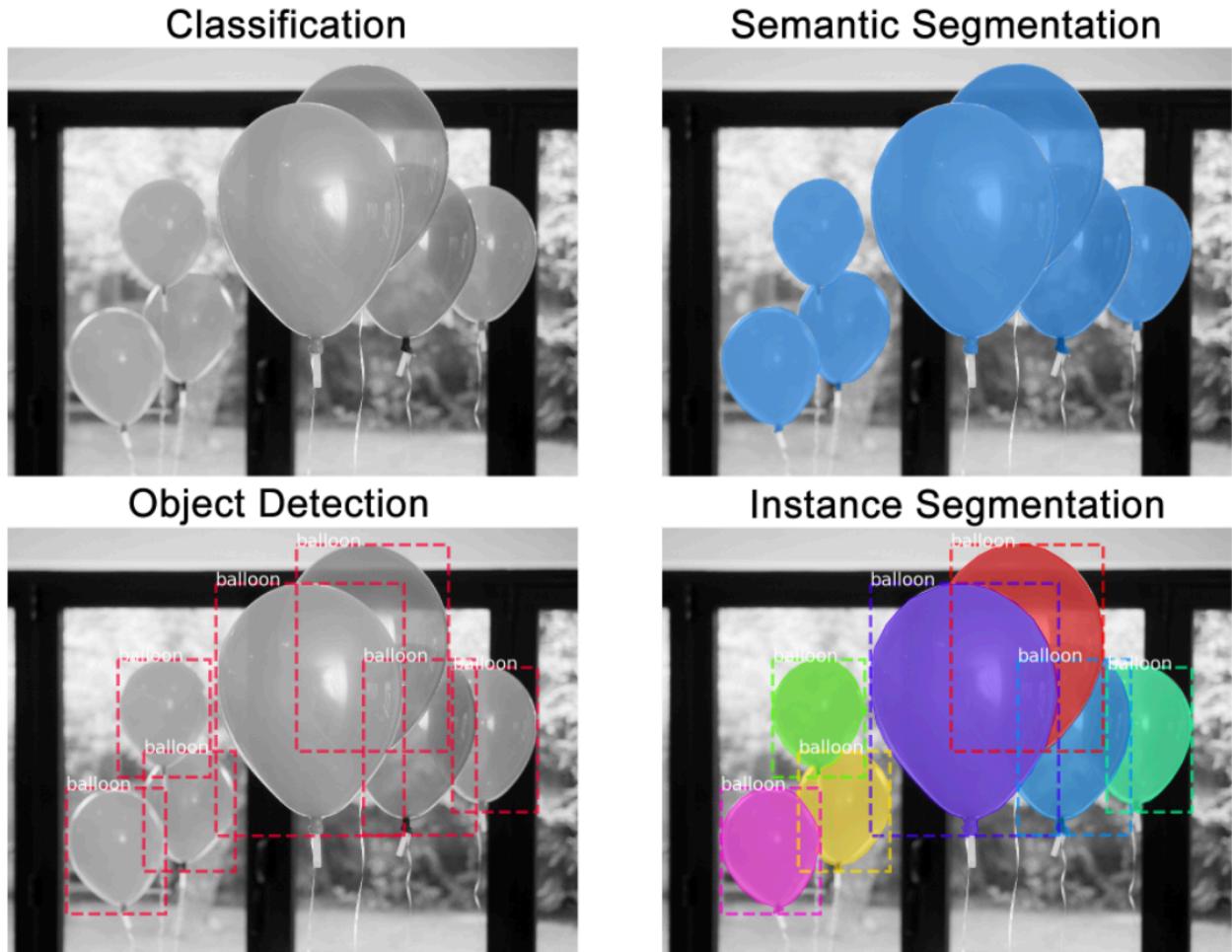


Deep Learning in Computer Vision Applications

Slides credit: Fatemeh Saleh, Australian National University

Applications of Deep Learning in Computer Vision

- **Classification:** There is a balloon in this image.
- **Semantic Segmentation:** These are all the balloon pixels.
- **Object Detection:** There are 7 balloons in this image at these locations. We're starting to account for objects that overlap.
- **Instance Segmentation:** There are 7 balloons at these locations, and these are the pixels that belong to each one.

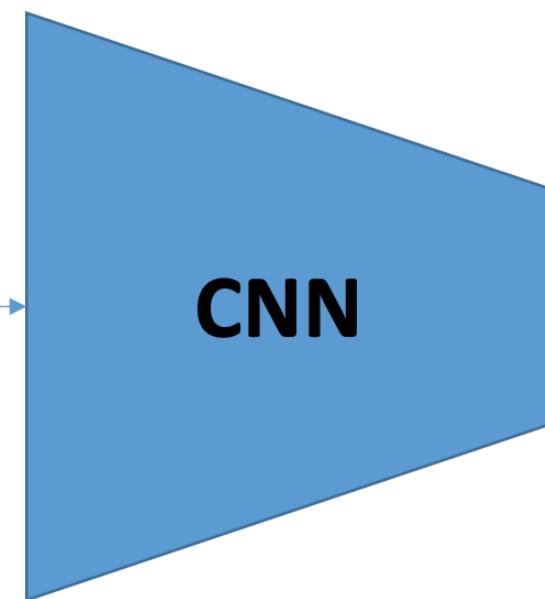


CNNs for image classification

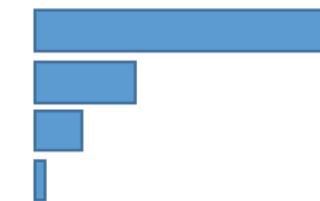
- Image classification problem
- ImageNet challenge
- CNN architectures
 - AlexNet
 - VGG
 - ResNet

Image Classification

- Given an image, what is the correct class label?



CAT



cat
dog
cheetah
monkey

ImageNet Challenge



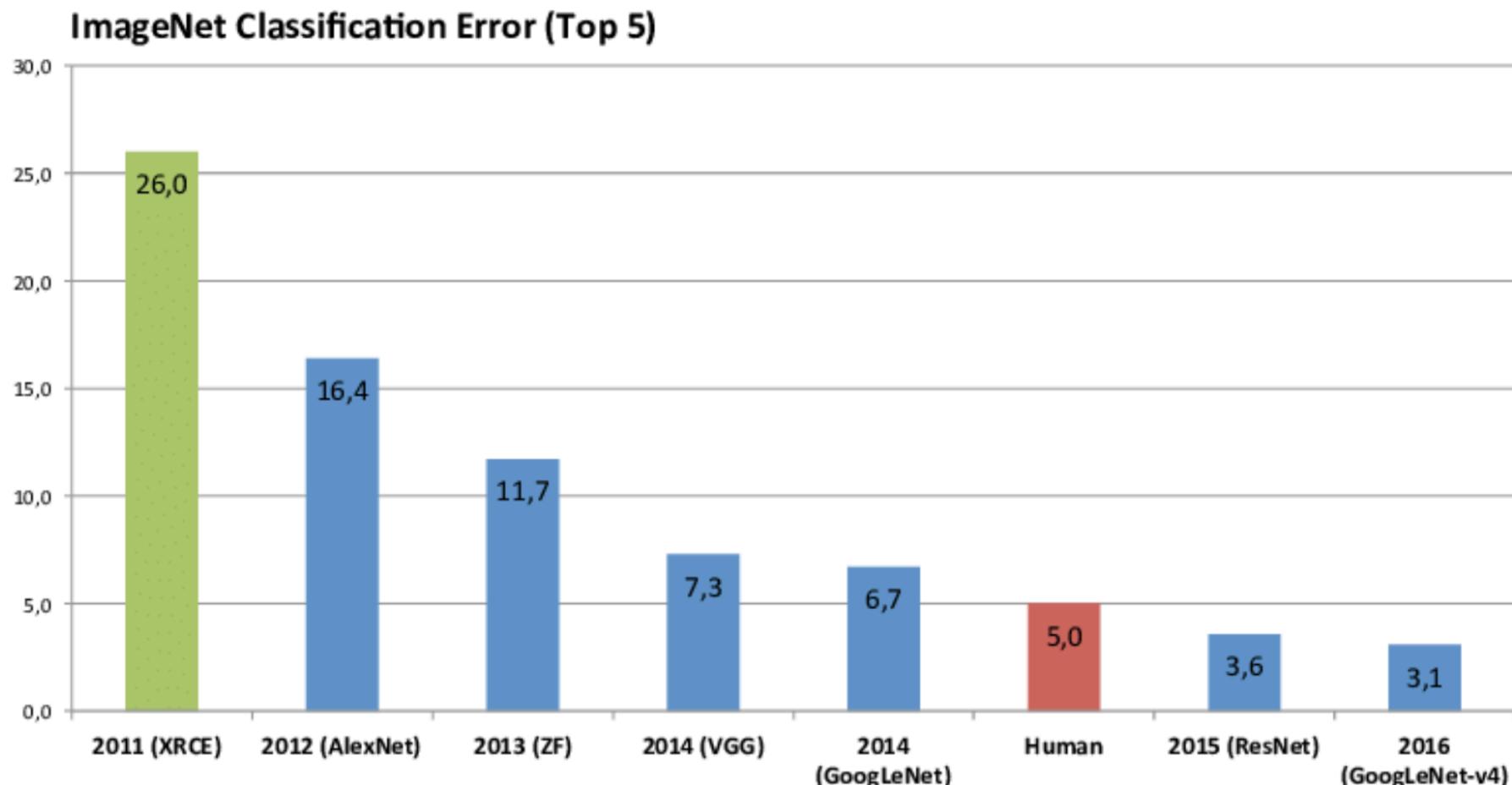
2019_week_11_DeepLearningApplications (page 15 of 82)

ImageNet Large Scale Visual Recognition Challenges

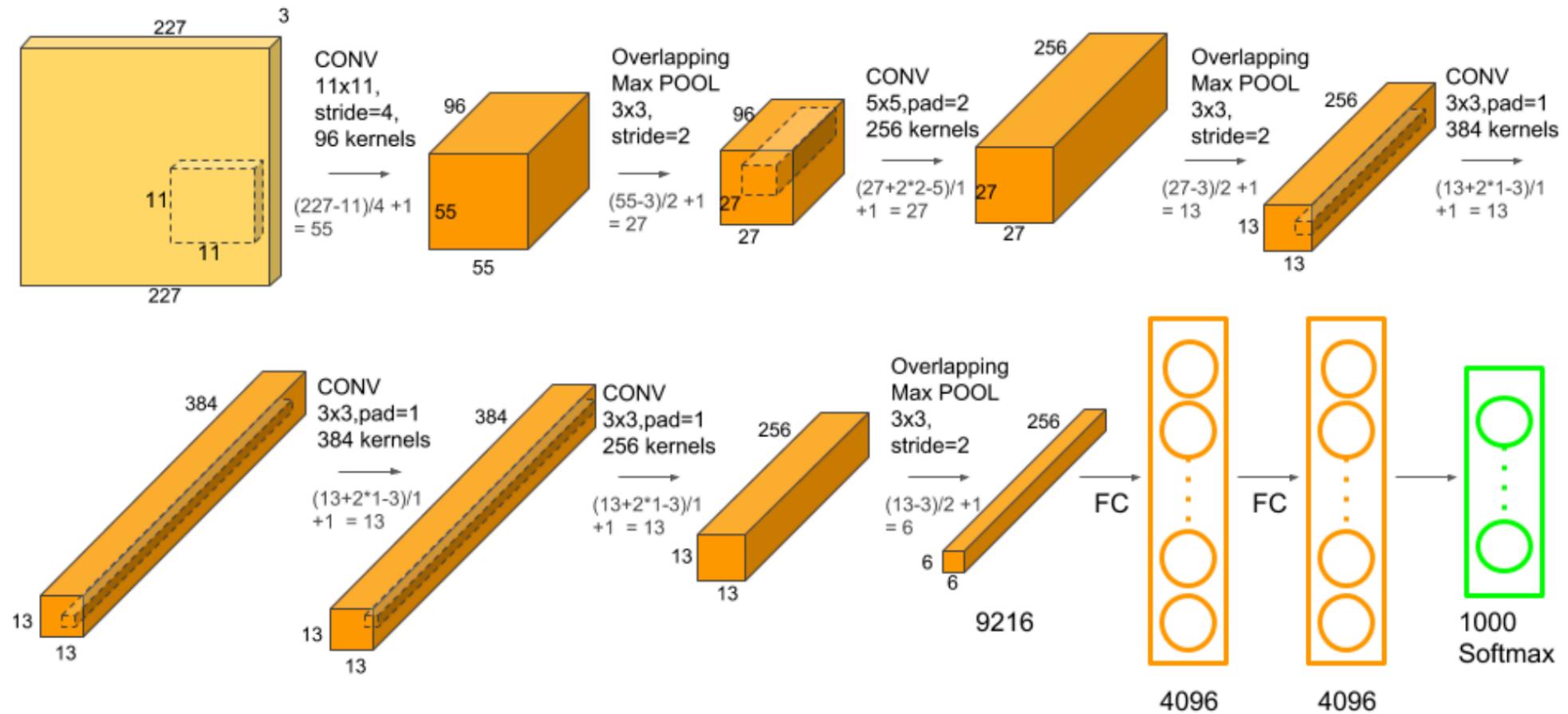


ImageNet Challenge

- Evaluation Metric: Top-k error

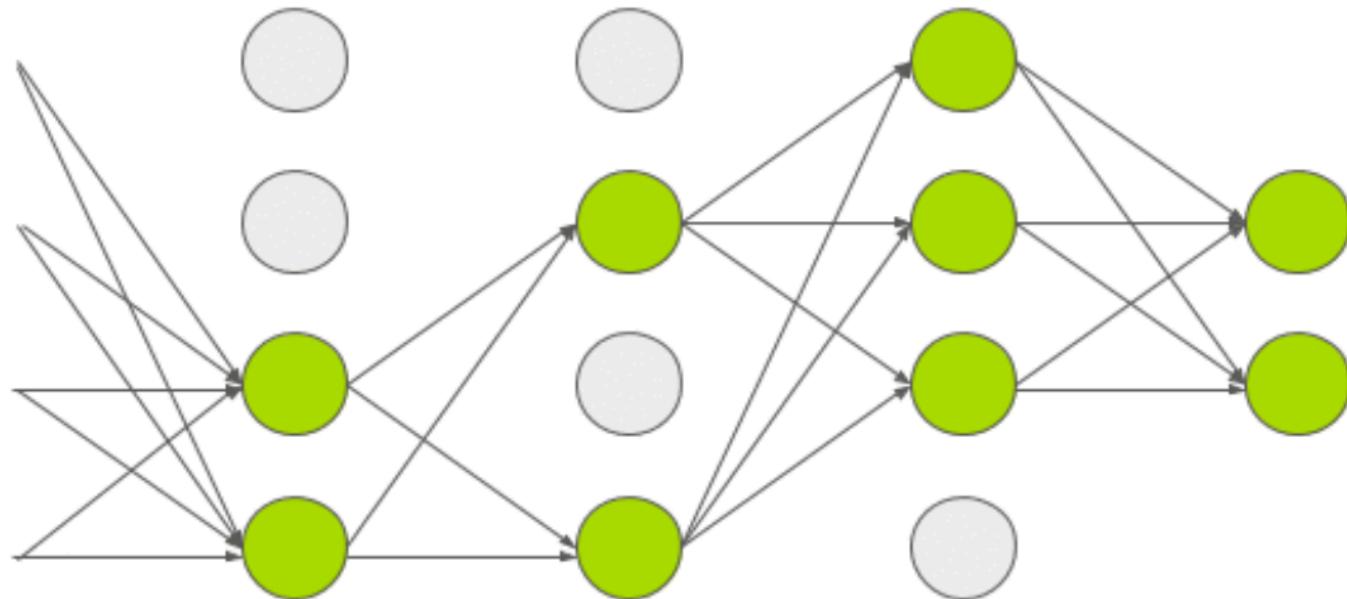


AlexNet

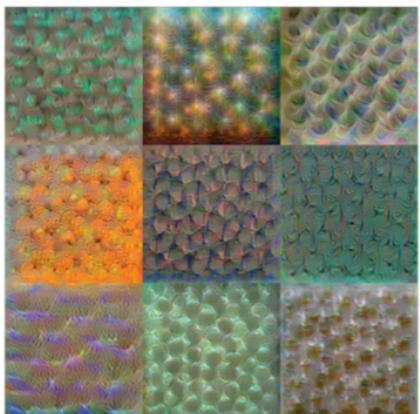
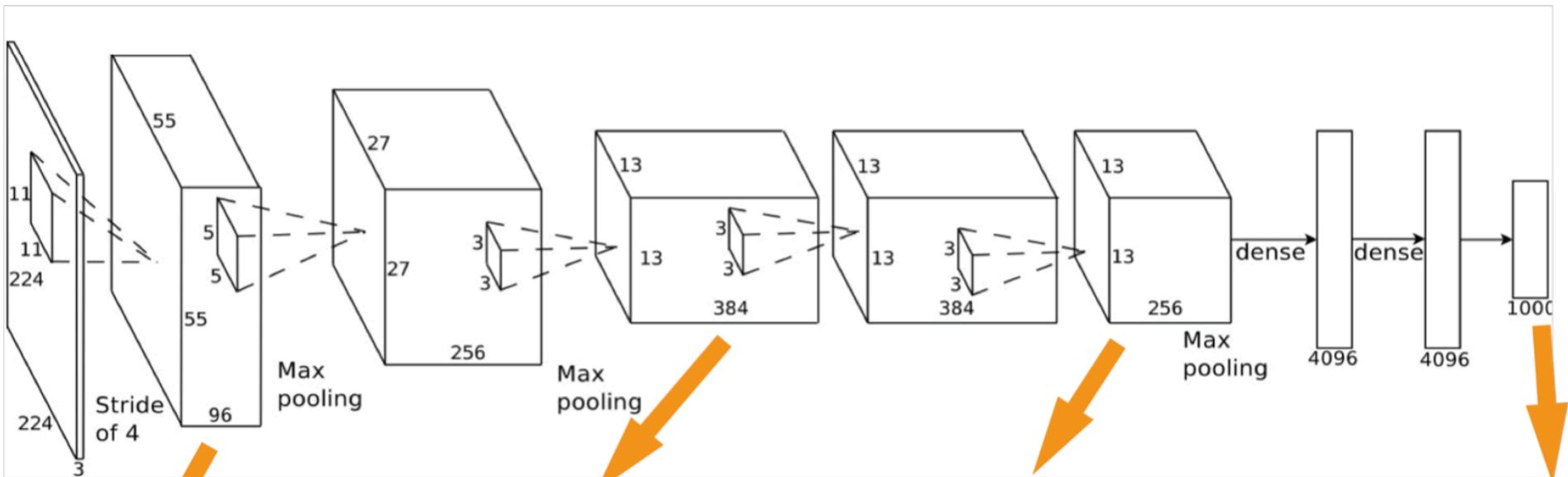


