Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

ROSS GIRSHICK, JEFF DONAHUE, TREVOR DARRELL, JITENDRA MALIK

PRESENTED BY: COLLIN MCCARTHY

Goal

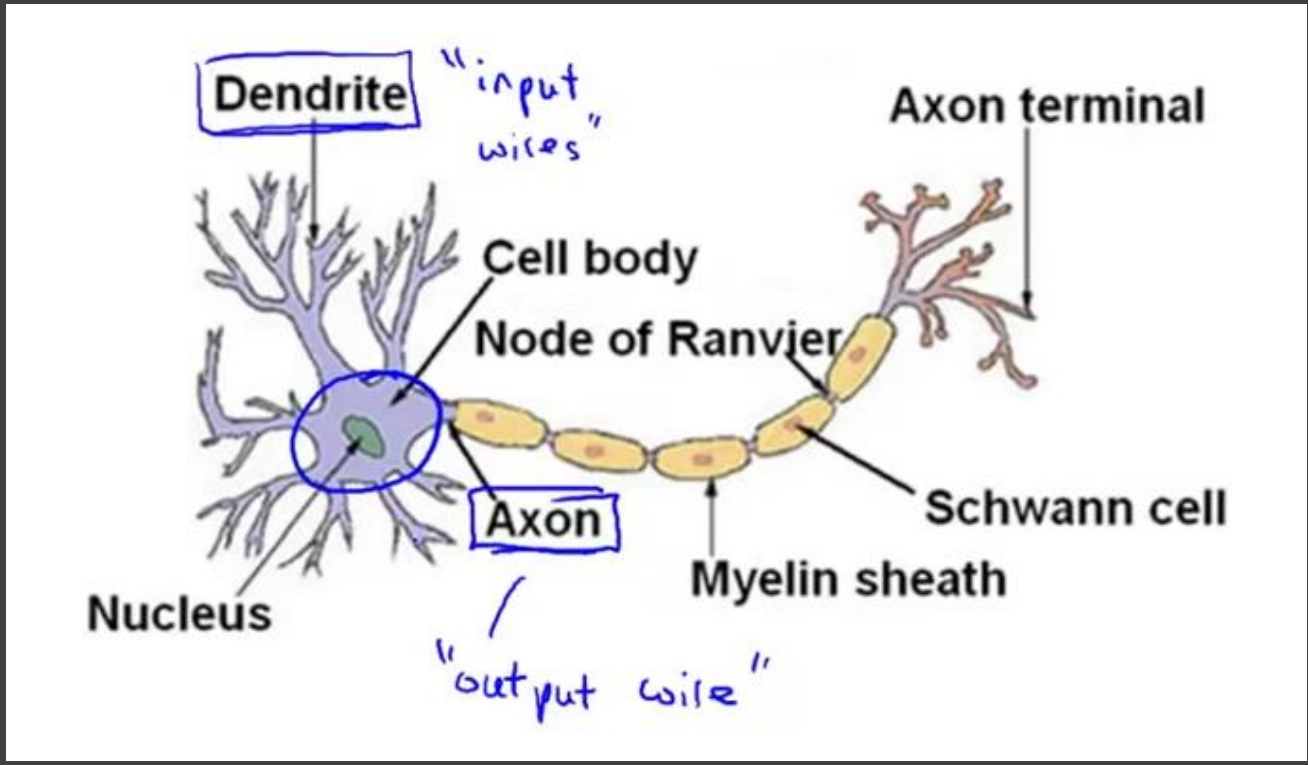
To produce a scalable, state-of-the-art detection algorithm

- CNN's using bottom-up regions
- Pre-training (general), fine-tuning (specific)

Background

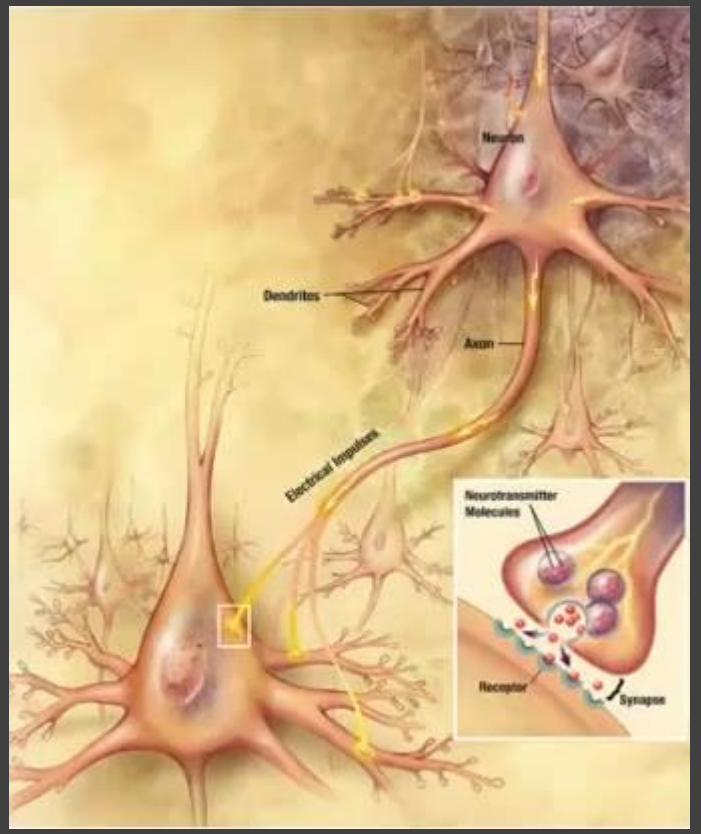
Neurons

Single Neuron



Ng, Andrew. Machine Learning, Lecture 8. Coursera.

Neural Network

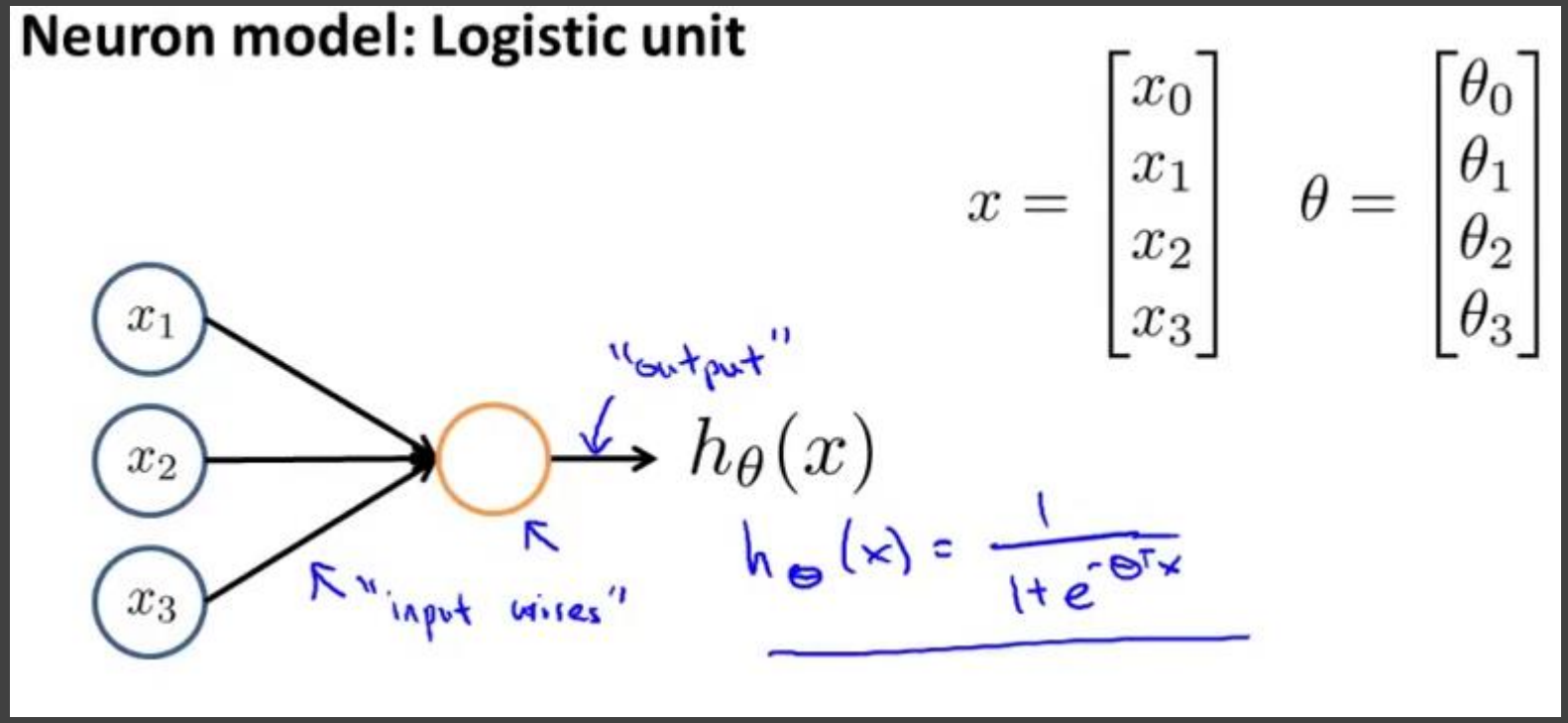


Ng, Andrew. Machine Learning, Lecture 8. Coursera.

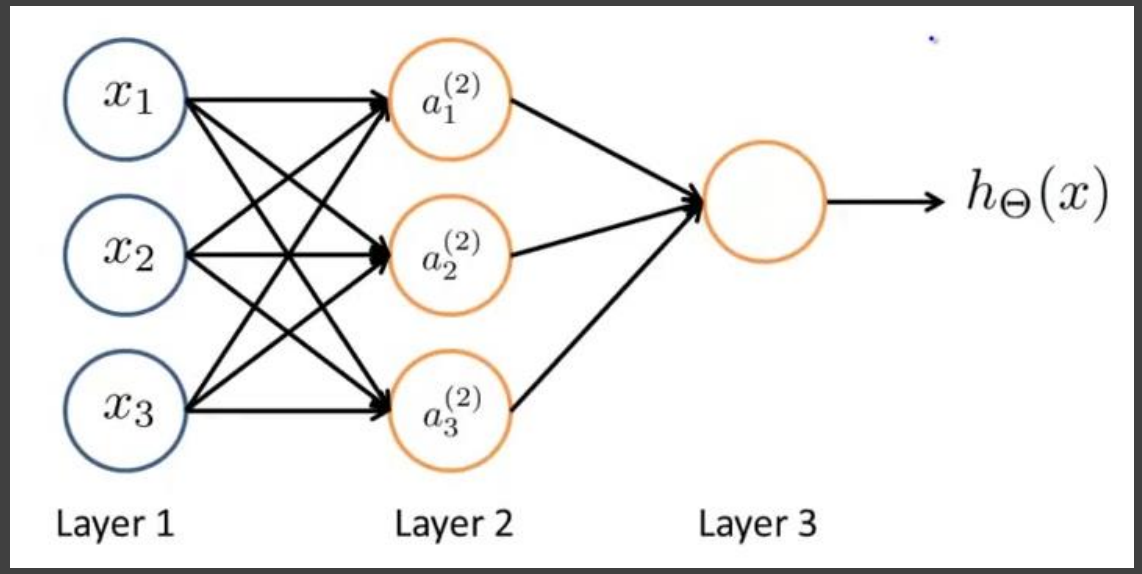
Artificial Neural Network

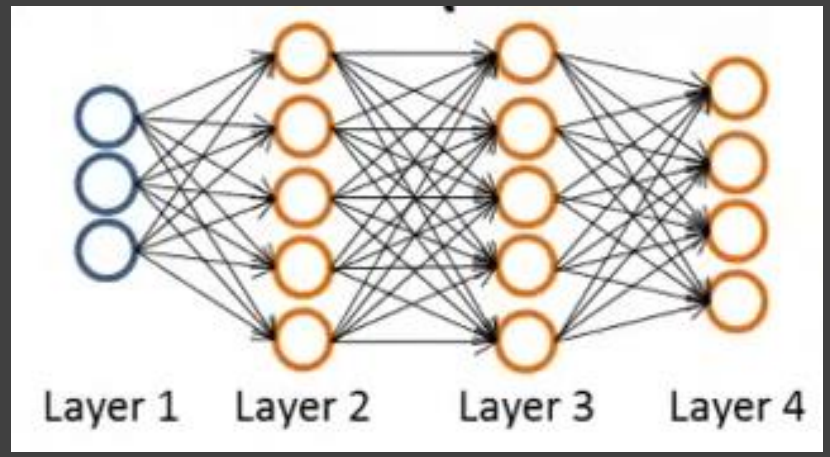
Sigmoid (logistic) activation function

- How much a neuron “fires”



