Object Detection

Xiaolong Wang

This Class: Object Detection

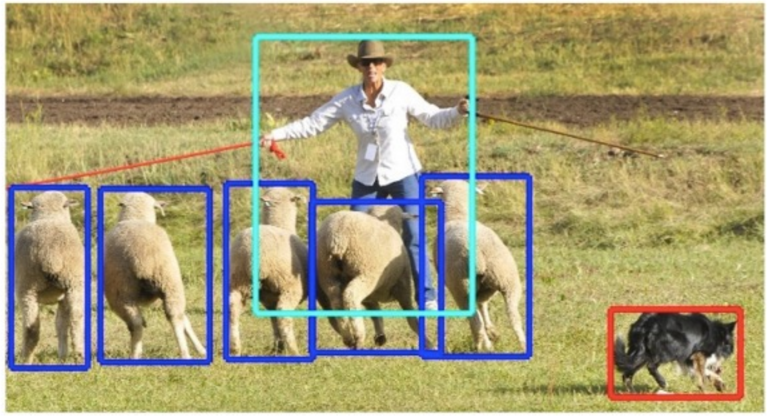
- Background and old fashion object detection
- 2-stage object detection
- FPN

Slides partially from: http://cs231n.stanford.edu/ https://slazebni.cs.illinois.edu/fall20/

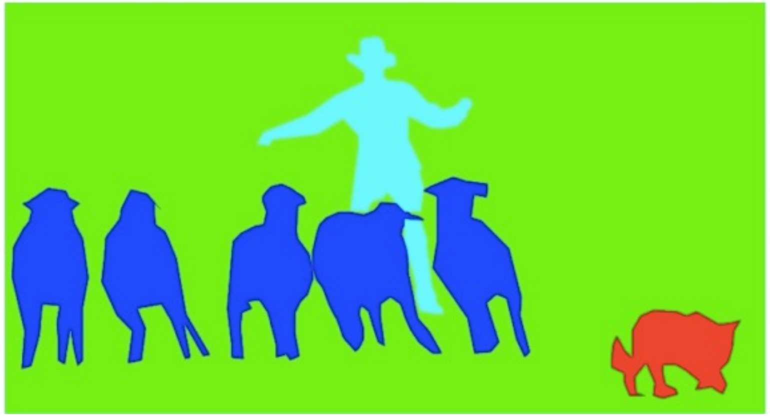
Background and old fashion object detection

