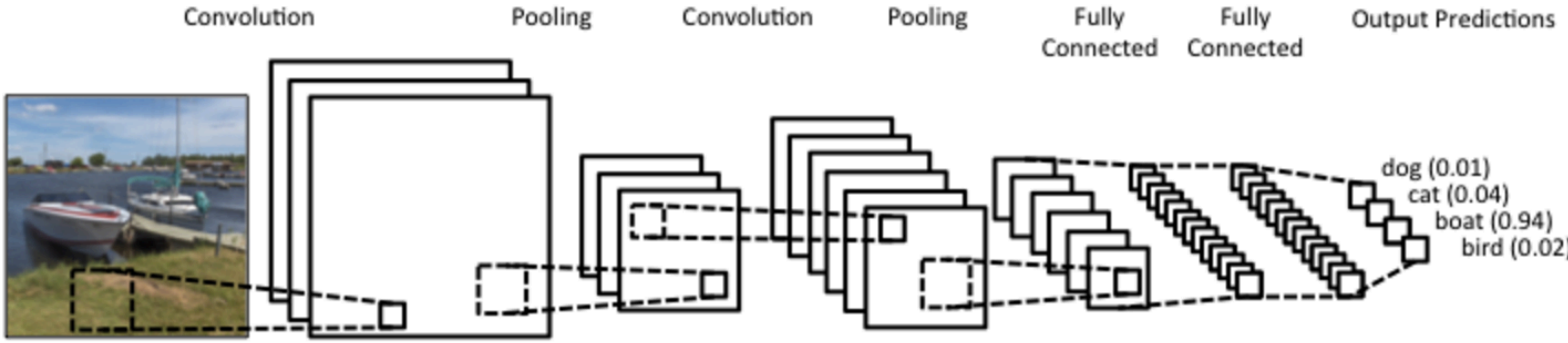
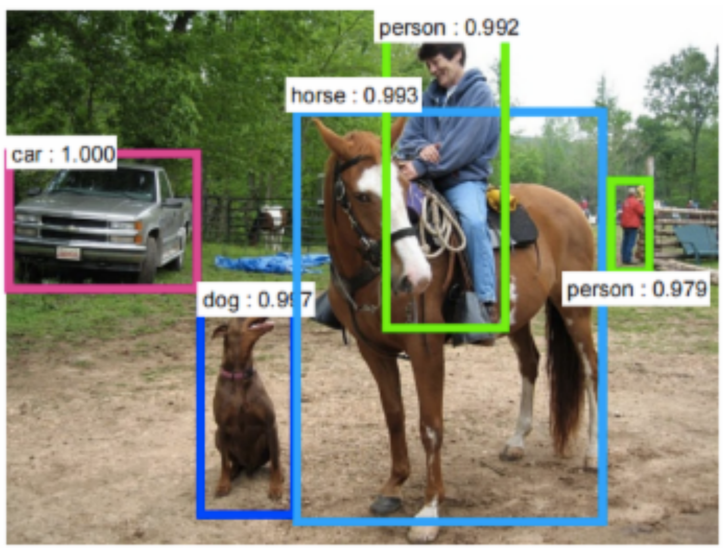
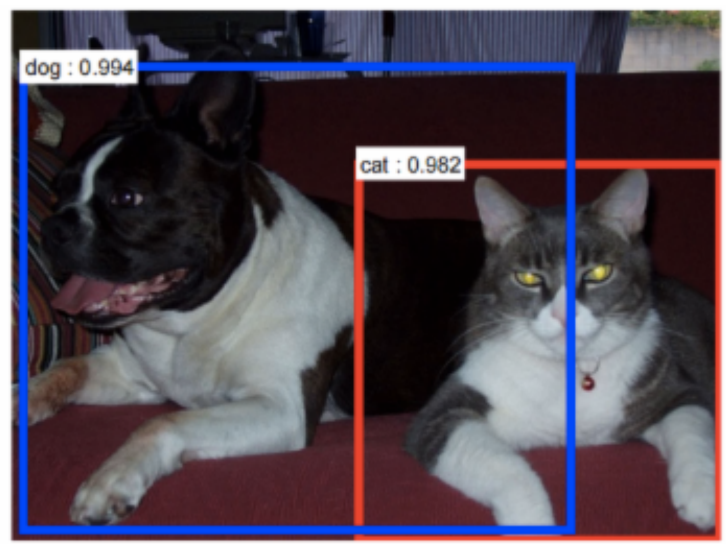
From image classification to object detection

Image classification



Object detection





Slides from L. Lazebnik

Image source

What are the challenges of object detection?

- Images may contain more than one class, multiple instances from the same class
- Bounding box localization
- Evaluation

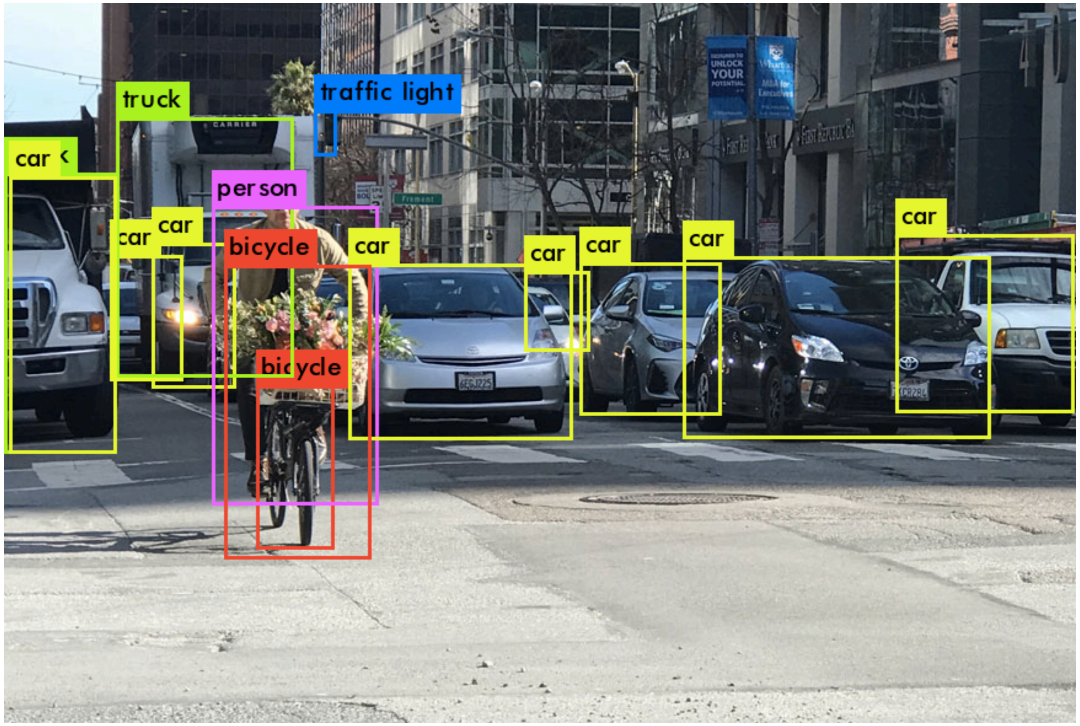


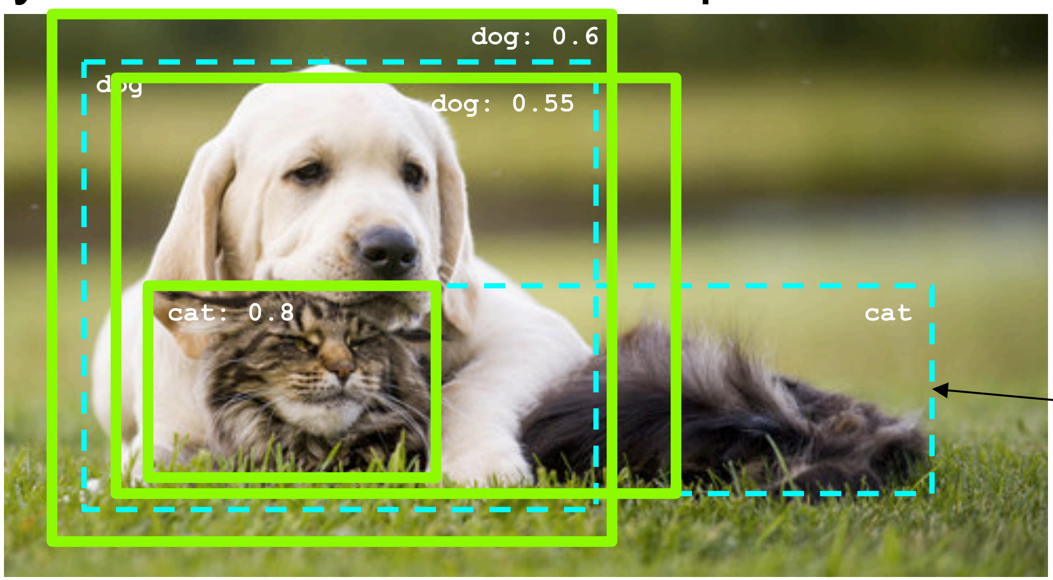
Image source

Outline

- Task definition and evaluation
- Generic object detection before deep learning
  - Sliding windows
  - HoG, DPMs (Components, Parts)
  - Region Classification Methods
- Deep detection approaches
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
  - SSD

Object detection evaluation

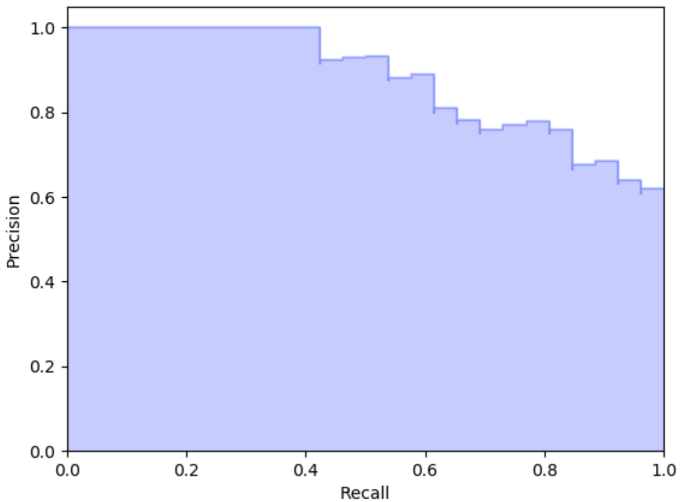
- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a true or false positive
  - PASCAL criterion: Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5
  - For multiple detections of the same ground truth box, only one considered a true positive



Ground truth (GT)

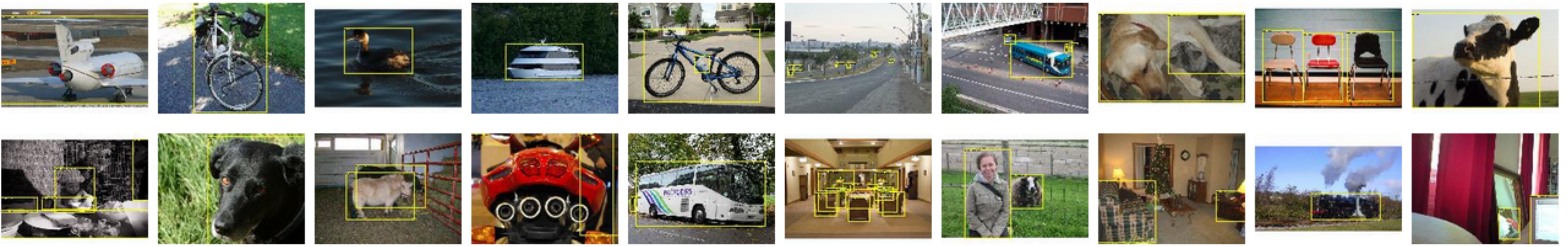
Object detection evaluation

- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a true or false positive
- 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

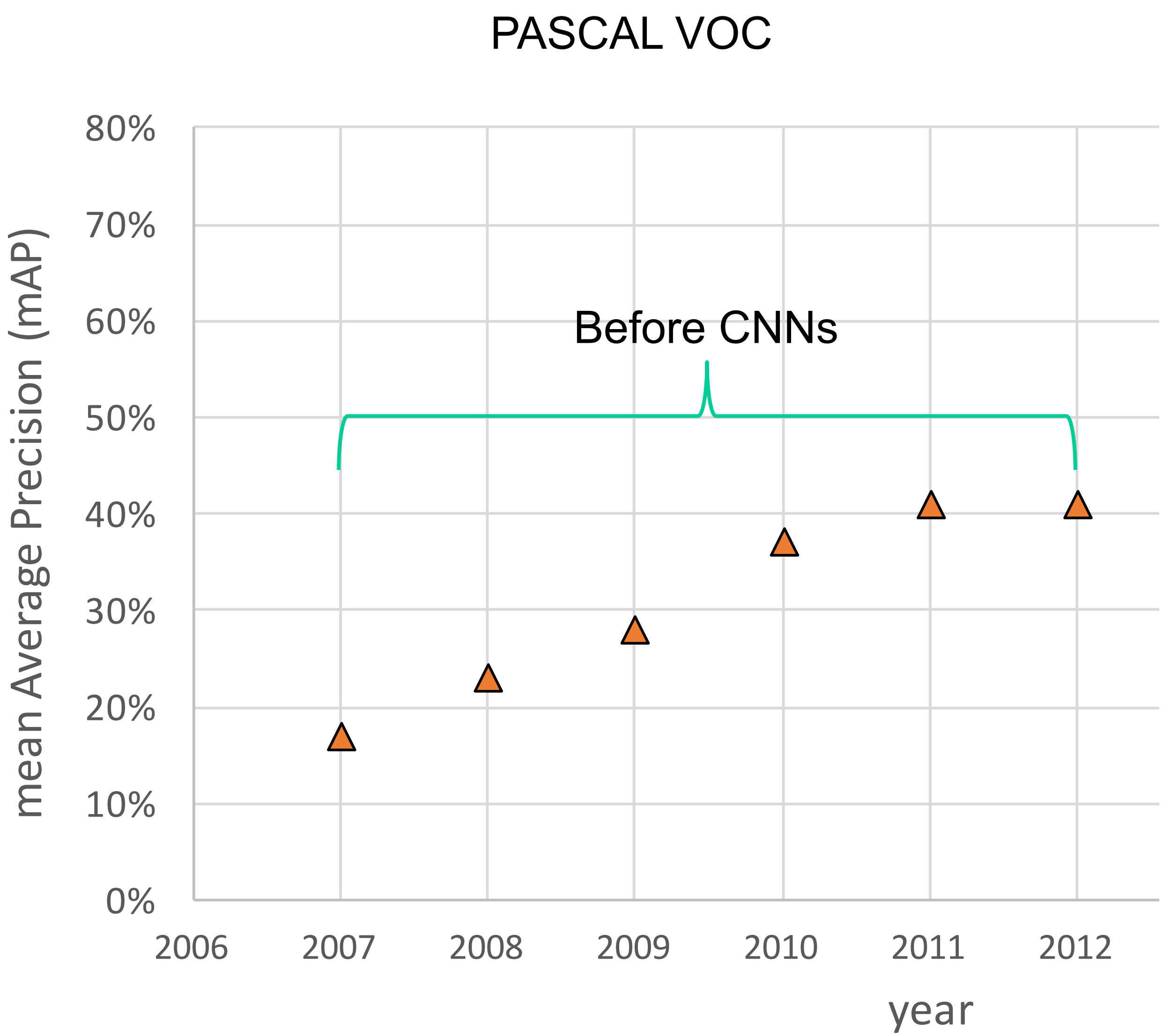
PASCAL VOC Challenge (2005-2012)



