Object detection

370: Intro to Computer Vision

Subhransu Maji

May 6, 2025

College of
INFORMATION AND
COMPUTER SCIENCES

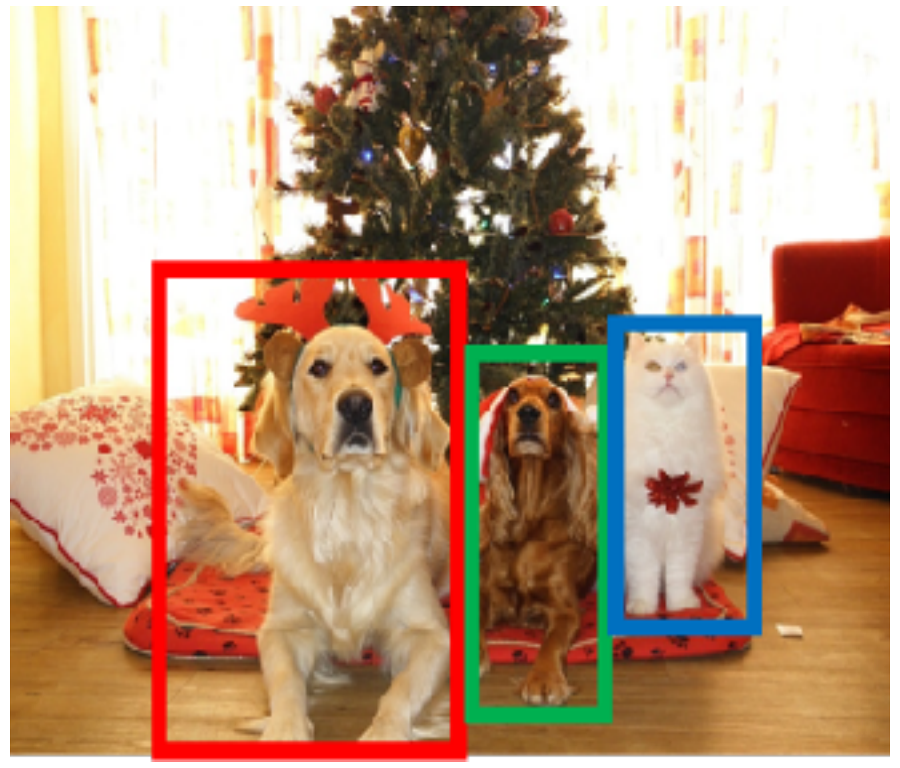
UMASS
AMHERST

Computer Vision Tasks

Classification

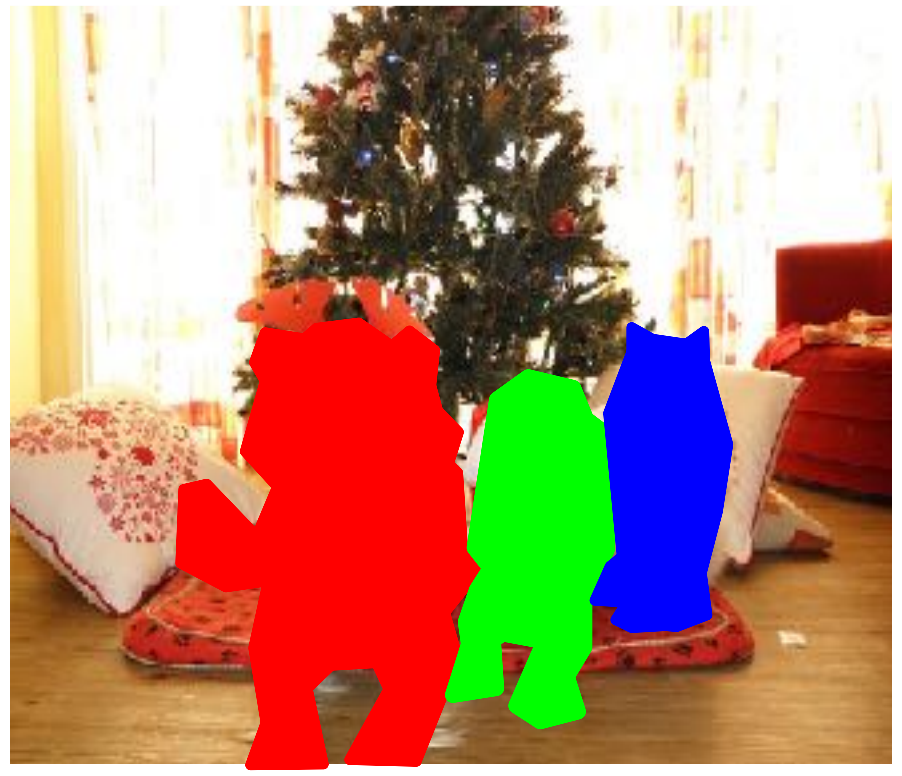
CAT

Object Detection



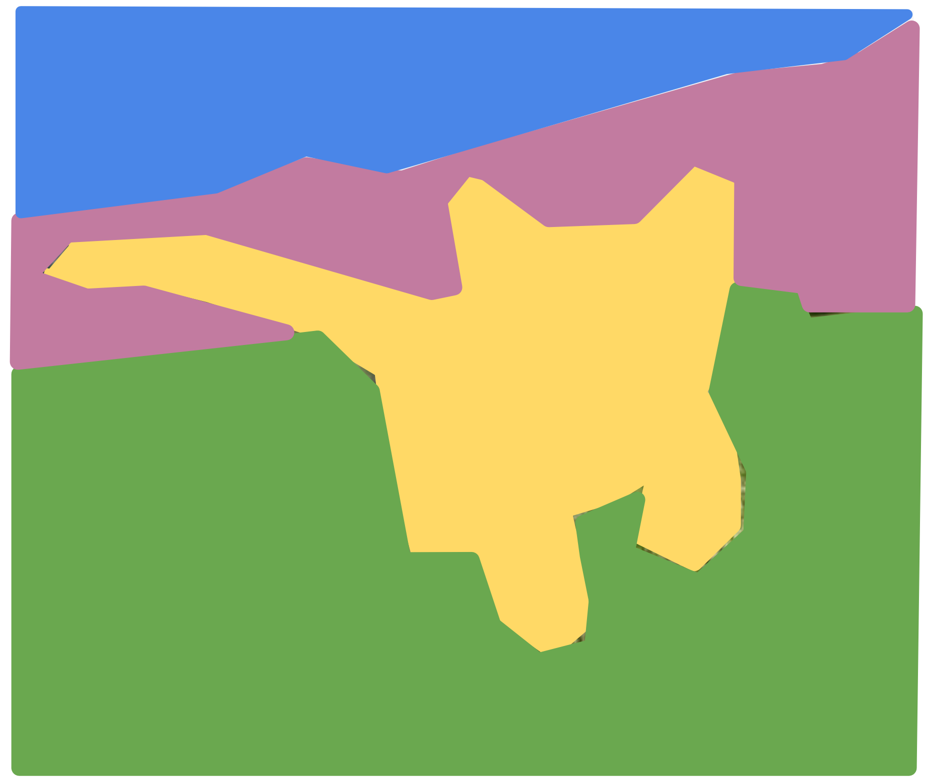
DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Semantic Segmentation