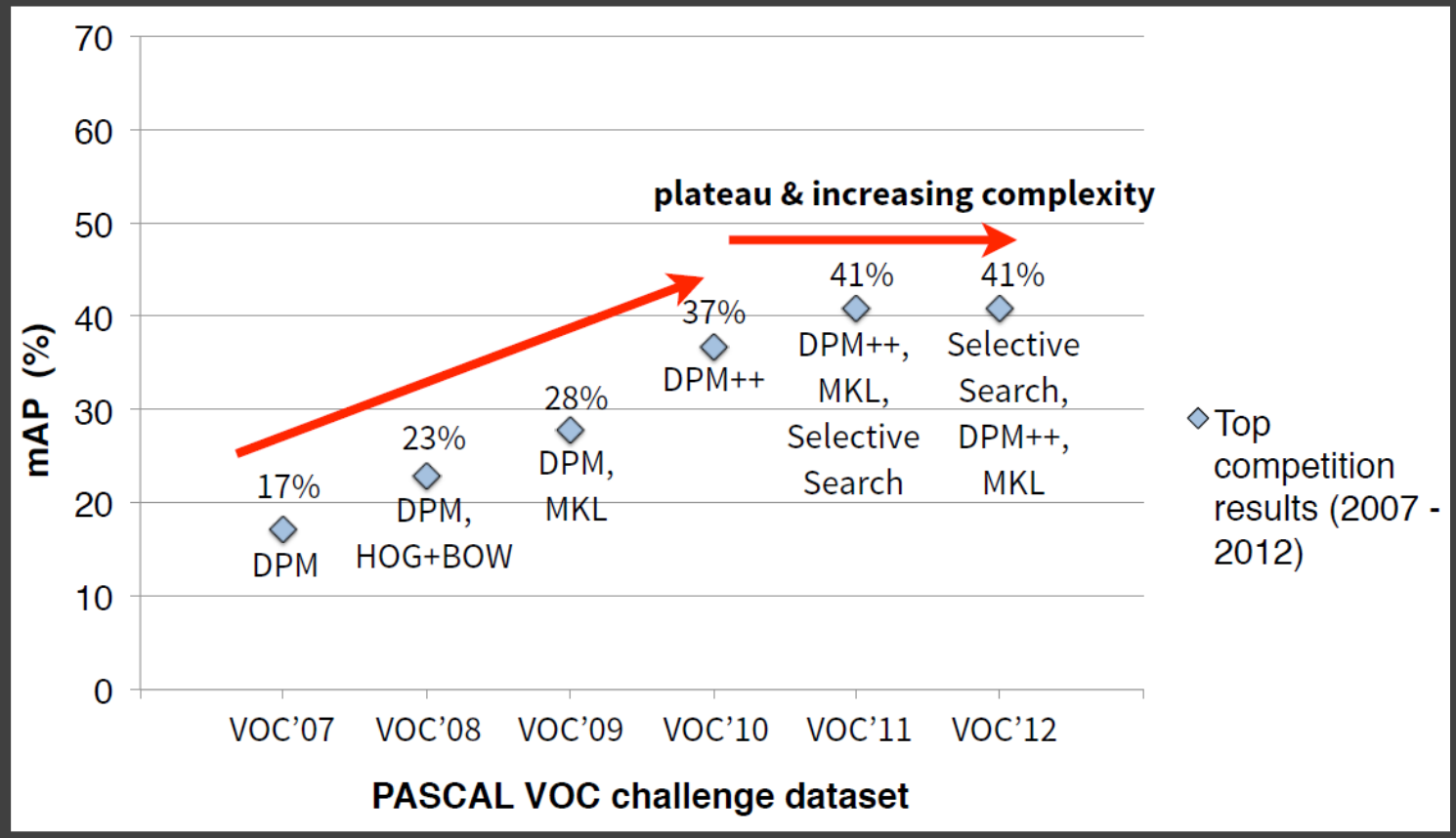
Artificial Neural Network



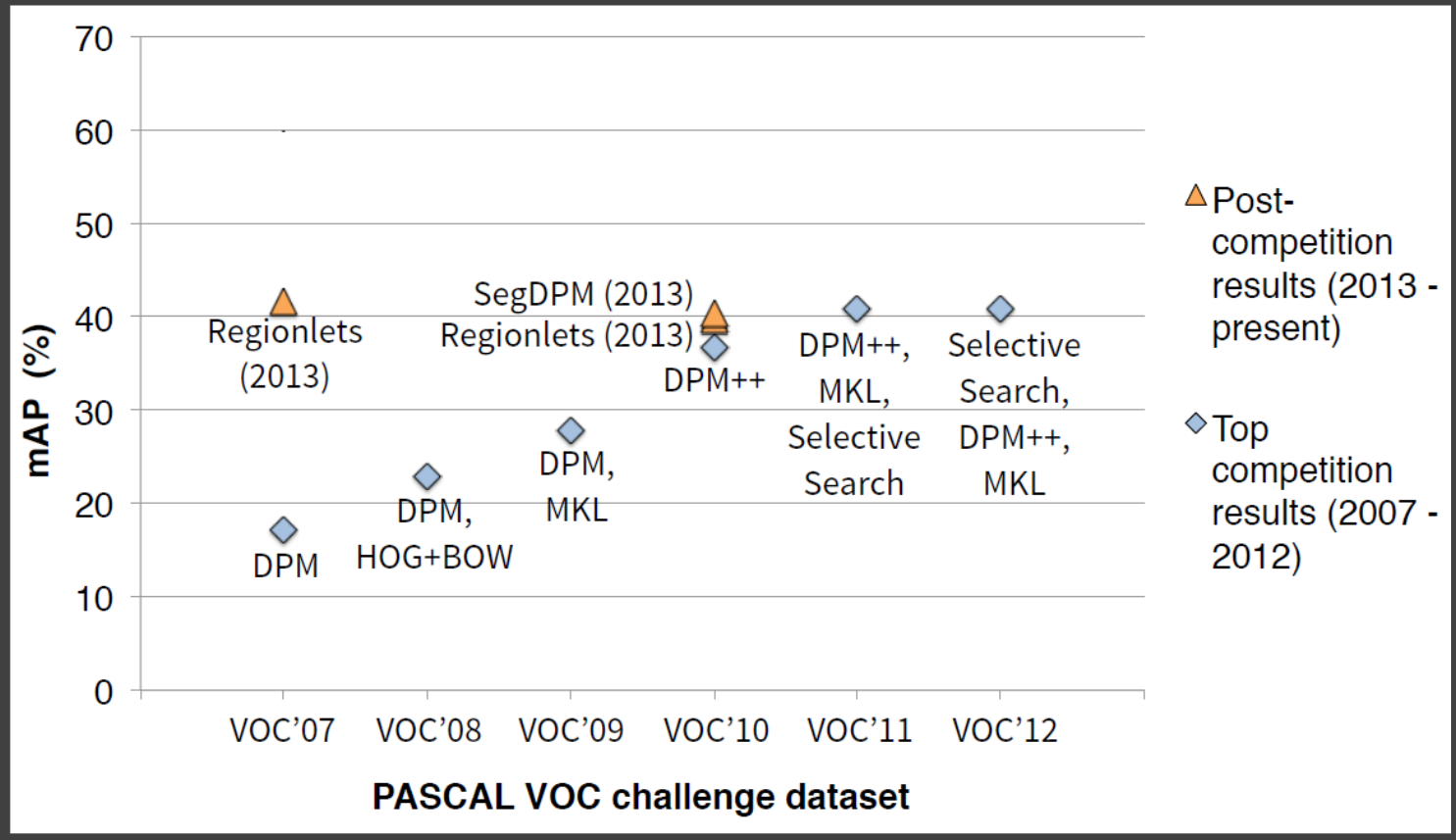


Previous Work

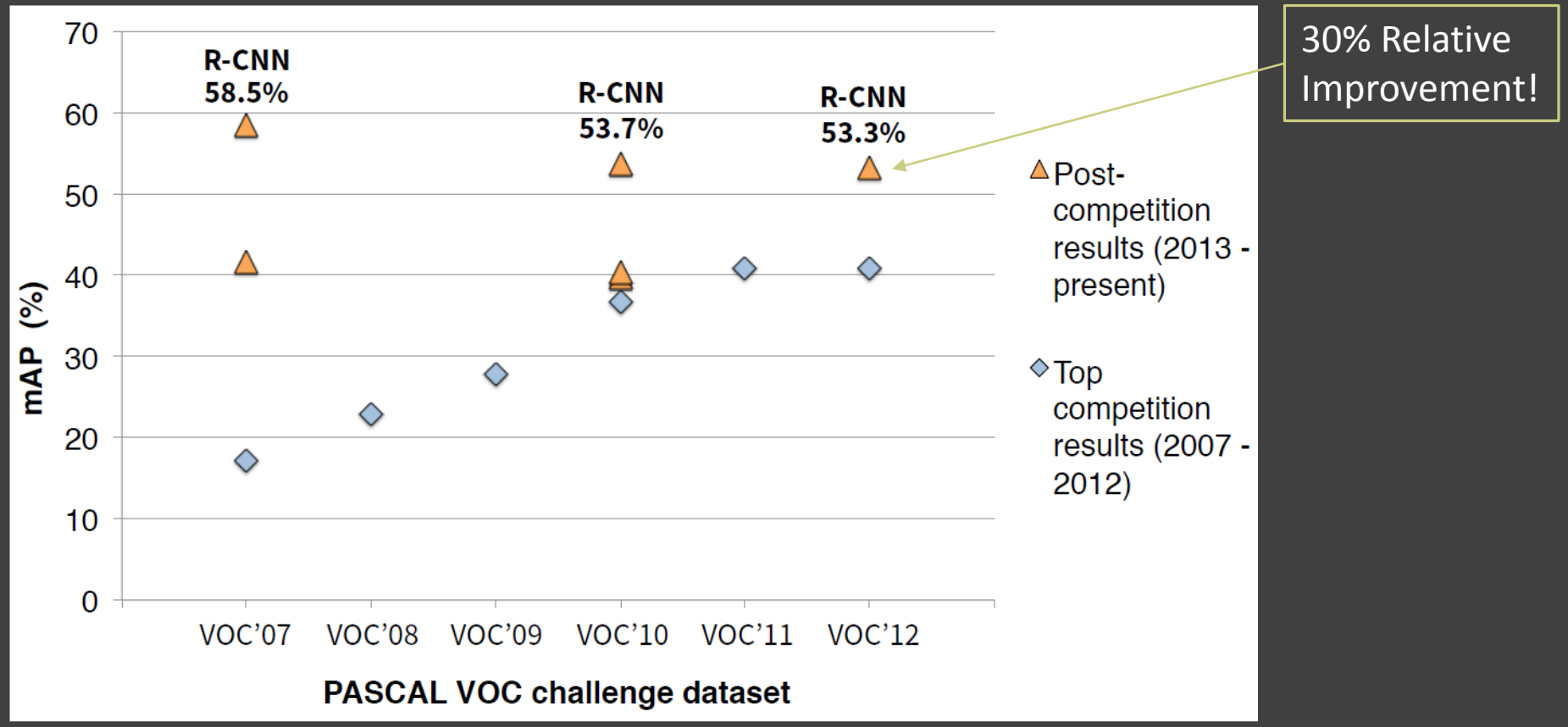
PASCAL VOC Challenge Results



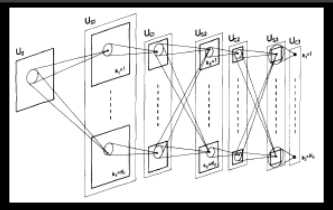
Post-Challenge Results (2013)



Regions with CNN Features (2014)

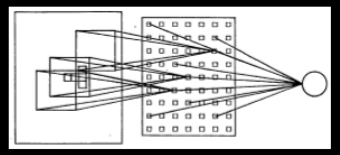


Previous Work on CNNs



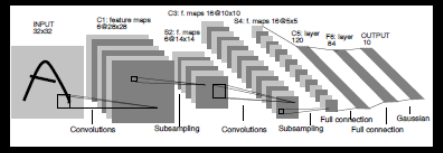
Fukushima 1980

Neocognitron



Rumelhart, Hinton, Williams 1986

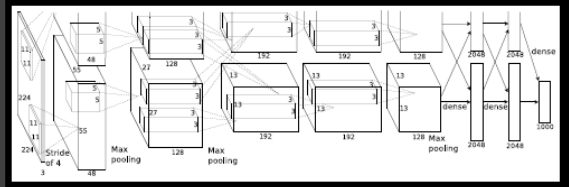
"T" versus "C" problem



LeCun et al. 1989-1998

Handwritten digit reading / OCR

...

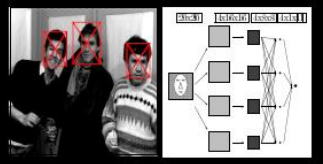


Krizhevksy, Sutskever, Hinton 2012

ImageNet classification breakthrough

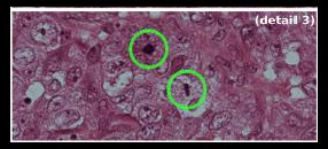
"SuperVision" CNN

Recent Work on CNNs for Object Detection



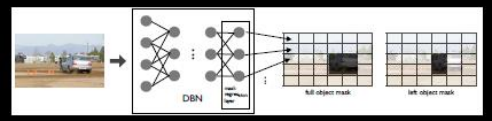
Vaillant, Monrocq, LeCun 1994
Multi-scale face detection

LeCun, Huang, Bottou 2004
NORB dataset



Cireşan et al. 2013
Mitosis detection

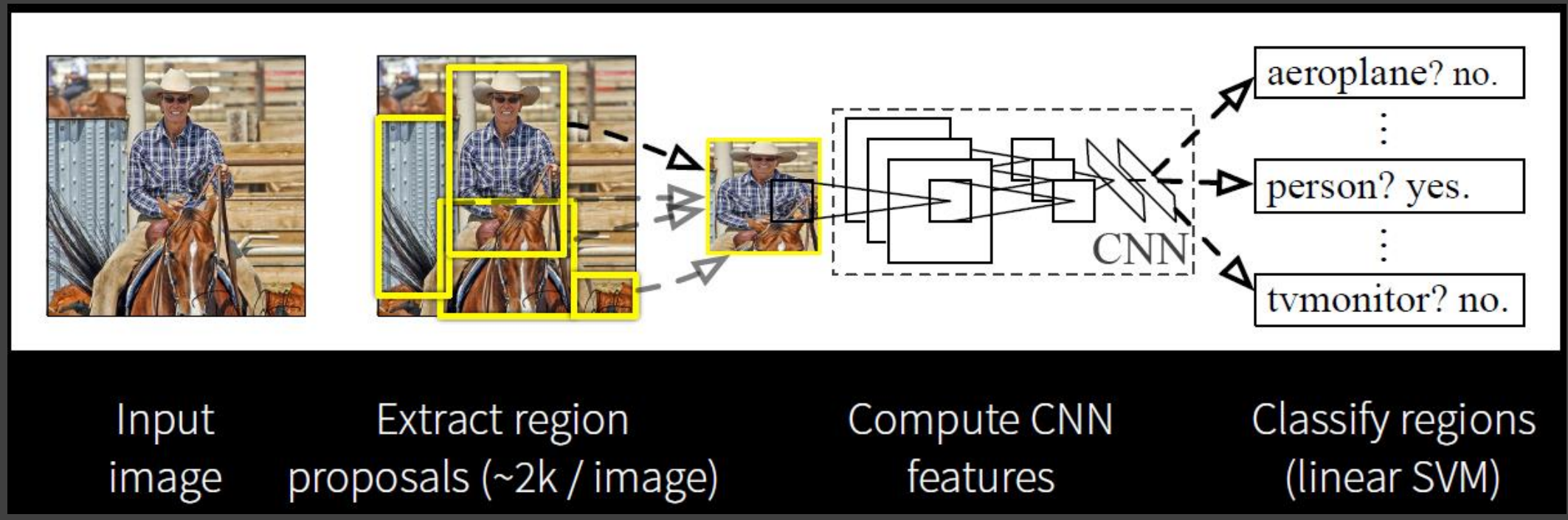
Sermanet et al. 2013
Pedestrian detection



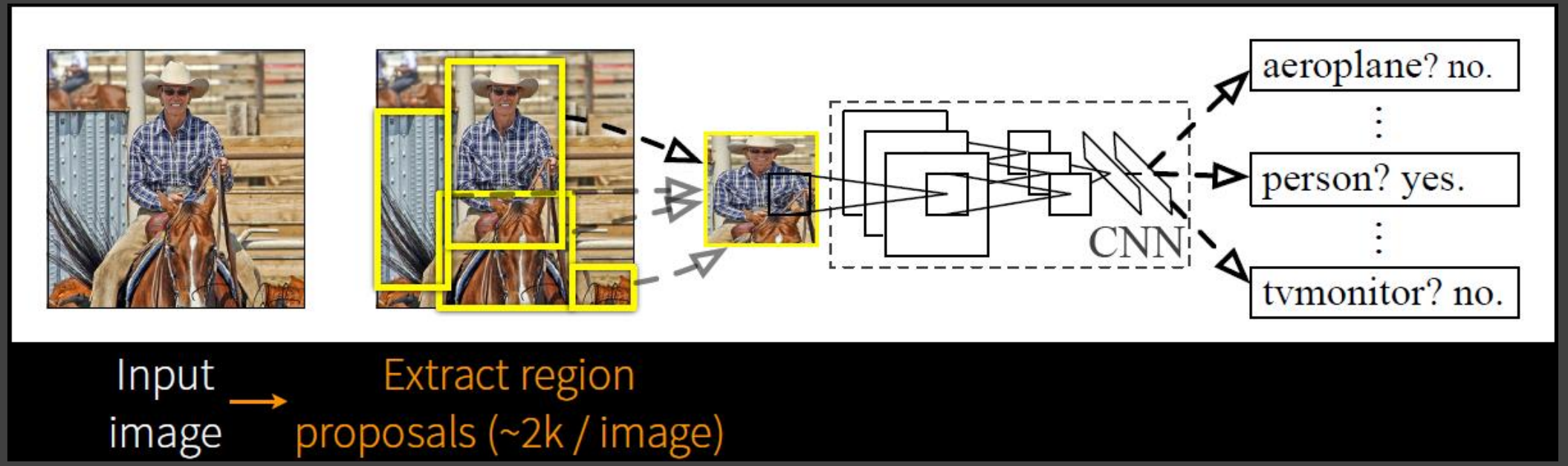
Szegedy, Toshev, Erhan 2013
PASCAL detection (VOC'07 mAP 30.5%)

Methods

Regions with CNN Features

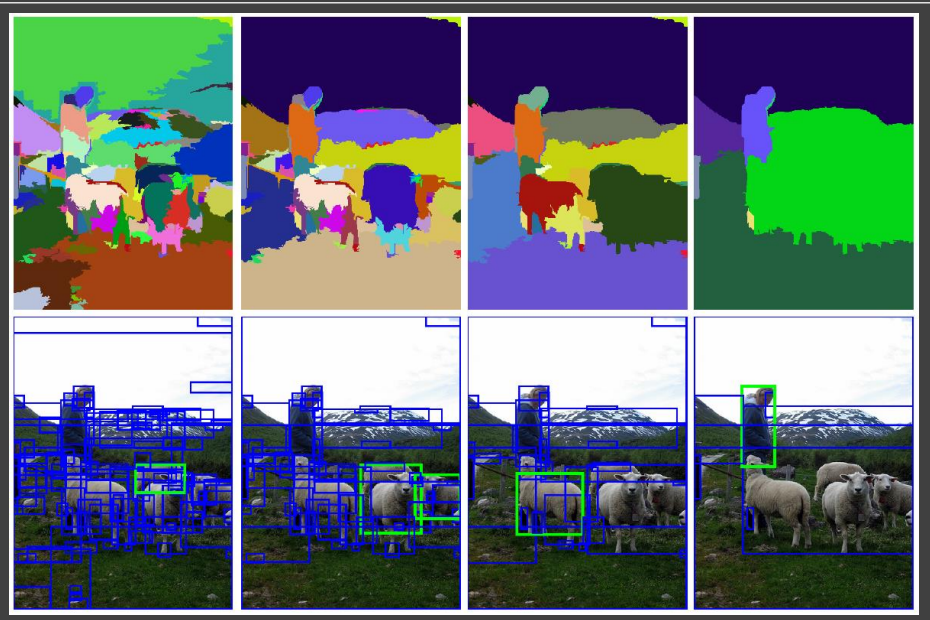


R-CNN: Step 1



Selective Search [van de Sande, Uijlings et al.]

