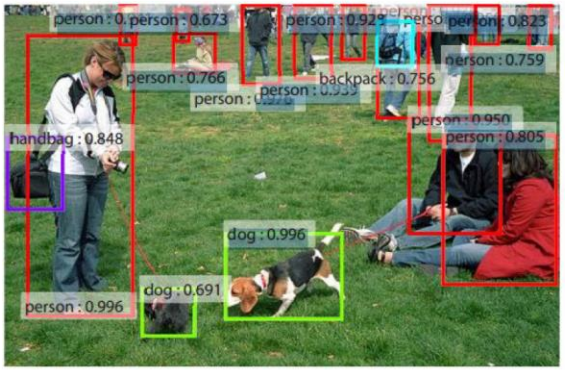
CENG 506 Deep Learning

Lecture 7 - Object Detection and Localization

Slides were prepared using the course material of
Stanford's CNN Course (CS231n by Fei-Fei, Johnson, Yeung)

Localization and Detection



Results from Faster R-CNN, Ren et al. 2015

Some Computer Vision Tasks

Classification

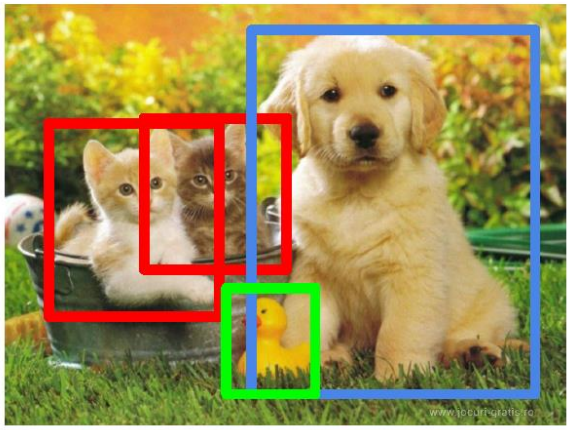
CAT

Classification + Localization

CAT

Single object

Object Detection



CAT, DOG, DUCK

Multiple objects

Note: We use the terms ‘localization’ and ‘detection’ in consistent with their use in ImageNet (ILSVRC) competition.

Some Computer Vision Tasks

Classification

Classification + Localization

Object Detection

Classification + Localization: Task

Classification: C classes

Input: Image

Output: Class label

Evaluation metric: Accuracy

Localization:

Input: Image

Output: Box in the image (x, y, w, h)

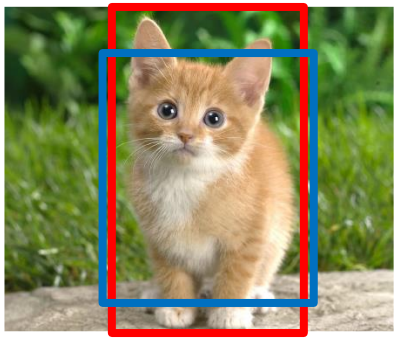
Evaluation metric: Intersection over Union

Classification + Localization: Do both

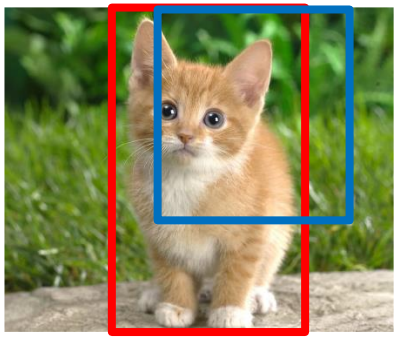
Our metric: Intersection over Union

This evaluation metric compares detected bounding box with the groundtruth box, and considers the detection as a true-positive if boxes' intersection/union ratio is higher than a threshold. Threshold is usually taken as 0.5.

$$\textbf{IoU} = \frac{\text{Box}_{\text{detected}} \cap \text{Box}_{\text{groundtruth}}}{\text{Box}_{\text{detected}} \cup \text{Box}_{\text{groundtruth}}}$$



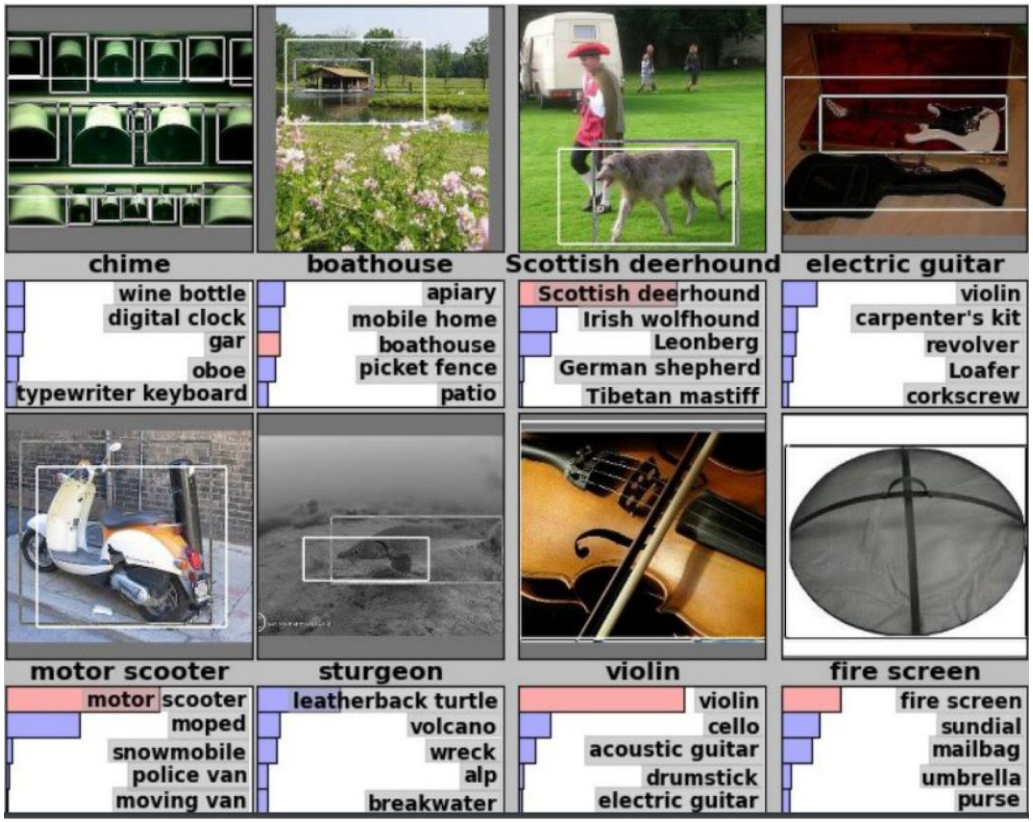
IoU=0.72



IoU=0.40

Classification + Localization: ImageNet

- 1000 classes
- Each image has 1 class, at least one bounding box
- ~800 train images / class
- Algorithm produces 5 (class, box) guesses
- Example is correct if at least one guess has correct class AND bounding box with at least 0.5 intersection over union (IoU)



Krizhevsky et. al. 2012

Idea #1: Localization as Regression

Input: image

Only one object,
simpler than detection

Neural Net

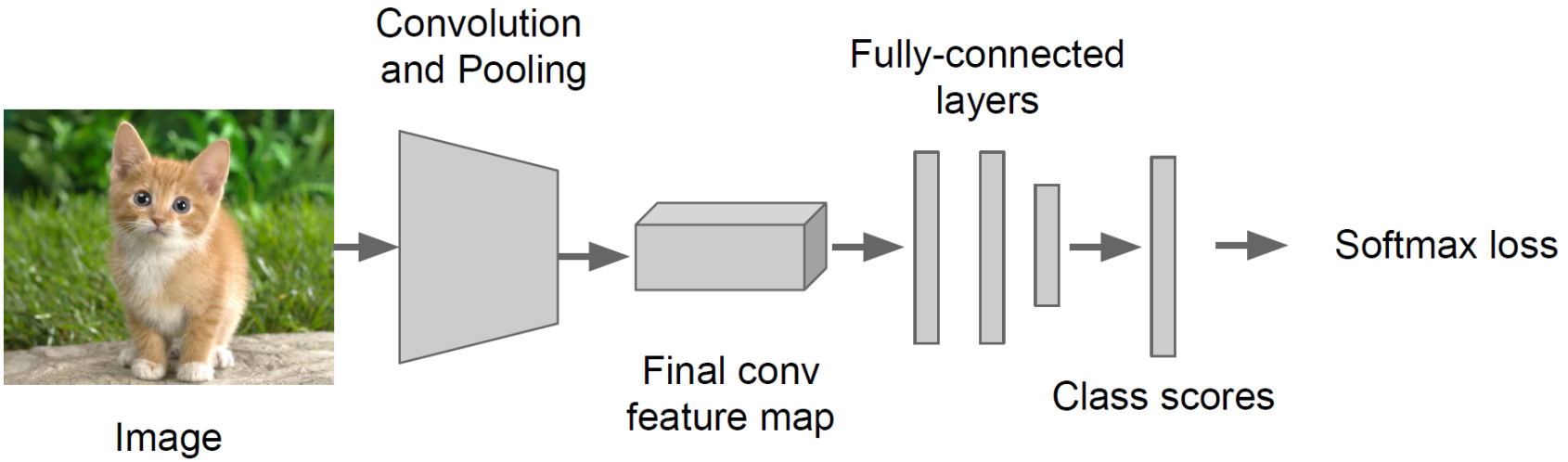
Output:
Box coordinates
(4 numbers)

Correct output:
box coordinates
(4 numbers)

Loss:
L2 distance

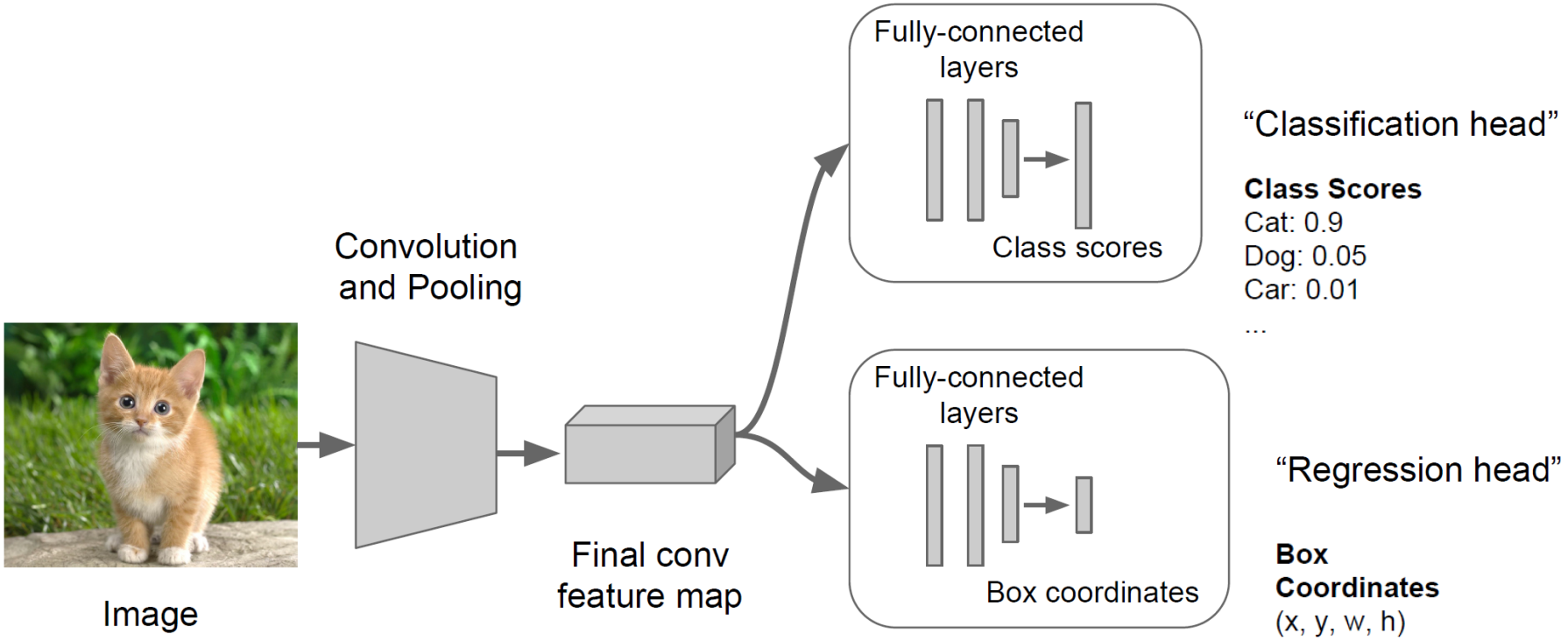
Simple Recipe for Classification+Localization

Step 1: Train (or download) a classification model (AlexNet, VGG, GoogLeNet)



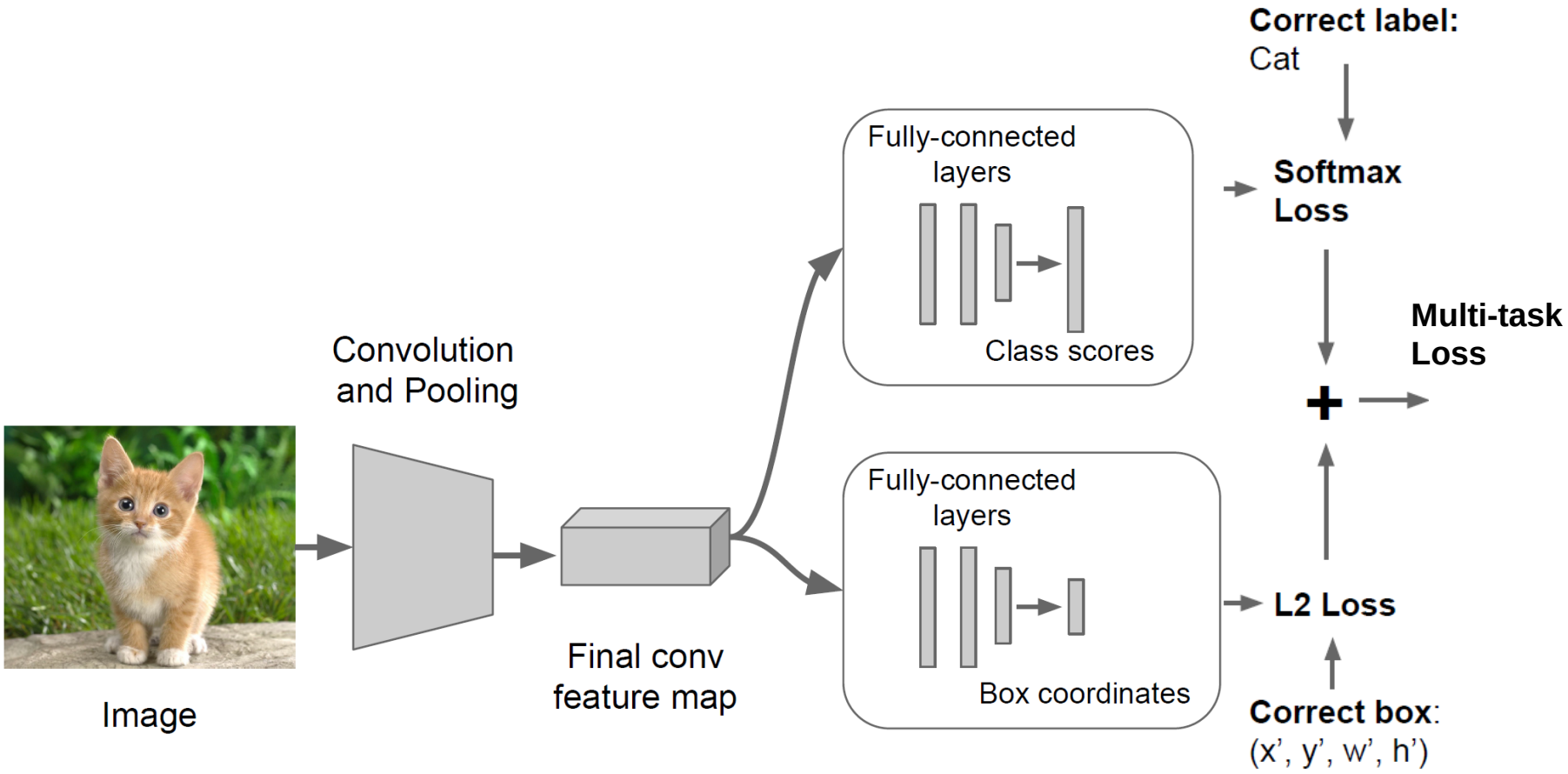
Simple Recipe for Classification+Localization