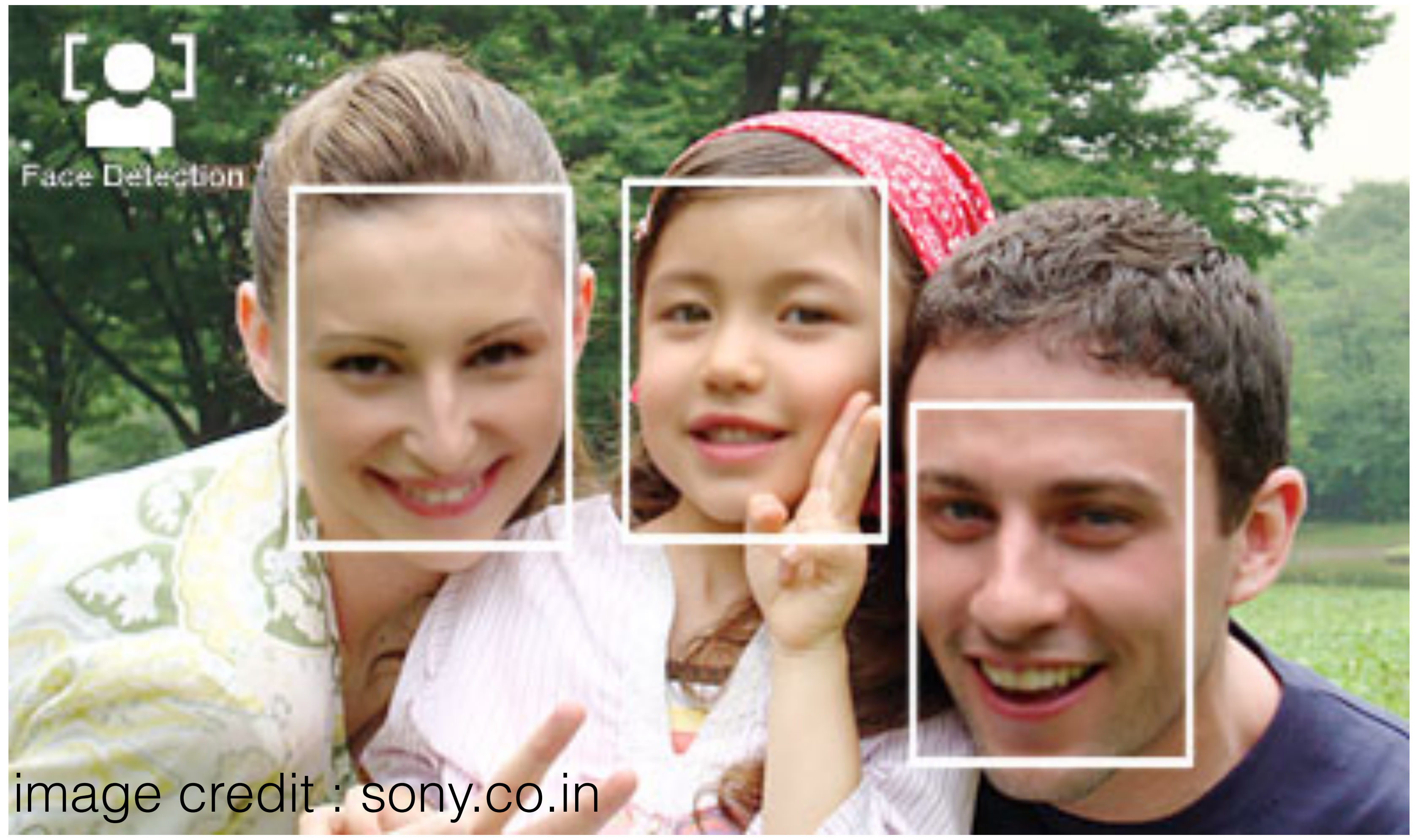
GRASS, CAT, TREE, SKY

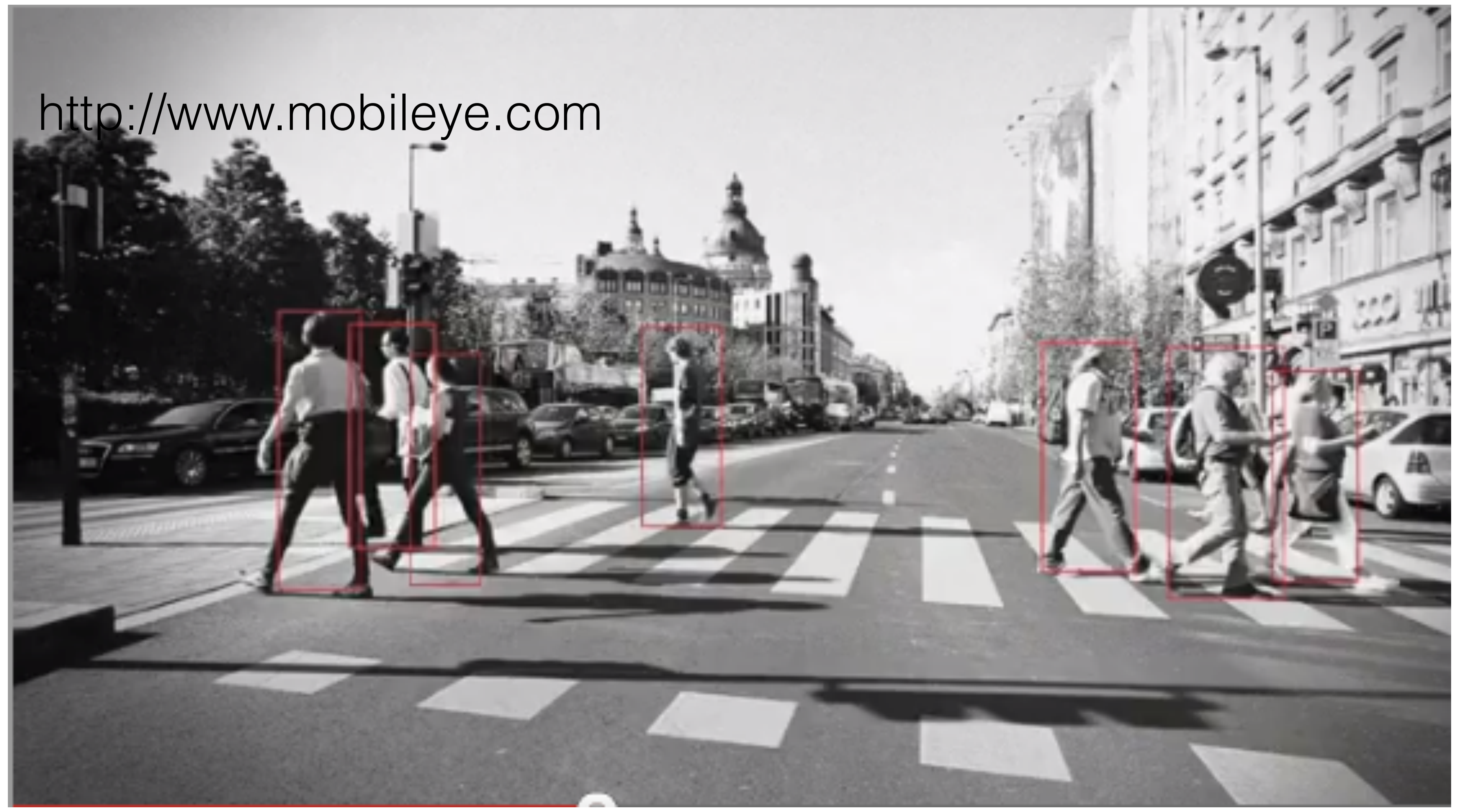
No spatial extent

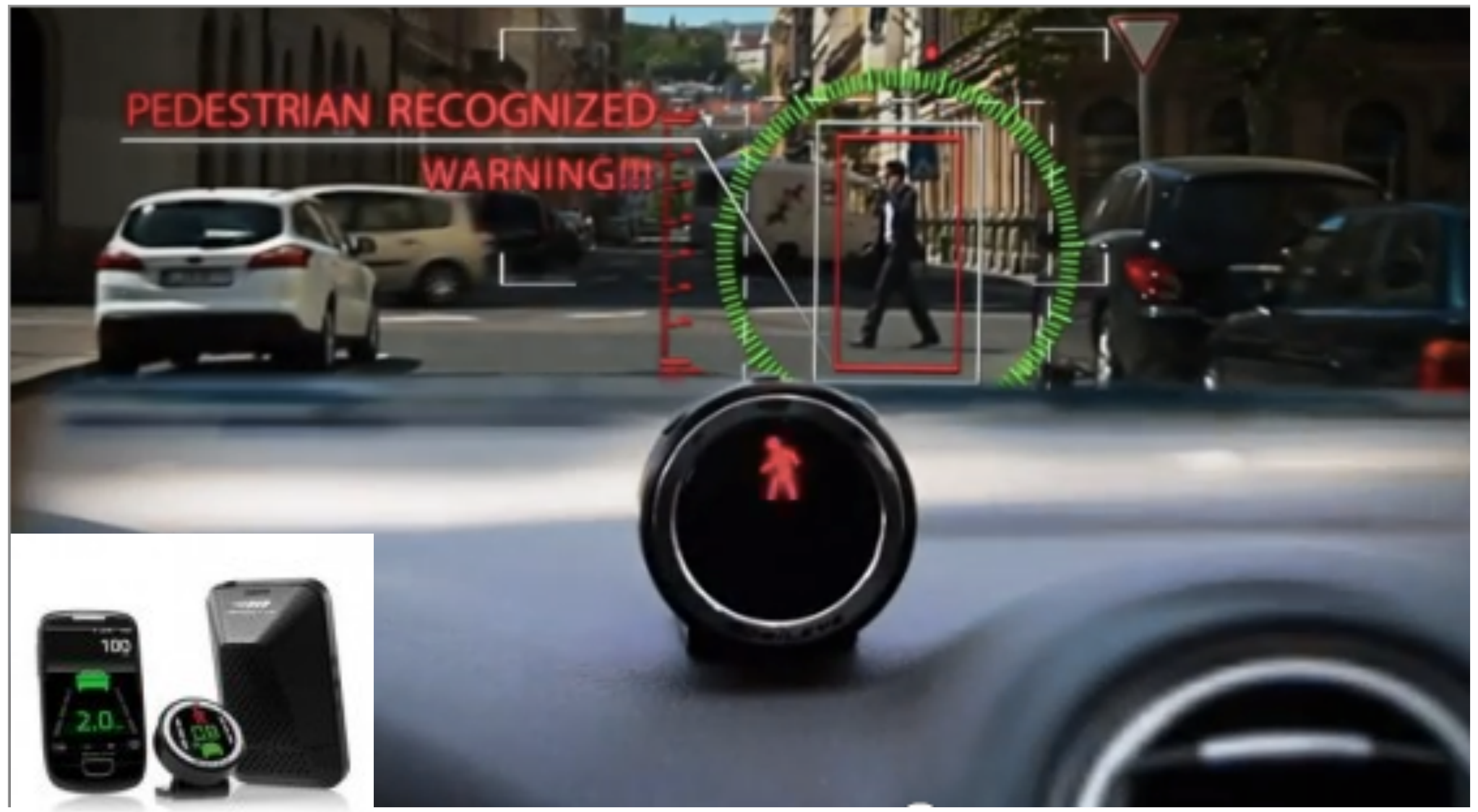
Multiple Objects

No objects, just pixels

Applications of Object Detection







Detection = Repeated Classification

face or not?

Challenges

Computational

- Large number of **location + scale** combinations
- A mega pixel image has a **millions** of candidate locations
- We should try to spend as little time as possible on each candidate

Accuracy

- The false positive rate of the classifier has to be very low
- 1 FP per image requires ~ 10^{-6} FP per candidate location

Lecture outline

Sliding-window detectors

- Case study: Dalal & Triggs, CVPR 2005
  - Detection as template matching
    - HOG feature pyramid
    - Non-maximum suppression
  - **Learning a template** — linear classifiers, hard negative mining