Selective Search

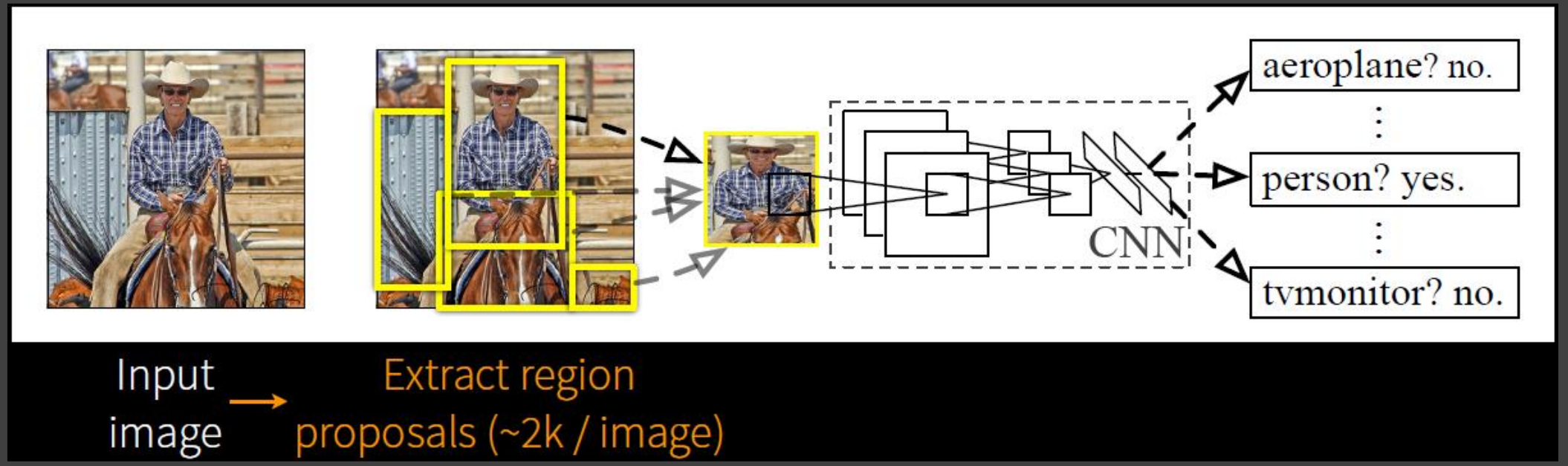


Approximate segmentation at multiple scales

- Capture more background
- Less expensive than exhaustive

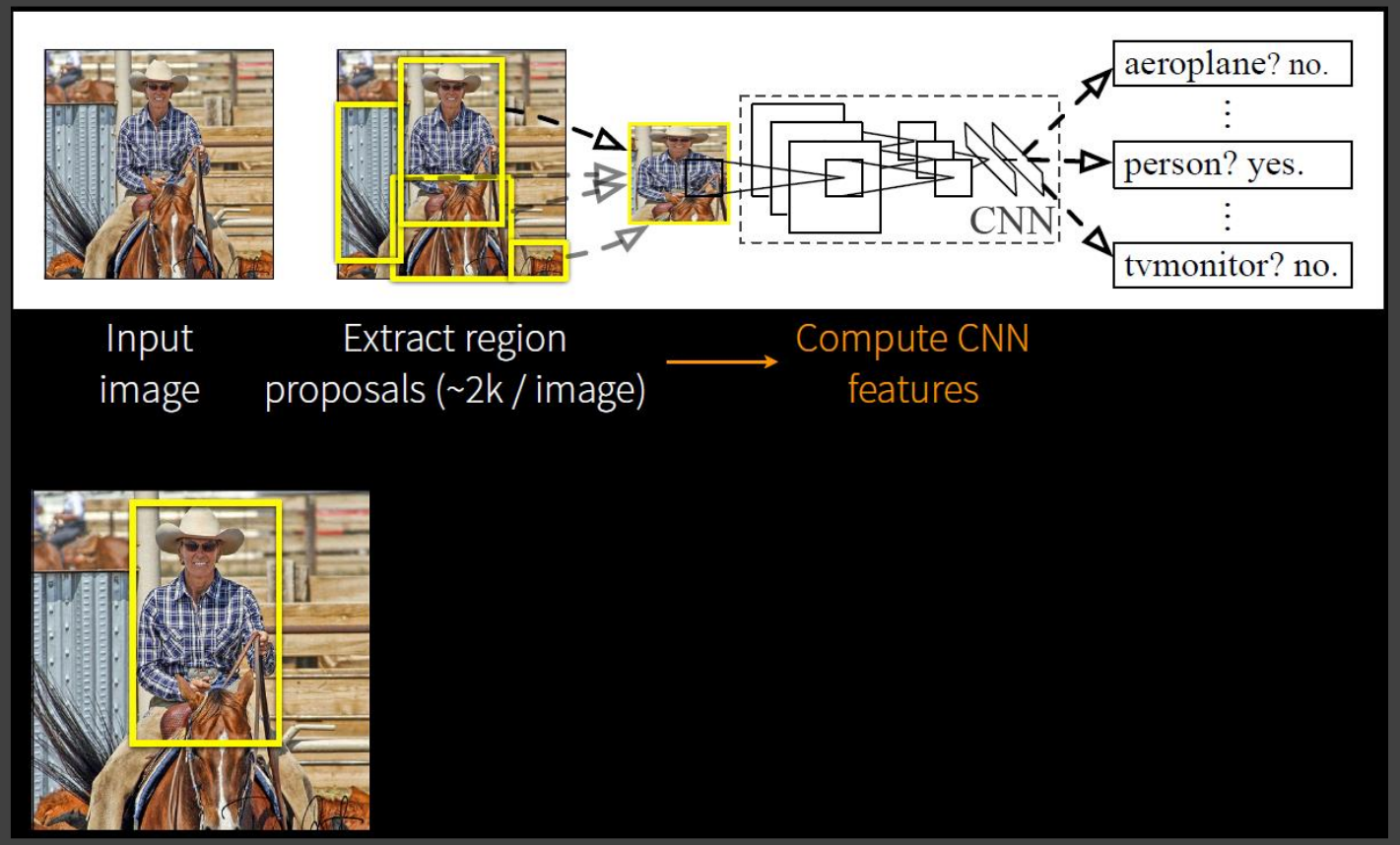
Uijlings, Jasper RR, et al. "Selective search for object recognition."*International journal of computer vision* 104.2 (2013): 154-171.

R-CNN: Step 1

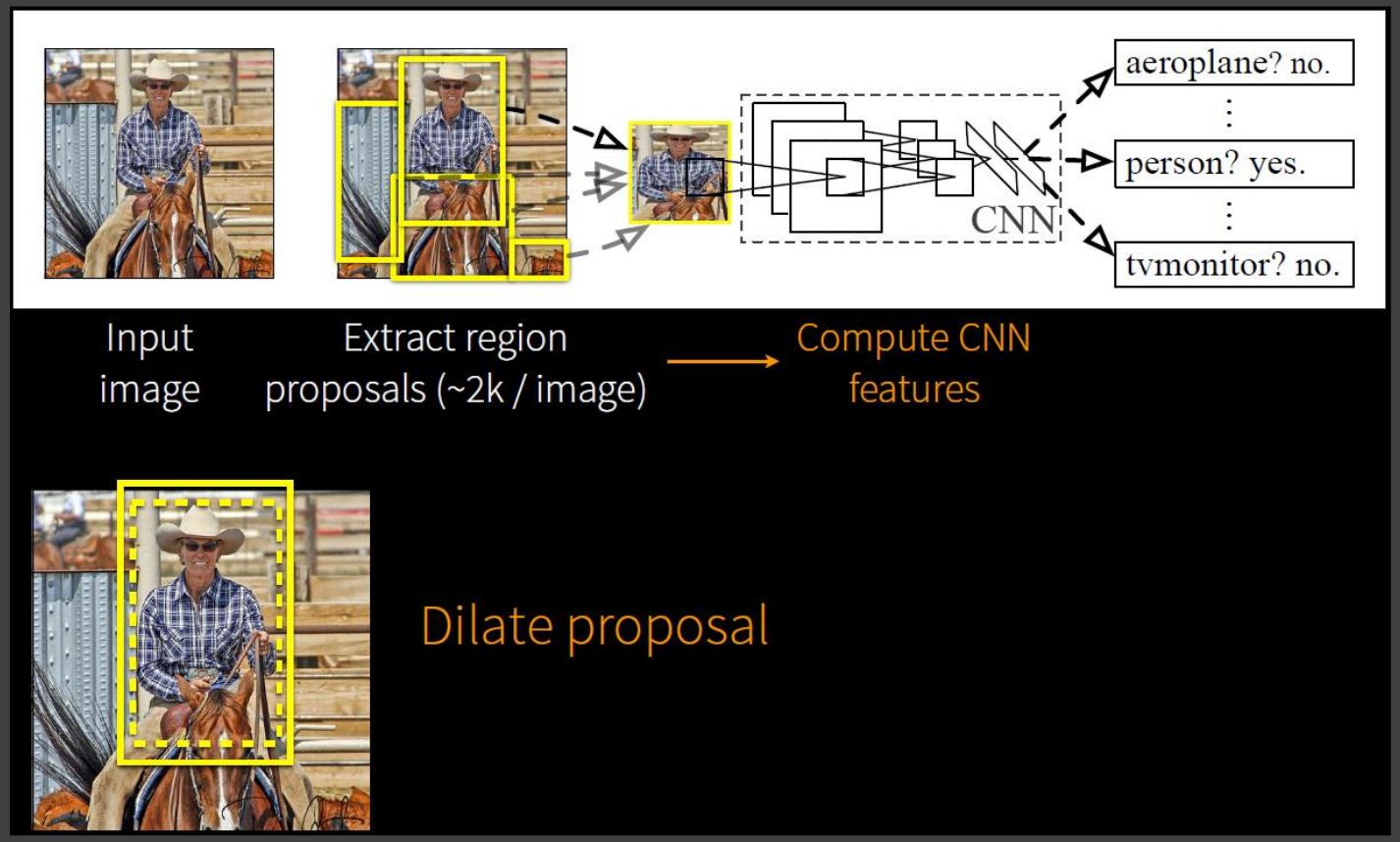


Selective Search [van de Sande, Uijlings et al.]