The task: Object Detection

image classification



object detection

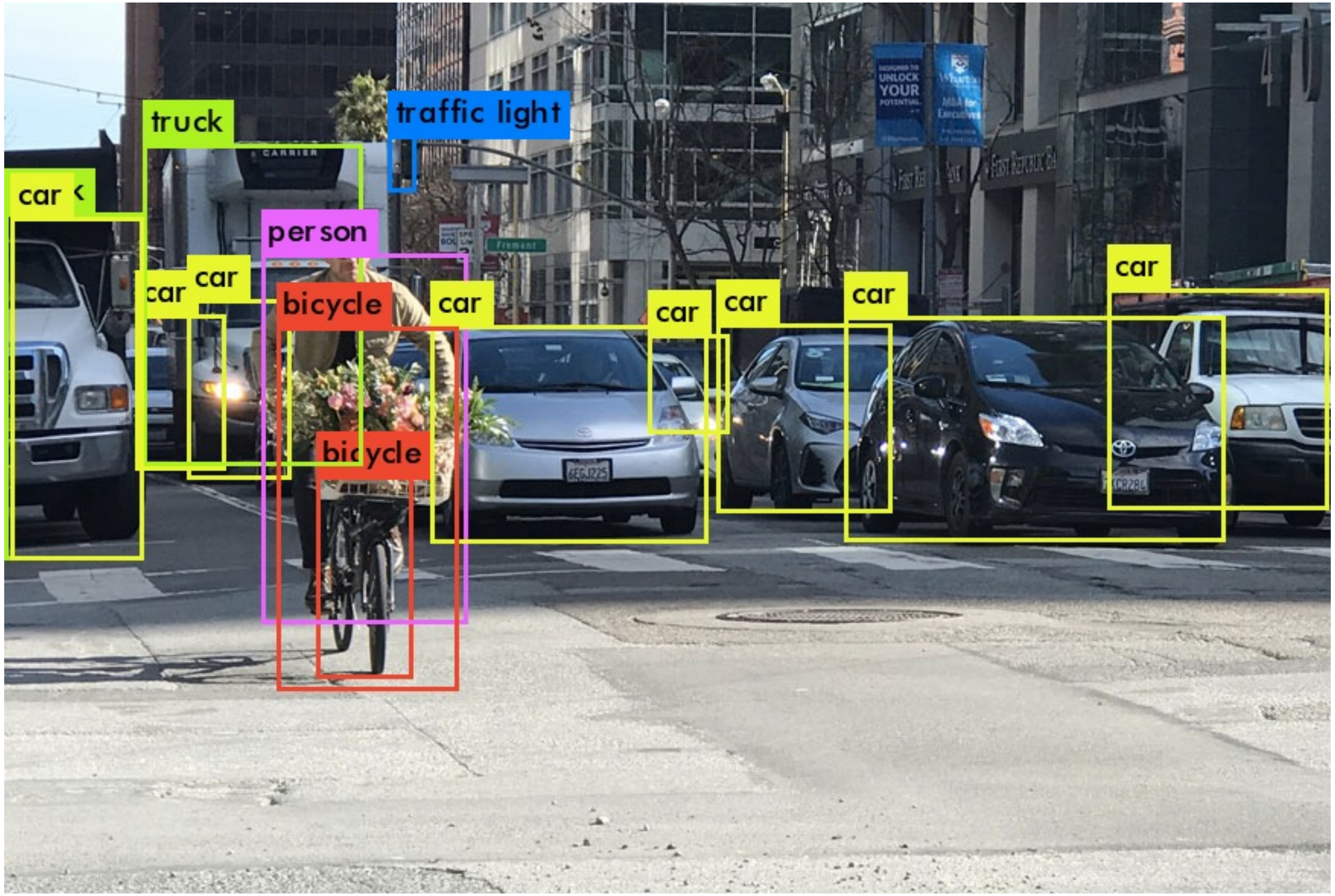


semantic segmentation

instance segmentation

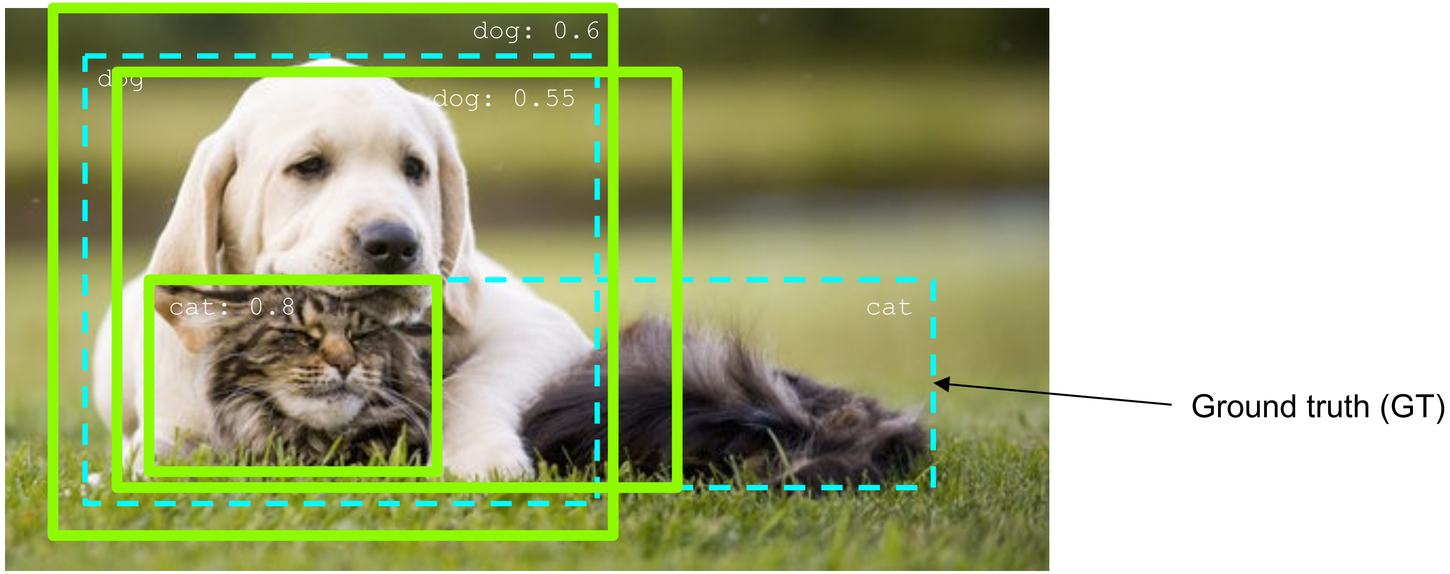
The task: Object Detection

Images may contain more than one class, multiple instances from the same class

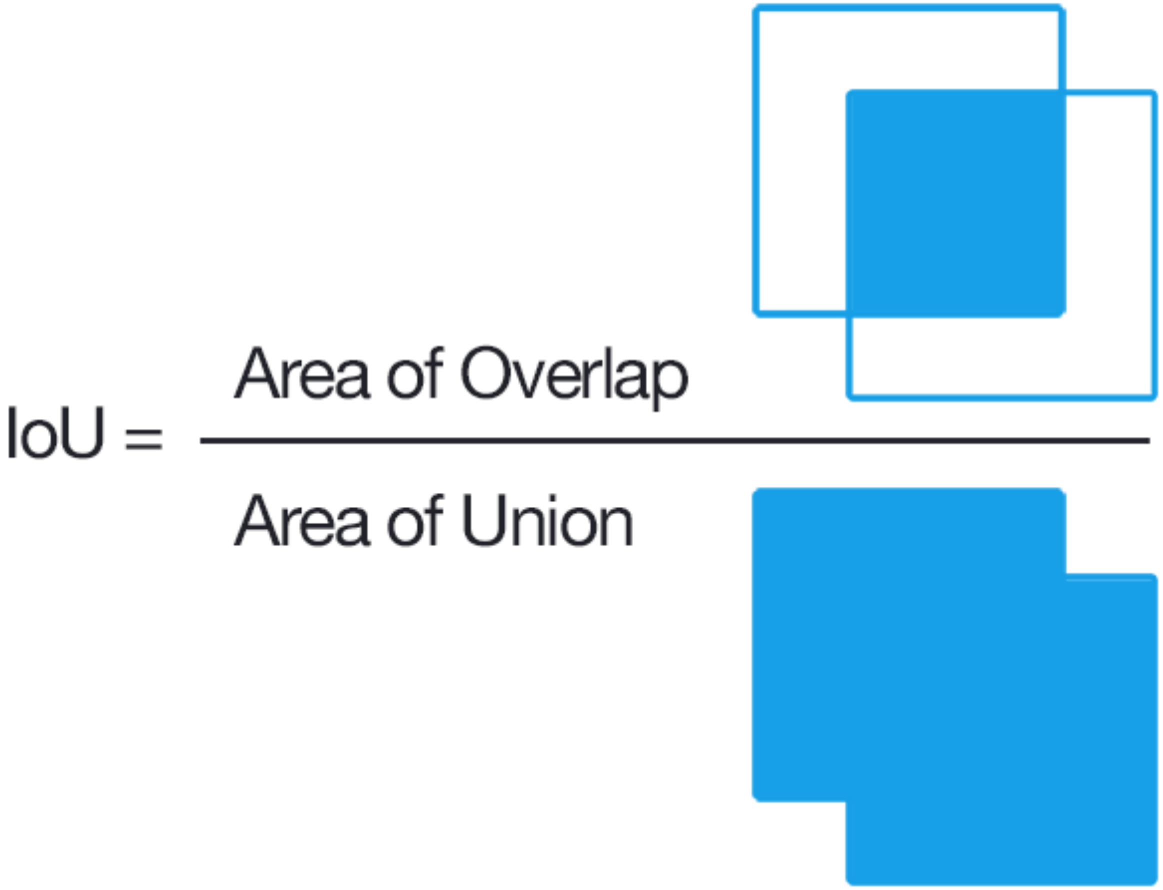


Evaluation

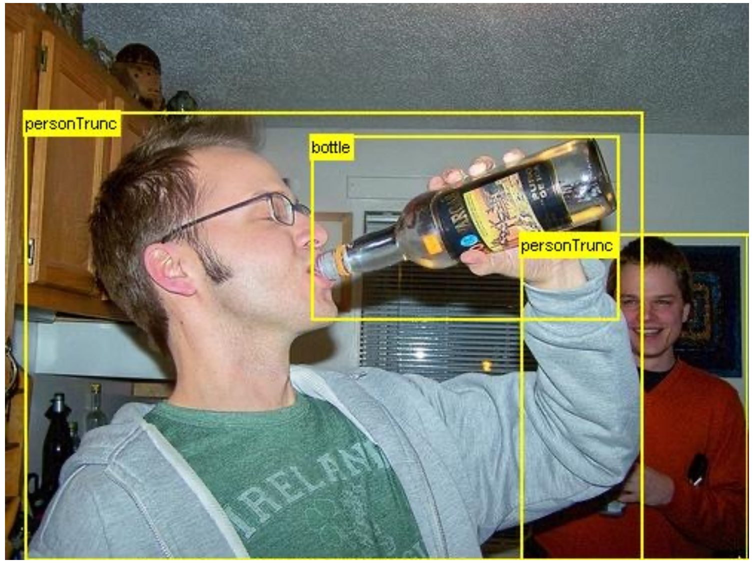
- At test time, predict bounding boxes, class labels, and confidence
- For each detection, determine whether it is a true or false positive
  - PASCAL criterion: Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5
  - For multiple detections of the same ground truth box, only one is considered a true positive

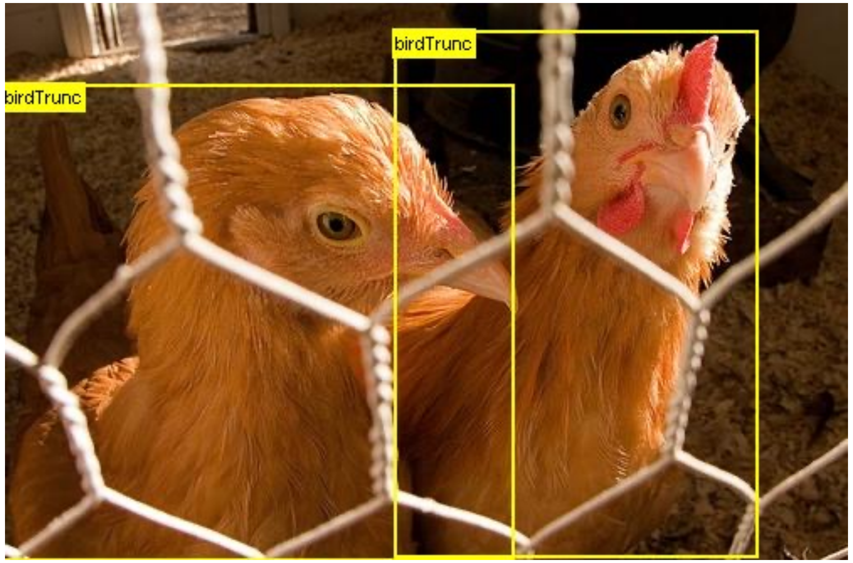


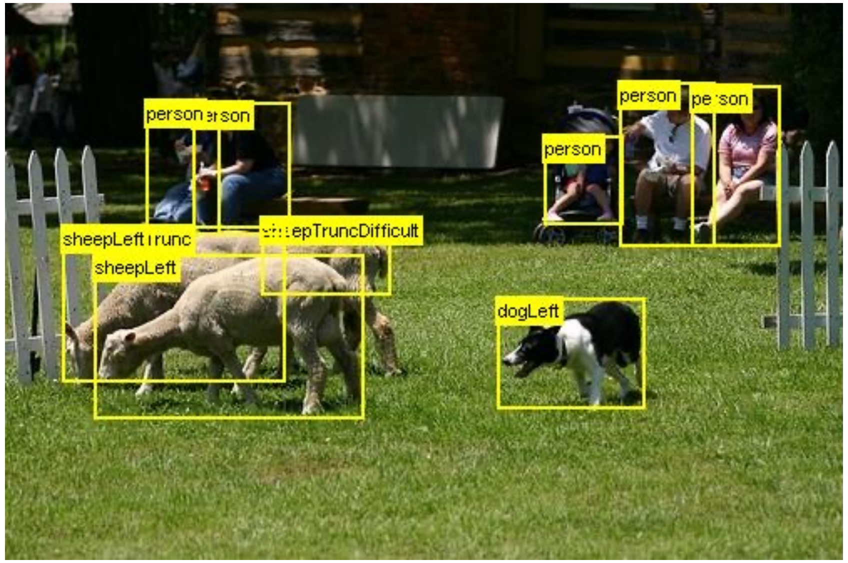
Evaluation



PASCAL VOC Challenge (2005-2012)



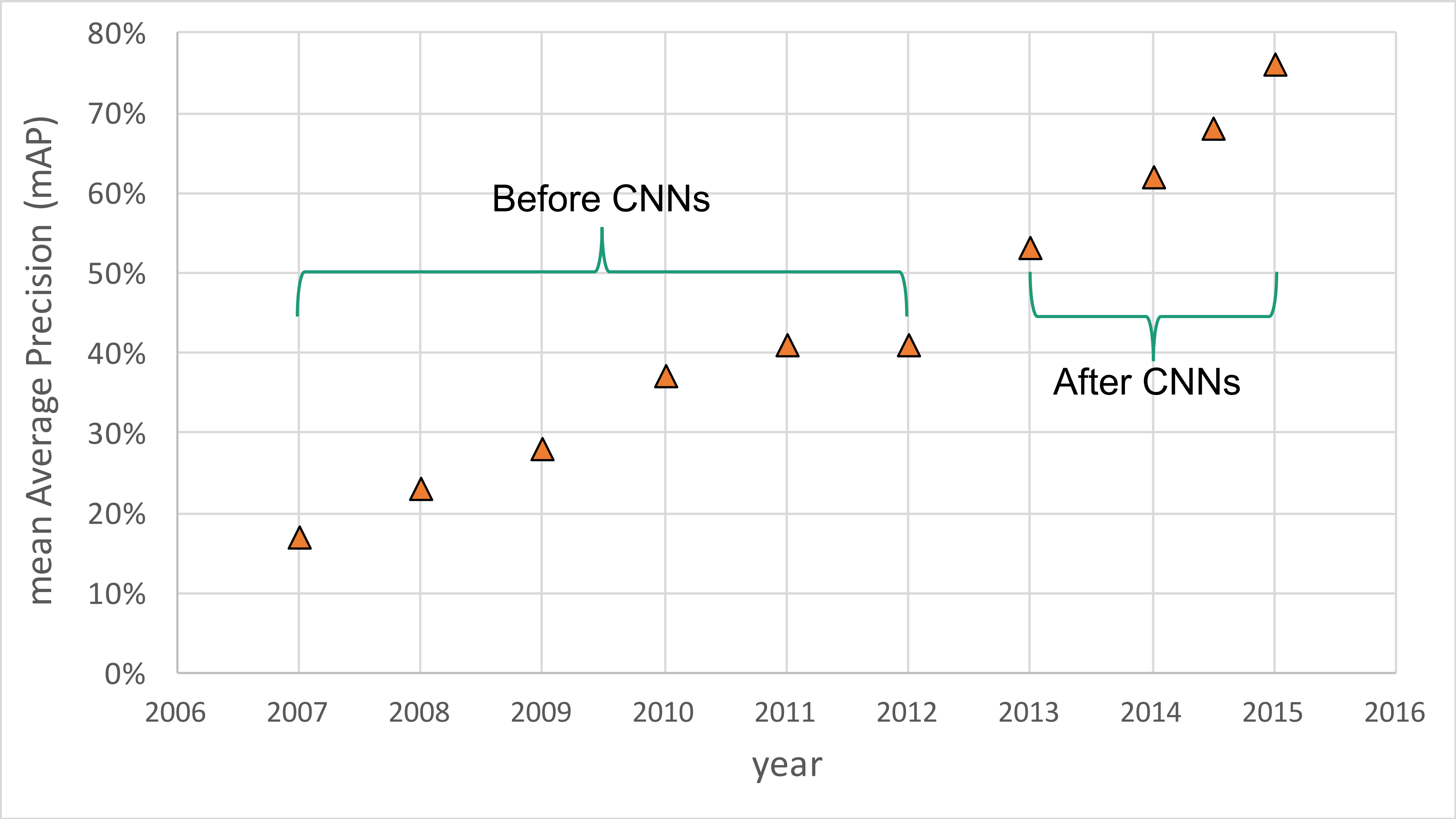




- 20 challenge classes:
  - *Person*
  - *Animals:* bird, cat, cow, dog, horse, sheep
  - *Vehicles:* airplane, bicycle, boat, bus, car, motorbike, train
  - *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor

- 11.5K training/validation images, 27K bounding boxes

PASCAL VOC Challenge (2005-2012)



COCO dataset

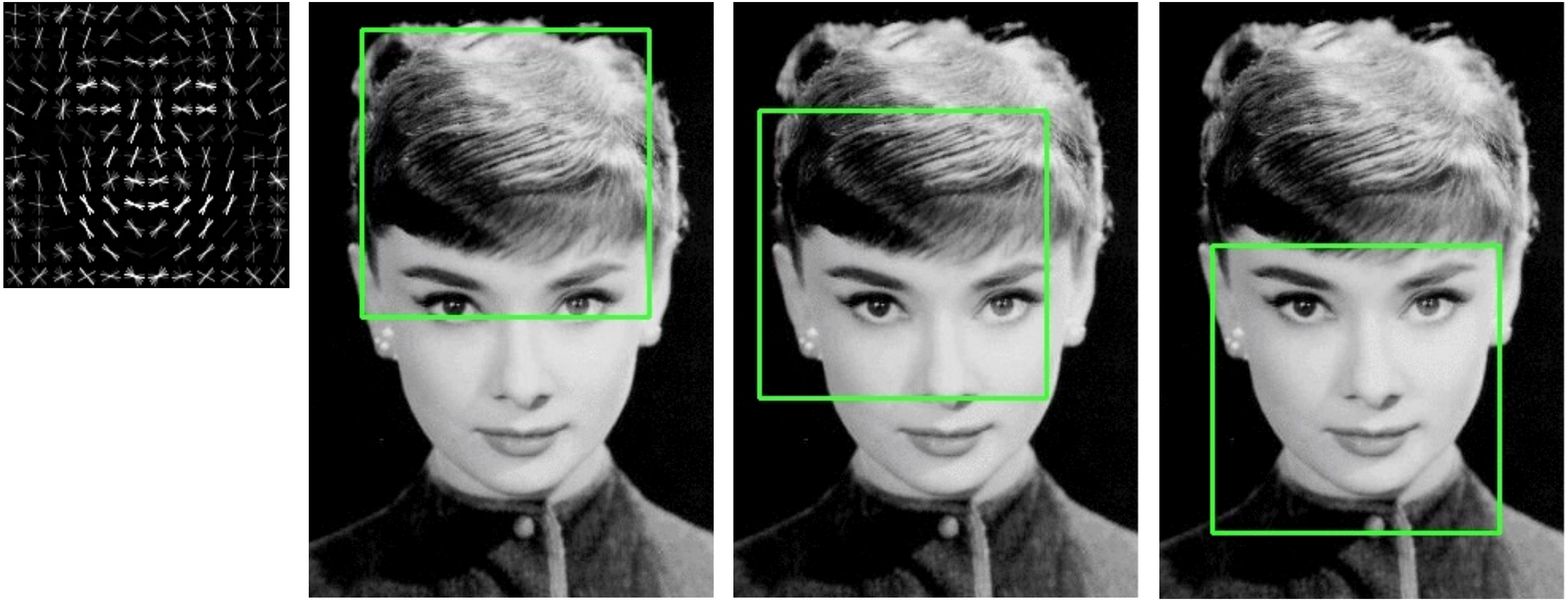
What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