- 20 challenge classes:
  - *Person*
  - *Animals:* bird, cat, cow, dog, horse, sheep
  - *Vehicles:* aeroplane, bicycle, boat, bus, car, motorbike, train
  - *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor
- Dataset size (by 2012): 11.5K training/validation images, 27K bounding boxes, 7K segmentations

http://host.robots.ox.ac.uk/pascal/VOC/

Progress on PASCAL detection



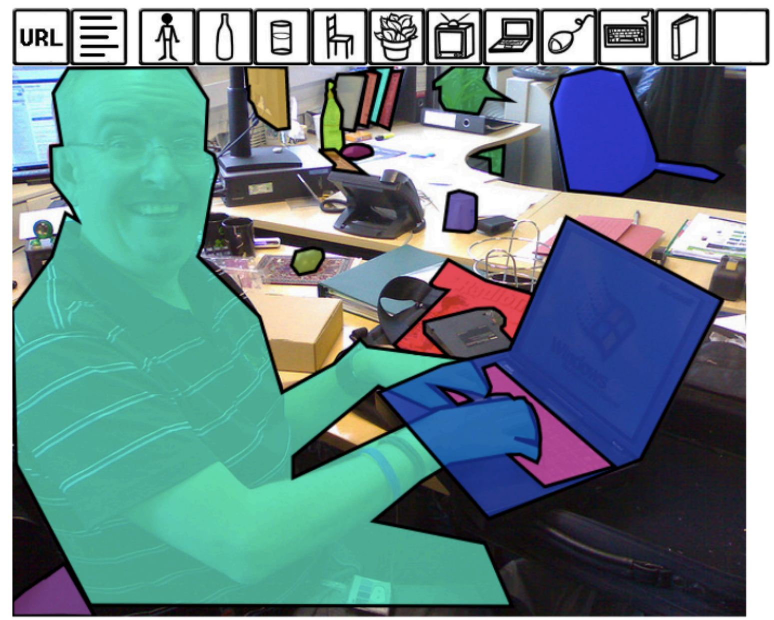
Newer benchmark: COCO

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



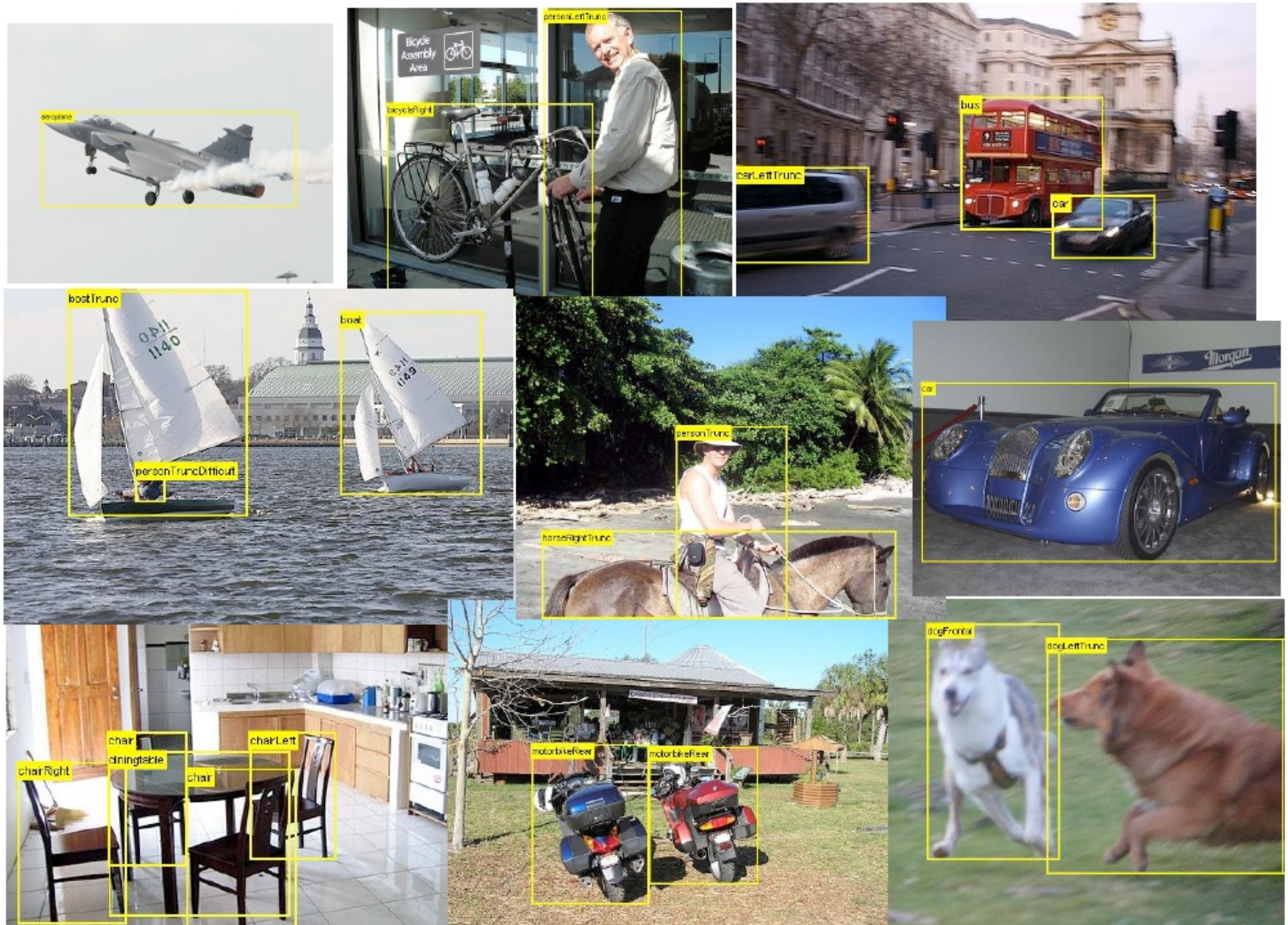
http://cocodataset.org/#home

COCO detection metrics

```
Average Precision (AP):
   AP                % AP at IoU=.50:.05:.95 (primary challenge metric)
   AP^IoU=.50        % AP at IoU=.50 (PASCAL VOC metric)
   AP^IoU=.75        % AP at IoU=.75 (strict metric)
AP Across Scales:
   AP^small          % AP for small objects: area < 32^2
   AP^medium         % AP for medium objects: 32^2 < area < 96^2
   AP^large          % AP for large objects: area > 96^2
Average Recall (AR):
   AR^max=1          % AR given 1 detection per image
   AR^max=10         % AR given 10 detections per image
   AR^max=100        % AR given 100 detections per image
AR Across Scales:
   AR^small          % AR for small objects: area < 32^2
   AR^medium         % AR for medium objects: 32^2 < area < 96^2
   AR^large          % AR for large objects: area > 96^2
```

- Leaderboard: http://cocodataset.org/#detection-leaderboard
  - Current best mAP: ~52%
- Official COCO challenges no longer include detection
  - More emphasis on instance segmentation and dense segmentation

Detection before deep learning

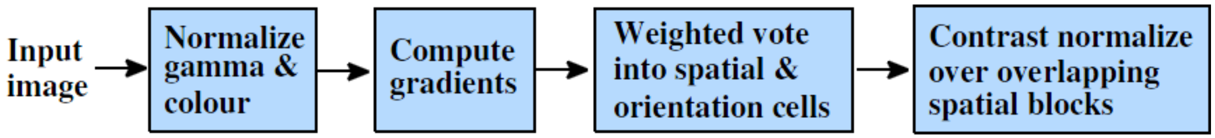


Conceptual approach: Sliding window detection

- Slide a window across the image and evaluate a detection model at each location
  - Thousands of windows to evaluate: efficiency and low false positive rates are essential
  - Difficult to extend to a large range of scales, aspect ratios

Histograms of oriented gradients (HOG)

- Partition image into blocks and compute histogram of gradient orientations in each block



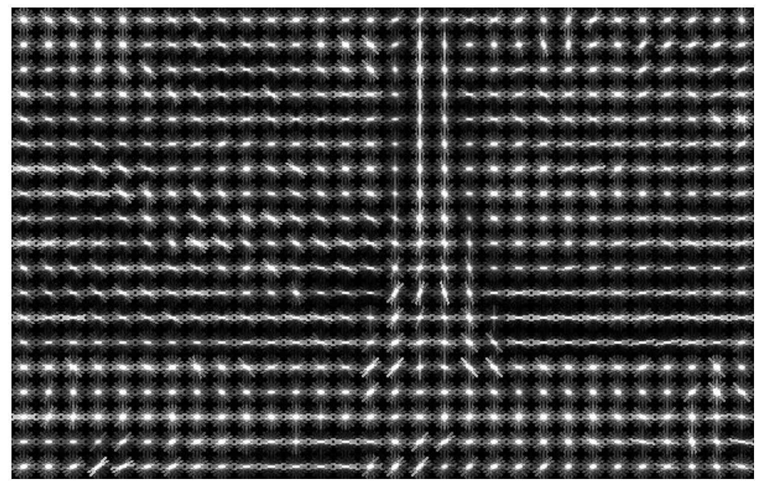


Image credit: N. Snavely

N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](), CVPR 2005

Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine

positive training examples

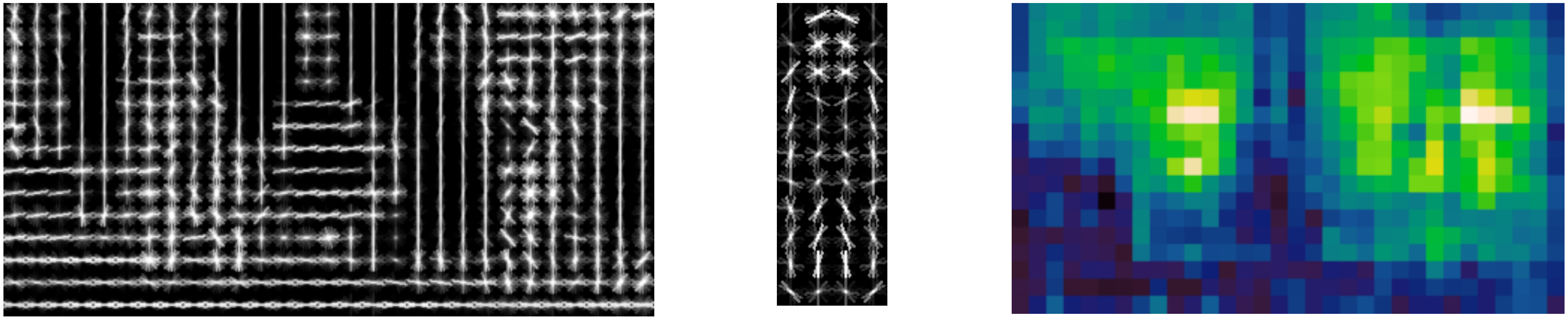
negative training examples

N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](), CVPR 2005

Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG *pyramid*

HOG feature map
Template
Detector response map



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Discriminative part-based models

- Single rigid template usually not enough to represent a category
  - Many objects (e.g. humans) are articulated, or have parts that can vary in configuration

  - Many object categories look very different from different viewpoints, or from instance to instance

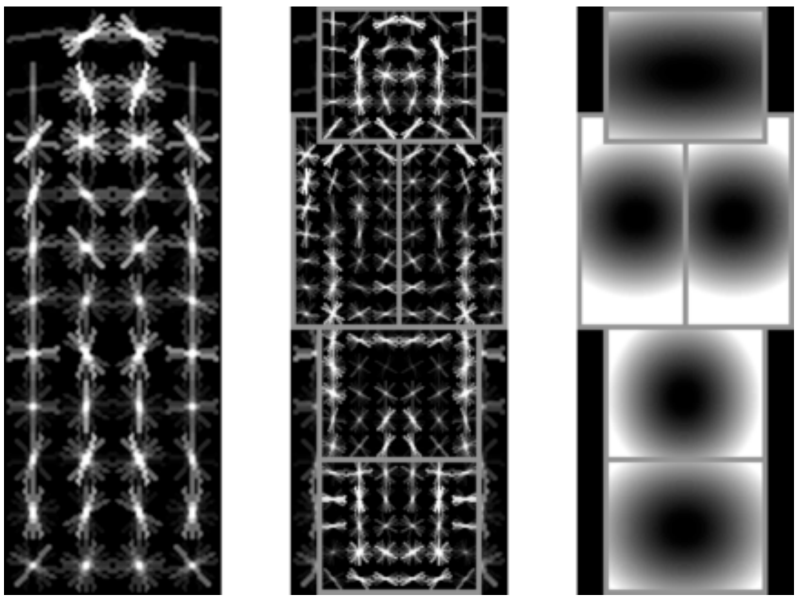
Slide by N. Snavely

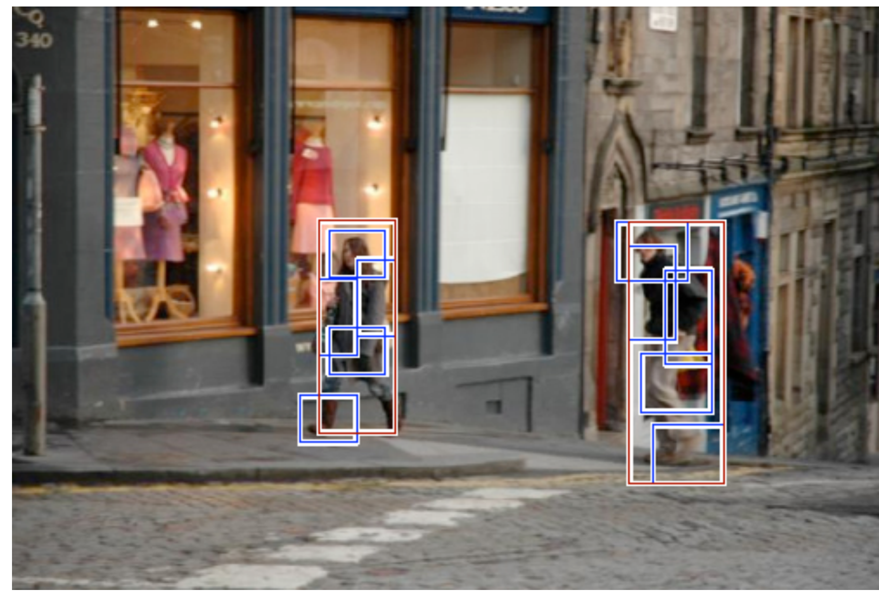
Discriminative part-based models

Root filter

Part filters

Deformation weights

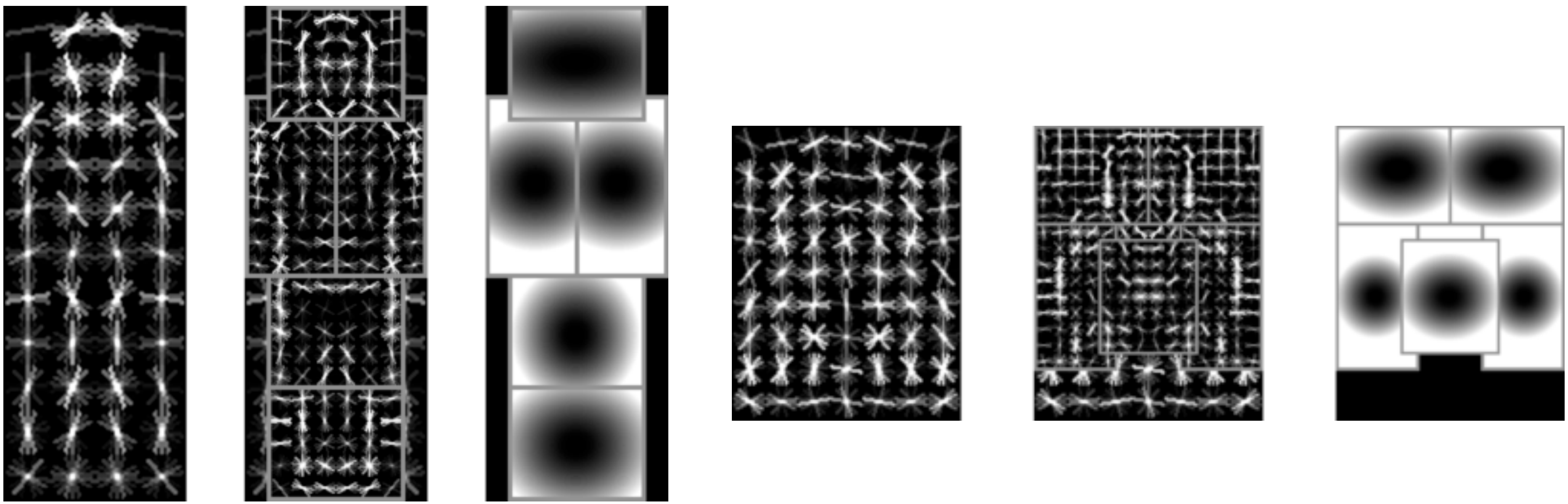




P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, [Object Detection with Discriminatively Trained Part Based Models](), PAMI 32(9), 2010

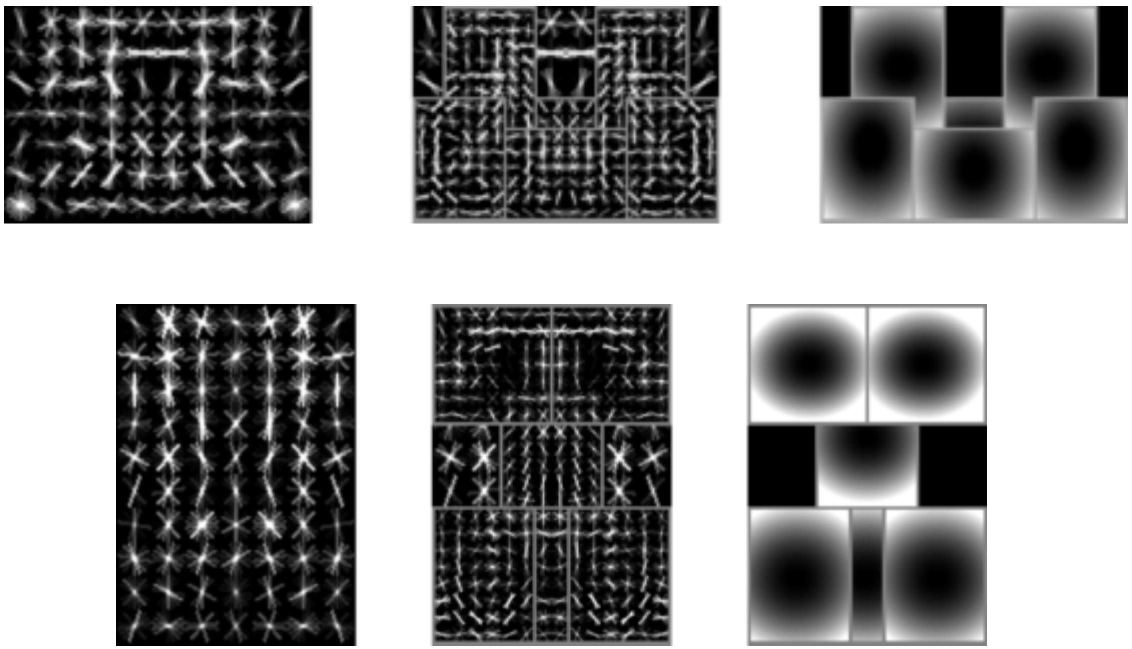
Discriminative part-based models

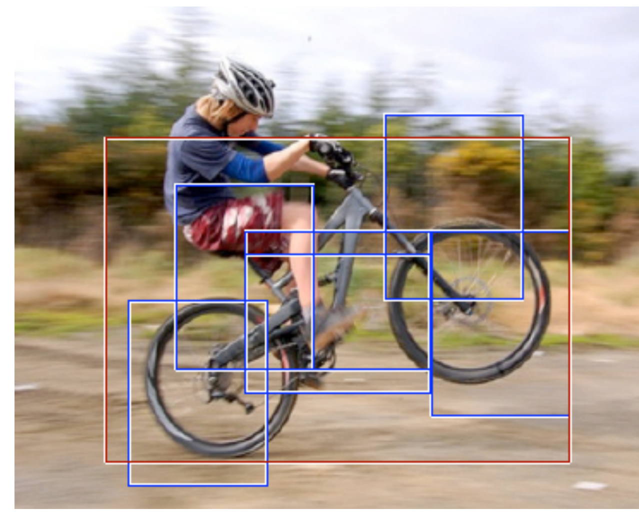
Multiple components



P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, [Object Detection with Discriminatively Trained Part Based Models](), PAMI 32(9), 2010