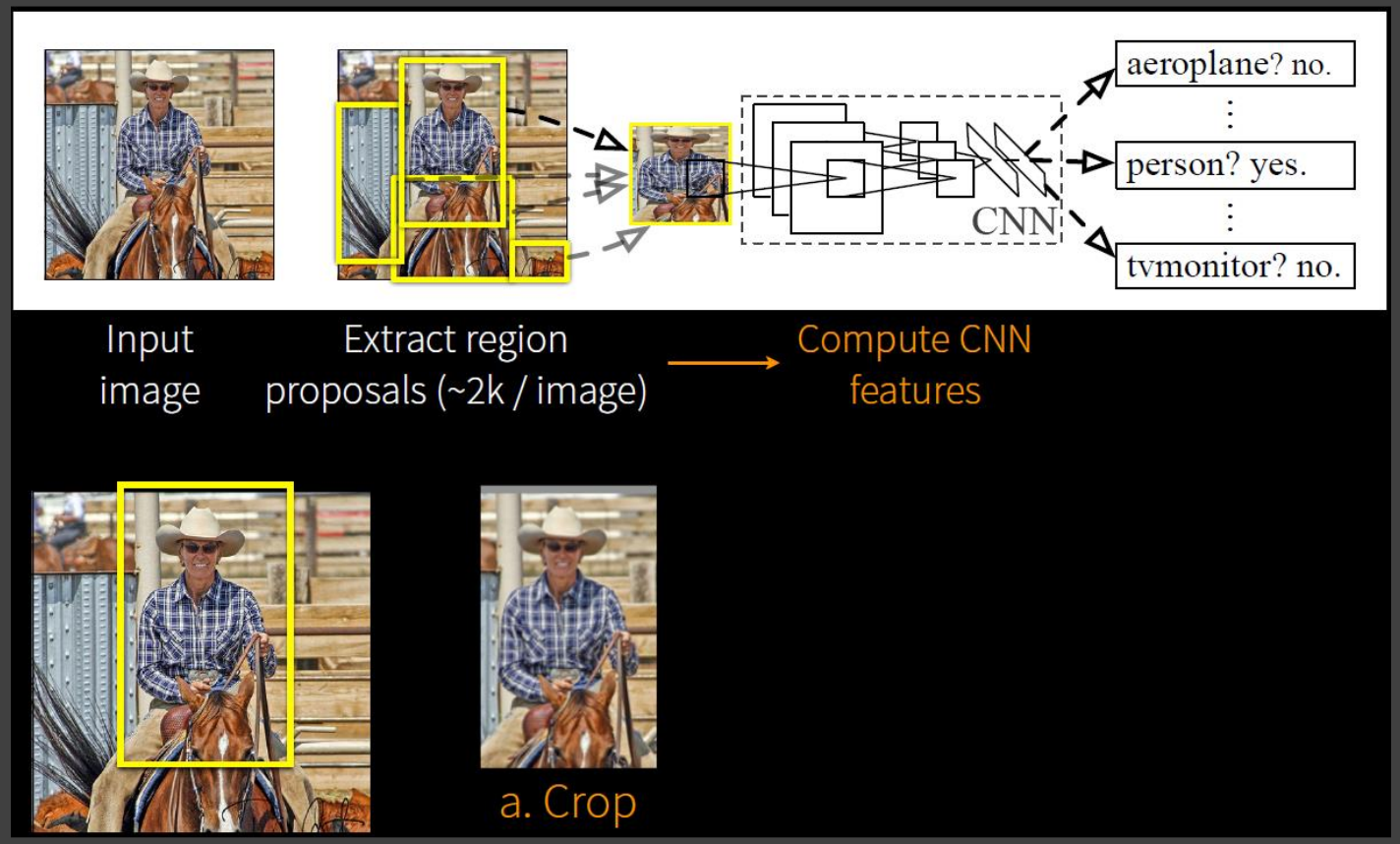
R-CNN: Step 2


aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features

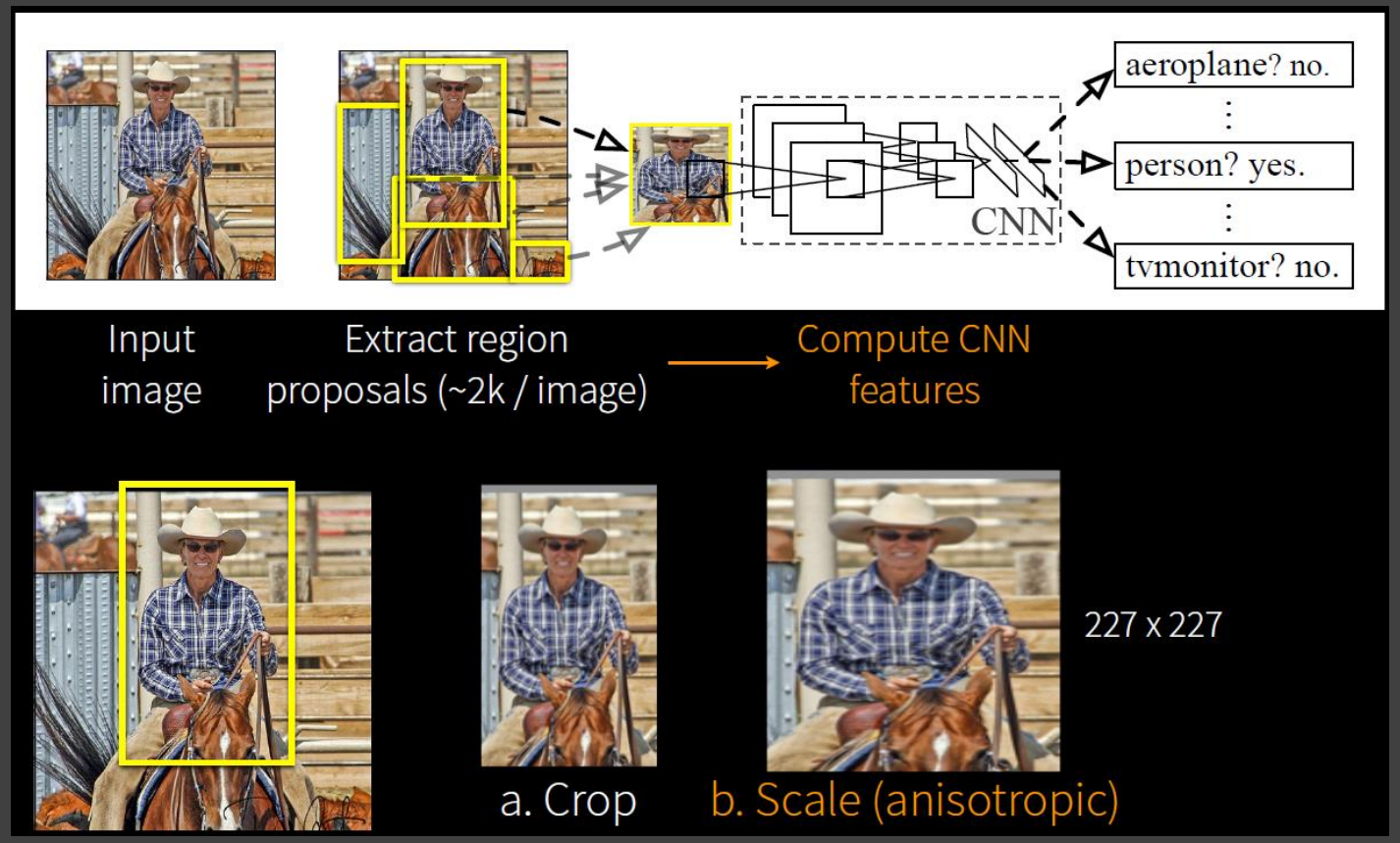
R-CNN: Step 2


aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features
Dilate proposal

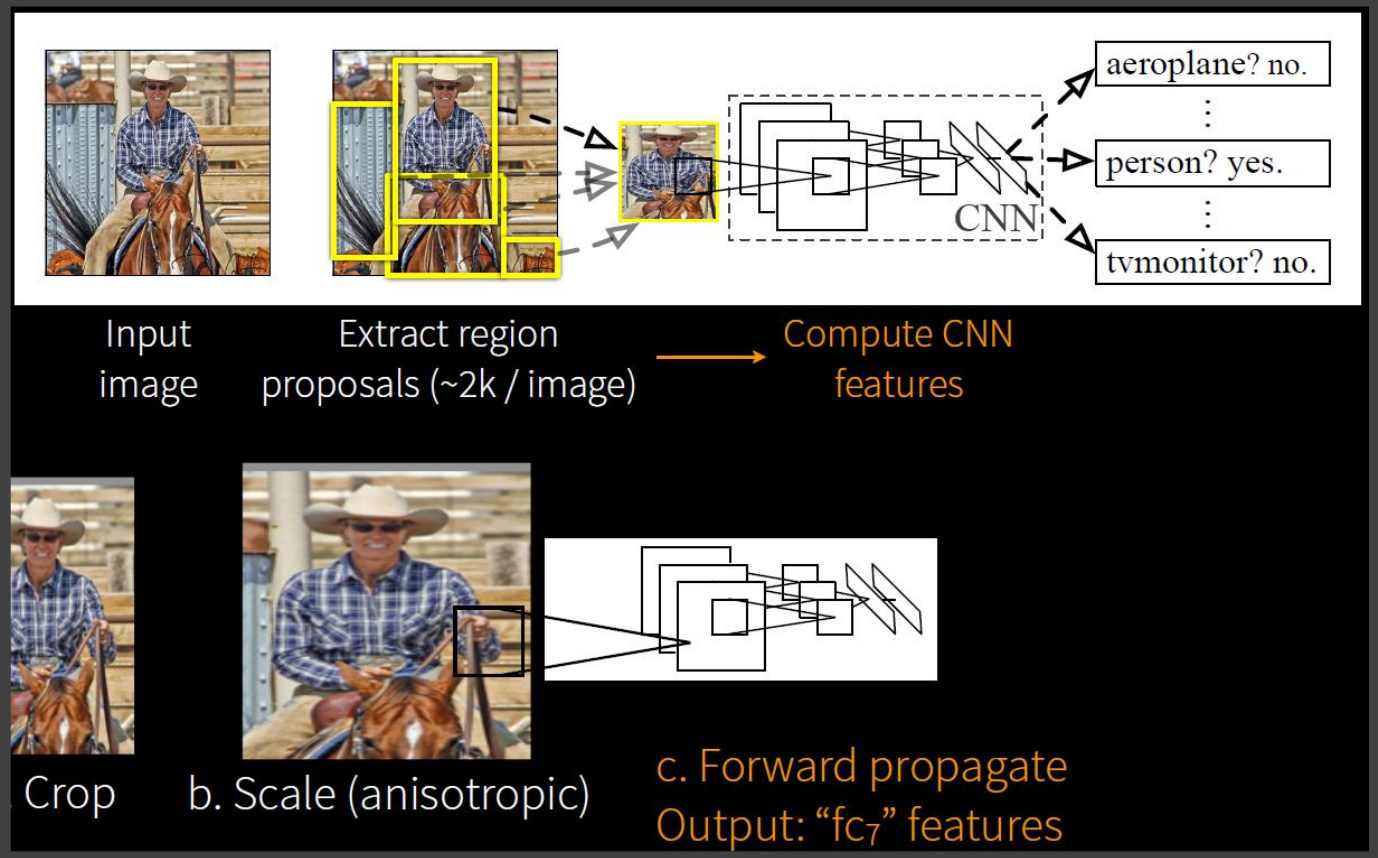
R-CNN: Step 2


aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features
a. Crop

R-CNN: Step 2

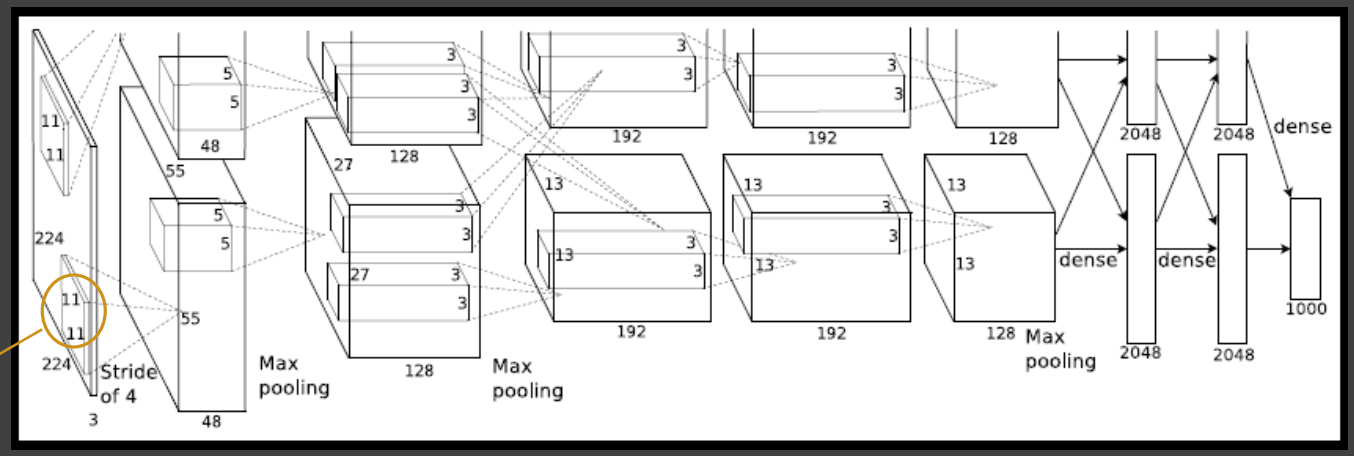

aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features
227 x 227
a. Crop
b. Scale (anisotropic)

R-CNN: Step 2


aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features
Crop
b. Scale (anisotropic)
c. Forward propagate
Output: "fc7" features

R-CNN: Step 2

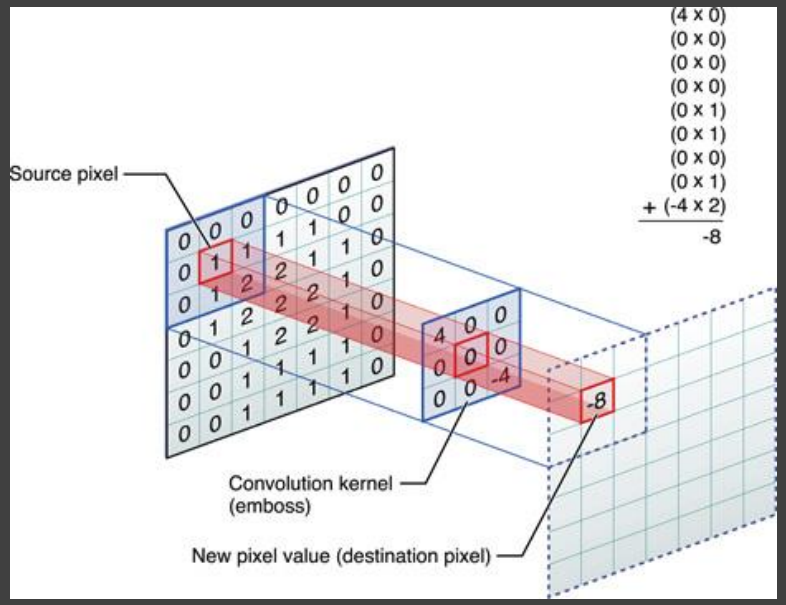
Forward Propagation



11x11 kernel size
w/ sliding window

R-CNN: Step 2

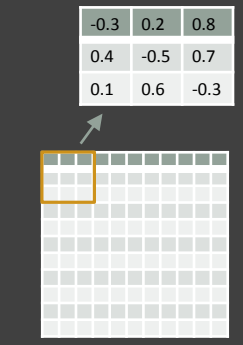
Kernel Convolution



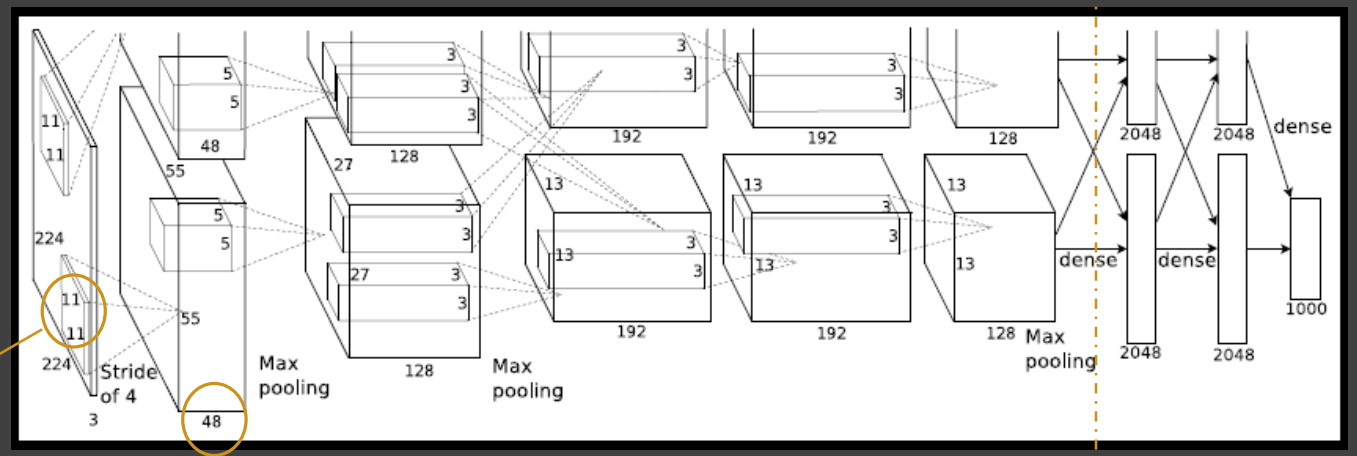
https://developer.apple.com/library/mac/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

R-CNN: Step 2

Forward Propagation



11x11 kernel size
w/ sliding window



48 kernels
(feature channels)

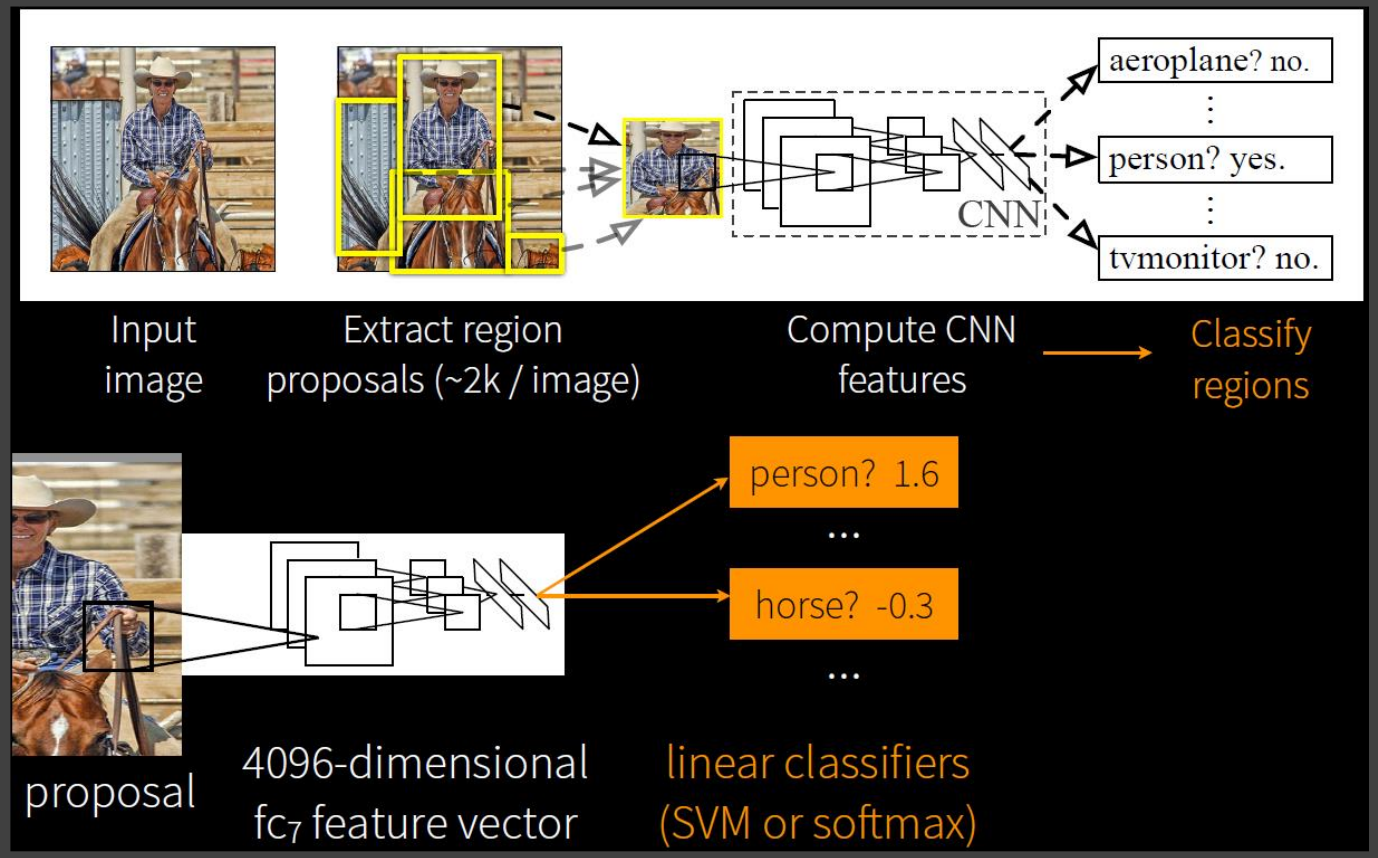
Feed-forward Convolutional Neural Net

Fully-connected Artificial Neural Net

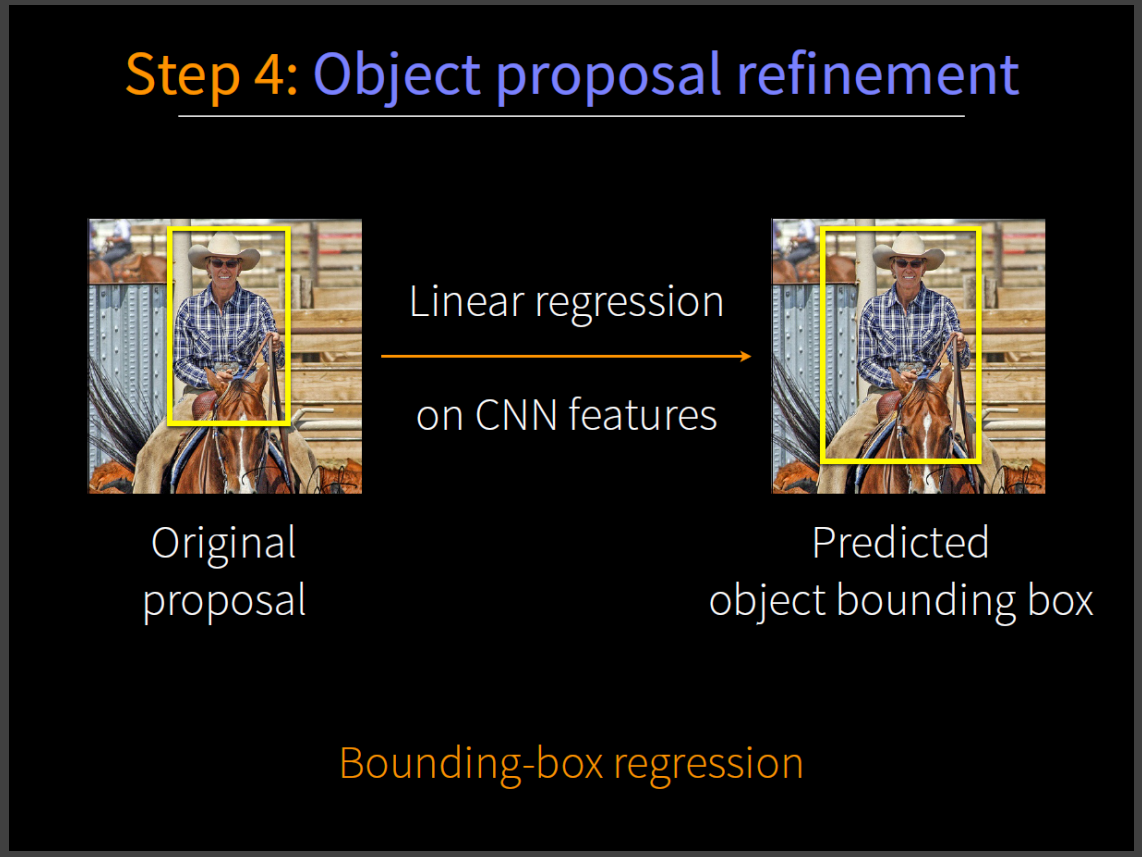
At each stage

- Higher level features, from convolution alone!
- Max pooling keeps best features
- Convolution kernels learned from training