Step 2: Attach new fully-connected "regression head" to the network



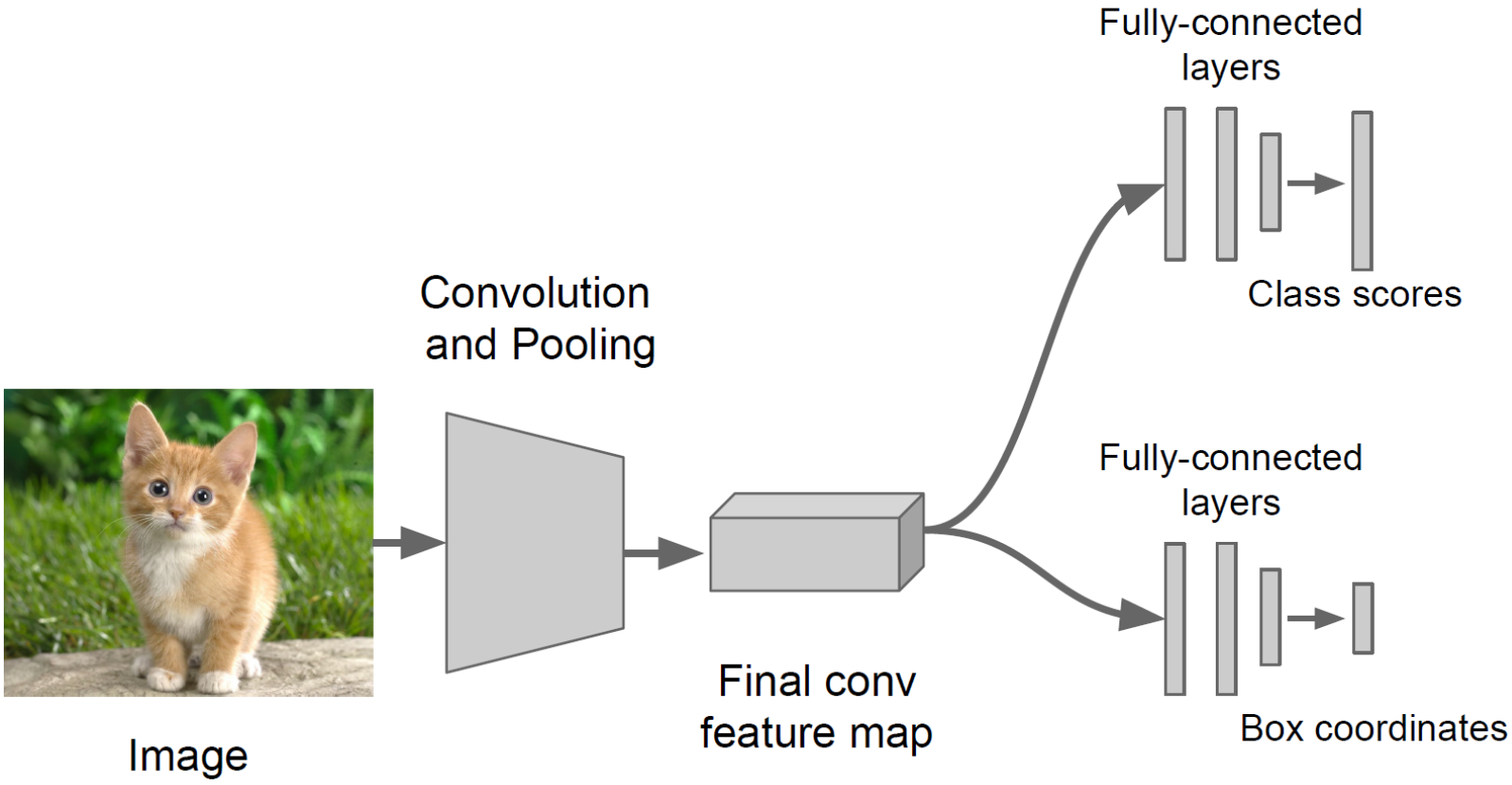
Simple Recipe for Classification+Localization

Step 3: Train the network with multi-task loss



Simple Recipe for Classification+Localization

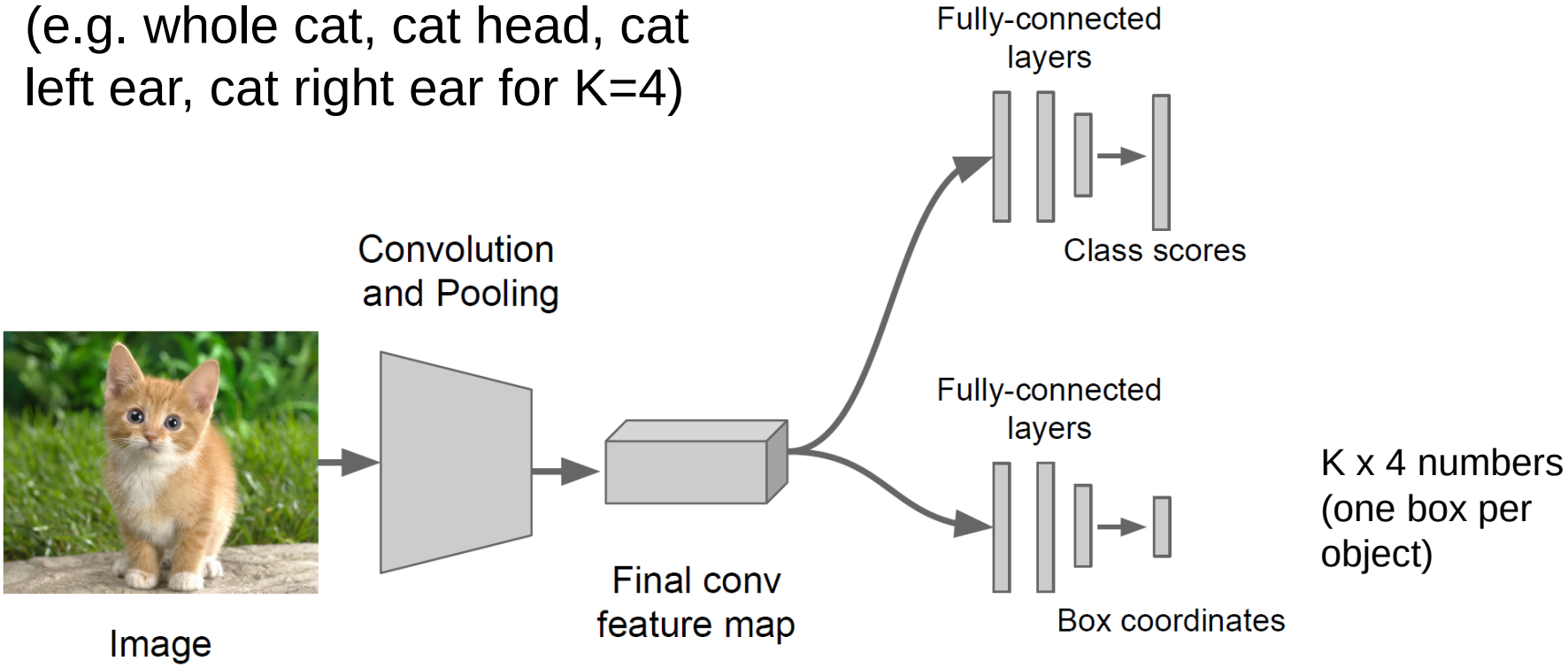
Step 4: At test time use both heads



Aside: Localizing multiple objects

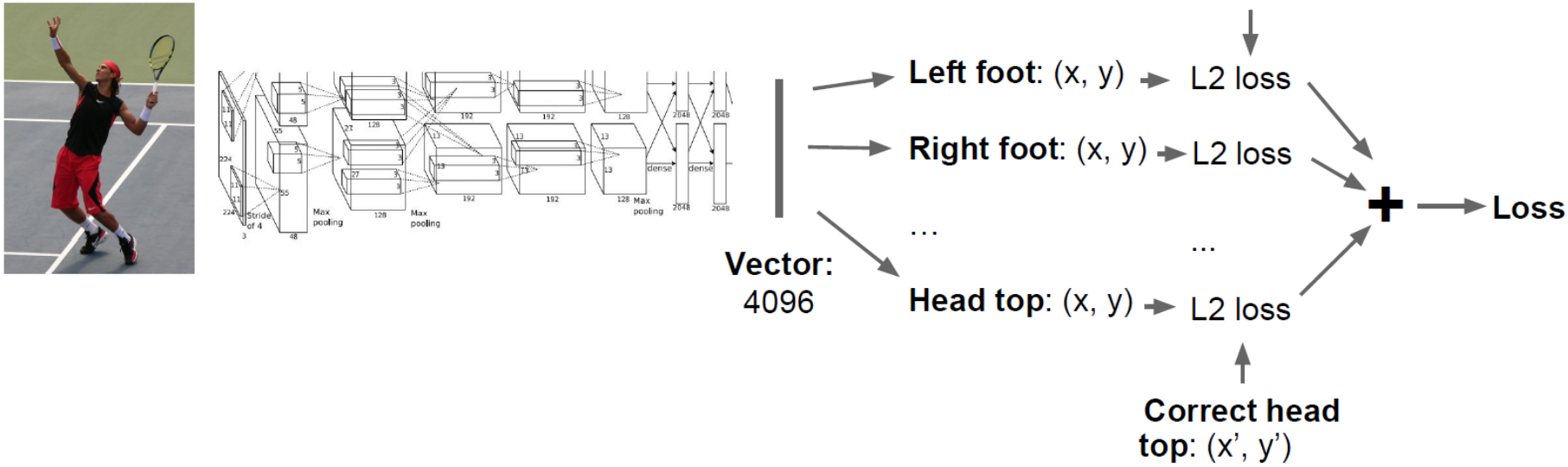
Want to localize **exactly** K objects in each image

(e.g. whole cat, cat head, cat left ear, cat right ear for K=4)



Aside: Human Pose Estimation

- Represent a person by K joints
- Regress (x, y) for each joint from last fully-connected layer



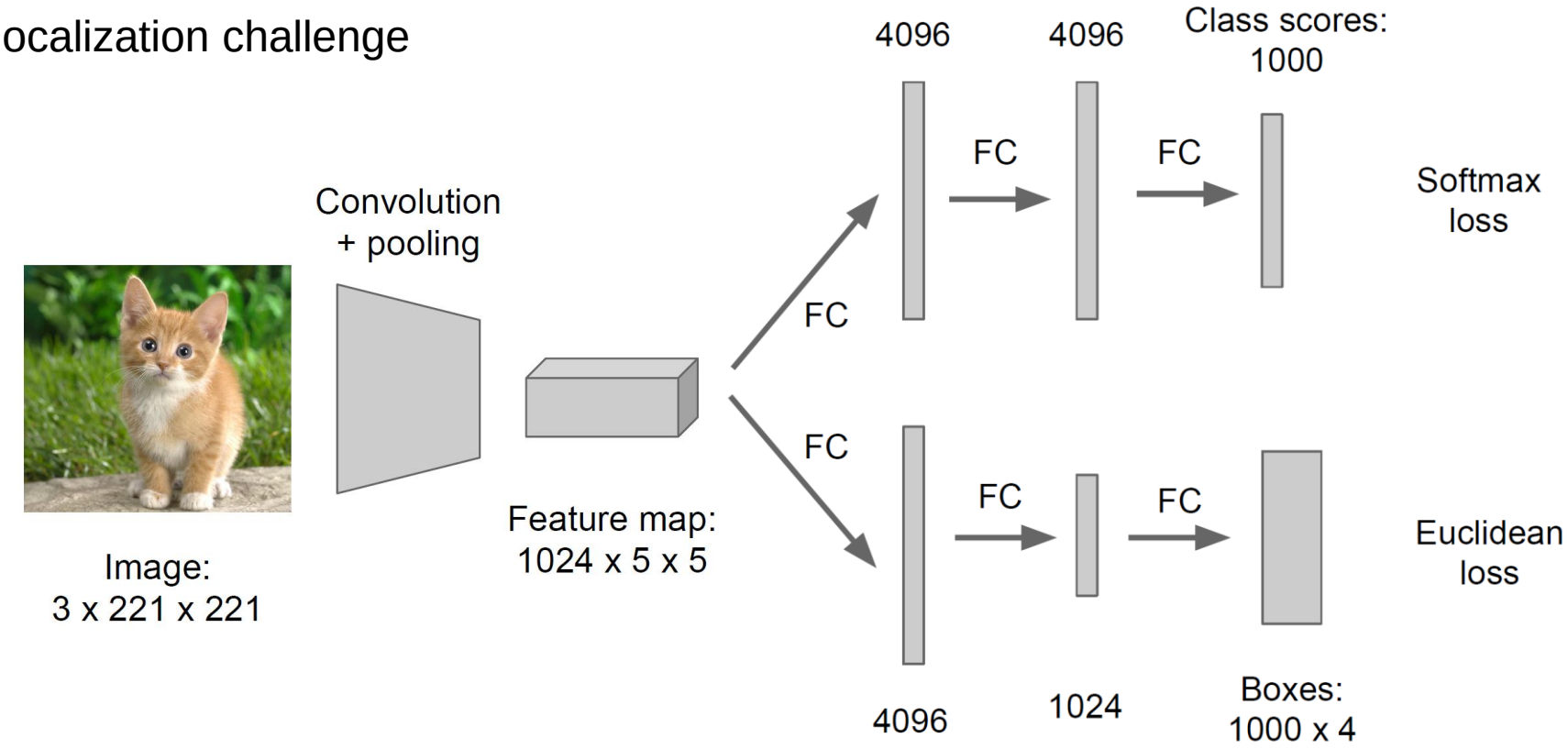
Toshev and Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

Idea #2: Sliding Window

- Run classification + regression network at multiple locations on a high resolution image
- Combine classifier and regressor predictions across all scales for final prediction