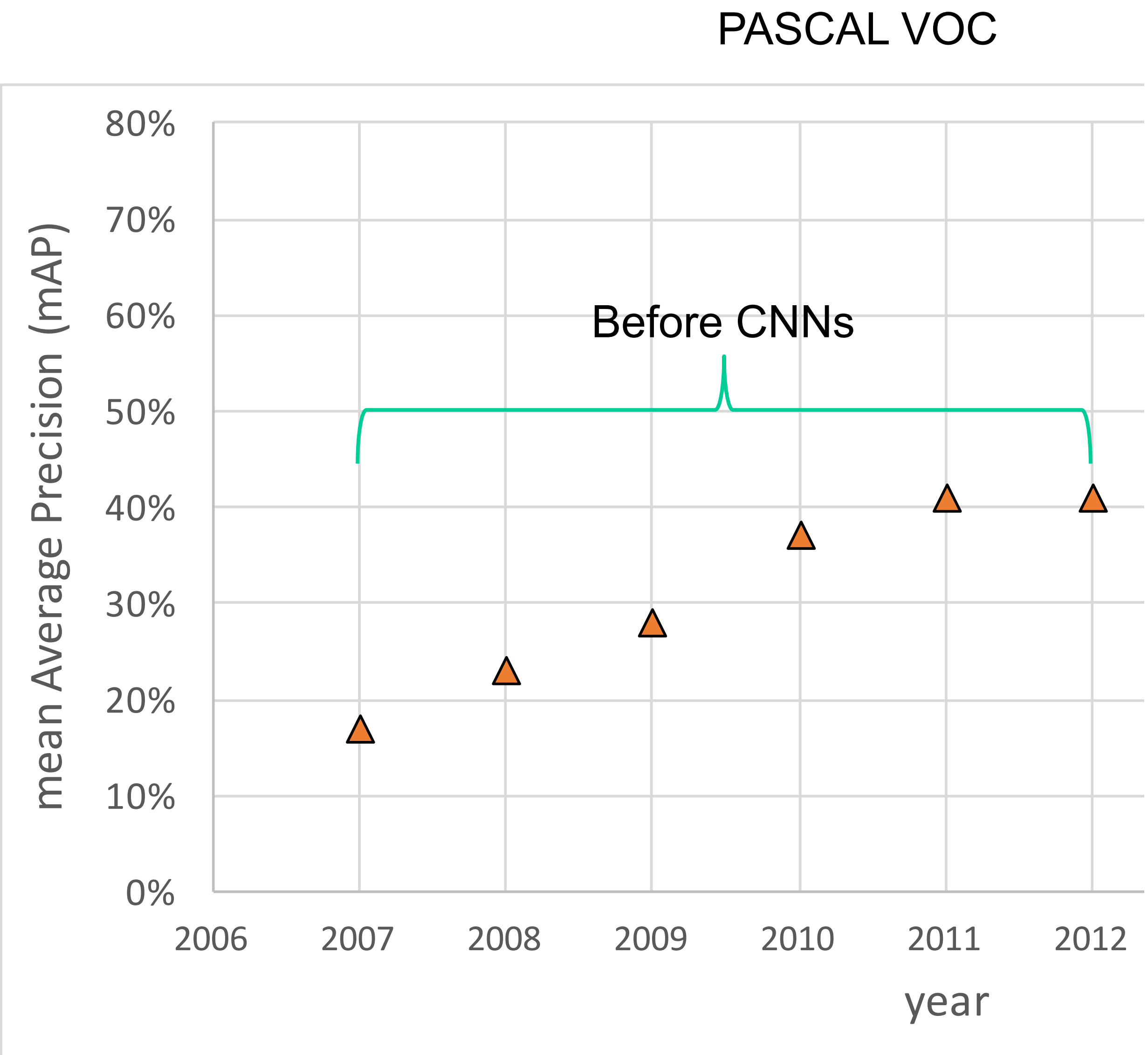
Discriminative part-based models



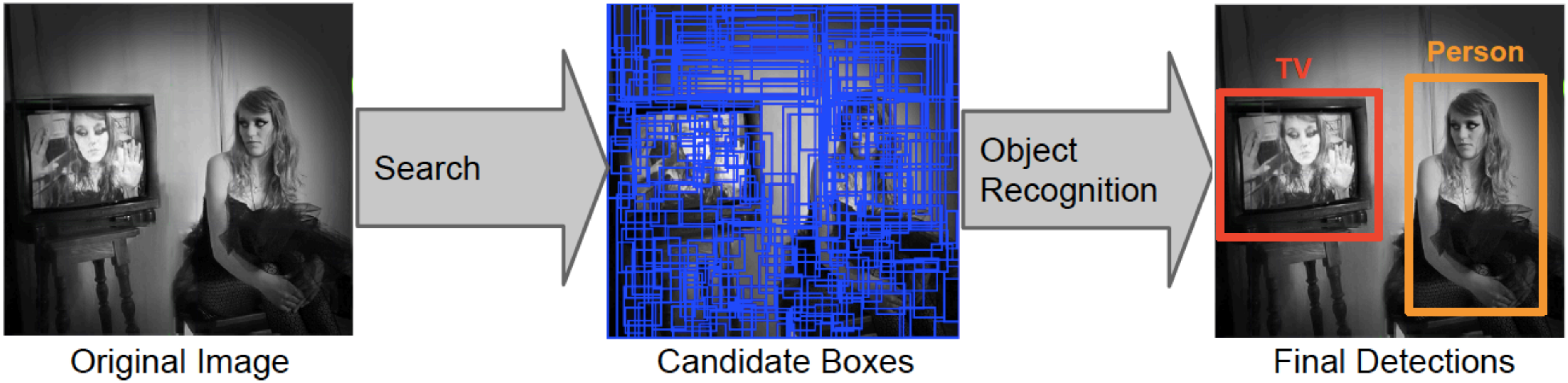


P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, [Object Detection with Discriminatively Trained Part Based Models](), PAMI 32(9), 2010

Progress on PASCAL detection



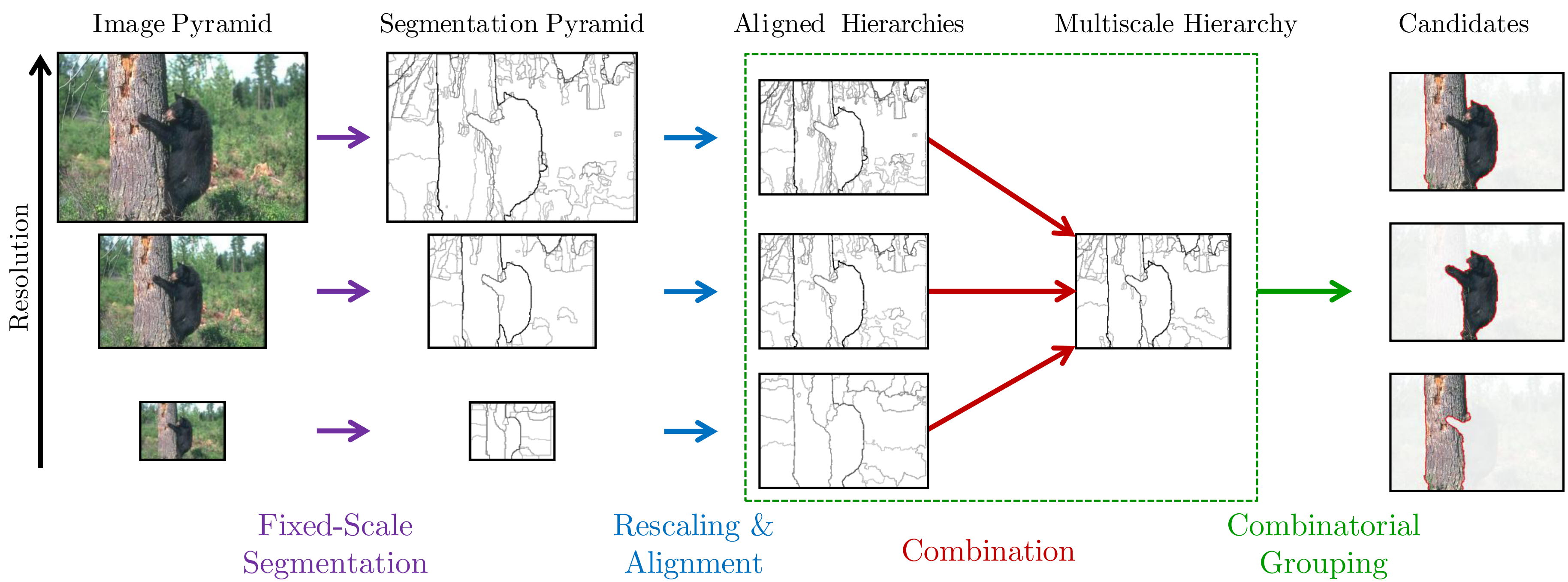
Conceptual approach: Proposal-driven detection



- Generate and evaluate a few hundred *region proposals*
  - Proposal mechanism can take advantage of low-level *perceptual organization* cues
  - Proposal mechanism can be category-specific or category-independent, hand-crafted or trained
  - Classifier can be slower but more powerful

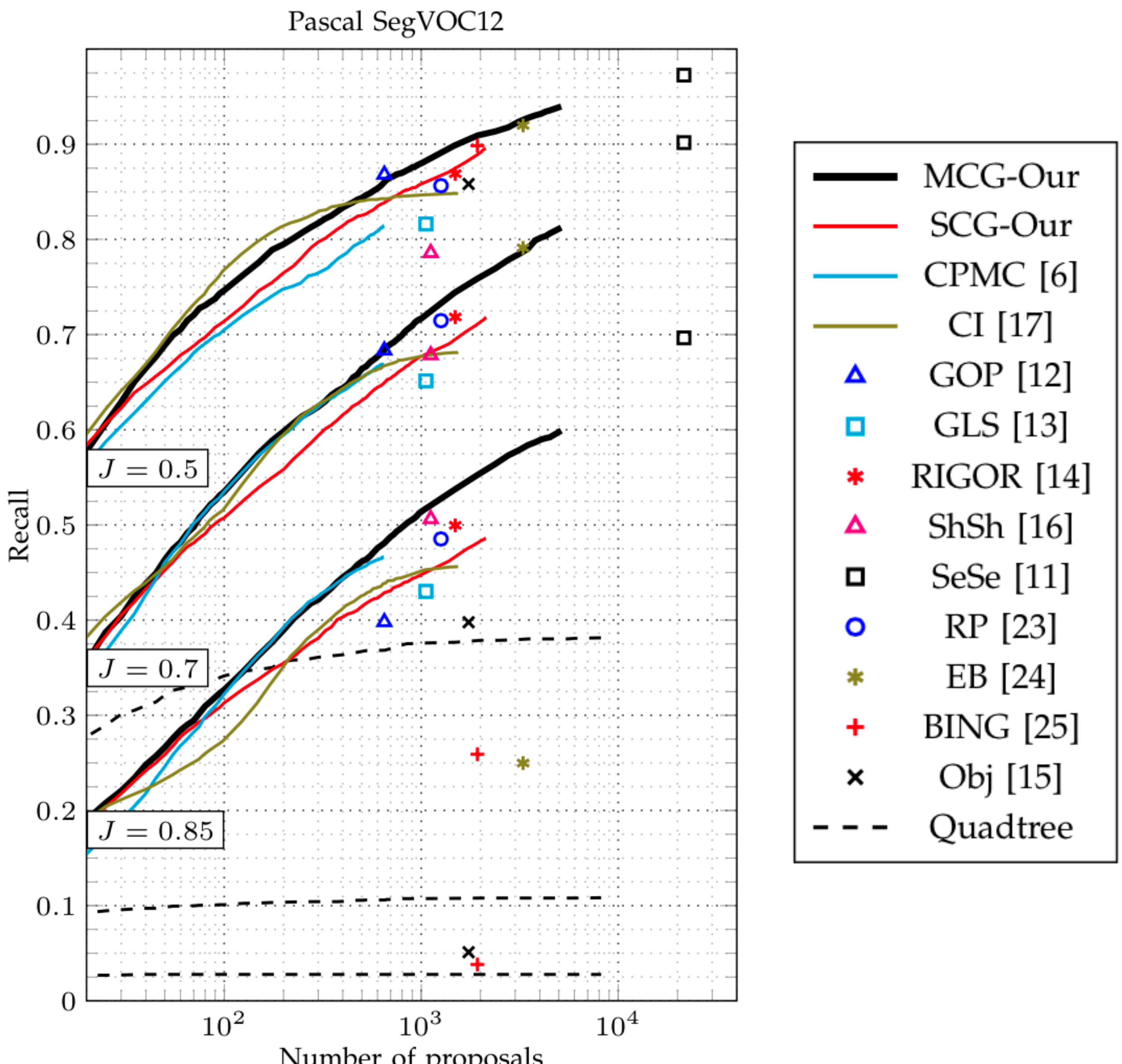
Multiscale Combinatorial Grouping

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



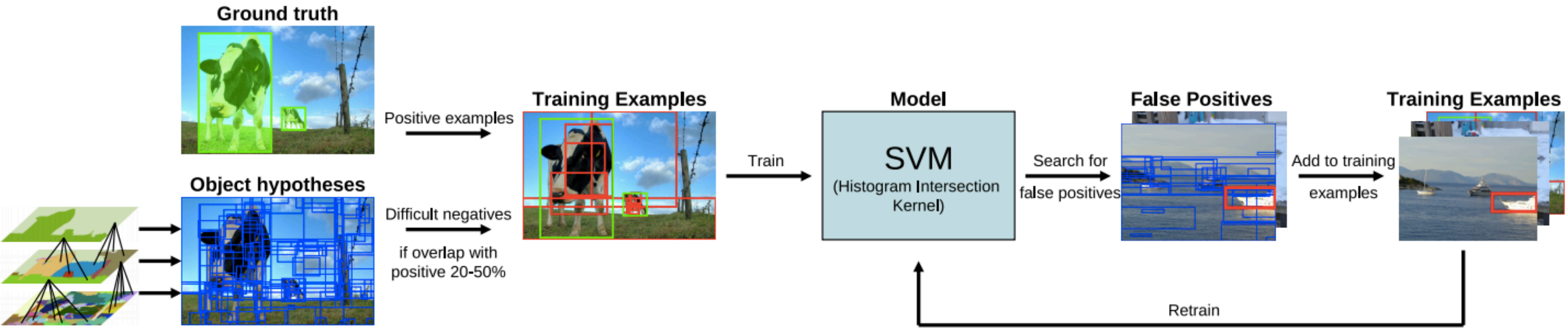
P. Arbelaez. et al., [Multiscale Combinatorial Grouping](), CVPR 2014

Region Proposals for Detection (Eval)



P. Arbelaez. et al., [Multiscale Combinatorial Grouping](), CVPR 2014

Region Proposals for Detection

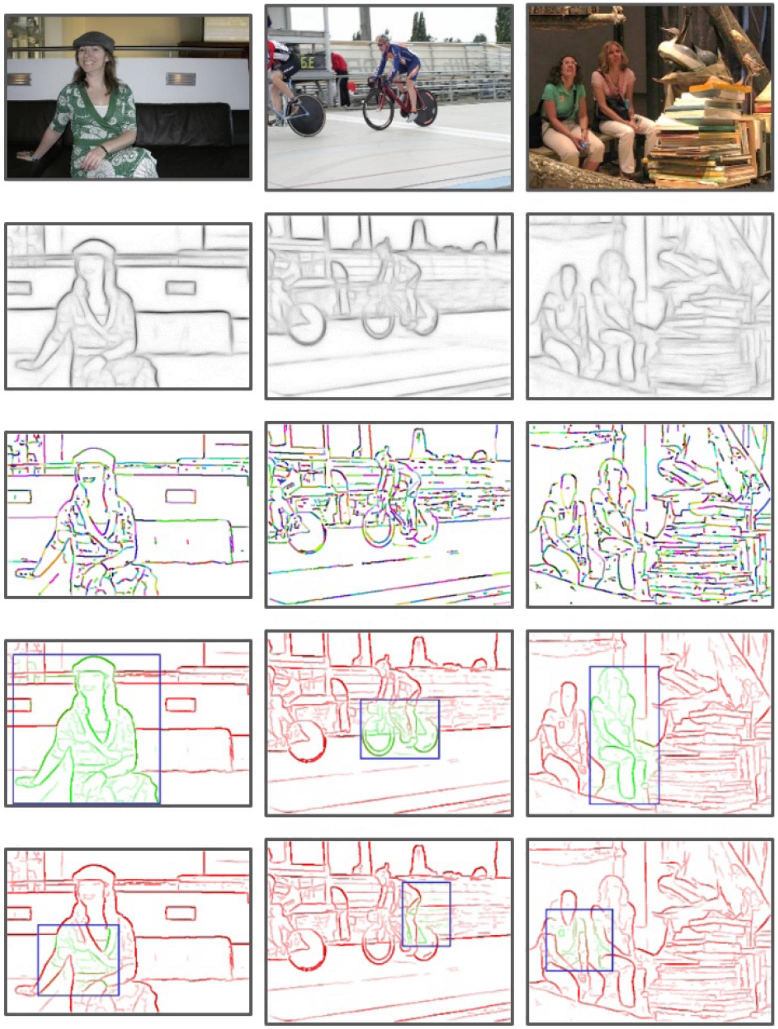


- Feature extraction: color SIFT, codebook of size 4K, spatial pyramid with four levels = 360K dimensions