R-CNN: Step 3


aeroplane? no.
person? yes.
tvmonitor? no.
CNN
Input image
Extract region proposals (~2k / image)
Compute CNN features
Classify regions
person? 1.6
horse? -0.3
proposal
4096-dimensional fc_7 feature vector
linear classifiers (SVM or softmax)

R-CNN: Step 4


Step 4: Object proposal refinement
Linear regression
on CNN features
Original proposal
Predicted object bounding box
Bounding-box regression

R-CNN: Step 4

Predicting Object Bounding Box

Ground Truth
Bounding Box

R-CNN: Step 4

Predicting Object Bounding Box

Ground Truth
Bounding Box

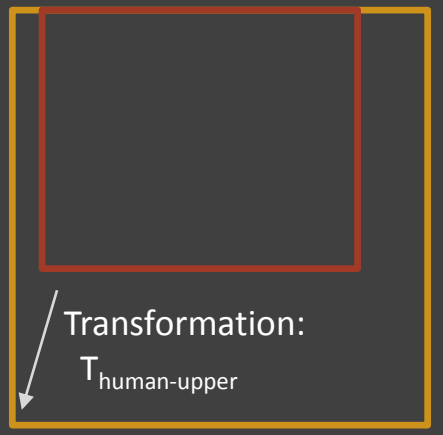
Features:
$F_{\text{human-upper}}$

R-CNN: Step 4

Predicting Object Bounding Box

Ground Truth
Bounding Box

Features:
$F_{human\text{-}upper}$



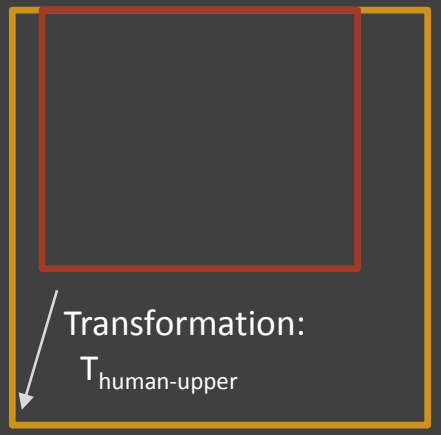
R-CNN: Step 4

Predicting Object Bounding Box

Ground Truth Bounding Box

Features: $F_{human\text{-}upper}$

Features: $F_{human\text{-}lower}$

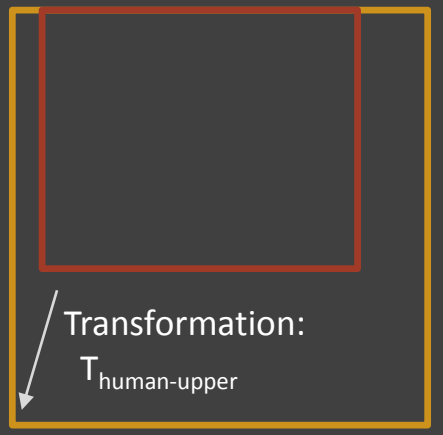


R-CNN: Step 4

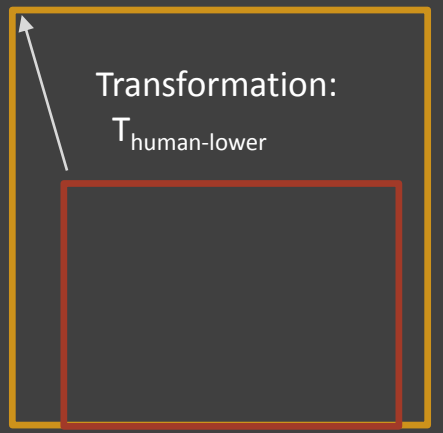
Predicting Object Bounding Box

Ground Truth Bounding Box

Features: $F_{human\text{-}upper}$



Features: $F_{human\text{-}lower}$

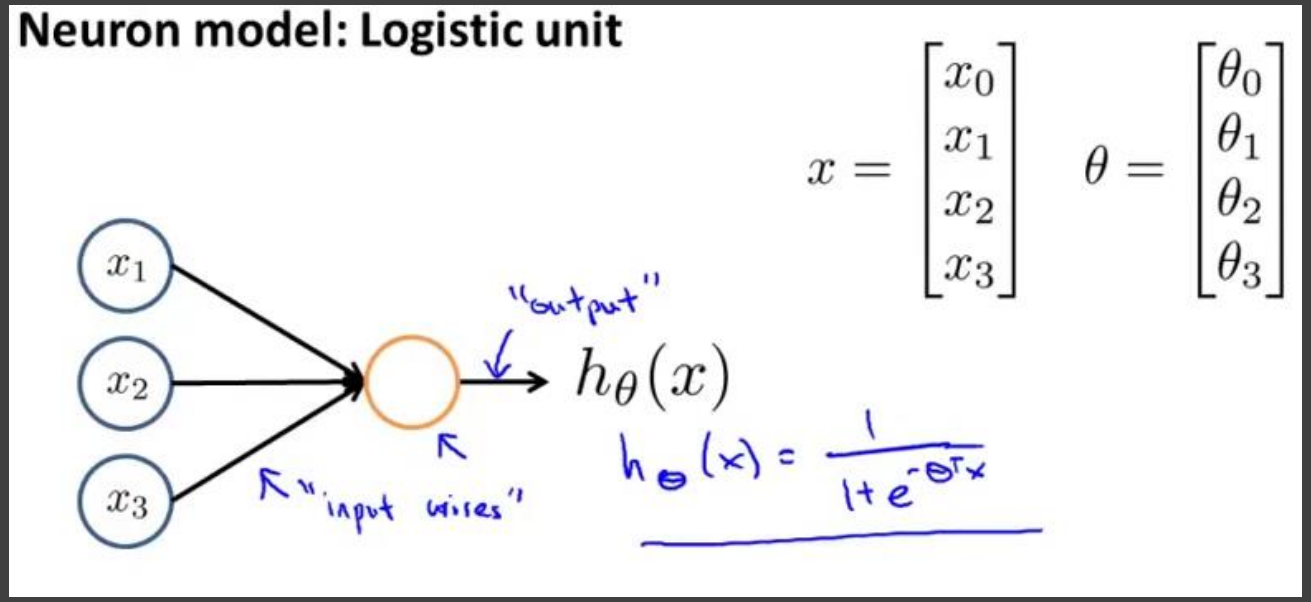


Training

Train What?

Convolution kernels

- To minimize a cost function
- Update kernels after every training image



Cost Function

Cost function

Logistic regression:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

Neural network:

$$h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$

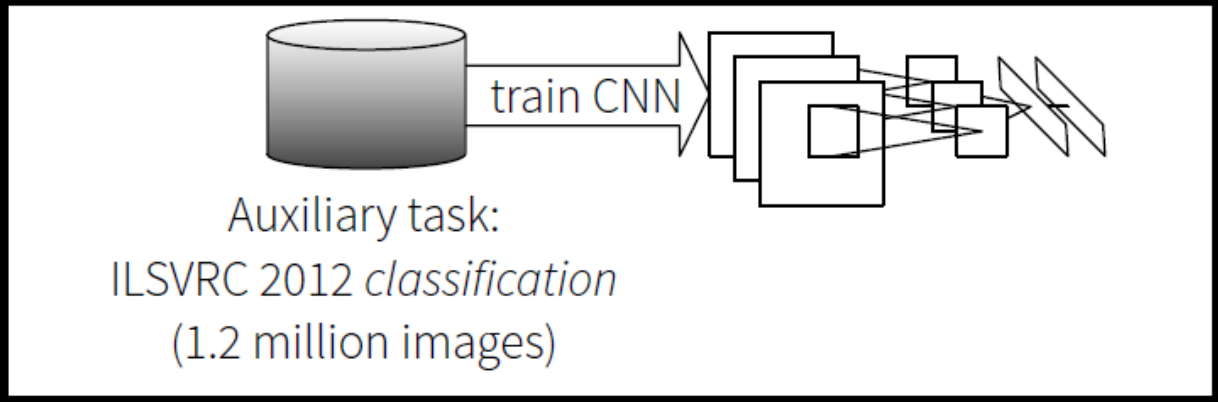
$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log((h_\Theta(x^{(i)}))_k + (1-y_k^{(i)}) \log(1-(h_\Theta(x^{(i)}))_k)\right]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

R-CNN Training: Step 1

Supervised pre-training

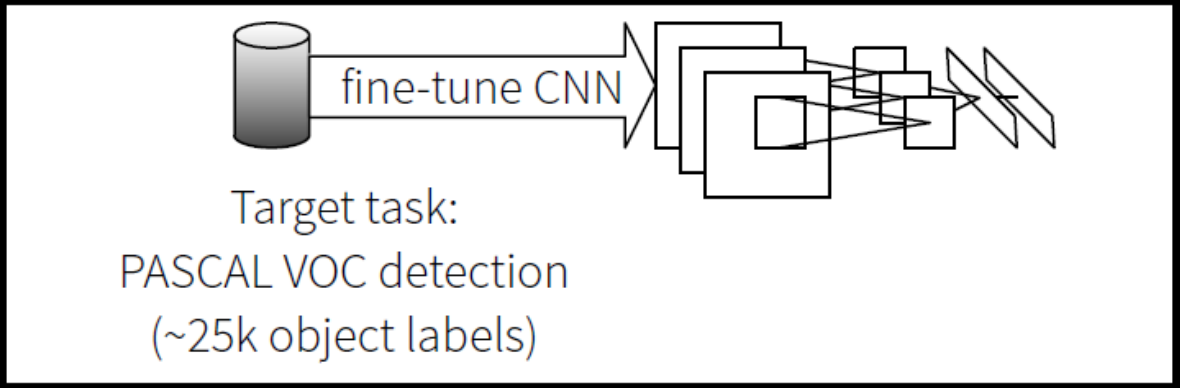
Train a SuperVision CNN* for the 1000-way ILSVRC image classification task



R-CNN Training: Step 2

Fine-tune the CNN for detection

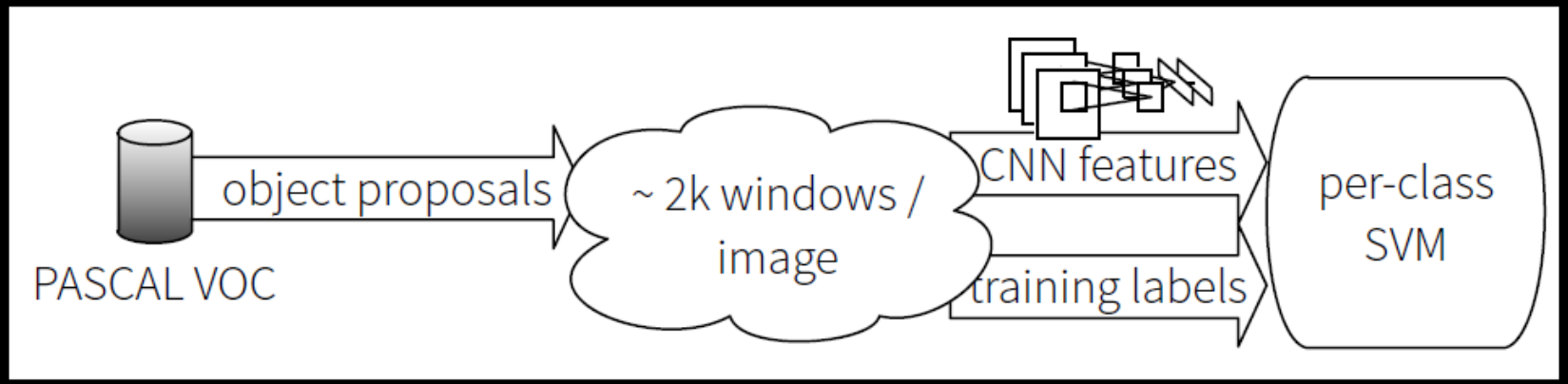
Transfer the representation learned for ILSVRC classification to PASCAL (or ImageNet detection)



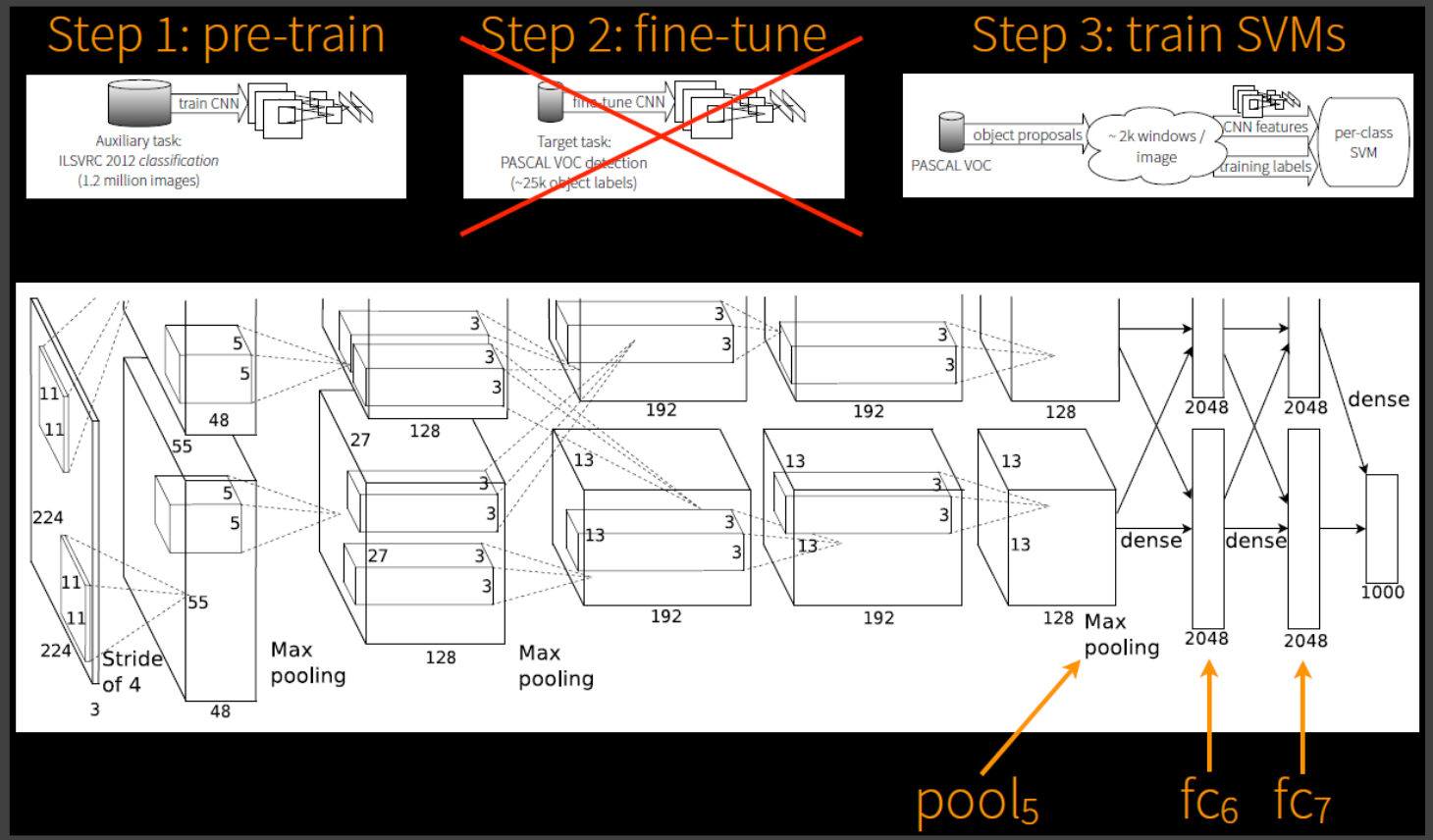
R-CNN Training: Step 3

Train detection SVMs

(With the softmax classifier from fine-tuning mAP decreases from 54% to 51%)