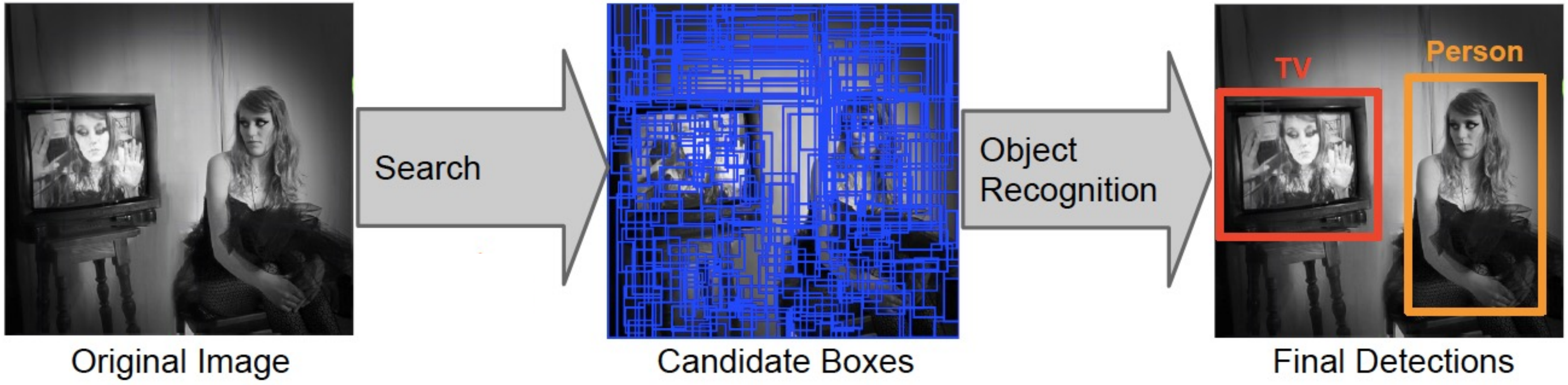
COCO
Common Objects in Context

Sliding window approach for detection



Histograms of Oriented Gradients. Dalal et al. 2005

Object proposal for object detection



- First generate a lot of region proposals (using low-level cues)
- Classification on each proposal

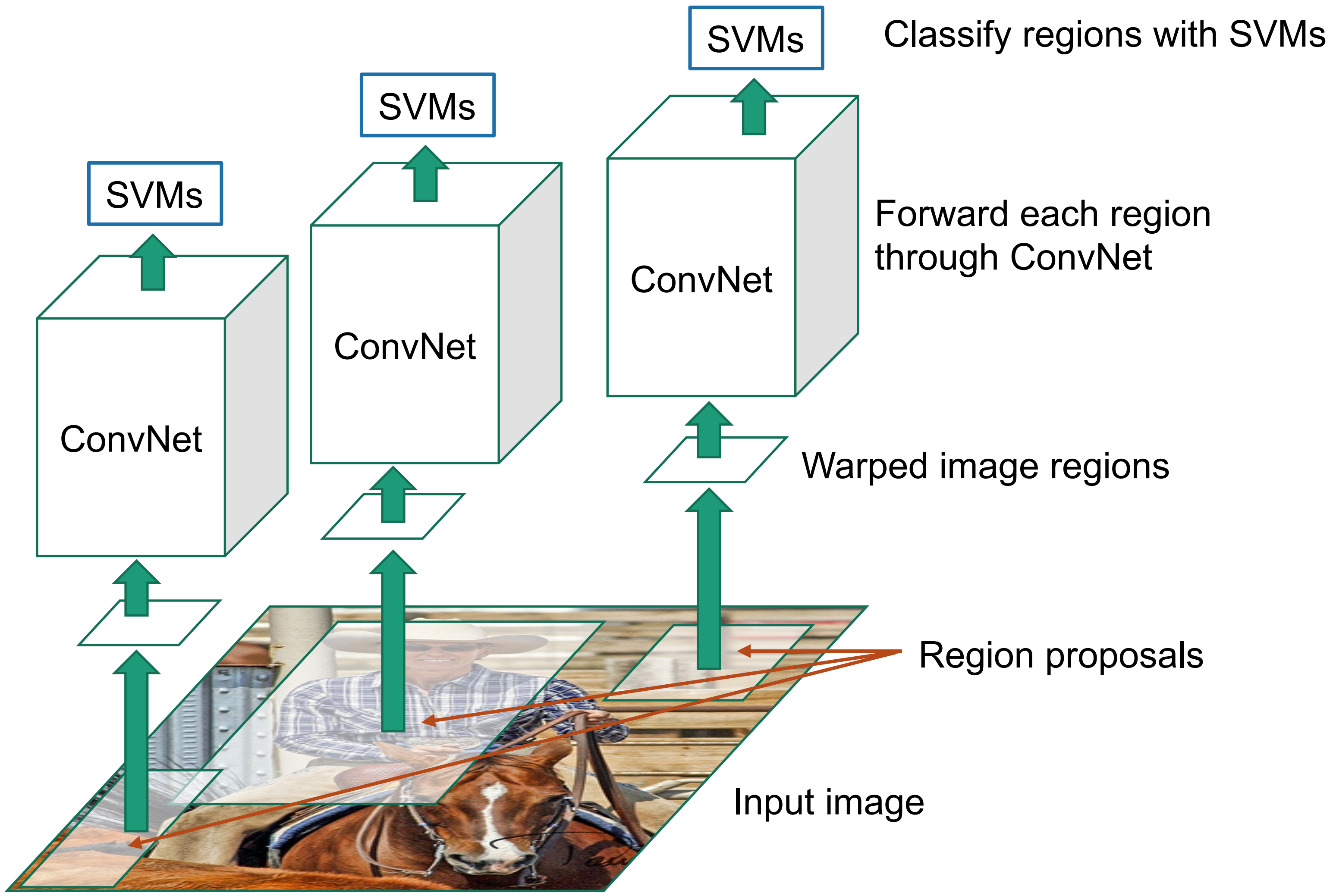
Uijlings et al., 2013

Selective search to generate object proposal for object detection

- Use hierarchical segmentation: start with small *superpixels* and merge based on diverse cues

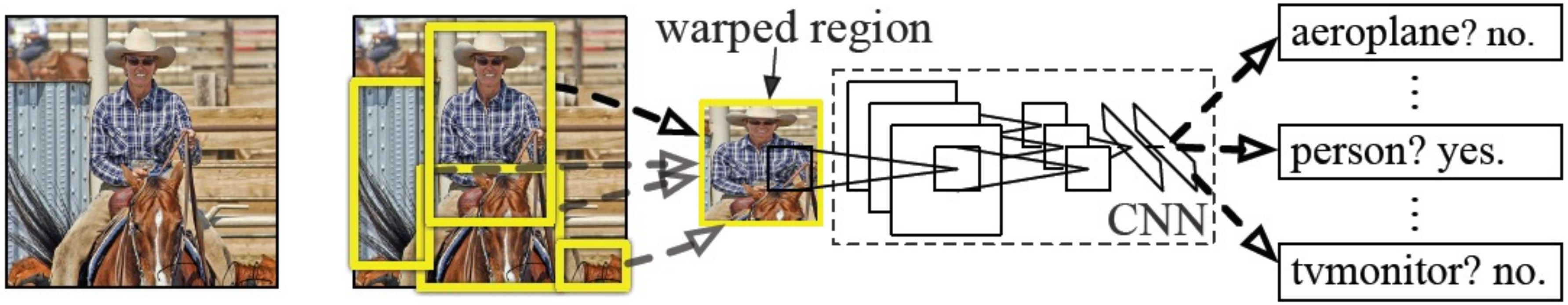
2-stage object detection

R-CNN: Region proposals + CNN



Girshick, et al, 2014

R-CNN: Region proposals + CNN

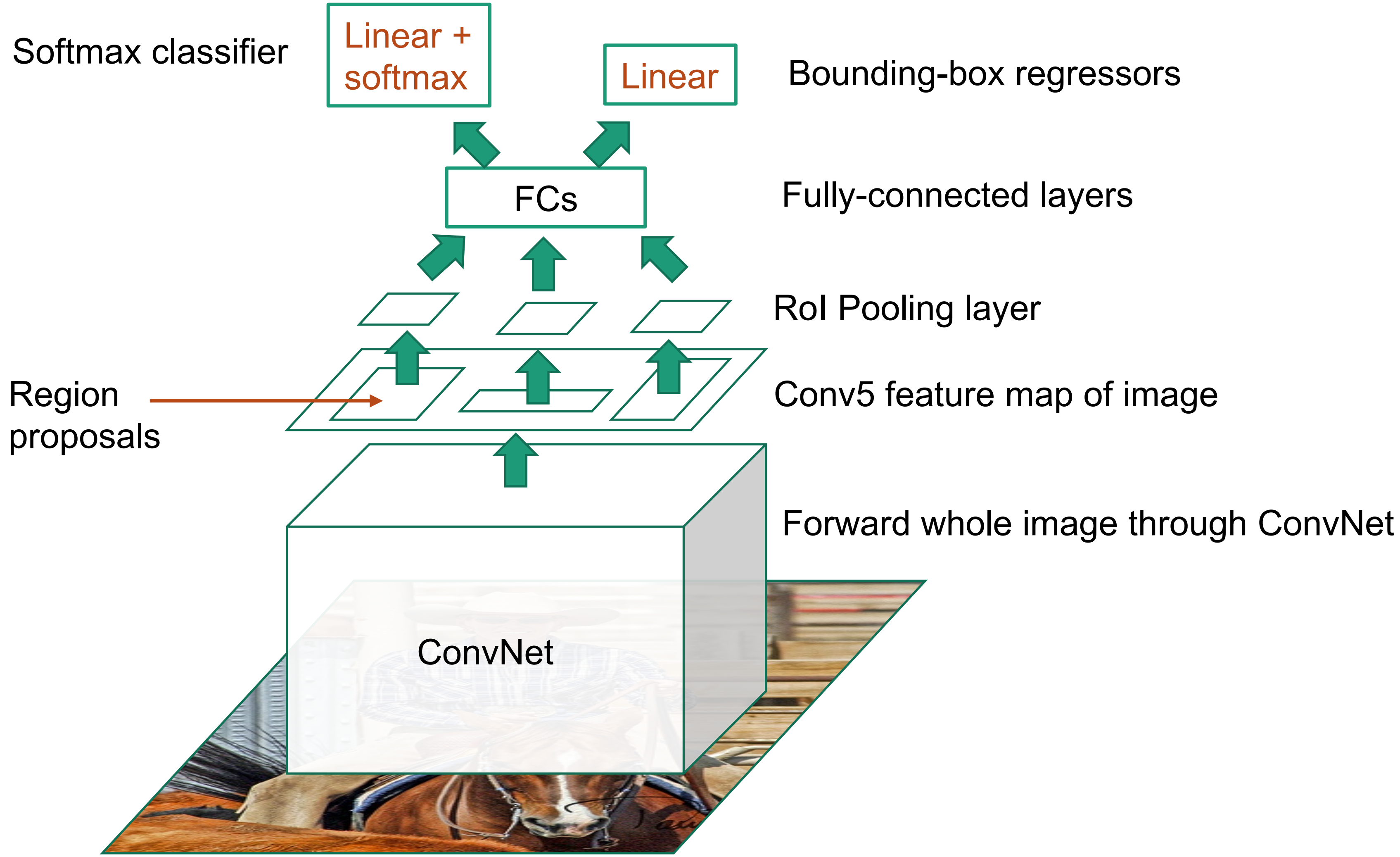


- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to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