AlexNet

- Dropout Layer
 - Reducing over-fitting by generalization



AlexNet



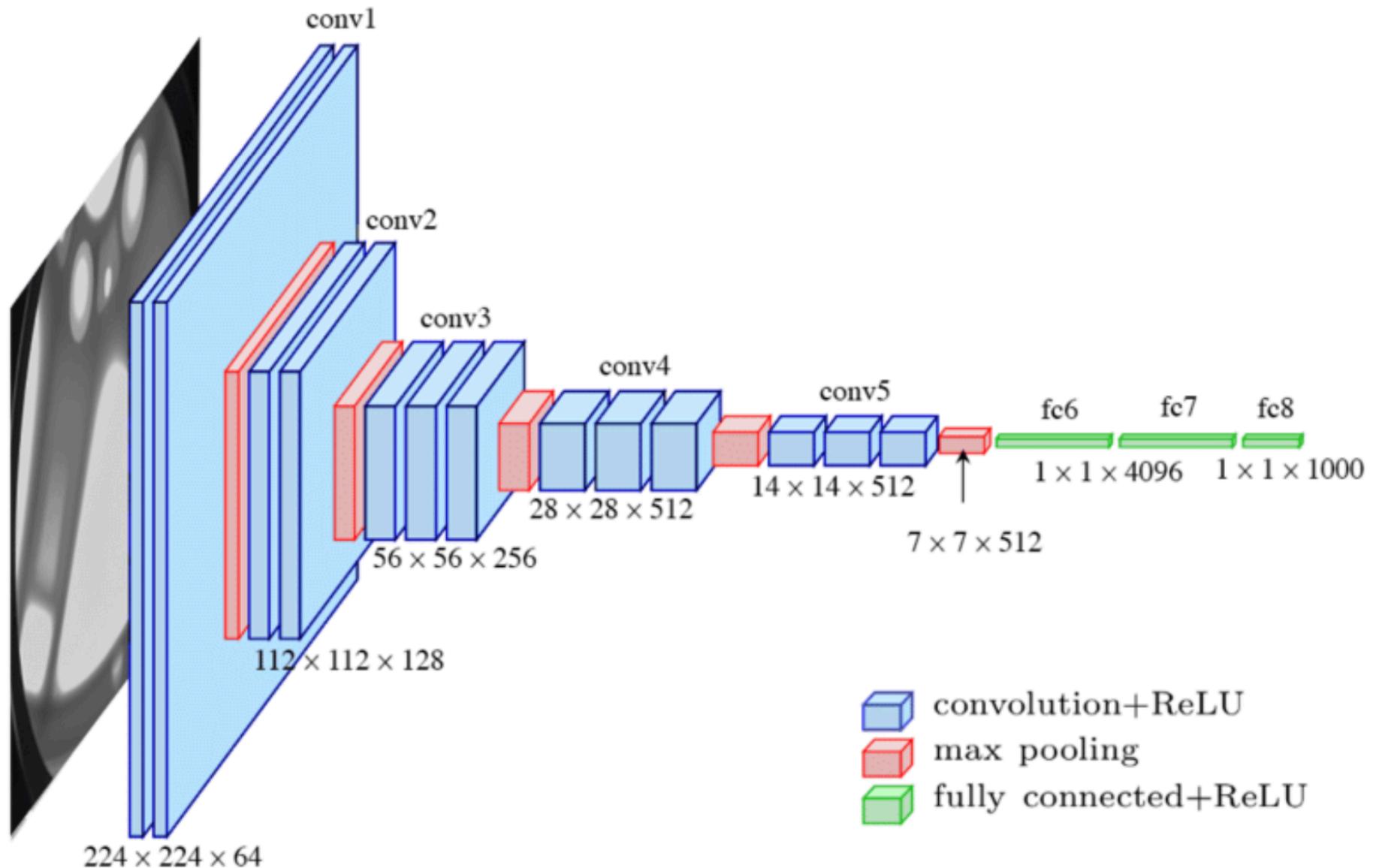
Numerical



Data-driven



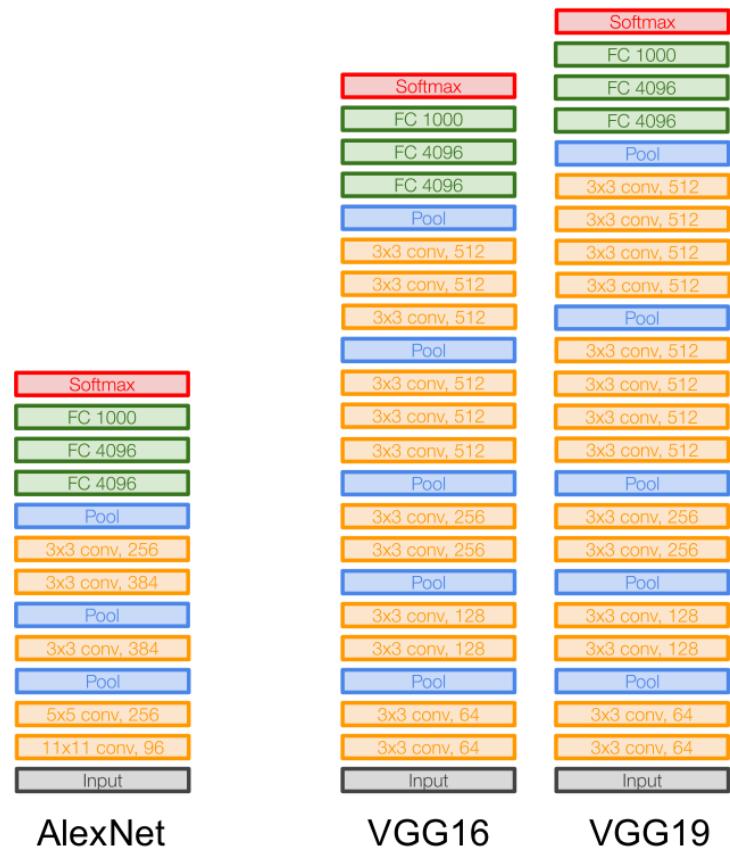
VGG



Case Study: VGGNet

[Simonyan and Zisserman, 2014]

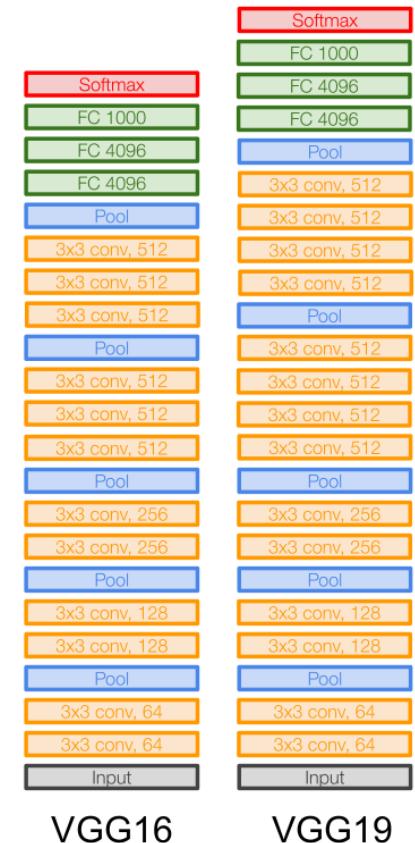
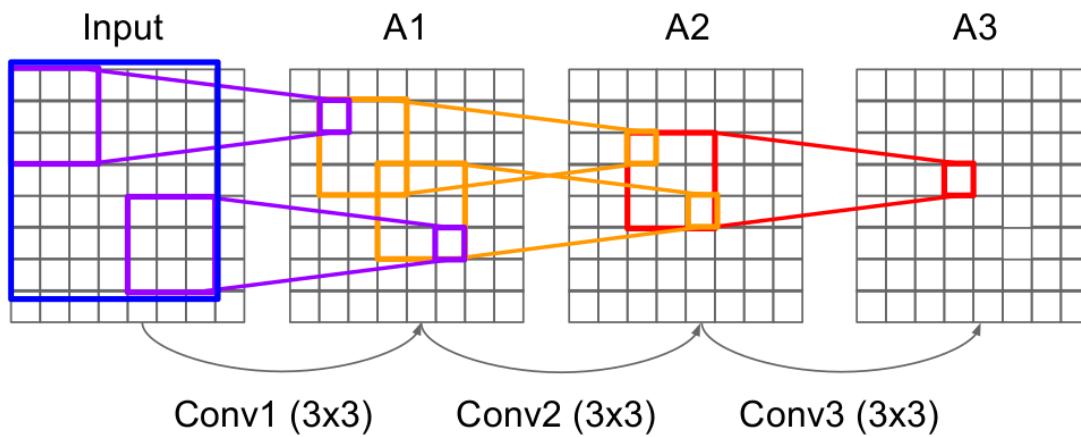
Q: Why use smaller filters? (3x3 conv)



Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field
of three 3x3 conv (stride 1) layers?



Case Study: VGGNet

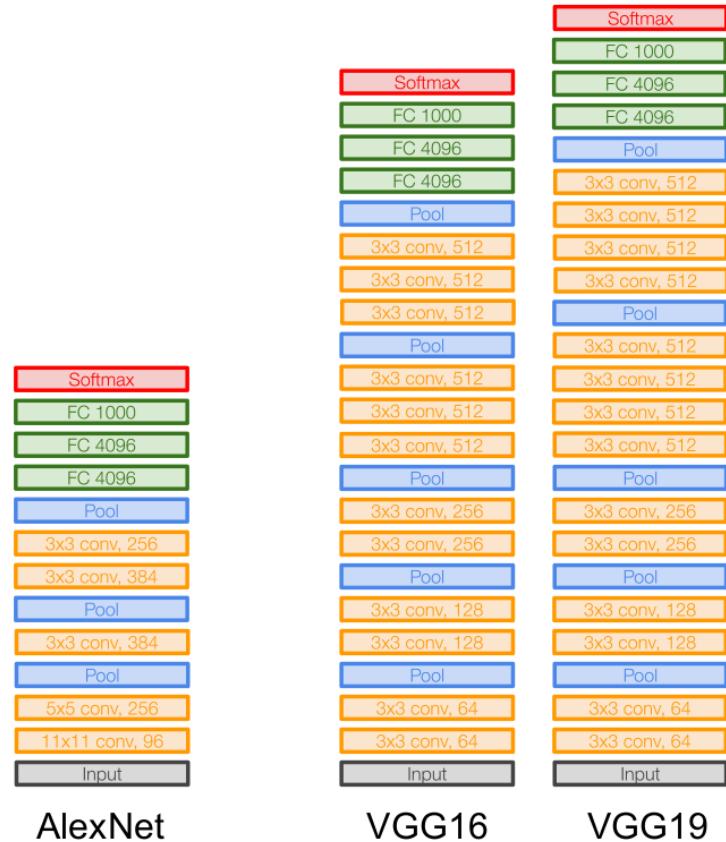
[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

Stack of three 3x3 conv (stride 1) layers
has same **effective receptive field** as
one 7x7 conv layer

But deeper, more non-linearities

And fewer parameters: $3 * (3^2 C^2)$ vs.
 $7^2 C^2$ for C channels per layer

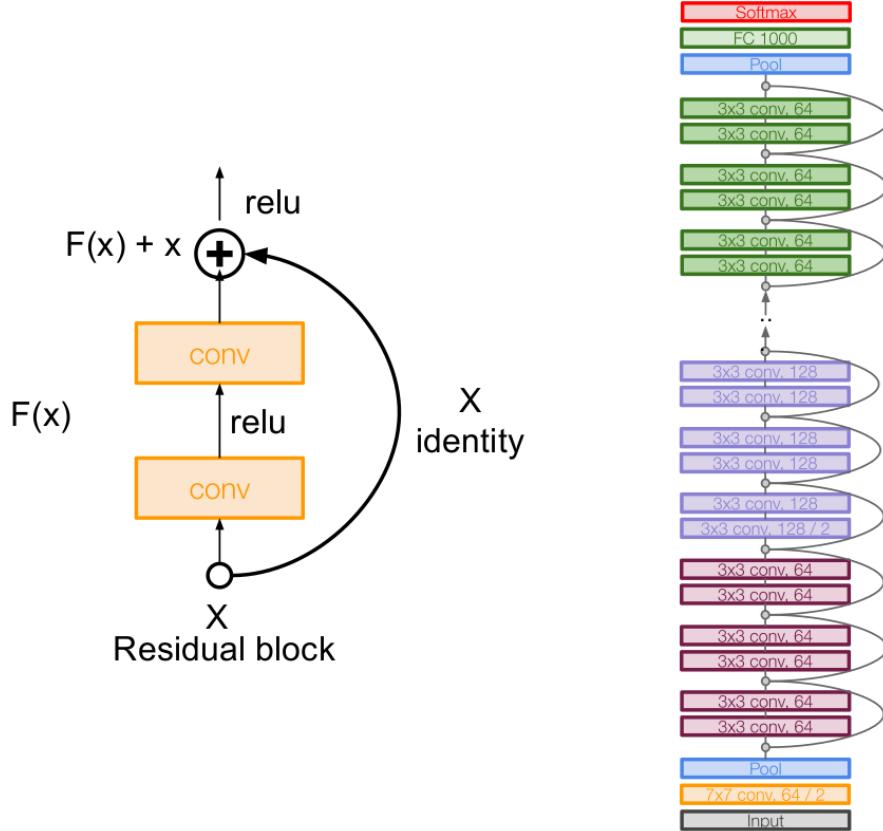


Case Study: ResNet

[He et al., 2015]

Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!



Case Study: ResNet

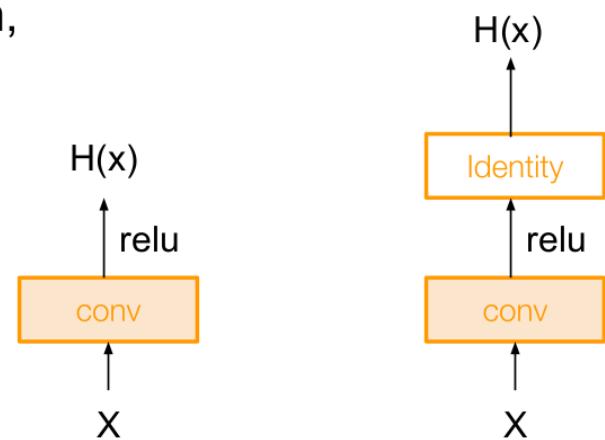
[He et al., 2015]

Fact: Deep models have more representation power (more parameters) than shallower models.

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize

What should the deeper model learn to be at least as good as the shallower model?

