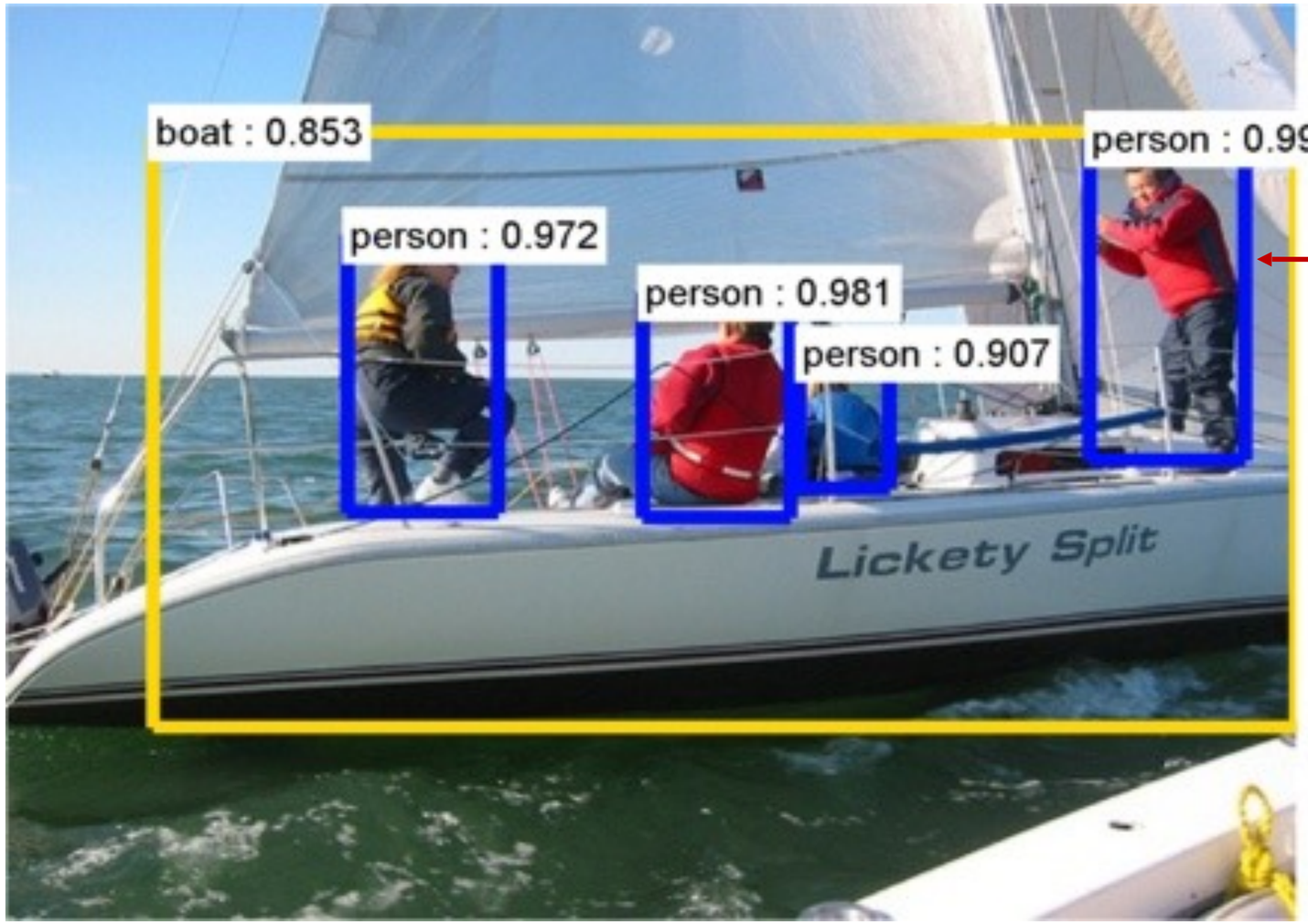
Lecture 10: Object Detection

Contains slides from S. Lazebnik, R. Girshick, B. Hariharan

Object detection with bounding boxes



What?

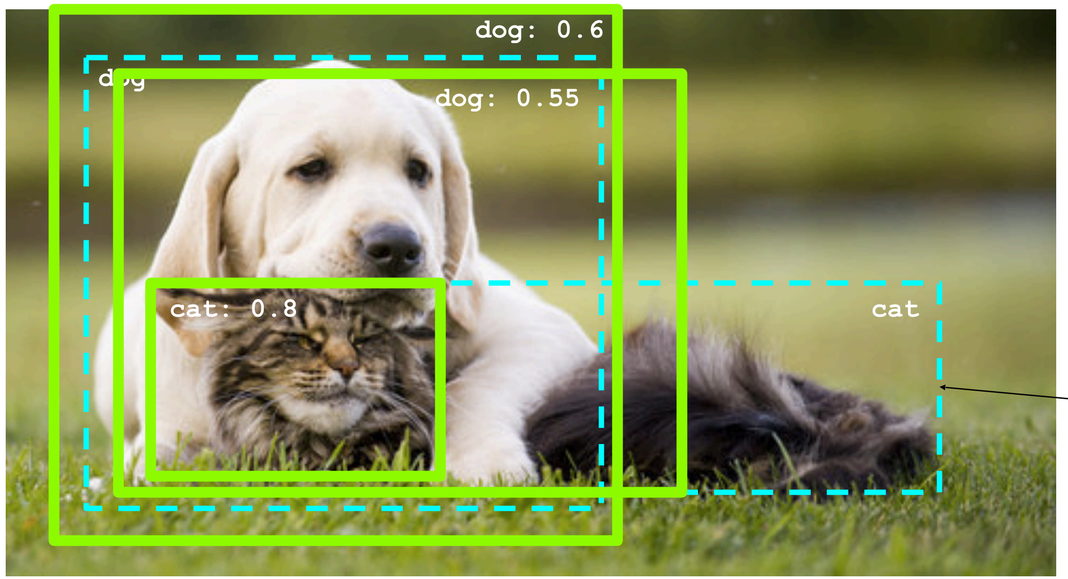
Where?

"Object detection"

Source: R. Girshick

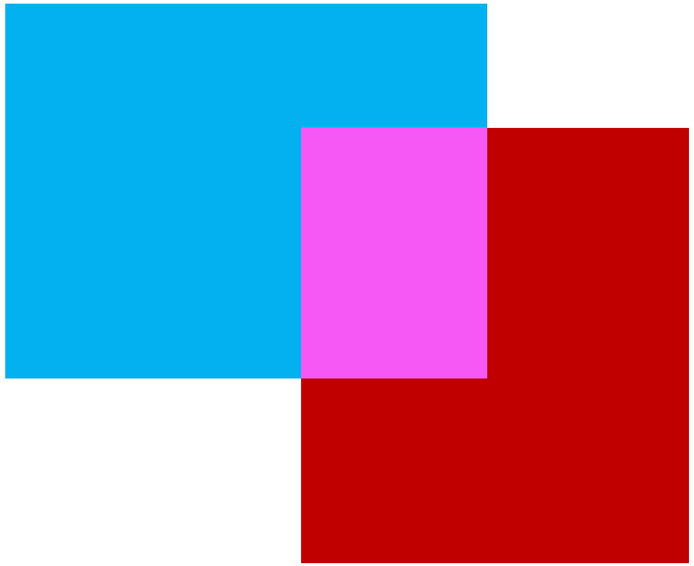
Evaluating an object detector

- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a true or false positive
  - **Intersection over union** (IoU): Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5



Source: S. Lazebnik

Evaluating an object detector



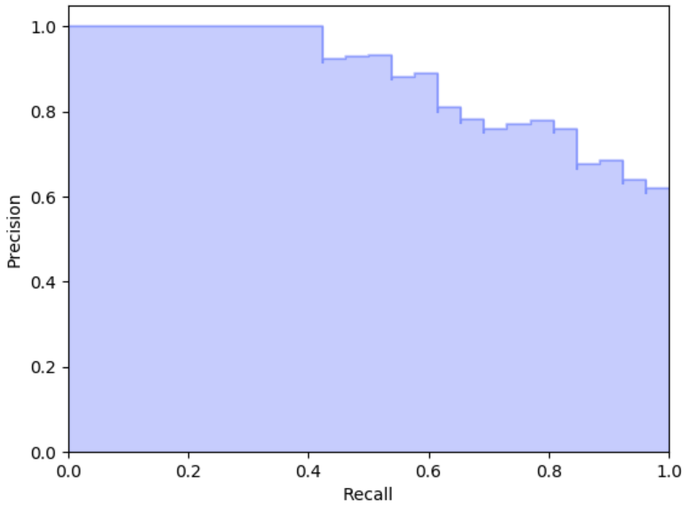
$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Intersection over union (also known as Jaccard similarity)

Source: B. Hariharan

Evaluating an object detector

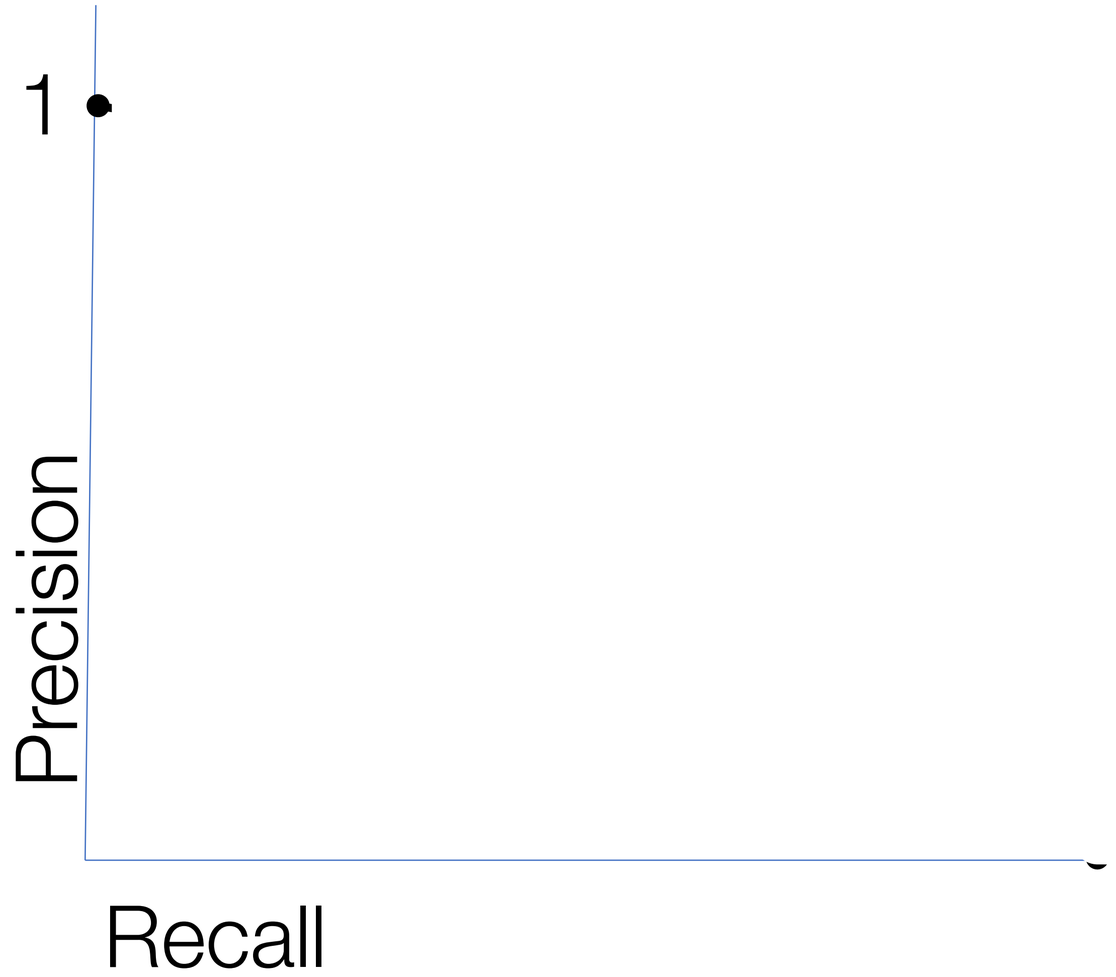
- For each class, plot Recall-Precision curve and compute Average Precision (area under the curve)
- Take mean of AP over classes to get mAP



Precision:
true positive detections / total detections
Recall:
true positive detections / total positive test instances

Source: S. Lazebnik

Average precision



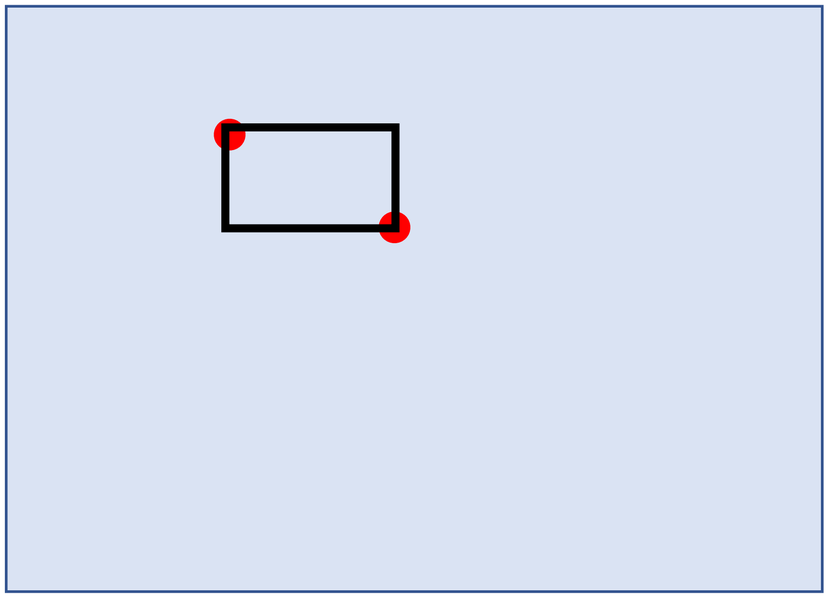
Source: B. Hariharan

Average precision

Source: B. Hariharan

Detection as classification

- Run through every possible box and classify
  - Well-localized object of class k or not?
- How many boxes?
  - Every pair of pixels = 1 box
  - $\begin{pmatrix} N \\ 2 \end{pmatrix}$ = $O(N^2)$
  - For 300 x 500 image, N = 150K
  - 2.25 x 10^{10} boxes!
- Related challenge: almost all boxes are negative!

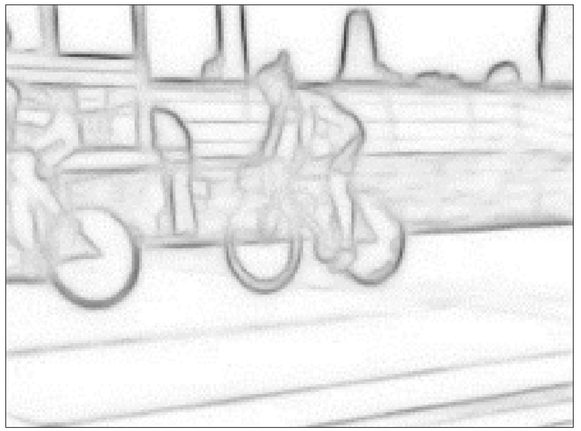


Source: B. Hariharan

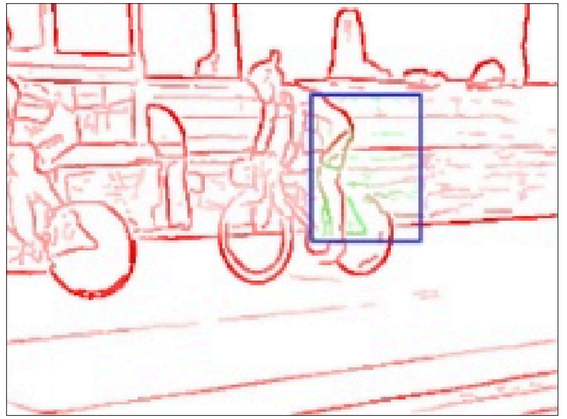
Selective search

Stage 1: generate candidate bounding boxes

Input image



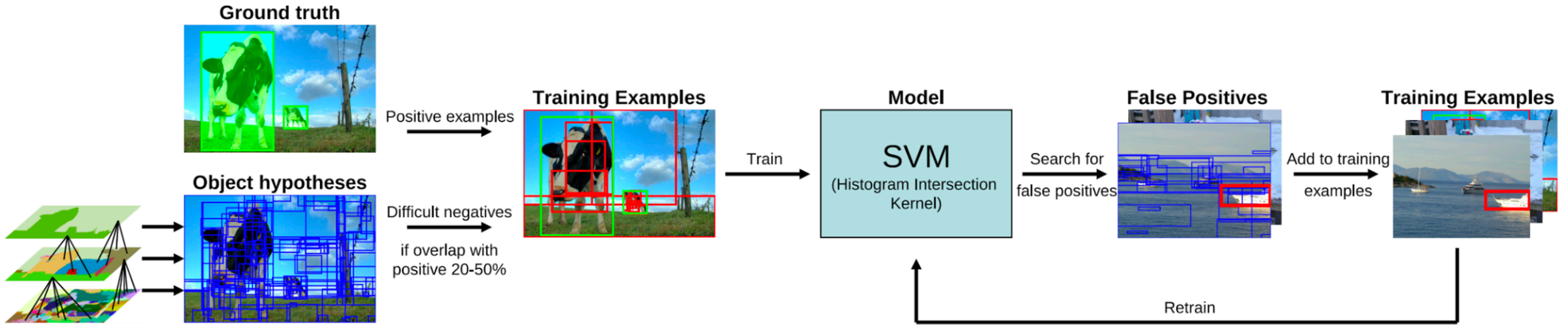
Edge detection



Bounding box proposal

[Zitnick and Dollar, "Edge Boxes…", 2014]