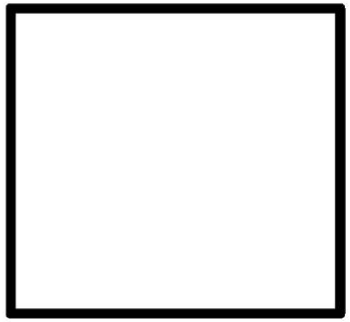
Sliding Window: Overfeat*

Winner of ILSVRC 2013
localization challenge



*Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

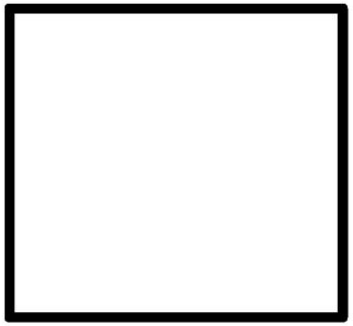
Sliding Window: Overfeat



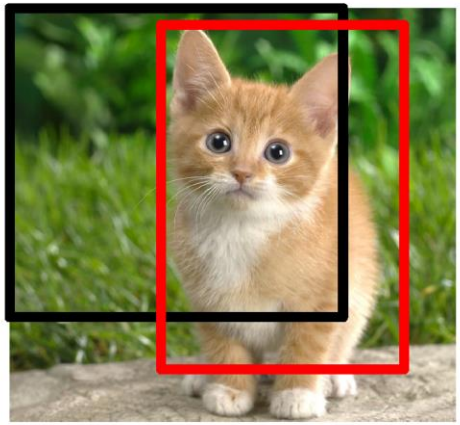
Network input:
3 x 221 x 221

Larger image:
3 x 257 x 257

Sliding Window: Overfeat



Network input:
3 x 221 x 221

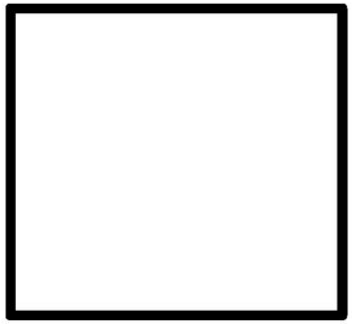


Larger image:
3 x 257 x 257

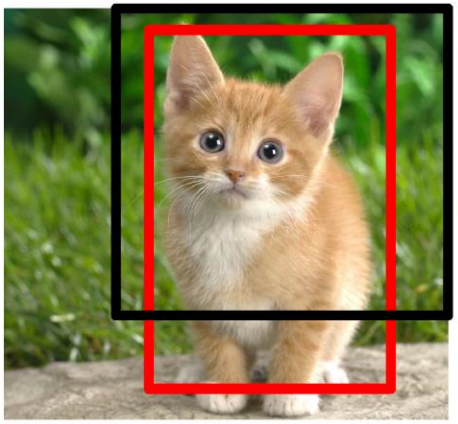
0.5	

Classification scores:
P(cat)

Sliding Window: Overfeat



Network input:
3 x 221 x 221

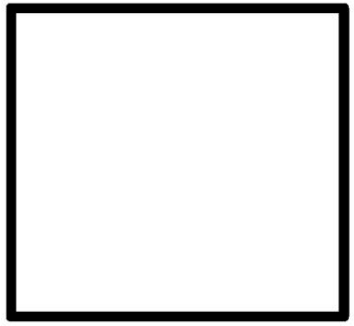


Larger image:
3 x 257 x 257

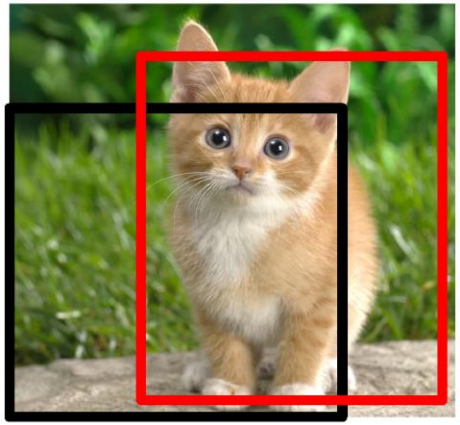
0.5	0.75

Classification scores:
P(cat)

Sliding Window: Overfeat



Network input:
3 x 221 x 221

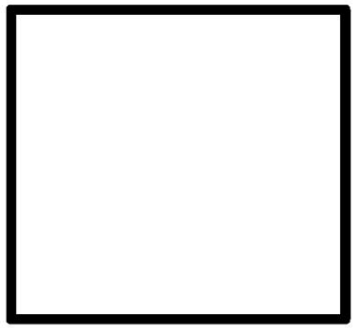


Larger image:
3 x 257 x 257

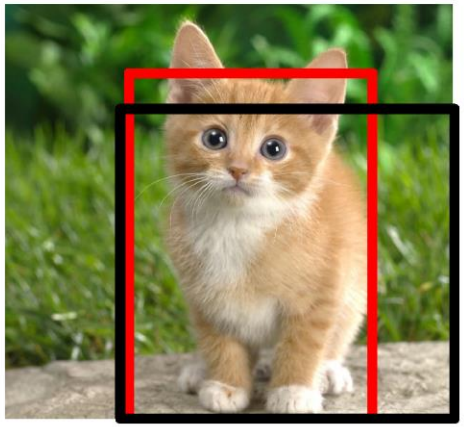
0.5	0.75
0.6	

Classification scores:
P(cat)

Sliding Window: Overfeat



Network input:
3 x 221 x 221

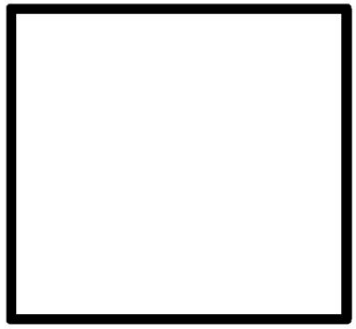


Larger image:
3 x 257 x 257

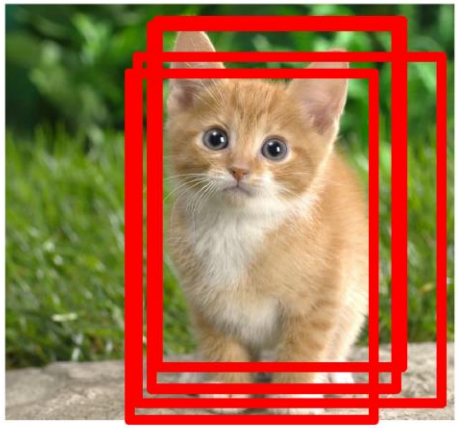
0.5	0.75
0.6	0.8

Classification scores:
P(cat)

Sliding Window: Overfeat



Network input:
3 x 221 x 221



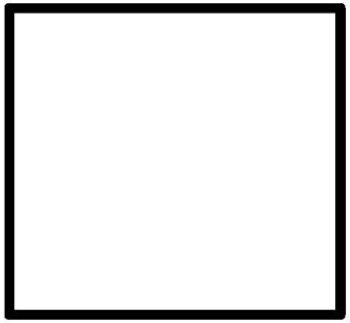
Larger image:
3 x 257 x 257

0.5	0.75
0.6	0.8

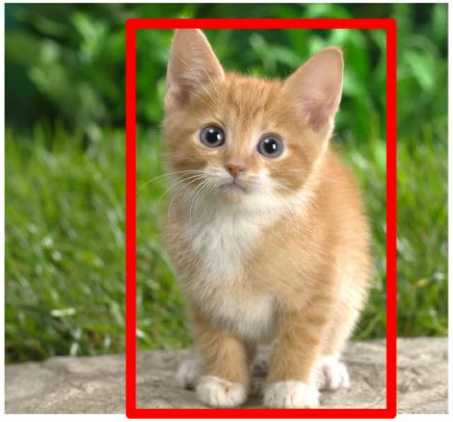
Classification scores:
P(cat)

Sliding Window: Overfeat

Greedily merge boxes and scores



Network input:
3 x 221 x 221



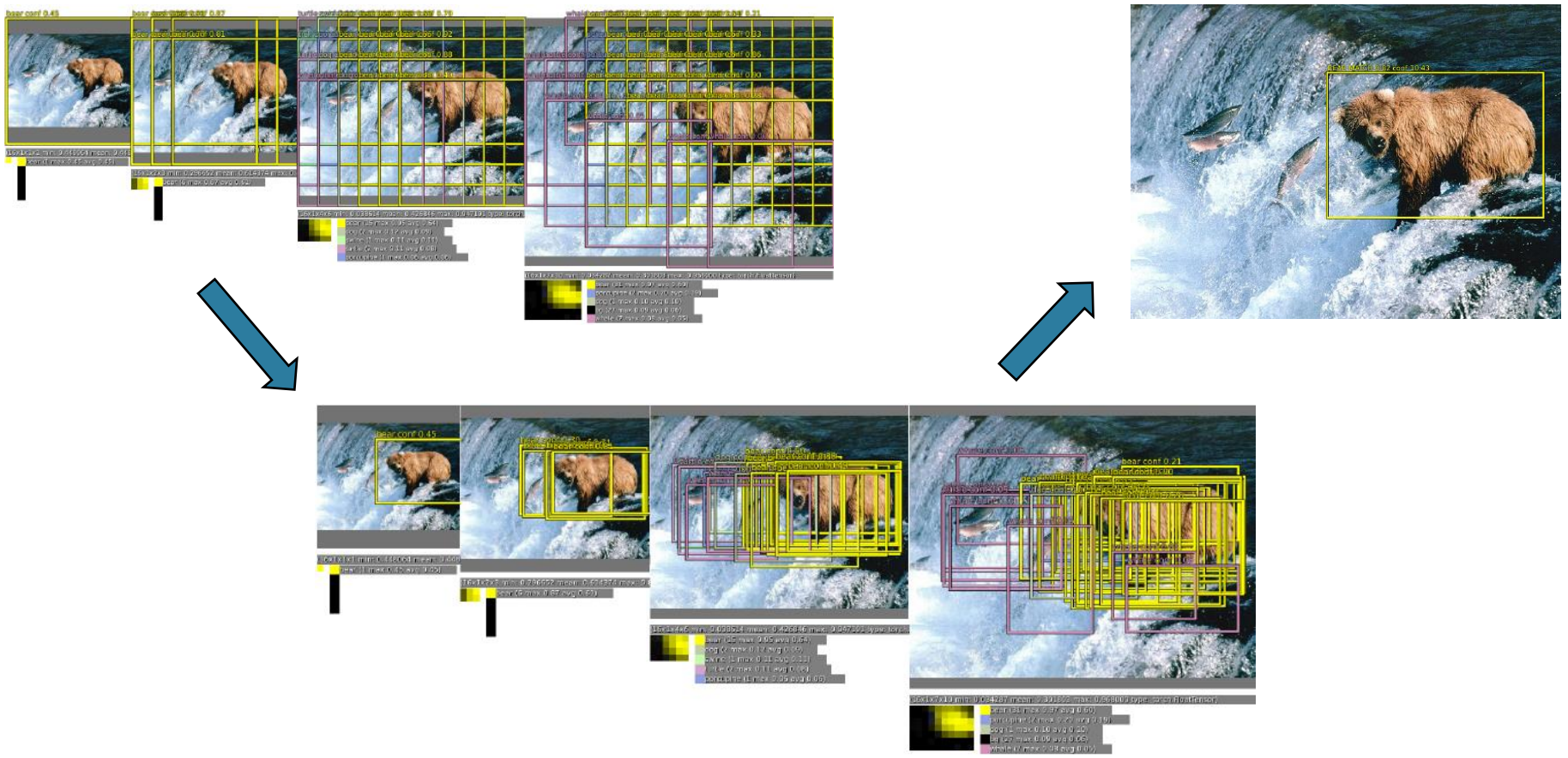
Larger image:
3 x 257 x 257

0.8

Classification score: P
(cat)

Sliding Window: Overfeat

In practice use many sliding windows and multiple scales

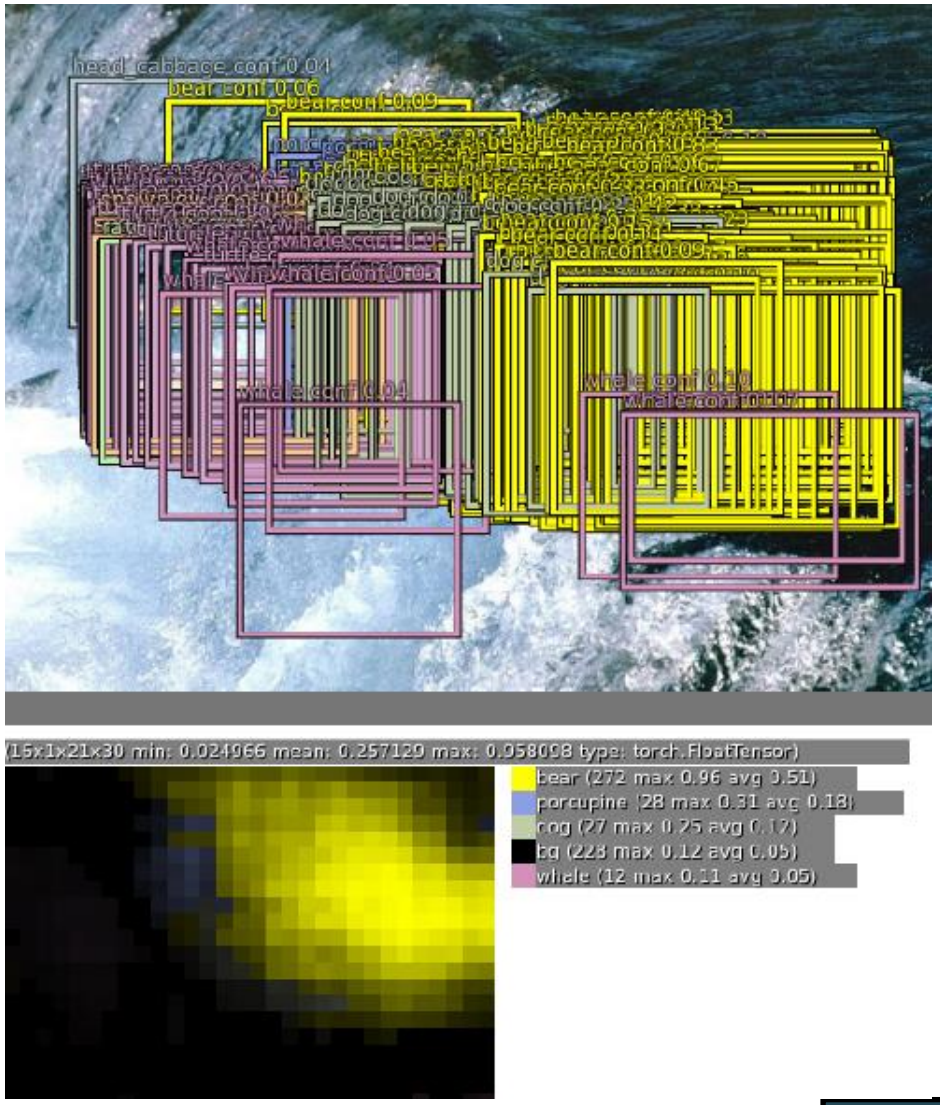


Sermanet et al,“Integrated Recognition, Localization and Detection using Convolutional Networks”,2014

Overfeat: How to merge bounding boxes?

- Merged bbox score is computed by cumulatively adding the detection class outputs associated with each bounding box predicted.
- Pink boxes (whale) disappear in the final stage due to their low confidence scores and they do not cohere as much.

DOWNSIDE: Computation time (need to test many positions and scales). Not used anymore!

