0.5)
- Combine all detections from all test images to draw a precision / recall curve for each class; AP is area under the curve

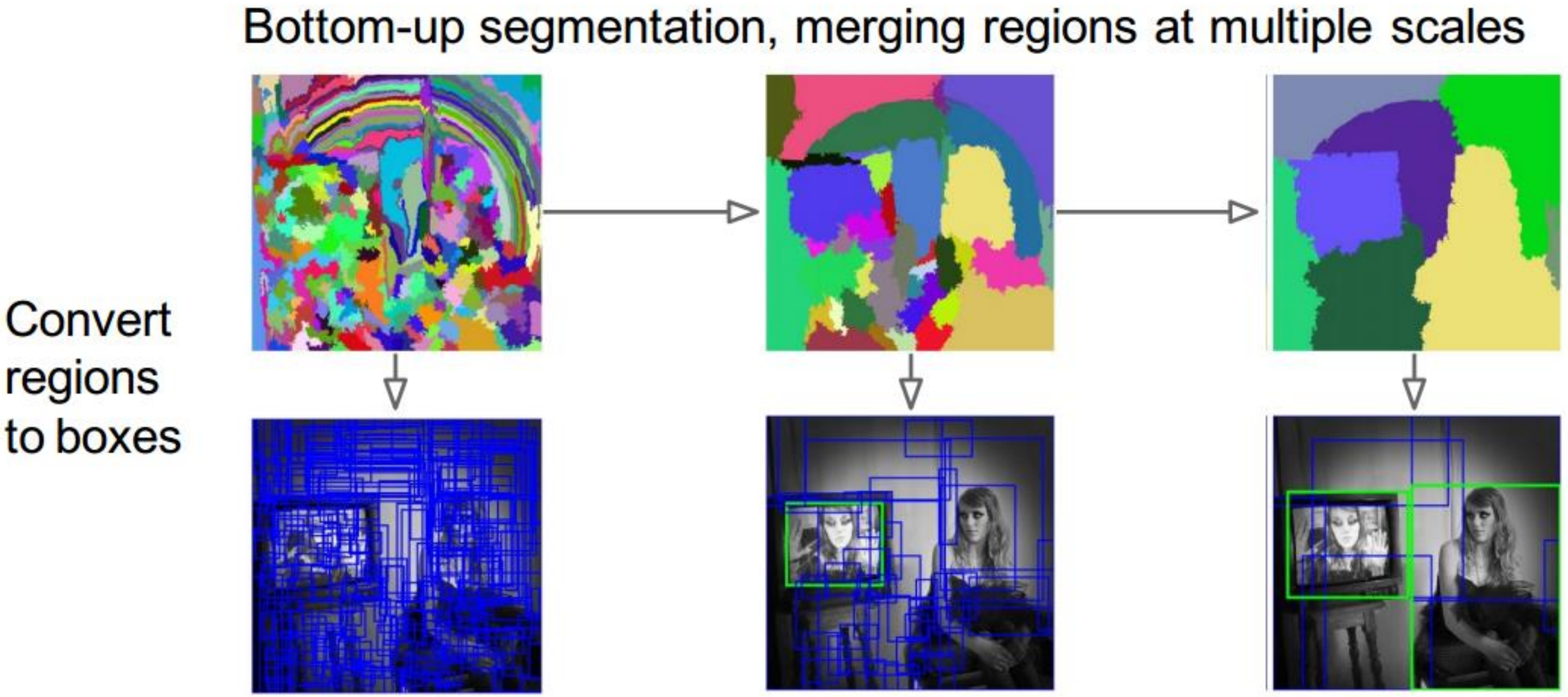
Detection as Classification

- **Problem**: Need to test many positions and scales, and use a computationally demanding classifier (CNN)

- **Solution**: Only look at a tiny subset of possible positions

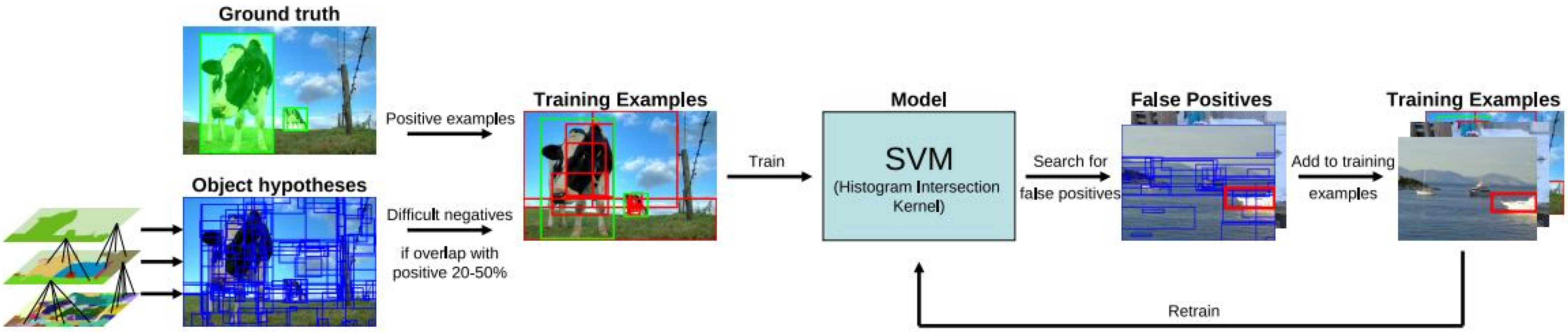
http://cs231n.github.io/neural-networks-3/ Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 48 1 Feb 2016

Region Proposals. Selective Search



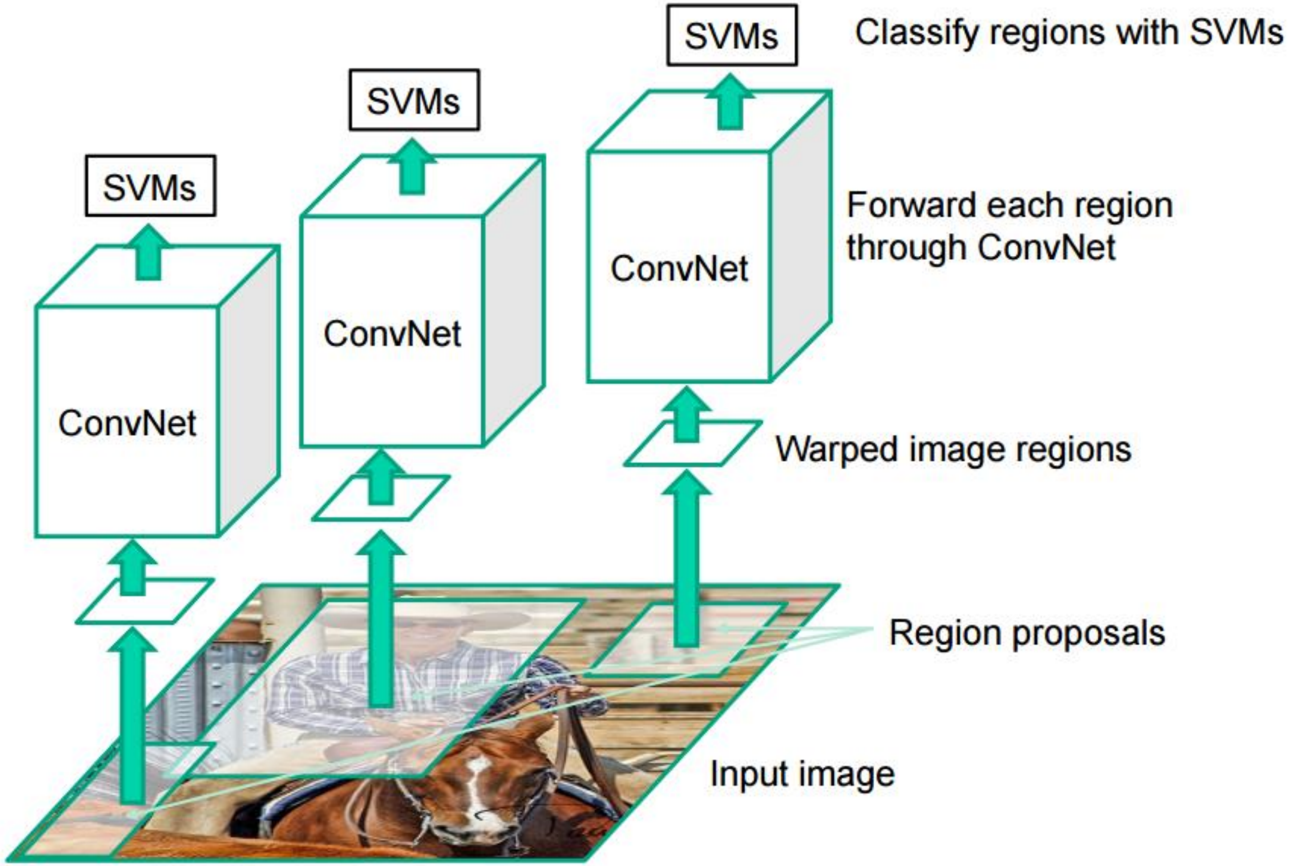
J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013