Evaluating a detector — some detection benchmarks

Region-based detectors

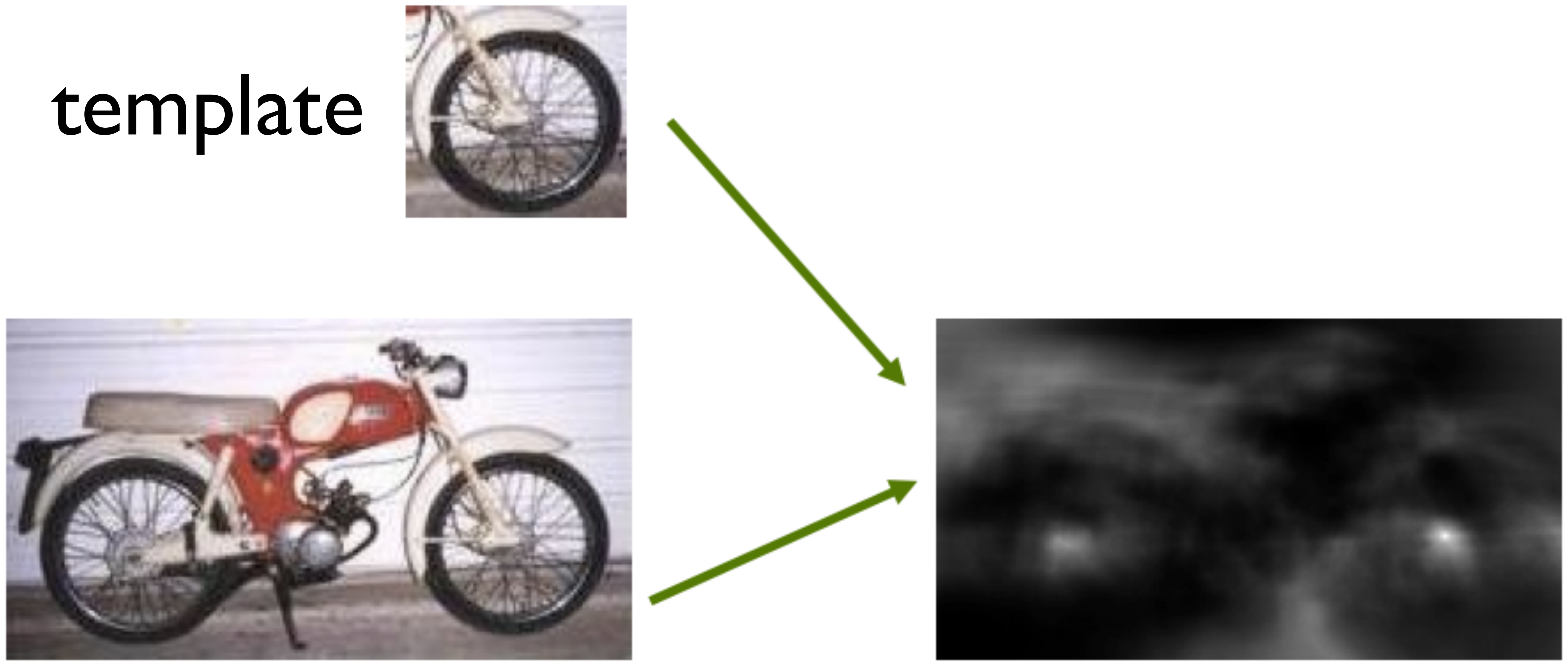
- Case study: Van de Sande et al., ICCV 2013
- Case study: R-CNN, Girshick et al., CVPR 2014

Detection as template matching

Consider matching with image patches

- What could go wrong?