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013

Another proposal method: EdgeBoxes

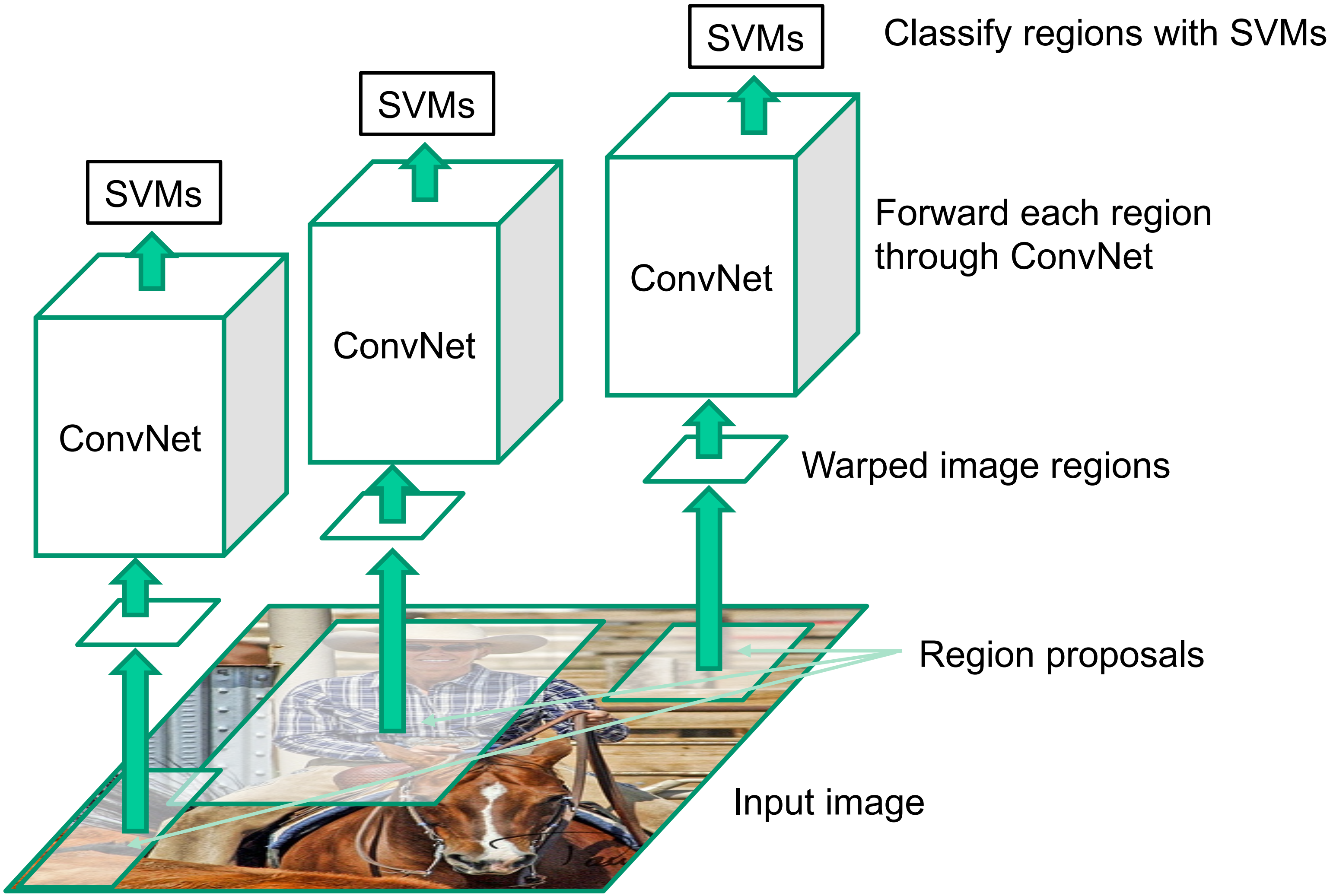
- Box score: number of edges in the box minus number of edges that overlap the box boundary
- Uses a trained edge detector
- Uses efficient data structures (incl. integral images) for fast evaluation
- Gets 75% recall with 800 boxes (vs. 1400 for Selective Search), is 40 times faster



C. Zitnick and P. Dollar, [Edge Boxes: Locating Object Proposals from Edges](), ECCV 2014

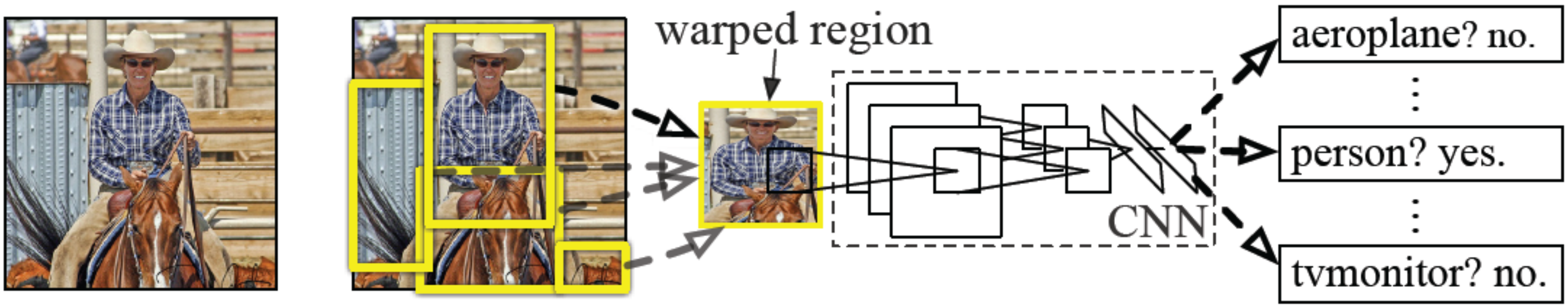
R-CNN: Region proposals + CNN features

Source: R. Girshick



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

R-CNN details

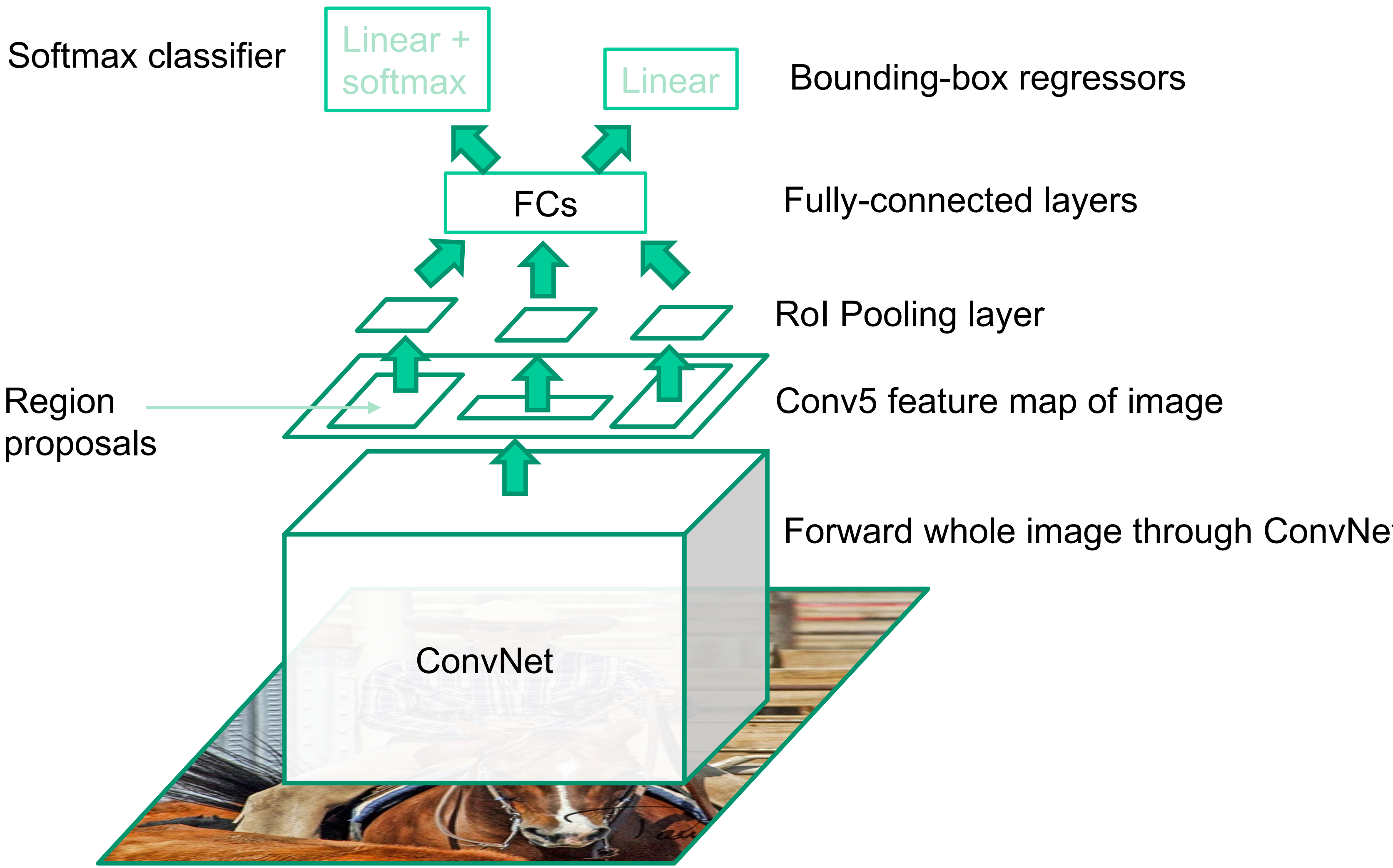


- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to 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 pros and cons

- Pros
  - Accurate!
  - 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)
    - Train post-hoc bounding-box regressions (least squares)
  - Training is slow (84h), takes a lot of disk space
    - 2000 CNN passes per image
  - Inference (detection) is slow (47s / image with VGG16)

Fast R-CNN

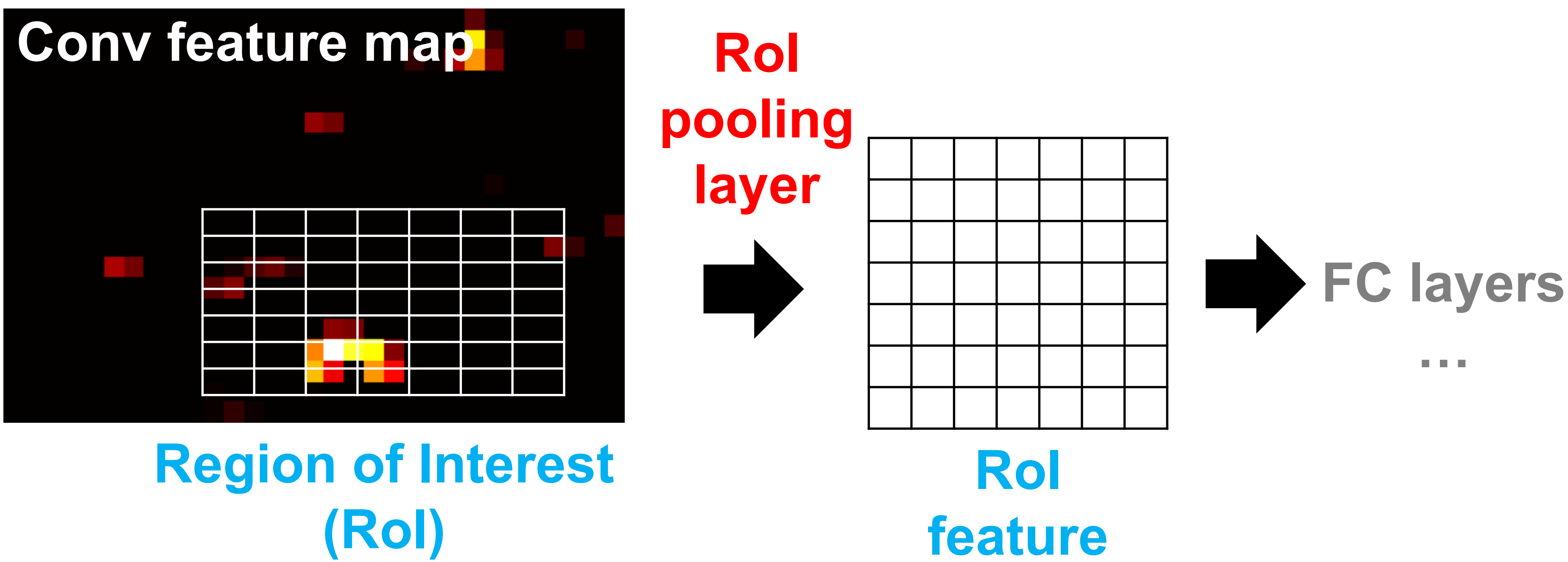


Source: R. Girshick

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
  - Use nearest-neighbor interpolation of coordinates, max pooling



Source: R. Girshick, K. He

RoI pooling illustration

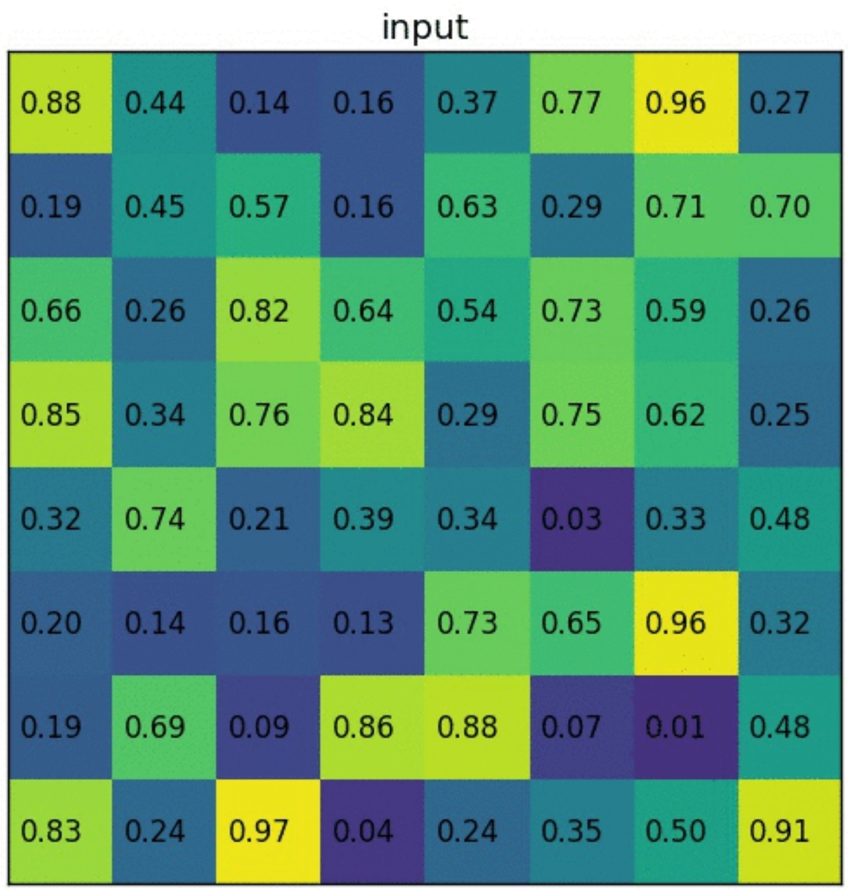
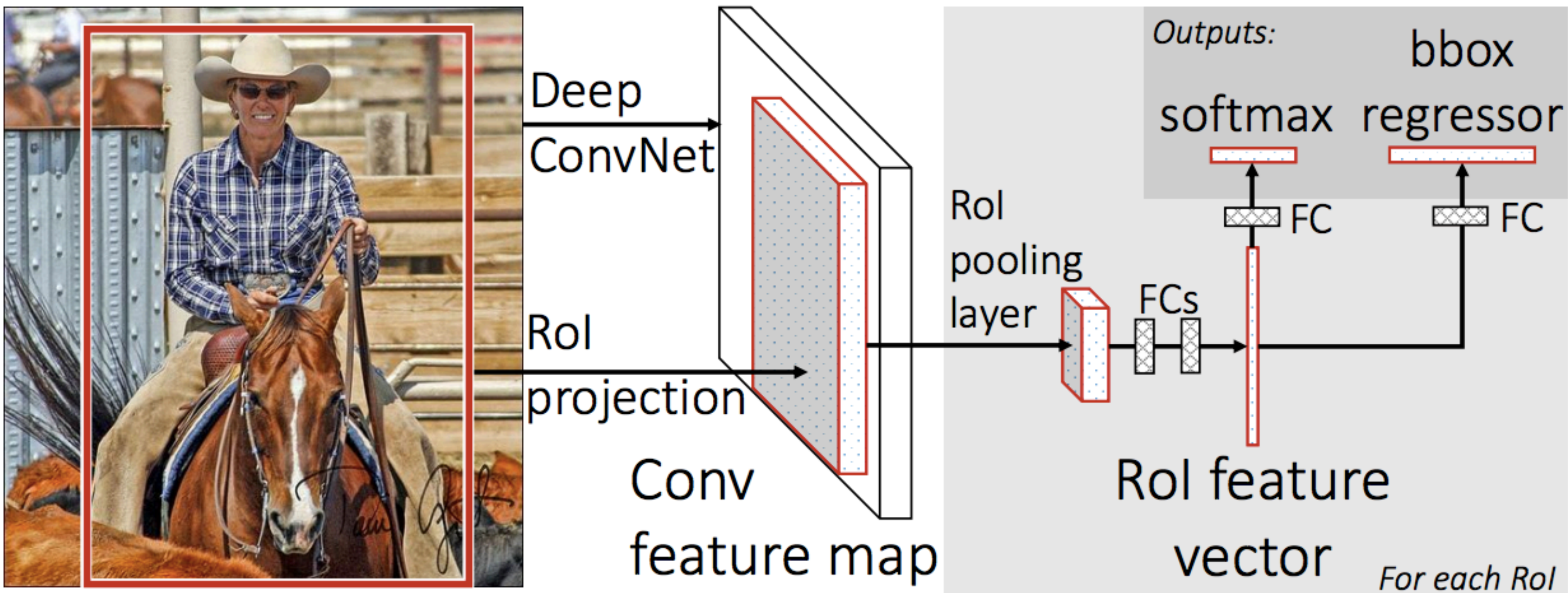