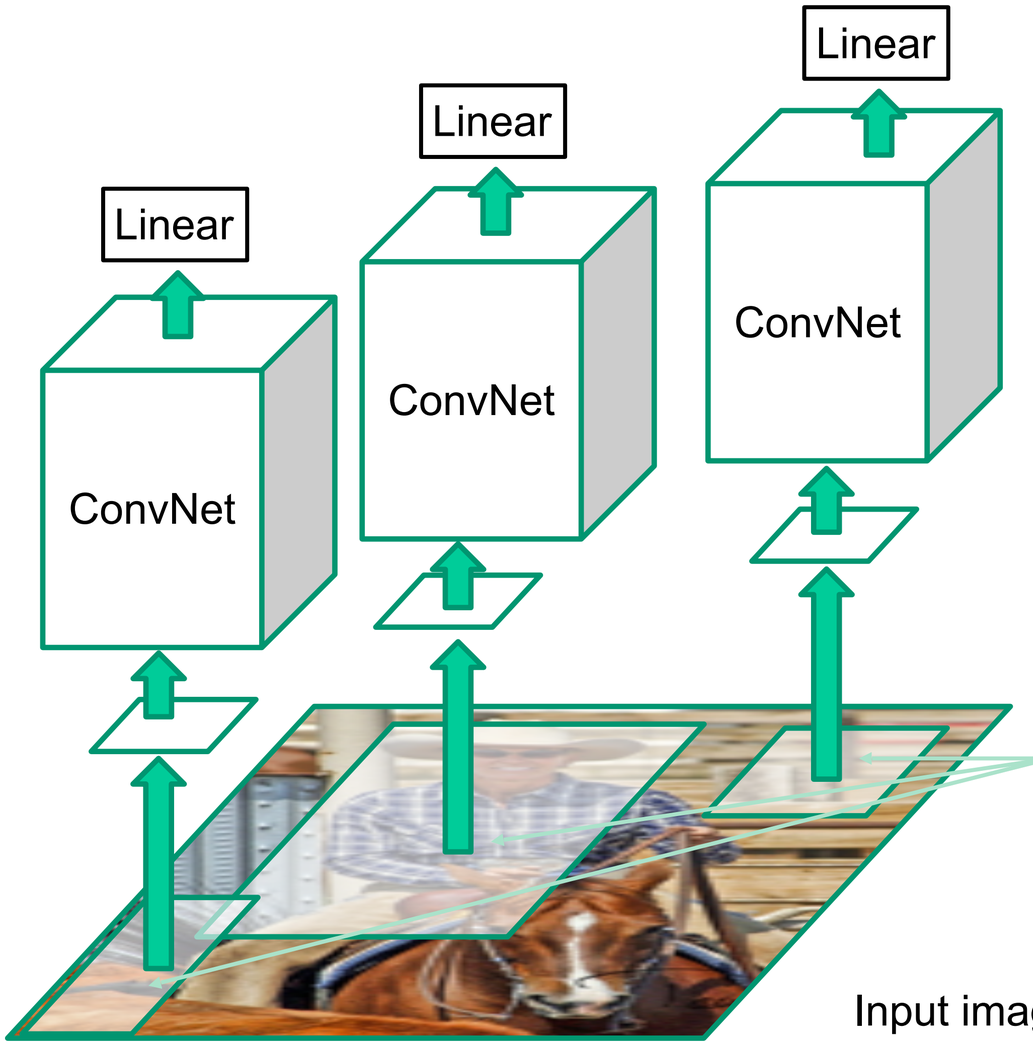
Stage 2: apply classifier only to each candidate bounding box



Source: Torralba, Freeman, Isola

[Uijlings et al., "Selective Search for Object Recognition", 2013]

R-CNN: Region proposals + CNN features



Classify regions with linear classifier

Forward each region through ConvNet

Warped image regions

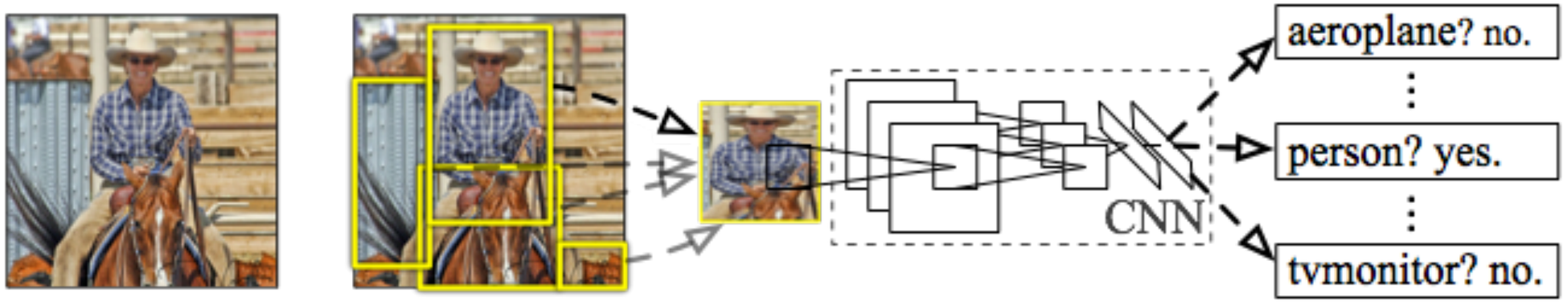
Region proposals from **selective search** (~2K rectangles that are likely to contain objects)

Input image

R. Girshick, J. Donahue, T. Darrell, and J. Malik, [Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation](), CVPR 2014.

Source: R. Girshick

R-CNN at test time



Input image

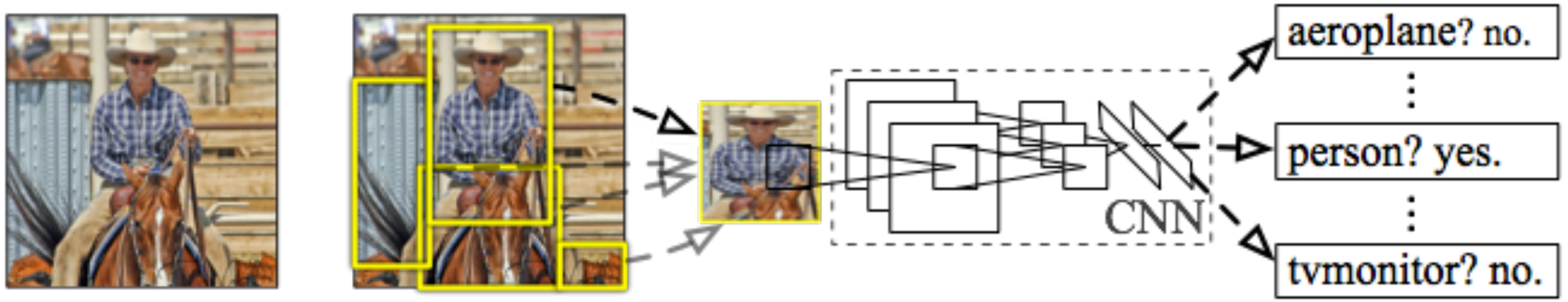
Extract region proposals (~2k / image)

Compute CNN features

a. Crop

Source: R. Girshick

R-CNN at test time



Input image

Extract region proposals (~2k / image)

Compute CNN features

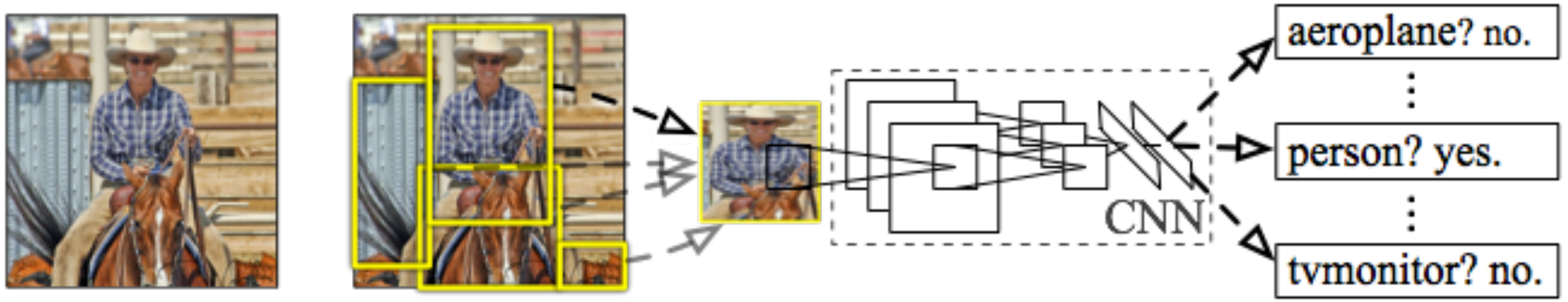
a. Crop

227 x 227

b. Scale (anisotropic)

Source: R. Girshick

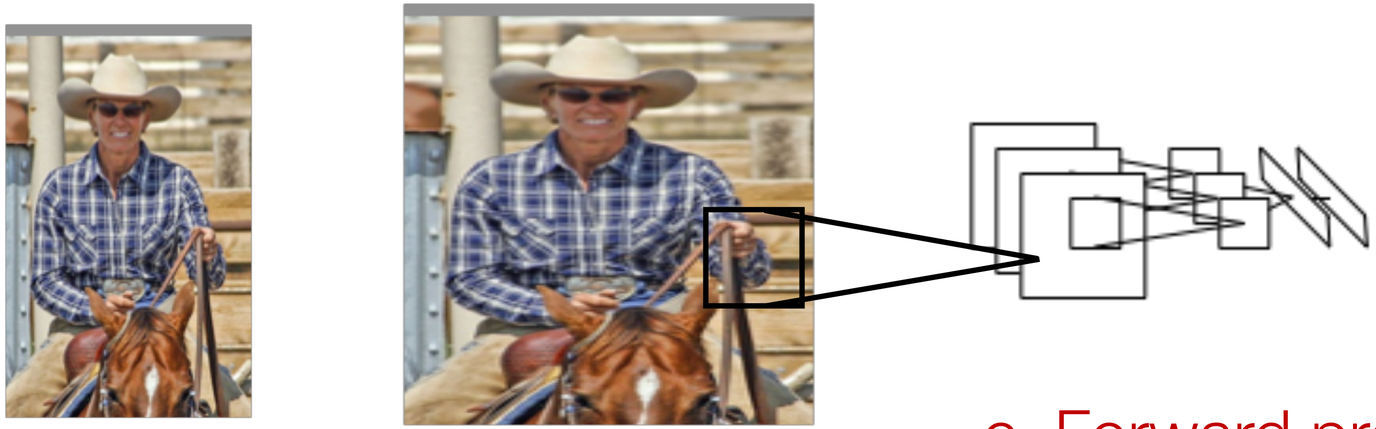
R-CNN at test time



Input image

Extract region proposals (~2k / image)

Compute CNN features



1. Crop

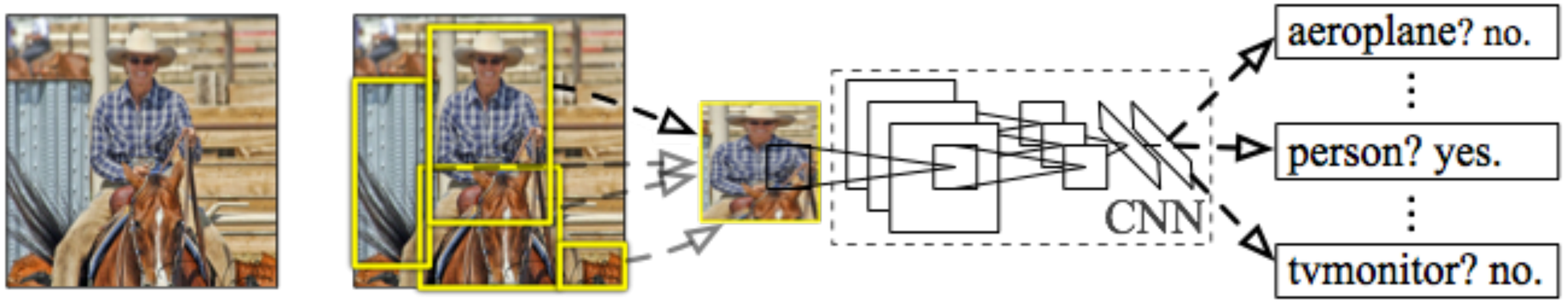
b. Scale (anisotropic)

c. Forward propagate

Output: "fc_7" features

Source: R. Girshick

R-CNN at test time

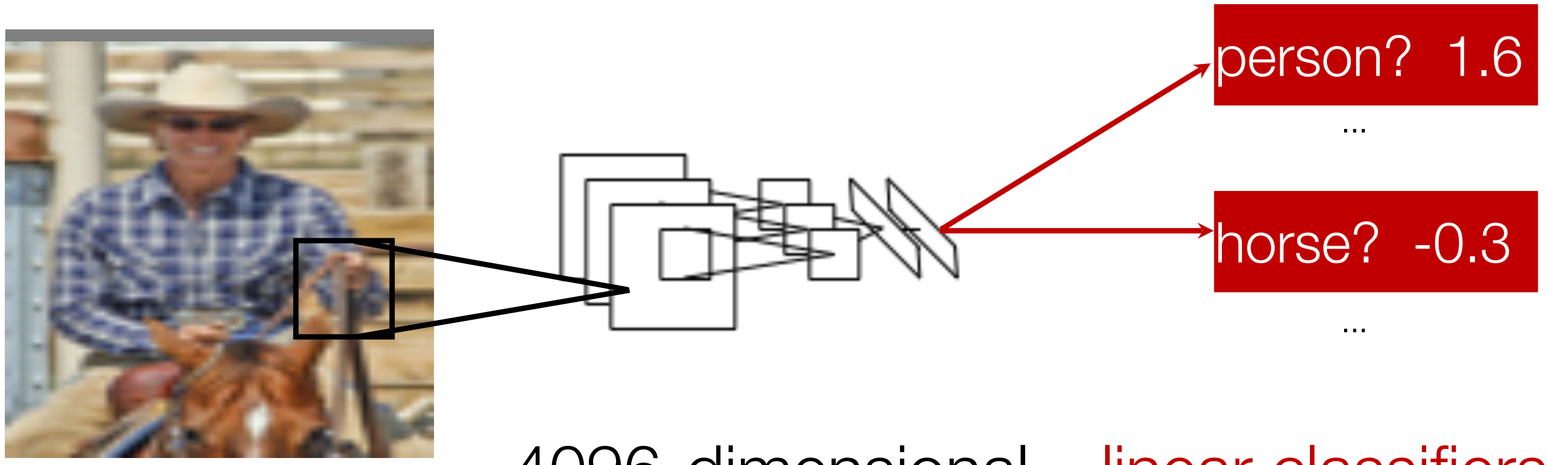


Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions



Warped proposal

4096-dimensional fc_7 feature vector

linear classifiers (SVM or softmax)