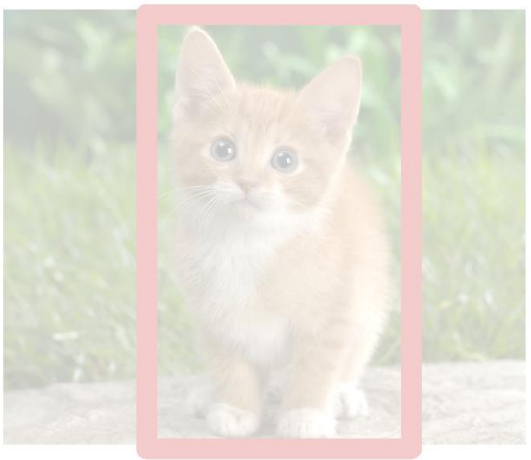


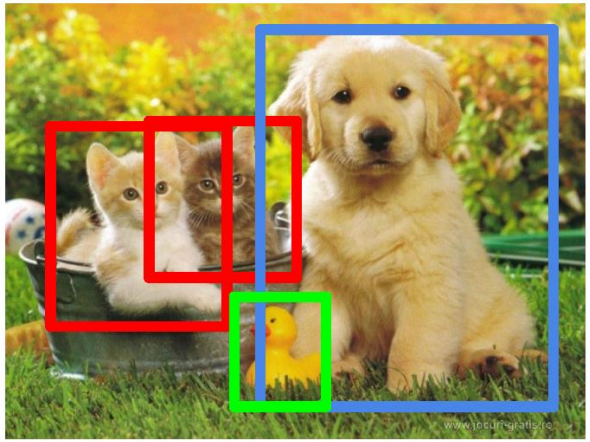
Computer Vision Tasks

Classification

Classification + Localization

Object Detection

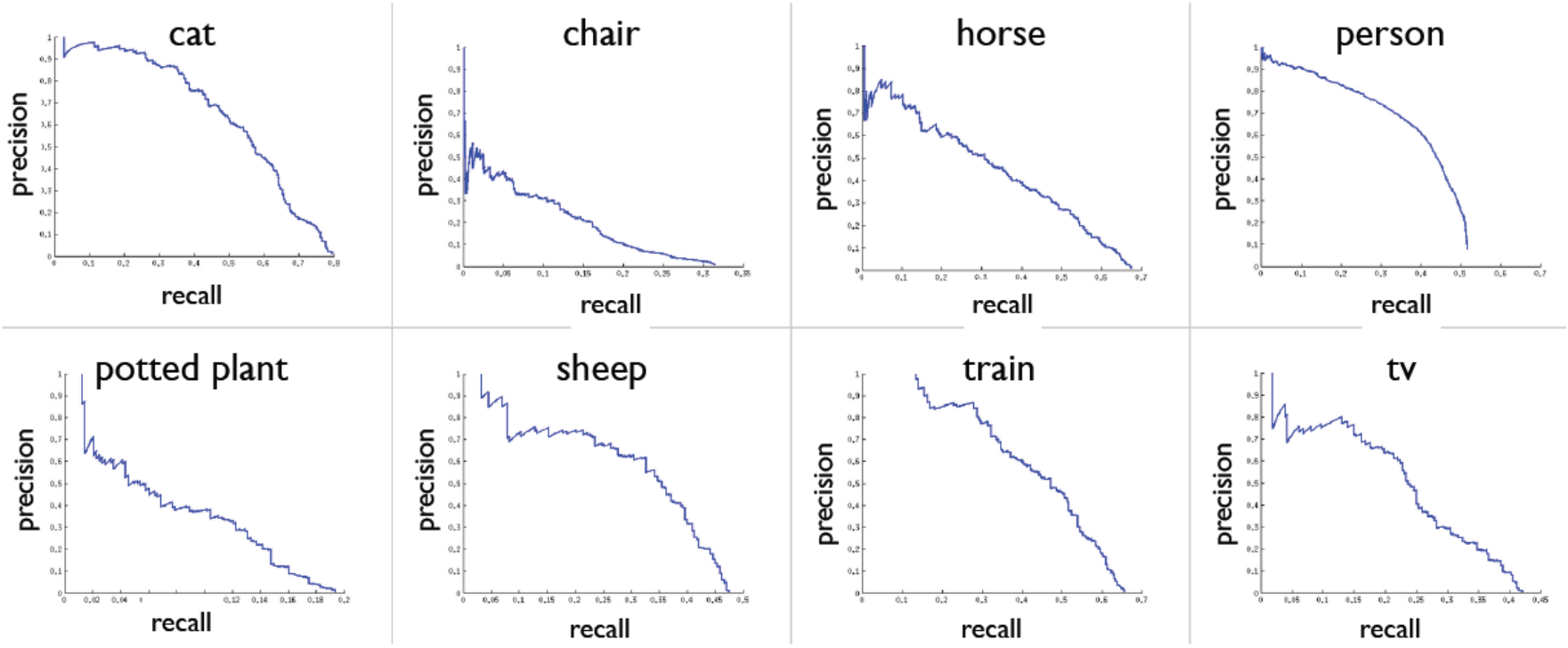




Object Detection

Detection task differs from localization. There can be any number of objects in each image (including zero), and false positives are penalized by the mean average precision (mAP) evaluation metric.

A detection is a true-positive if it has IoU greater than 0.5 (mAP@0.5).

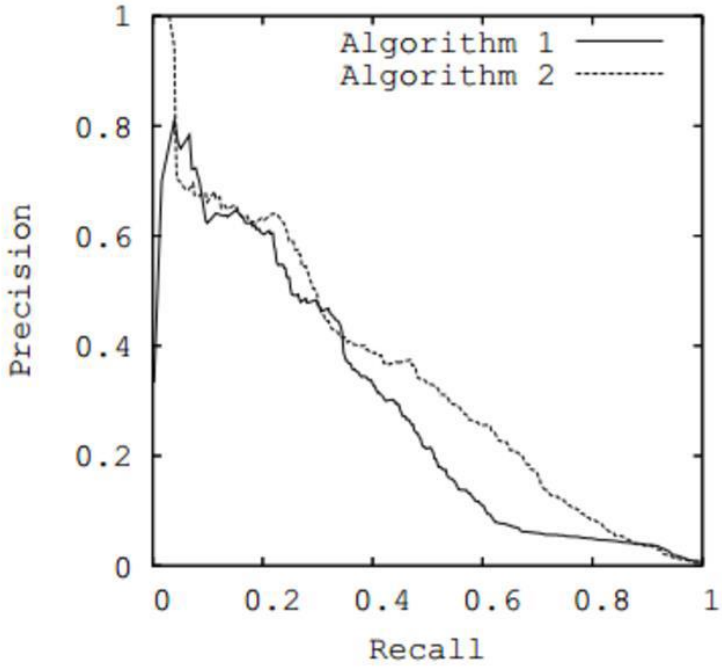


Object Detection

Algorithms are compared with Precision-Recall curves:

$$\text{Precision} = \frac{\#\,\text{True positives}}{\#\,\text{Predicted positives}}$$

$$\text{Recall} = \frac{\#\,\text{True positives}}{\#\,\text{Actual positives}}$$



Average Precision (AP): Area under curve

mean Average Precision (mAP) is simply all the AP values averaged over different classes.

Detection as Classification? (sliding windows)

CAT? NO

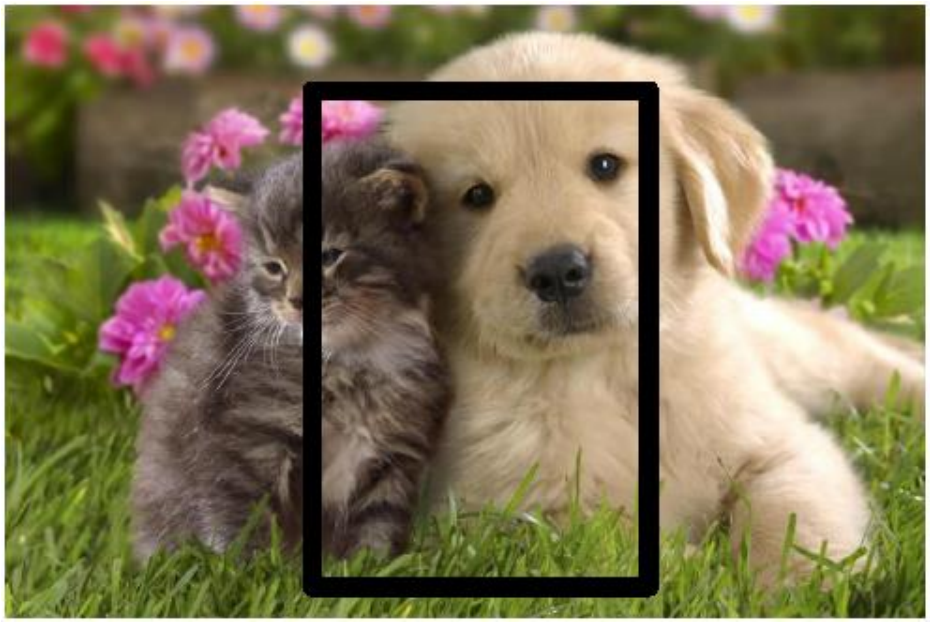
DOG? NO

Detection as Classification? (sliding windows)

CAT? YES!

DOG? NO

Detection as Classification? (sliding windows)



CAT? NO

DOG? NO

Problem: Need to test many positions and scales. Computationally expensive (remember Overfeat*).

*Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

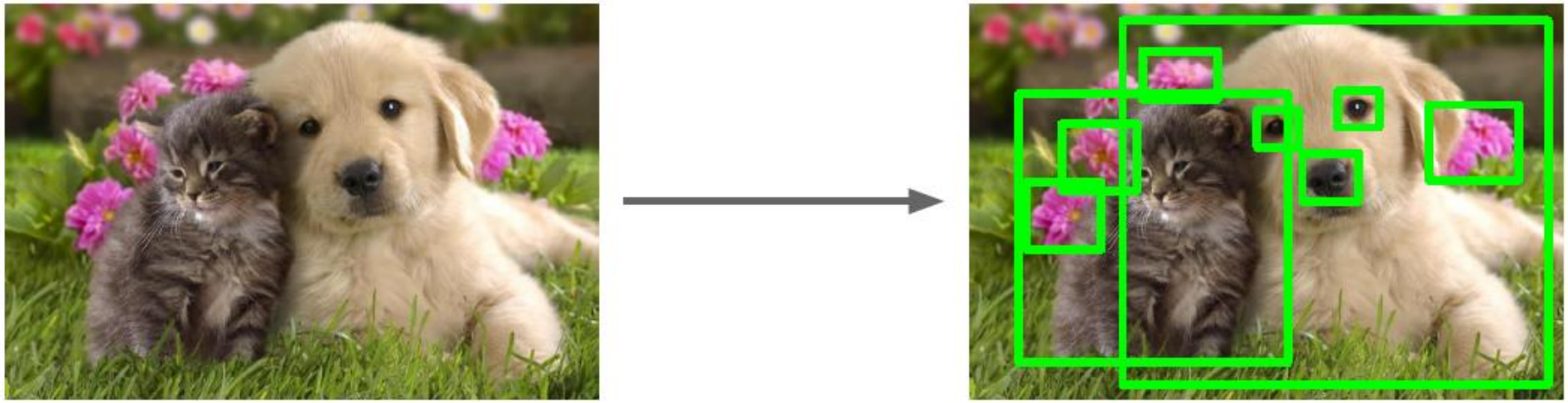
Detection as Classification? (sliding windows)

Problem: Need to test many positions and scales and use a computationally demanding classifier (CNN) while finishing detection in a reasonable time.
Want to test a few images in a second for instance.

Solution: Only look at a tiny subset of possible positions.

Region Proposals

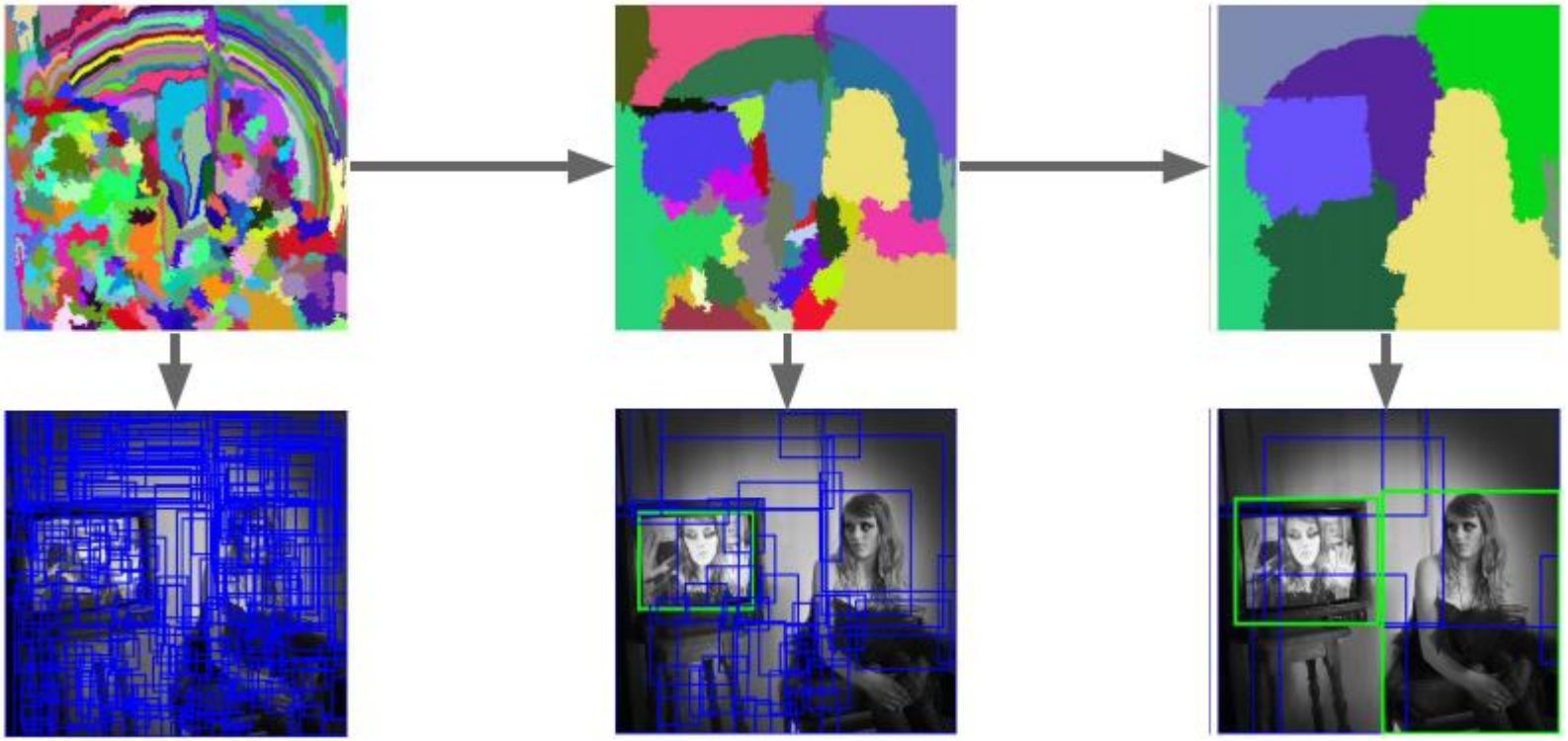
- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector



Region Proposals: Selective Search*

Converts regions to boxes

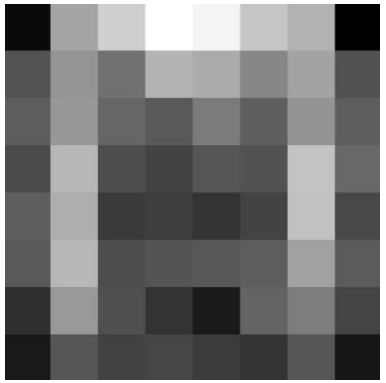
Bottom-up segmentation, merging regions at multiple scales

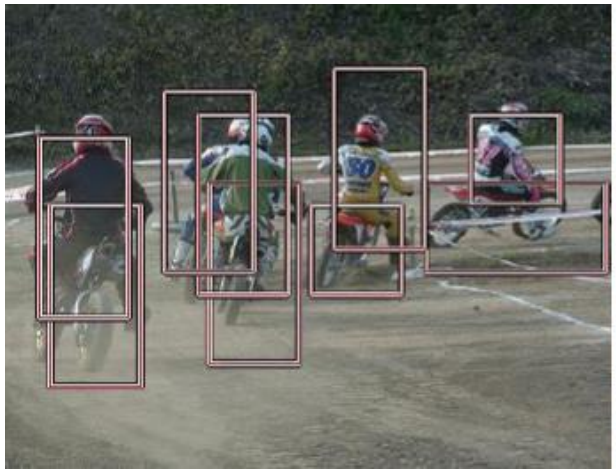


* Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

Region Proposals: BING*

Check objectness in a sliding window fashion.
Warp and resize each window to 8x8 pixels.
Use the gradients as a simple 64D feature and compare with the trained model.