template

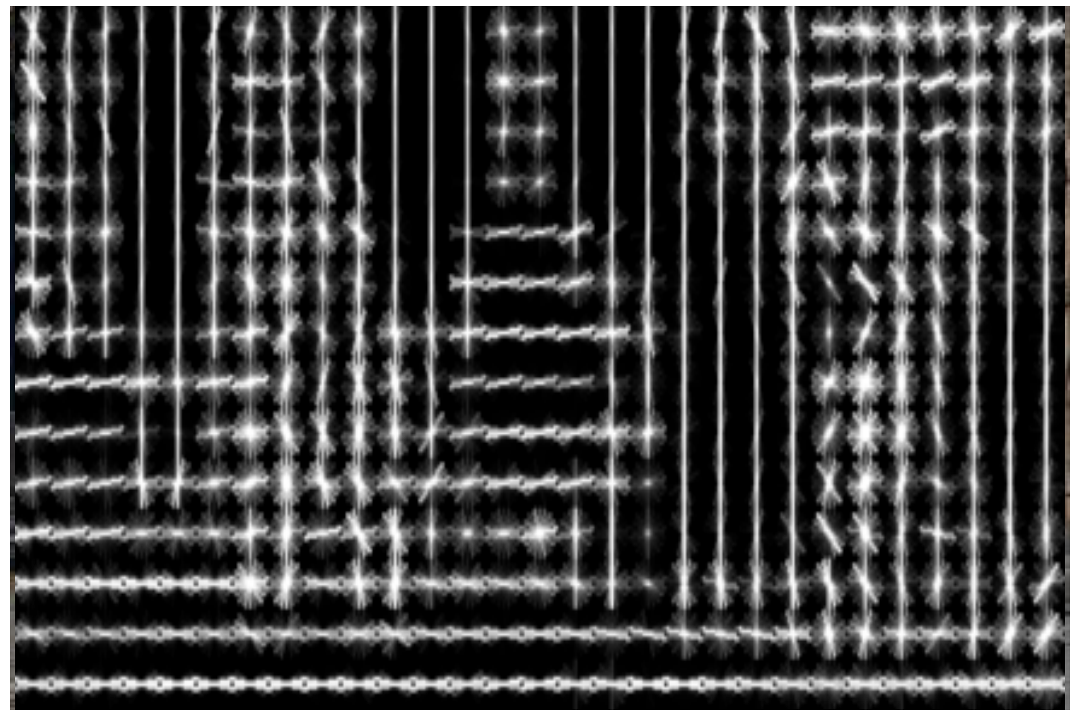


image

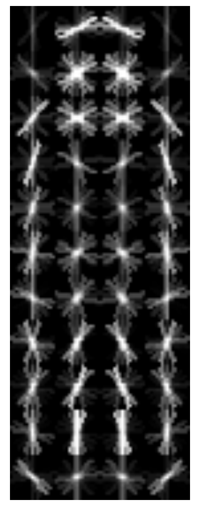
match quality
e.g., cross correlation

Template matching with HOG

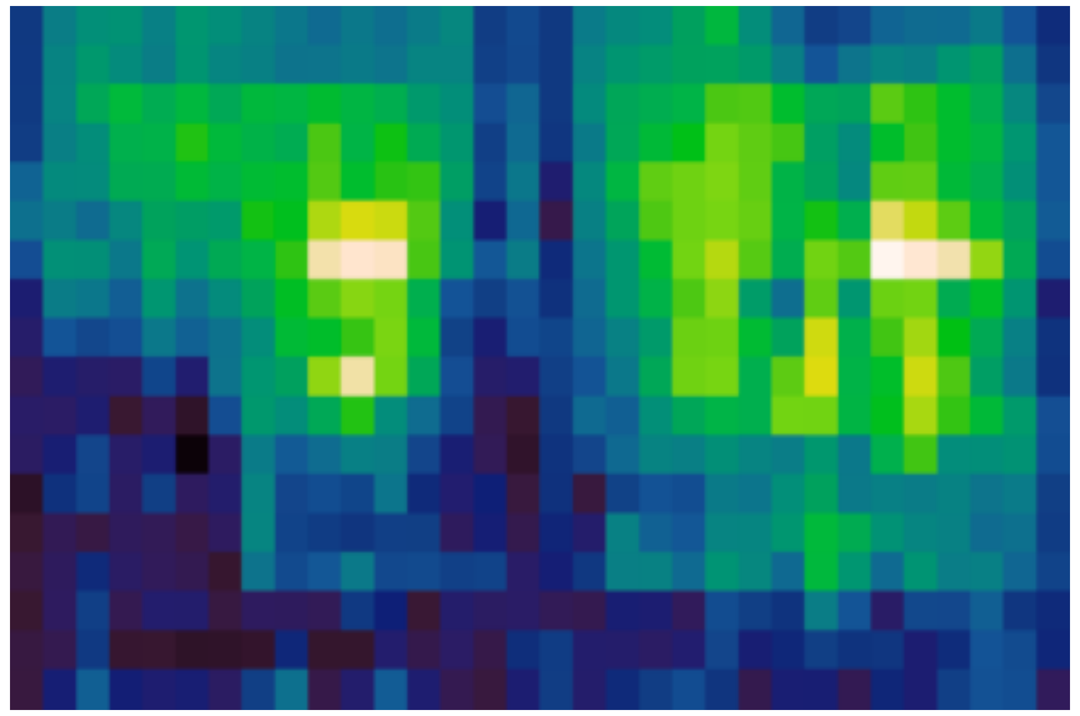
HOG feature map



Template



Detector response map



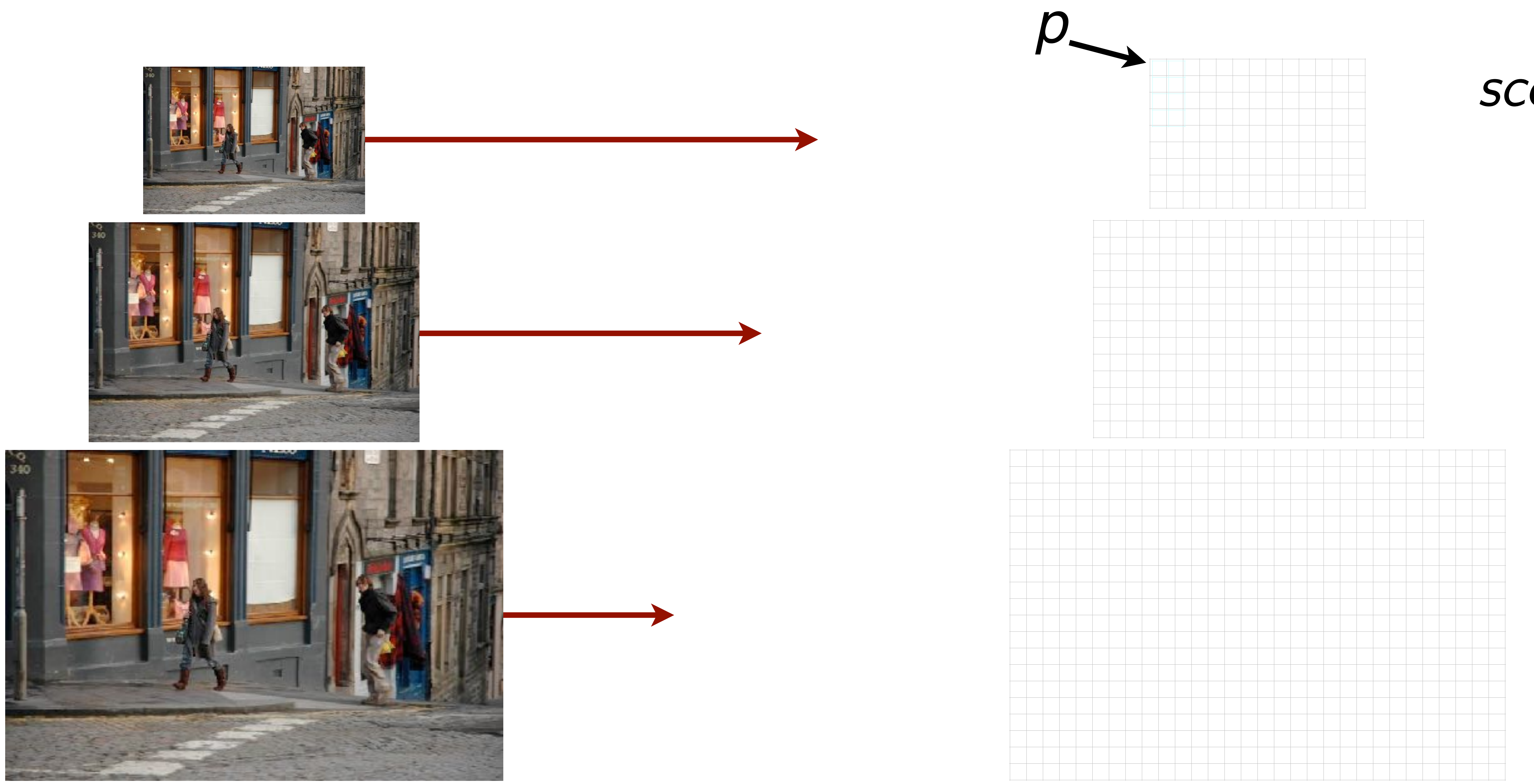
Compute the HOG feature map for the image

Convolve the template with the feature map to get score

Find peaks of the response map (non-max suppression)

What about multi-scale?