Selective search detection pipeline



Selective search + SIFT + bag-of-words + SVMs

R-CNN



- Regions: ~2000 Selective Search proposals
- Network: AlexNet pre-trained on ImageNet (1000 classes), fine-tuned on PASCAL (21 classes)
- Final detector: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- Bounding box regression to refine box locations
- Performance: mAP of 53.7% on PASCAL 2010 (vs. 35.1% for Selective Search and 33.4% for DPM).

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014

R-CNN pros and cons

- Pros
  - Accurate!
  - Any deep architecture can immediately be “plugged in”
- Cons
  - Ad hoc training objectives
    - Fine-tune network with softmax classifier (log loss)
    - Train post-hoc linear SVMs (hinge loss)
    - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
  - 2000 convnet passes per image
- Inference (detection) is slow (47s / image with VGG16)

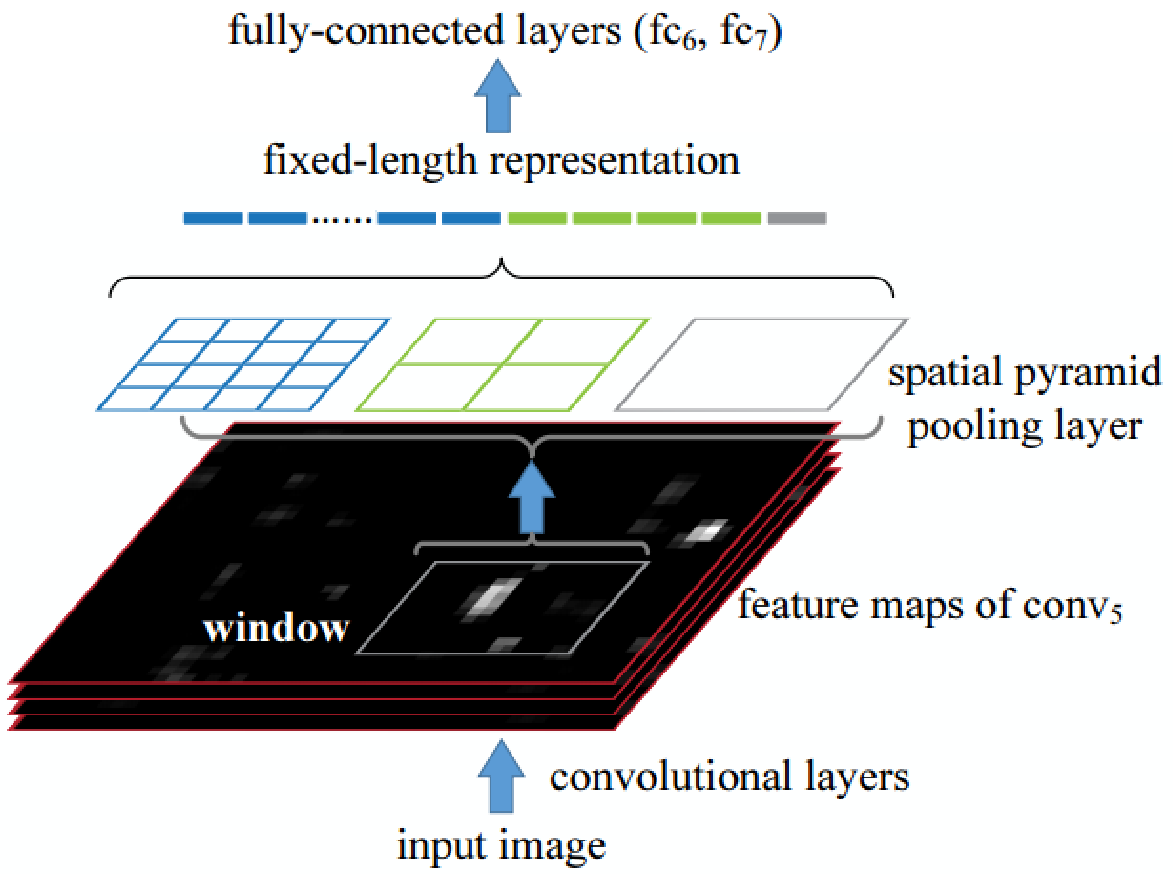
Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014

Spatial Pyramid Pooling Layer

In each candidate window, used a 4-level spatial pyramid:

- 1×1
- 2×2
- 3×3
- 6×6

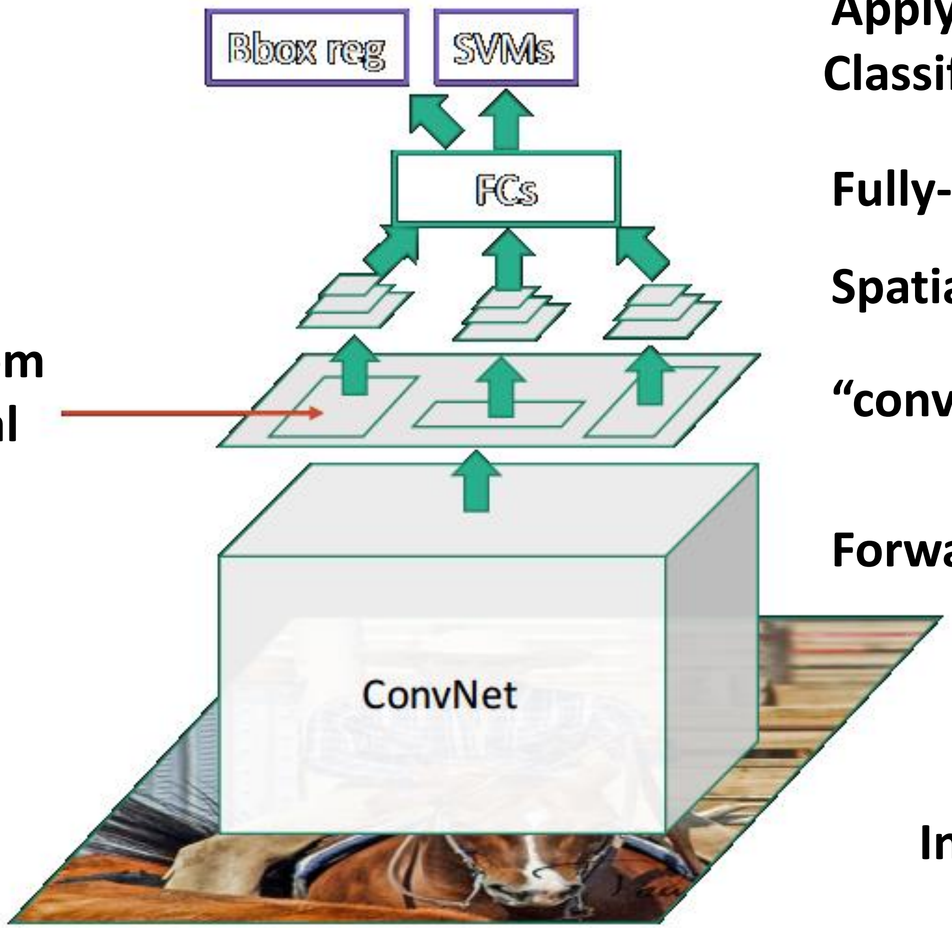
Totally 50 bins to pool the features. This generates a 12,800- d (256×50) representation for each window



K. Grauman and T. Darrell, *"The pyramid match kernel: Discriminative classification with sets of image features,"* in ICCV, 2005.