Image source

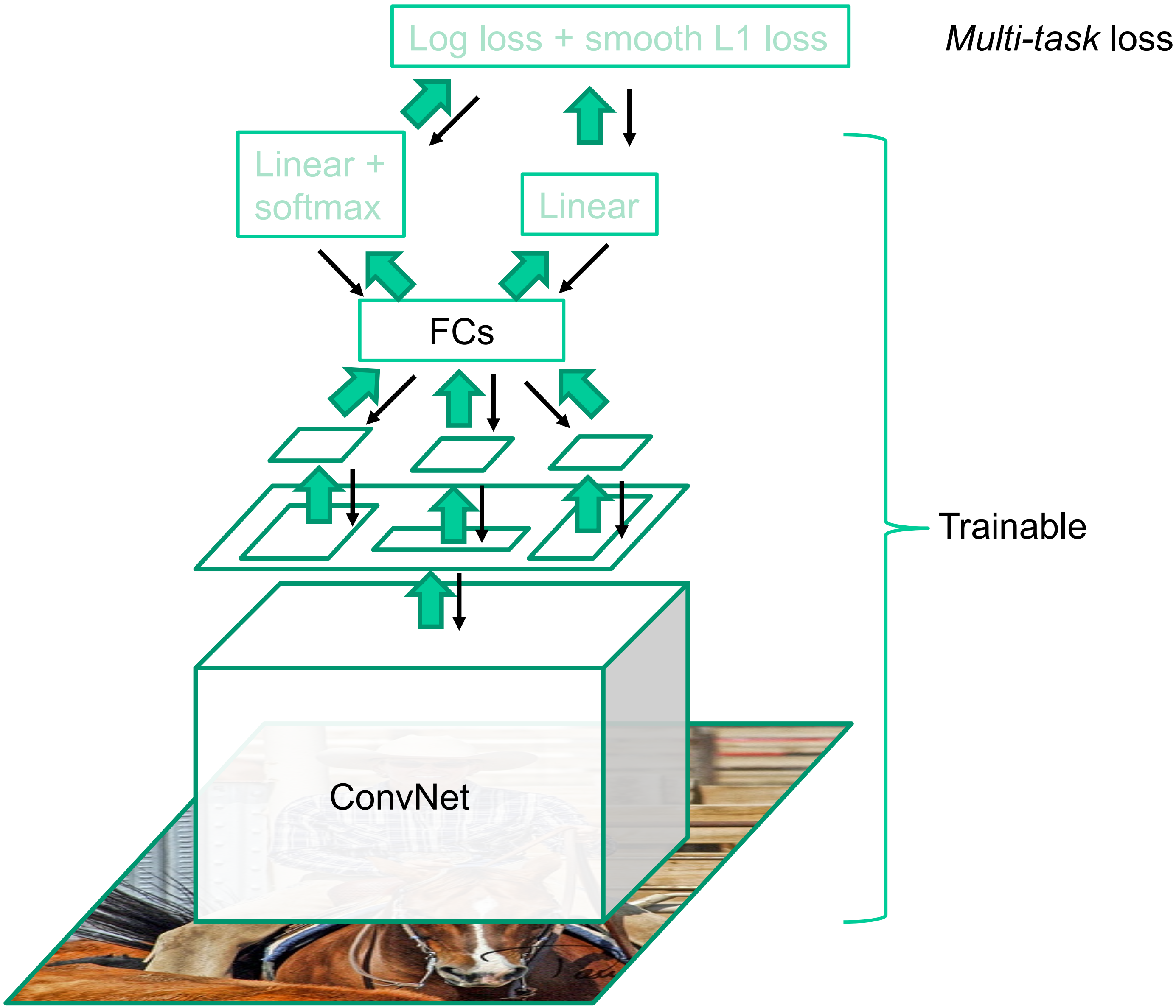
Prediction

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



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

Fast R-CNN training



Source: R. Girshick

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

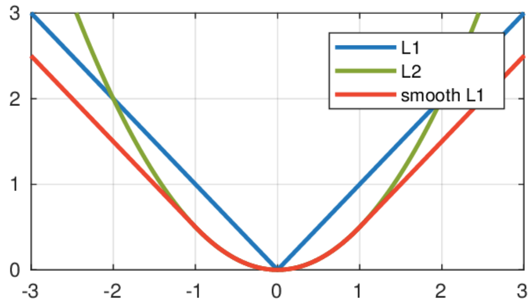
Multi-task loss

- Loss for ground truth class y, predicted class probabilities $P(y)$, ground truth box b, and predicted box $\hat{b}$:

$$L(y, P, b, \hat{b}) = \underbrace{-\log P(y)}_{\text{softmax loss}} + \lambda \mathbb{I}[y \geq 1] \underbrace{L_{\text{reg}}(b, \hat{b})}_{\text{regression loss}}$$

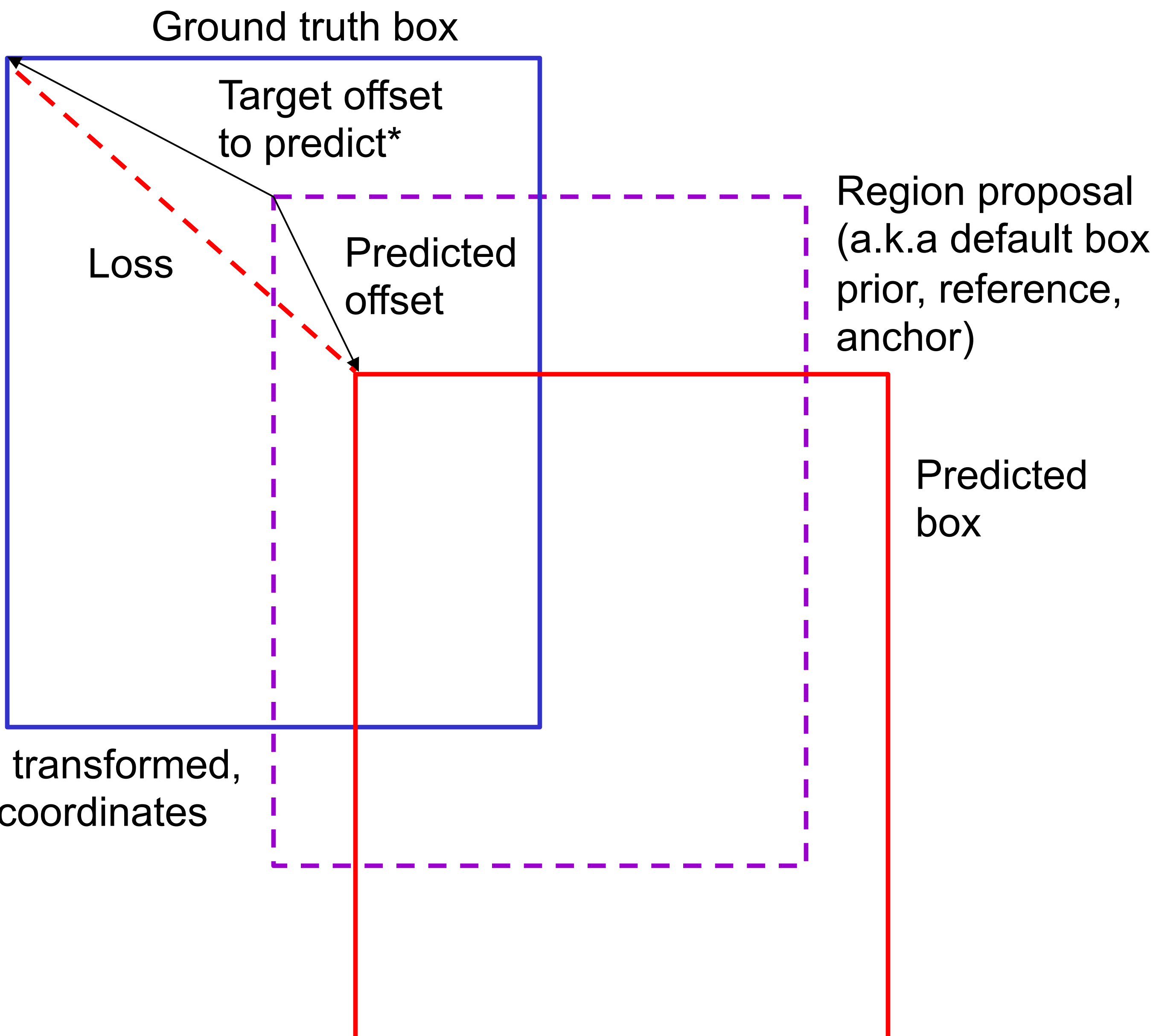
- Regression loss: *smooth L1 loss* on top of log space offsets relative to proposal

$$L_{\text{reg}}(b, \hat{b}) = \sum_{i=\{x,y,w,h\}} \text{smooth}_{L_1}(b_i - \hat{b}_i)$$



$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Bounding box regression



*Typically in transformed, normalized coordinates

Fast R-CNN results

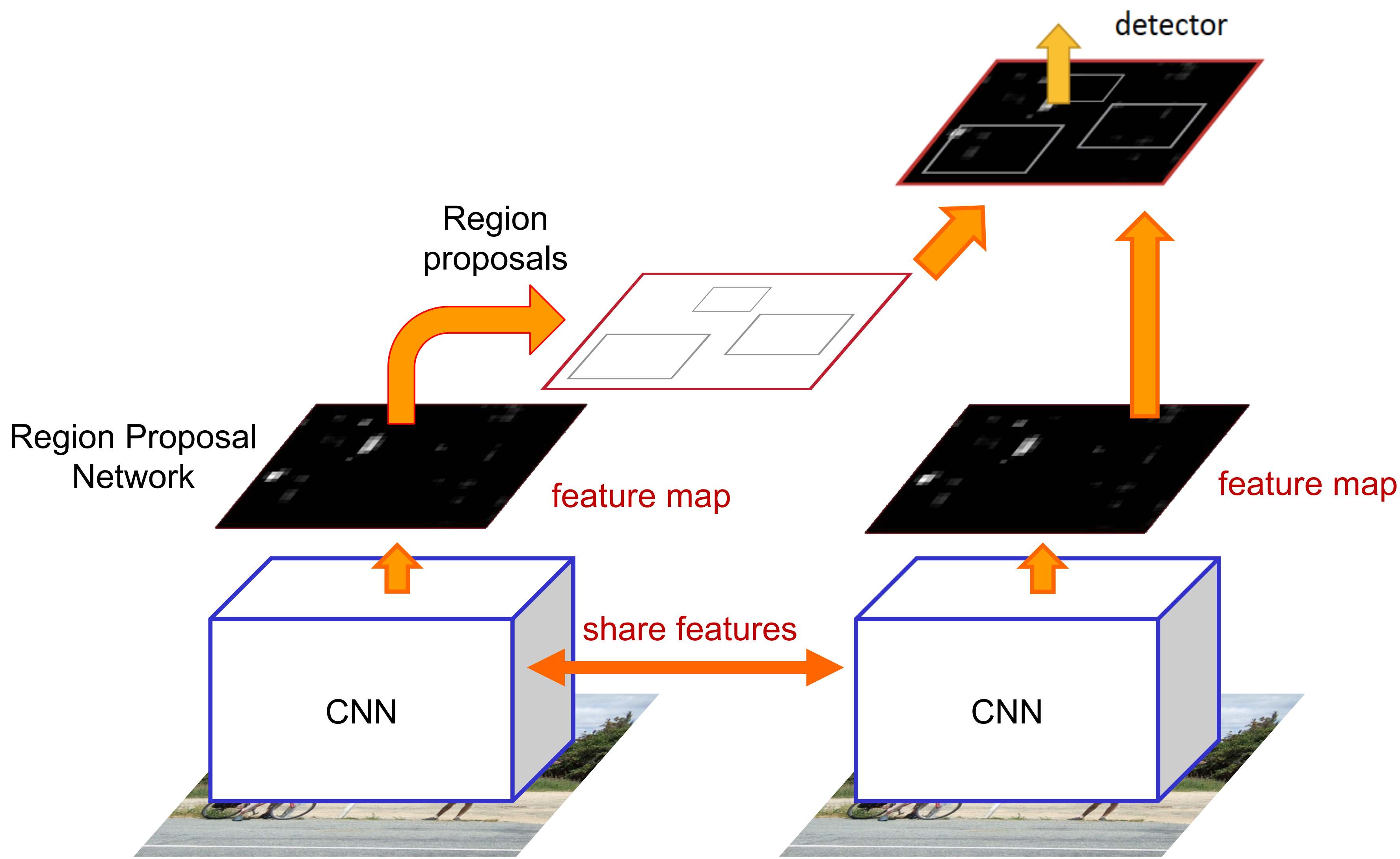
	Fast R-CNN	**R-CNN**
Train time (h)	**9.5**	84
- Speedup	**8.8x**	1x
Test time / image	**0.32s**	47.0s
Test speedup	**146x**	1x
mAP	**66.9%**	66.0%

(vs. 53.7% for AlexNet)

Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

Source: R. Girshick

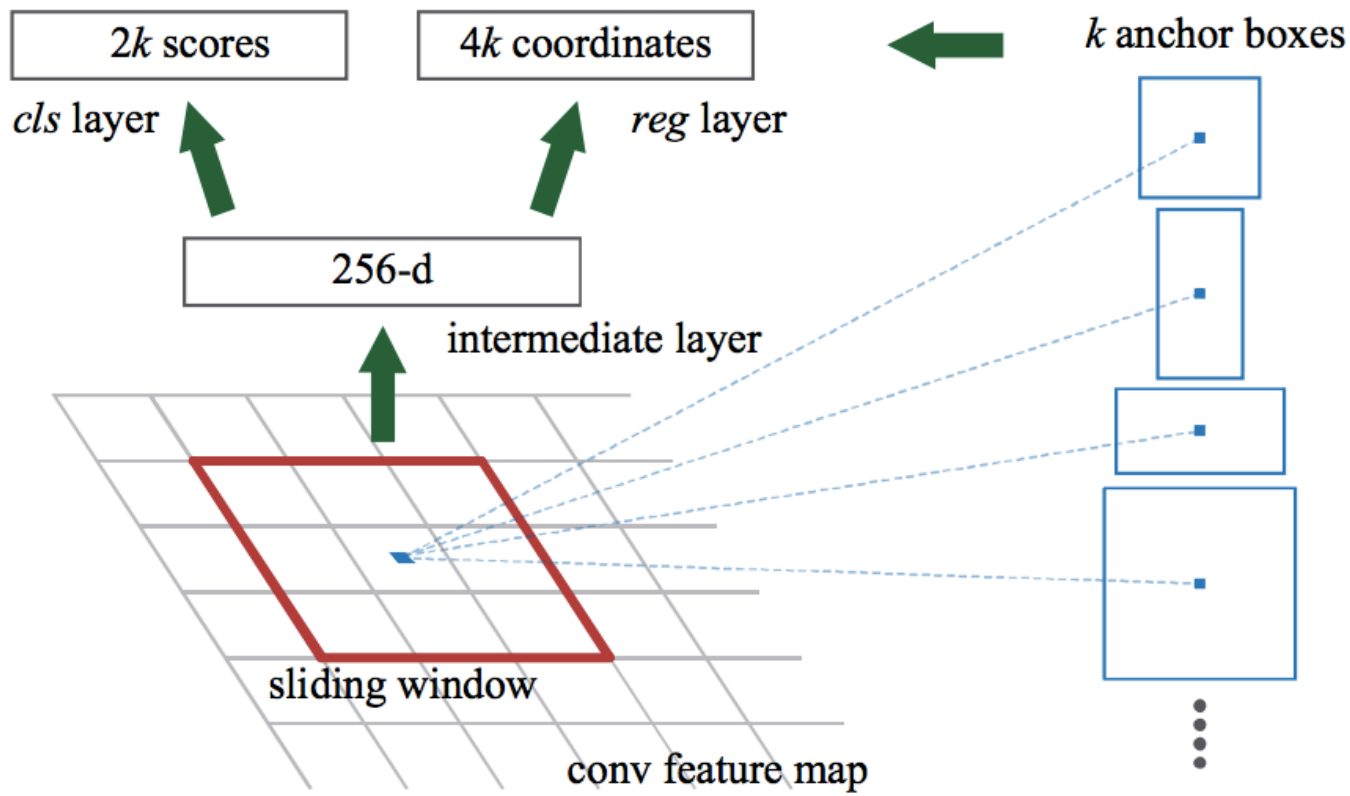
Faster R-CNN



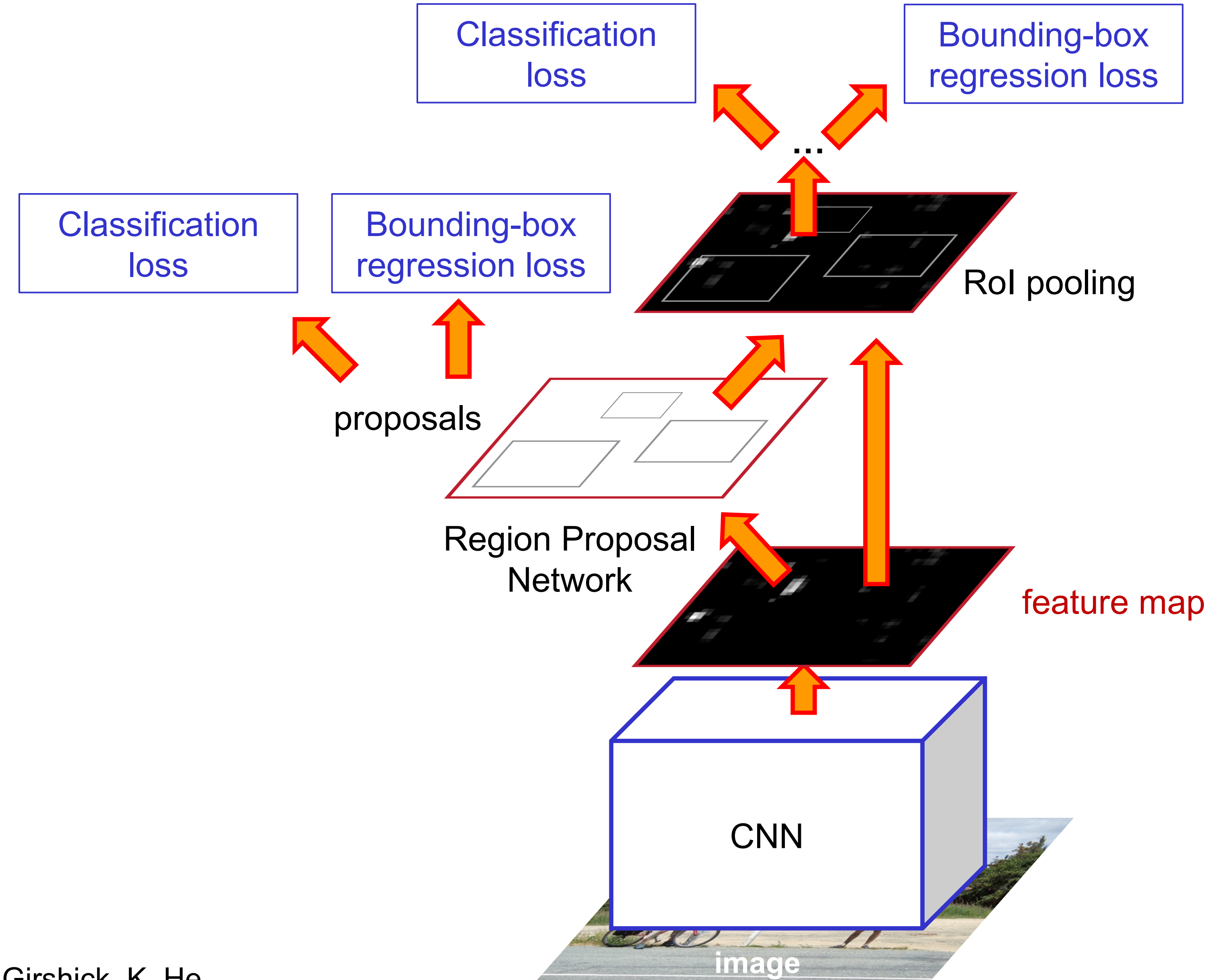
S. Ren, K. He, R. Girshick, and J. Sun, [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks], NIPS 2015