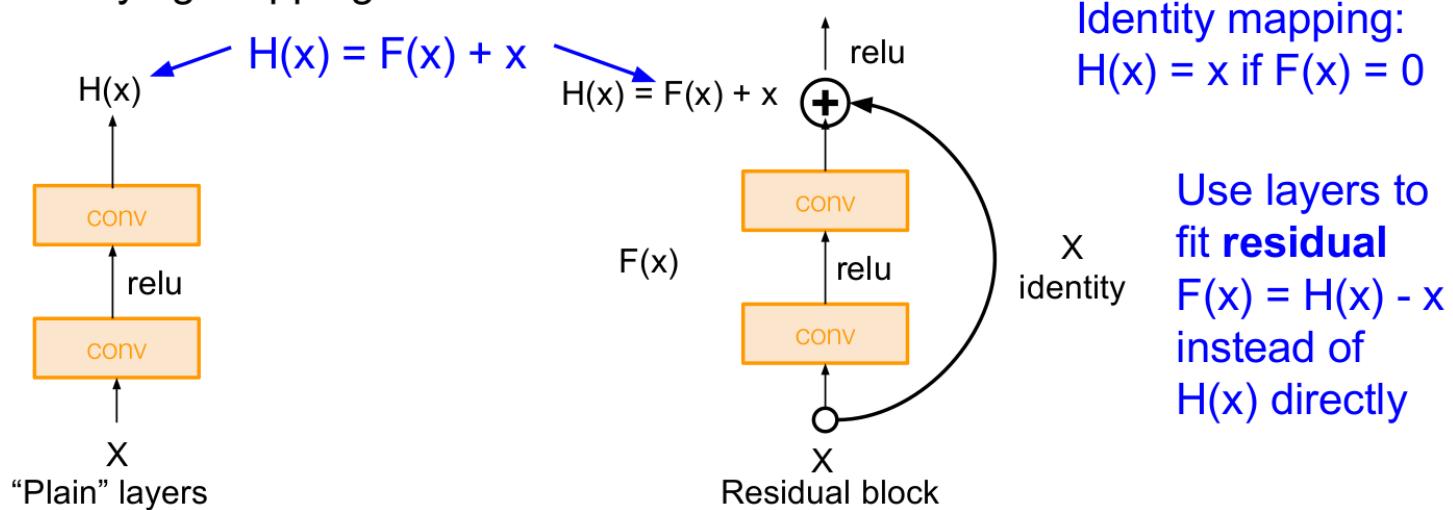
A solution by construction is copying the learned layers from the shallower model and setting additional layers to identity mapping.



Case Study: ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



Main takeaways

AlexNet showed that you can use CNNs to train Computer Vision models.

ZFNet, VGG shows that bigger networks work better

GoogLeNet is one of the first to focus on efficiency using 1x1 bottleneck convolutions and global avg pool instead of FC layers

ResNet showed us how to train extremely deep networks

- Limited only by GPU & memory!
- Showed diminishing returns as networks got bigger

After ResNet: CNNs were better than the human metric and focus shifted to Efficient networks:

- Lots of tiny networks aimed at mobile devices: **MobileNet, ShuffleNet**

Neural Architecture Search can now automate architecture design

CNNs for Object Detection

- Object Detection Problem
- Evaluation Metrics
- CNN Architectures for Object Detection (two-stage methods)
 - RCNN
 - Fast RCNN
 - Faster RCNN
- Instance level semantic segmentation

Object Detection

- Classification vs. Detection



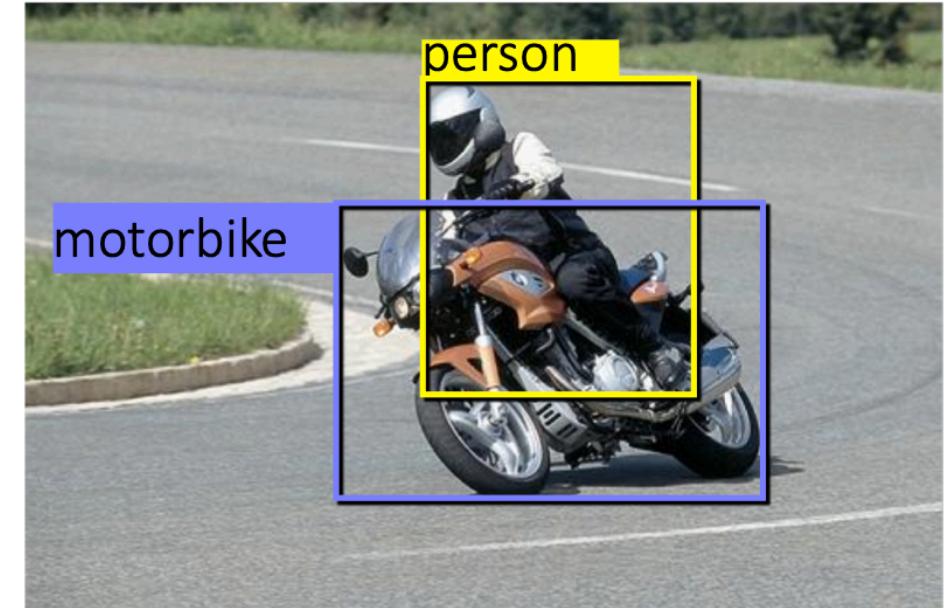
Object Detection

- Problem formulation

{ airplane, bird, motorbike, person, sofa }



Input



Desired output

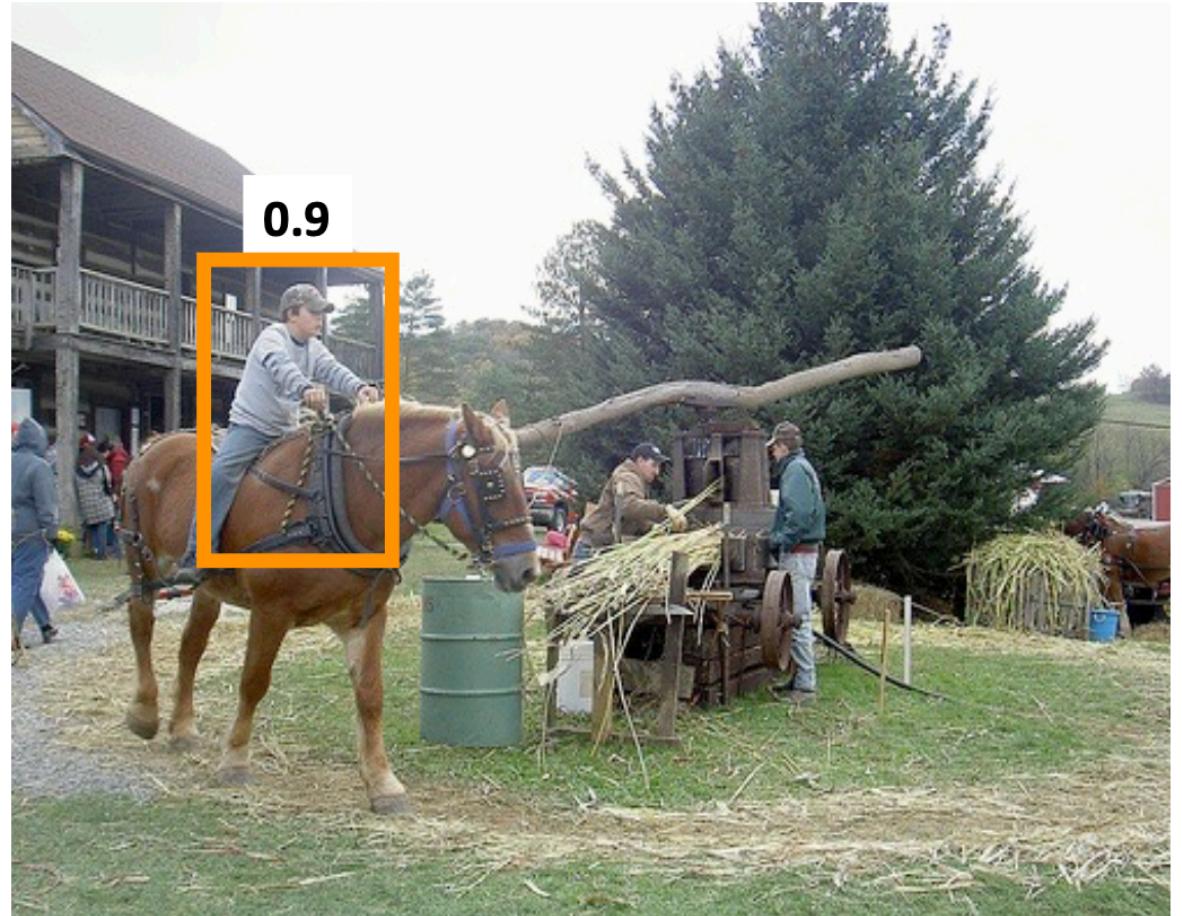
Evaluation Metrics

- Test image (previously unseen)