Multi-scale template matching



$$score(I, p) = \mathbf{w} \cdot \phi(I, p)$$

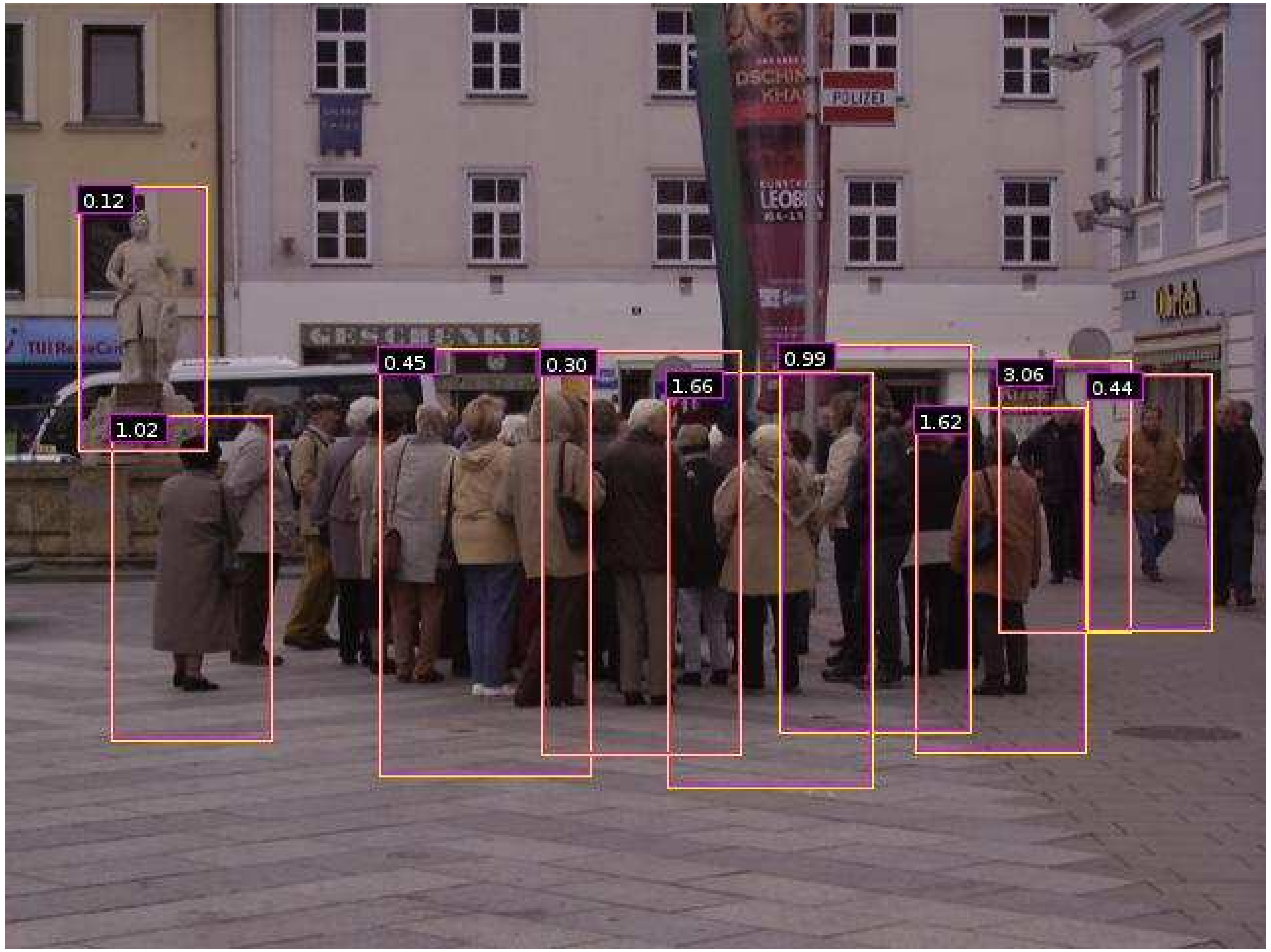
Image pyramid HOG feature pyramid

Compute HOG of the whole image at multiple resolutions

Score each sub-windows of the feature pyramid

Threshold the score and perform non-maximum suppression

Example pedestrian detections

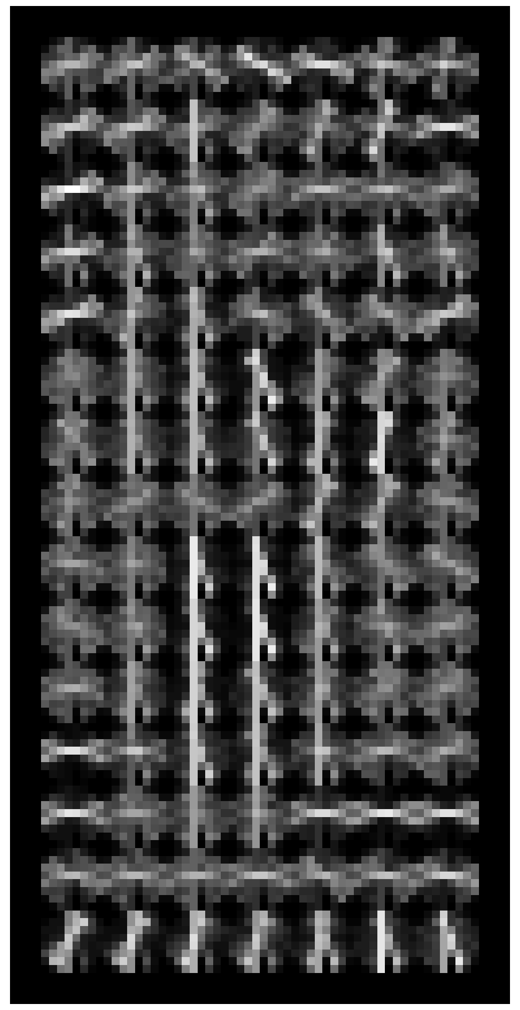


[Dalal05]

Learning a template

Pos = {…  …}

Annotations

is this template good?

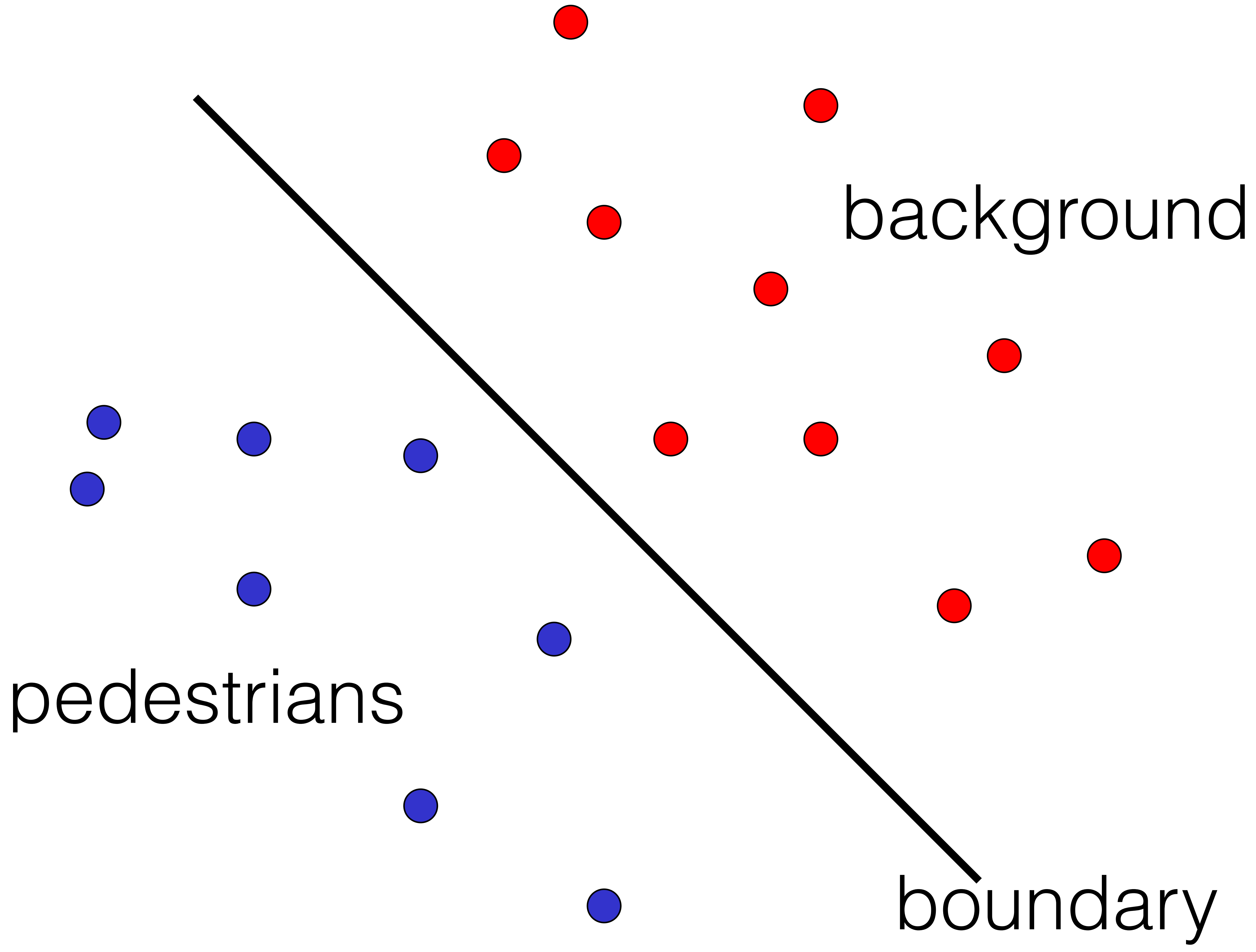
Cropped positive   HOG

[Dalal05]

Learning a template

Score high on pedestrians and low on background patches

Discriminative learning setting — lets use linear classifiers!