Source: R. Girshick

R-CNN at test time: proposal refinement

Linear regression

on CNN features

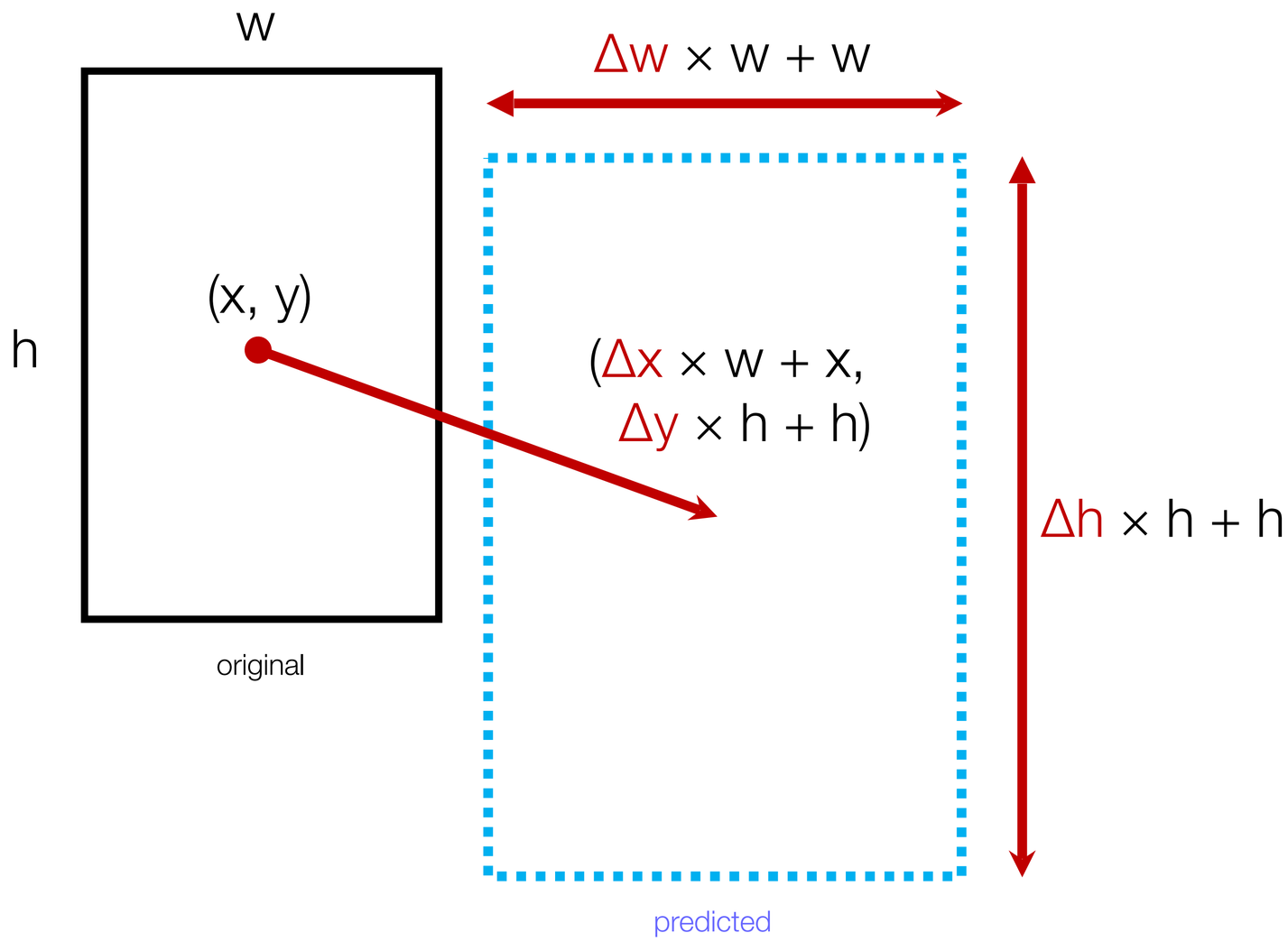
Original proposal

Predicted object bounding box

Bounding-box regression

Source: R. Girshick

Bounding-box regression



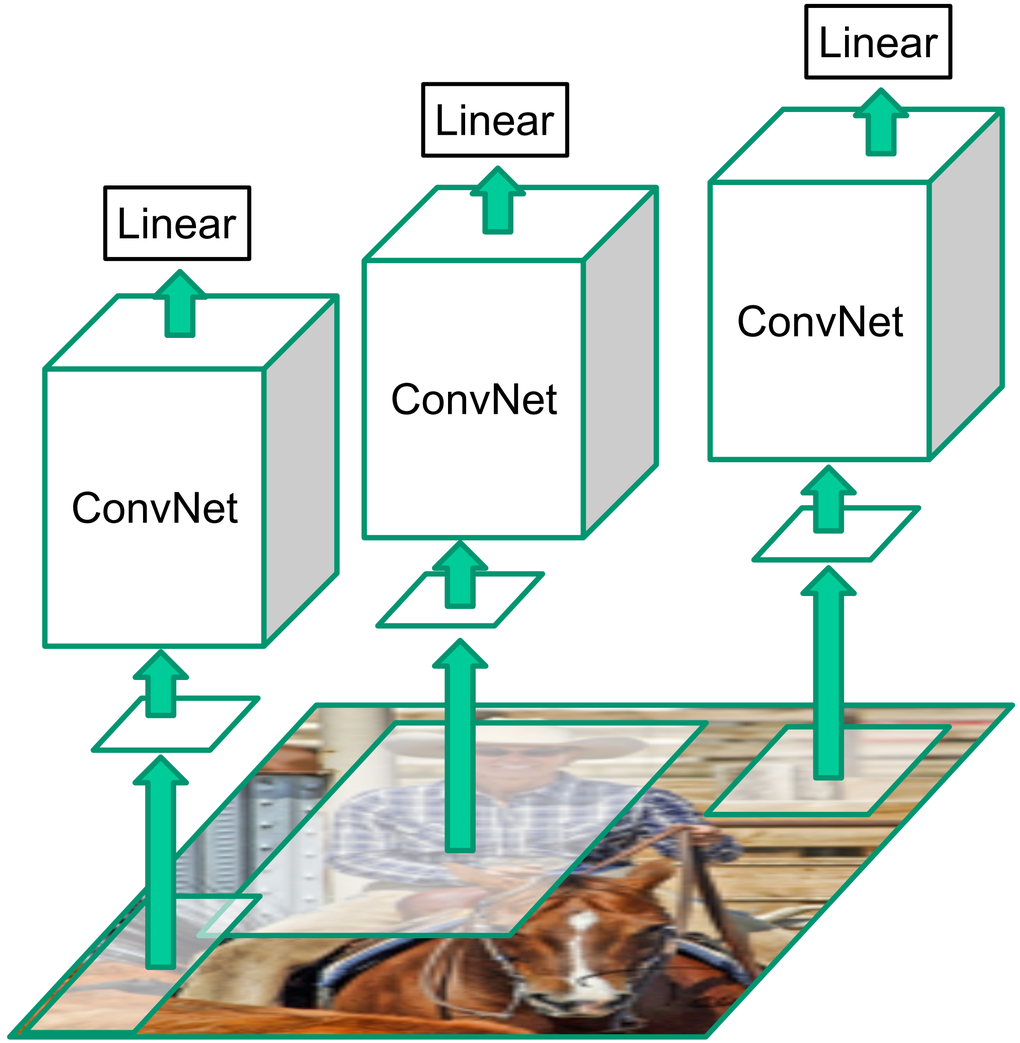
Source: R. Girshick

Non-maximum suppression

If two boxes overlap significantly (e.g. > 50% IoU), drop the one with the lower score. Usually use greedy algorithm.

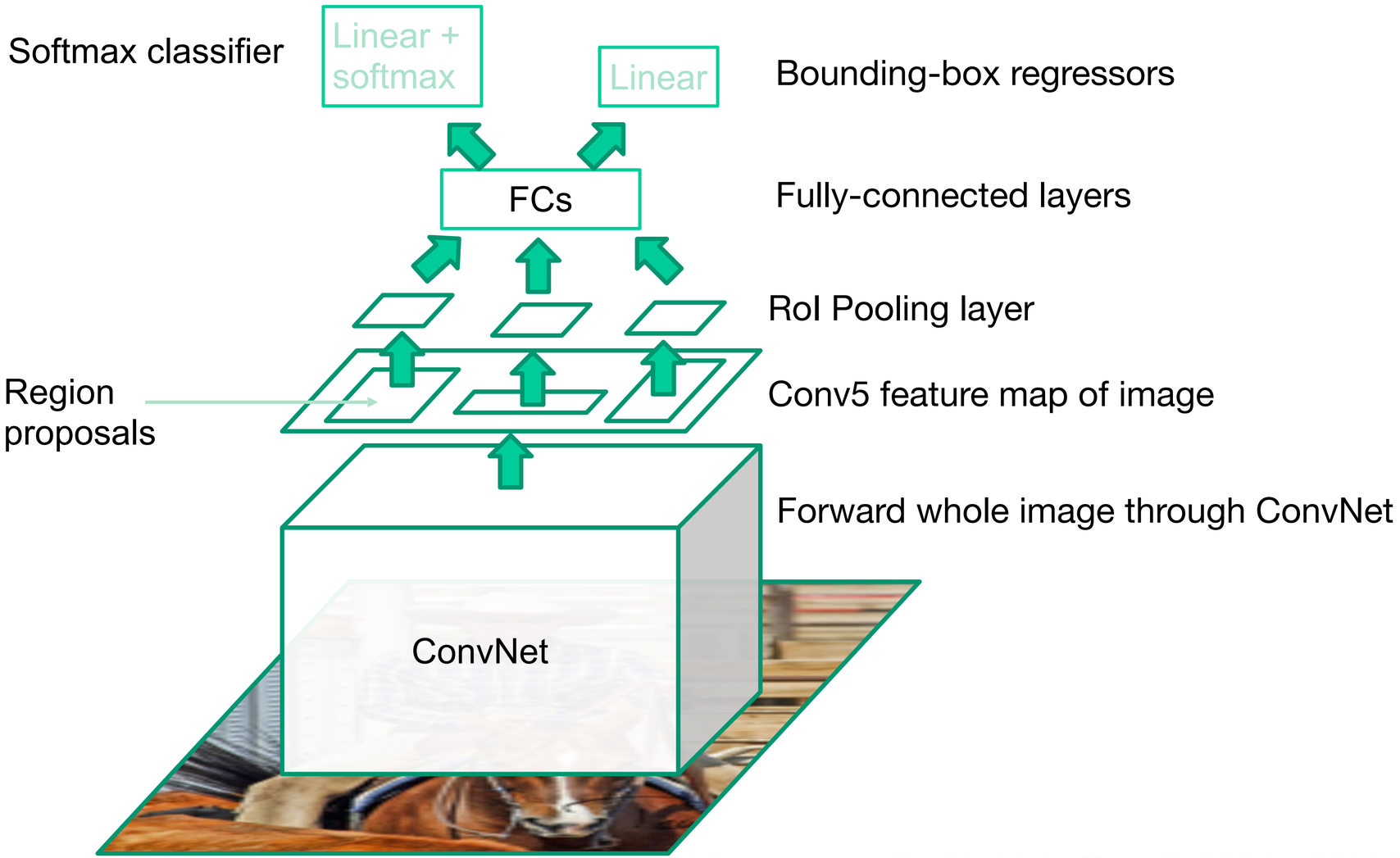
Source: B. Hariharan

Problems with R-CNN



1. Slow! Have to run CNN per window
2. Hand-crafted mechanism for region proposal might be suboptimal.

“Fast” R-CNN: reuse features between proposals

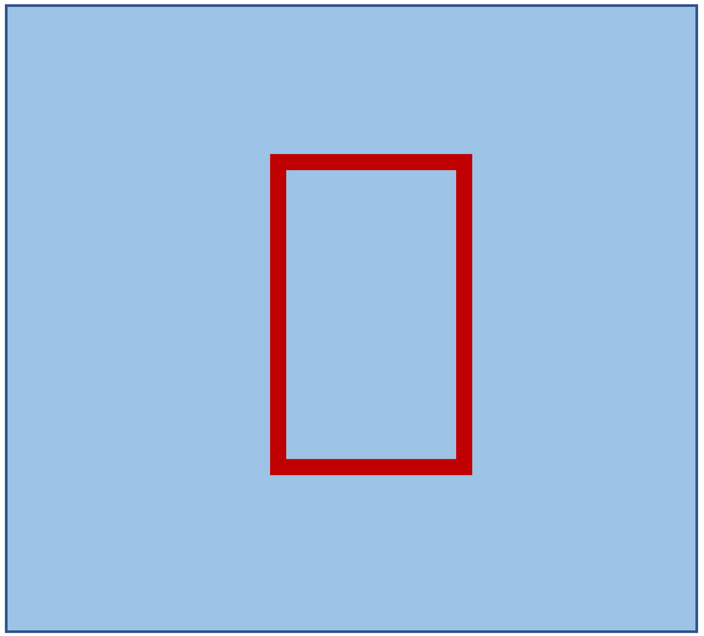


Source: R. Girshick

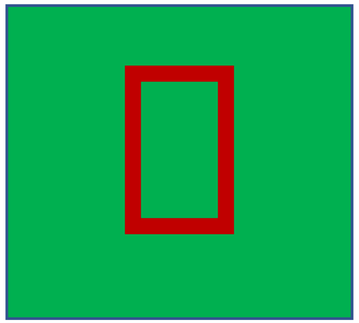
R. Girshick, Fast R-CNN, ICCV 2015

ROI Pooling

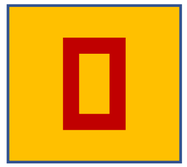
- How do we crop from a feature map?
- Step 1: Resize boxes to account for subsampling



Layer 1



Layer 2

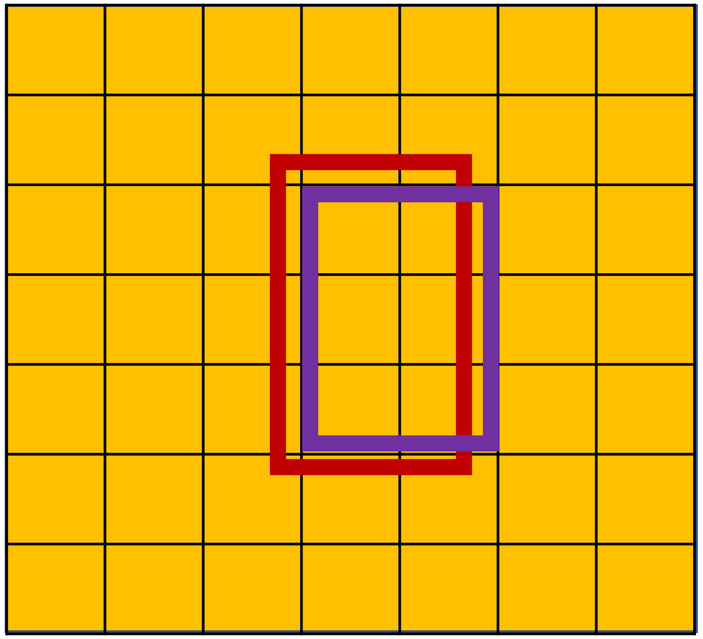


Layer 3

Source: B. Hariharan

ROI Pooling

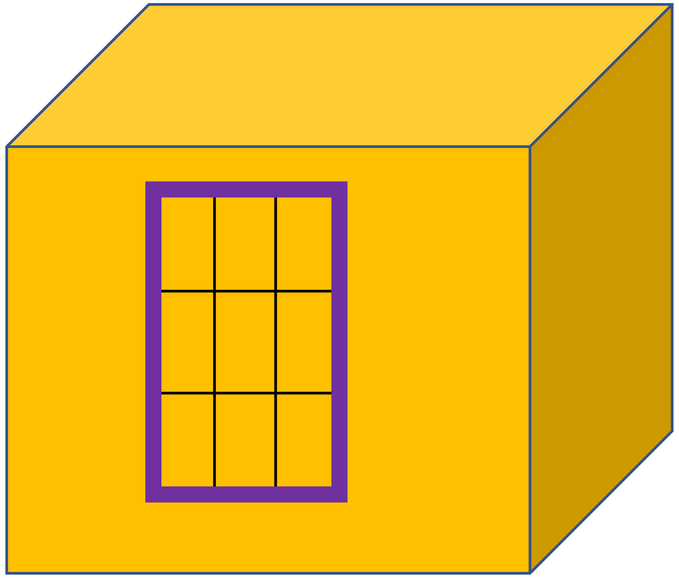
- How do we crop from a feature map?
- Step 2: Snap to feature map grid



Source: B. Hariharan

ROI Pooling

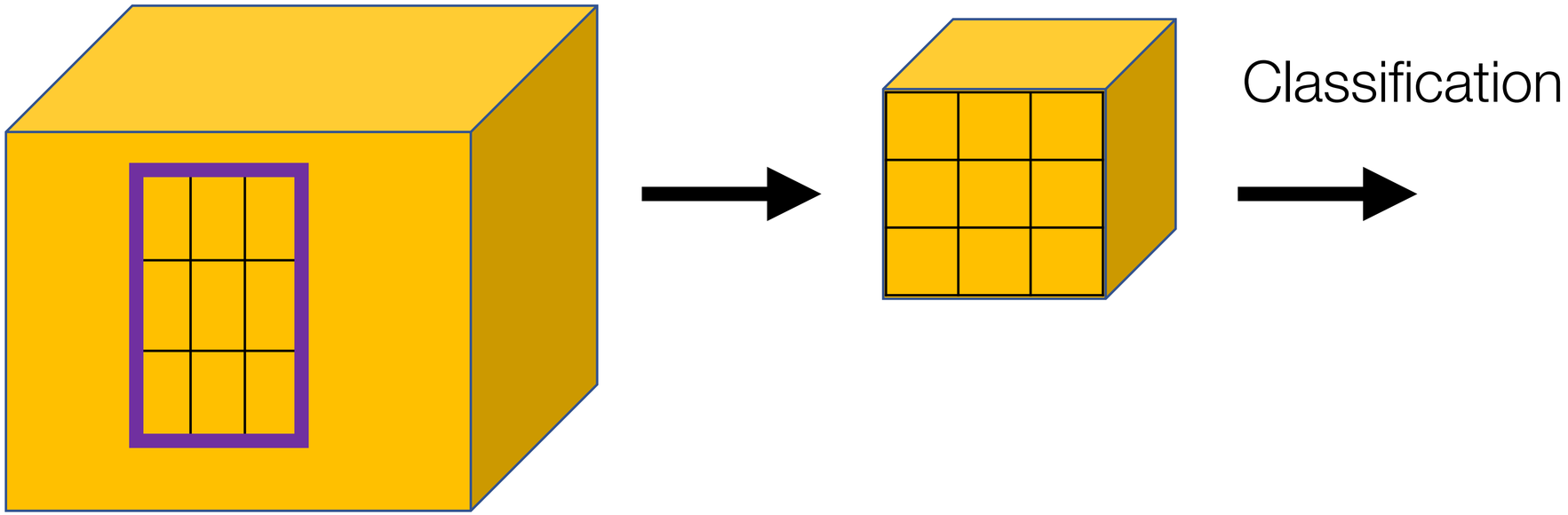
- How do we crop from a feature map?
- Step 3: Overlay a new grid of fixed size