S. Lazebnik, C. Schmid, and J. Ponce, *"Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,"* in CVPR, 2006.

SPP-net



ROIs from proposal method

Apply b-box regressors
Classify regions with SVMs

Fully-connected layers

Spatial Pyramid Pooling layer

“conv5” feature map of image

Forward whole image through ConvNet

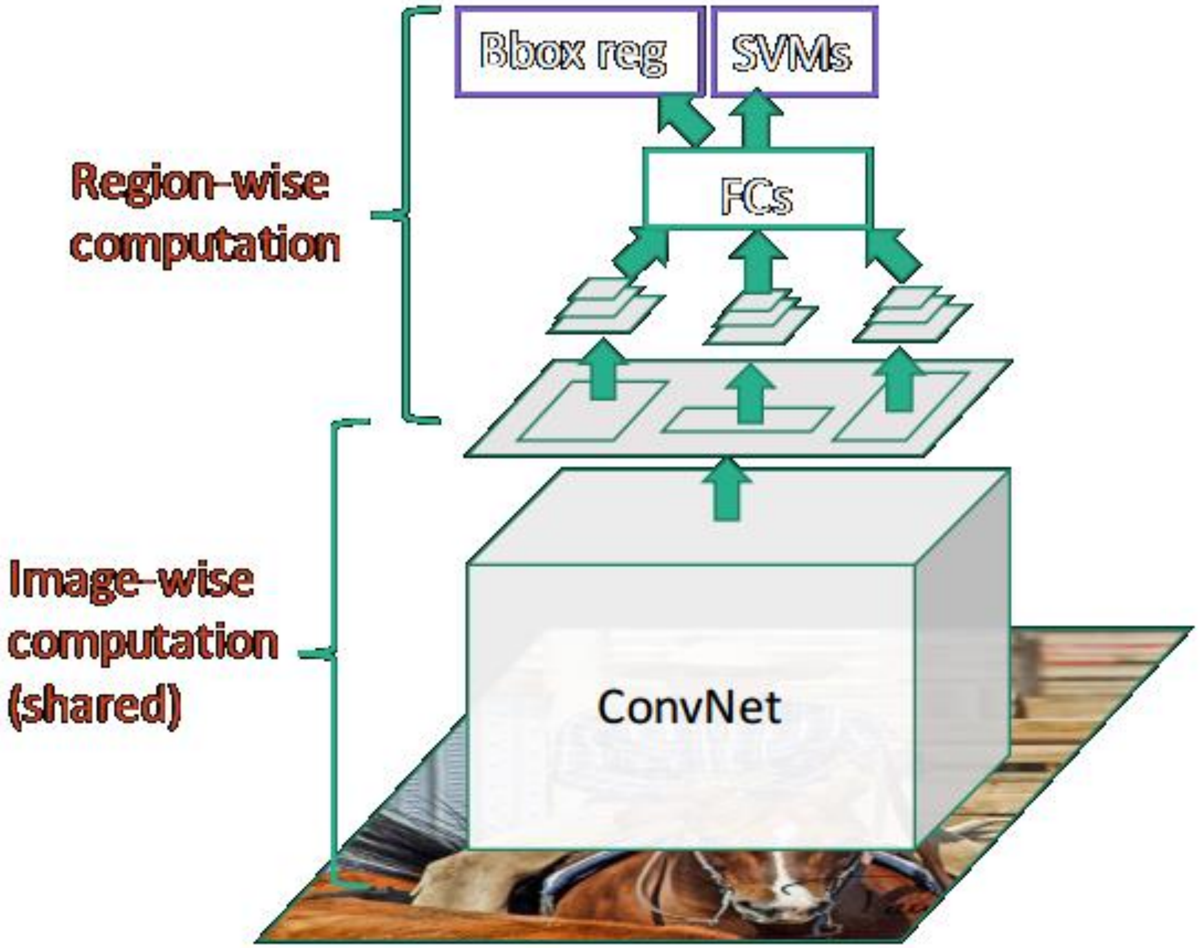
Input image

He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014

What's good about SPP-net?

Pascal VOC 2007 results

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (ZF-5)
$ftfc_7$	54.5	55.2	55.1
$ftfc_7$ bb	58.0	**59.2**	**59.2**
conv time (GPU)	0.053s	0.293s	14.37s
fc time (GPU)	0.089s	0.089s	0.089s
total time (GPU)	0.142s	0.382s	14.46s
speedup (*vs.* RCNN)	**102**×	**38**×	-

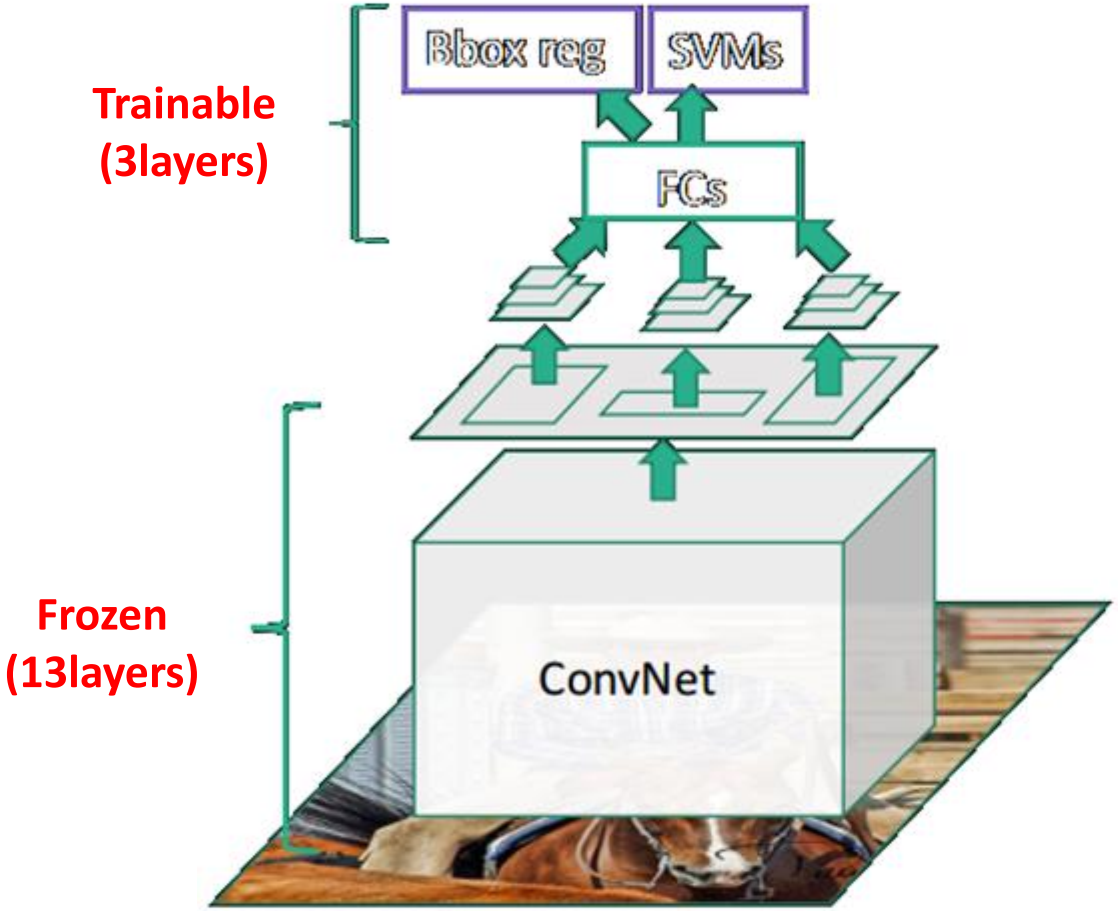


its realy faster...