* Cheng, M. M., Zhang, Z., Lin, W.Y., Torr, P. “BING: Binarized Normed Gradient for Objectness Estimation at 300 fps”. CVPR 2014.

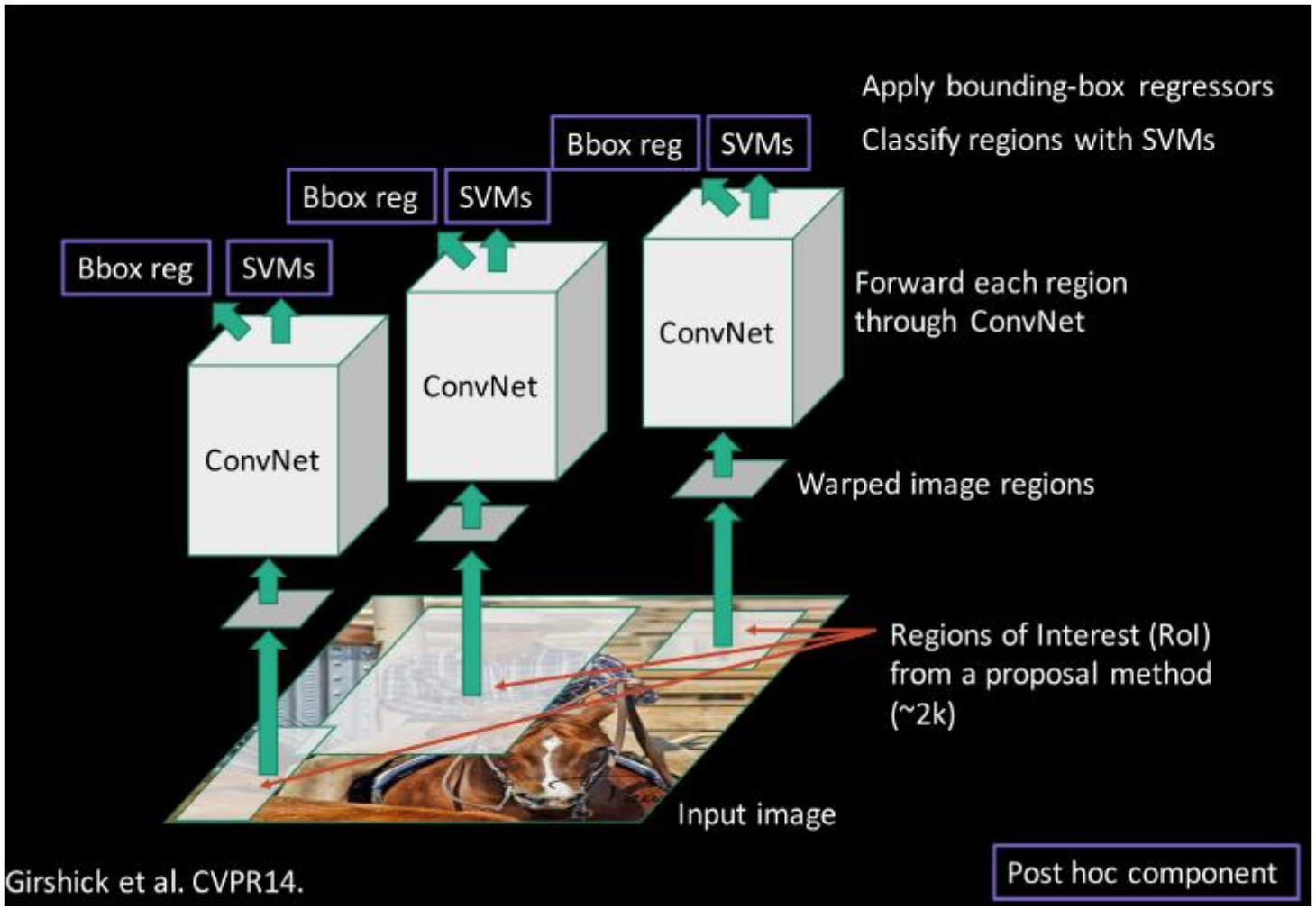
Region Proposals: Many alternatives*

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
Bing [18]	Window scoring		✓	✓	0.2	⋆⋆⋆	⋆	·
CPMC [19]	Grouping	✓	✓	✓	250	-	⋆⋆	⋆
EdgeBoxes [20]	Window scoring		✓	✓	0.3	⋆⋆	⋆⋆⋆	⋆⋆⋆
Endres [21]	Grouping	✓	✓	✓	100	-	⋆⋆⋆	⋆⋆
Geodesic [22]	Grouping	✓		✓	1	⋆	⋆⋆⋆	⋆⋆
MCG [23]	Grouping	✓	✓	✓	30	⋆	⋆⋆⋆	⋆⋆⋆
Objectness [24]	Window scoring		✓	✓	3	·	⋆	·
Rahtu [25]	Window scoring		✓	✓	3	·	·	⋆
RandomizedPrim's [26]	Grouping	✓		✓	1	⋆	⋆	⋆⋆
Rantalankila [27]	Grouping	✓		✓	10	⋆⋆	·	⋆⋆
Rigor [28]	Grouping	✓		✓	10	⋆	⋆⋆	⋆⋆
SelectiveSearch [29]	Grouping	✓	✓	✓	10	⋆⋆	⋆⋆⋆	⋆⋆⋆
Gaussian				✓	0	·	·	⋆
SlidingWindow				✓	0	⋆⋆⋆	·	·
Superpixels		✓			1	⋆	·	·
Uniform				✓	0	·	·	·

Soon, we will see how to propose regions by CNNs

*Hosang et al, “What makes for effective detection proposals?”, PAMI 2015

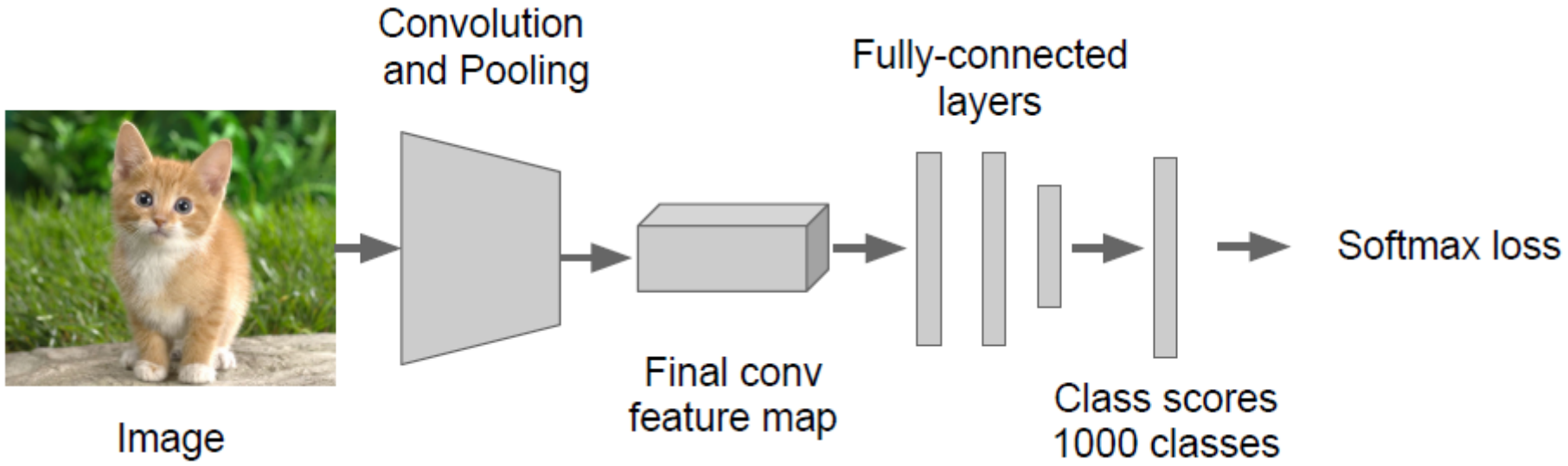
Putting it together: R-CNN*



*Girshick et al, “R-CNN. Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN Training

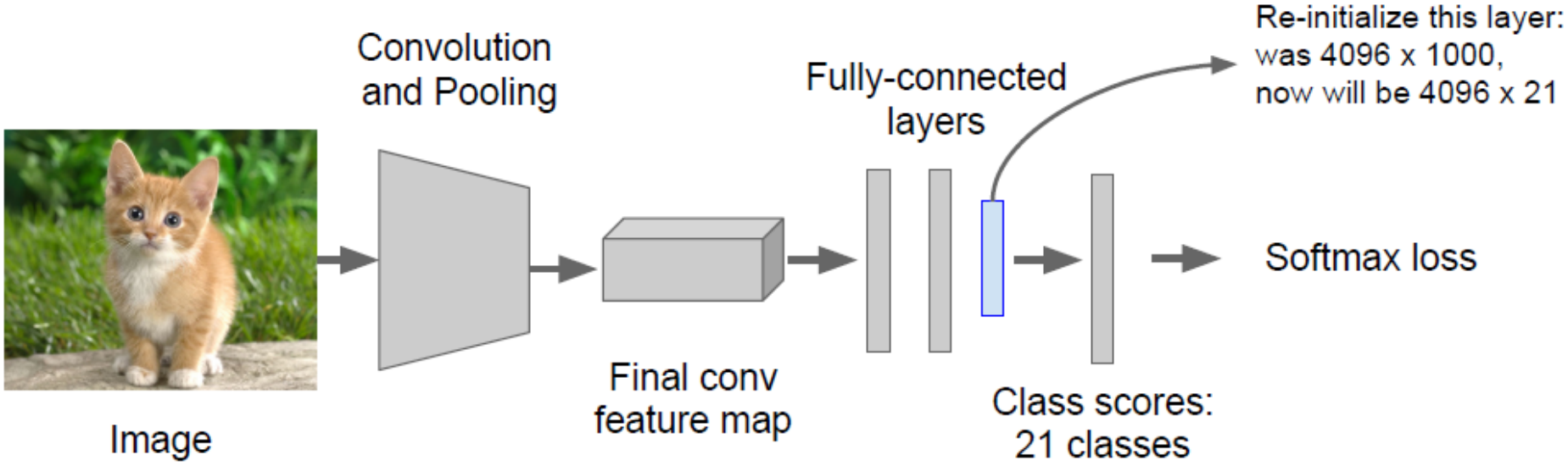
Step 1: Train (or download) a classification model for ImageNet (AlexNet)



R-CNN Training

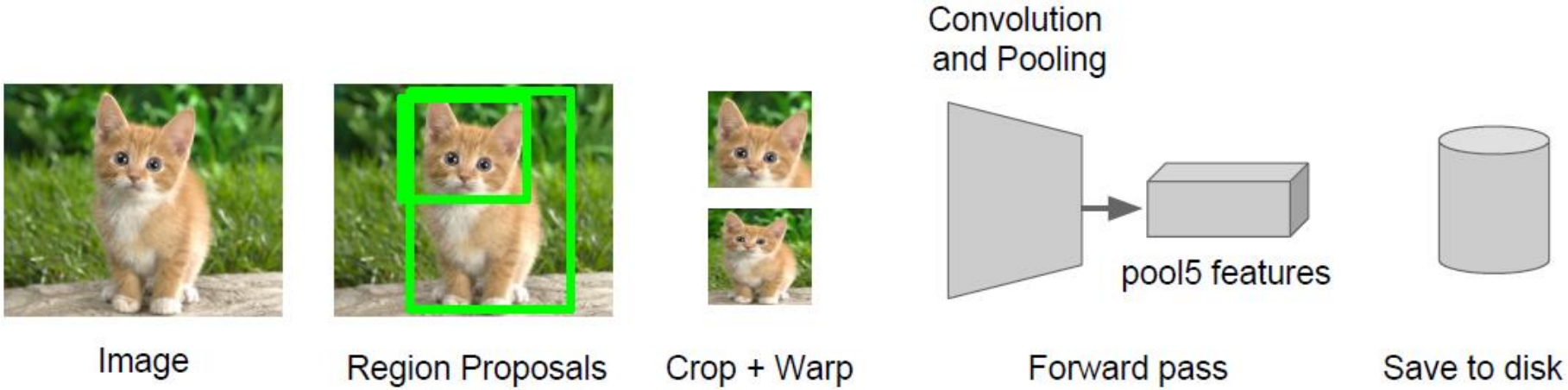
Step 2: Fine-tune model for detection

- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images



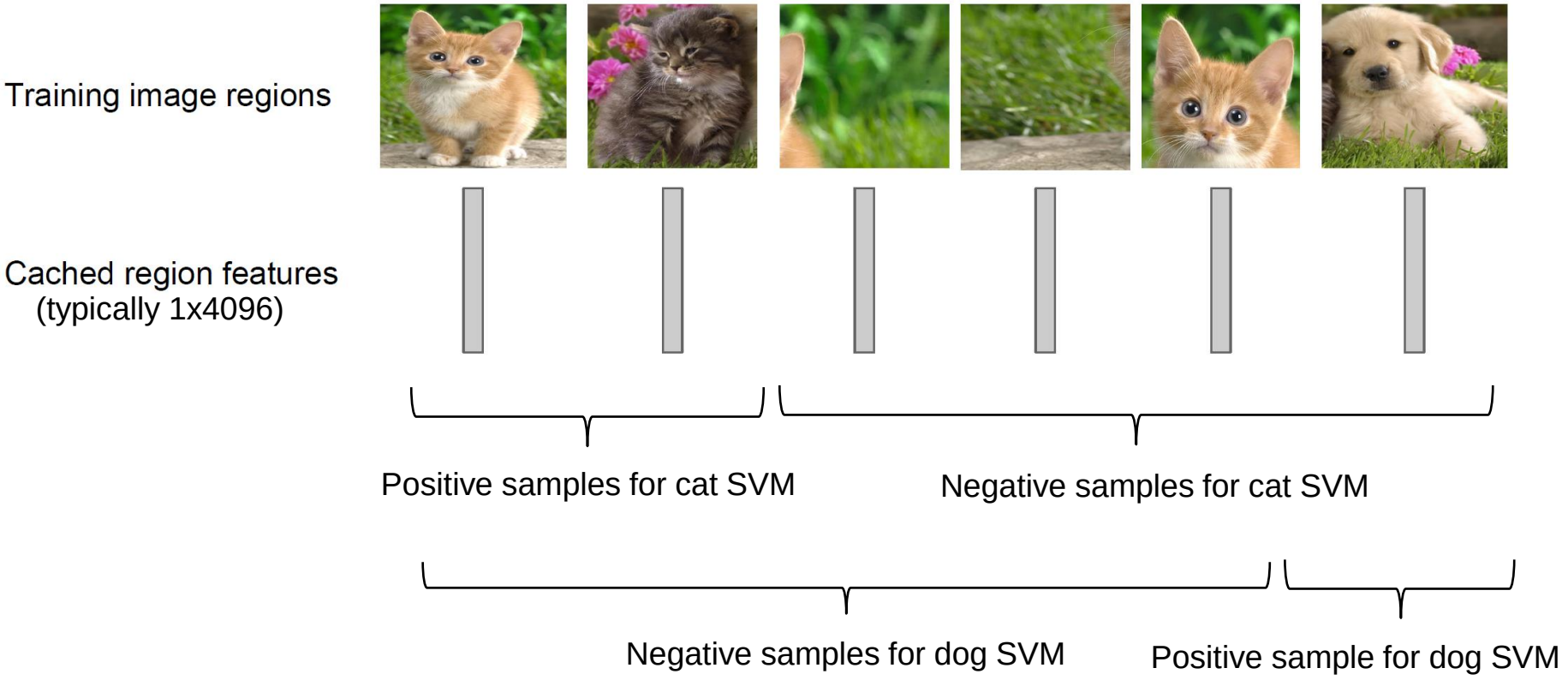
R-CNN Training

Step 3: Extract features
- Extract region proposals for all images
- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk



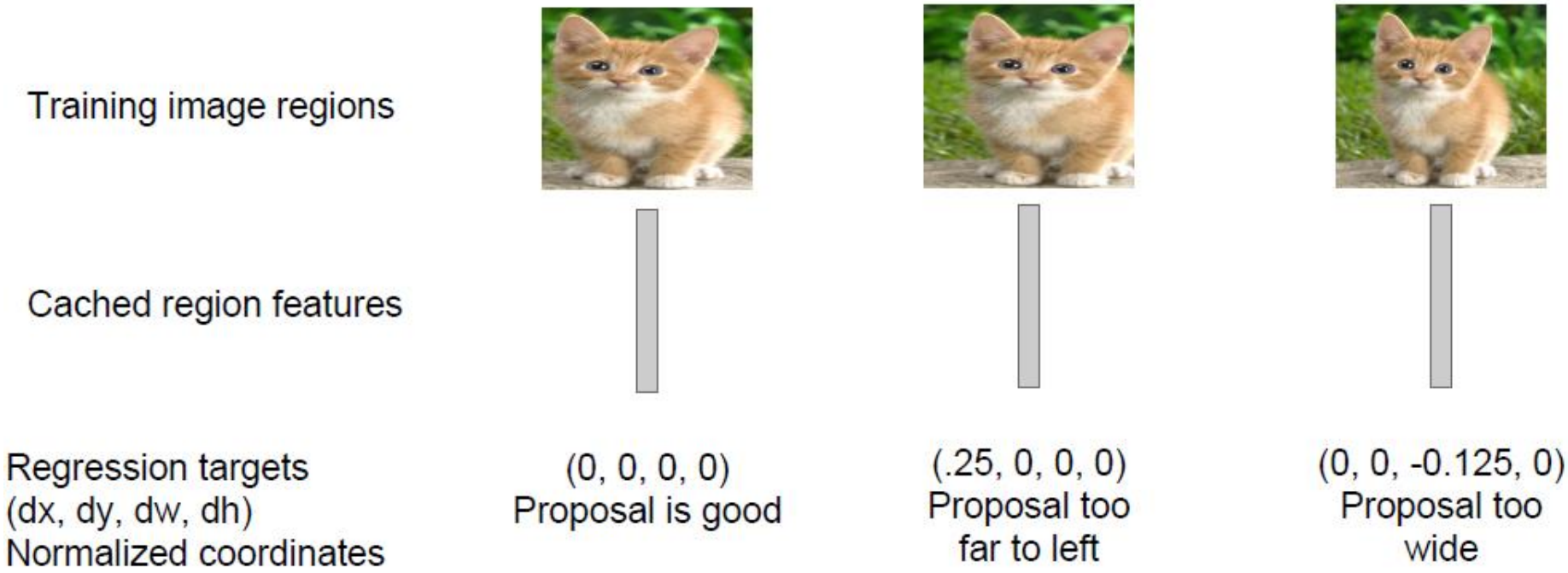
R-CNN Training

Step 4: Train one binary SVM per class to classify region features



R-CNN Training

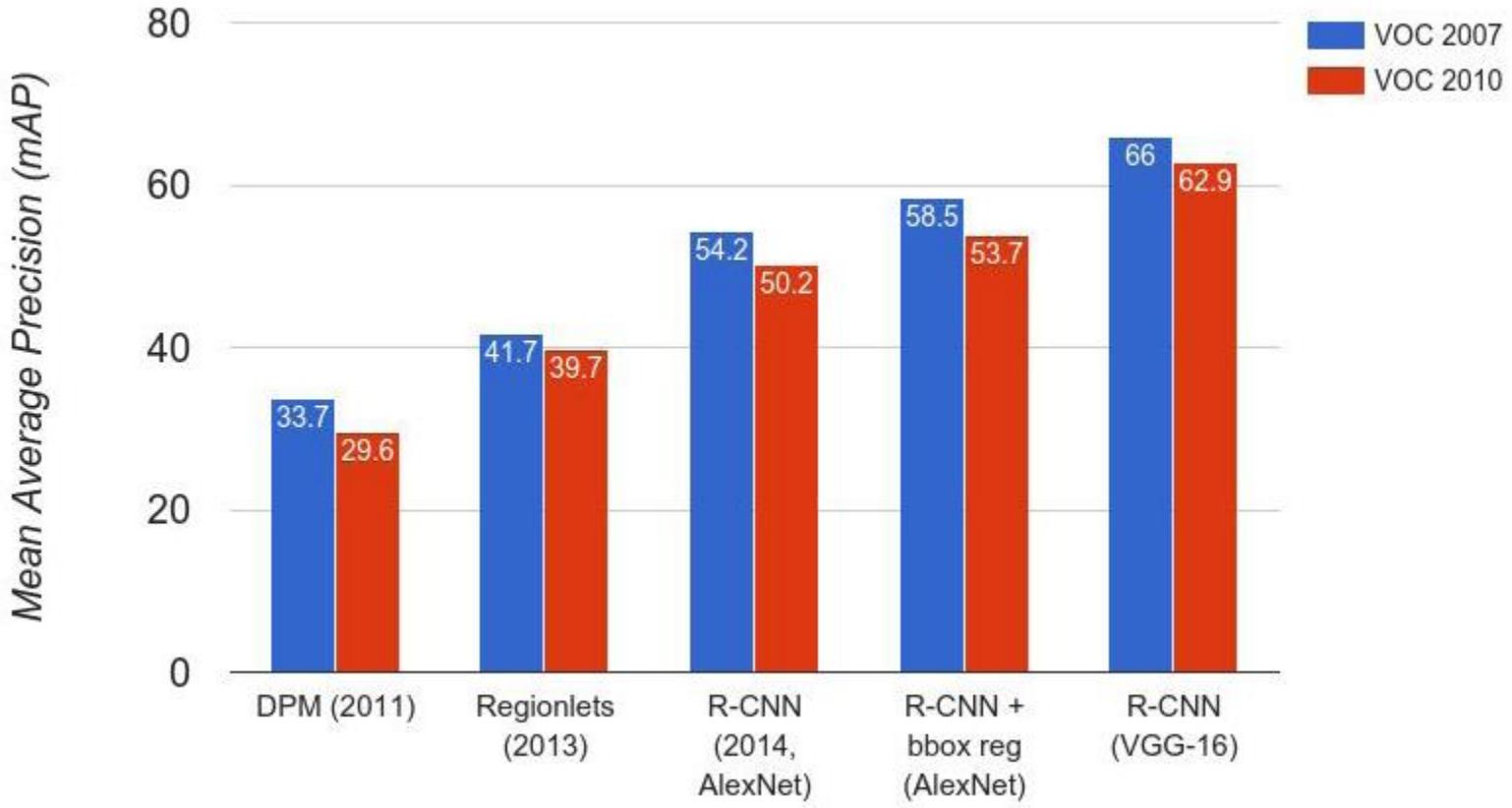
Step 5 (bbox regression): For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for “slightly wrong” proposals



R-CNN Results on PASCAL VOC dataset

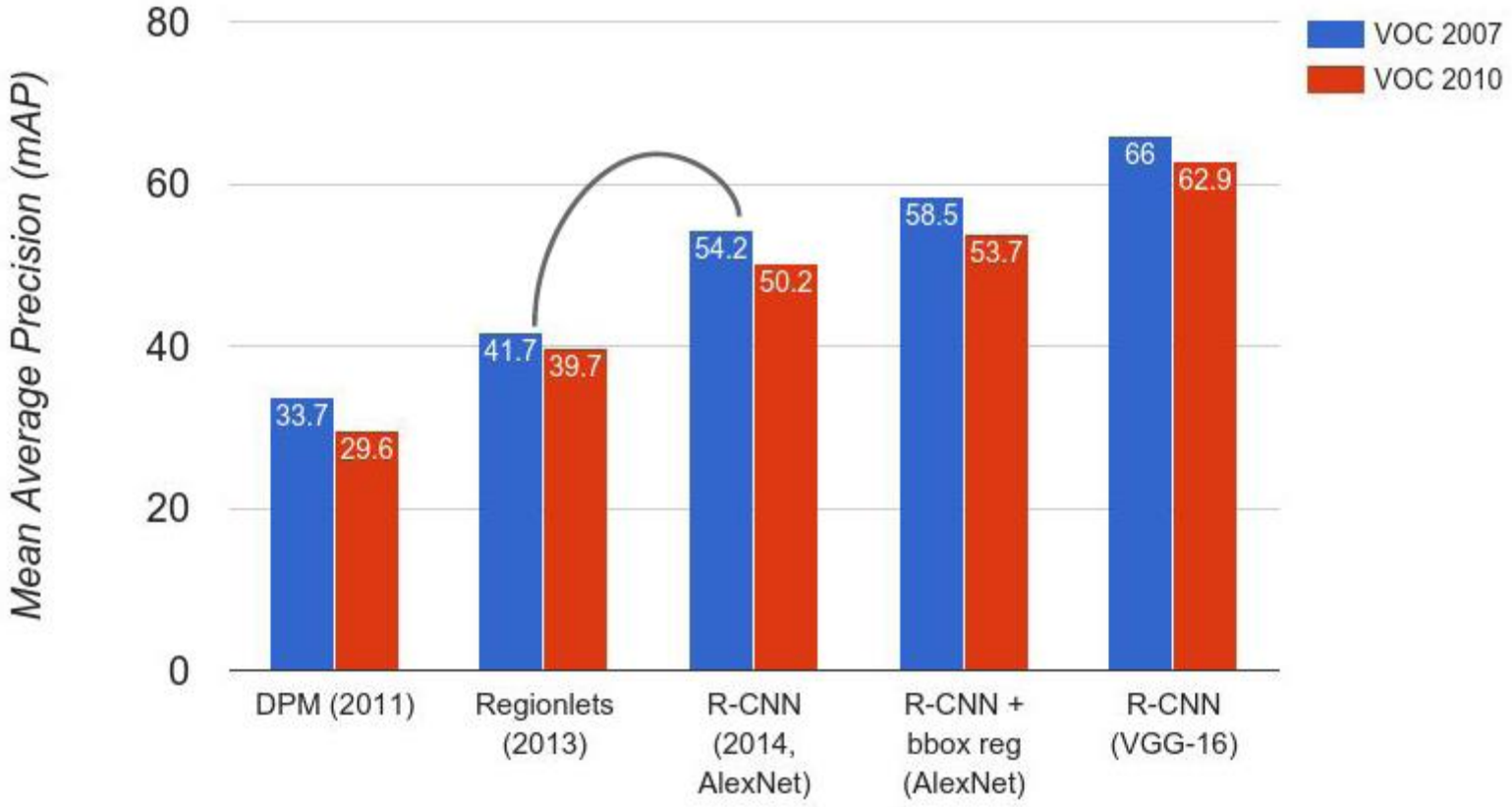
Evaluation metric is “mean average precision” (mAP).

Average precision (AP) is computed separately for each class.