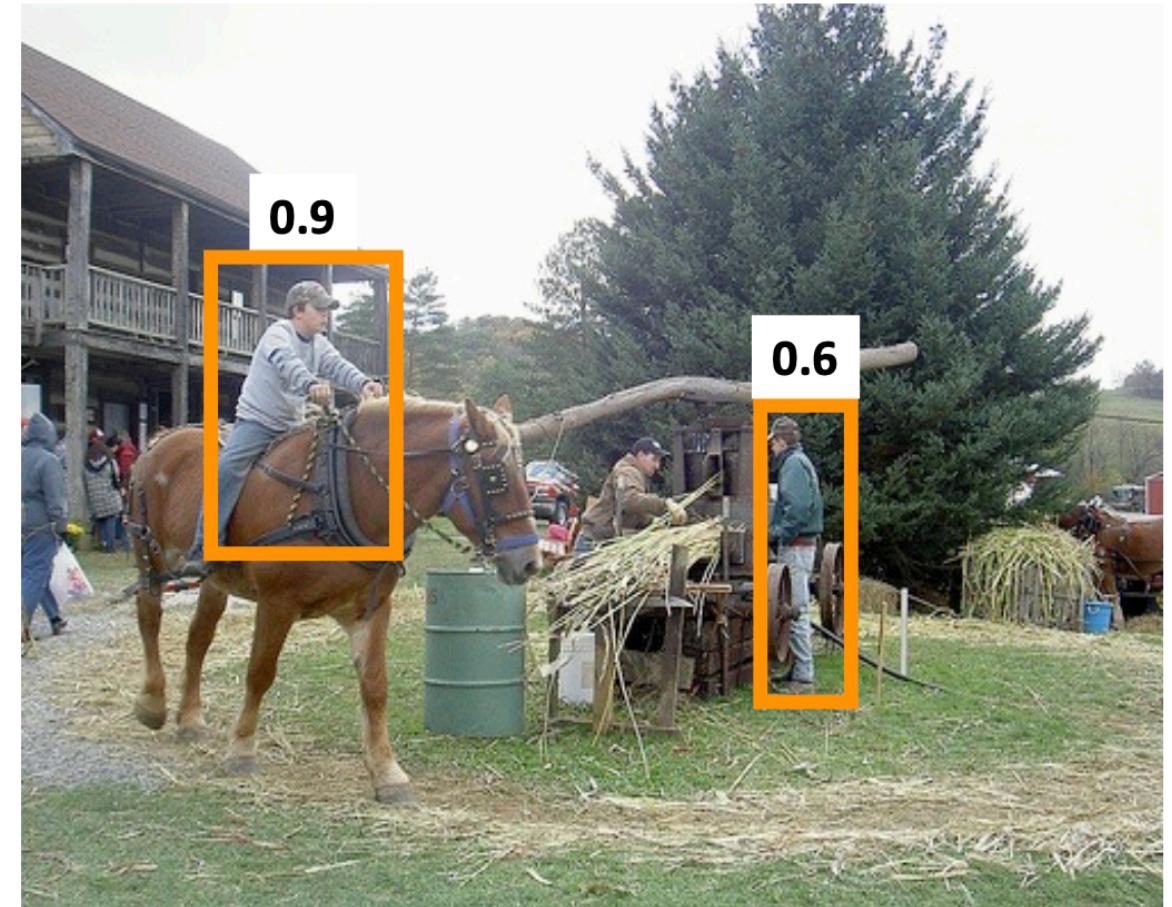
Evaluation Metrics

- First detection ...
- ‘person’ detector predictions



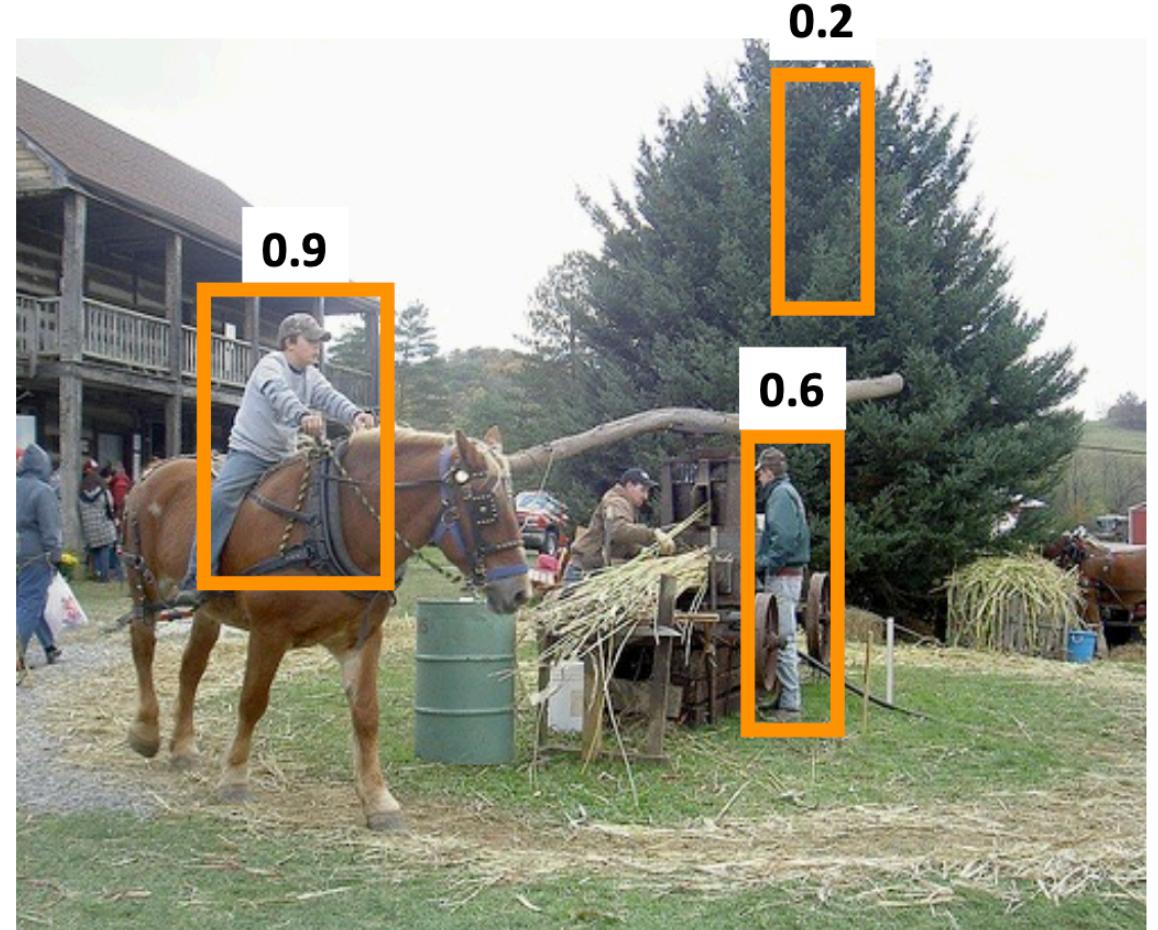
Evaluation Metrics

- Second detection ...
- ‘person’ detector predictions



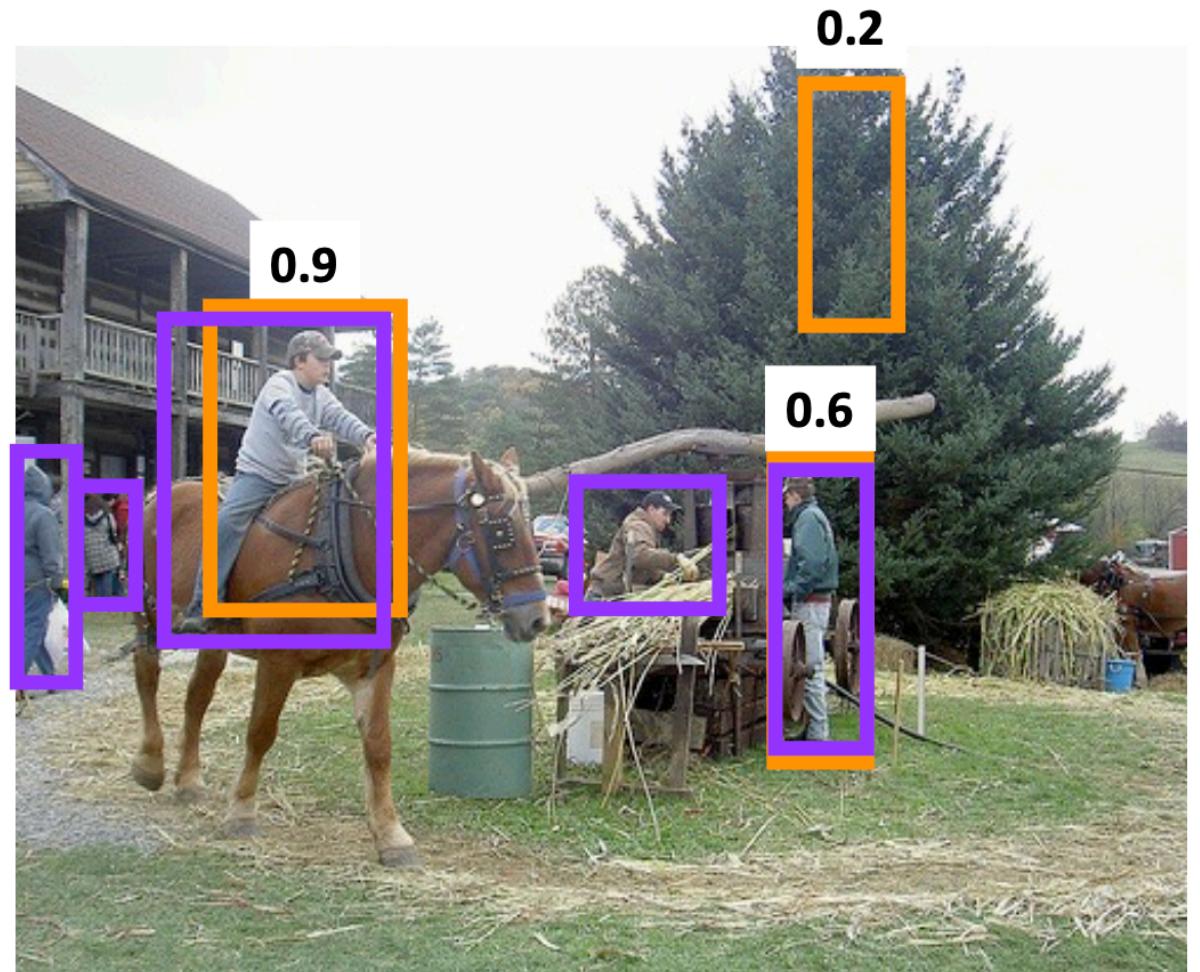
Evaluation Metrics

- Third detection ...
- ‘person’ detector predictions



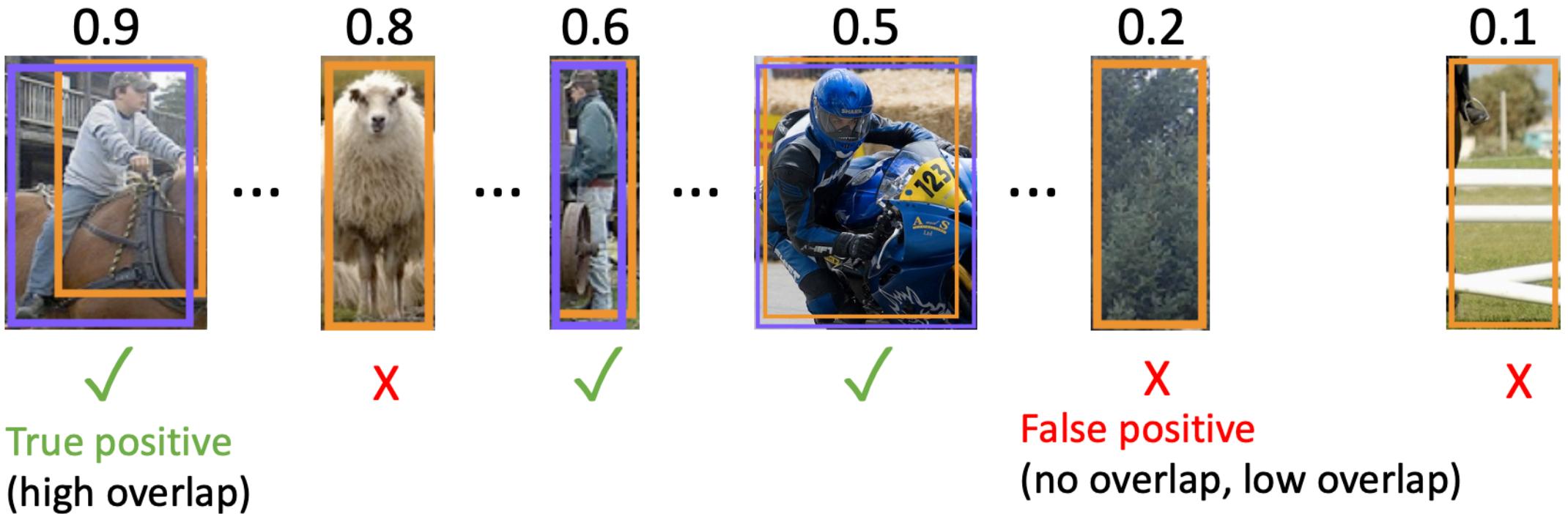
Evaluation Metrics

- Compare to ground-truth
 - ‘person’ detector predictions
 - Ground-truth ‘person’ boxes



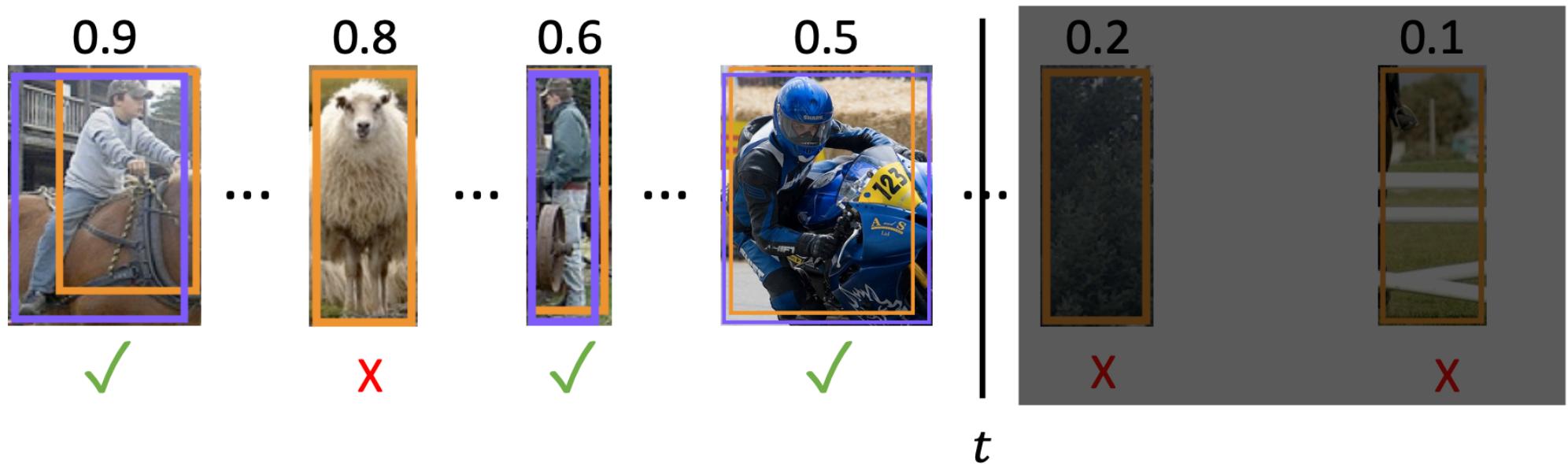
Evaluation Metrics

- Sort by confidence



$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

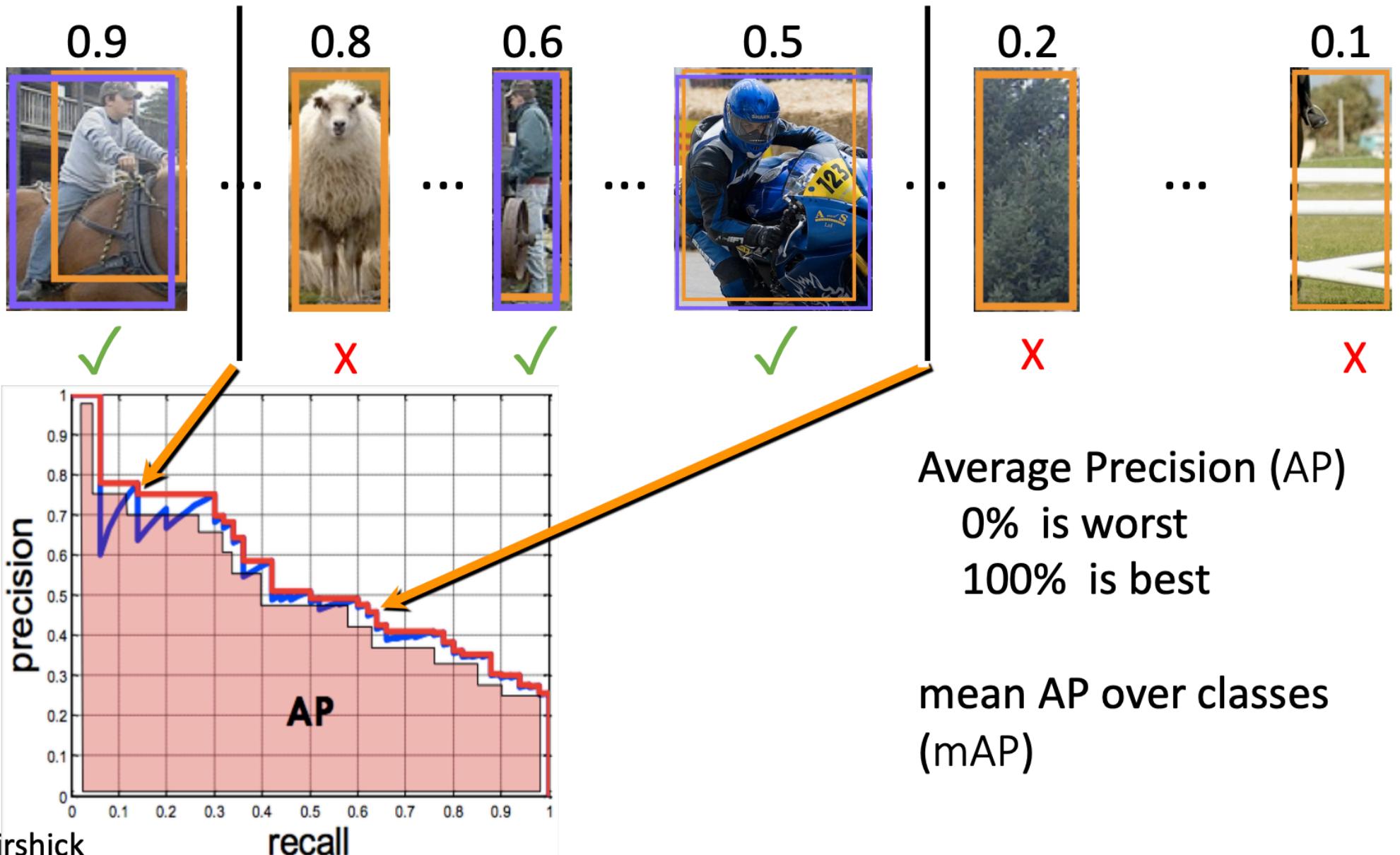
Evaluation Metrics



$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t}$$

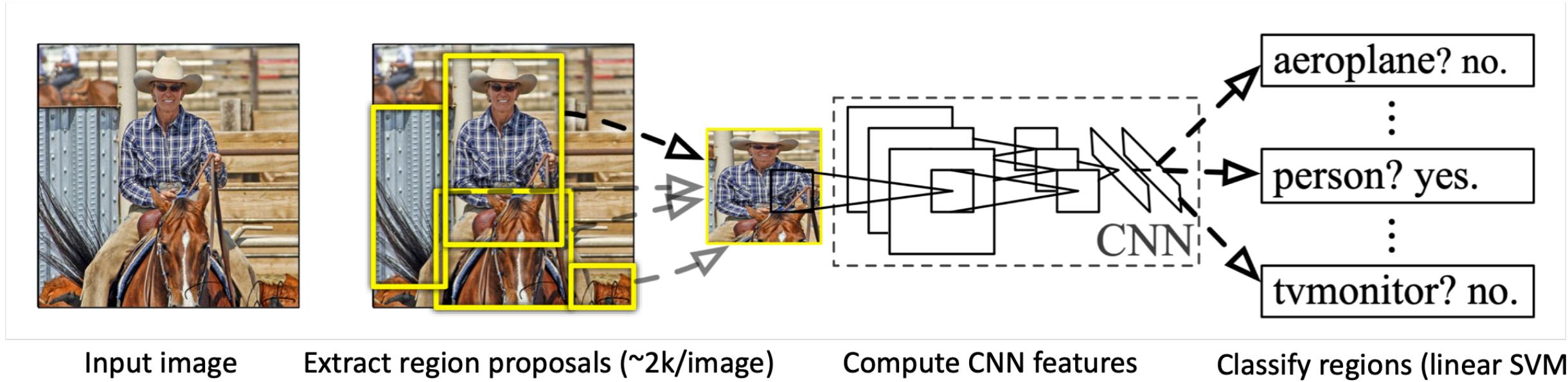
$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects}$$