He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014

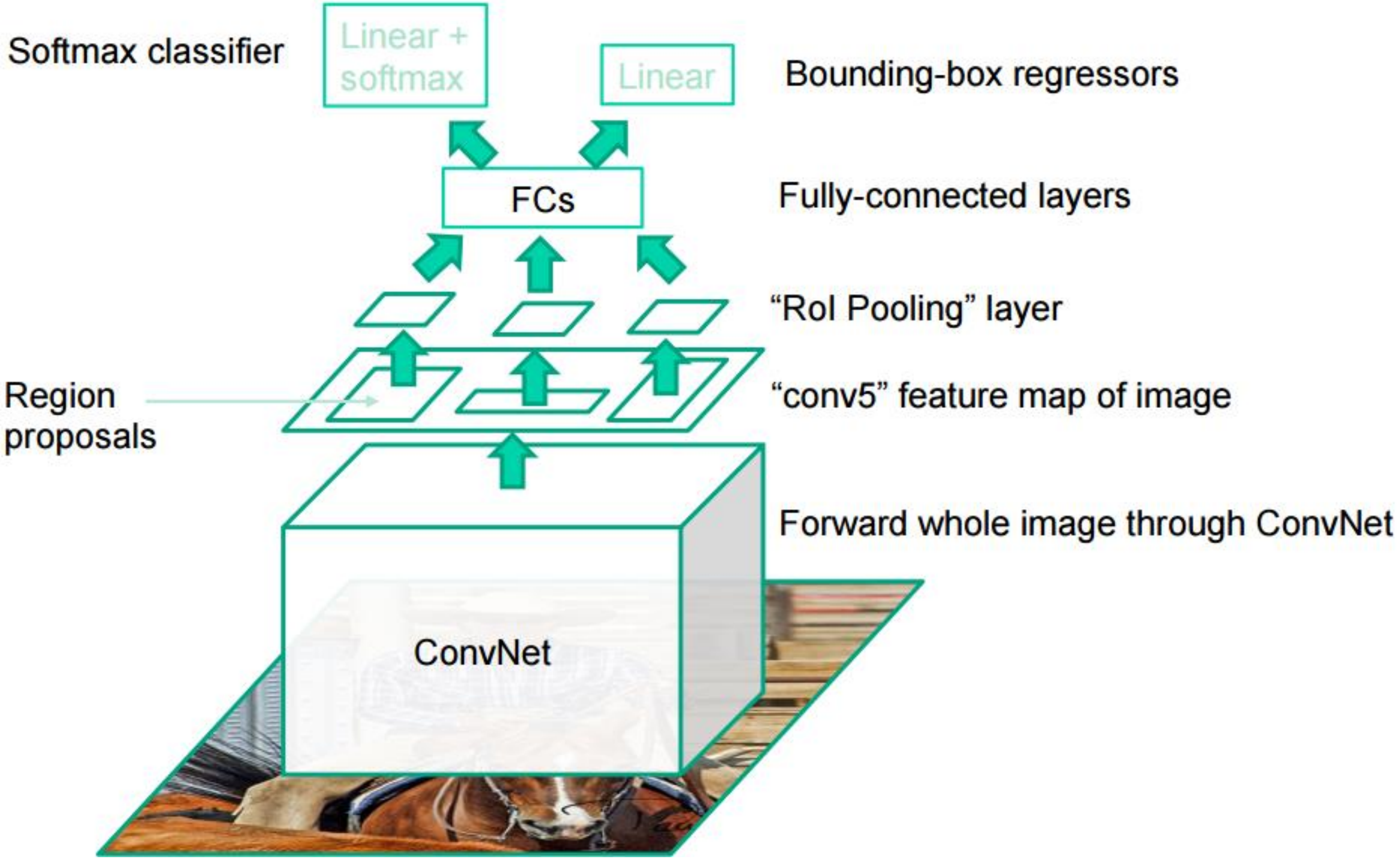
What's wrong with SPP'net?

- Inherits the rest of R-CNN's problems
- **Introduces a new problem**: cannot update parameters below SPP layer during training



He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014

Fast R-CNN



- Fast test time, like SPP-net
- One network, trained in one stage
- Higher mean average precision than slow R-CNN and SPP-net

R. Girshick. Fast R-CNN. arXiv:1504.08083, 2015

Fast R-CNN Results

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**
Better!	mAP (VOC 2007)	66.0	**66.9**

Using VGG-16 CNN on Pascal VOC 2007 dataset

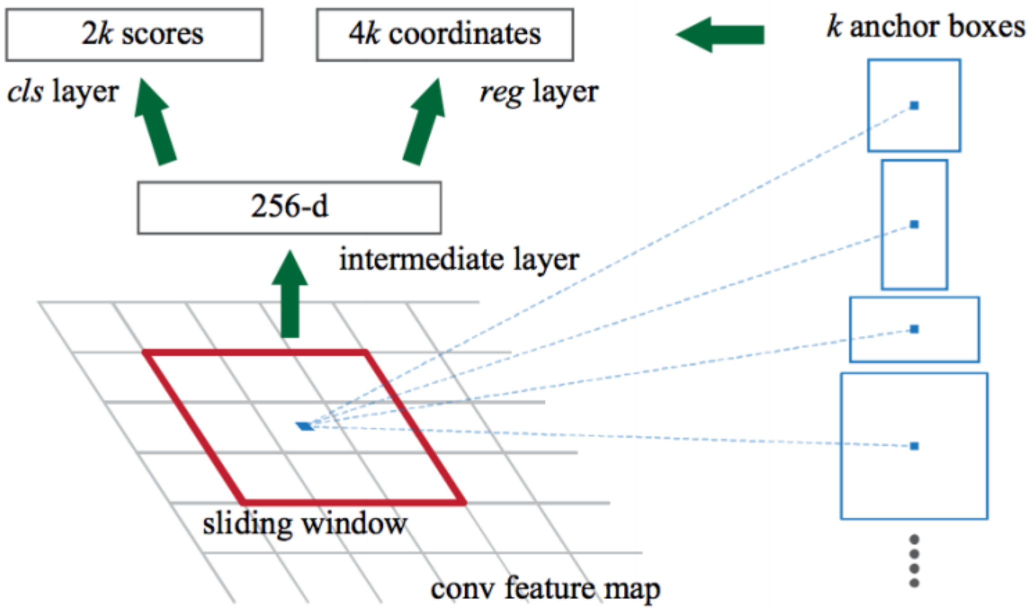
Fast R-CNN Problem

Test-time speeds don’t include region proposals

	R-CNN	Fast R-CNN
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Region proposal network