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)



One network, four losses



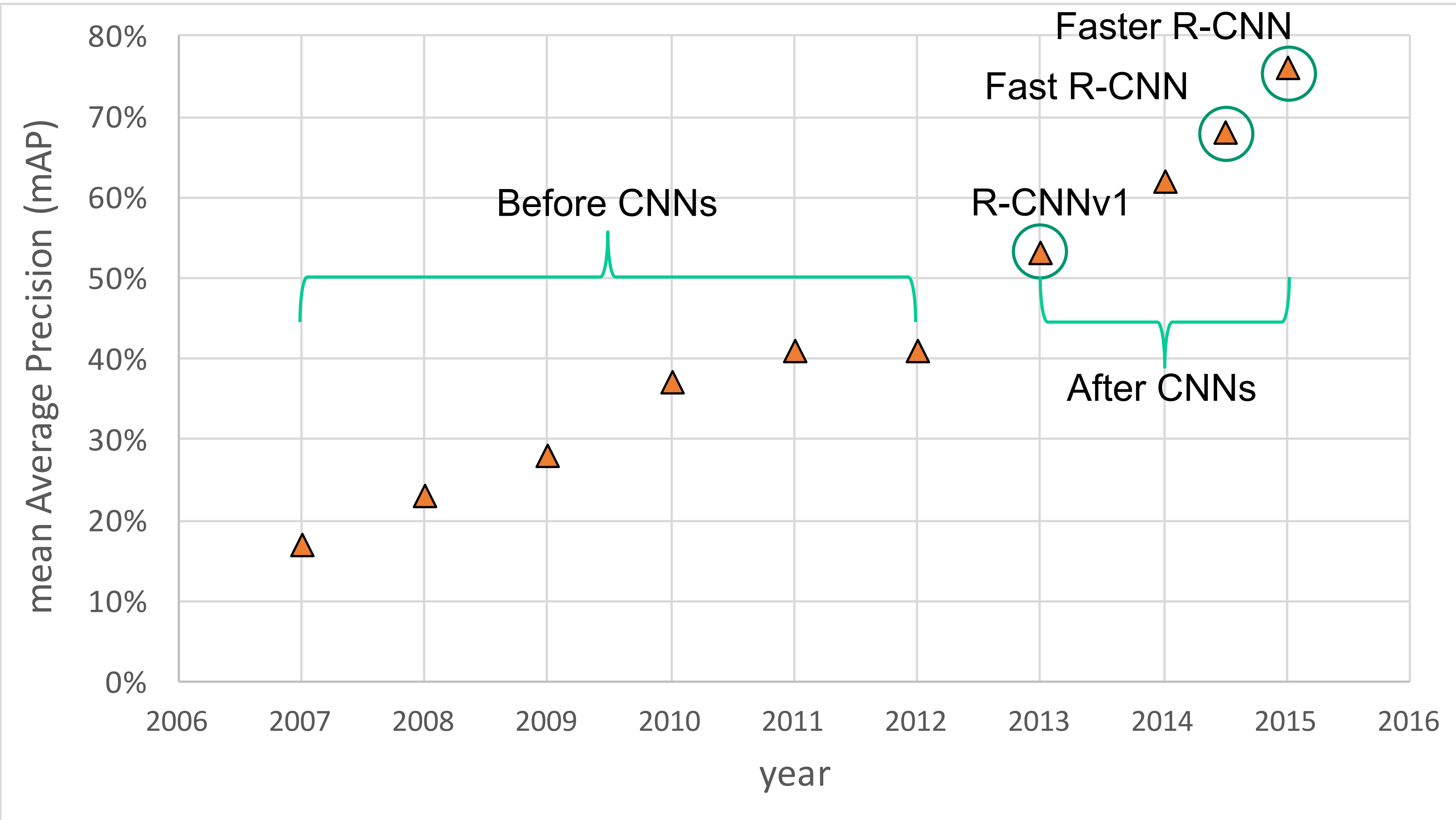
Source: R. Girshick, K. He

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**

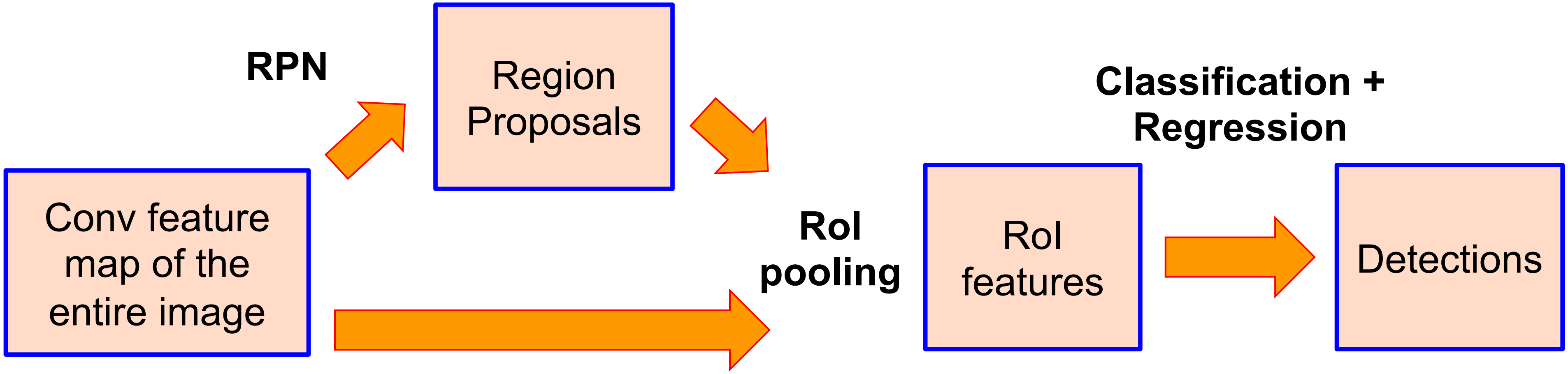
detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

Object detection progress



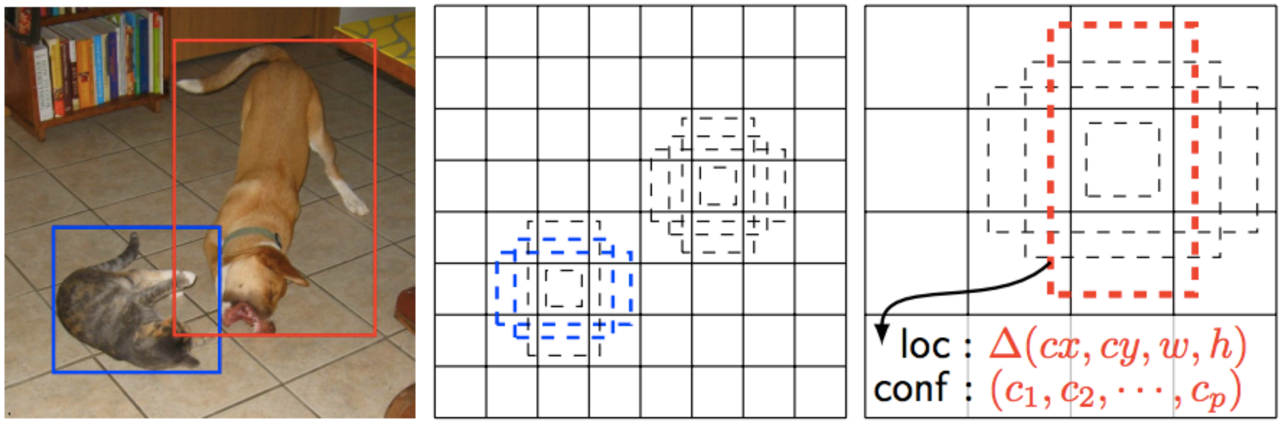
Streamlined detection architectures

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



- Is it possible do detection in one shot?

SSD

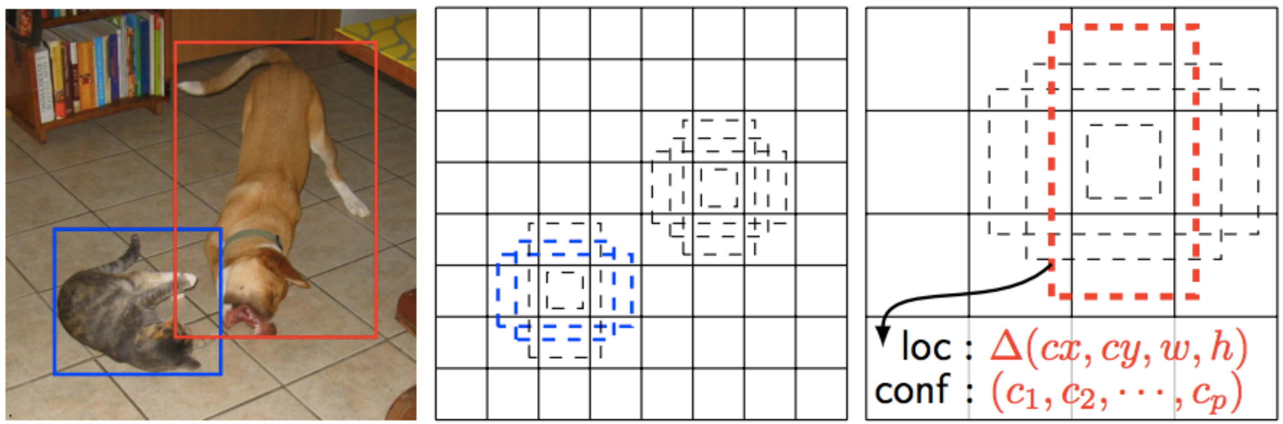


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

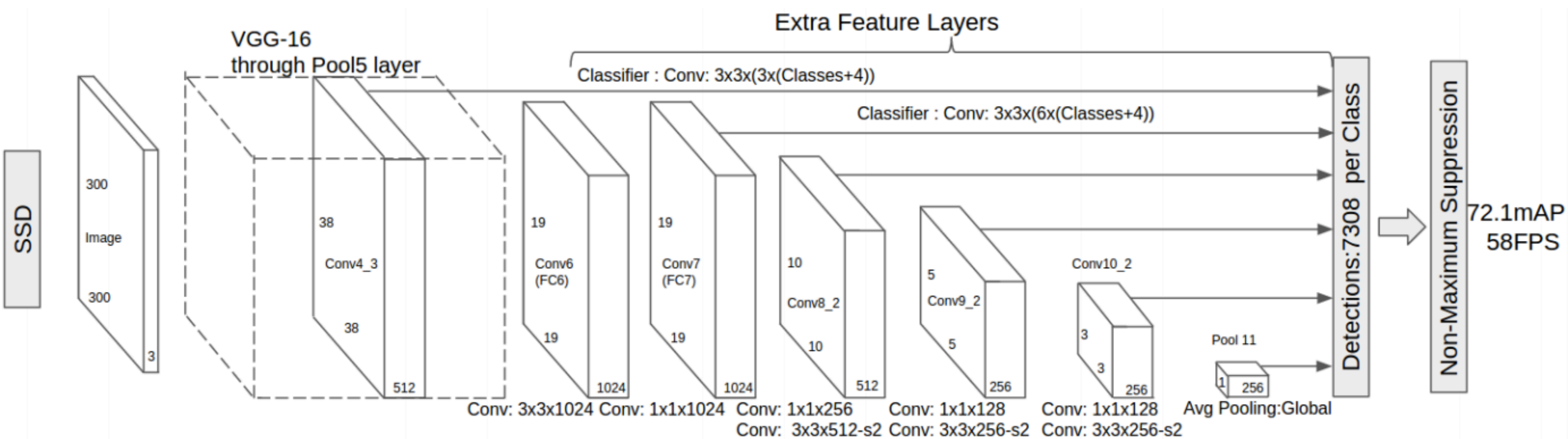
- Similarly to RPN, use anchors and directly predict class-specific bounding boxes.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, [SSD: Single Shot MultiBox Detector](), ECCV 2016.

SSD



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map



W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, [SSD: Single Shot MultiBox Detector](), ECCV 2016.

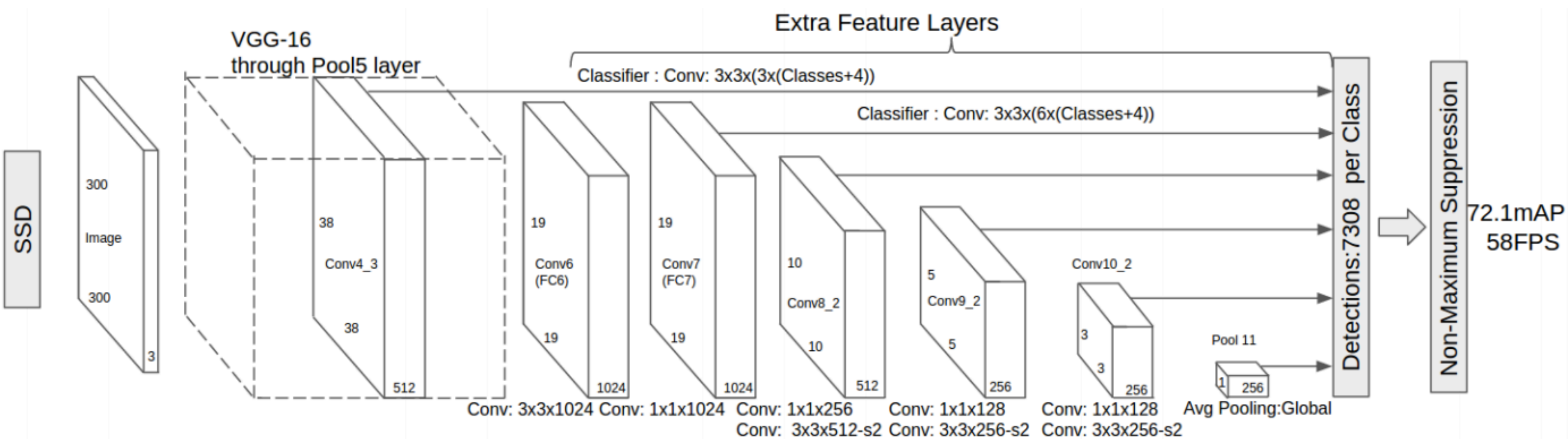
SSD: Results (PASCAL 2007)

- More accurate *and* faster than YOLO and Faster R-CNN

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

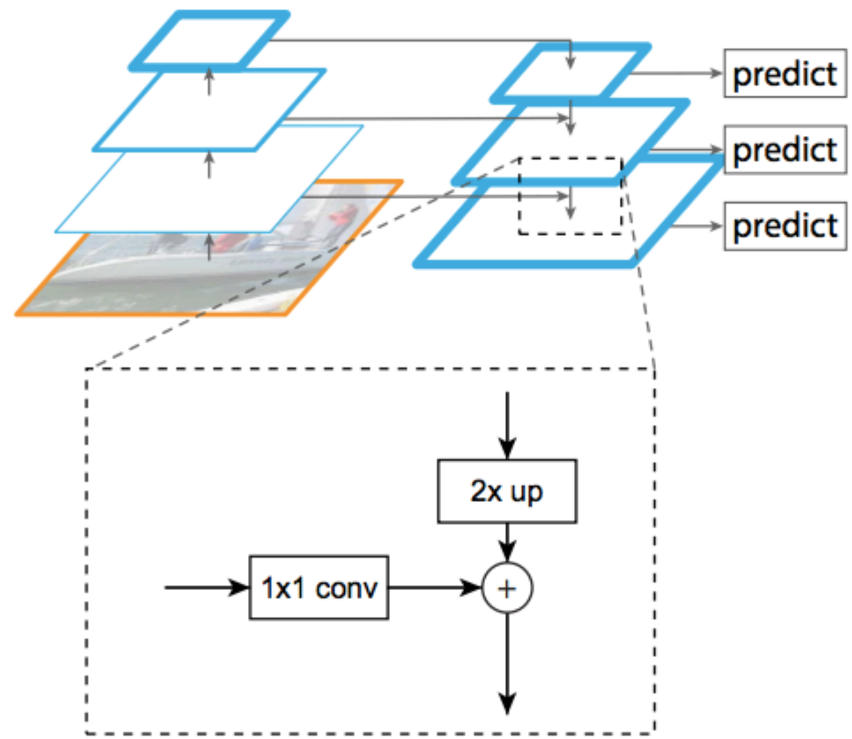
Multi-resolution prediction

- SSD predicts boxes of different size from different conv maps, but each level of resolution has its own predictors and higher-level context does not get propagated back to lower-level feature maps
- Can we have a more elegant multi-resolution prediction architecture?