Evaluation Metrics



RCNN

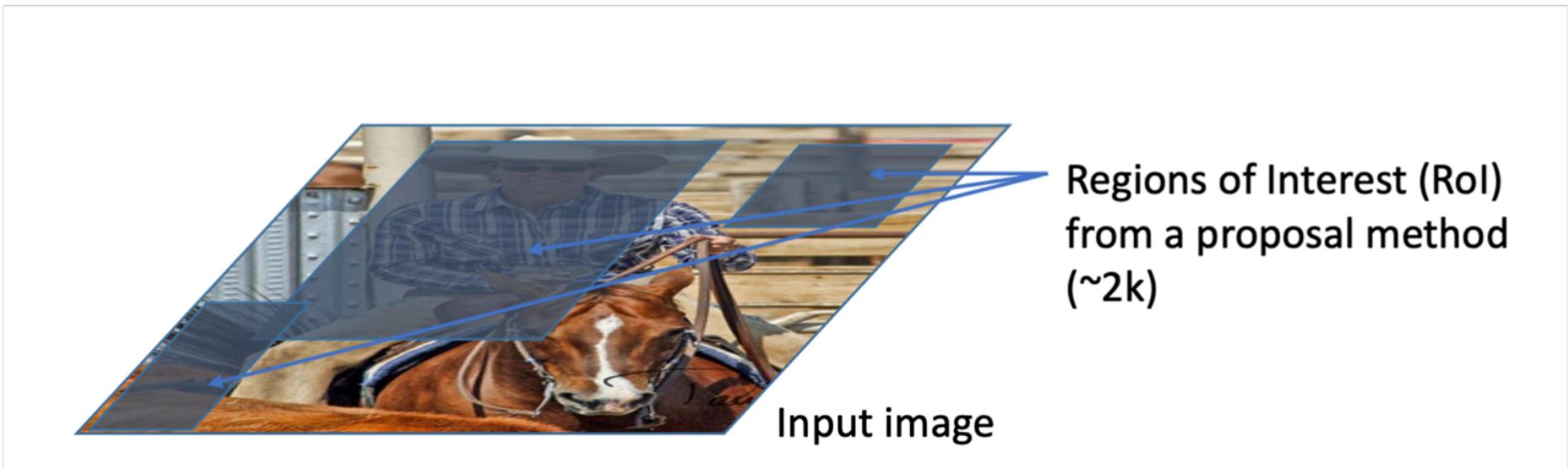
- Regions with CNN features



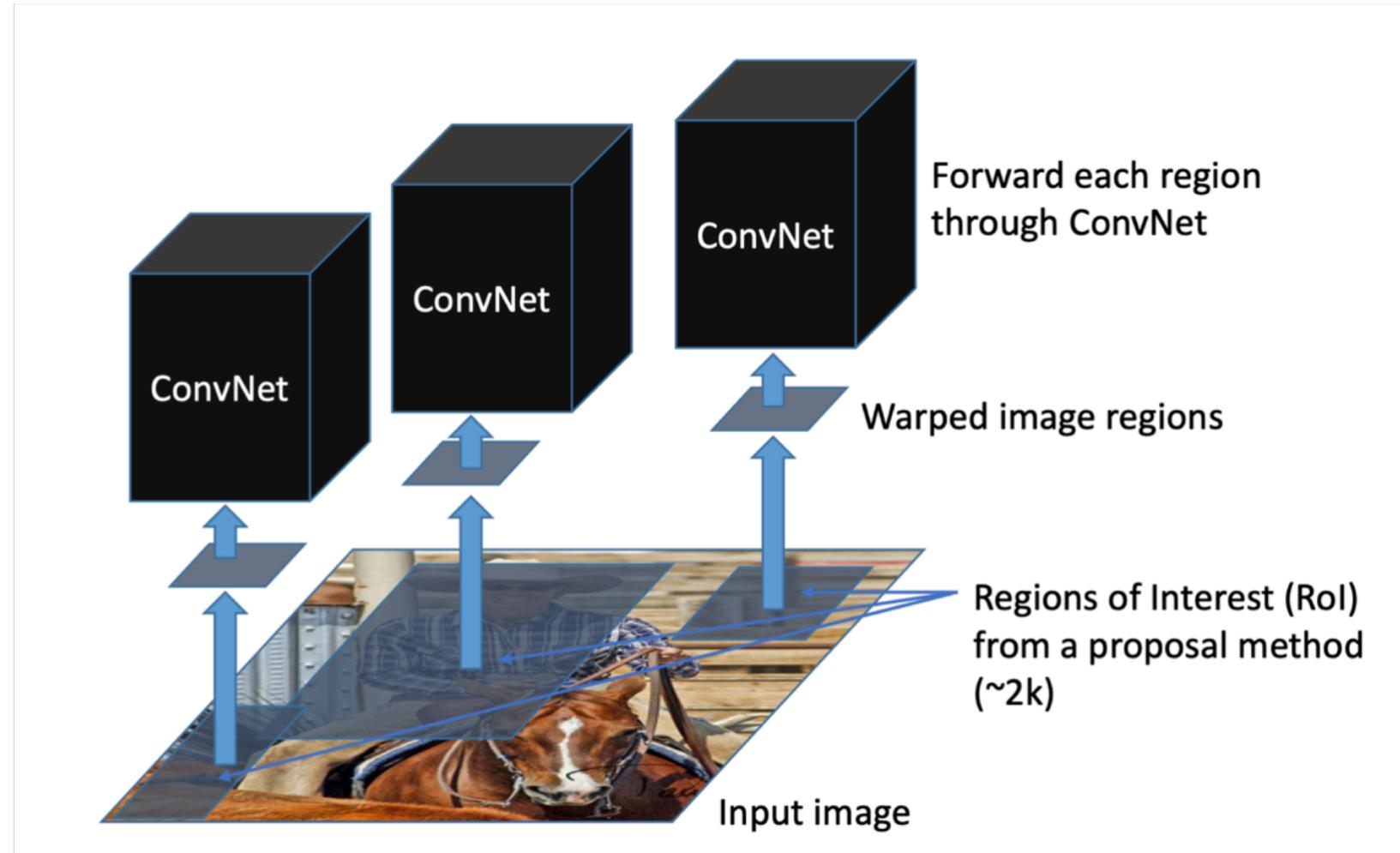
RCNN



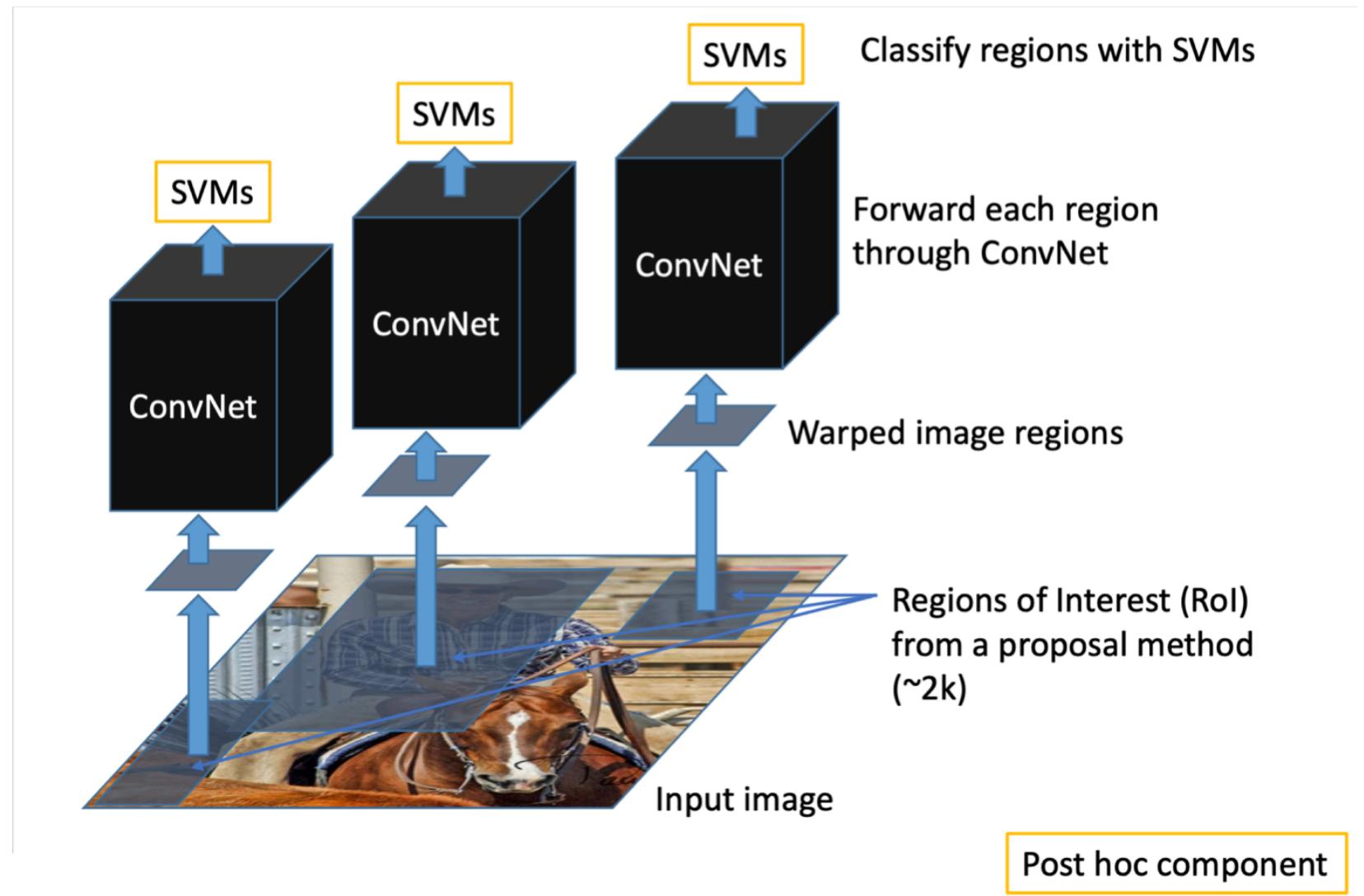
RCNN



RCNN

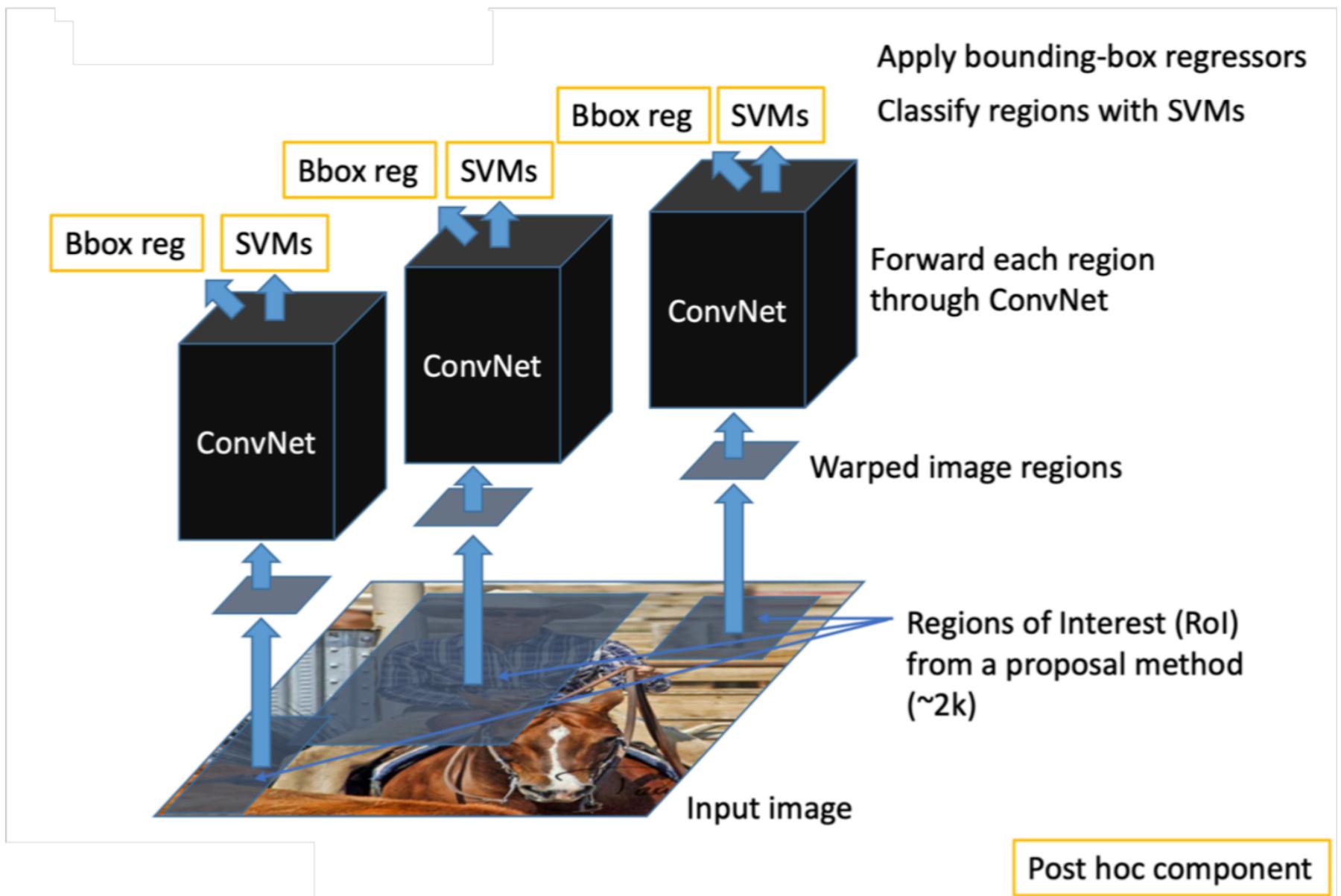


RCNN



Slide credit: Ross Girshick

RCNN



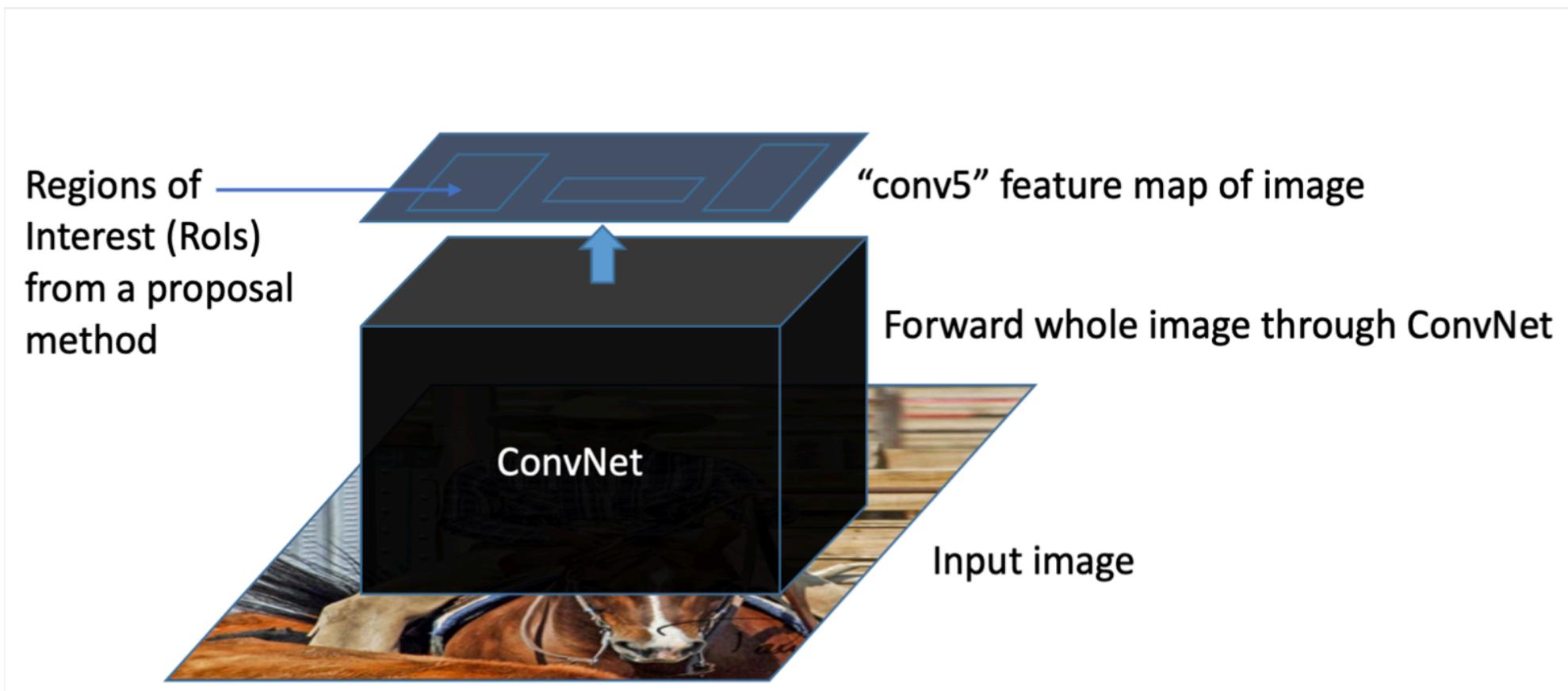
RCNN

- Problems
 - Ad hoc training objectives
 - Fine-tune network with softmax classifier
 - Train linear SVMs
 - Train bounding-box regressors
 - Training is slow (84h), takes a lot of disk space
 - Inference (detection) is slow, 47s/image with VGG16

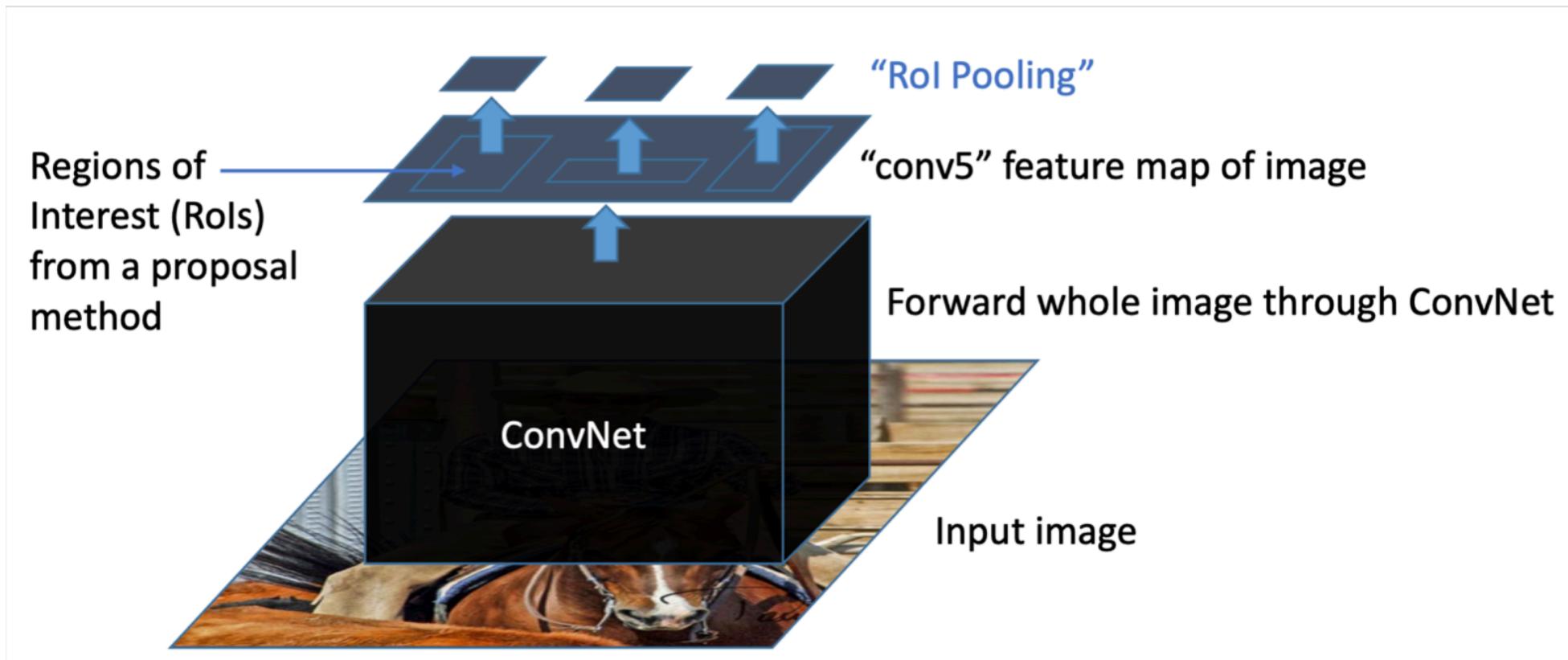
Fast RCNN

- Fast test-time
- One network, trained in one stage
- Higher mean average precision than R-CNN