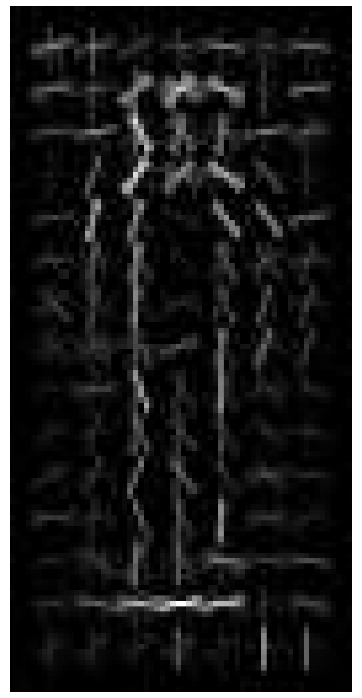


Issue: too many background patches

Initial training

Pos = {…  …}

Neg = {… random background patches …}



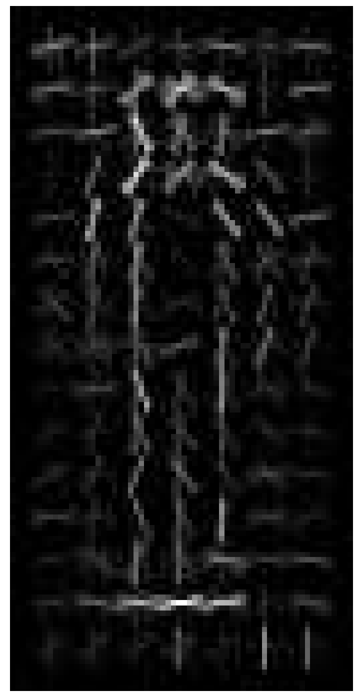
Test on cropped windows

SVM (l_2 reg, l_1 loss)