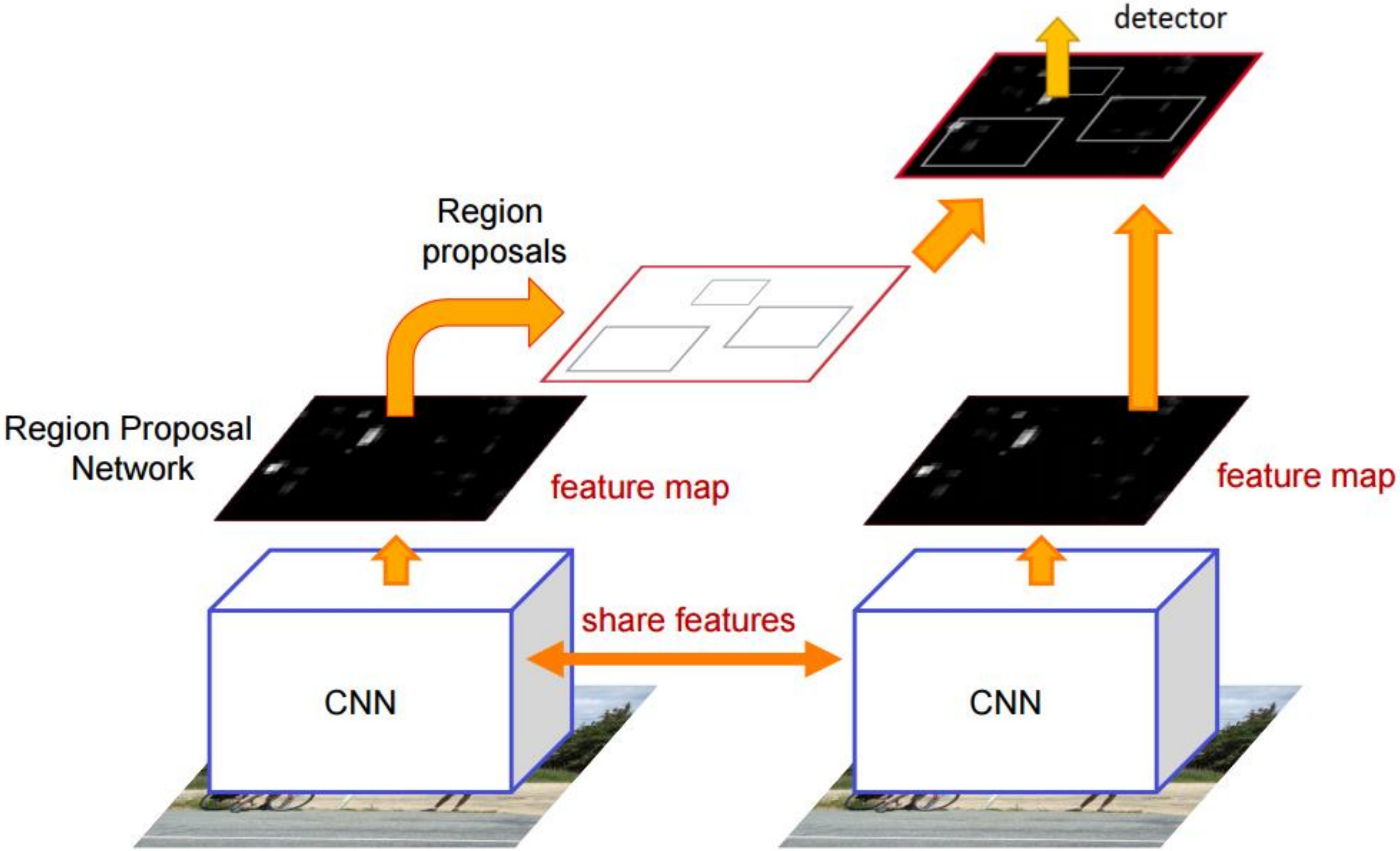
- Slide a small window over the conv5 layer
  - Predict object/no object
  - Regress bounding box coordinates
  - Box regression is with reference to *anchors* (3 scales x 3 aspect ratios)



~ 10 ms per image

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

Faster R-CNN



S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

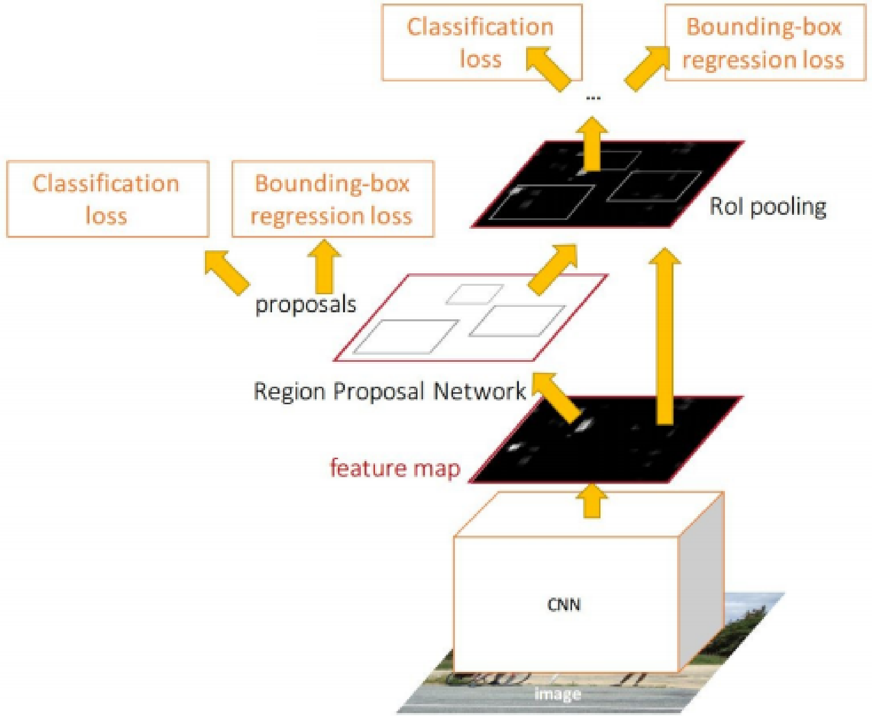
Faster R-CNN Training

In the paper: Ugly pipeline

- Use alternating optimization to train RPN, then Fast R-CNN with RPN proposals, etc.
- More complex than it has to be

Since publication: Joint training!
One network, four losses

- RPN classification (anchor good / bad)
- RPN regression (anchor -> proposal)
- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal -> box)