R-CNN: Region proposals + CNN

- Pros
  - Much more accurate than previous approaches!
  - Any deep architecture can immediately be “plugged in”
- Cons
  - Not a single end-to-end system
    - Fine-tune network with softmax classifier (log loss)
    - Train post-hoc linear SVMs (hinge loss)
  - Training was slow (84h), took up a lot of storage
    - 2000 CNN passes per image
  - Inference (detection) was slow (47s / image with VGG16)

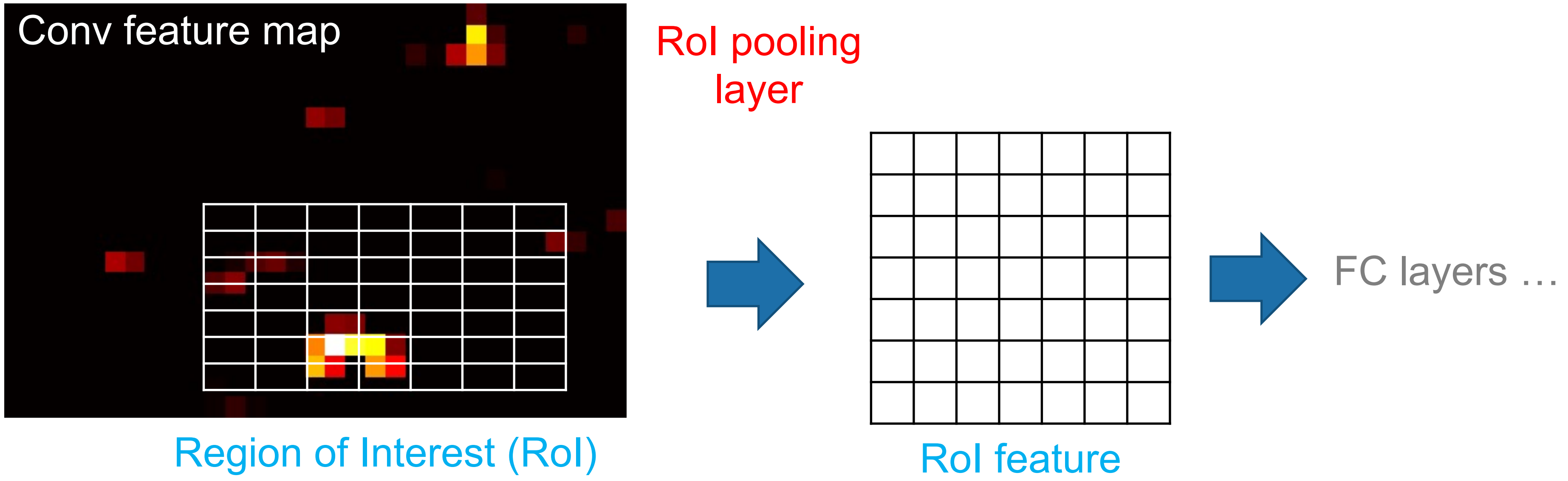
Fast R-CNN



R. Girshick, Fast R-CNN, ICCV 2015

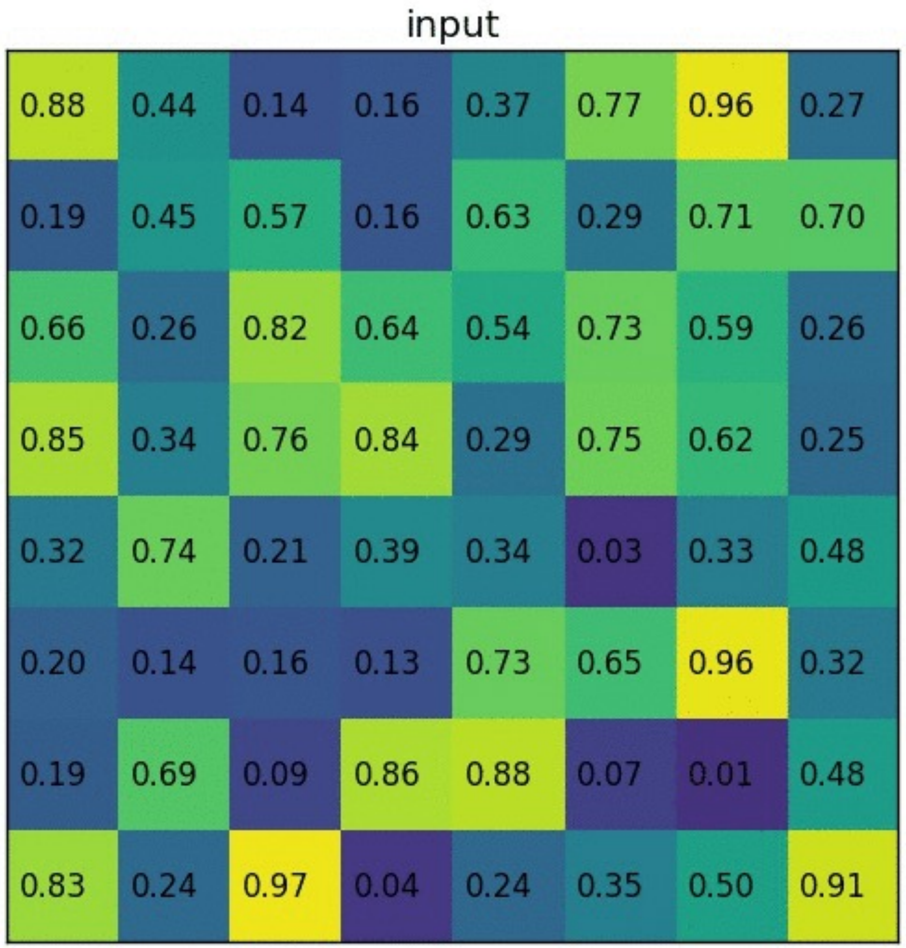
RoI pooling

"Crop and resample" a fixed-size feature representing a region of interest out of the outputs of the last conv layer



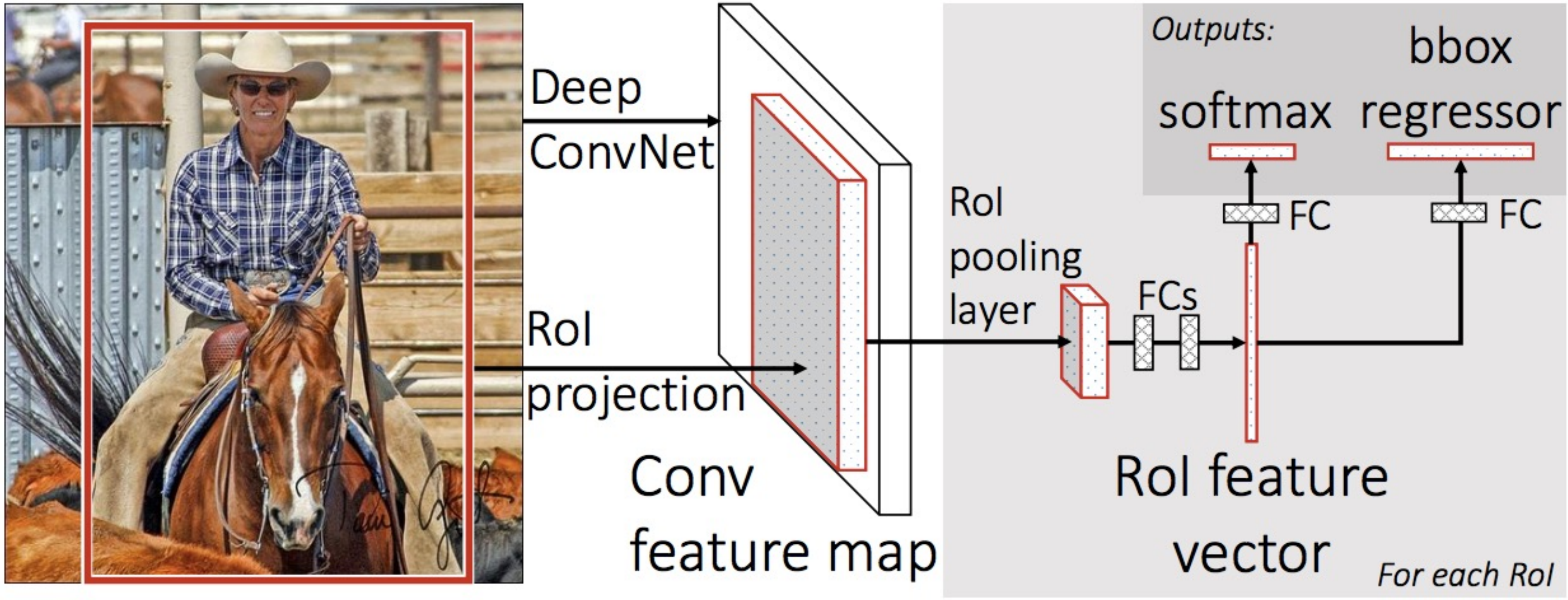
Source: R. Girshick, K. He

RoI pooling

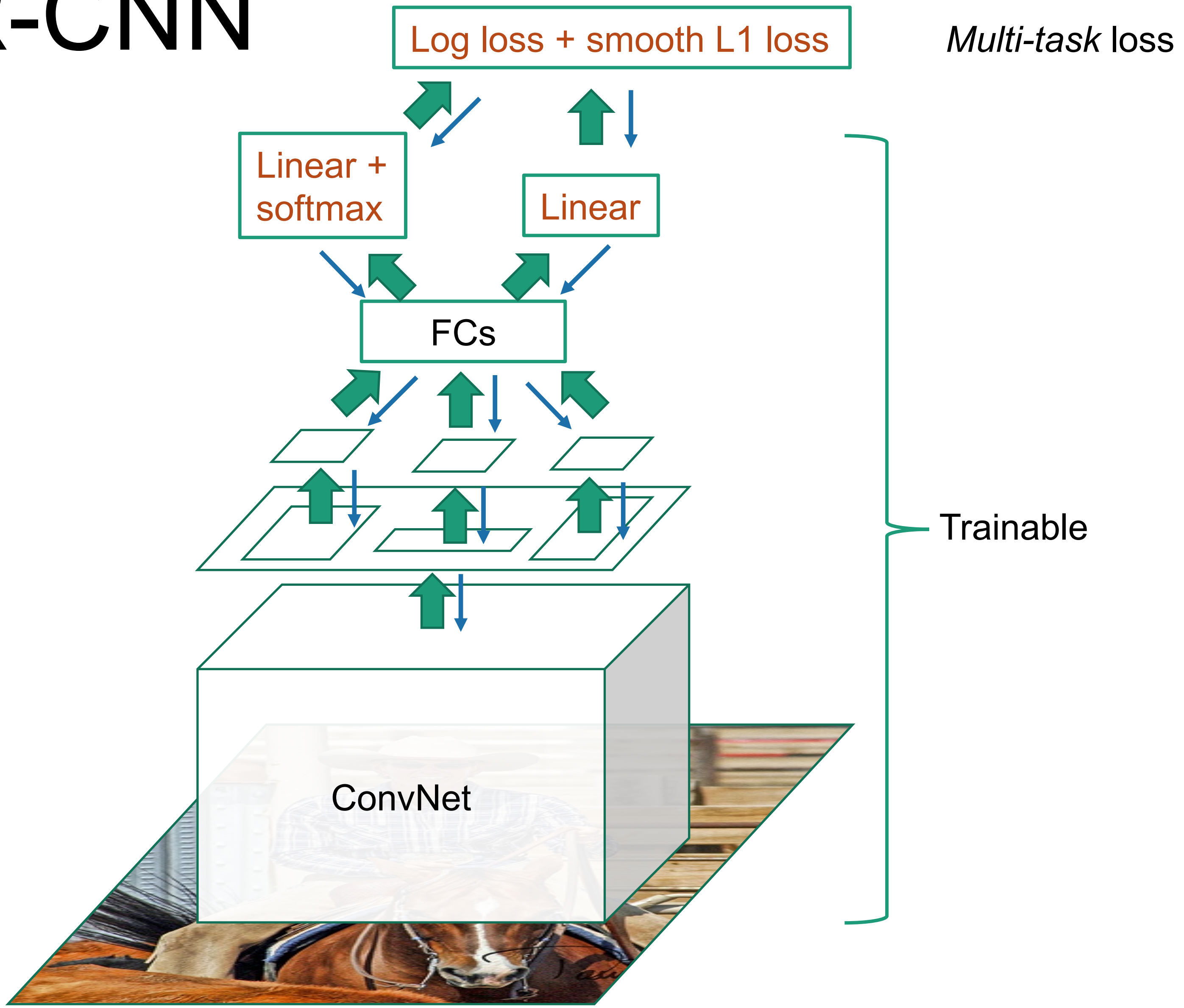

input
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91

Fast R-CNN

For each RoI, network predicts probabilities for $C + 1$ classes (class 0 is background) and four bounding box offsets for C classes