Source: B. Hariharan

ROI Pooling

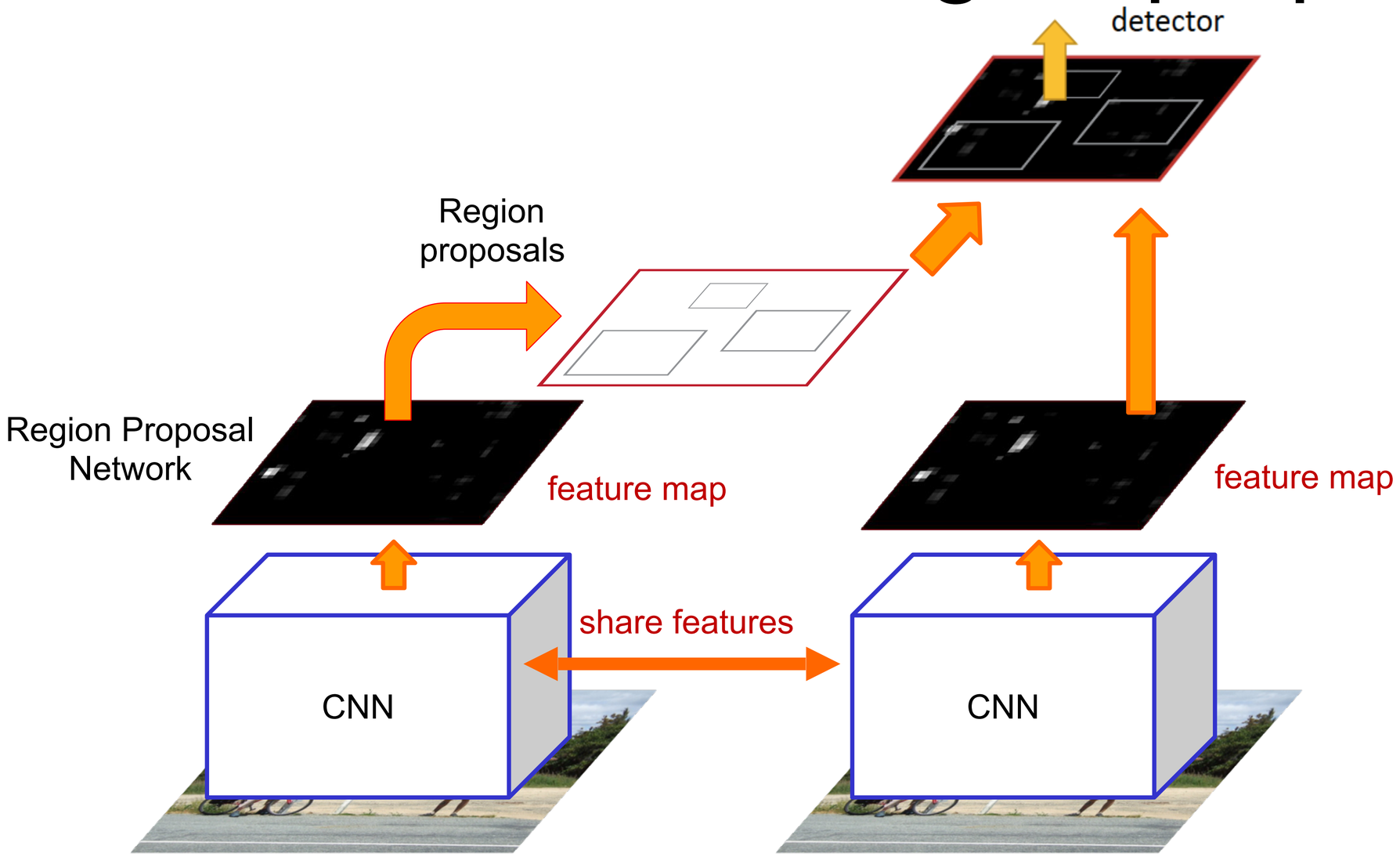
- How do we crop from a feature map?
- Step 4: Take max in each cell



See more here: https://deepsense.ai/region-of-interest-pooling-explained/

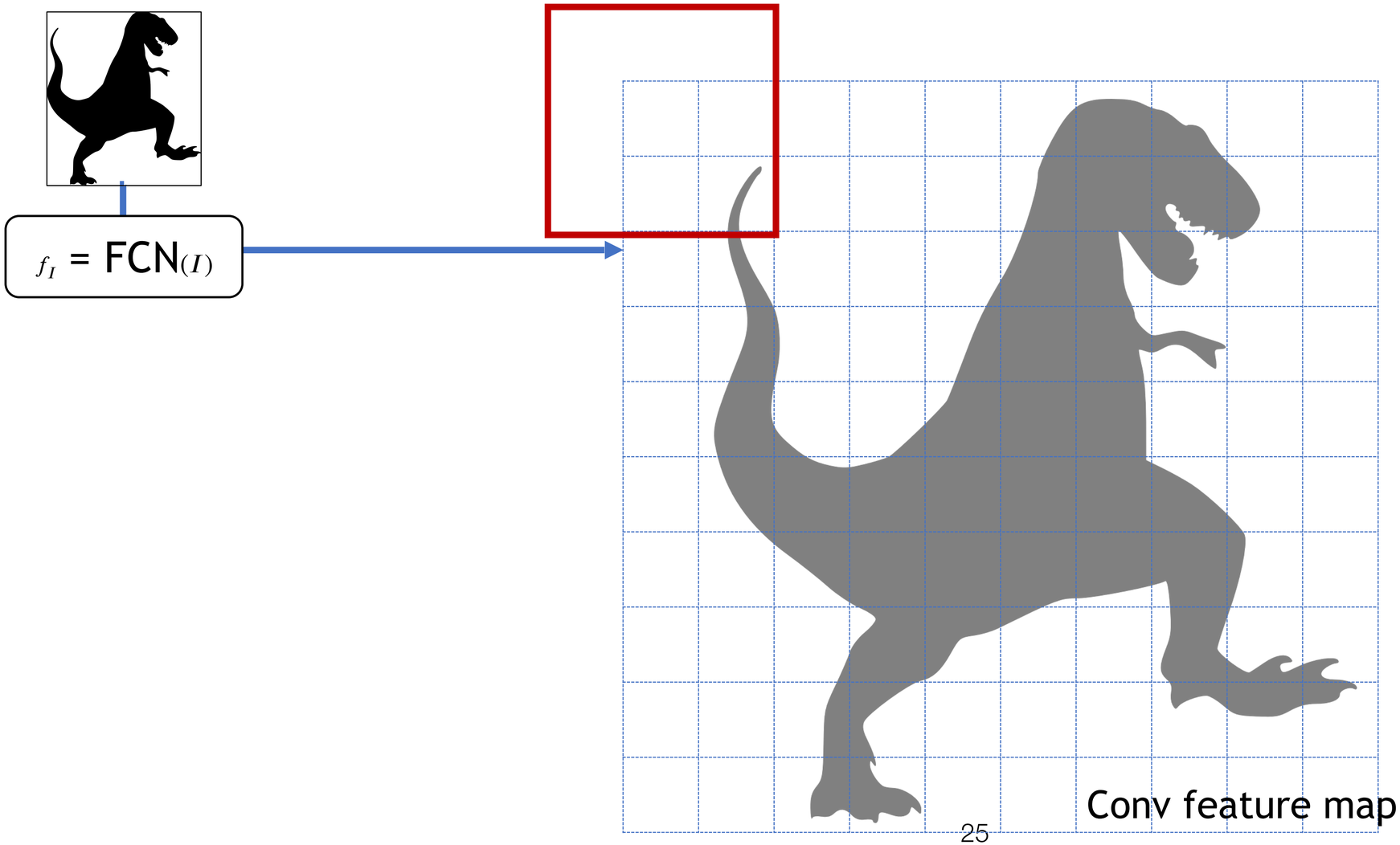
Source: B. Hariharan

“Faster” R-CNN: learn region proposals



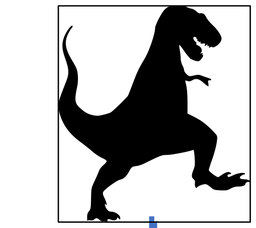
S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

RPN: Region Proposal Network

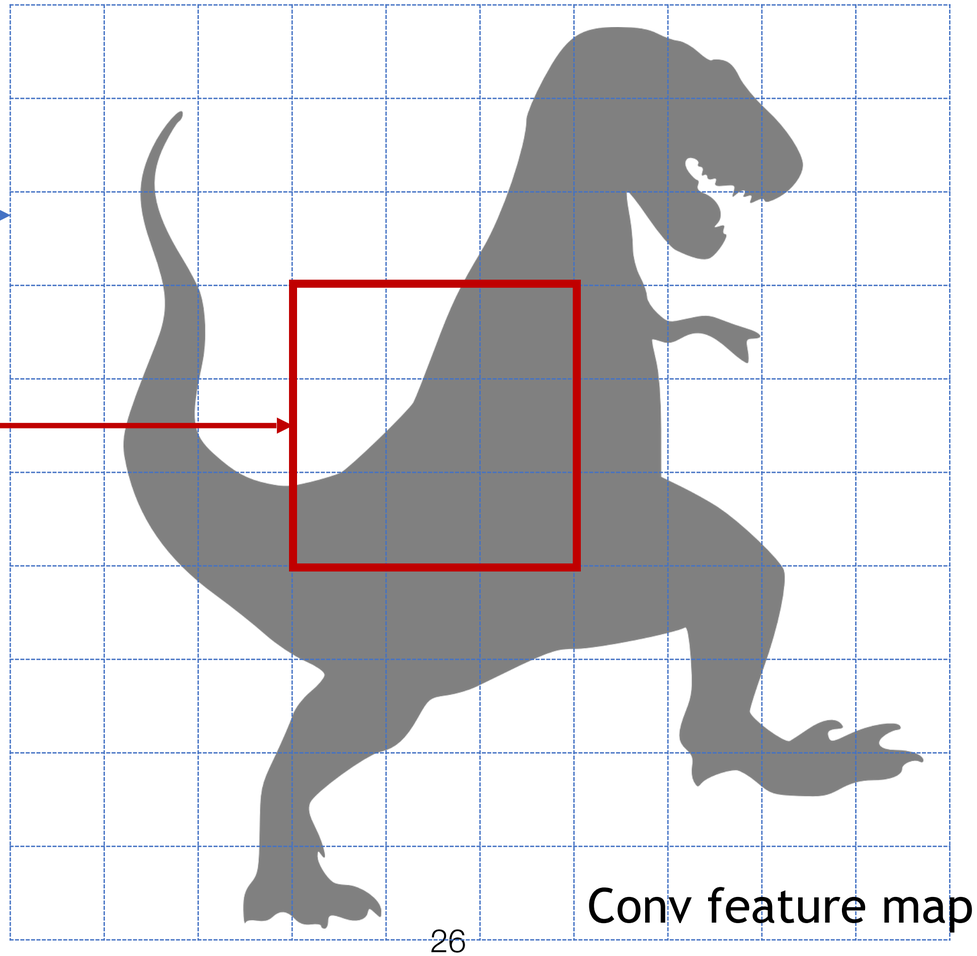


Source: R. Girshick

RPN: Region Proposal Network



f_I = FCN(I)



3x3 “sliding window”

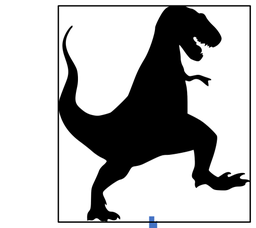
Scans the feature map looking for objects

Conv feature map

Source: R. Girshick

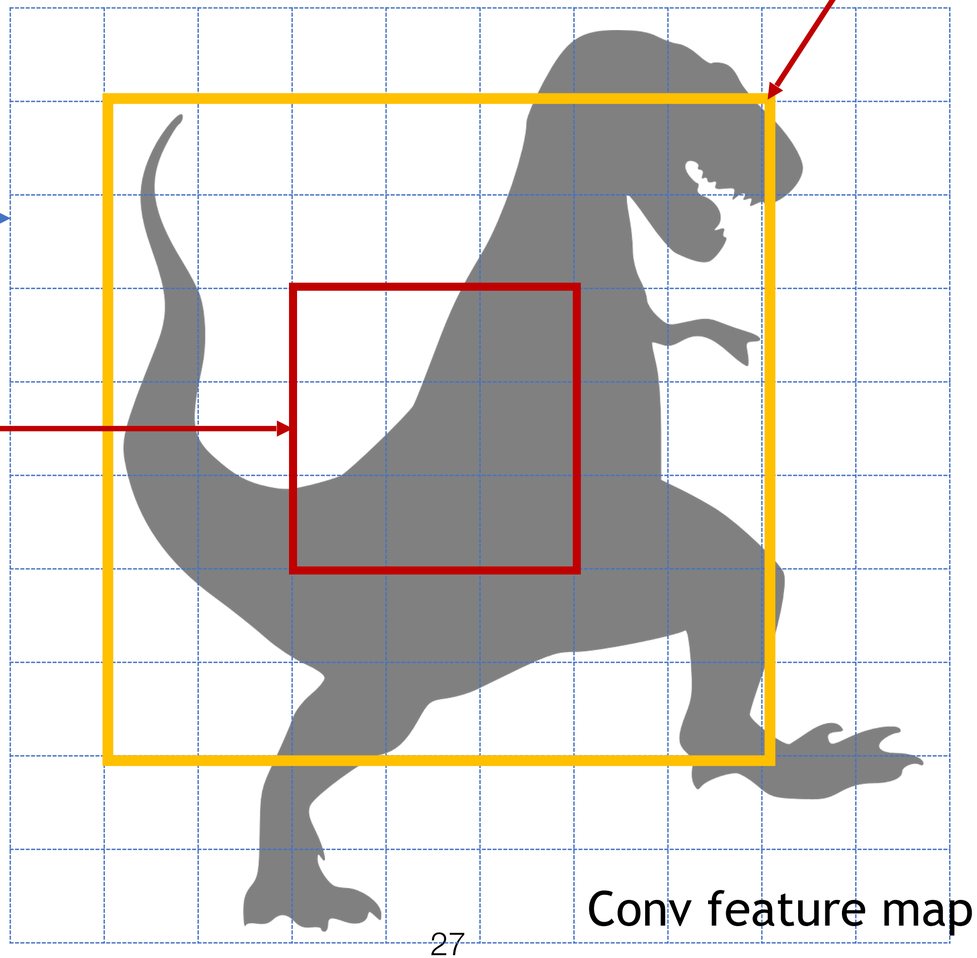
RPN: Anchor Box

Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*



f_I = FCN(I)

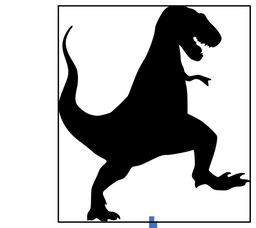
3x3 “sliding window”
Scans the feature map looking for objects