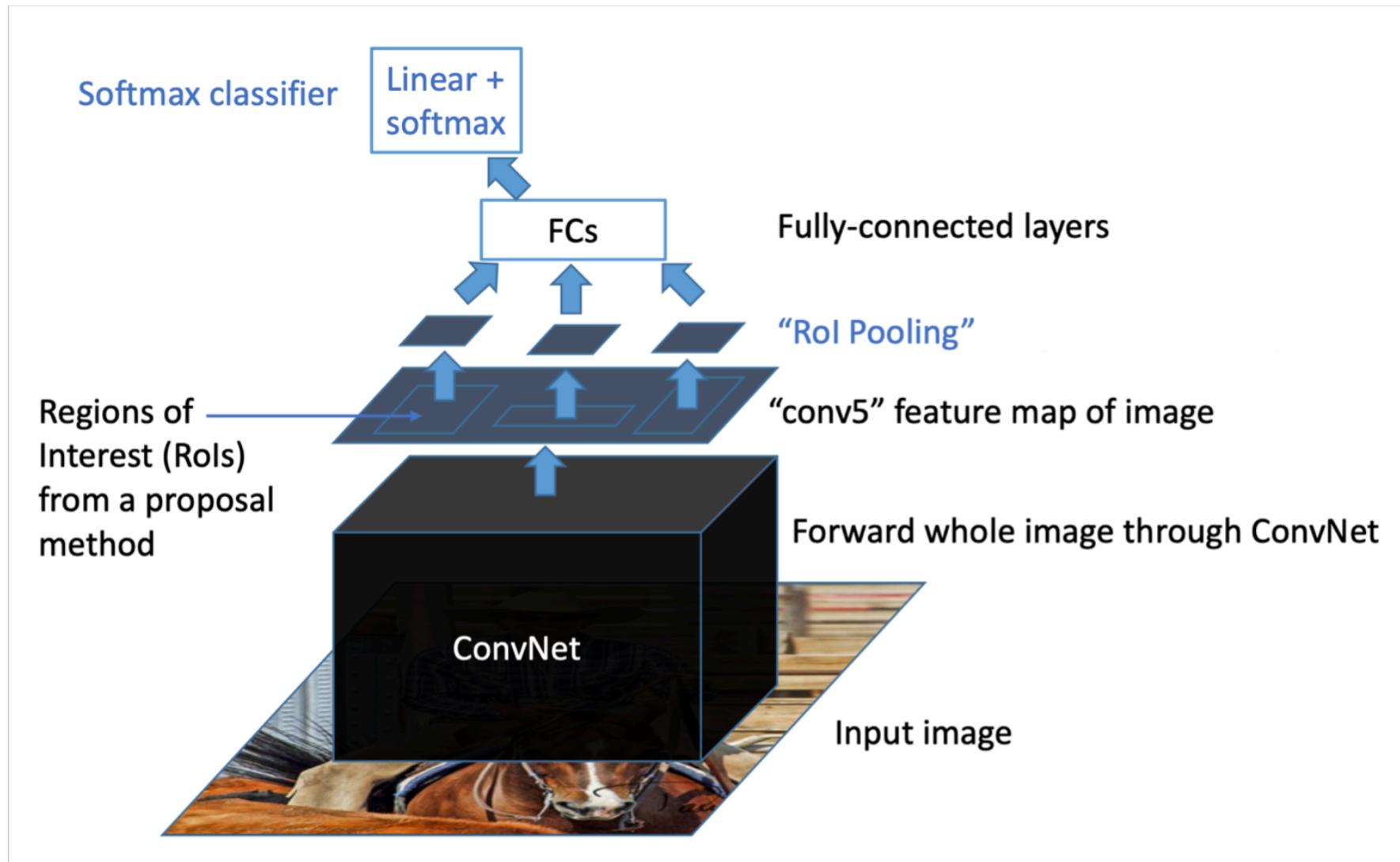
Fast RCNN (test time)



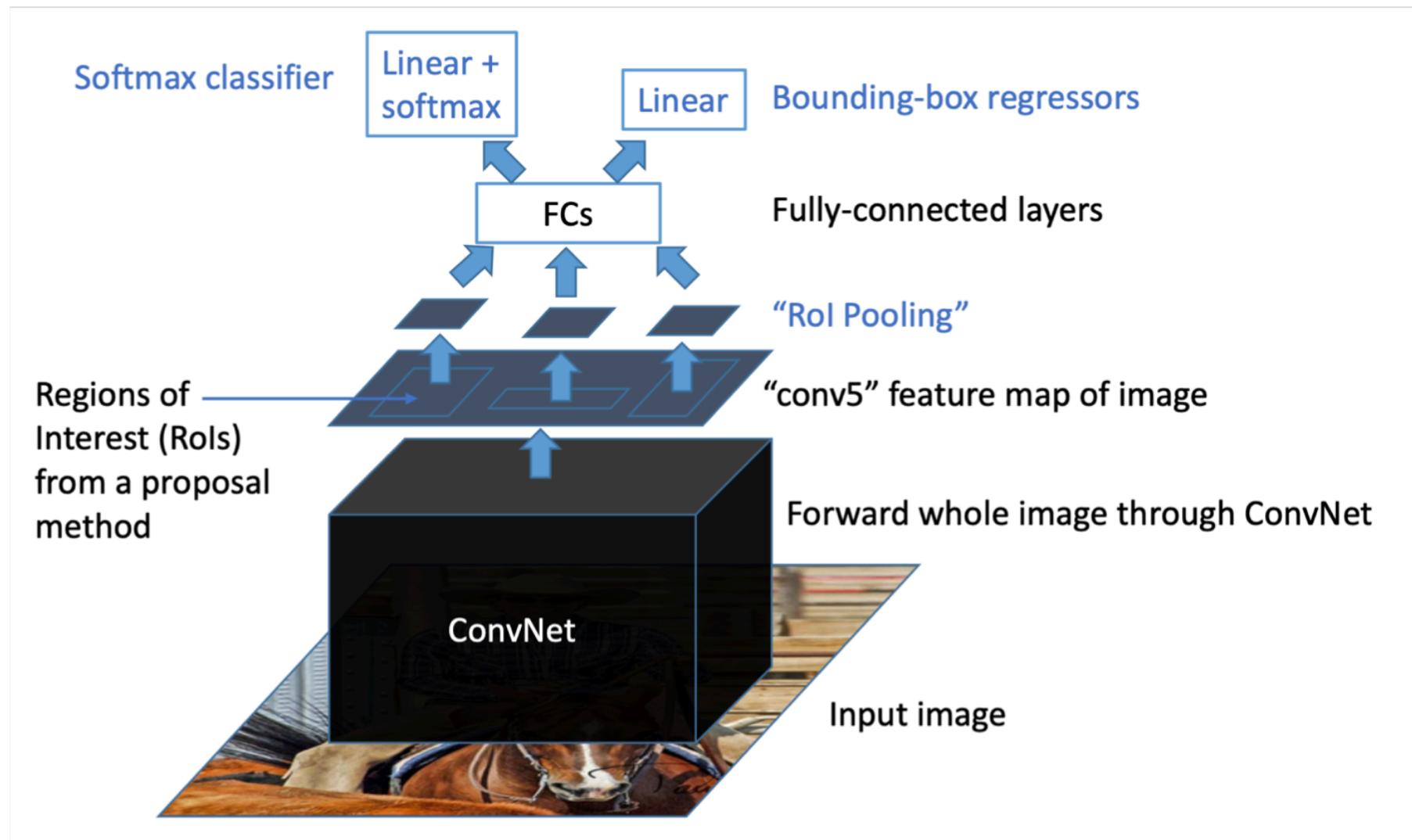
Fast RCNN (test time)



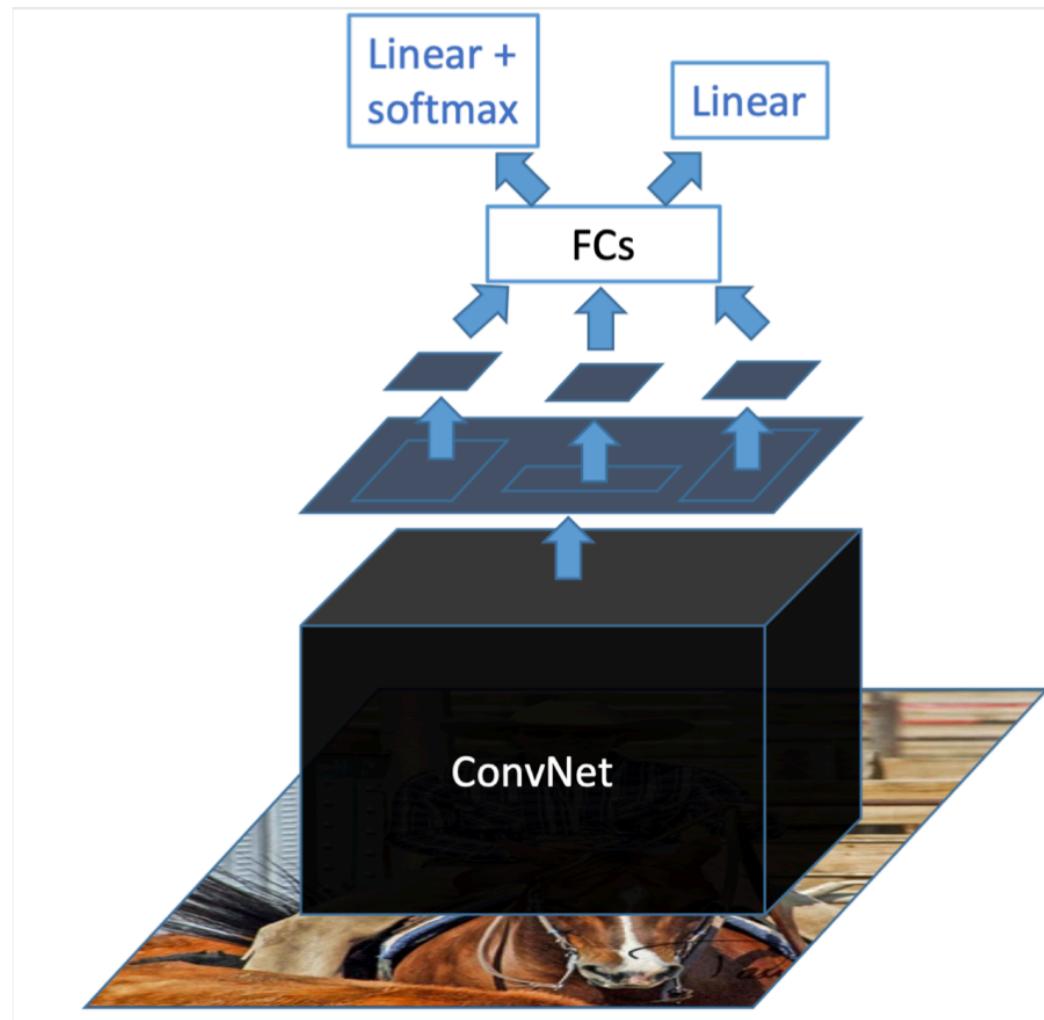
Fast RCNN (test time)



Fast RCNN (test time)

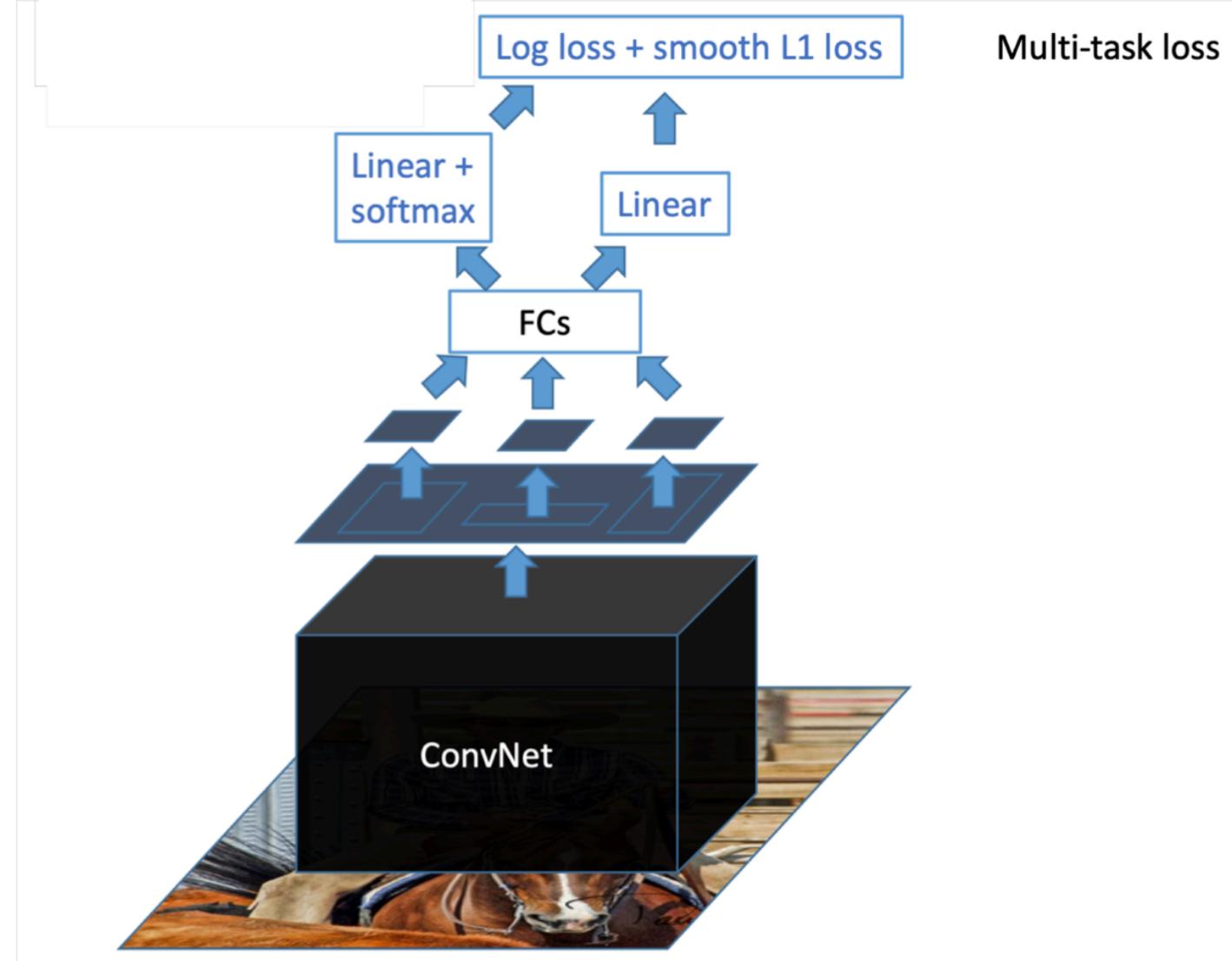


Fast RCNN (training)



Slide credit: Ross Girshick

Fast RCNN (training)



Results (RCNN vs Fast RCNN)

	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

Fast RCNN

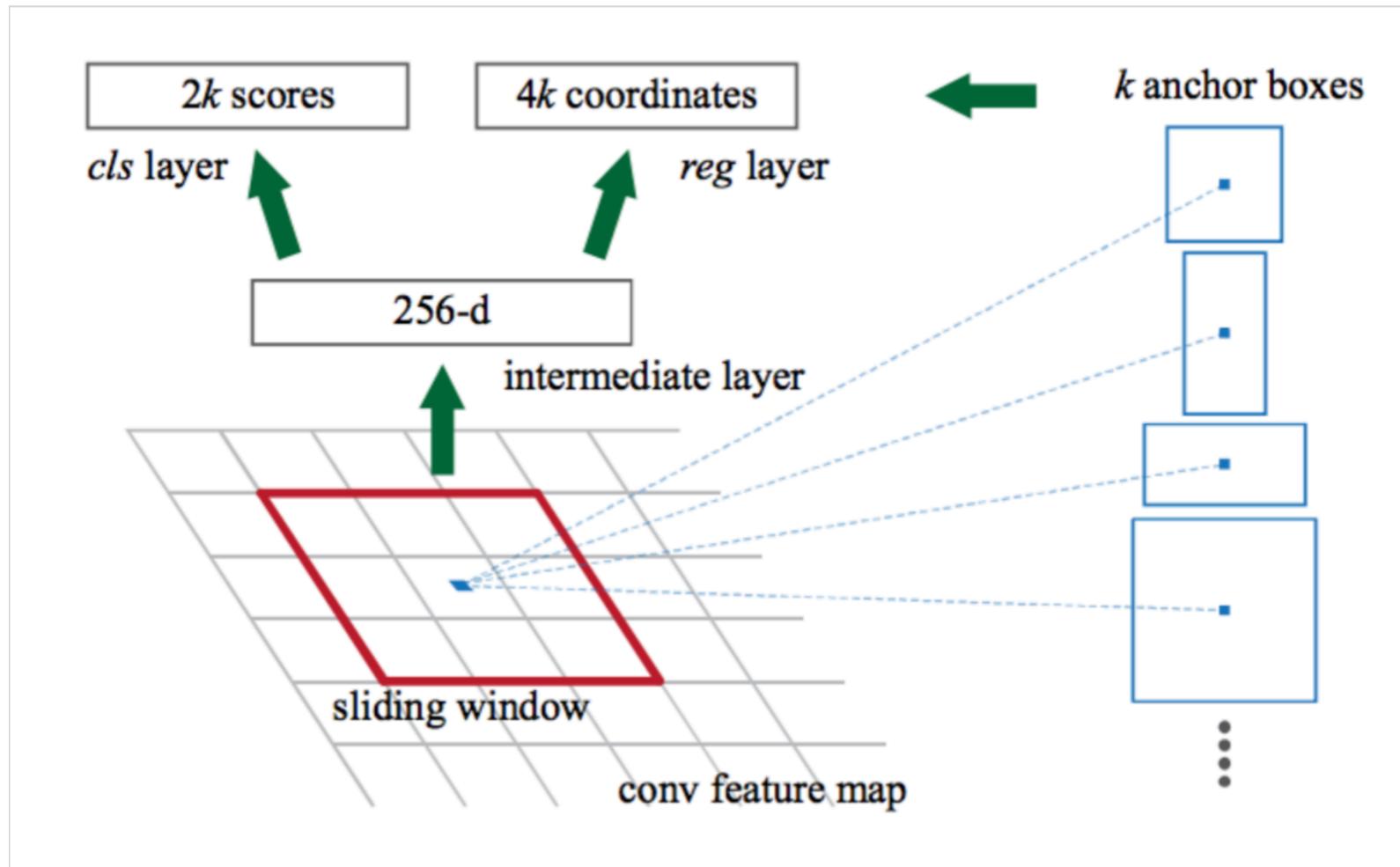
- Problem
 - Test-time speeds do not include region proposals

	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:
 - Just make the CNN do region proposals too!