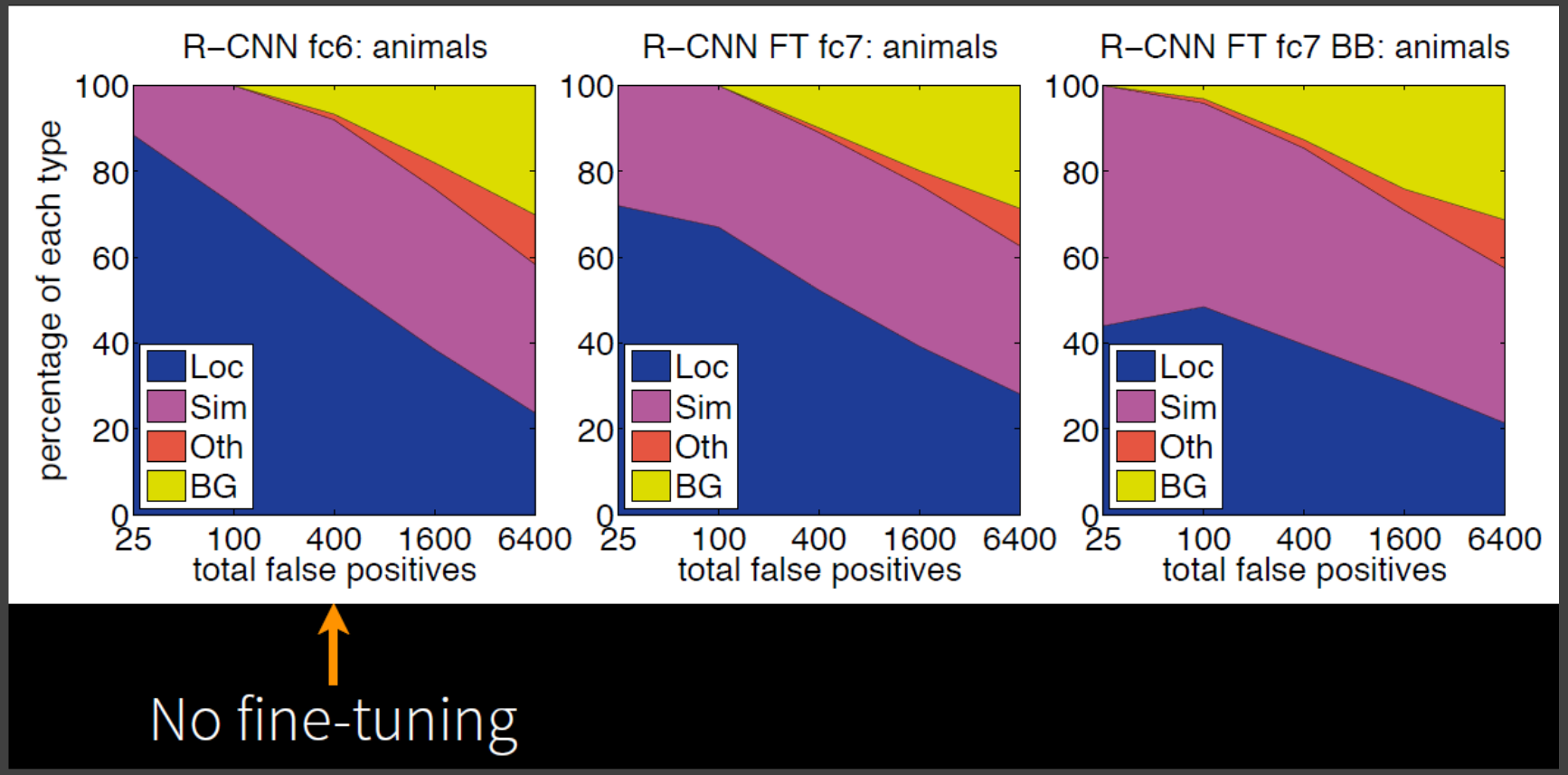
Tuning: Worth it?



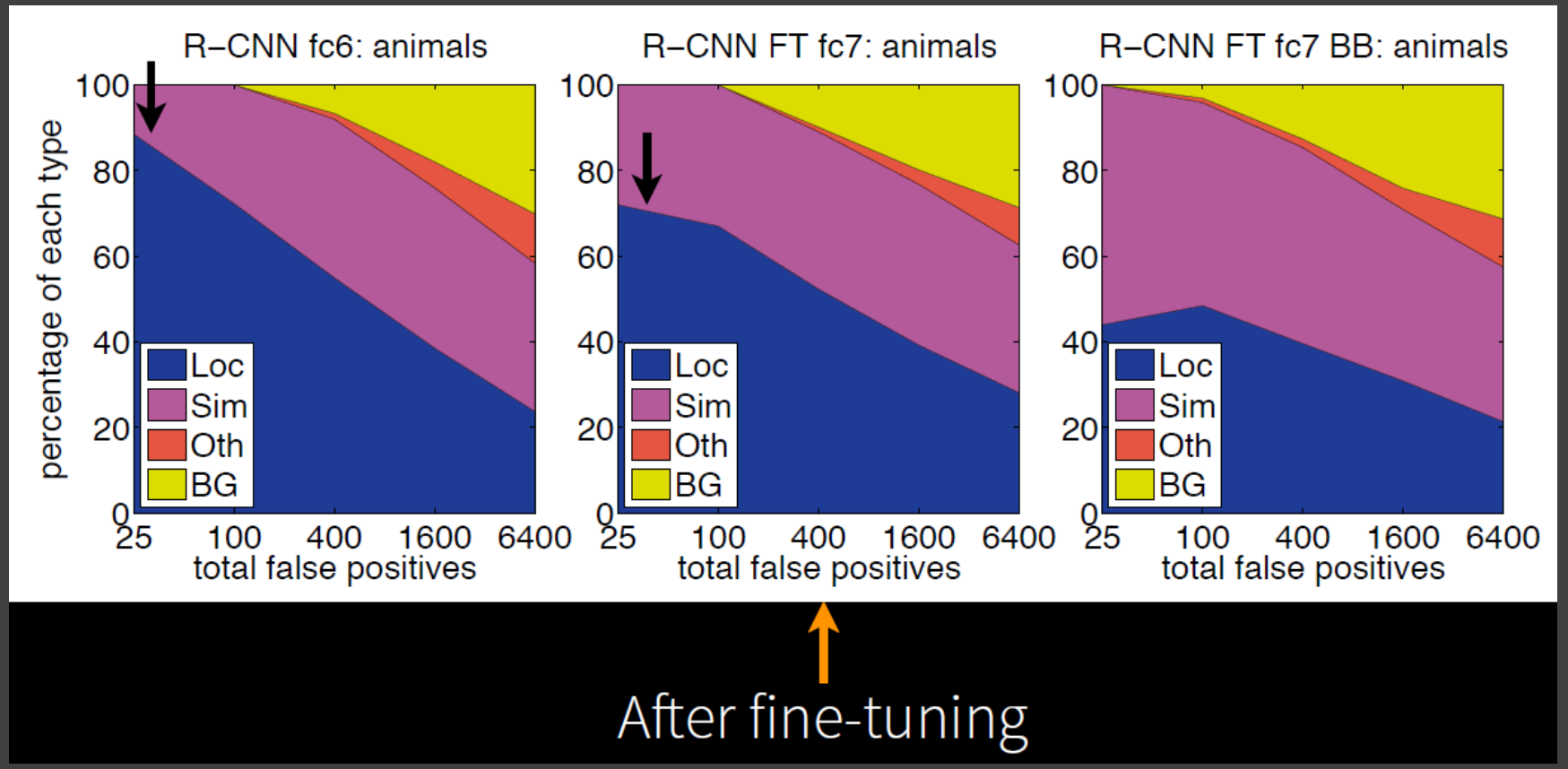
Tuning: Worth it?

fine-tuned		VOC 2007	VOC 2010
	Regionlets (Wang et al. 2013)	41.7%	39.7%
	SegDPM (Fidler et al. 2013)		40.4%
	R-CNN $pool_5$	44.2%	
	R-CNN fc_6	46.2%	
	R-CNN fc_7	44.7%	
✓	R-CNN FT $pool_5$	47.3%	
✓	R-CNN FT fc_6	53.1%	
✓	R-CNN FT fc_7	**54.2%**	**50.2%**

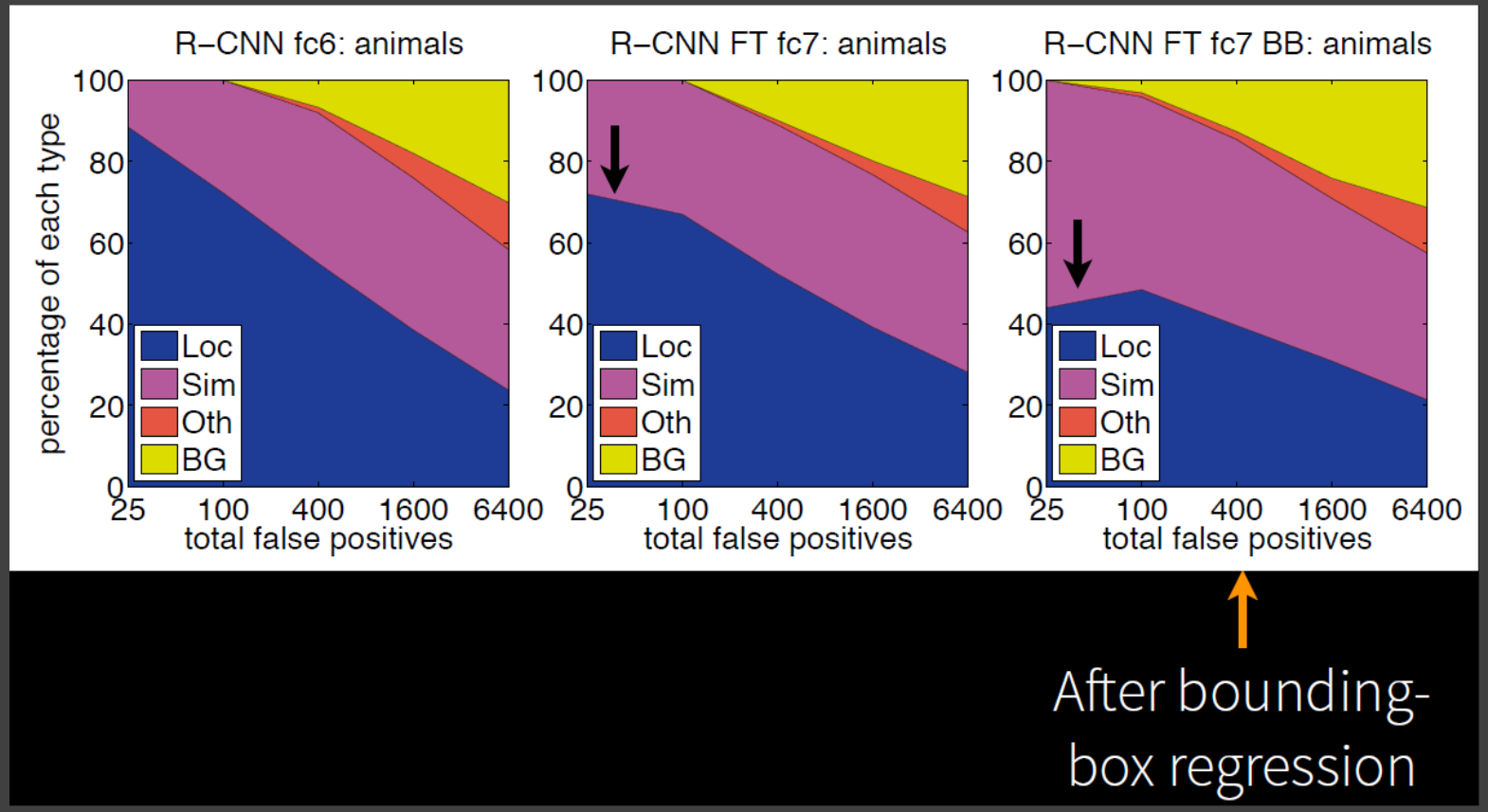
Tuning: Worth it?



Tuning: Worth it?



Tuning: Worth it?



Performance

R-CNN, PASCAL Results

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
SegDPM (Fidler et al. 2013)		40.4%
R-CNN	54.2%	50.2%
R-CNN + bbox regression	**58.5%**	**53.7%**

ImageNet Detection

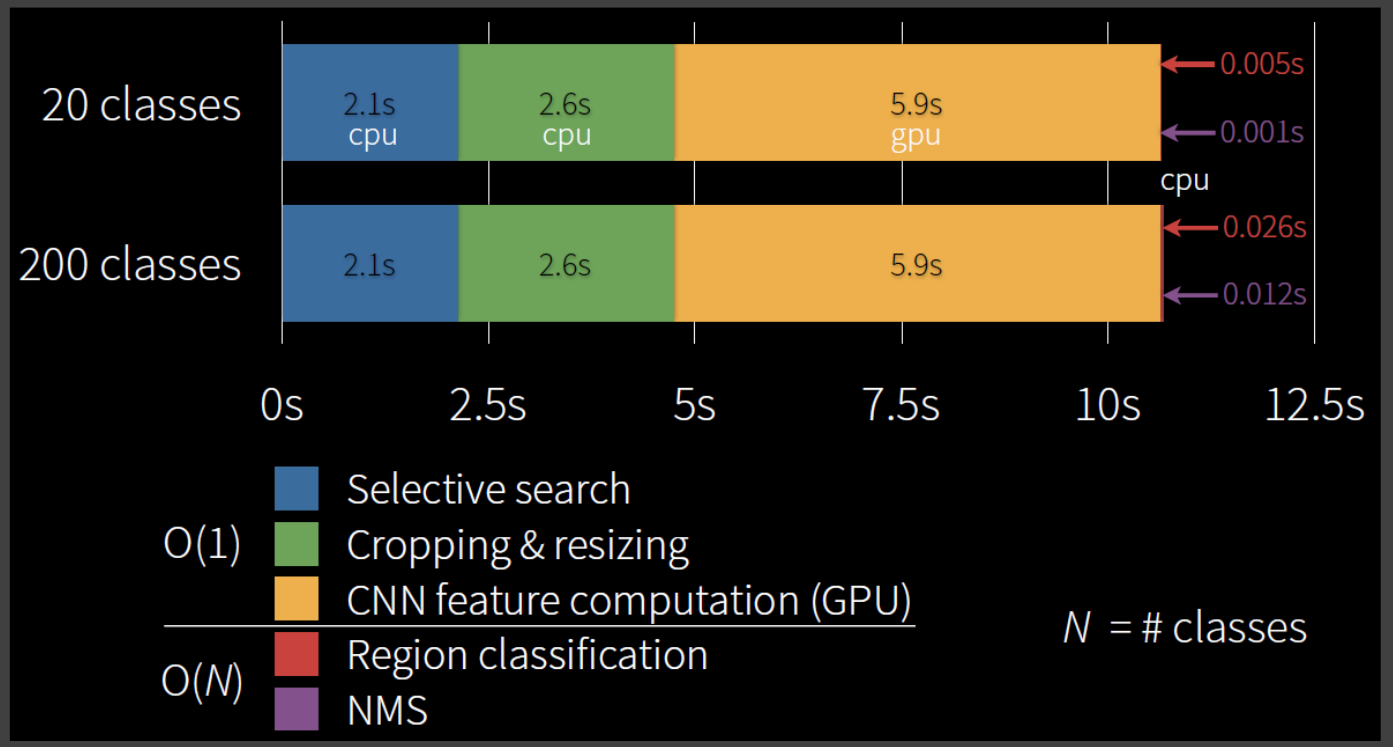
200 object categories instead of *20*

- Can this approach work?