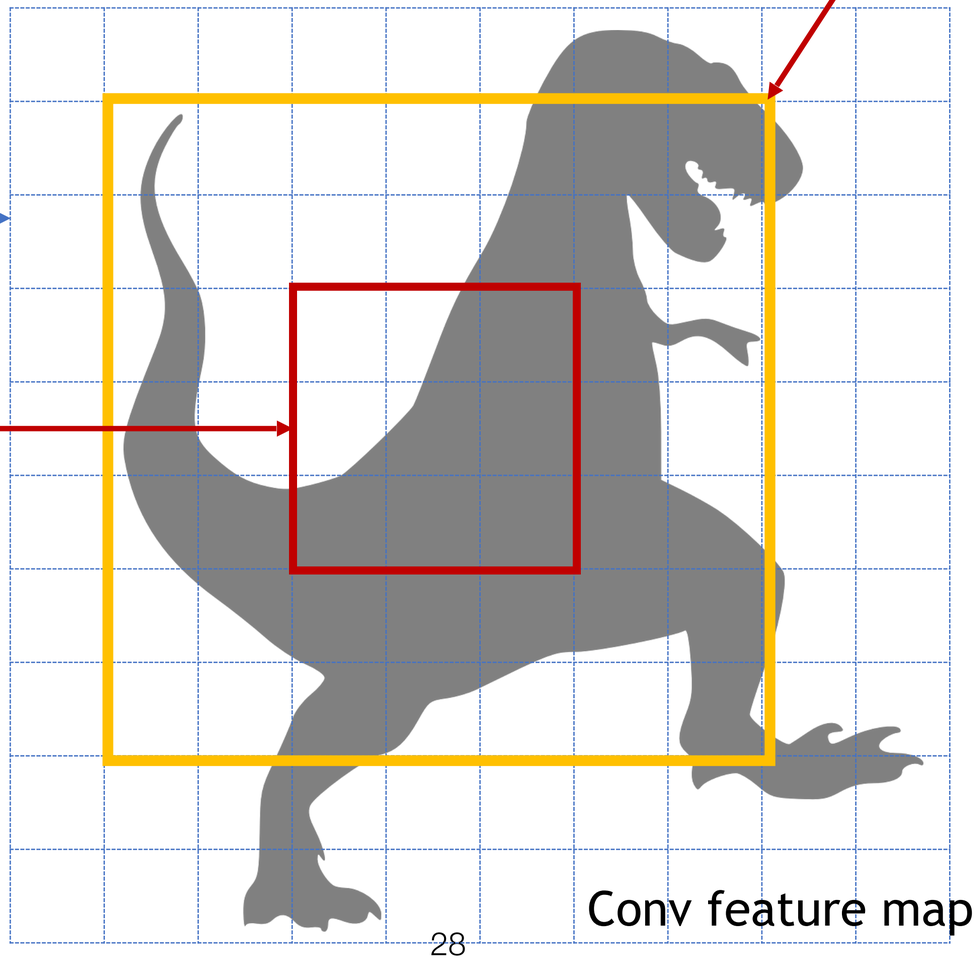
Conv feature map

Source: R. Girshick

RPN: Anchor Box



f_I = FCN(I)



Conv feature map

Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

3x3 “sliding window”

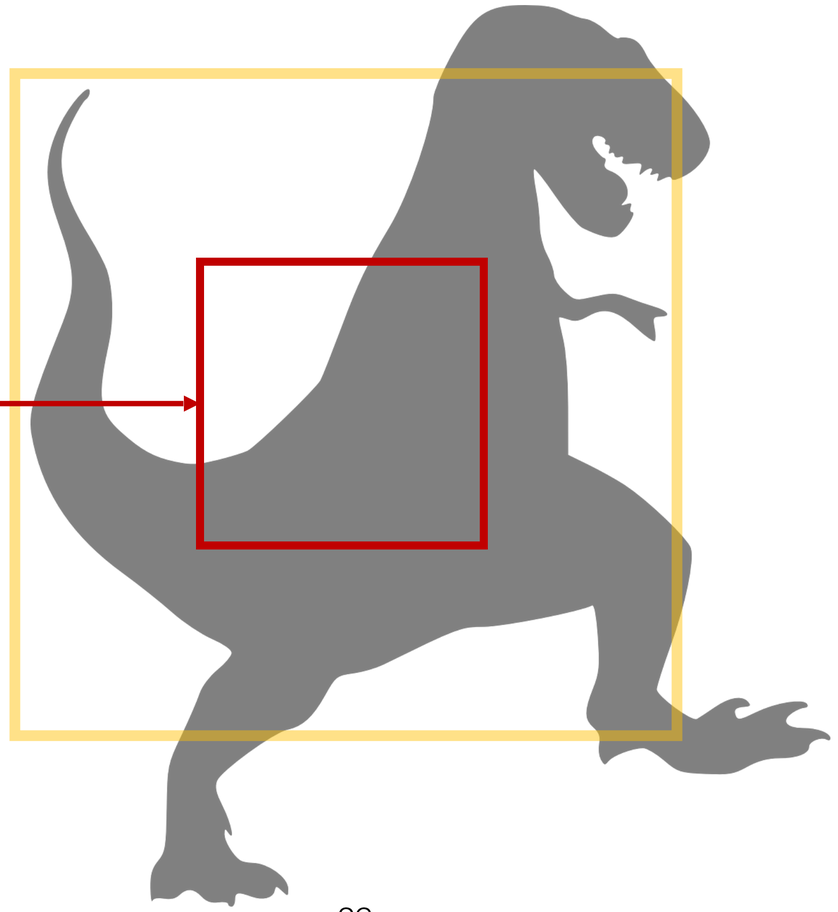
- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

Source: R. Girshick

RPN: Prediction (on object)

Objectness score

P(object) = 0.94



3x3 “sliding window”

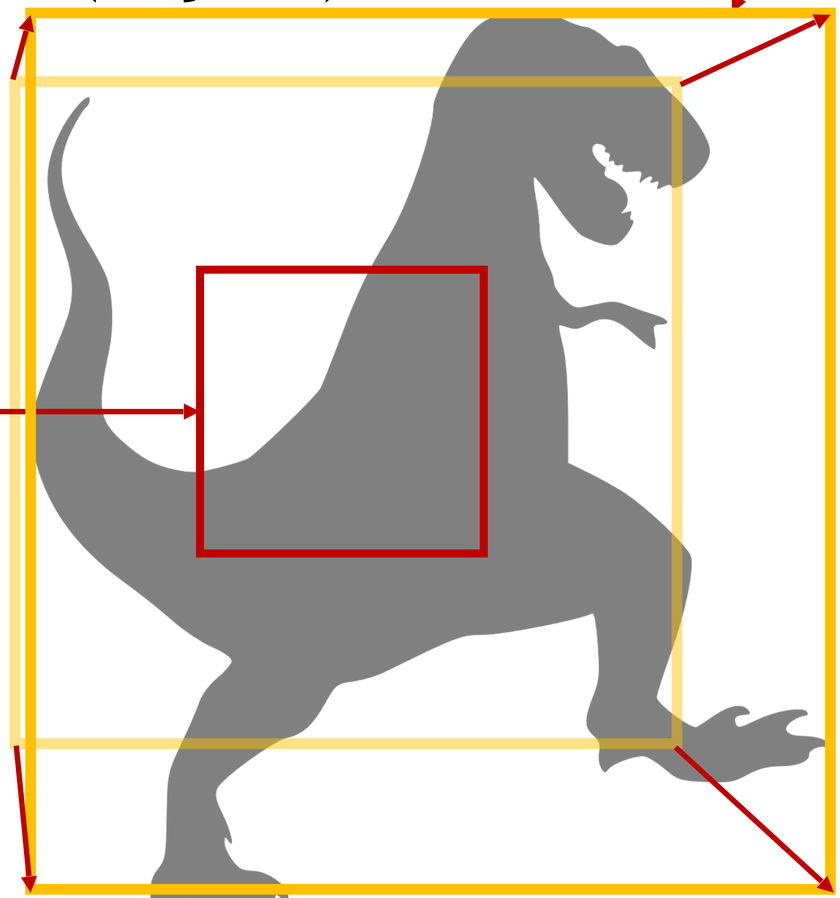
- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

Source: R. Girshick

RPN: Prediction (on object)

Anchor box: transformed by box regressor

P(object) = 0.94



3x3 “sliding window”

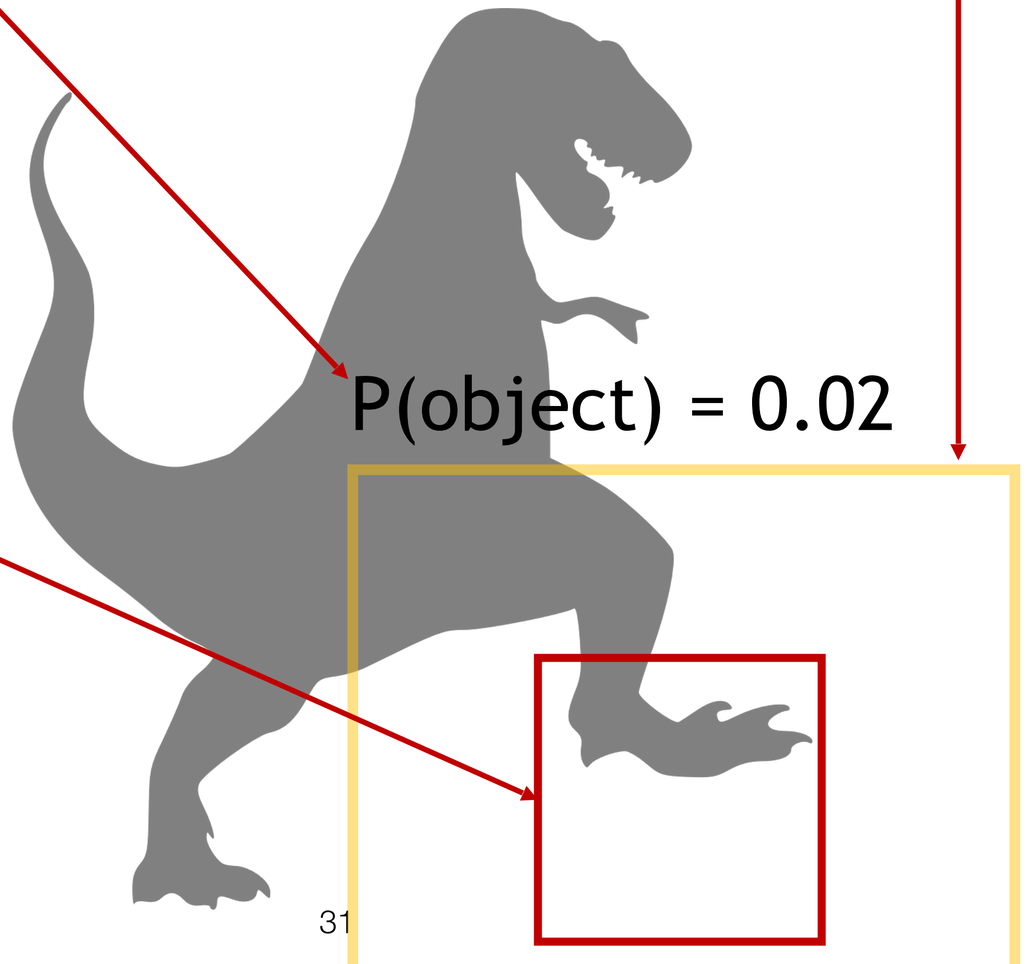
- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

Source: R. Girshick

RPN: Prediction (off object)

Anchor box: transformed by box regressor

Objectness score

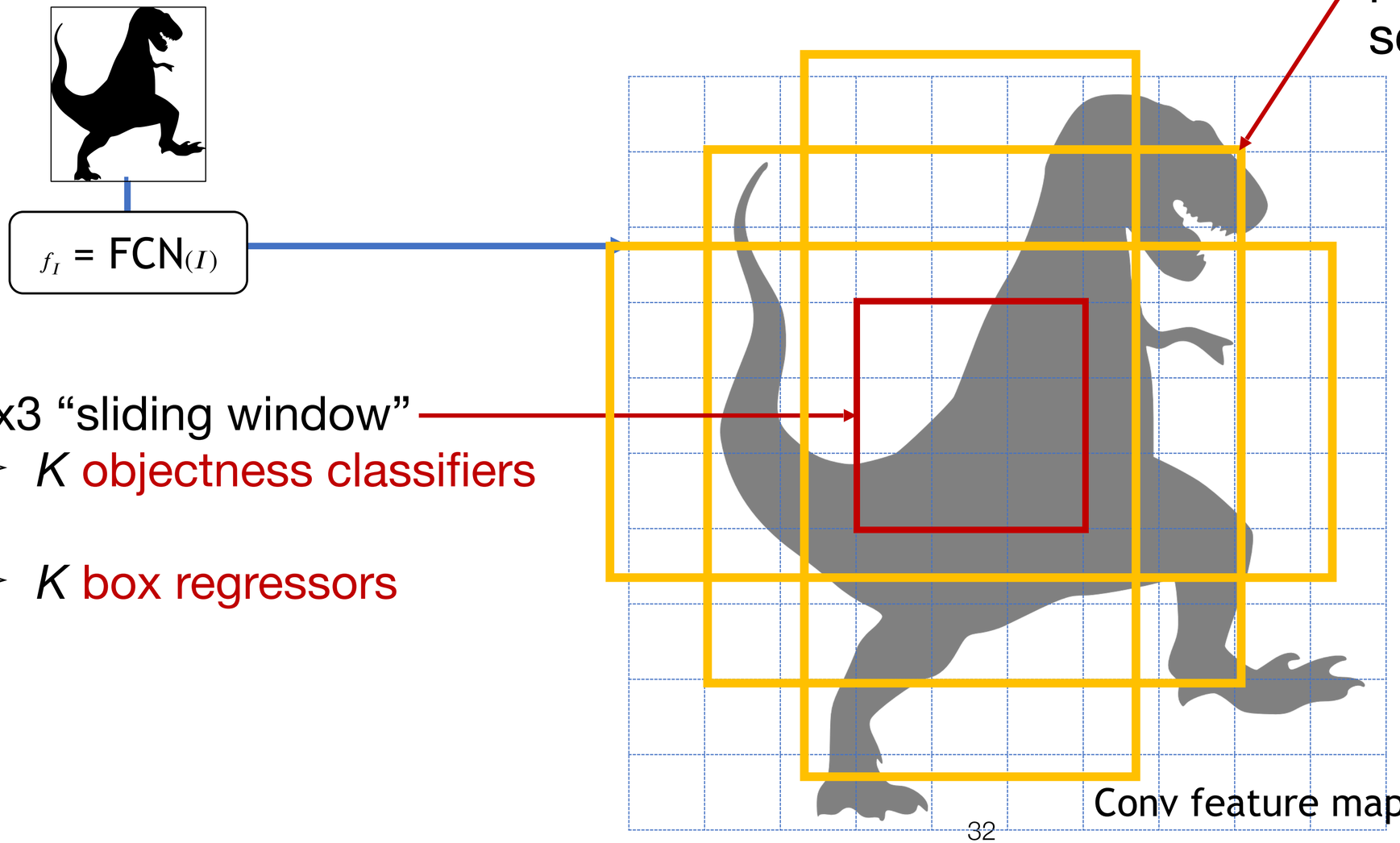


3x3 “sliding window”

- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)