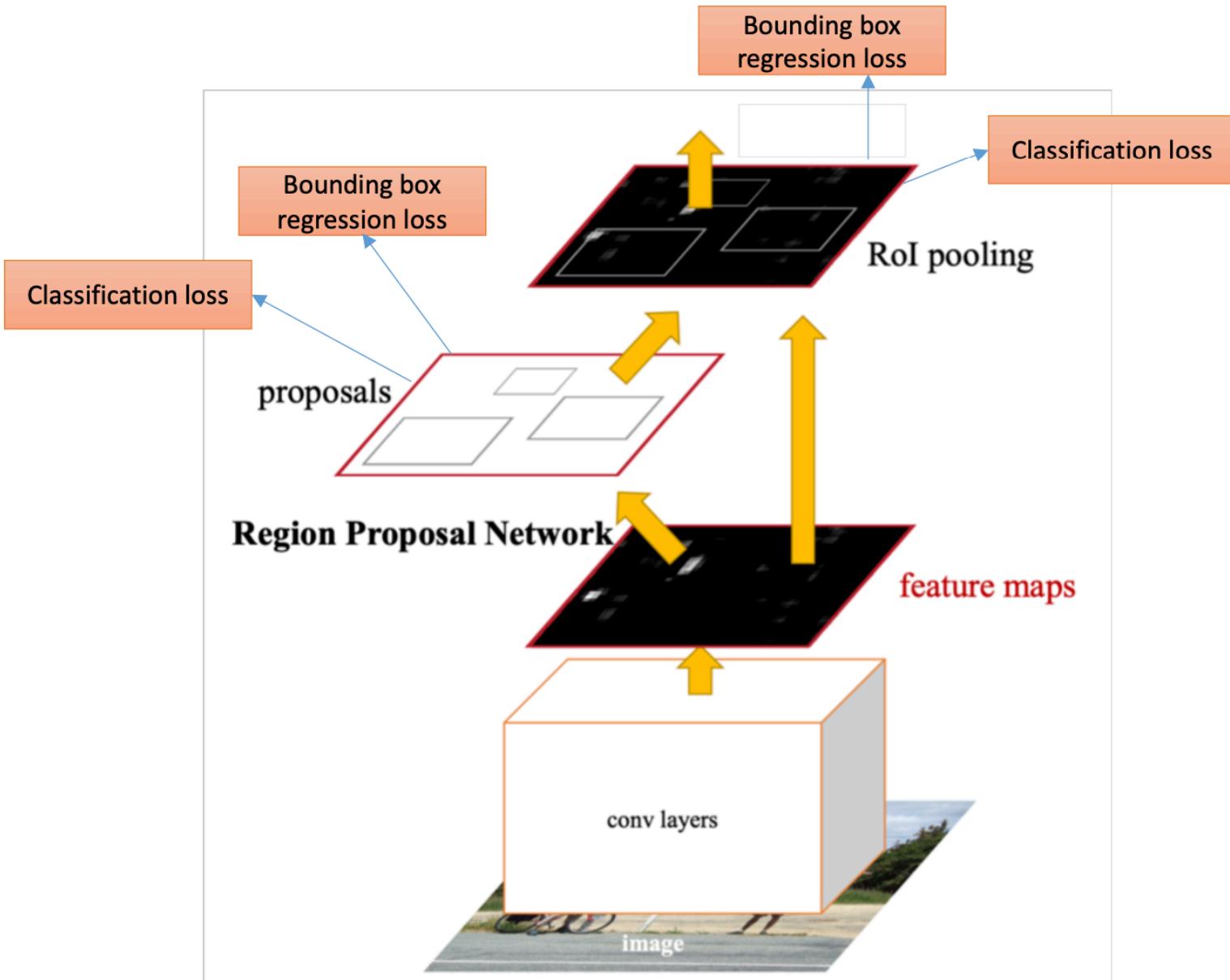
Faster RCNN

- Region Proposal Network (RPN)



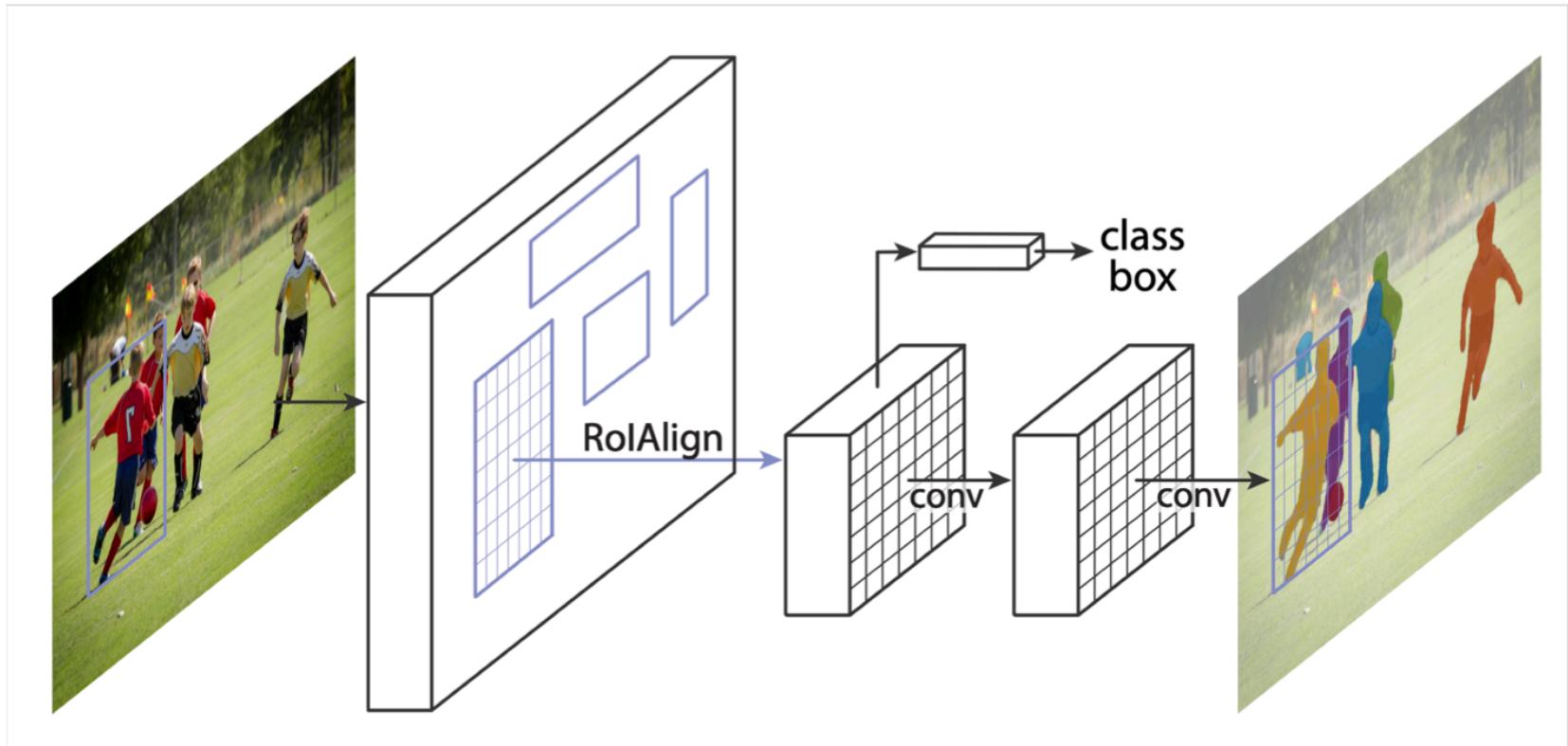
Faster RCNN

- **Training**
- One network, four losses
 - RPN classification
 - RPN regression
 - Fast R-CNN classification
 - Fast R-CNN regression



Object Instance Segmentation: Mask RCNN

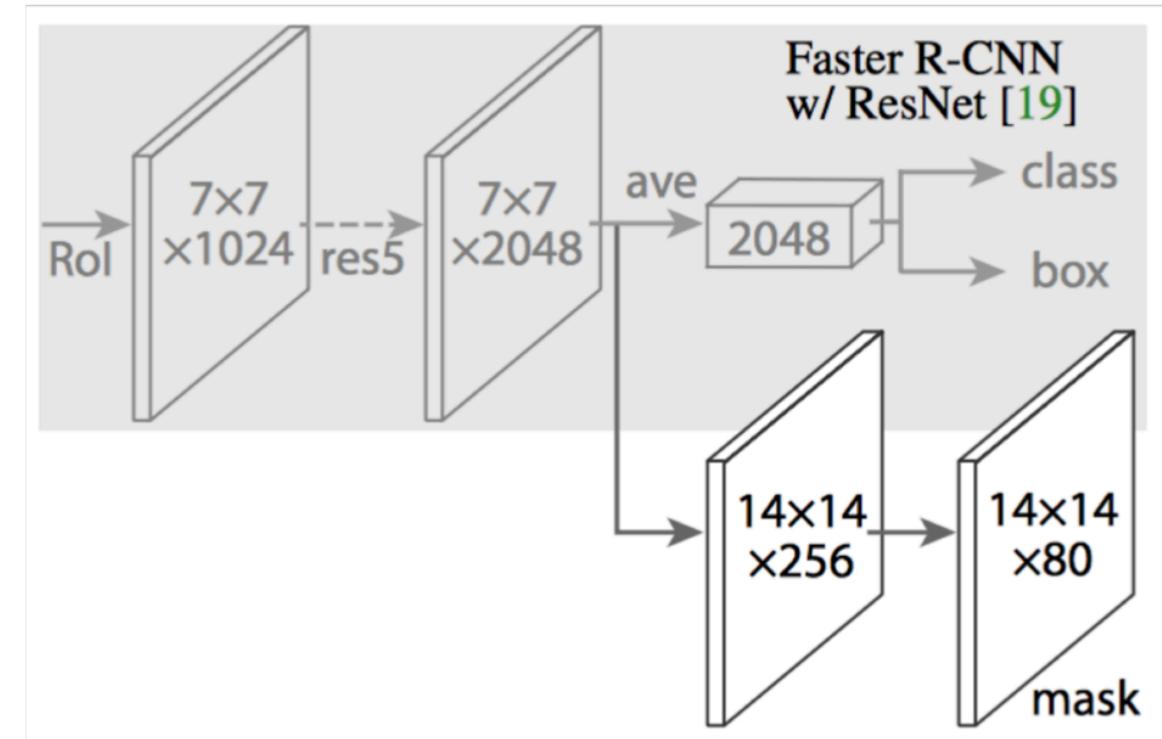
- Extending Faster R-CNN for Pixel Level Segmentation



Object Instance Segmentation: Mask RCNN

- Faster R-CNN has two outputs for each candidate object, a **class label** and a **bounding-box offset**
- In the Mask R-CNN they add a third branch that outputs the **object mask**.
- During training, they define a multi-task loss on each sampled RoI:

$$L = L_{cls} + L_{box} + L_{mask}$$



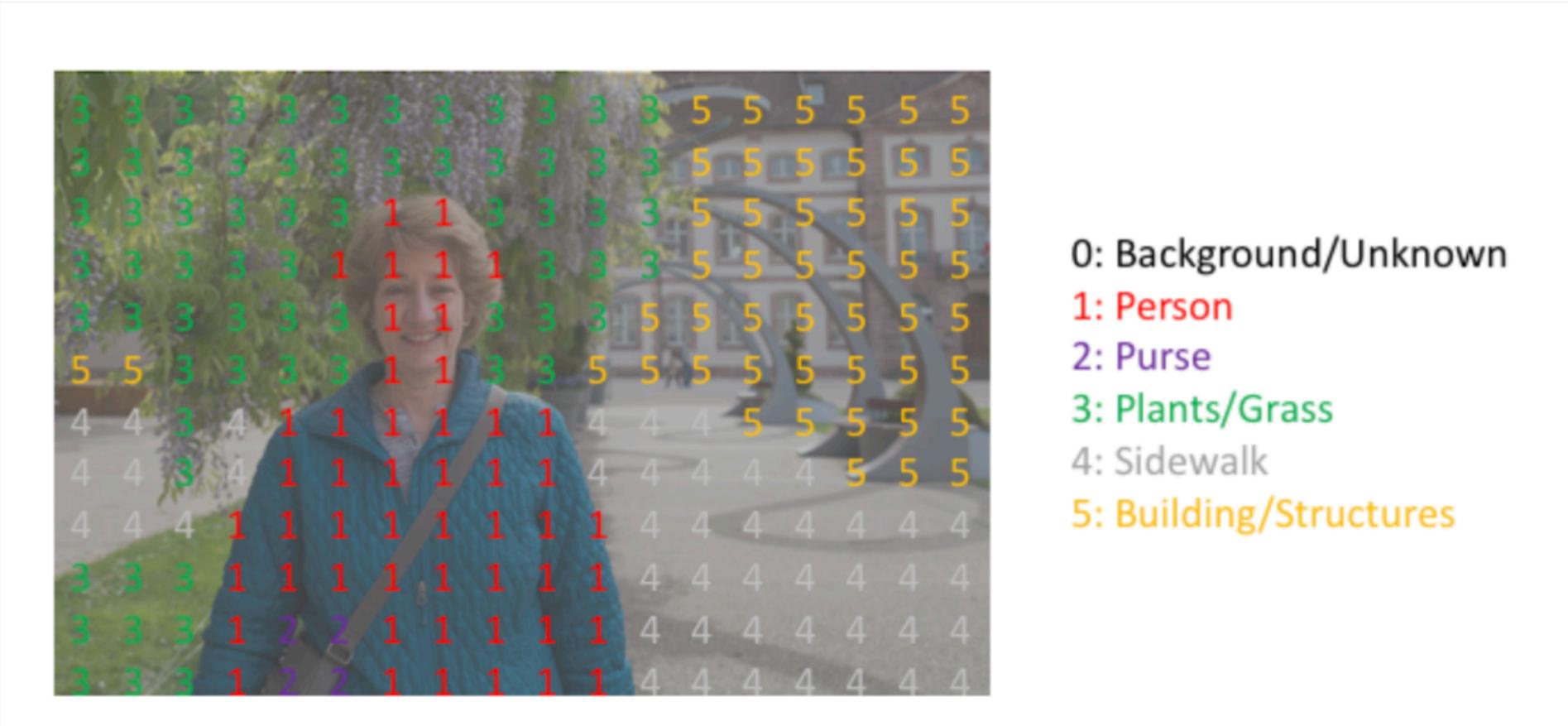
CNNs for Semantic Segmentation

CNNs for Semantic Segmentation

- Semantic Segmentation Problem
- Evaluation Metrics
- CNN Architectures for Semantic Segmentation
 - FCN
 - U-net

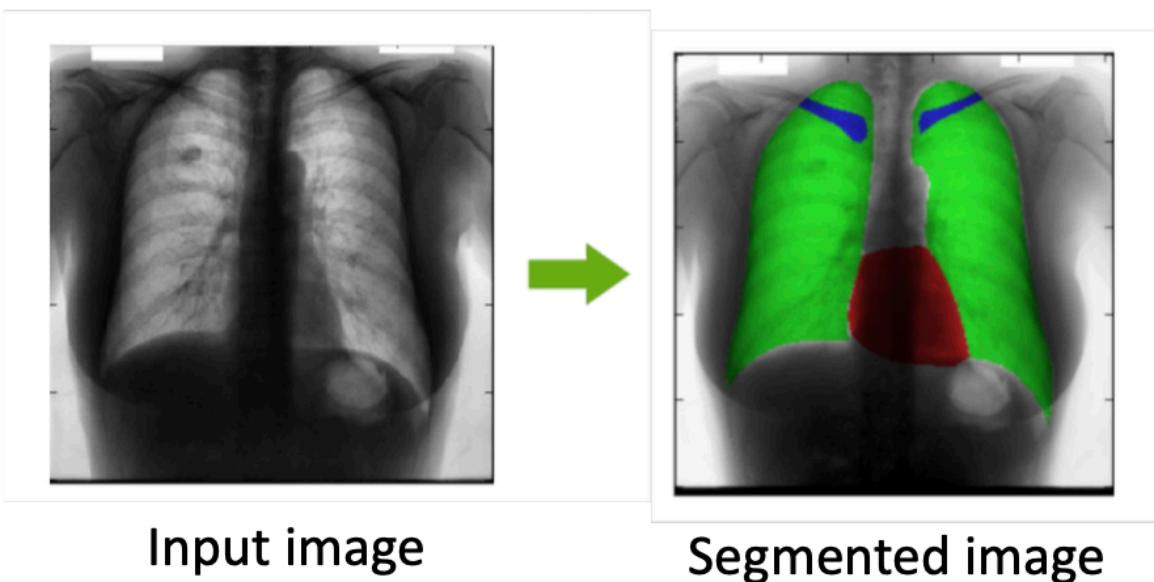
Semantic Segmentation Problem

- Classifying each pixel of an image into a category corresponding to an object or a part of the image (road, sky, ...).