ImageNet Detection

200 object categories instead of *20*

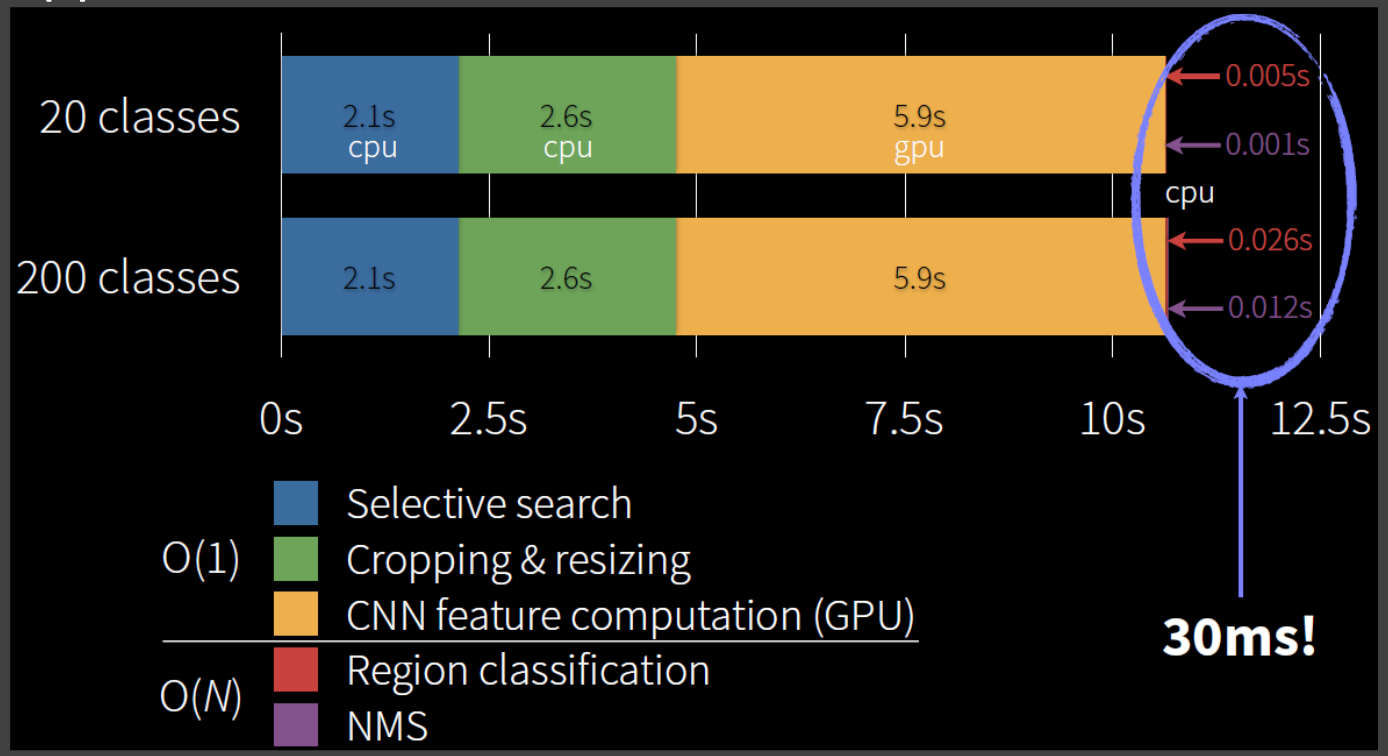
- Can this approach work? Yes!



ImageNet Detection

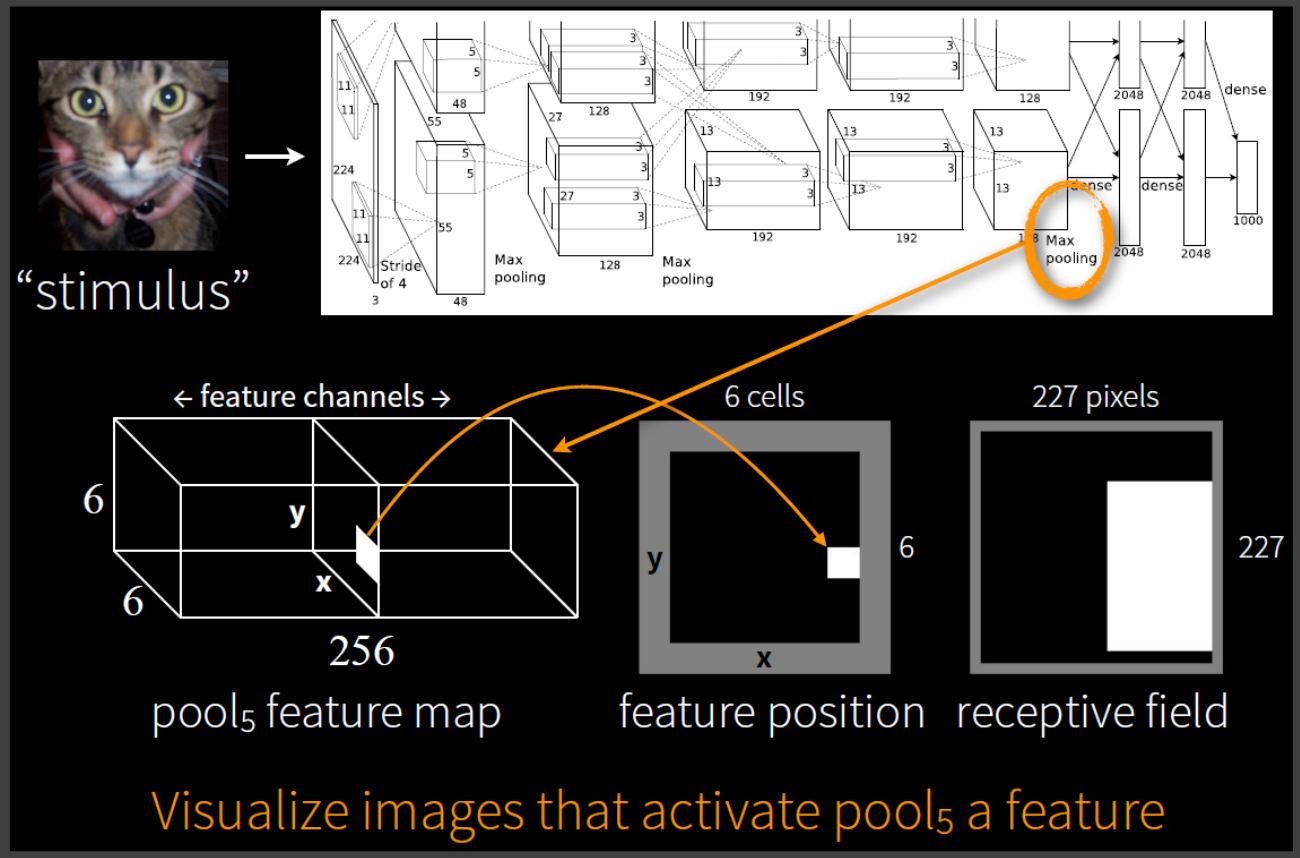
200 object categories instead of *20*

- Can this approach work? Yes!

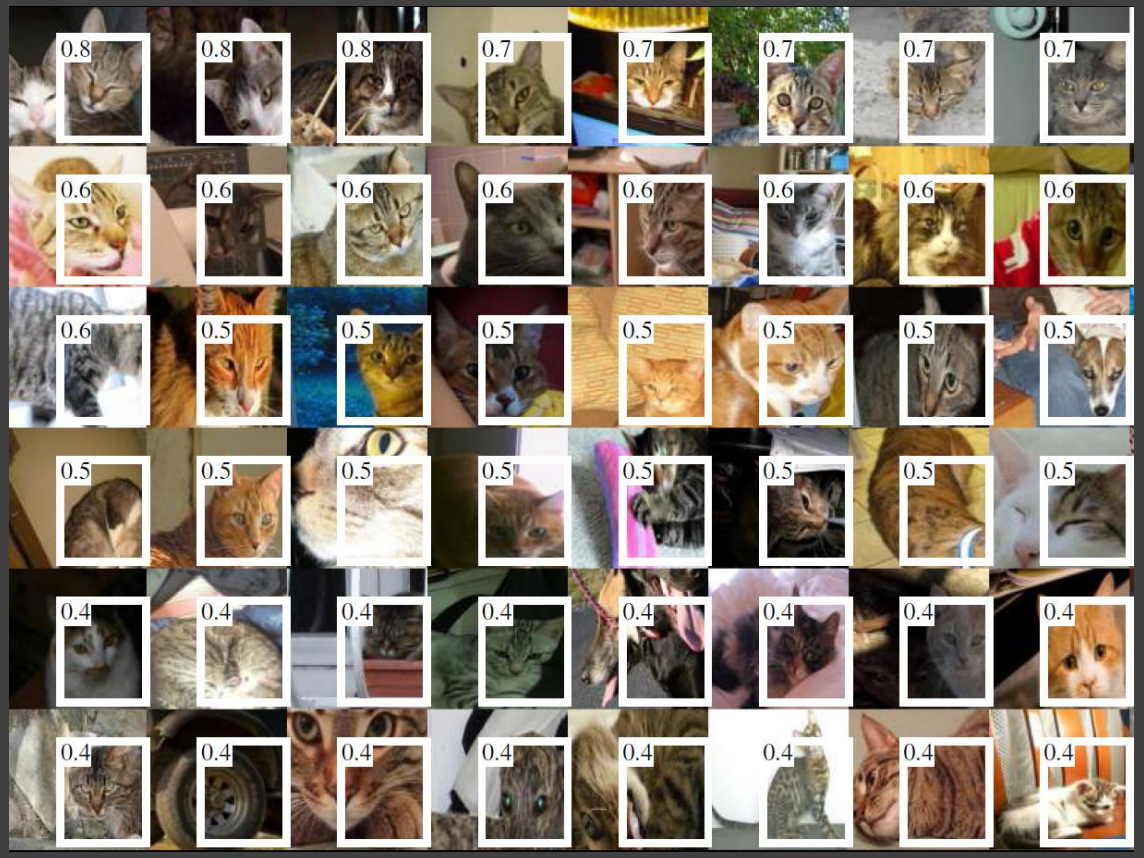


Results

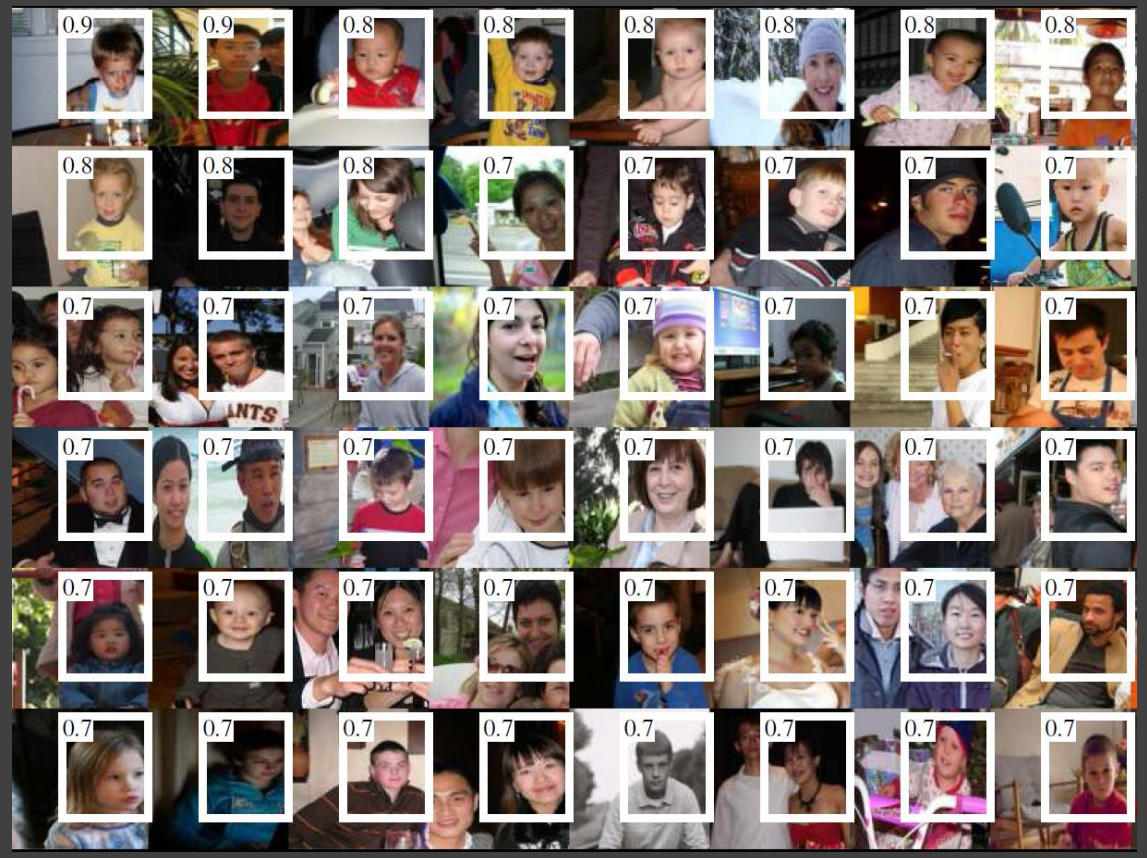
What did the CNN learn?



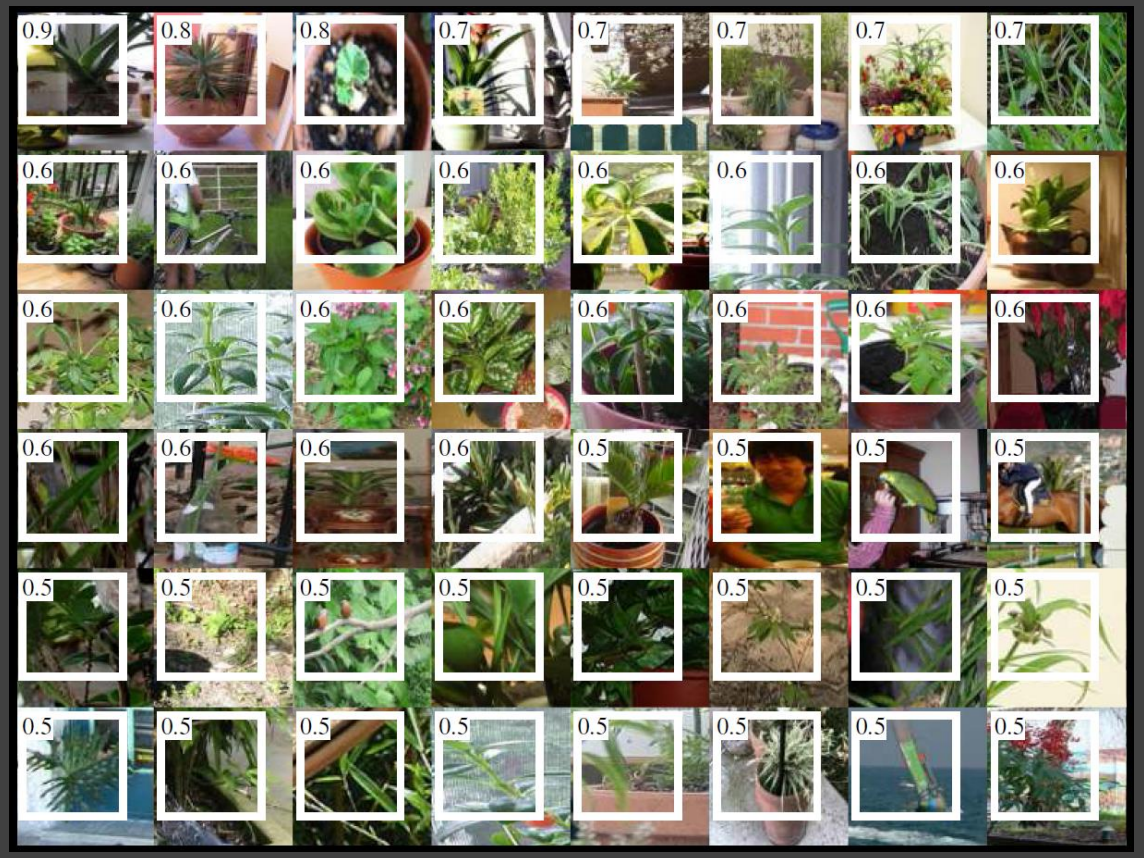
What did the CNN learn?