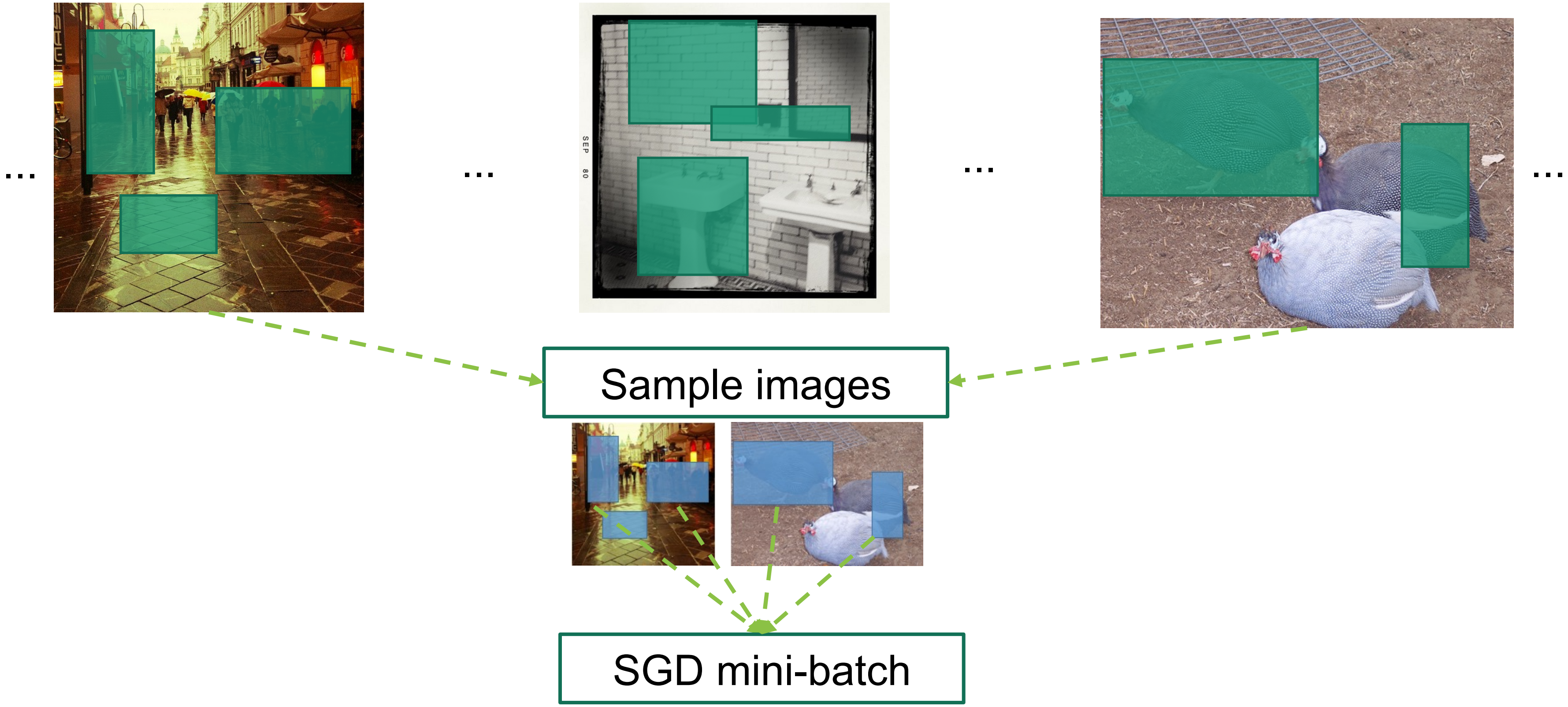
Fast R-CNN



R. Girshick, [Fast R-CNN](), ICCV 2015

Mini-batch sampling

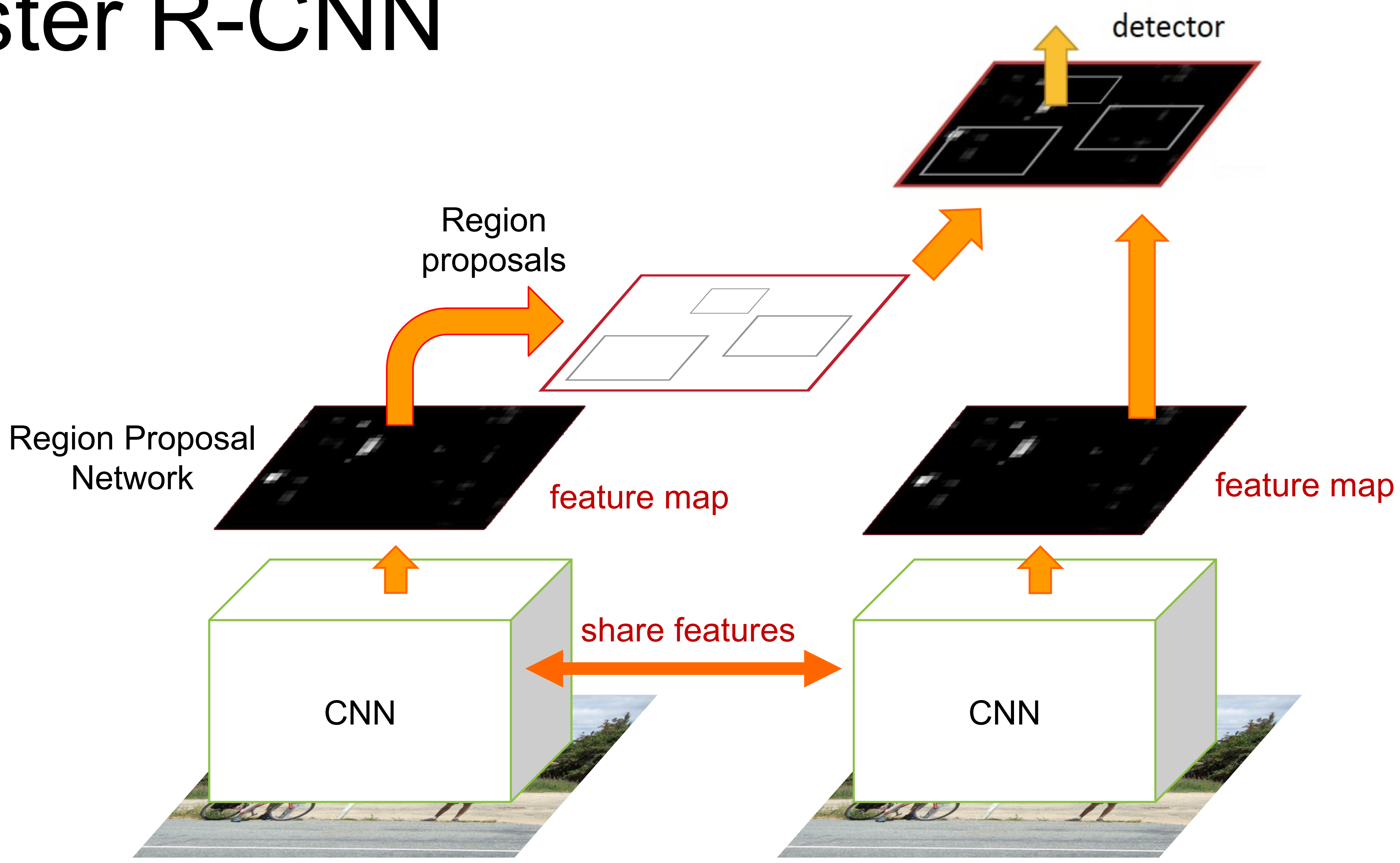
- Sample a few images (e.g., 2)
- Sample many regions from each image (64)



Fast R-CNN results with VGG16

	Fast R-CNN	R-CNN
Train time (h)	**9.5**	84
- Speedup	**8.8x**	
Test time / image	**0.32s**	47.0s
- Test speedup	**146x**	
mAP	**66.9%**	66.0%