Faster R-CNN Results

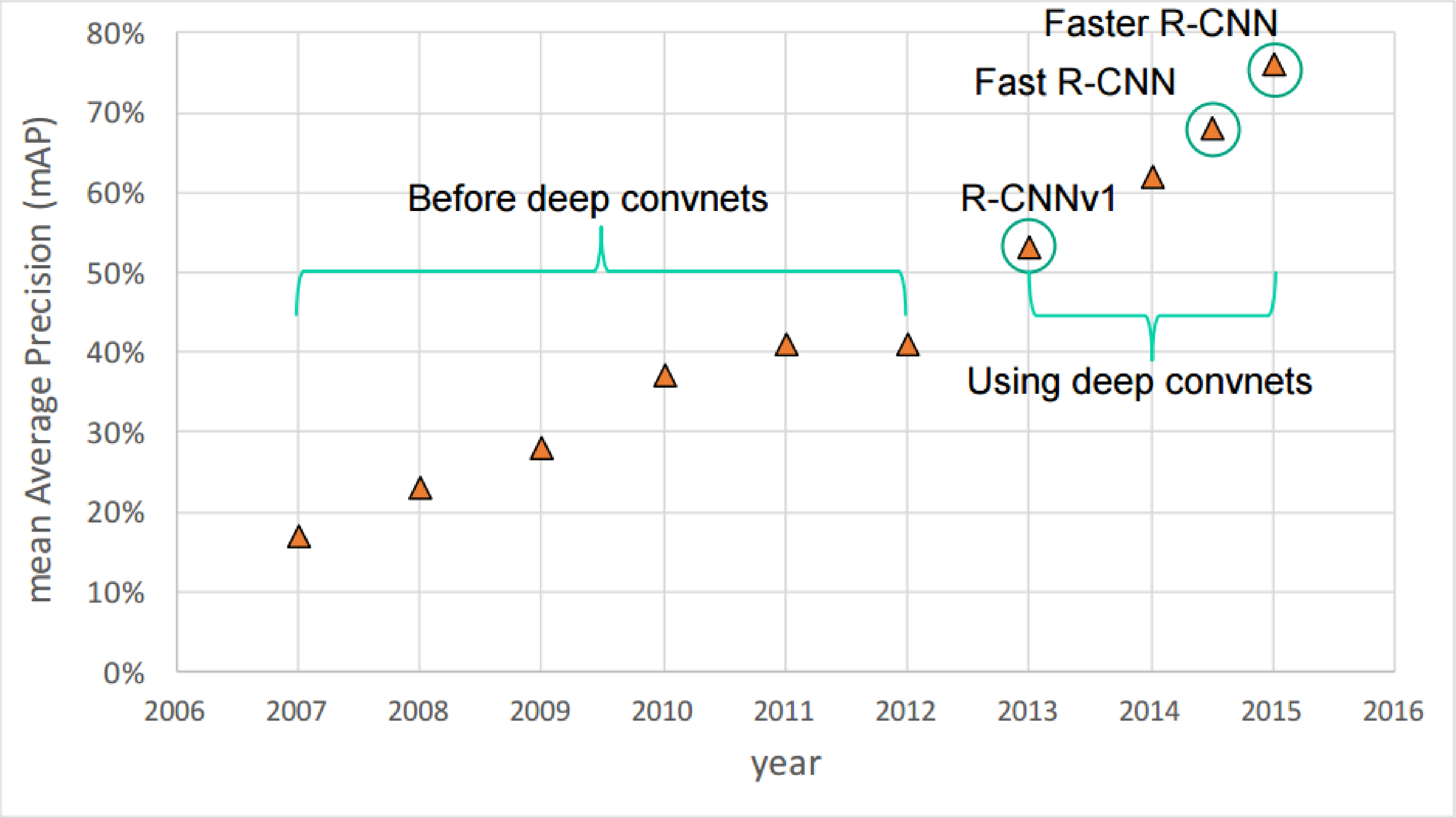
	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Faster R-CNN Results MS COCO

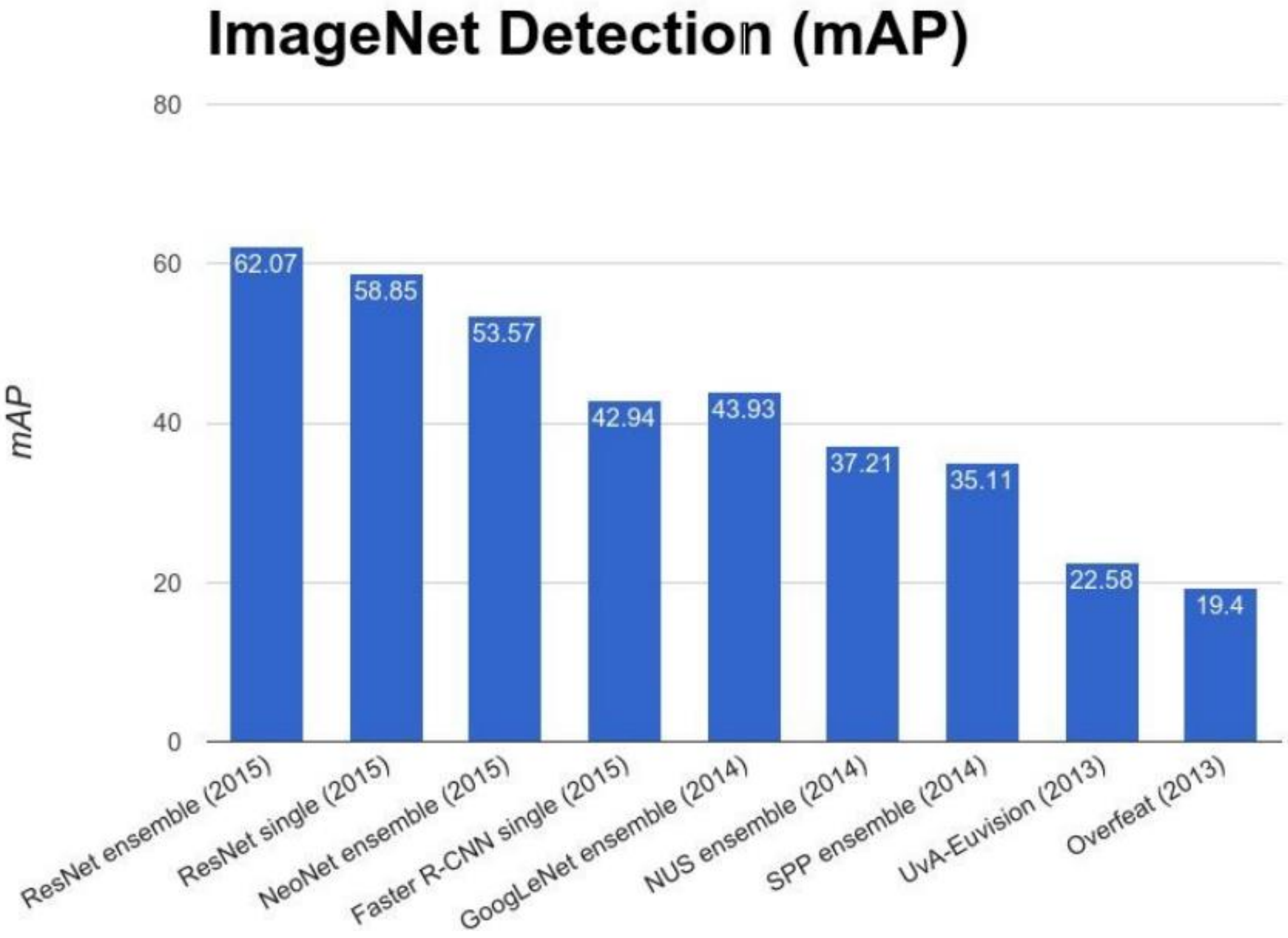
ResNet 101 + Faster R-CNN

training data	COCO train		COCO trainval	
test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**