Feature pyramid networks

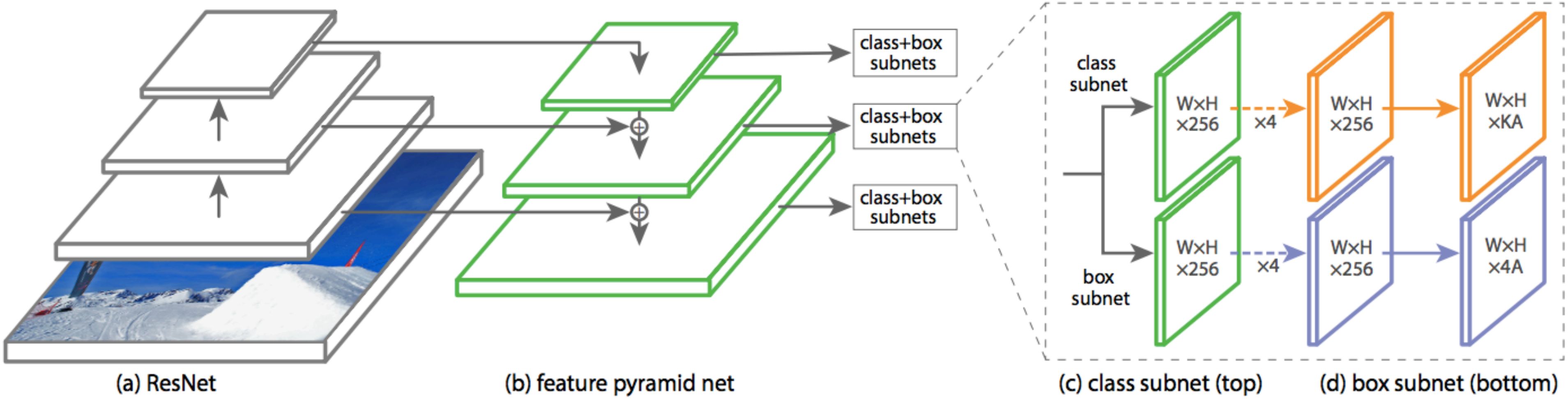
- Improve predictive power of lower-level feature maps by adding contextual information from higher-level feature maps
- Predict different sizes of bounding boxes from different levels of the pyramid (but share parameters of predictors)



T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object detection, CVPR 2017.

RetinaNet

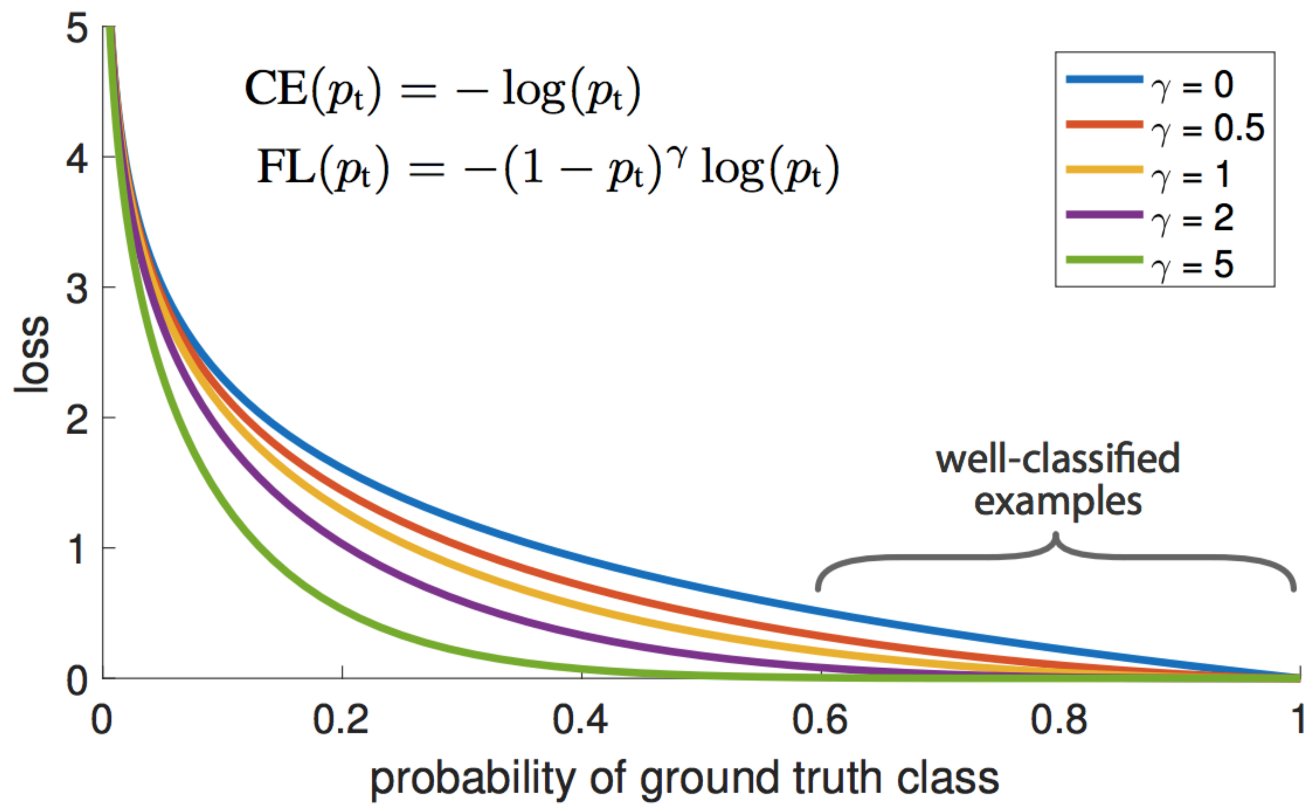
- Combine feature pyramid network with *focal loss* to reduce the standard cross-entropy loss for well-classified examples



T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, [Focal loss for dense object detection](), ICCV 2017.

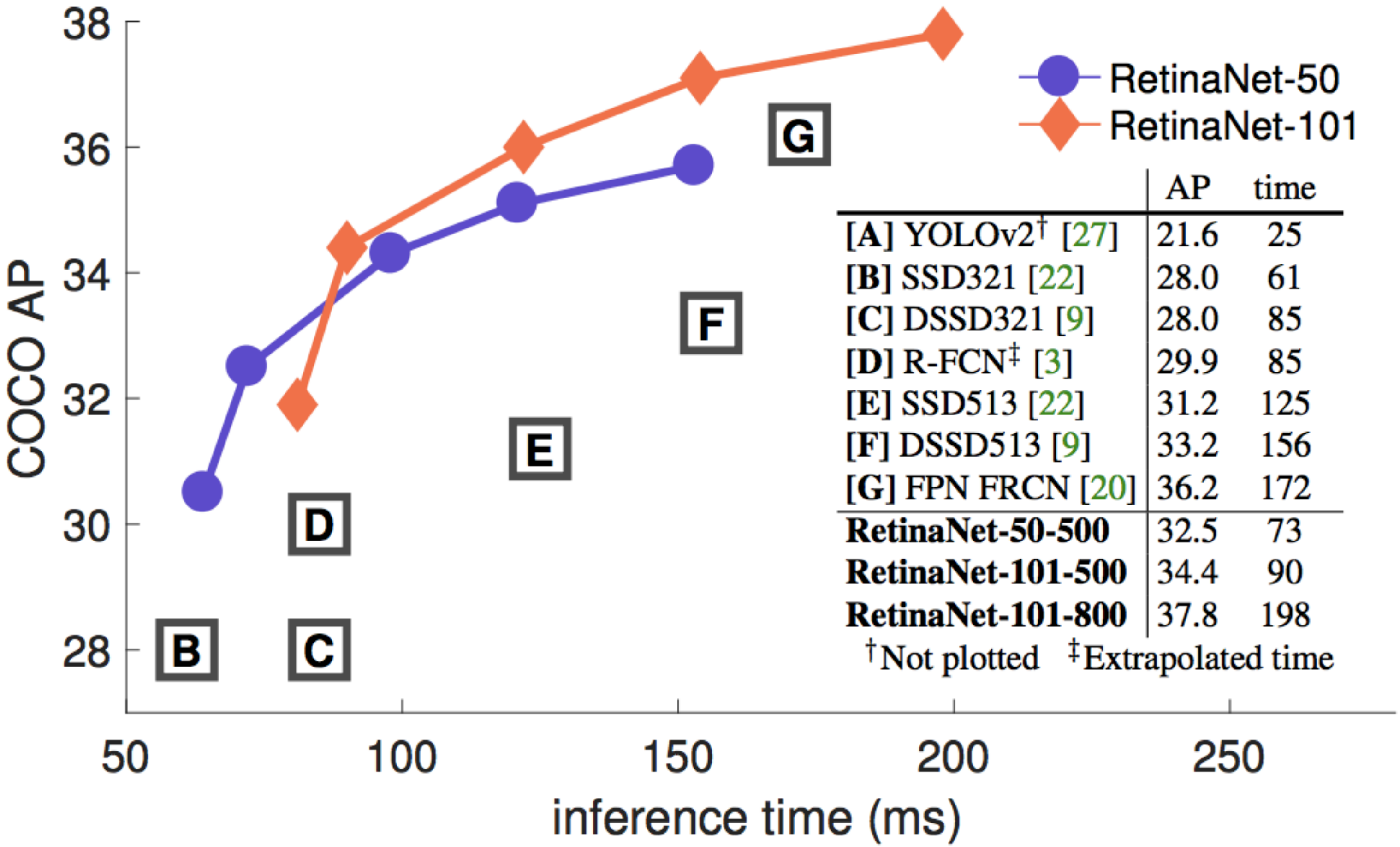
RetinaNet

- Combine feature pyramid network with *focal loss* to reduce the standard cross-entropy loss for well-classified examples



T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, [Focal loss for dense object detection](), ICCV 2017.

RetinaNet: Results



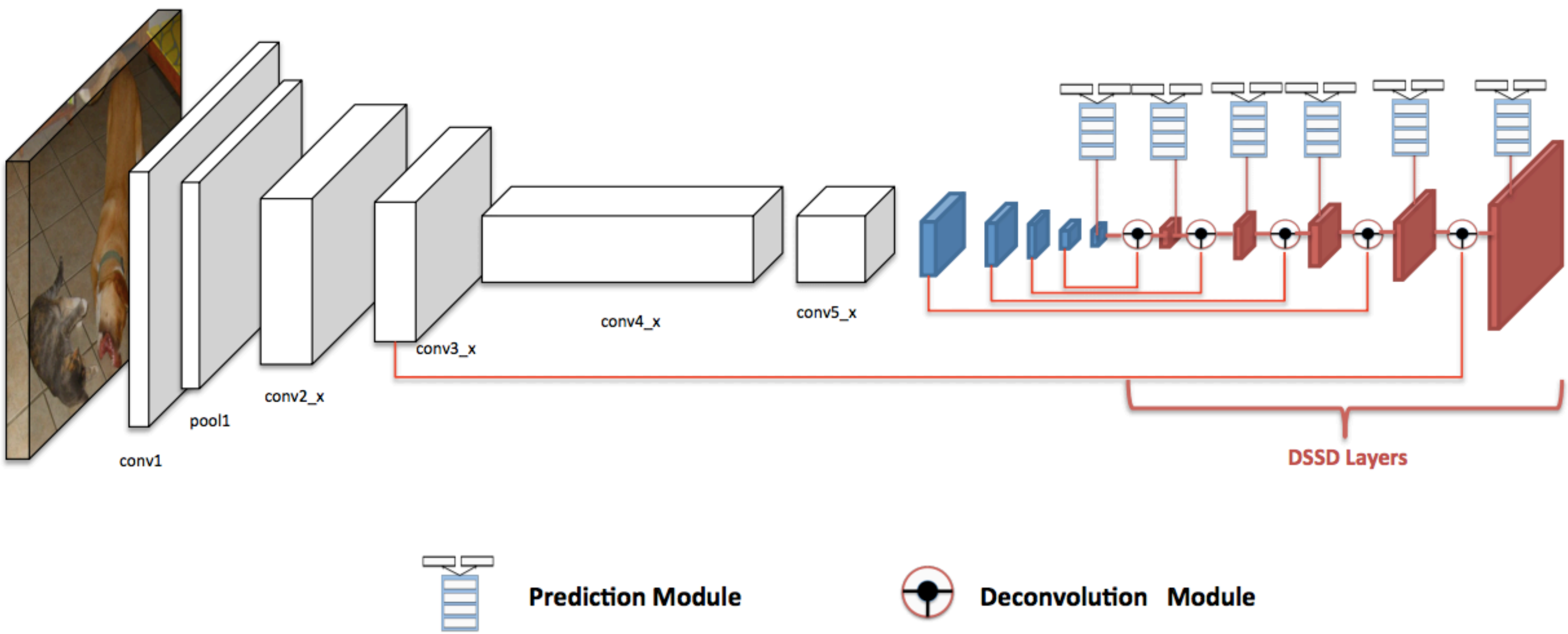
	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, [Focal loss for dense object detection](), ICCV 2017.

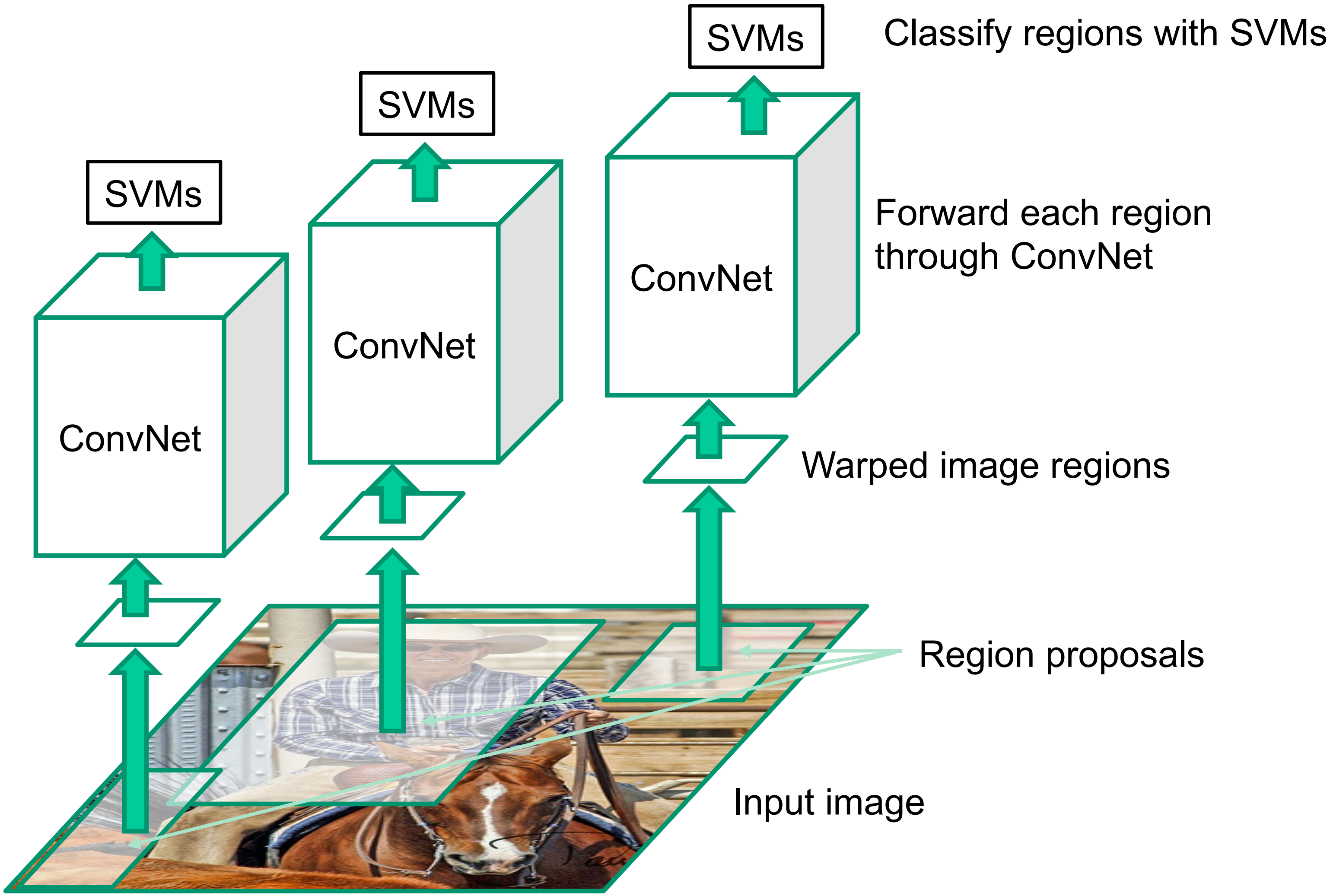
Deconvolutional SSD

- Improve performance of SSD by increasing resolution through learned “deconvolutional” layers



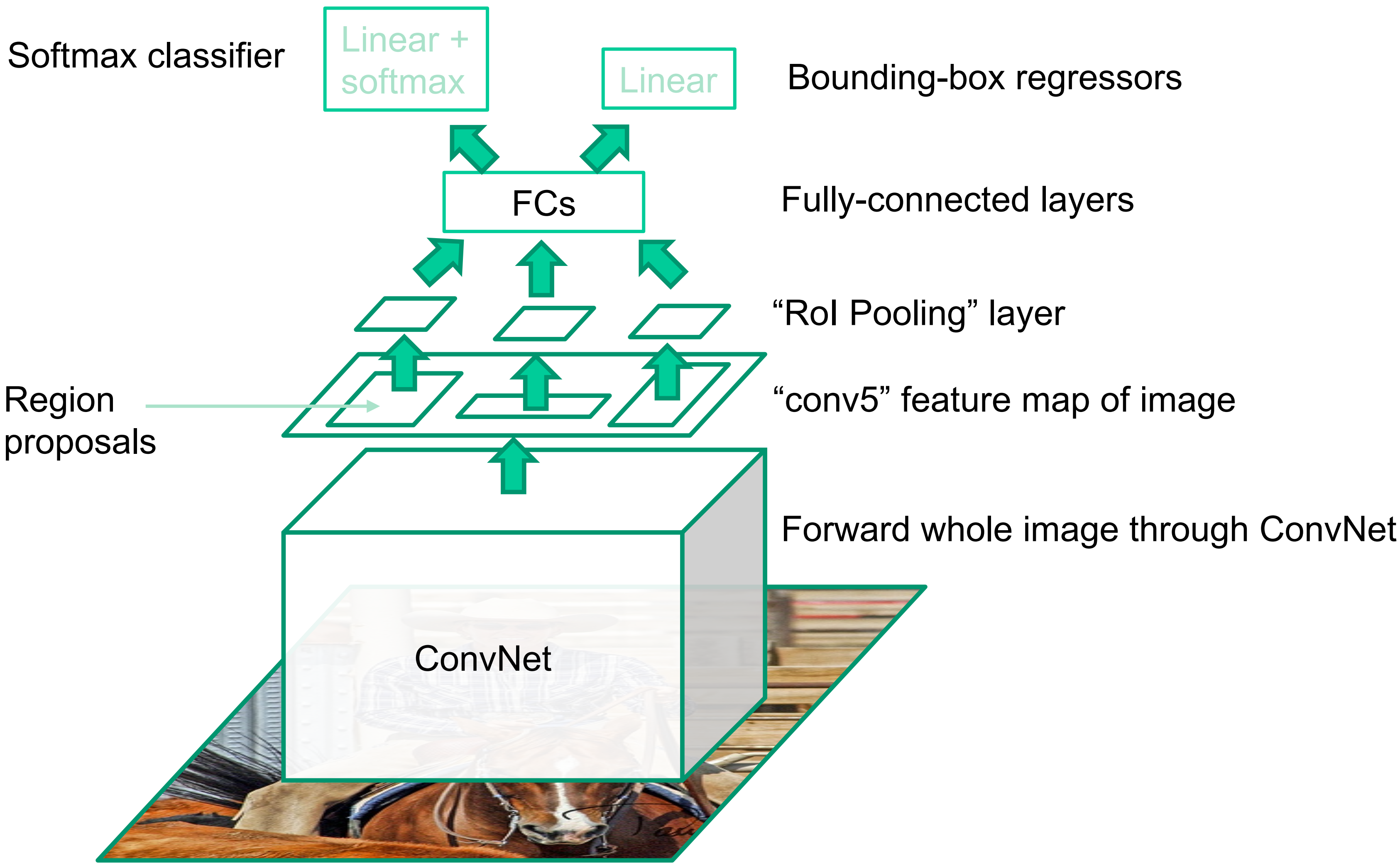
C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A. Berg, [DSSD: Deconvolutional single-shot detector](), arXiv 2017.