Faster R-CNN

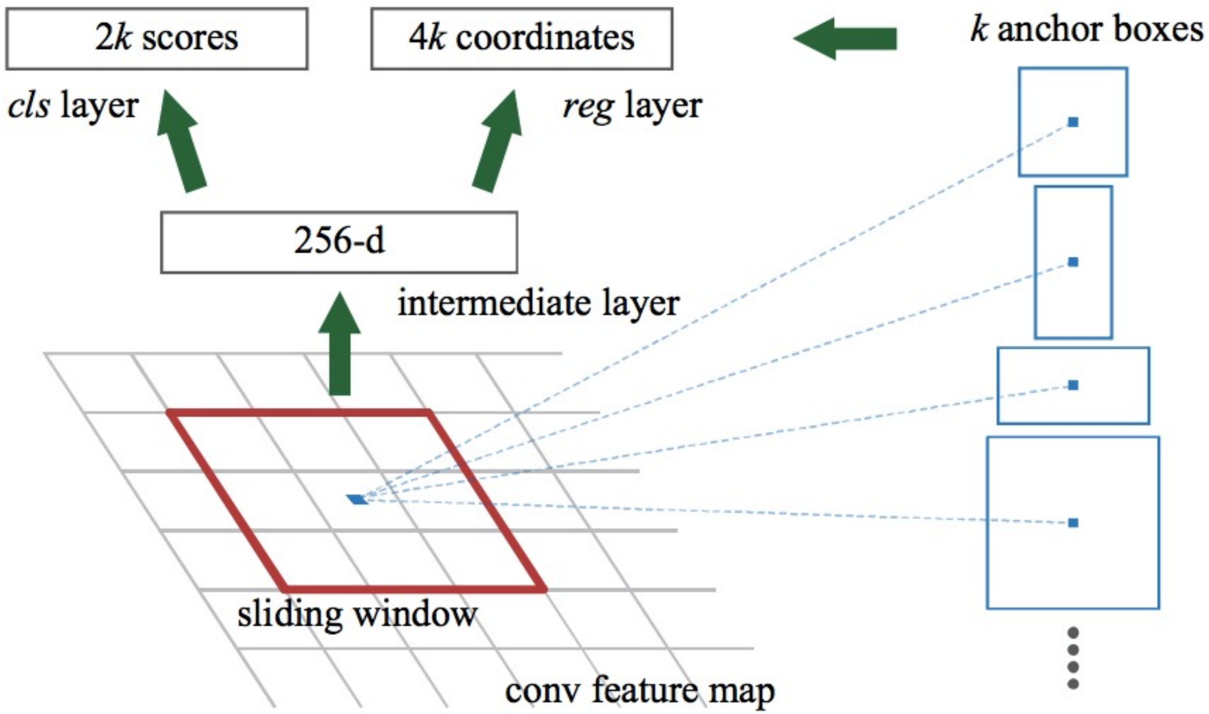


Ren, et al., NIPS 2015

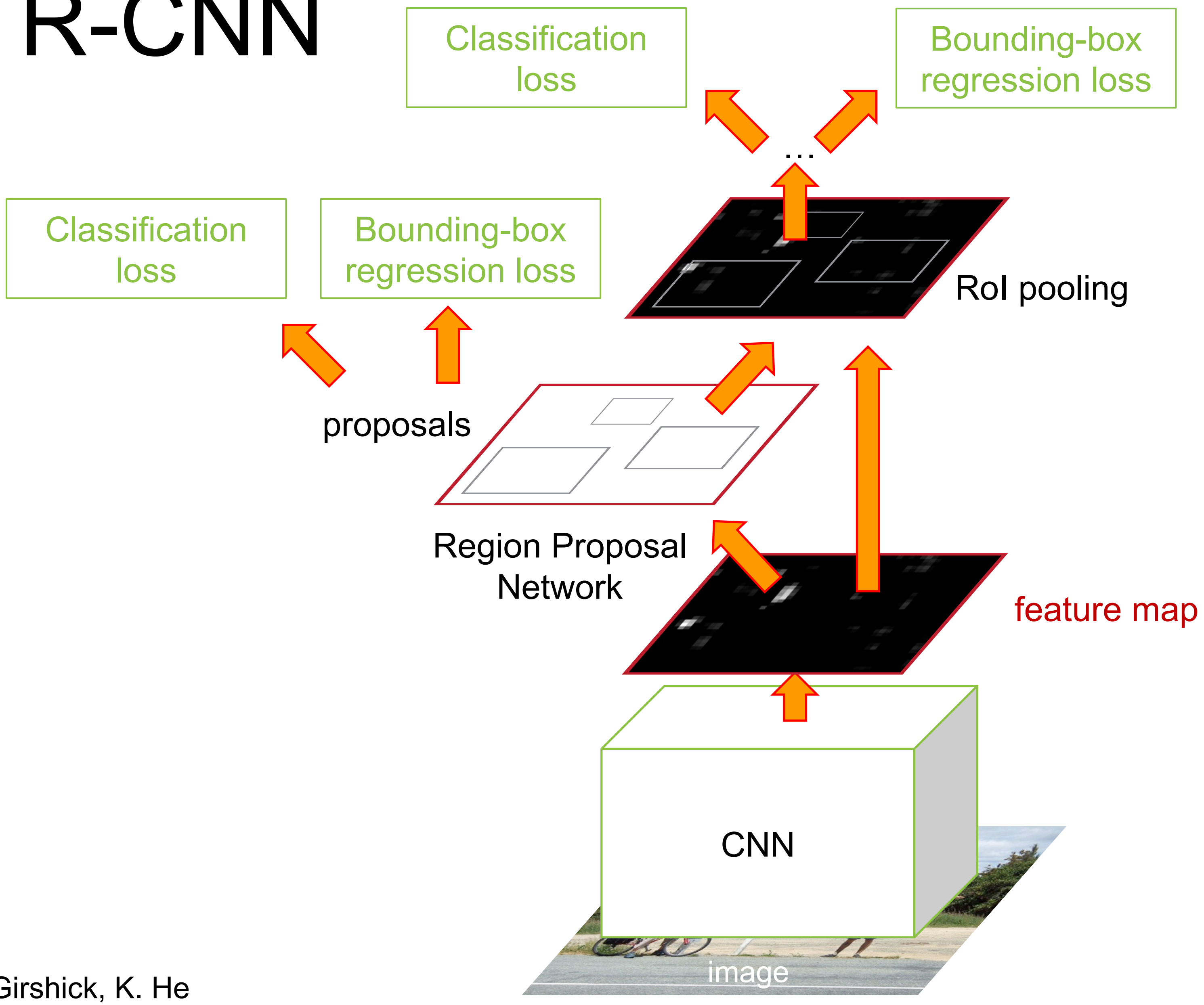
Region proposal network (RPN)

Slide a small window (3x3) over the conv5 layer

- Predict object/no object
- Regress bounding box coordinates with reference to *anchors* (3 scales x 3 aspect ratios)



Faster R-CNN



Source: R. Girshick, K. He

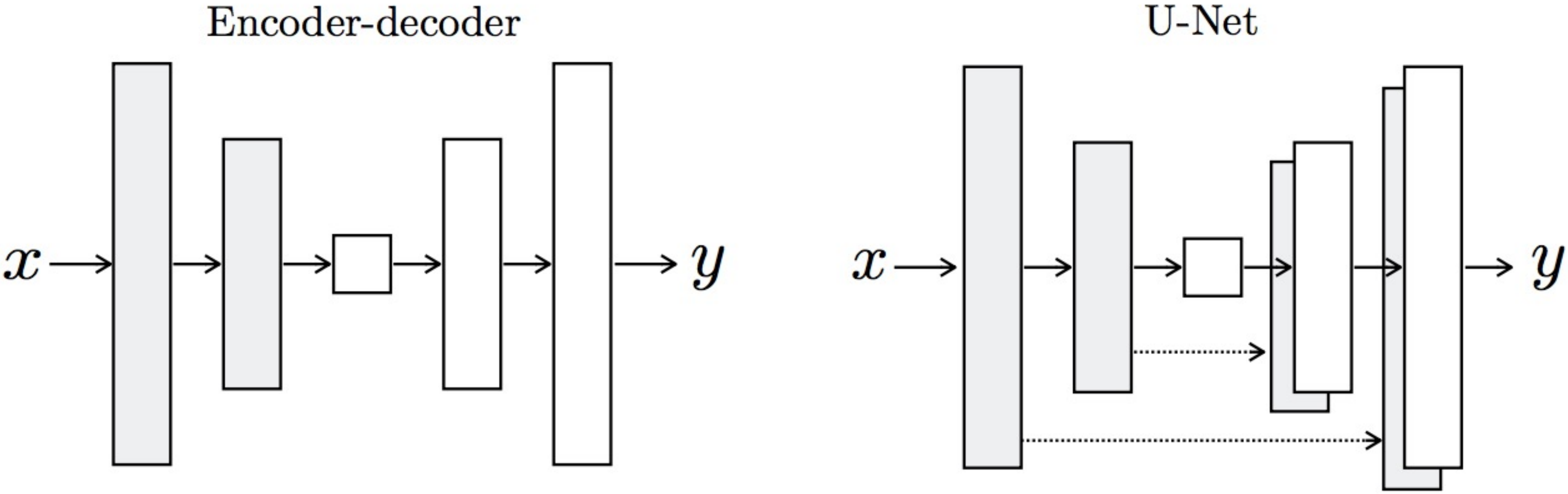
Faster R-CNN results

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

FPN

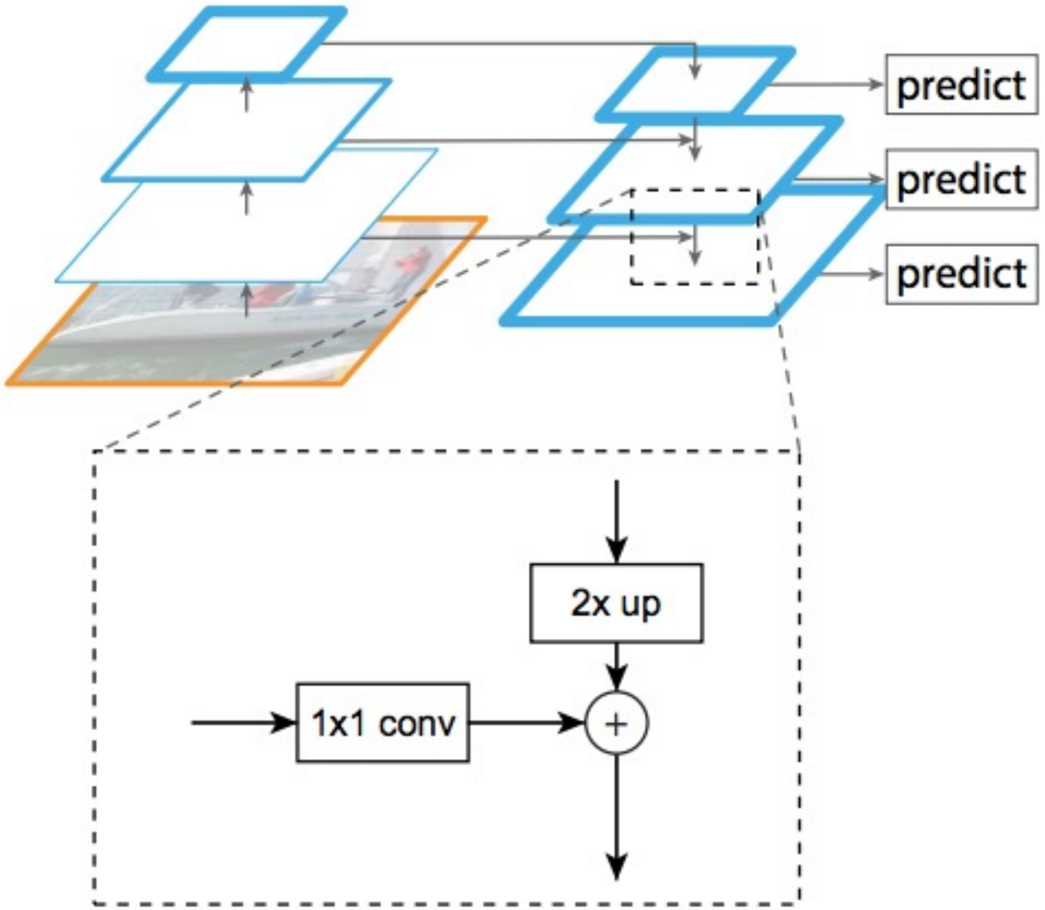
Feature pyramid networks



https://phillipi.github.io/pix2pix/

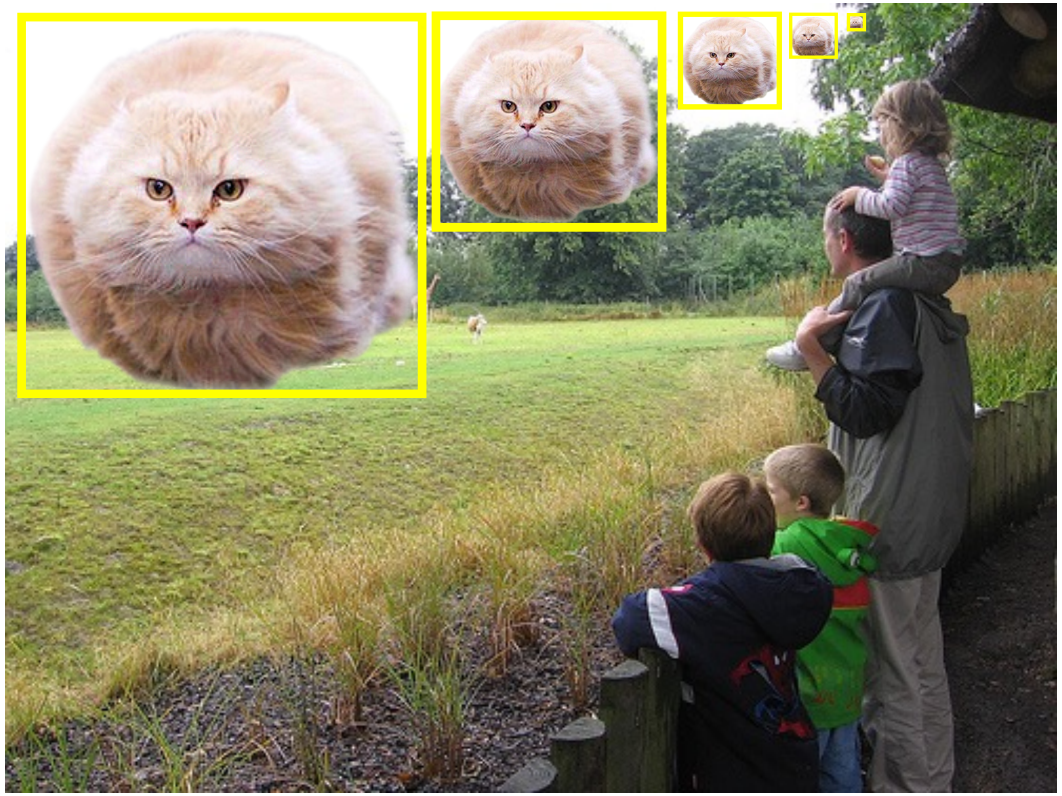
Feature pyramid networks

- Improve predictive power of lower-level feature maps by adding contextual information from higher-level feature maps
- Predict different sizes of bounding boxes from different levels of the pyramid (but share parameters of predictors)