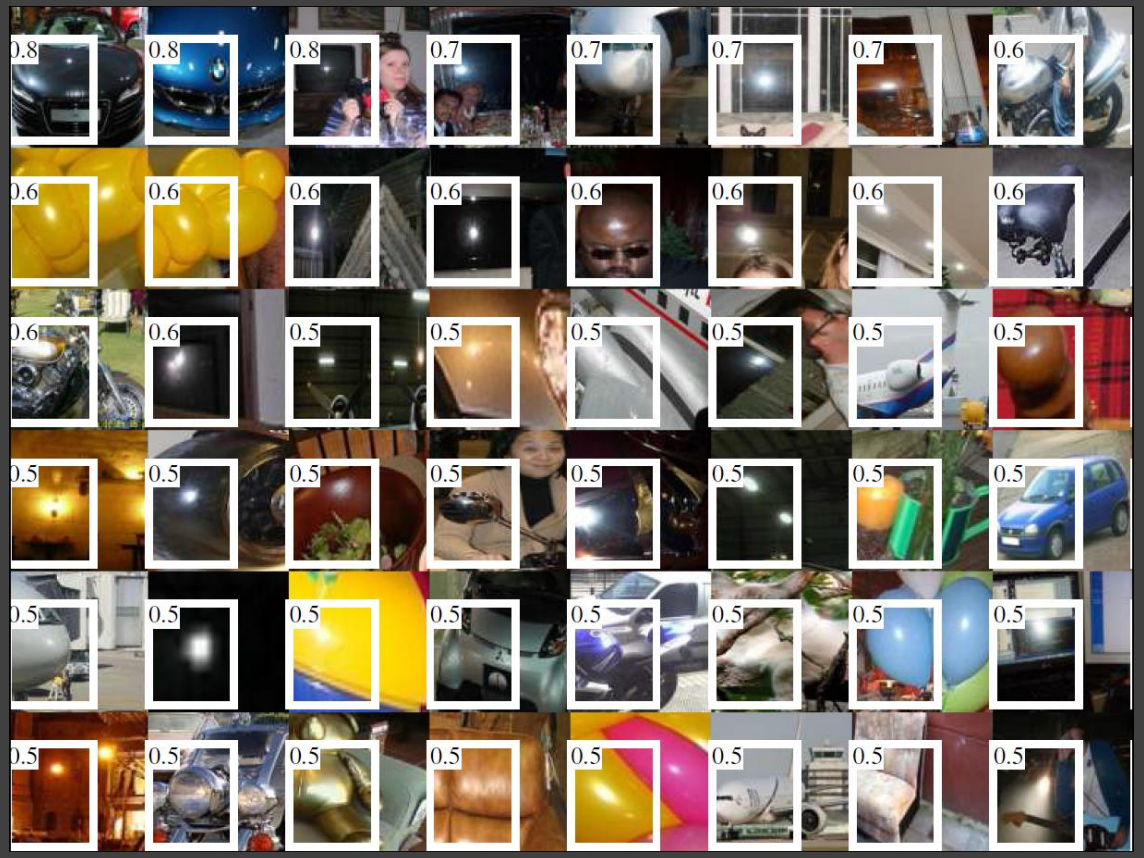
What did the CNN learn?



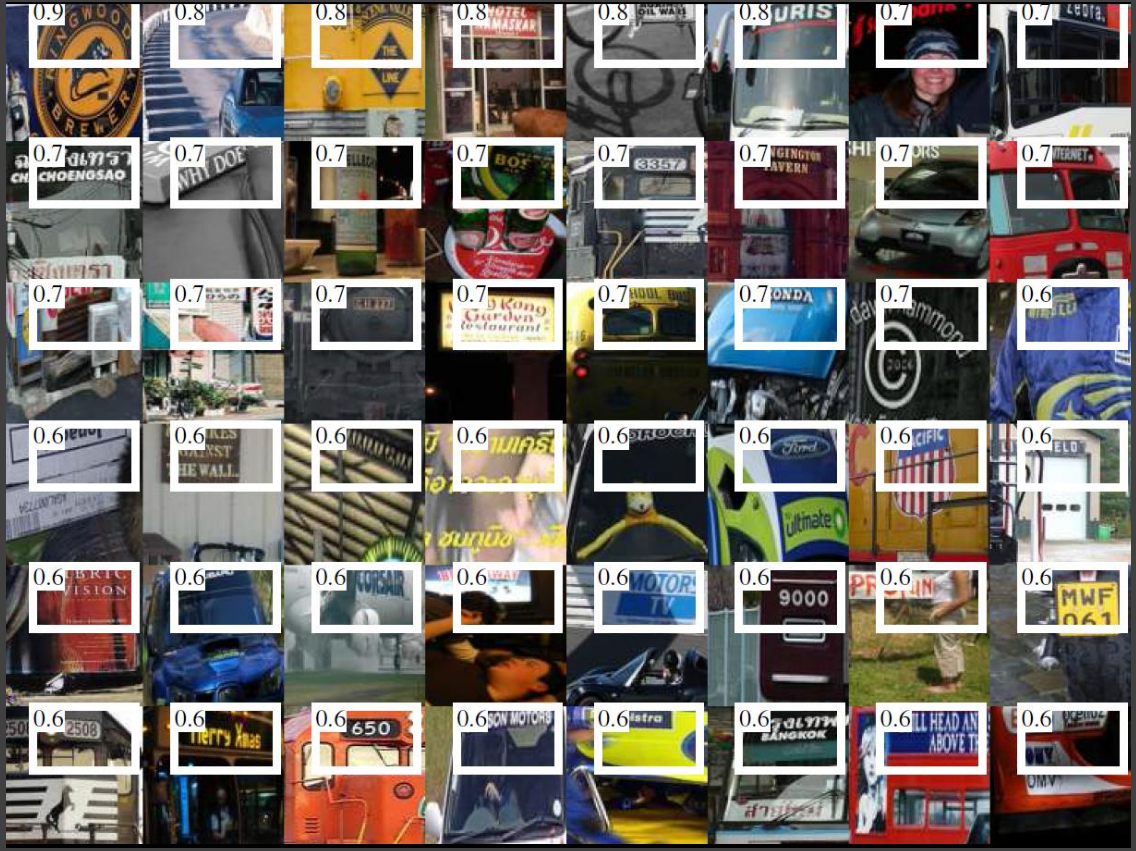
What did the CNN learn?



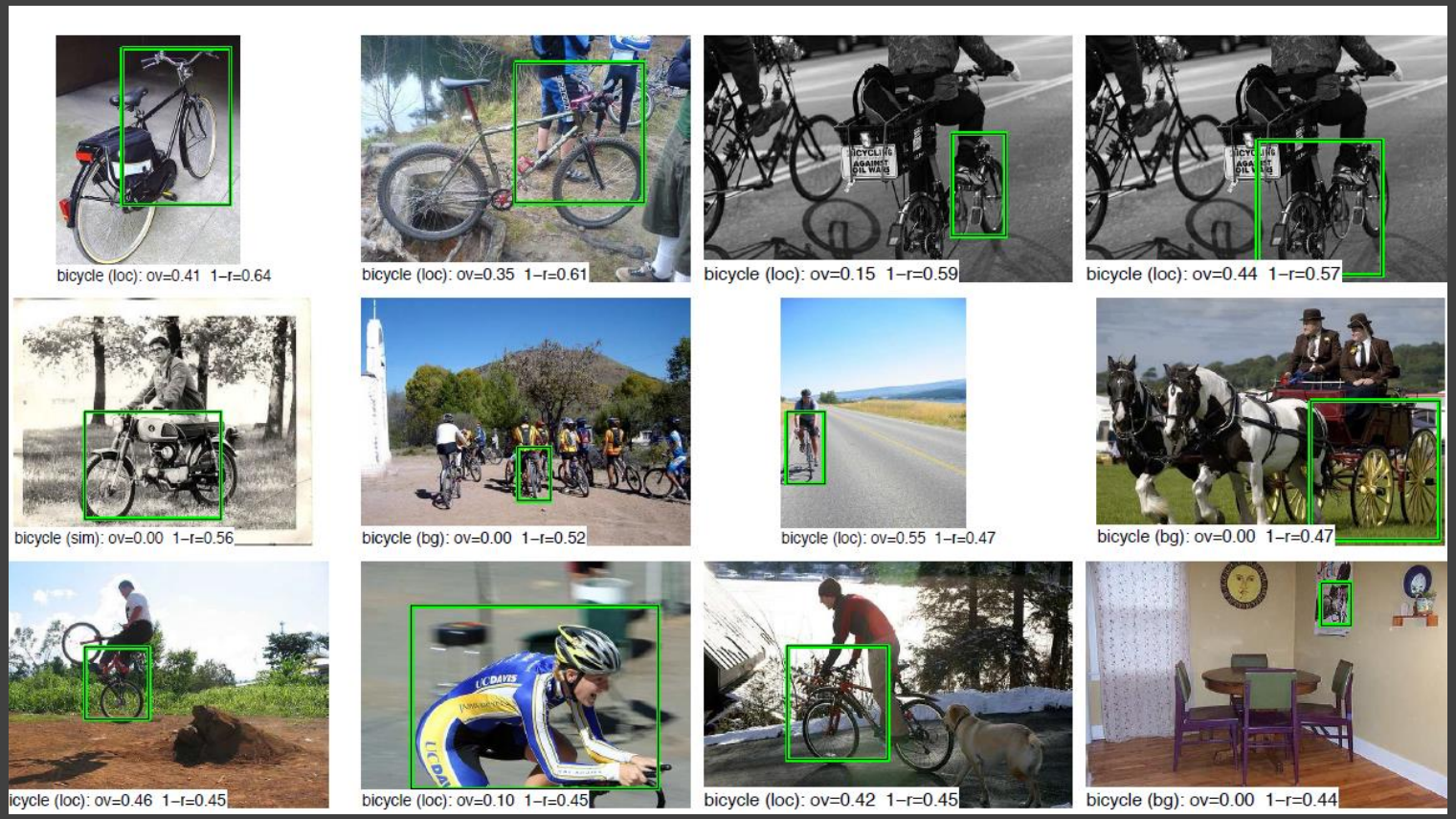
What did the CNN learn?



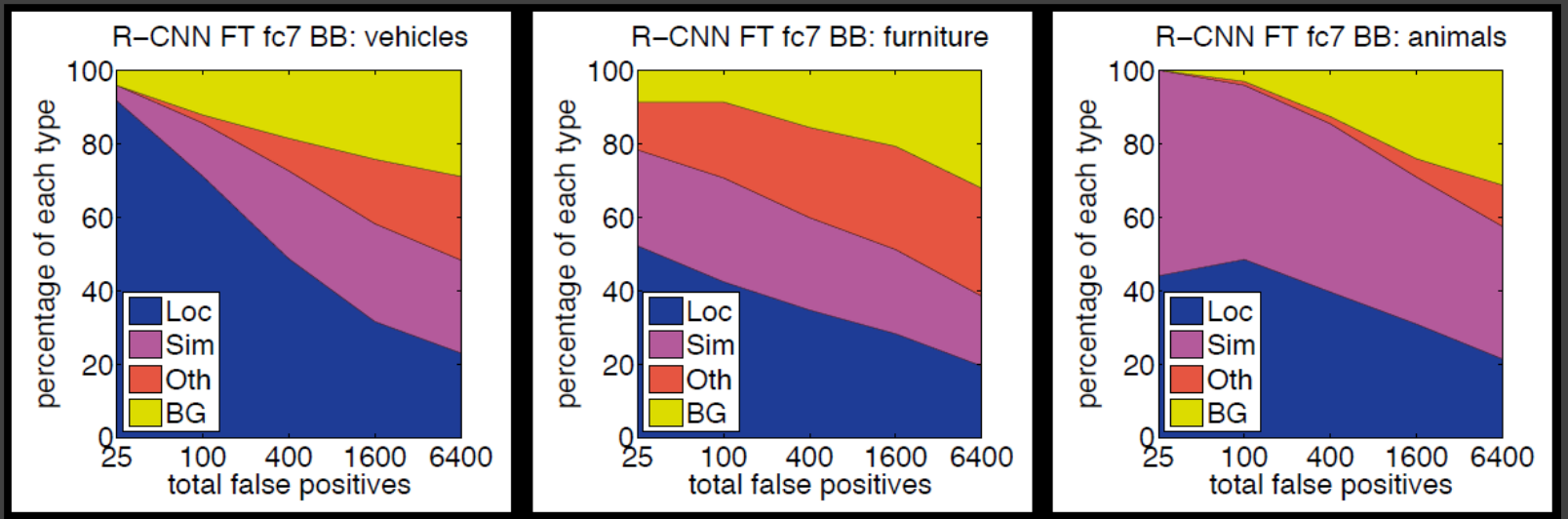
What did the CNN learn?



False-Positives

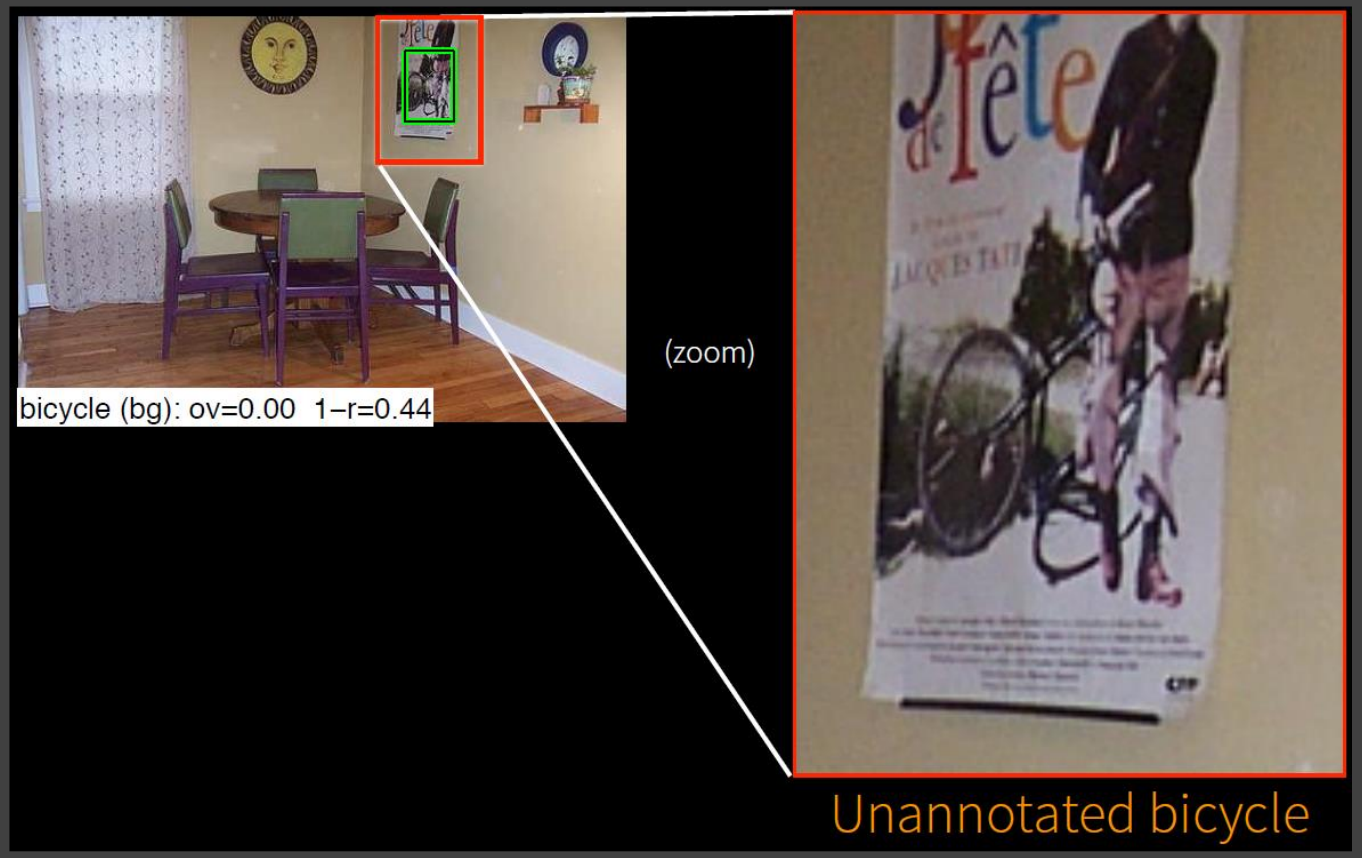


False-Positive Distribution



Analysis software: D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing Error in Object Detectors. ECCV, 2012.

False-Positive?



False-Positive?

1949 French comedy by Jacques Tati

Conclusion

R-CNN Conclusion

- Dramatically better PASCAL mAP
- Outperforms other CNN-based methods
- Detection speed manageable (~11s/image on GPU)
- Scales very well (30ms for 20 → 200 classes!)
- Relatively simple and open source

Questions