Review: R-CNN



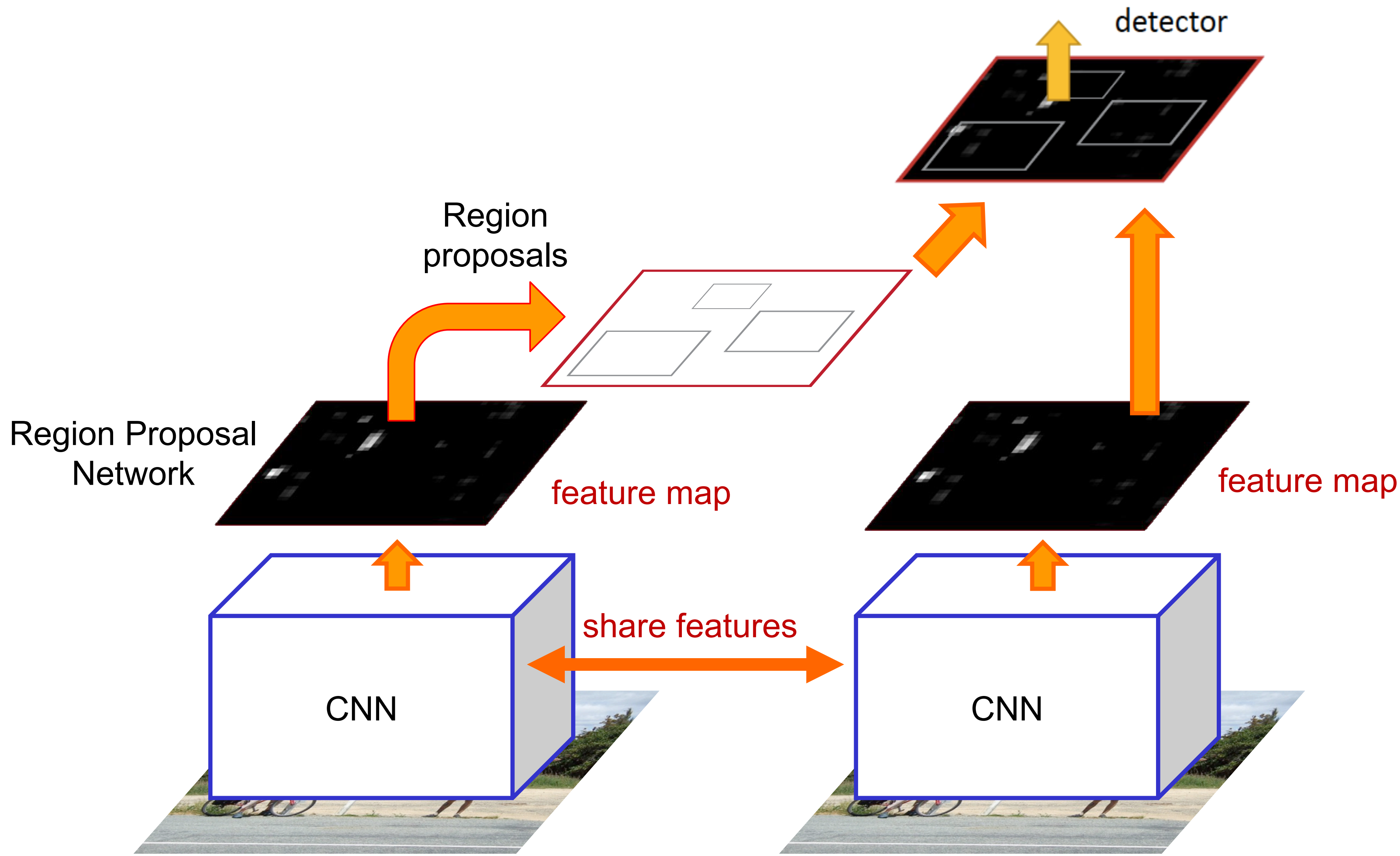
R. Girshick, J. Donahue, T. Darrell, and J. Malik, [Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation](), CVPR 2014.

Review: Fast R-CNN



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

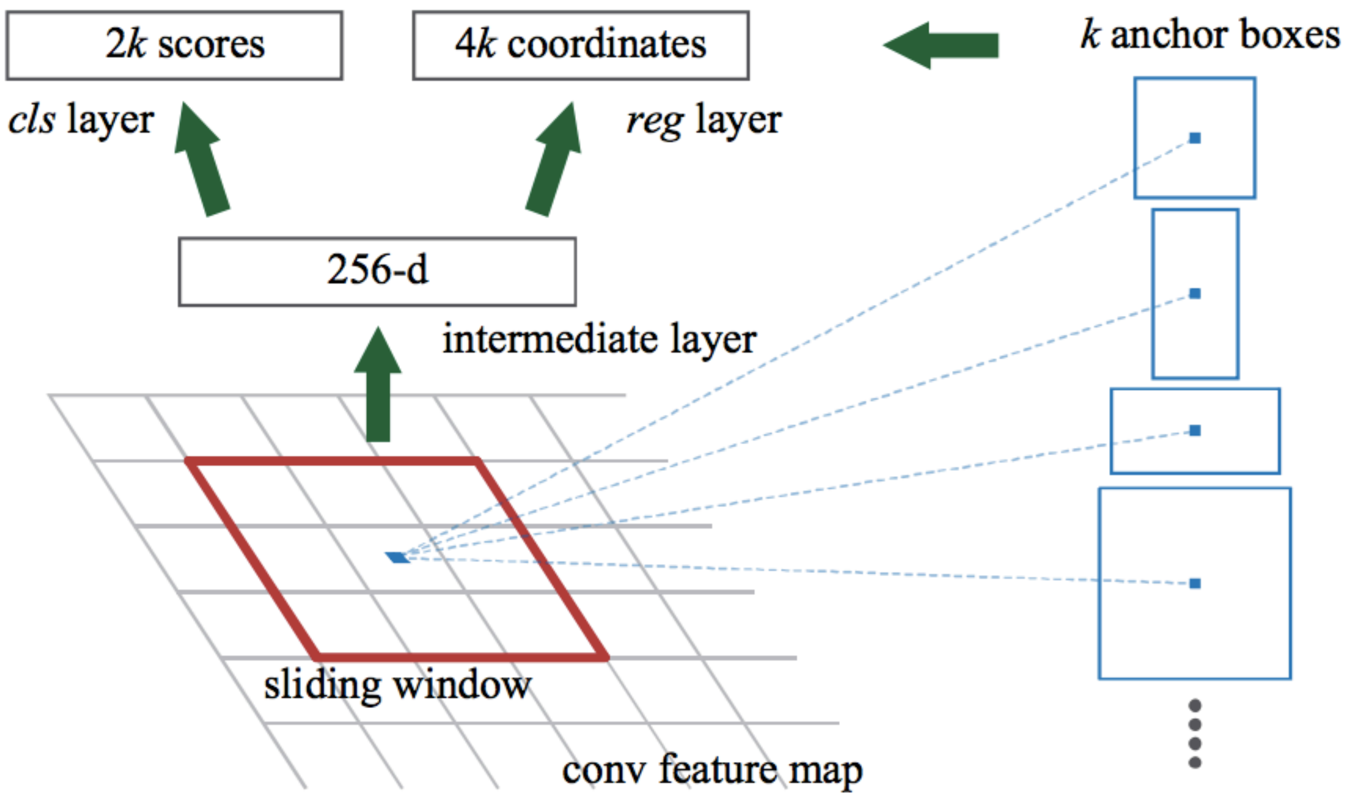
Review: Faster R-CNN



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

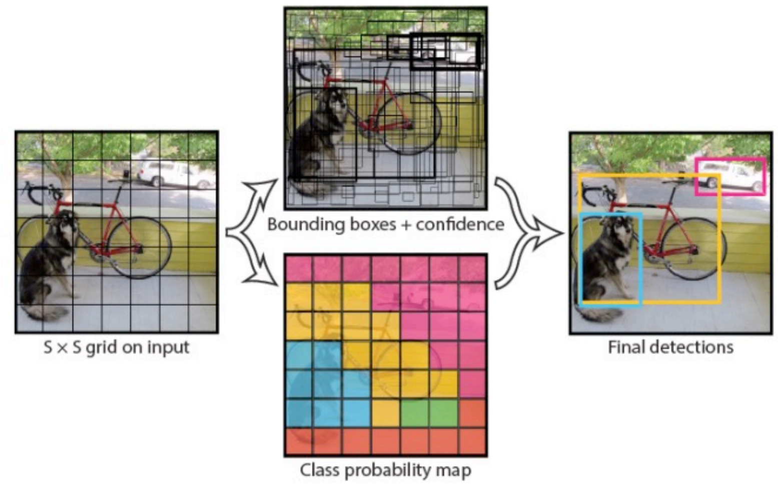
Review: RPN

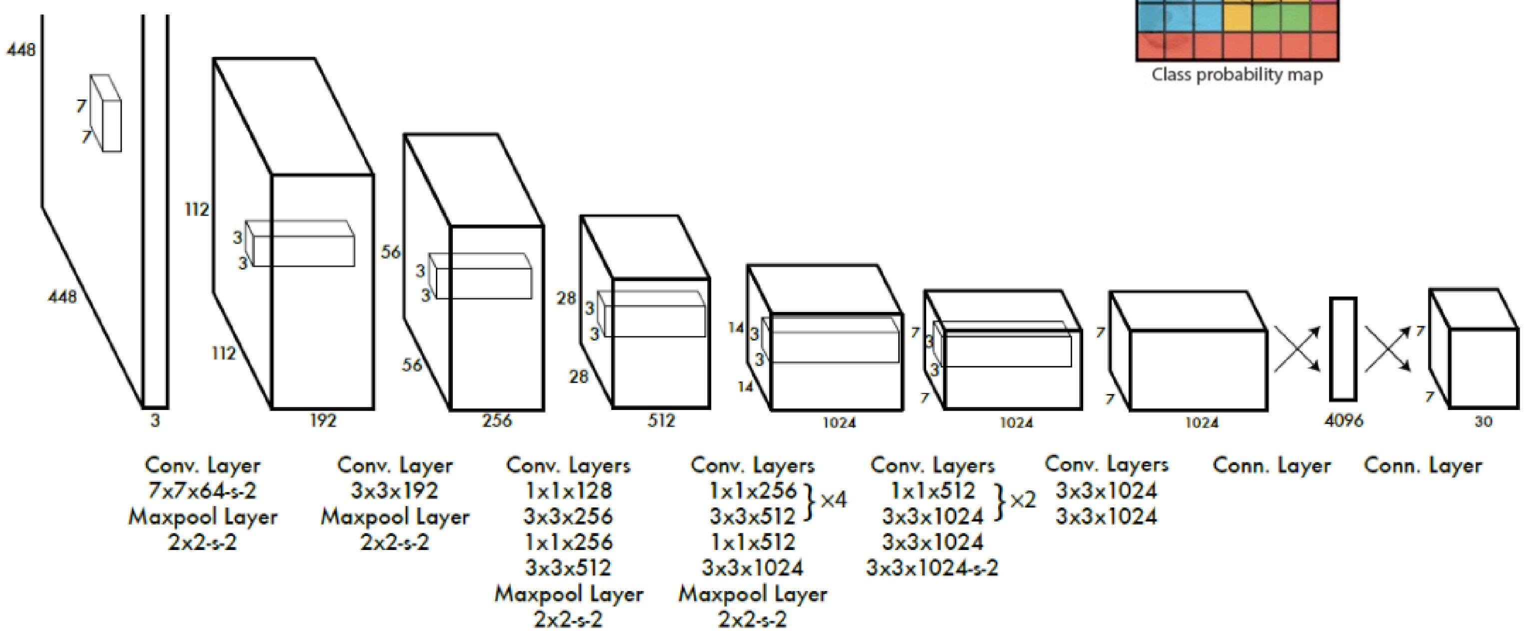
- Slide a small window (3x3) over the conv5 layer
  - Predict object/no object
  - Regress bounding box coordinates with reference to *anchors* (3 scales x 3 aspect ratios)



Review: YOLO

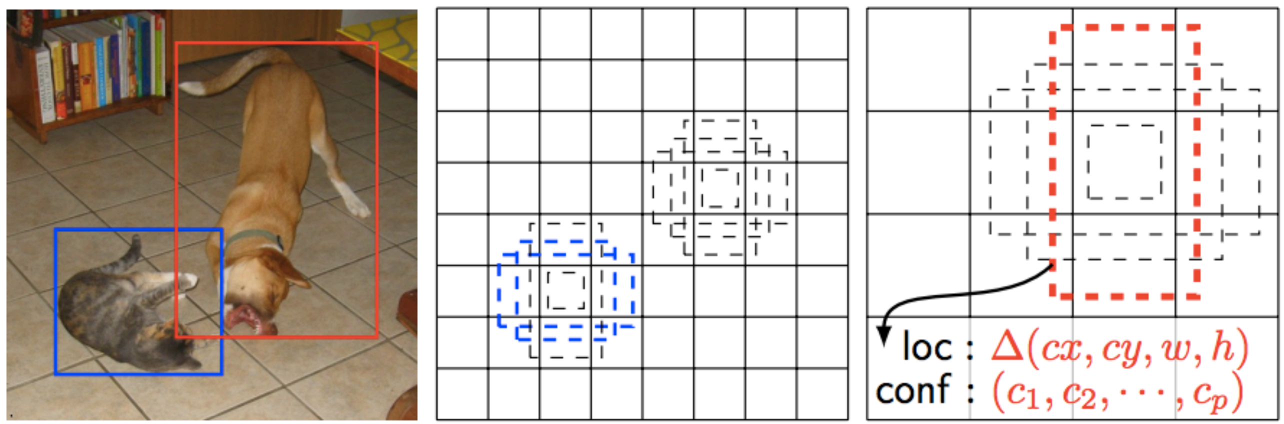
1. Take 7x7 conv feature map
2. Add two FC layers to predict, at each location, a score for each class and 2 bboxes w/ confidences
   - For PASCAL, output is 7x7x30 (30 = 20 + 2*(4+1))



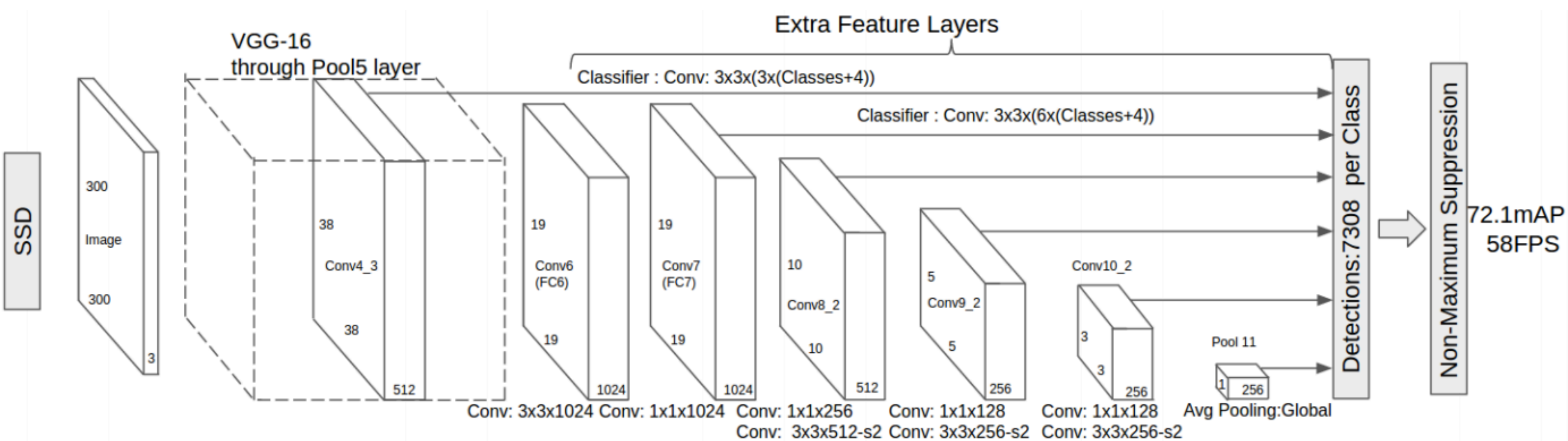


J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, [You Only Look Once: Unified, Real-Time Object Detection](), CVPR 2016

Review: SSD



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map



W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, SSD: Single Shot MultiBox Detector, ECCV 2016.

Summary: Object detection with CNNs

- R-CNN: region proposals + CNN on cropped, resampled regions
- Fast R-CNN: region proposals + RoI pooling on top of a conv feature map
- Faster R-CNN: RPN + RoI pooling
- Next generation of detectors
  - Direct prediction of BB offsets, class scores on top of conv feature maps
  - Get better context by combining feature maps at multiple resolutions