Lin et al., CVPR 2017

Feature pyramid networks

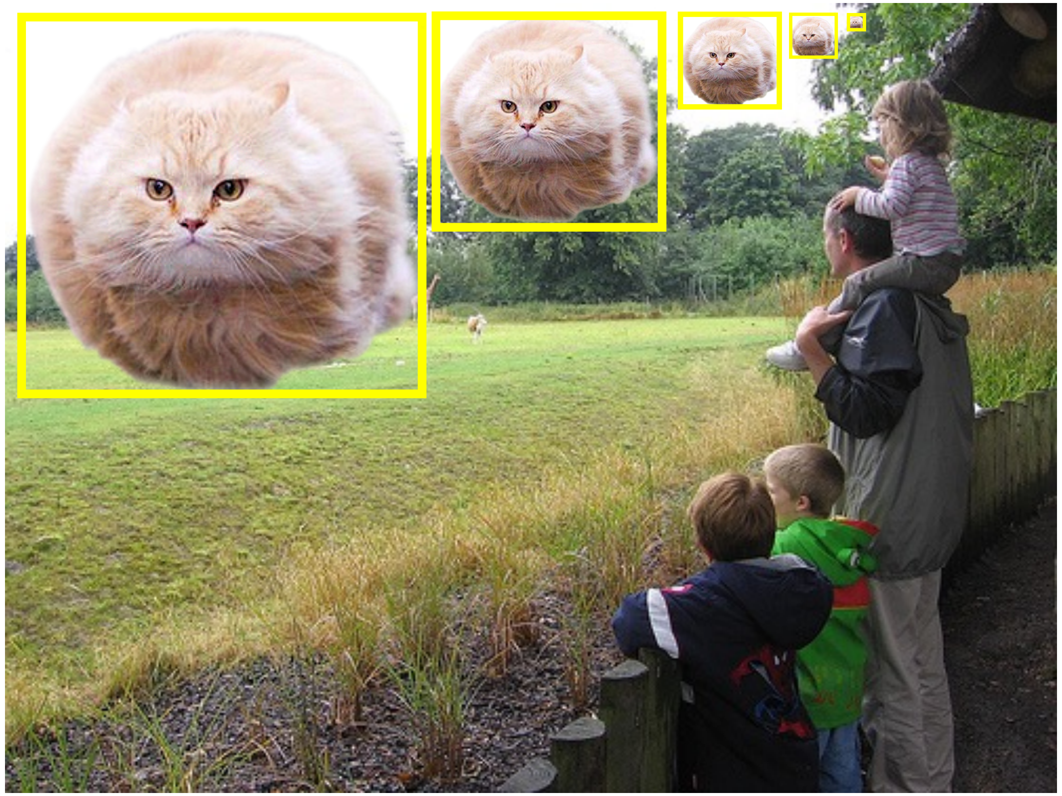


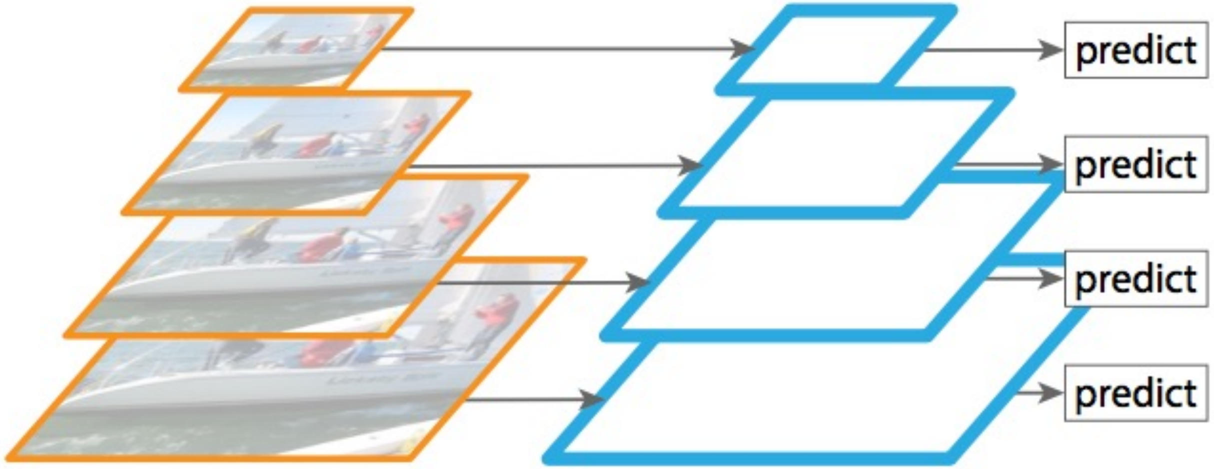
Detectors need to
1. classify and
2. localize

objects over a wide range of scales

FPN improves this ability

Strategy 1: Image Pyramid



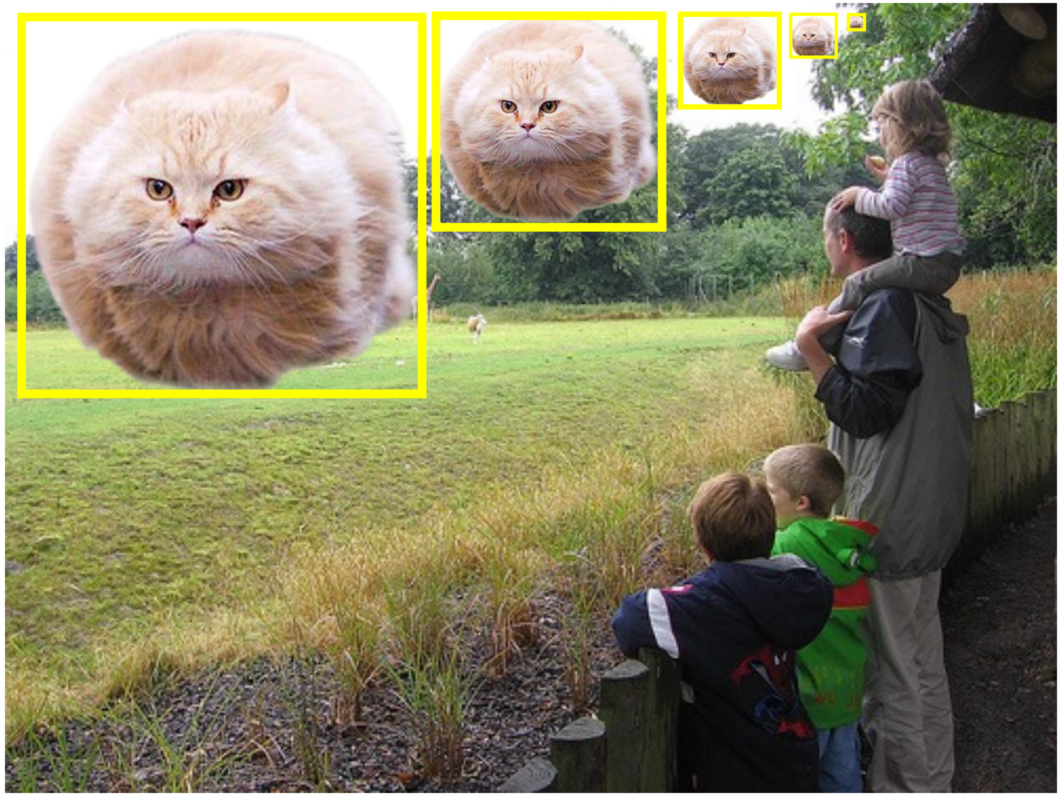


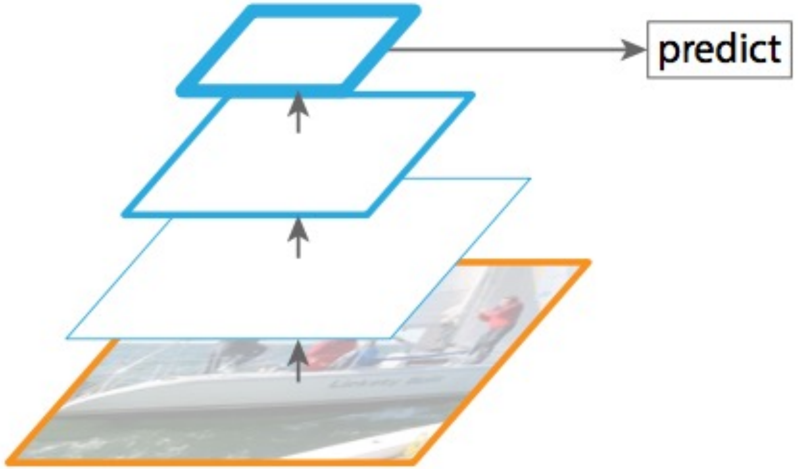
(a) Featurized image pyramid

Standard solution – *slow!*

(E.g., Viola & Jones, HOG, DPM, SPP-net, multi-scale Fast R-CNN, …)

Strategy 2: Multi-scale Features (Single-scale Map)



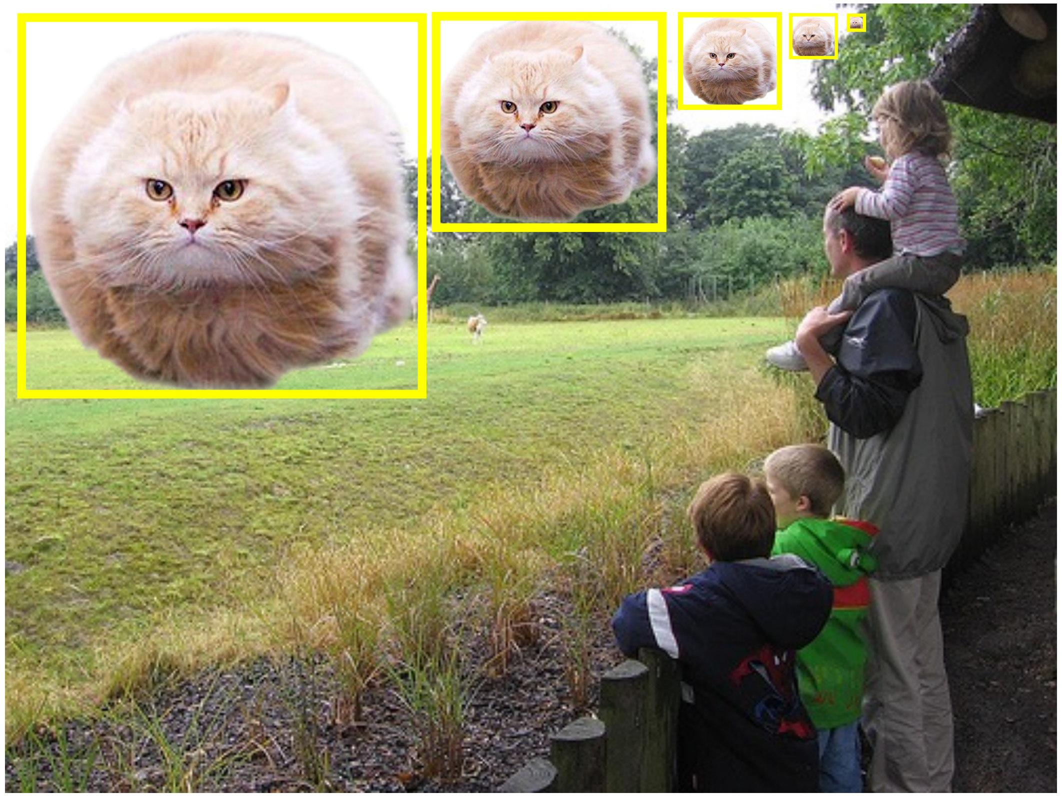


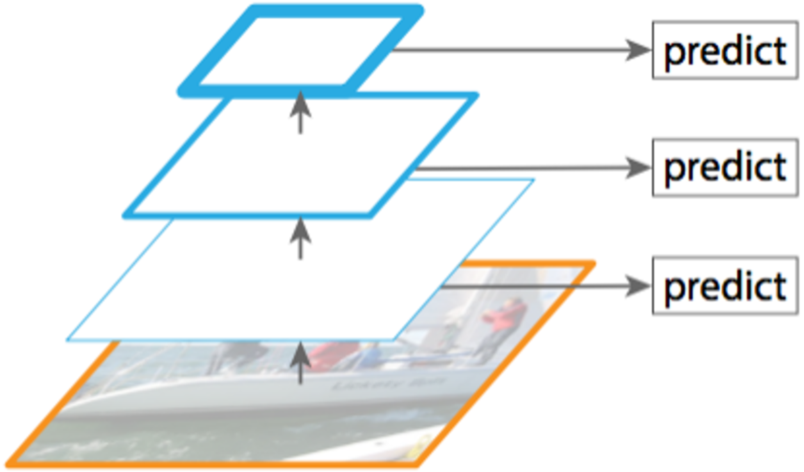
(b) Single feature map

Leave it all to the features – *fast, suboptimal*

(E.g., Fast/er R-CNN, YOLO, …)