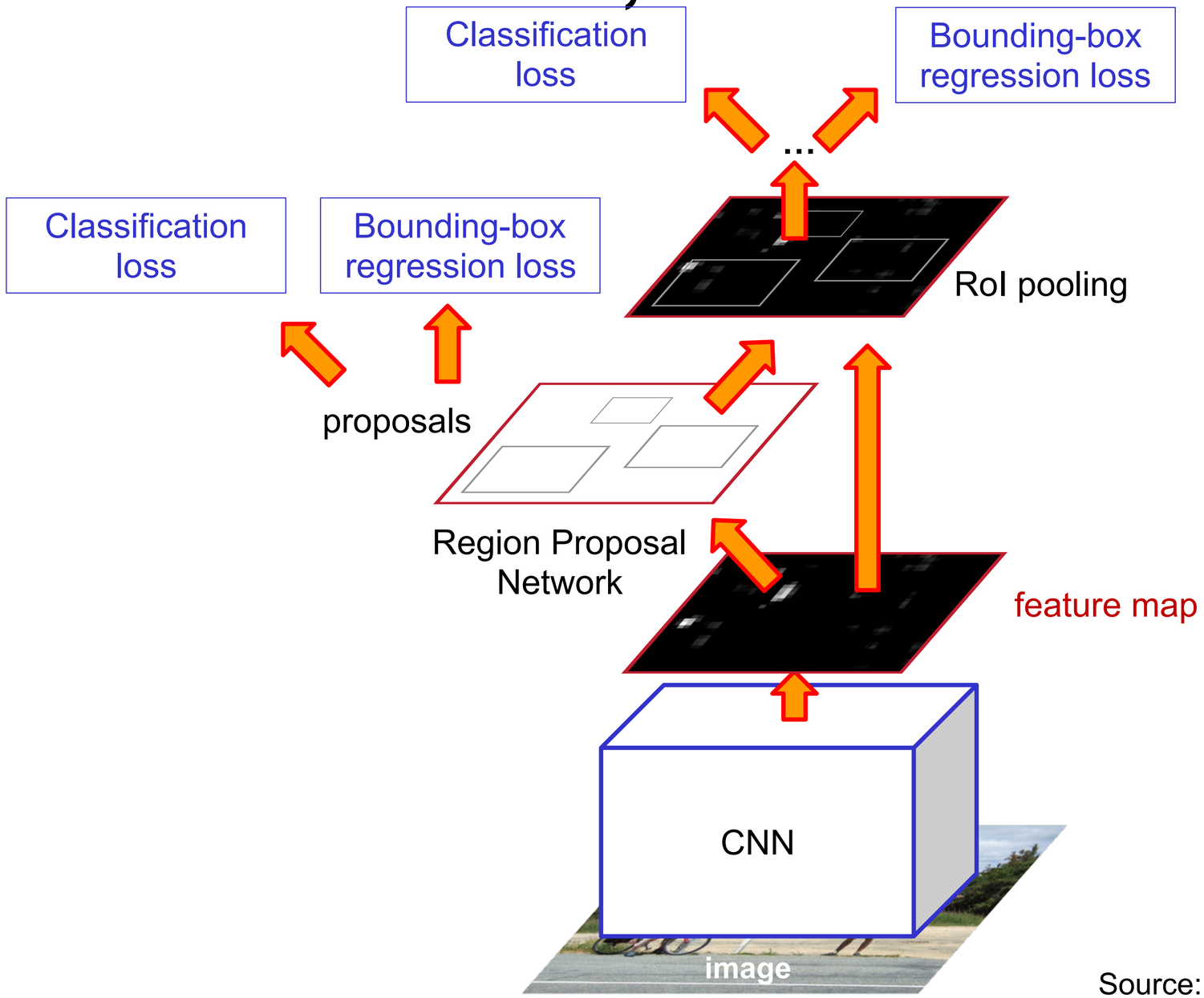
Source: R. Girshick

RPN: Multiple Anchors



Source: R. Girshick

One network, four losses



Source: R. Girshick, K. He, S. Lazebnik

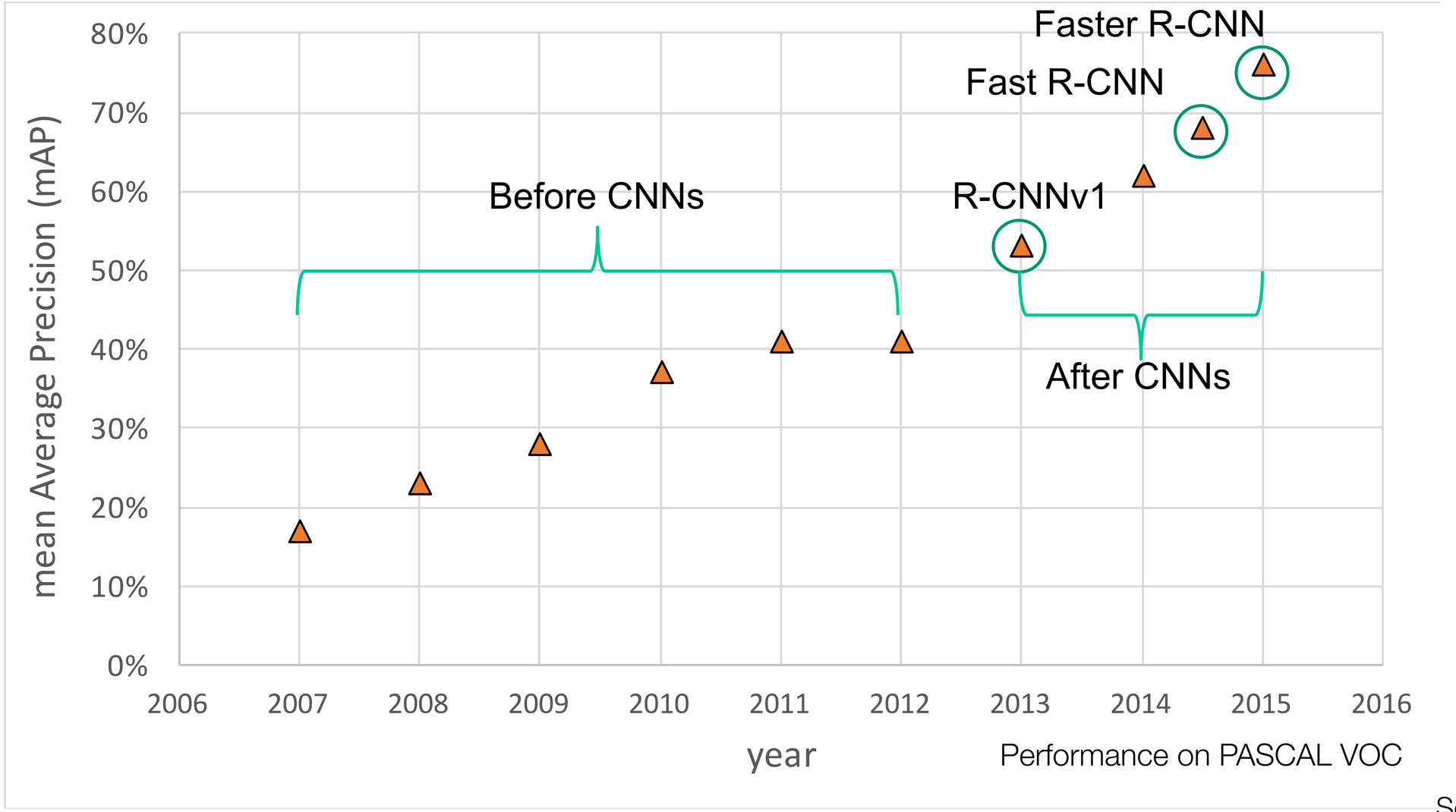
Faster R-CNN results

system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	**69.9**	**73.2**

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

Source: S. Lazebnik

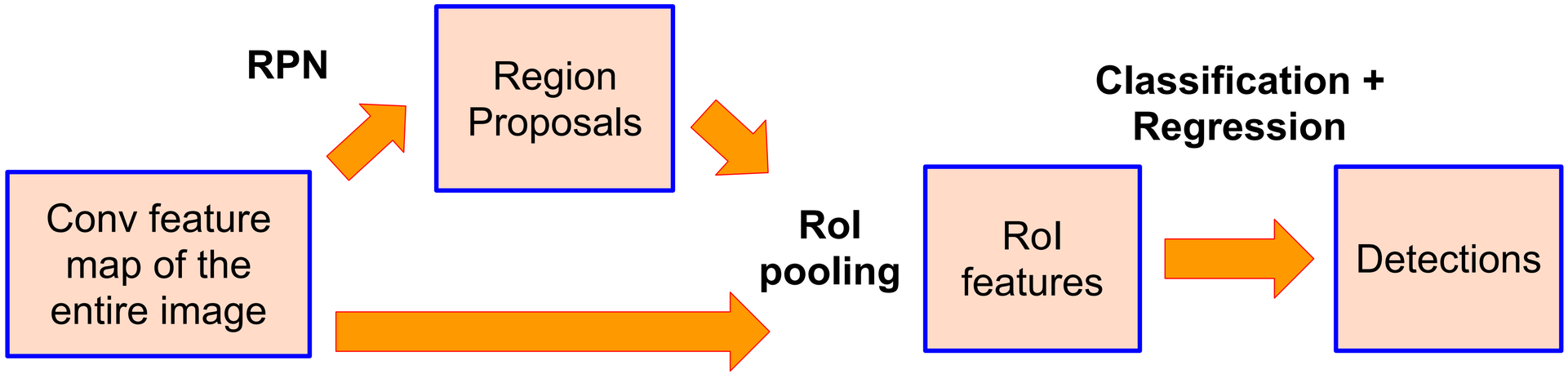
Object detection progress



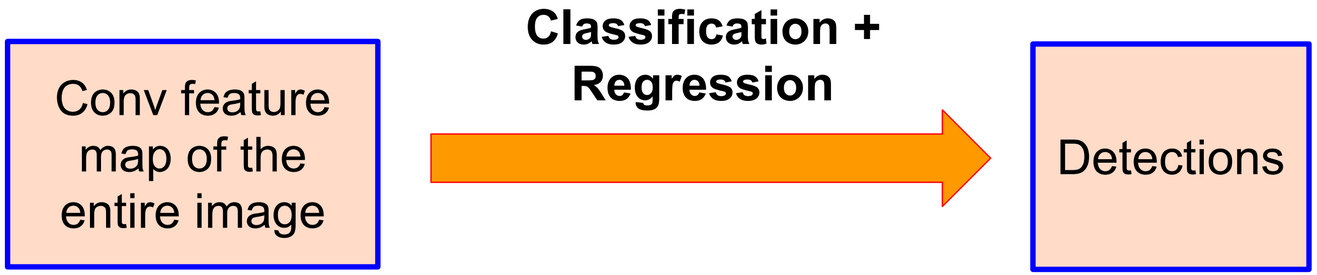
Source: S. Lazebnik

Streamlined detection architectures

- The Faster R-CNN pipeline separates proposal generation and region classification:



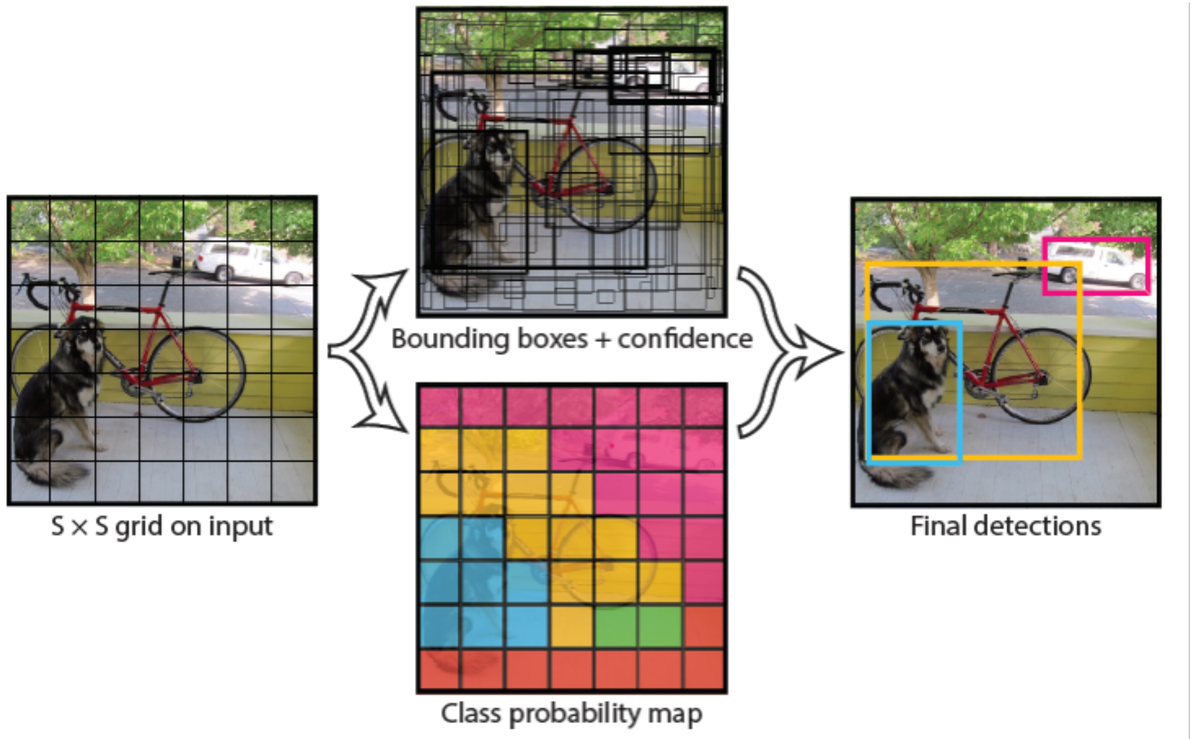
- Is it possible do detection in one shot?



Source: S. Lazebnik

Single-stage object detector

- Divide the image into a coarse grid and directly predict class label and a few candidate boxes for each grid cell



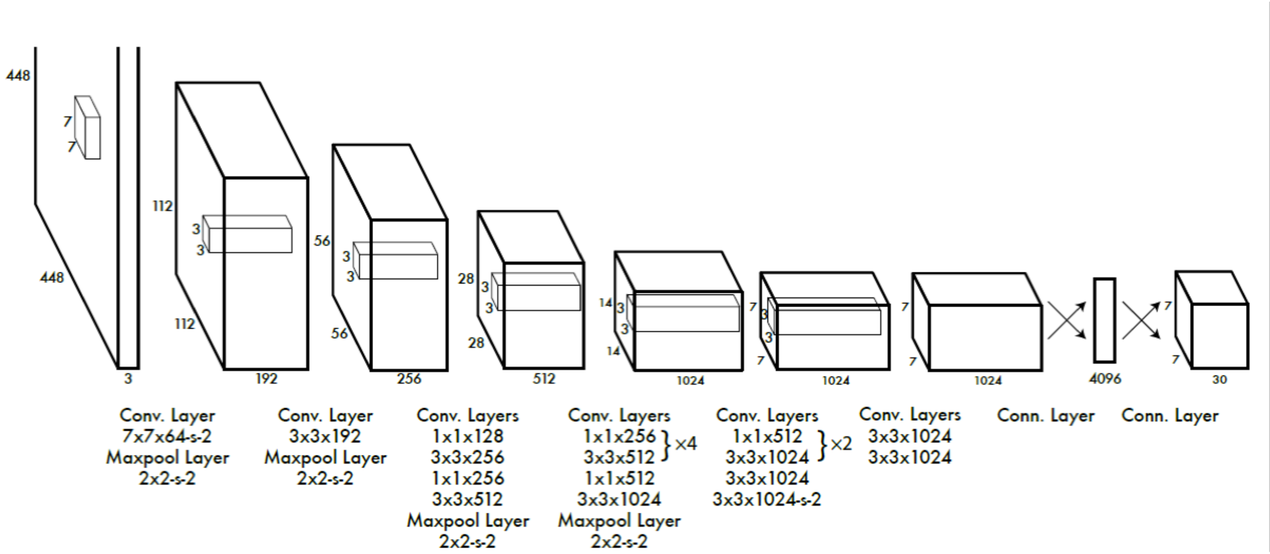
J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016

Source: S. Lazebnik

YOLO detector

1. Take conv feature maps at 7x7 resolution
2. Predict, at each location, a score for each class and 2 bboxes w/ confidences
   - For PASCAL, output is 7x7x30 (30 = 20 + 2*(4+1))

- 7x speedup over Faster R-CNN (45-155 FPS vs. 7-18 FPS)
  but less accurate (e.g. 65% vs. 72 mAP%)



J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016

Source: S. Lazebnik

Challenges in object detection

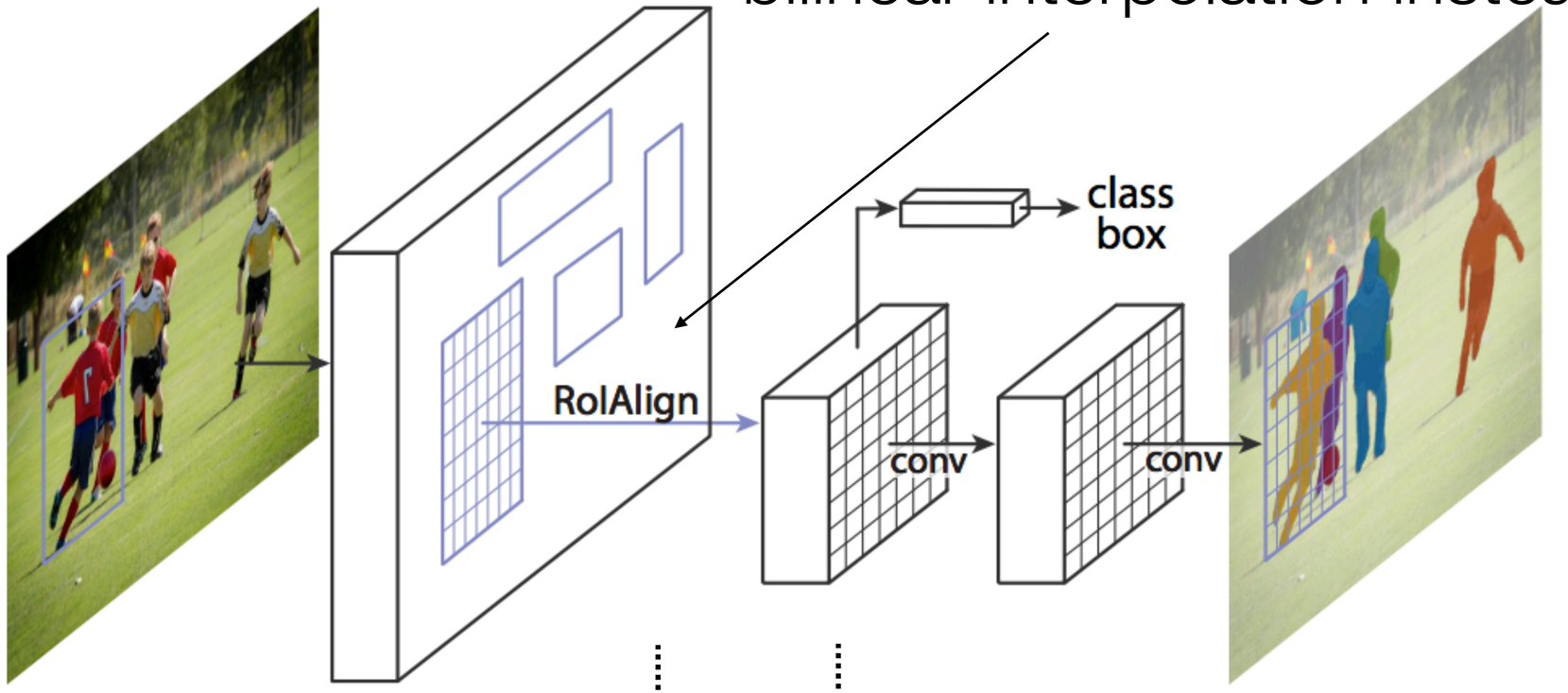
Beyond bounding boxes: instance segmentation