Object detection progress PASCAL VOC 2007



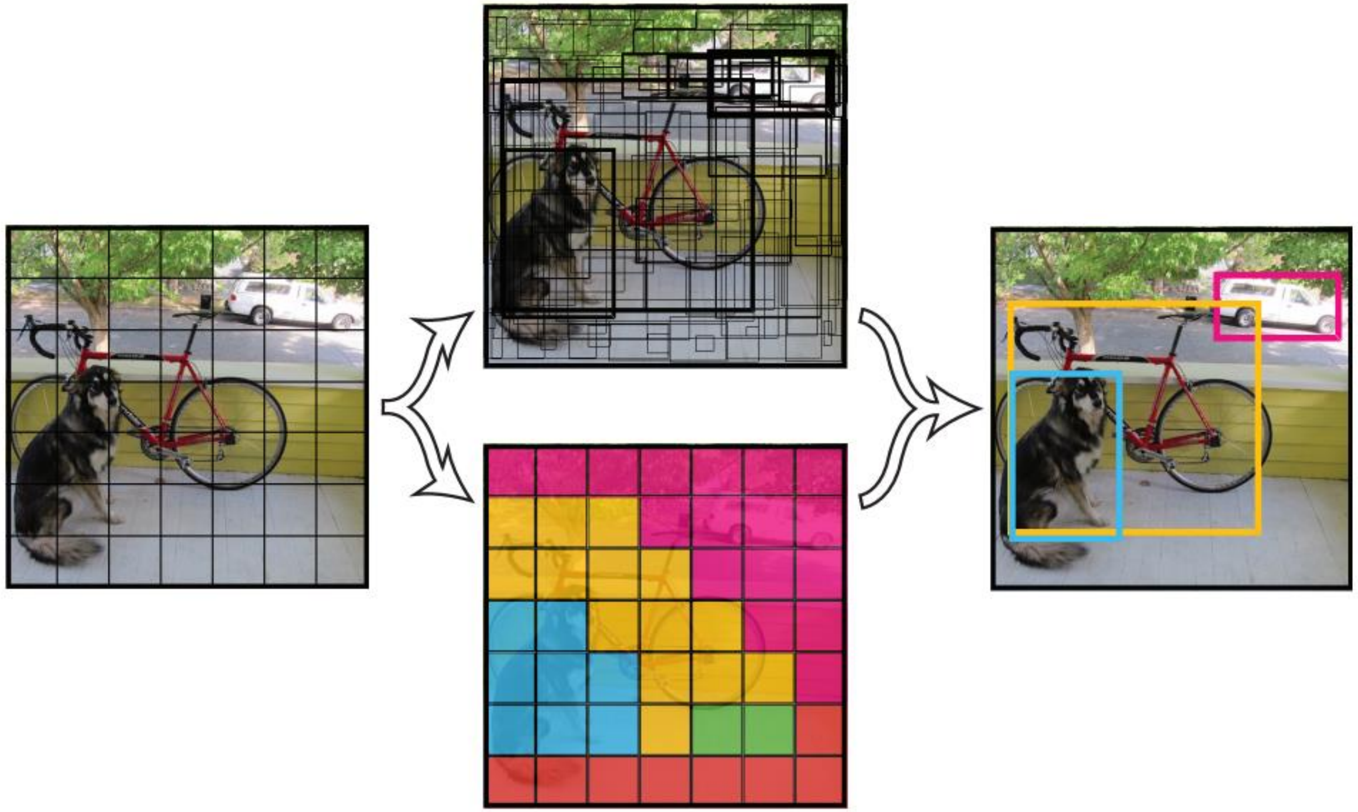
ImageNet Detection 2013 - 2015



Next trends

- **Fully convolutional detection networks**
  - You Only Look Once (YOLO)
  - Single Shot Multibox Detector (SSD)

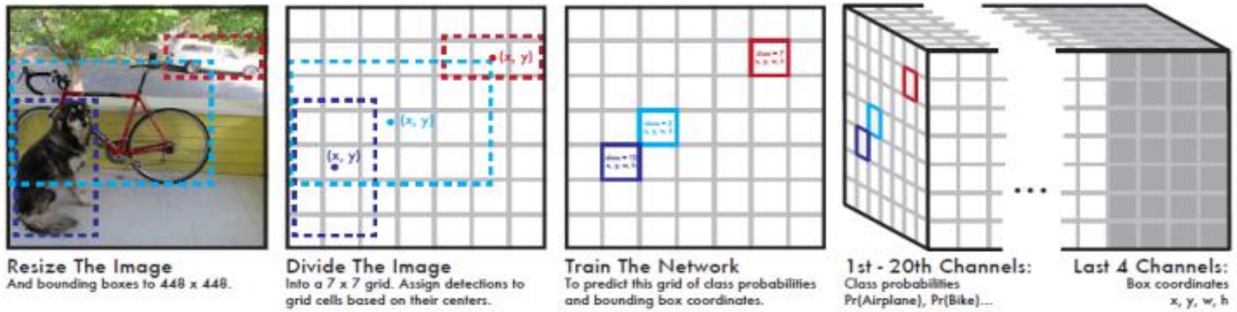
You Only Look Once YOLO



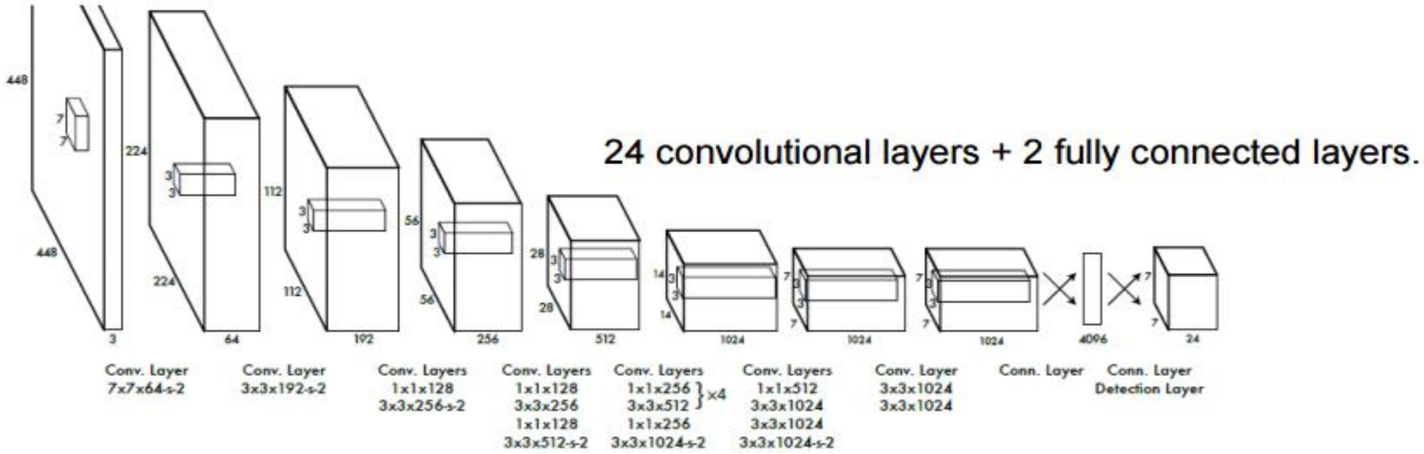
- Divide image into SxS grid
  If the center of an object falls into a grid cell, it will be the responsible for the object.
- Each grid cell predict:
  - B-boxes
  - Confidence scores
  - Class probability

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015

YOLO Design



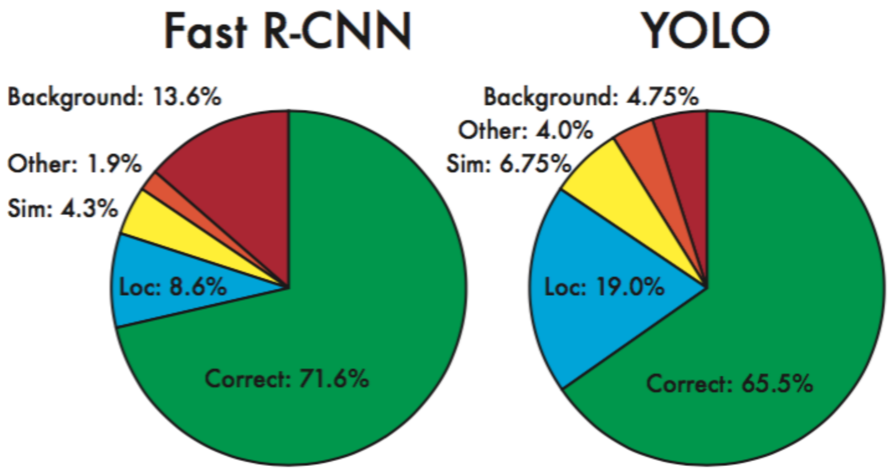
A regression problem to a 7724 tensor which encodes bounding boxes and class probabilities for all objects in the image.