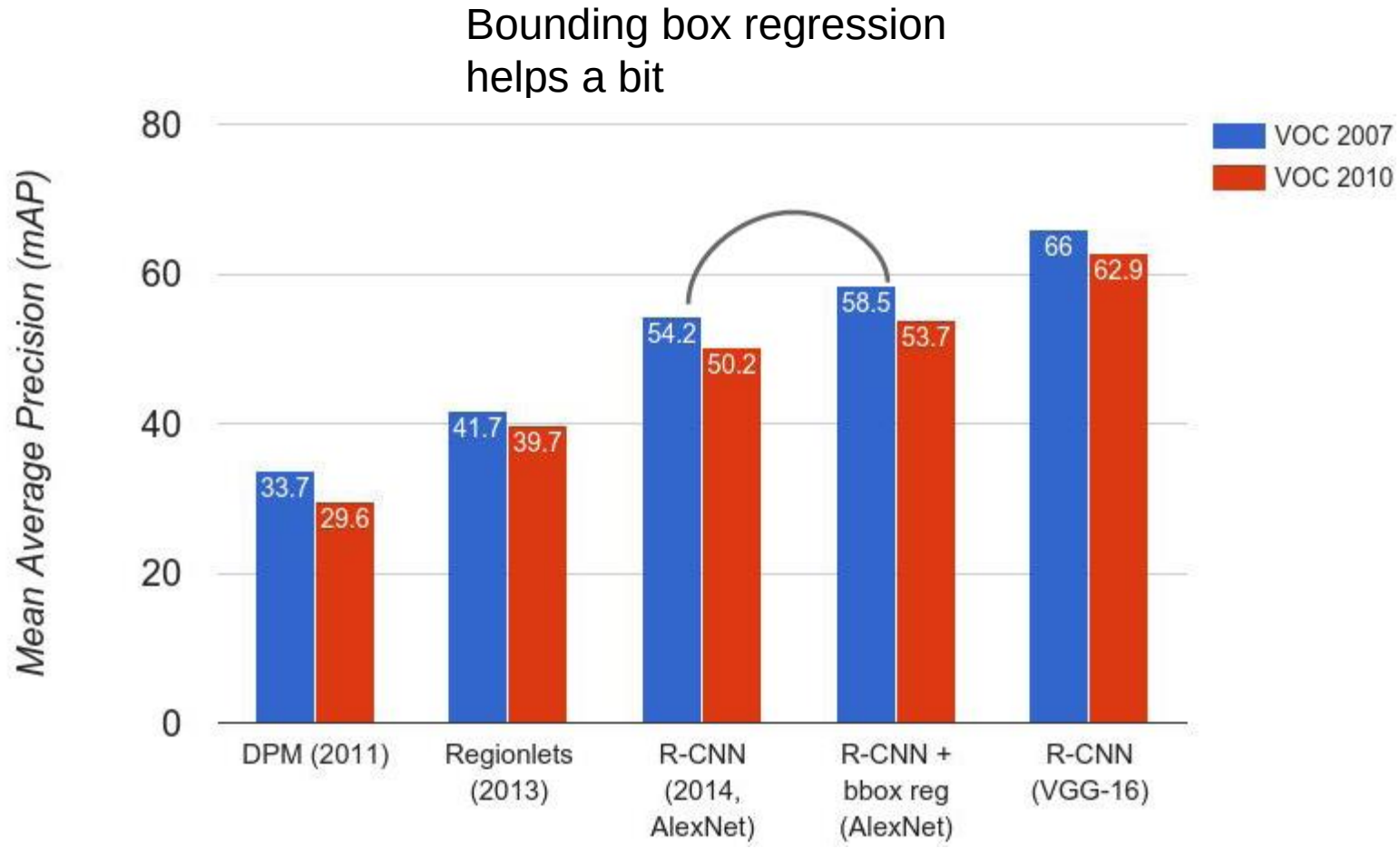


R-CNN Results on PASCAL VOC dataset

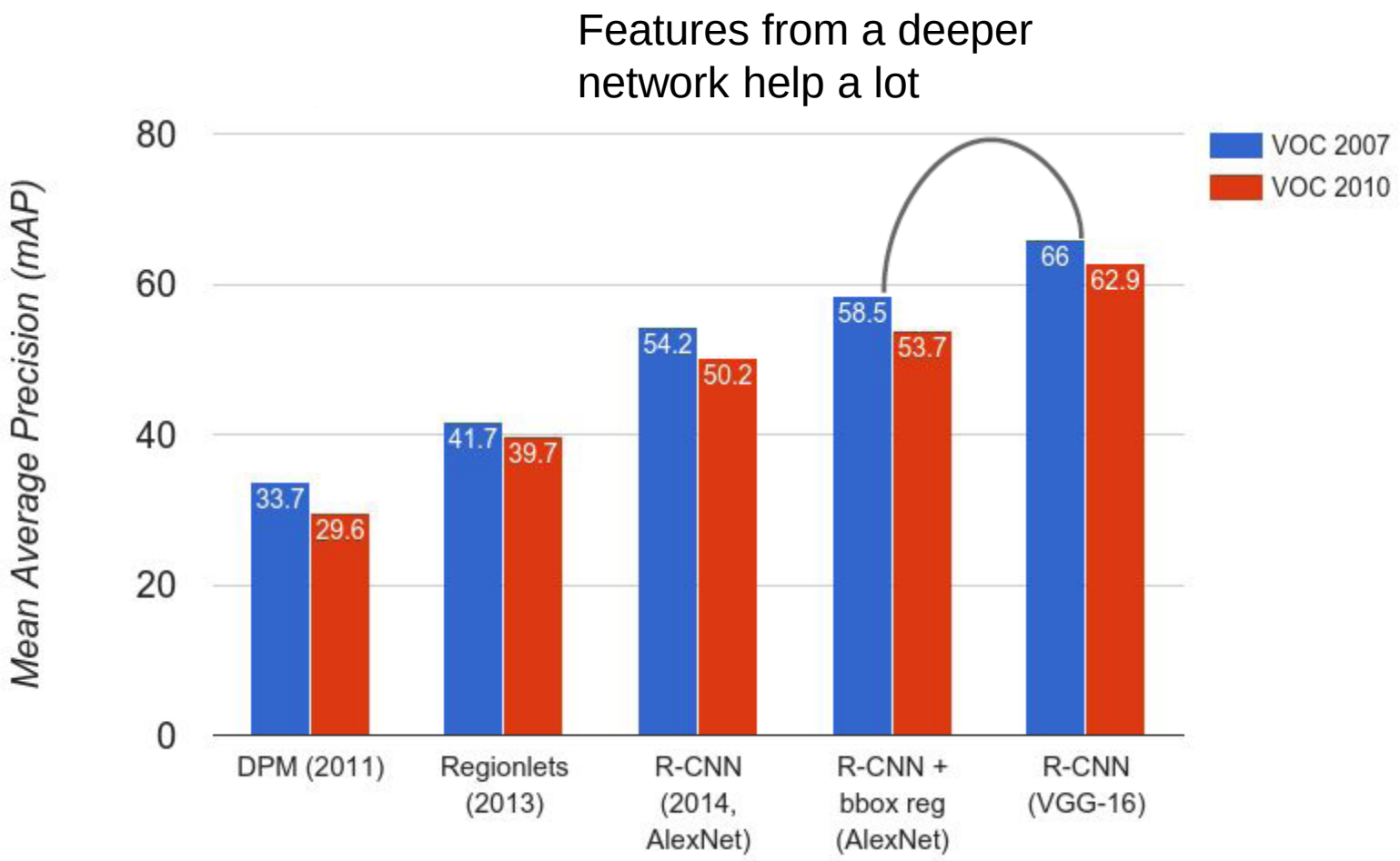
Big improvement compared
to pre-CNN methods



R-CNN Results on PASCAL VOC dataset



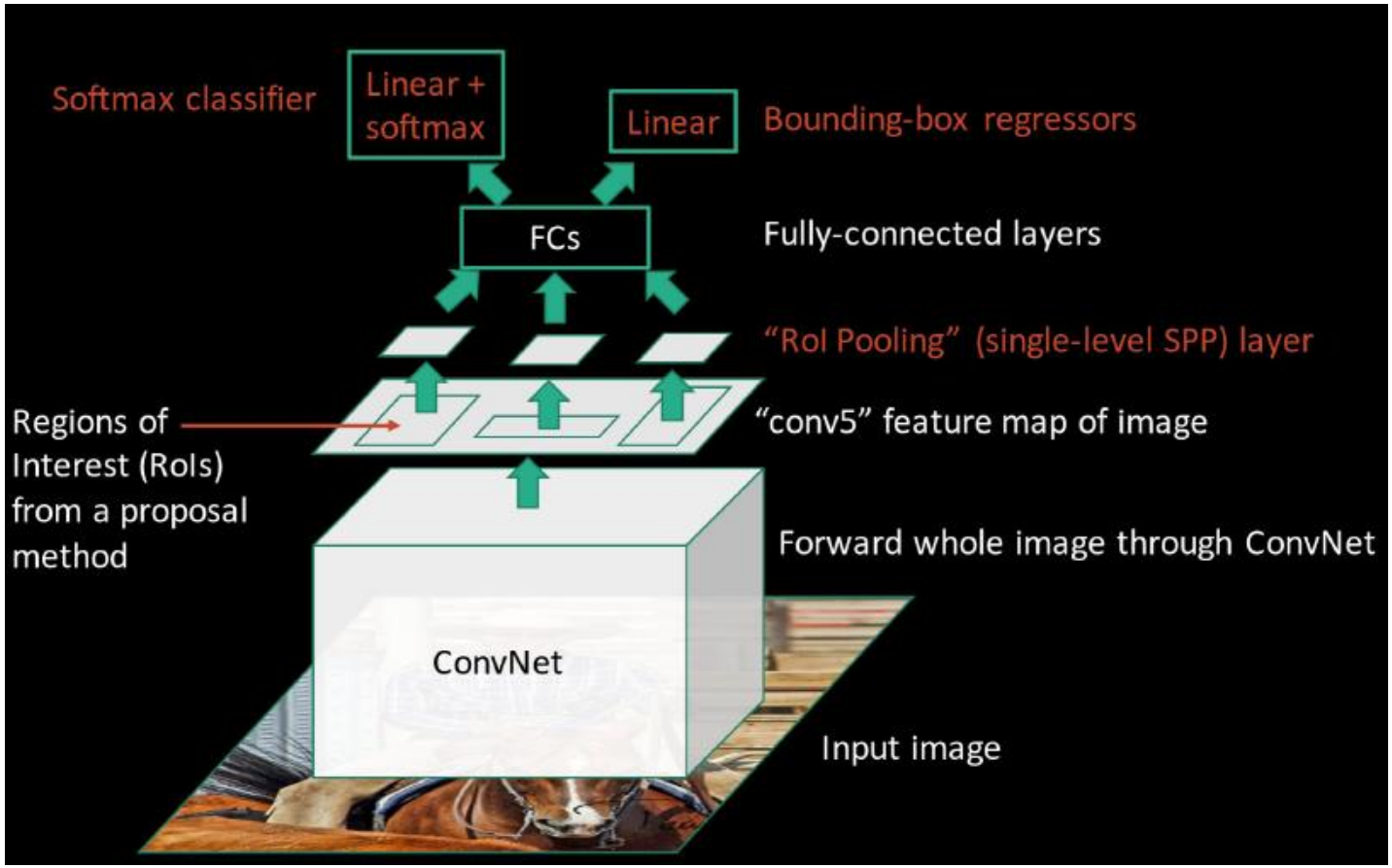
R-CNN Results on PASCAL VOC dataset



R-CNN Problems

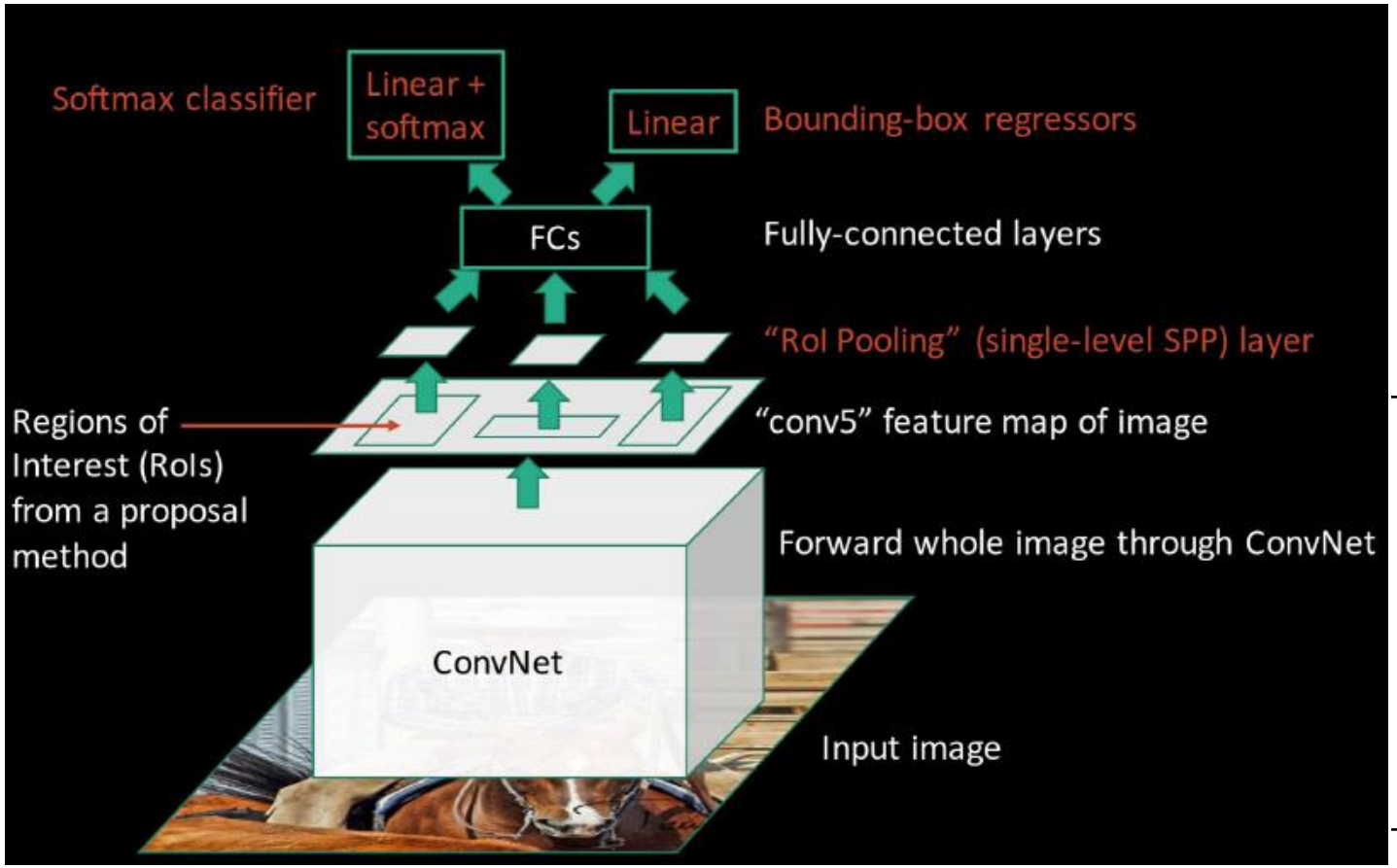
1. Slow at test-time: need to run full forward pass of CNN for each region proposal separately
   (47s / image with VGG16 [Simonyan & Zisserman. ICLR15])
2. SVMs and regressors are post-hoc: CNN features not updated in response to SVMs and regressors

Fast R-CNN (test time)



Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN (test time)

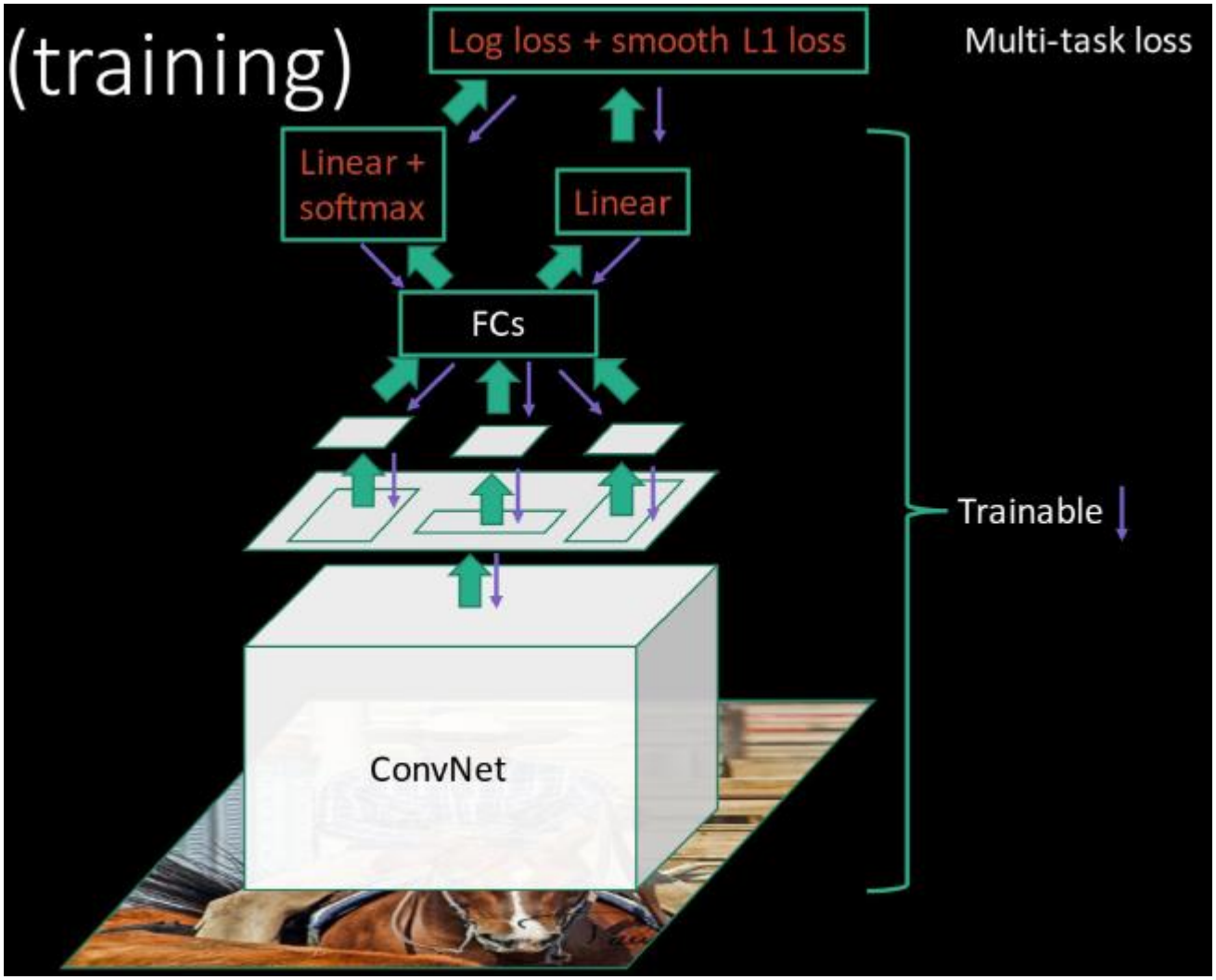


R-CNN Problem #1: Slow at test-time due to separate forward passes of the CNN

Solution: Share computation of conv layers between proposals in an image

Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN (training)



R-CNN Problem #2: Post-hoc training: CNN not updated in response to final classifiers and regressors

Solution:
No one-vs-rest SVMs.
Just train the whole system all at once with multi-task loss!