Predict segmentation mask for each object
From COCO [Lin et al., 2014]

Source: B. Hariharan

Instance segmentation

ROI pooling with tiny change:
bilinear interpolation instead of max



Faster R-CNN

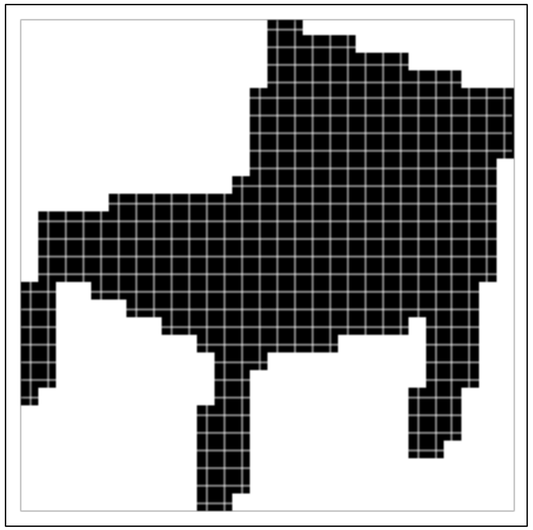
Extra “head” on network
predicts binary mask

[He et al., “Mask R-CNN”, 2017]

Example Mask Training Targets

Image with training proposal

28x28 mask target

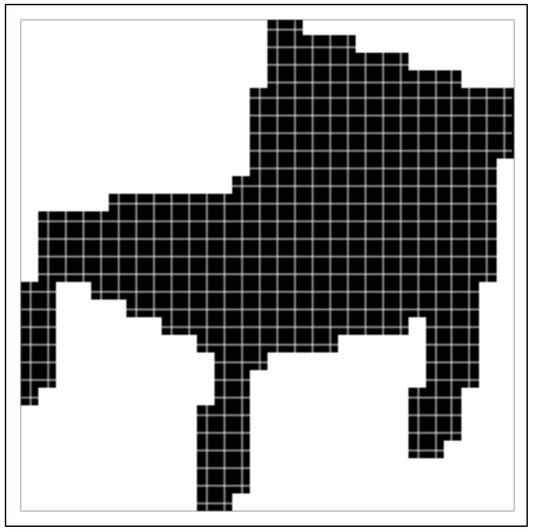


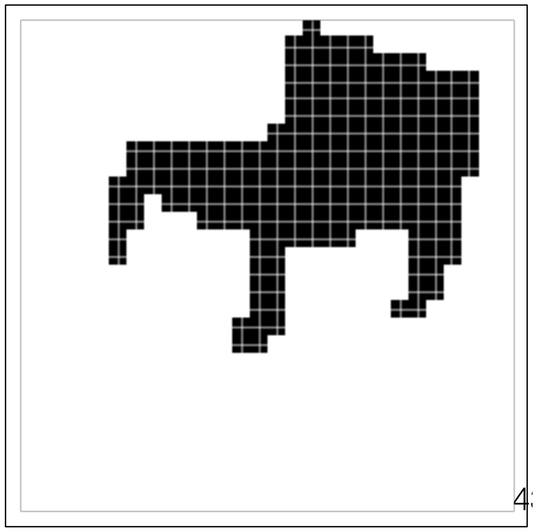
Source: R. Girshick

Example Mask Training Targets

Image with training proposal

28x28 mask target





Source: R. Girshick

Example Mask Training Targets

Image with training proposal

28x28 mask target

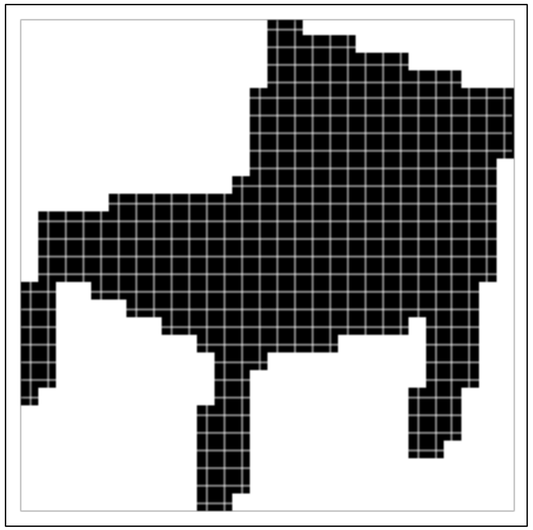
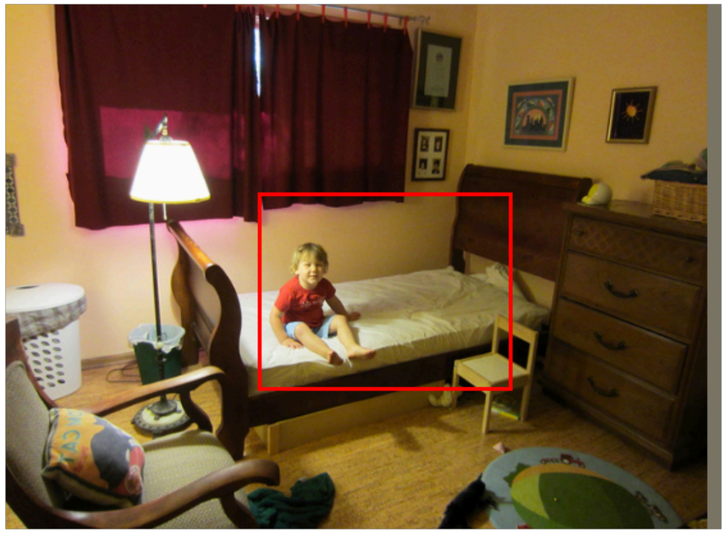
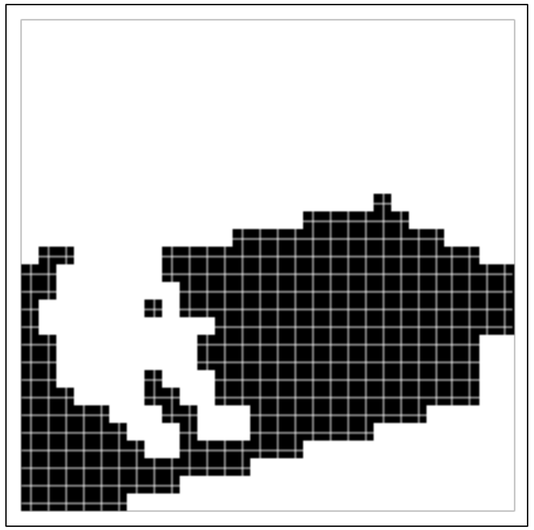
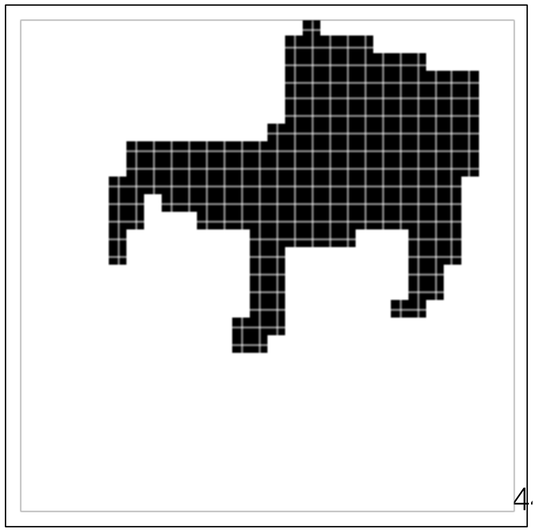