J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015

Fast R-CNN & YOLO

	mAP	Combined	Gain
Fast R-CNN	-	71.8	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**



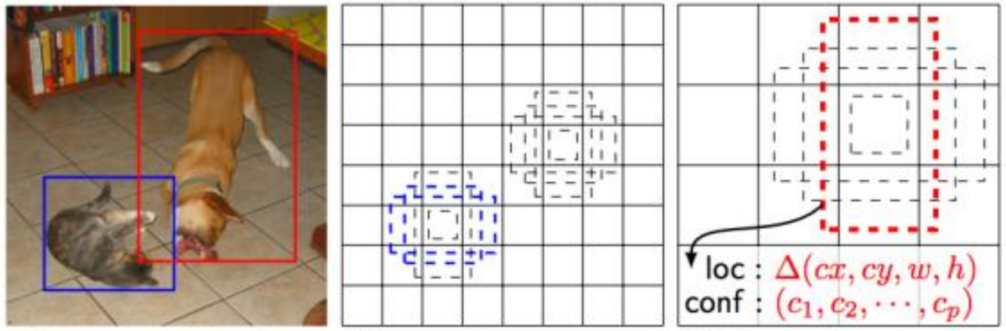
Speed > 45 fps

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015

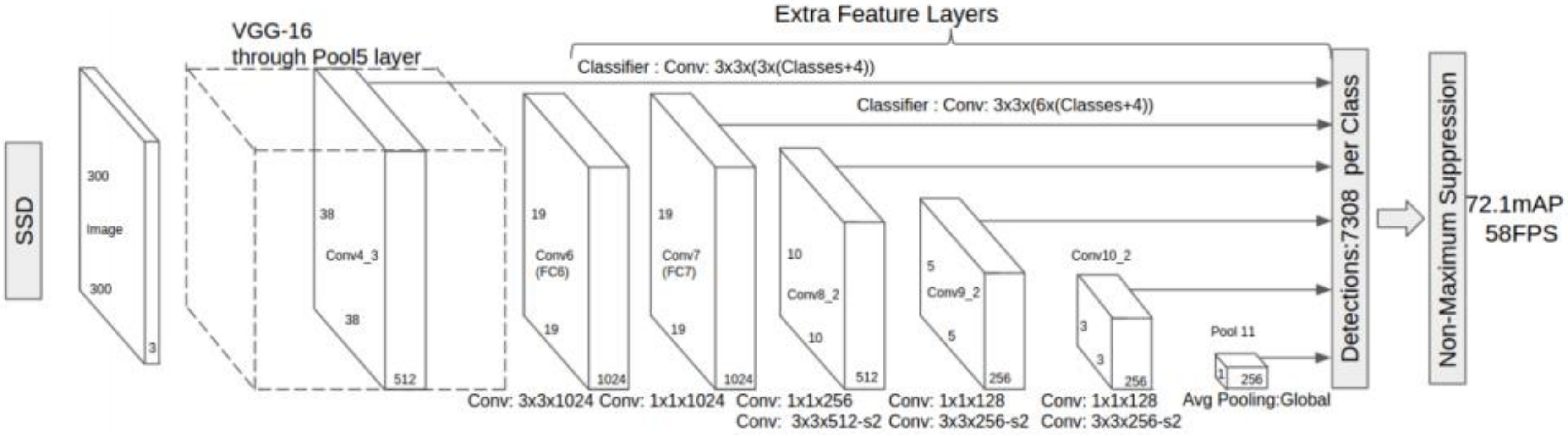
Limitations

- Struggle with small objects
- Struggle with different aspects and ratios of objects
- Loss function is an approximation
- Loss function threats errors in different boxes ratio at the same.

Single Shot MultiBox Detector (SSD)

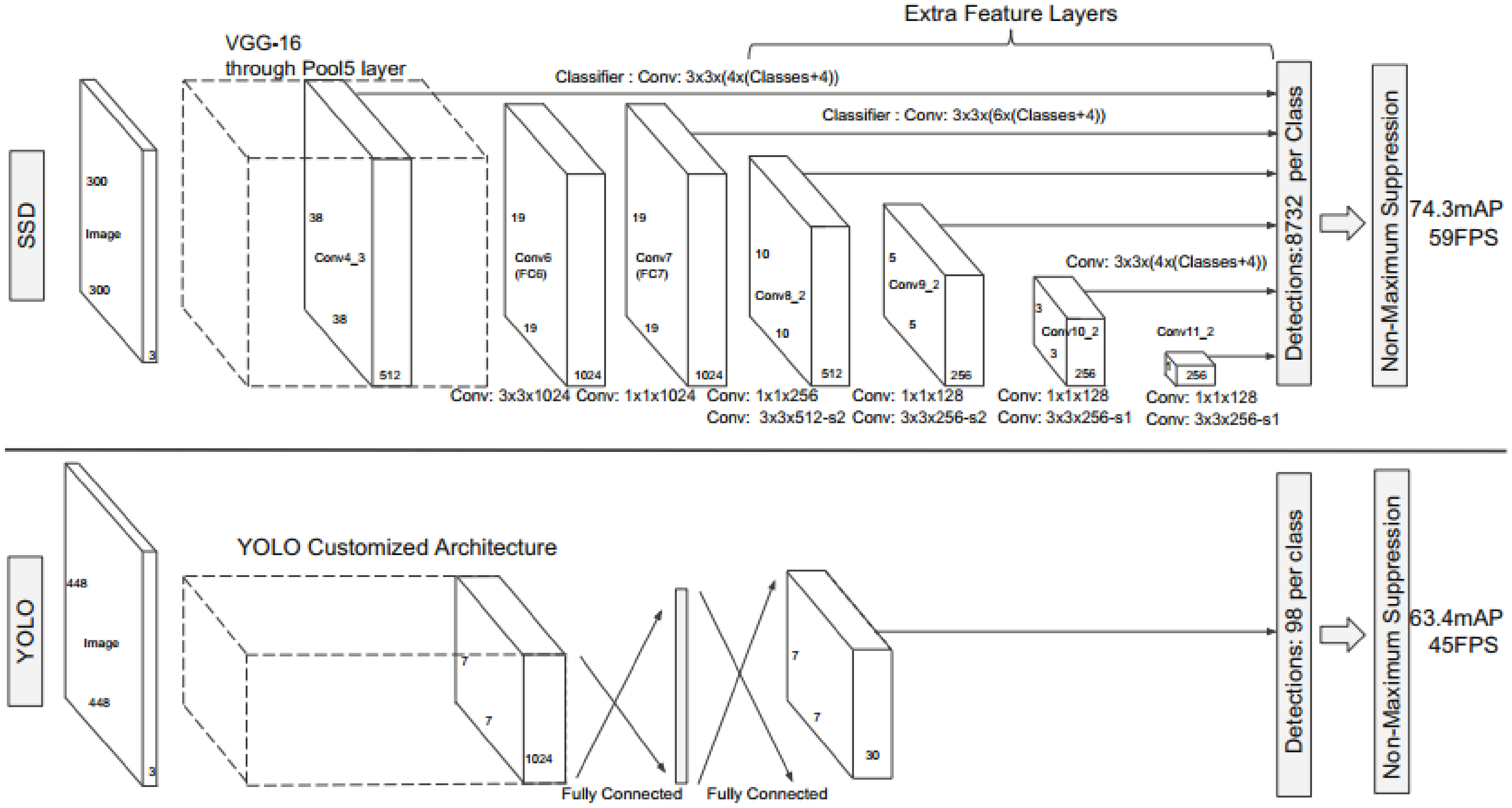


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map



W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015

SSD vs YOLO Architecture



W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015

SSD Results Pascal

PASCAL VOC 2007

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

PASCAL VOC 2012

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**

SSD Results on MS COCO

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

Sources

- R-CNN
  - Caffe + MATLAB : https://github.com/rbgirshick/rcnn
- Faster R-CNN
  - Caffe + MATLAB: https://github.com/ShaoqingRen/faster_rcnn
  - Caffe + Python: https://github.com/rbgirshick/py-faster-rcnn
  - Torch: https://github.com/andreaskoepf/faster-rcnn.torch
  - TensorFlow: https://github.com/smallcorgi/Faster-RCNN_TF
- YOLO
  - Darknet: https://github.com/pjreddie/darknet
  - TensorFlow: https://github.com/gliese581gg/YOLO_tensorflow
  - Caffe: https://github.com/xingwangsfu/caffe-yolo
- SSD
  - Caffe: https://github.com/weiliu89/caffe/tree/ssd

THANK YOU

Yurii Pashchenko
george.pashchenko@gmail.com