Semantic Segmentation Problem

- Applications:
 - self-driving vehicles
 - human-computer interaction
 - virtual reality
 - bio Medical Image Diagnosis



Evaluation

- **Per Class Pixel Accuracy**
- The percent of pixels which were correctly classified for each class

$$\text{pixel accuracy}_i = \frac{\text{the number of pixels of class } i \text{ predicted to belong to class } i}{\text{the total number of pixels of class } i}$$

- Suitable for datasets with no background class but has a strong drawback for datasets with a large background class

Evaluation

- **The Intersection over Union (IoU)**
 - also referred to as the Jaccard index



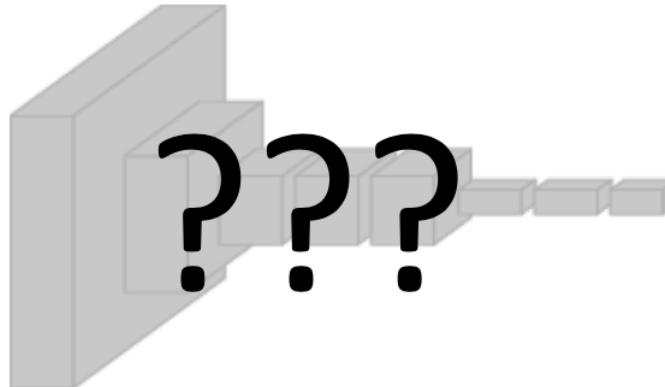
$$IoU = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}}$$

Semantic Segmentation Problem



$3 \times H \times W$

< 1/5 second

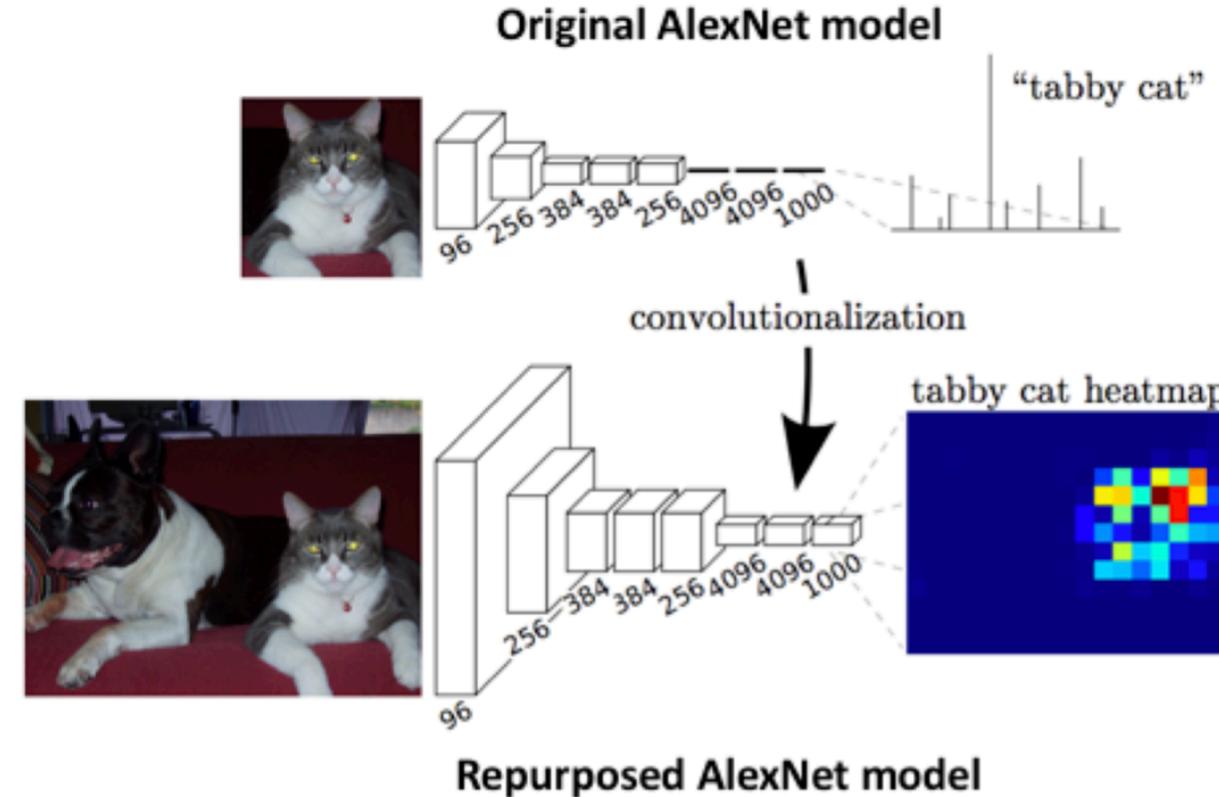


end-to-end learning



$H \times W$

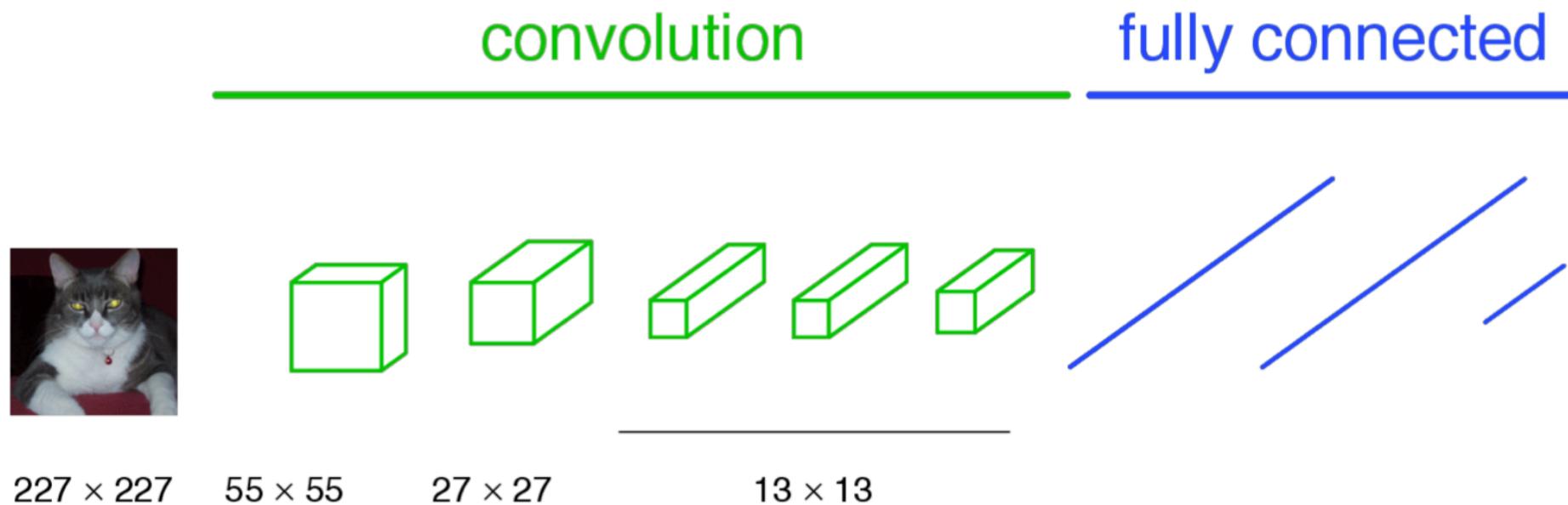
Fully Convolutional Network (FCN)



The encoder produces a **coarse** feature map which is then refined by the decoder module.

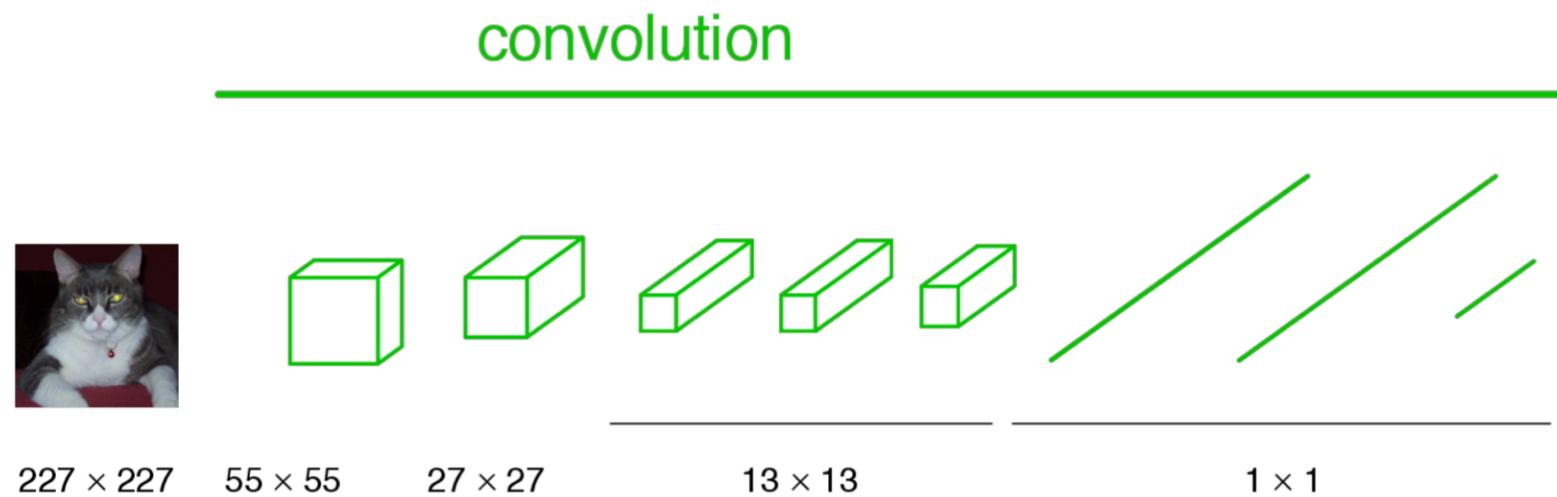
Fully Convolutional Network (FCN)

- a classification network



Fully Convolutional Network (FCN)

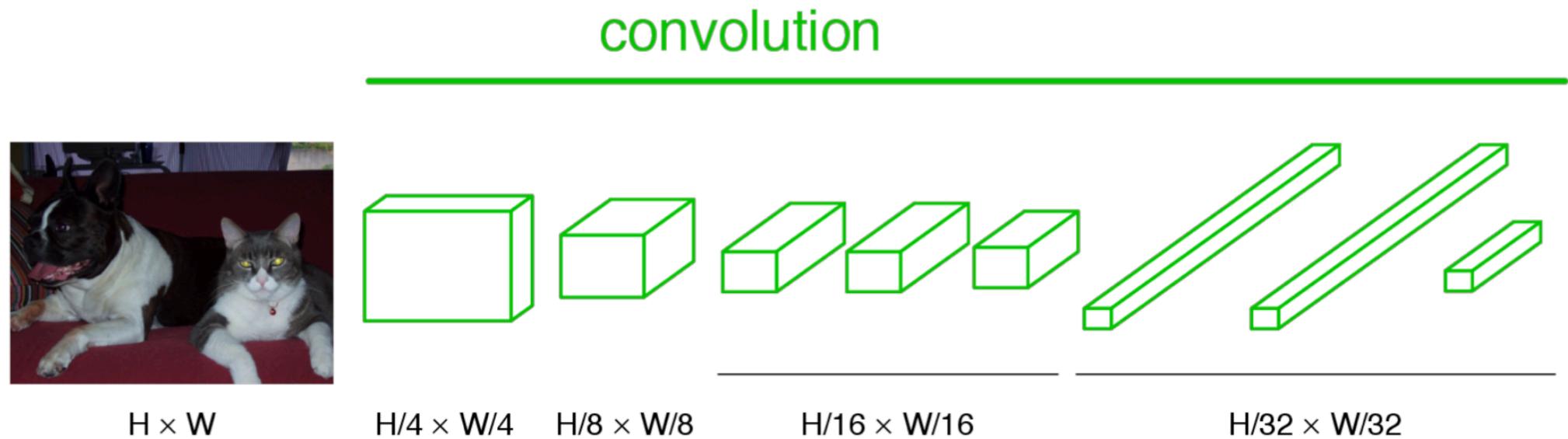
- becoming fully convolutional



Slide credit: Jonathan Long

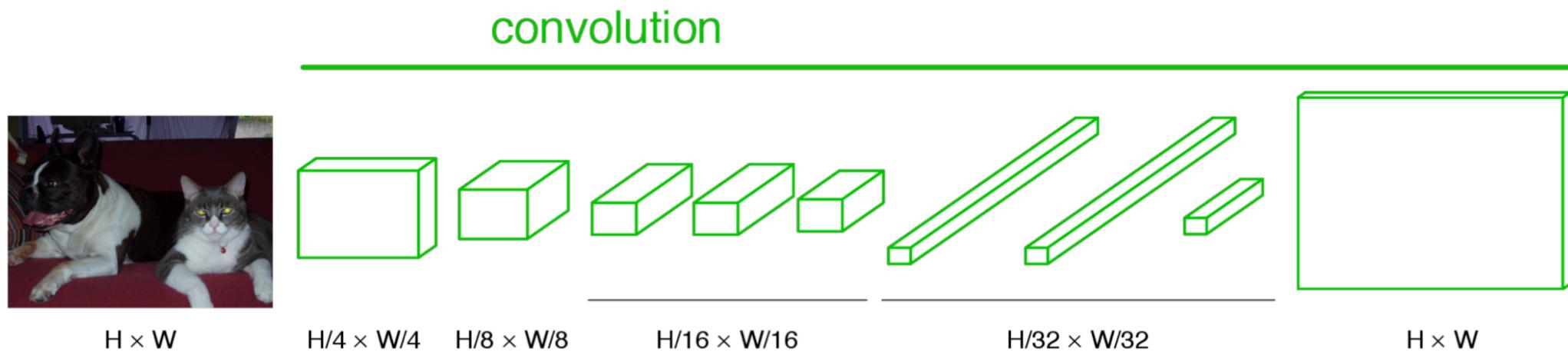
Fully Convolutional Network (FCN)

- becoming fully convolutional



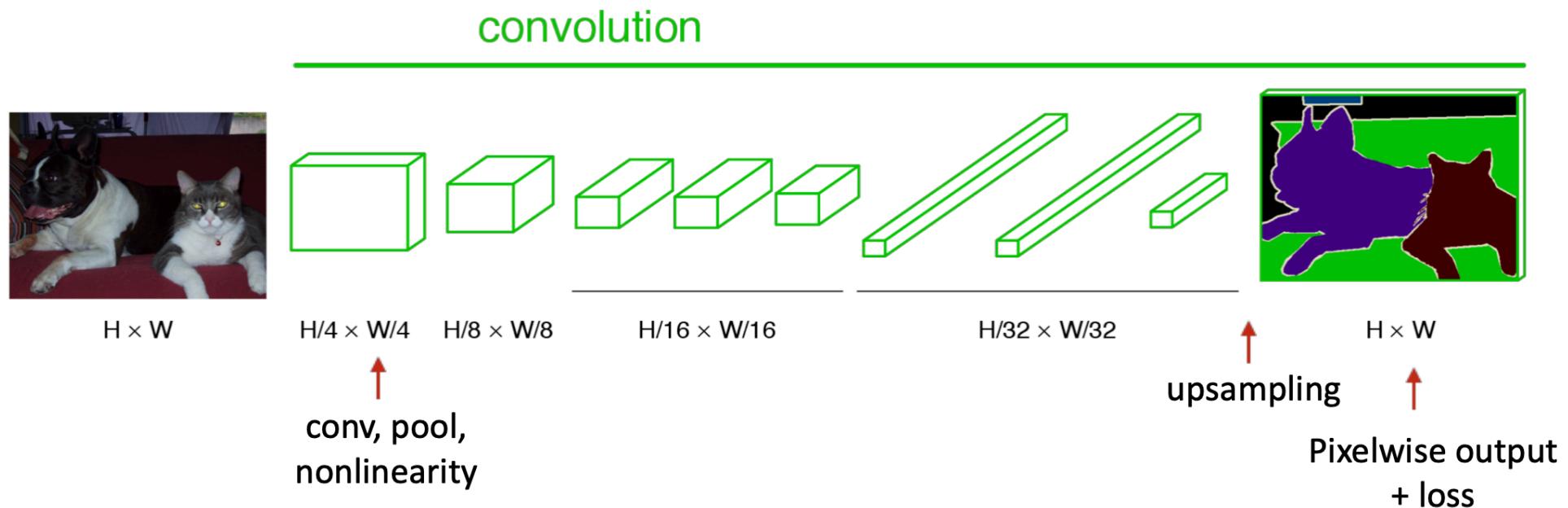
Fully Convolutional Network (FCN)

- Upsampling output



Fully Convolutional Network (FCN)

- End-to-end, pixel-to-pixel network



U-Net

- U-net composed in two parts:
 - contracting or downsampling
 - expanding or upsampling
- Feature maps from the downsampling part of the network are copied within the upsampling part
 - to avoid loosing pattern information.
- Finally, a 1×1 convolution processes the feature maps
 - to generate a segmentation map and thus categorise each pixel of the input image.