Image with training proposal



28x28 mask target





Source: R. Girshick

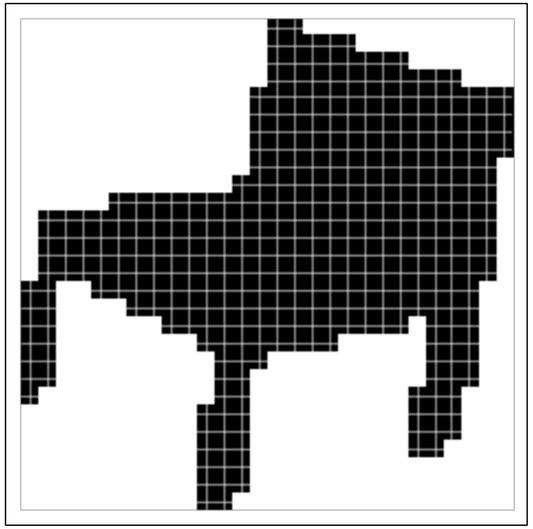
Example Mask Training Targets

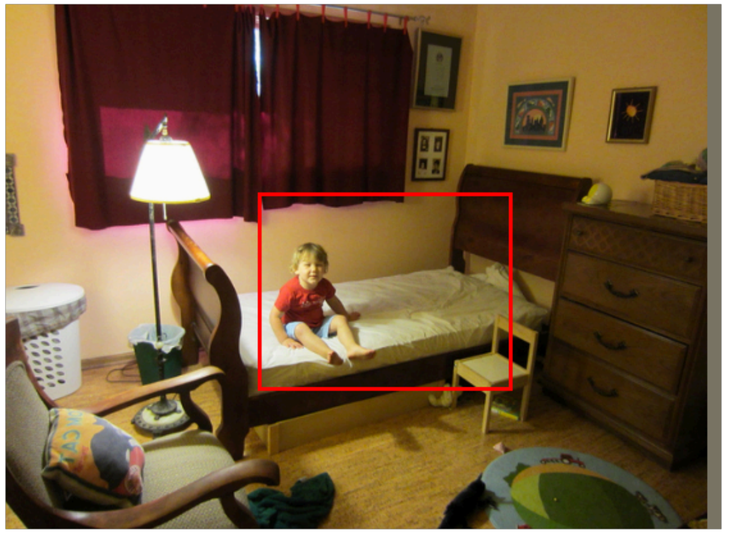
Image with training proposal

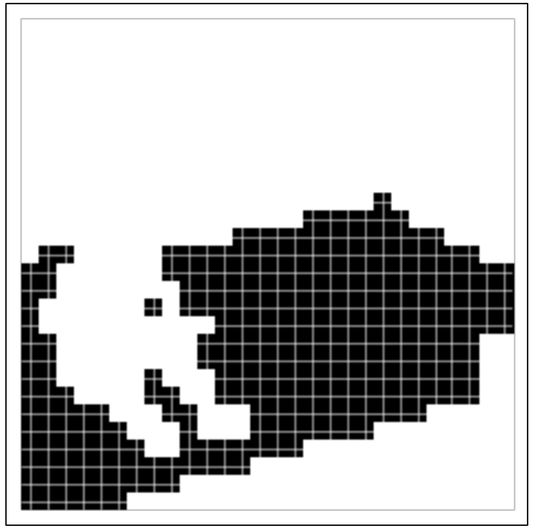
28x28 mask target

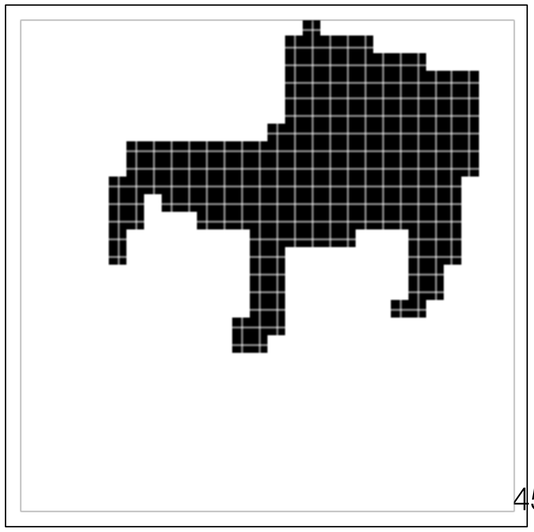
Image with training proposal

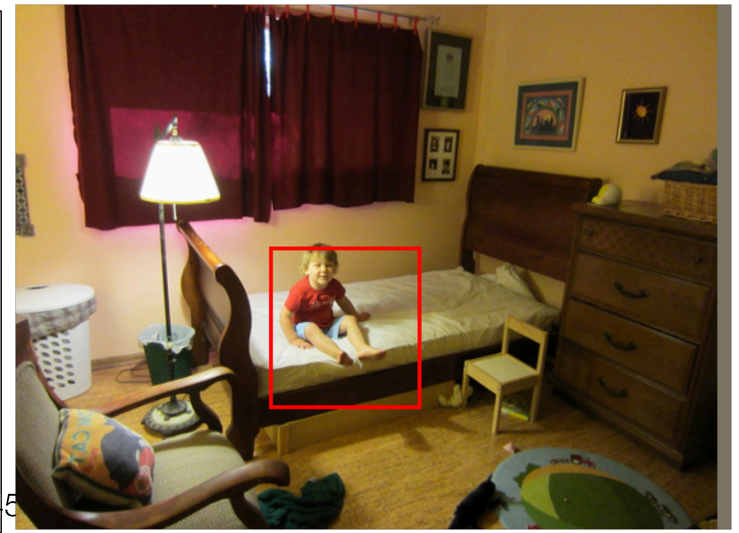
28x28 mask target

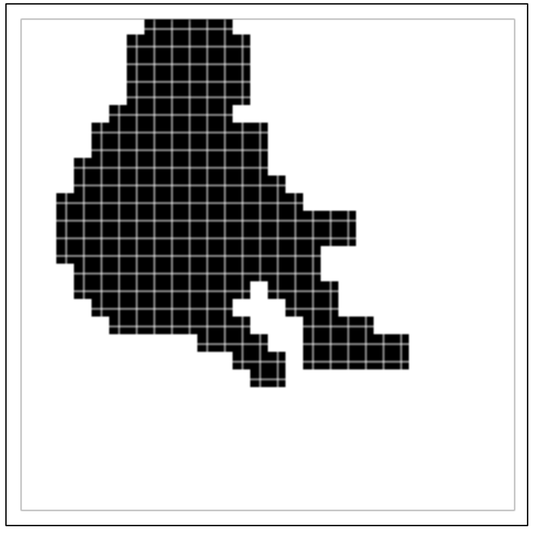


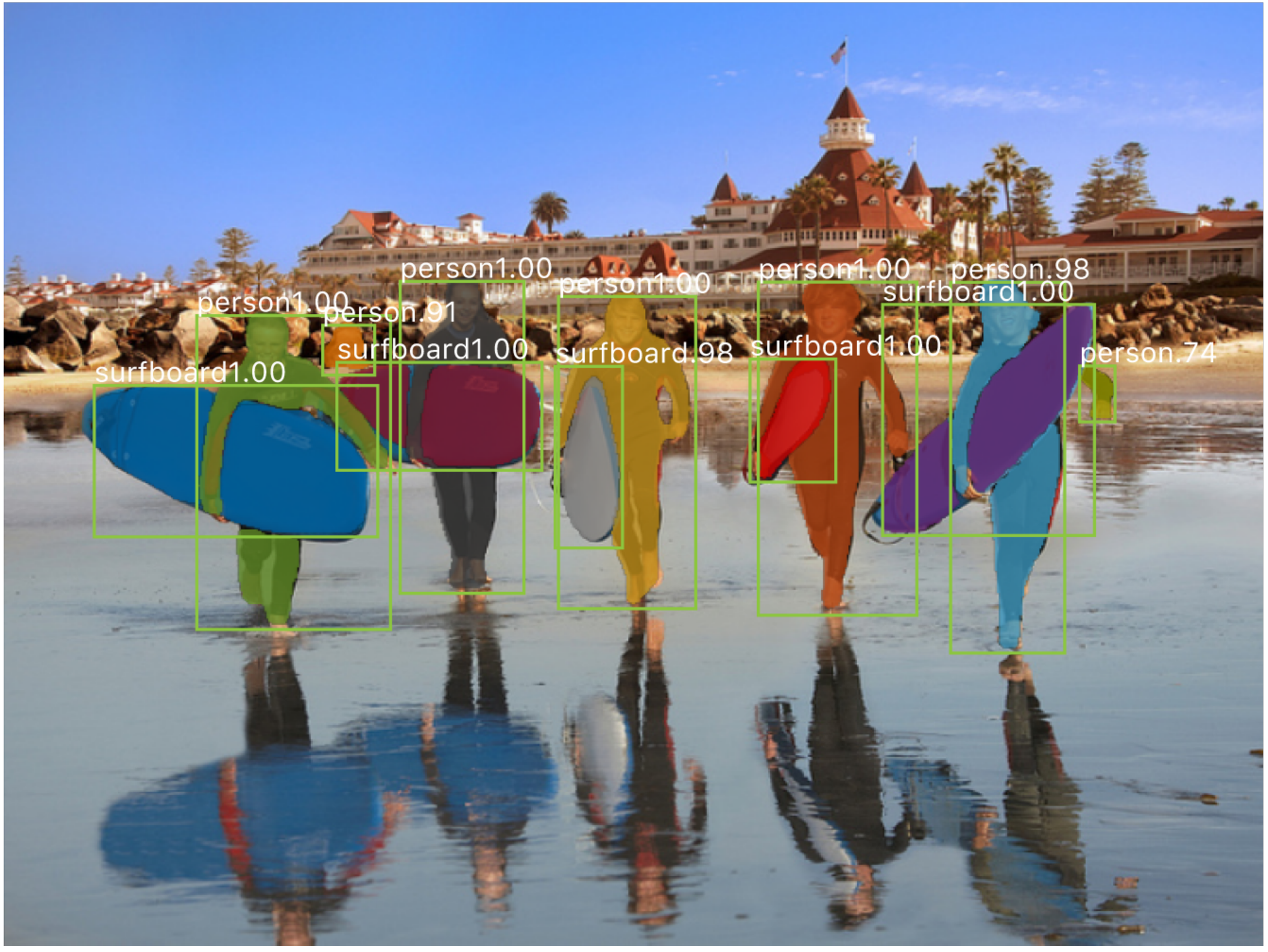


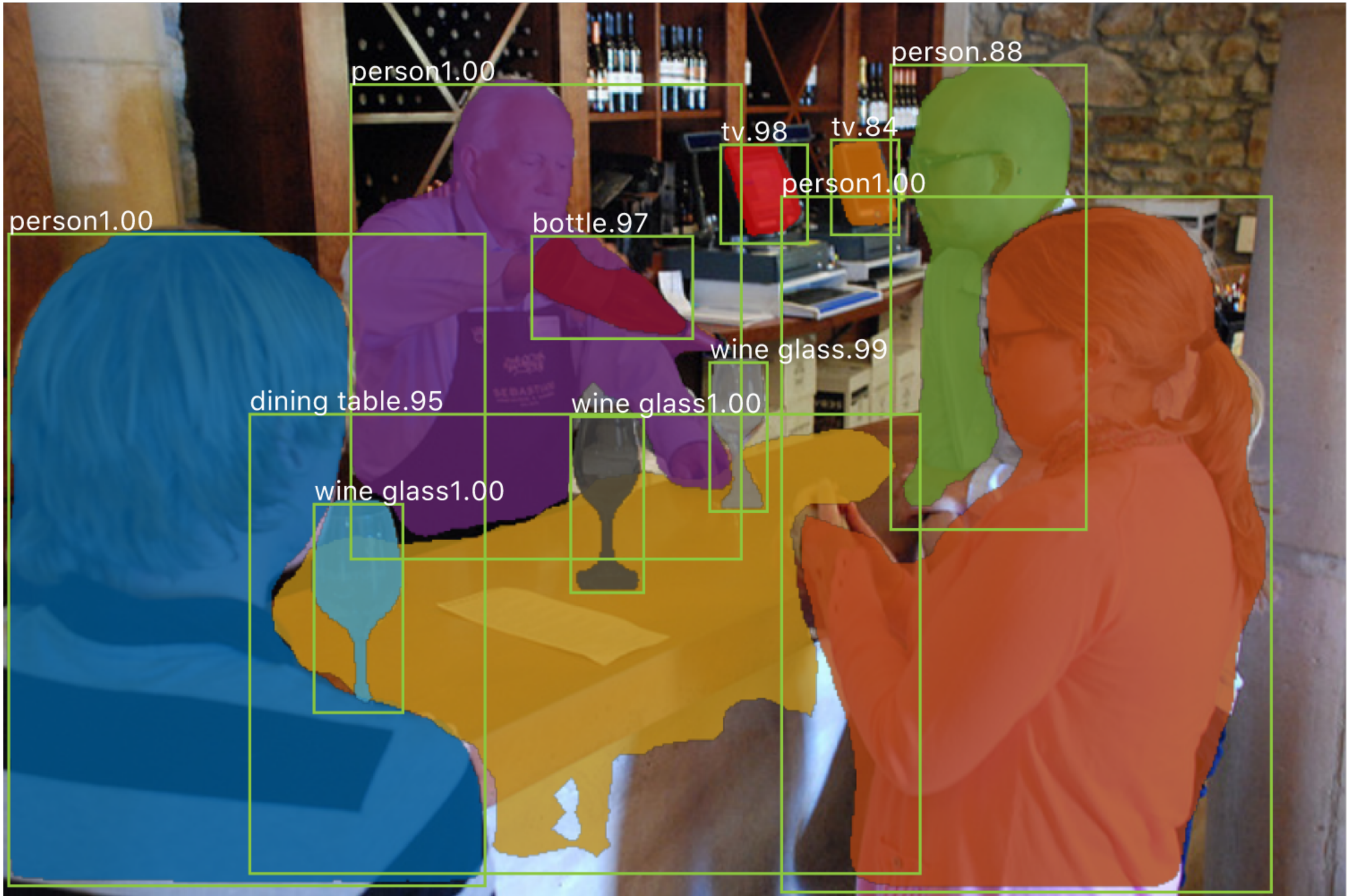




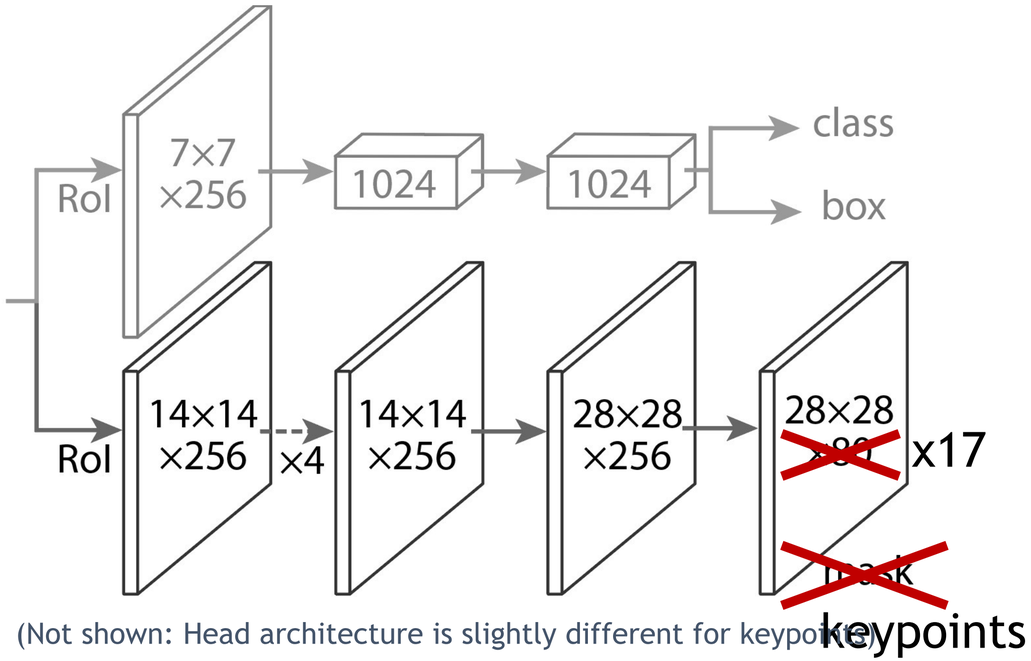




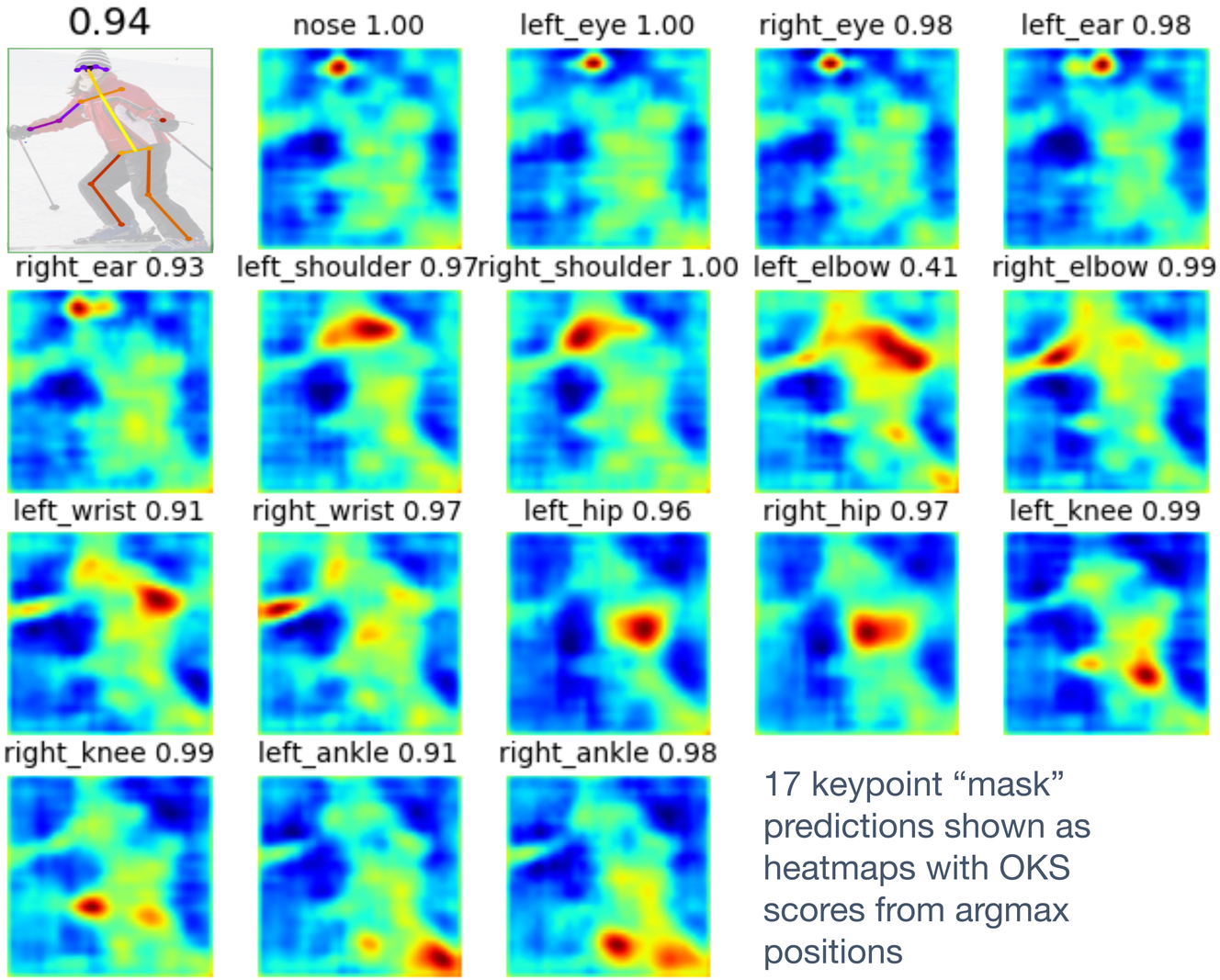

person1.00
person1.00
person1.00
person.98
surfboard1.00
person1.00
person.91
surfboard1.00
surfboard.98
surfboard1.00
person.74
surfboard1.00


person.88
person1.00
tv.98
tv.84
person1.00
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00

Human Pose



(Not shown: Head architecture is slightly different for keypoints)



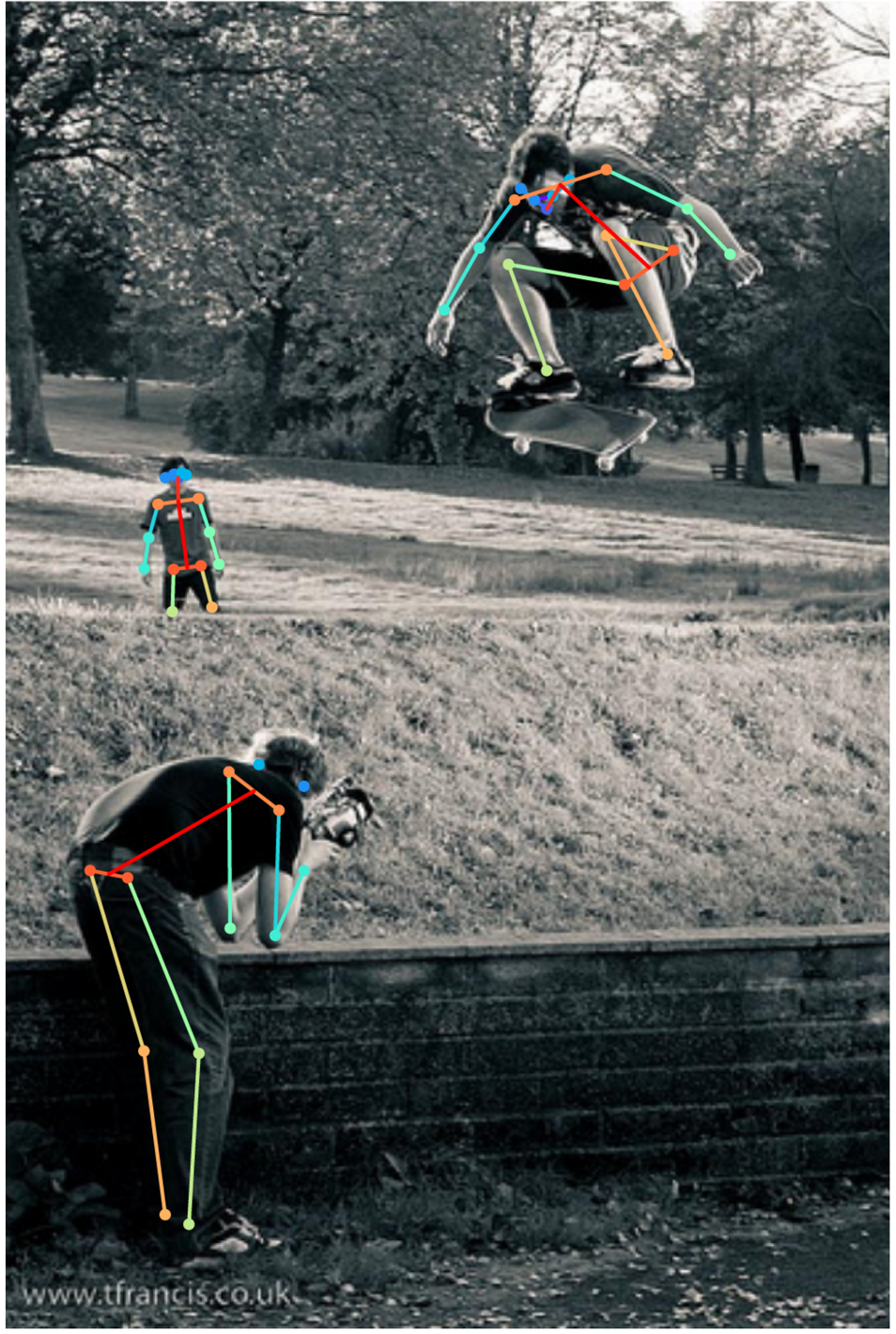
17 keypoint “mask” predictions shown as heatmaps with OKS scores from argmax positions

➢ Add keypoint head (28x28x17)

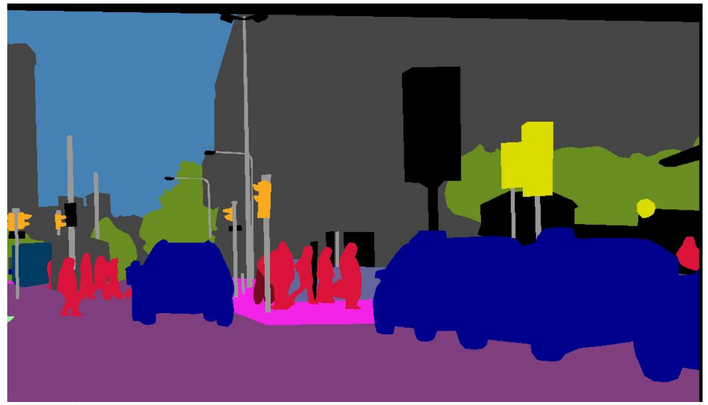
➢ Predict one “mask” for each keypoint

➢ Softmax over spatial locations (encodes one keypoint per mask “prior”)

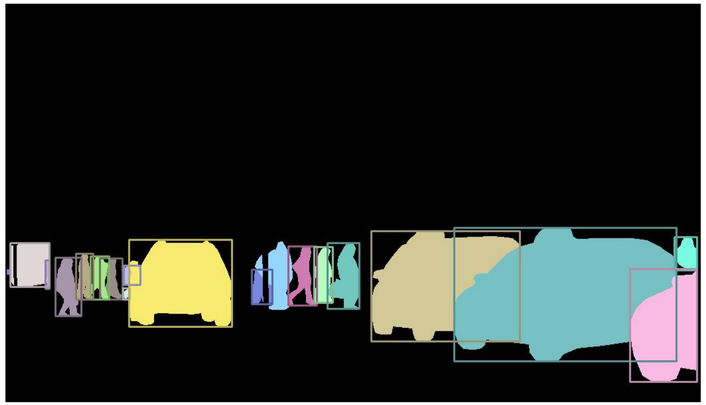
Source: R. Girshick


www.tfrancis.co.uk

Panoptic Segmentation



Semantic segmentation “stuff”



Instance detection, “things”

panoptic segmentation [Kriilov et al. 2018]
predict label + instance id per pixel

Source: ?


UPSNet-101-M: Cityscapes