Girschick, “Fast R-CNN”, ICCV 2015

Fast R-CNN Results

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

Faster!

FASTER!

a bit better!

Using VGG-16 CNN on Pascal VOC 2007 dataset

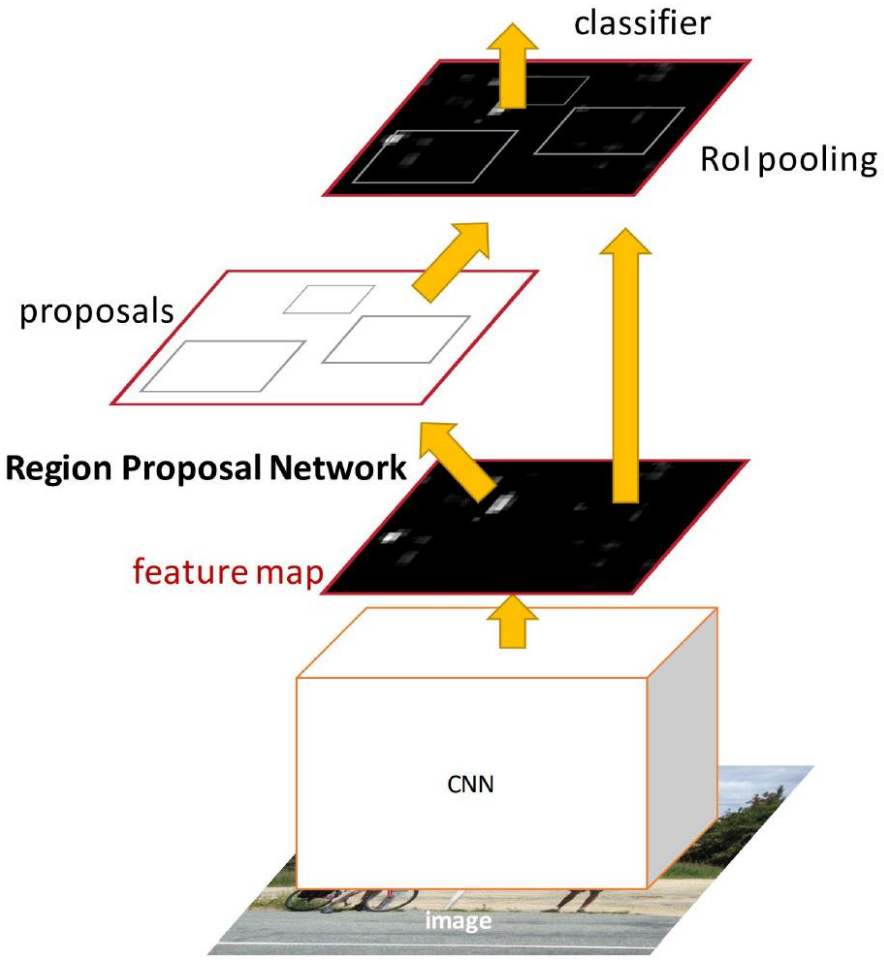
Fast R-CNN Problem

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**

Solution: Make the CNN do region proposals too!

Faster R-CNN*



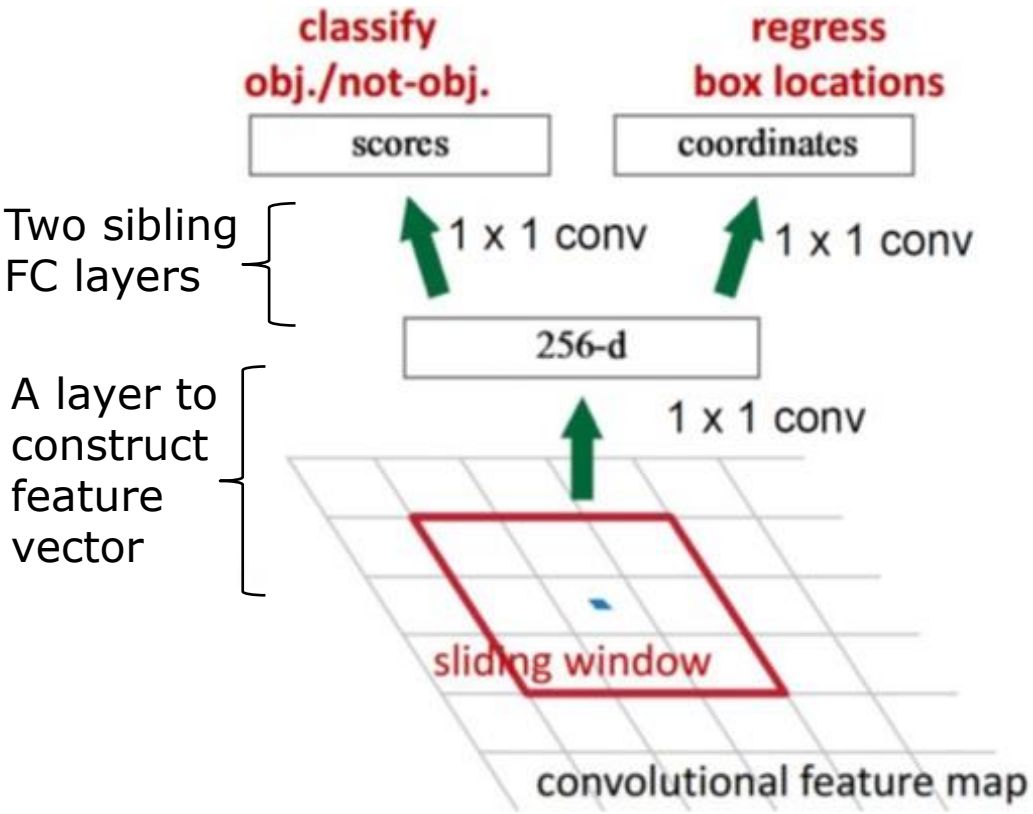
Insert a **Region Proposal Network (RPN)** after the last convolutional layer

RPN trained to produce region proposals directly; no need for external region proposals!

After RPN, use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN

*Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Faster R-CNN: Region Proposal Network

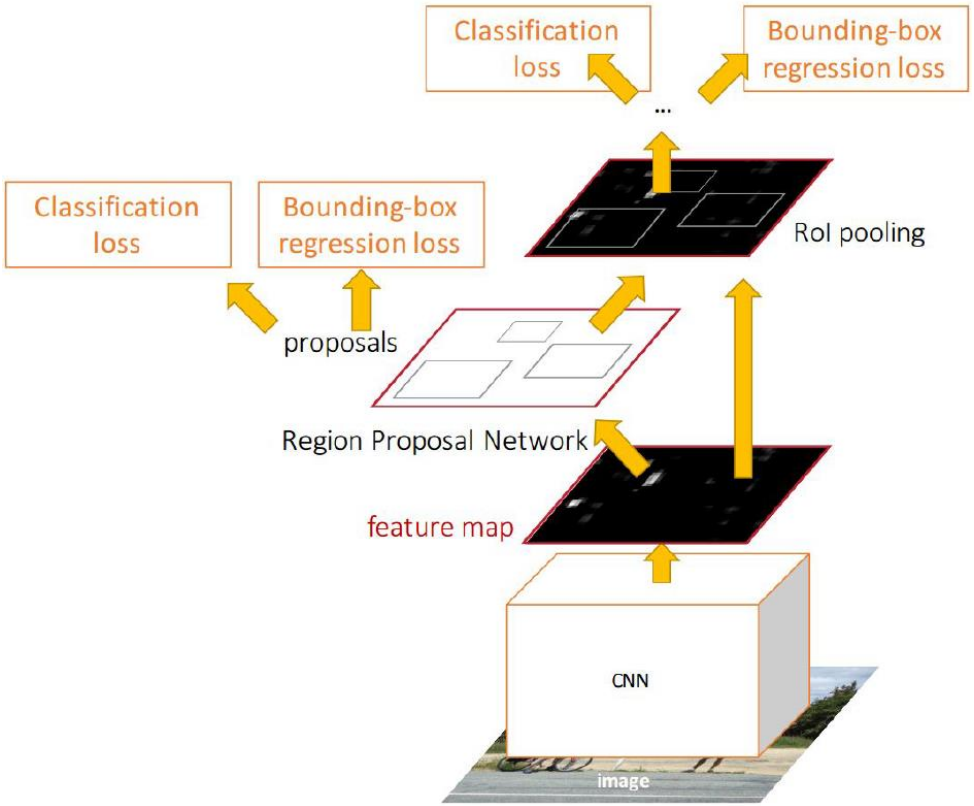


Slide a small window on the feature map

Build a small network to:
- classify object or not-object, and
- regress bbox locations

Box regression provides finer localization with reference to the center of the sliding window

Faster R-CNN: Training



Jointly training four losses

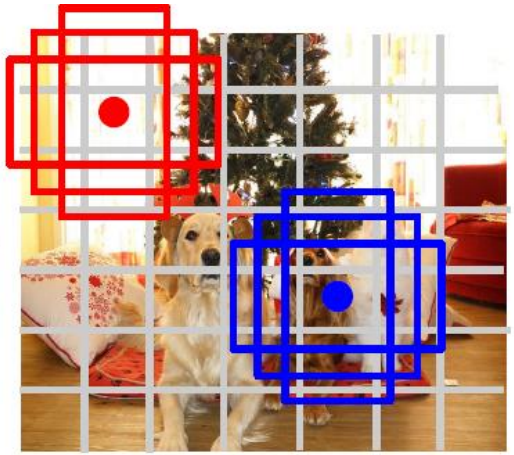
- RPN classification (object / not object)
- RPN regress box coordinates
- Final classification score (over classes)
- Final box coordinates (proposal -> box)

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**

Detection without Proposals: YOLO / SSD

Input image
3 x H x W



Divide image into grid
7 x 7

Get a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:

- Regress from each of the B base boxes to a final box with 5 numbers:

(dx, dy, dh, dw, confidence)

- Predict scores for each of C classes

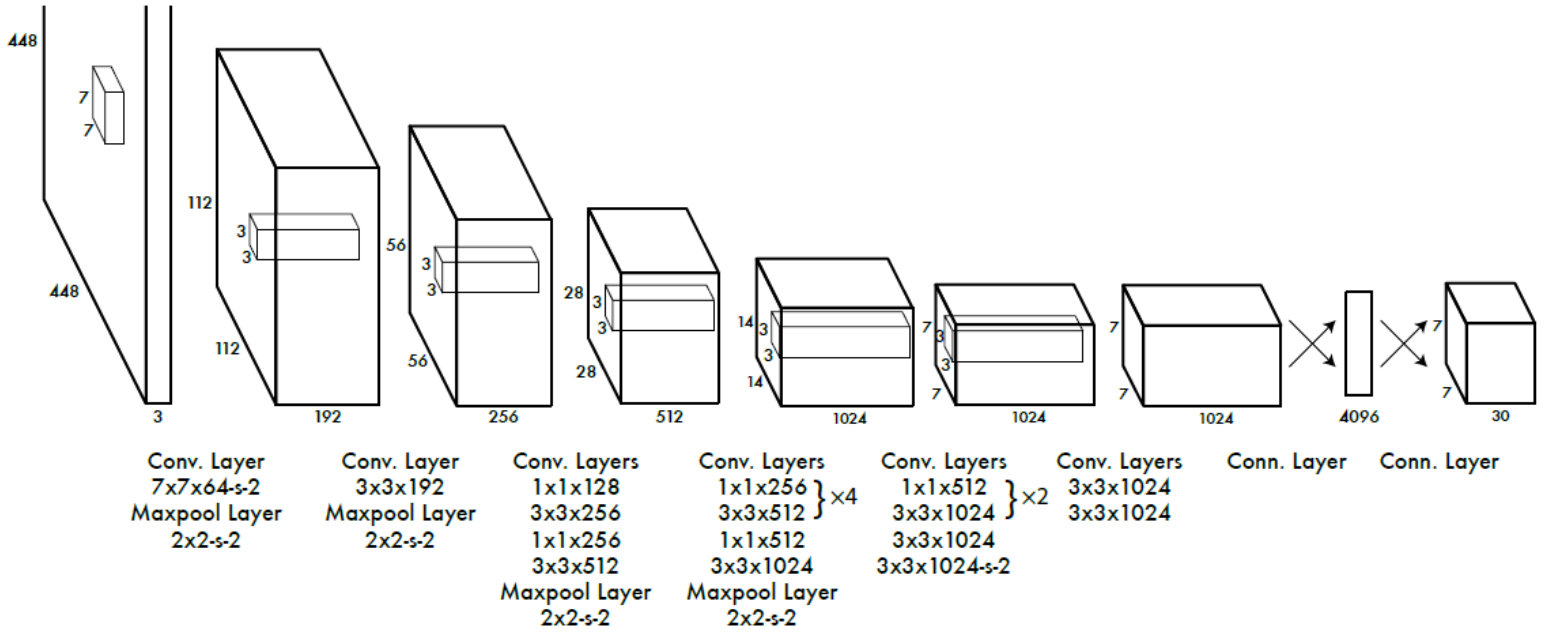
Output:
7 x 7 x (5 * B + C)

Redmon et al, “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016
Liu et al, “SSD: Single-Shot MultiBox Detector”, ECCV 2016

Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!

Regression from image to 7 x 7 x (5 * B + C) tensor.



In YOLO paper, CNN has 24 conv layers followed by 2 FC layers.

Redmon et al, “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

Detection without Proposals: YOLO / SSD

SSD and YOLO are faster than Faster R-CNN.

SSD and YOLOv2 are as good as Faster R-CNN.

YOLO has updated versions: v3, v4, .. (latest is v9 I suppose :)

Detection Frameworks	mAP	FPS
Fast R-CNN [5]	70.0	0.5
Faster R-CNN VGG-16[15]	73.2	7
Faster R-CNN ResNet[6]	76.4	5
YOLO [14]	63.4	45
SSD300 [11]	74.3	46
SSD500 [11]	76.8	19
YOLOv2 288 × 288	69.0	91
YOLOv2 352 × 352	73.7	81
YOLOv2 416 × 416	76.8	67
YOLOv2 480 × 480	77.8	59
YOLOv2 544 × 544	**78.6**	40

scores on PASCAL VOC 2007

Table source: Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, arXiv:1612.08242v1, 2016

* A Guide to the YOLO Family of Computer Vision Models (dataphoenix.info), 2023

Object Detection: Datasets

Classification and localization tasks share the same dataset, while detection dataset has additional data where objects can be smaller.

	PASCAL VOC (2010)	ImageNet Detection (ILSVRC 2014)	MS-COCO (2014)
Number of classes	20	**200**	80
Number of images (train + val)	~20k	**~470k**	~120k
Mean objects per image	2.4	1.1	**7.2**

Object Detection: Lots of Choices

Backbone (Base Network)

- VGG16
- ResNet-101
- Inception
- Inception v2, v3
- ResNet
- MobileNet

Object detection architecture

- Faster R-CNN
- SSD
- YOLO

Variables

Image Size
Region Proposals
…