Strategy 3: Naïve In-network Pyramid



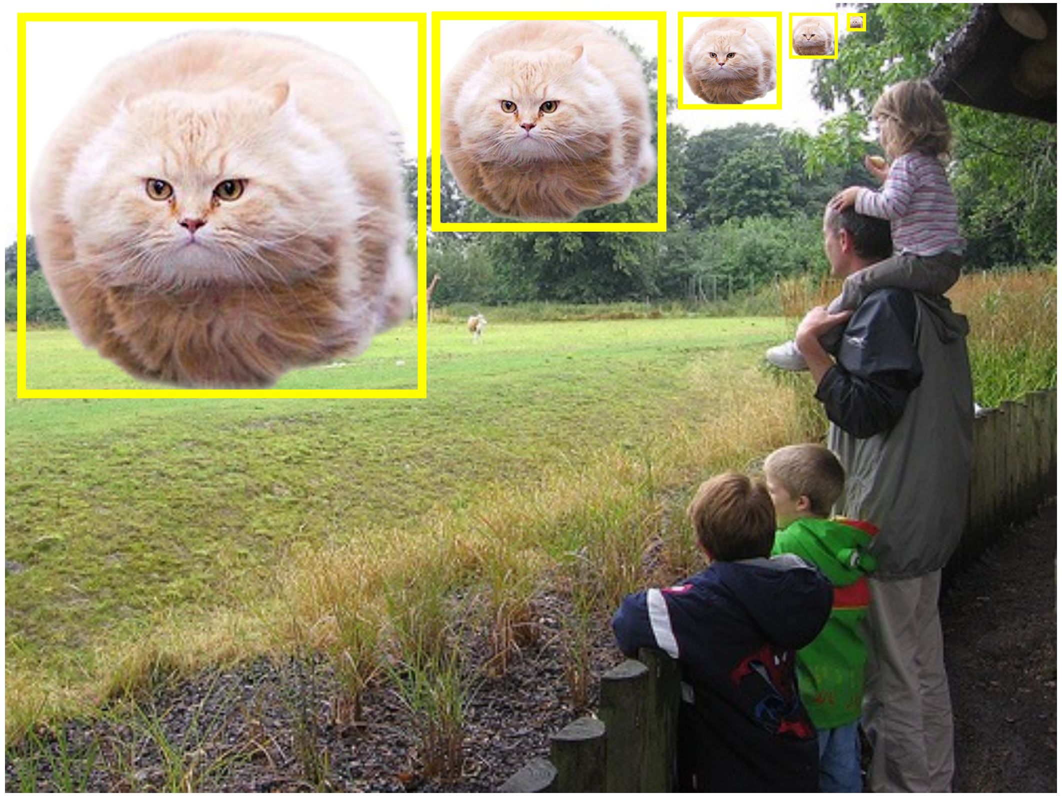


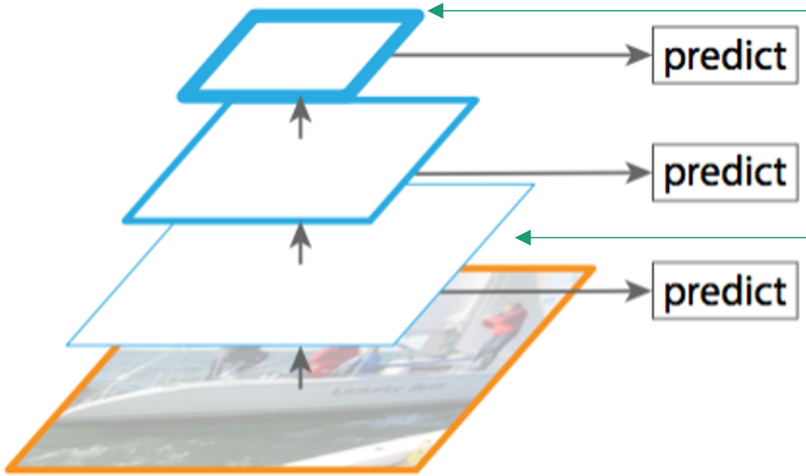
(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 3: Naïve In-network Pyramid





Low resolution,
Strong features

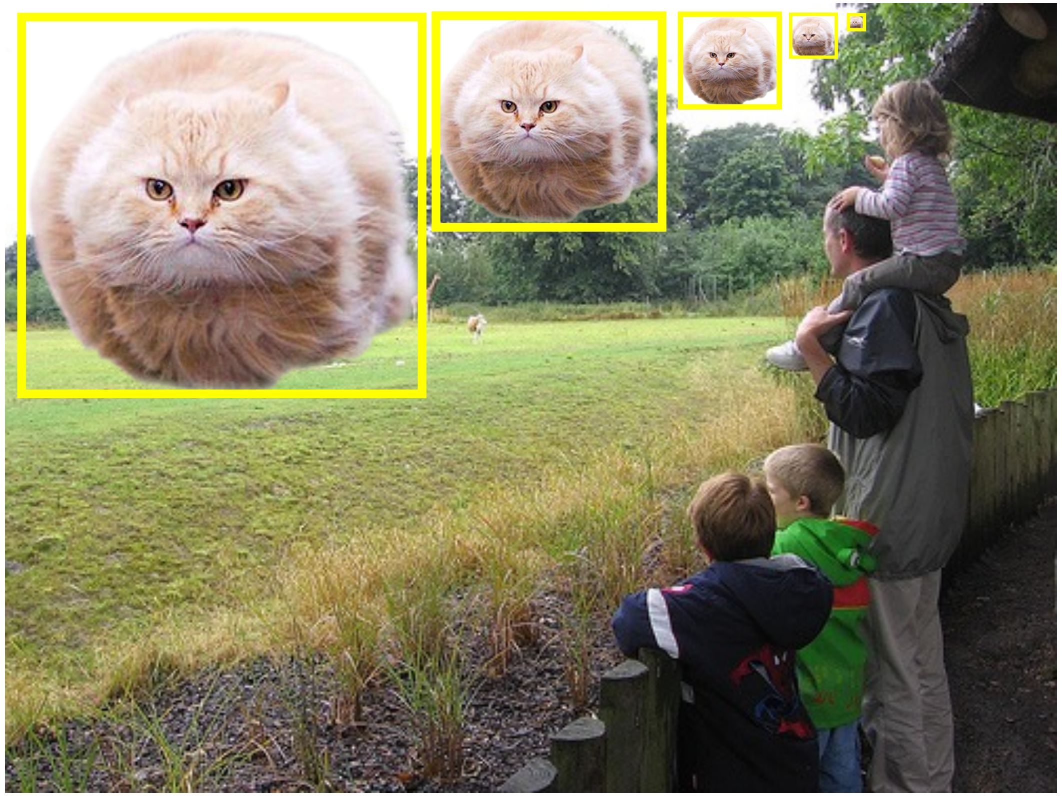
High resolution,
Weak features

(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 4: Feature Pyramid Network

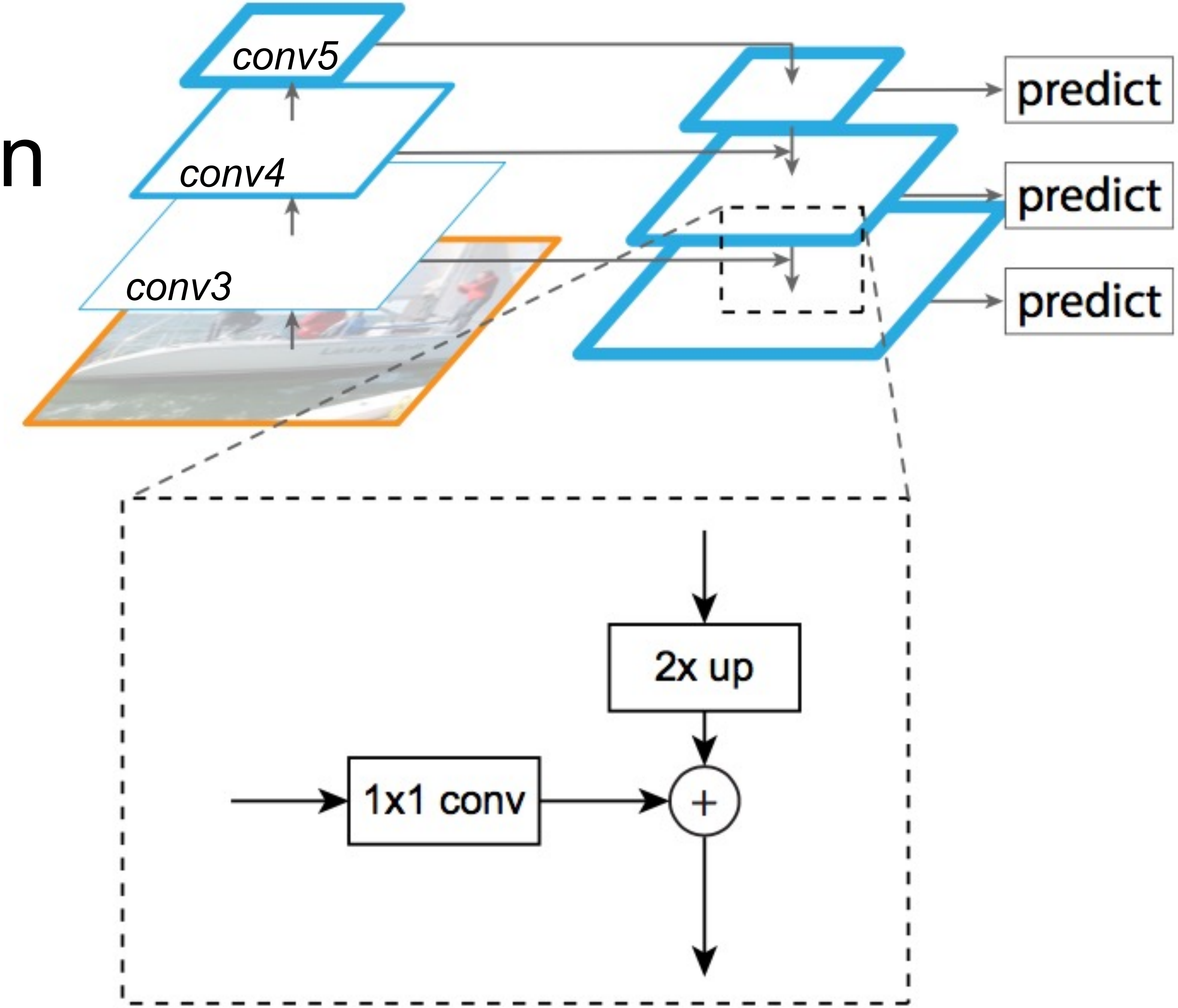


(d) Feature Pyramid Network

Top-down enrichment of high-res features – *fast, less suboptimal*

Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

FPN Top-down Refinement Module



Combine low and high resolution features.

Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

Summary

- Background and old fashion object detection
- 2-stage object detection
- FPN