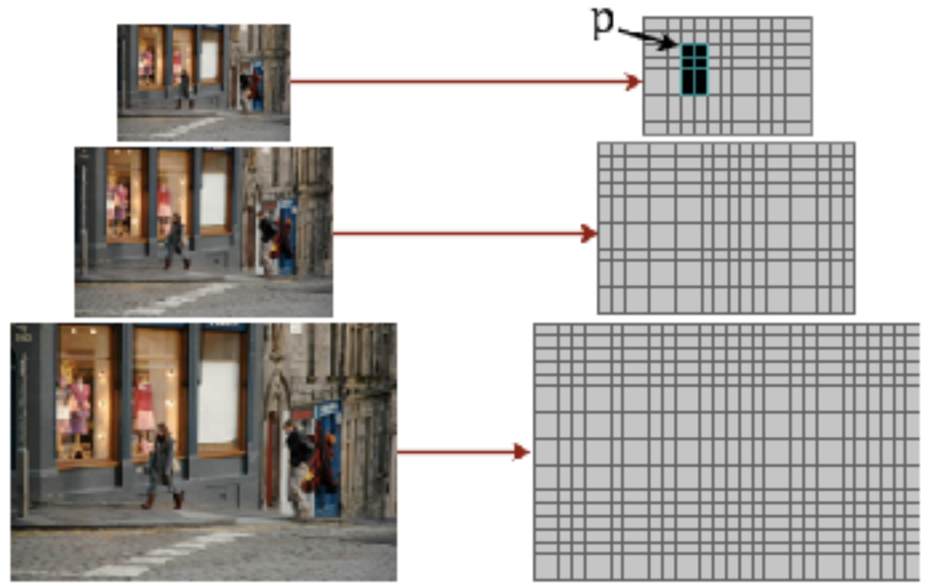
Mining hard negatives

Pos = {...  ...}

Neg_{rand} = {... random background patches ...}

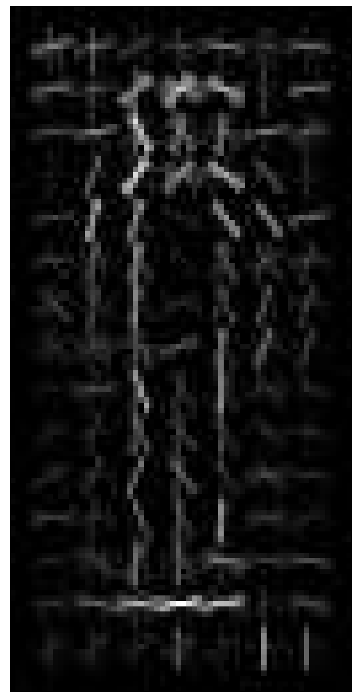


SVM



"Hard" negatives

+ Neg_{hard} = {... windows with score >= -1 ...}



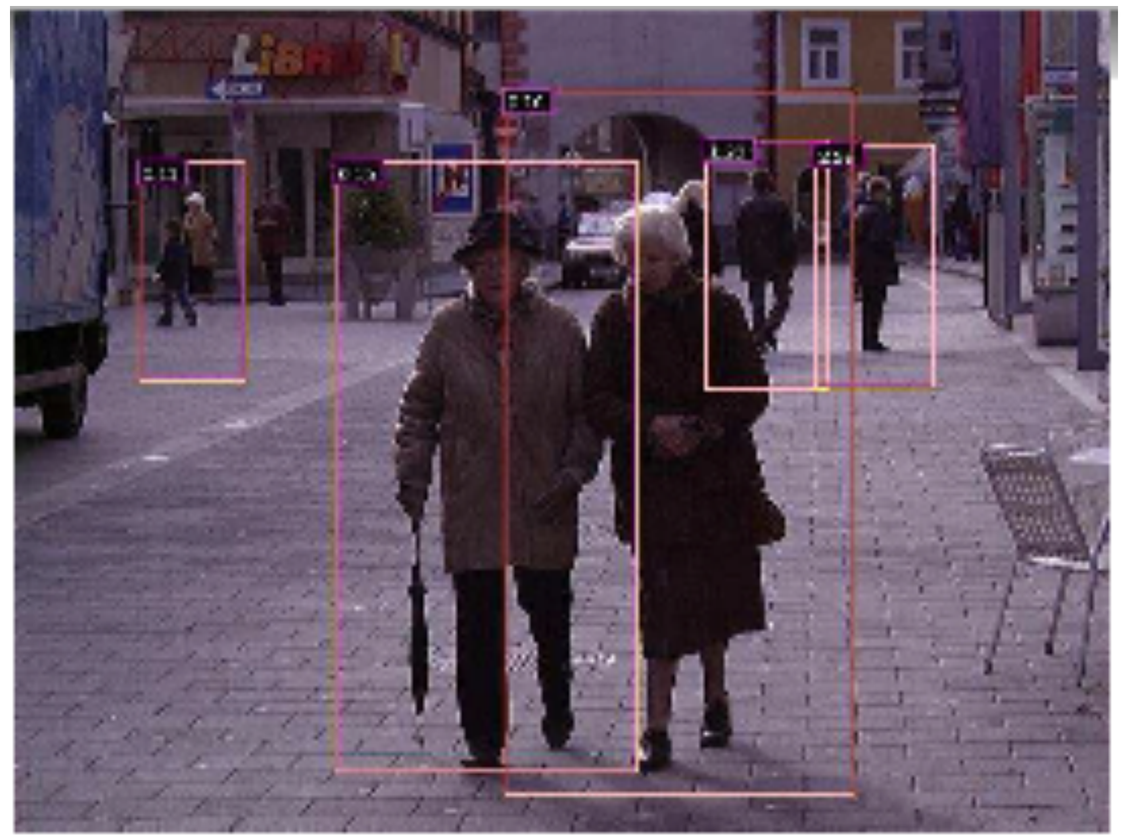
SVM

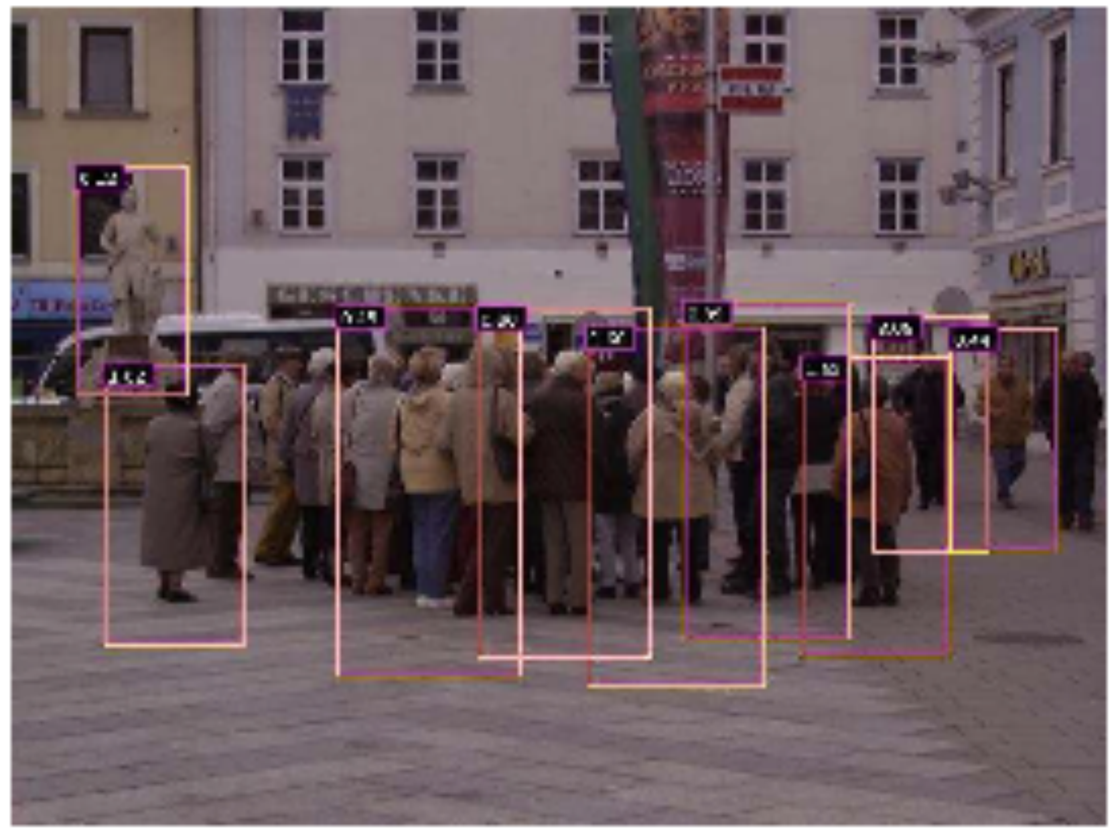
INRIA person dataset

N. Dalal and B. Triggs, CVPR 2005

A dataset of people in:

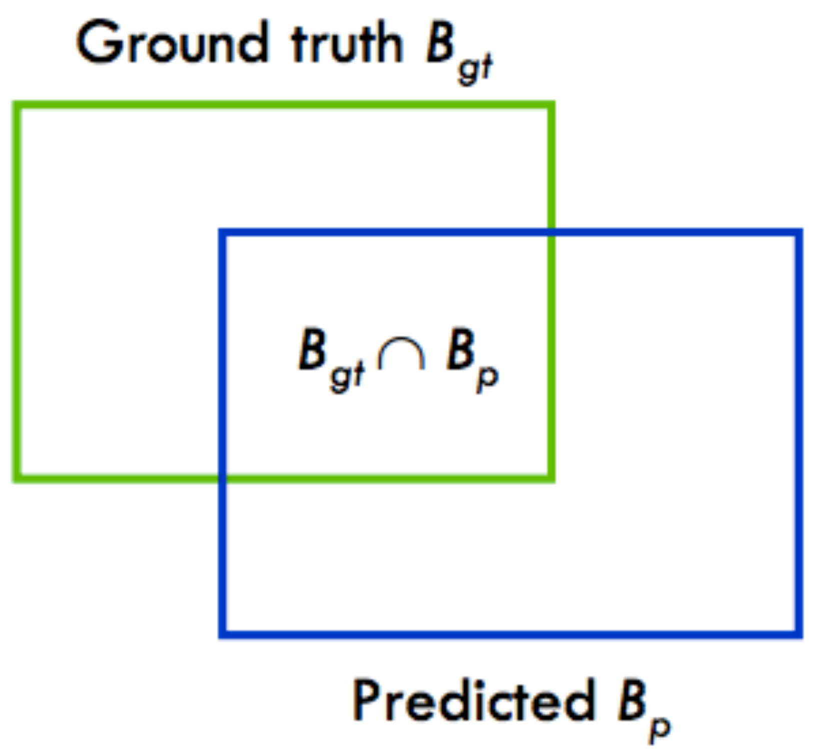
- Wide variety of articulated poses
- Variable appearance/clothing
- Complex backgrounds
- Unconstrained illumination
- Occlusions, different scales



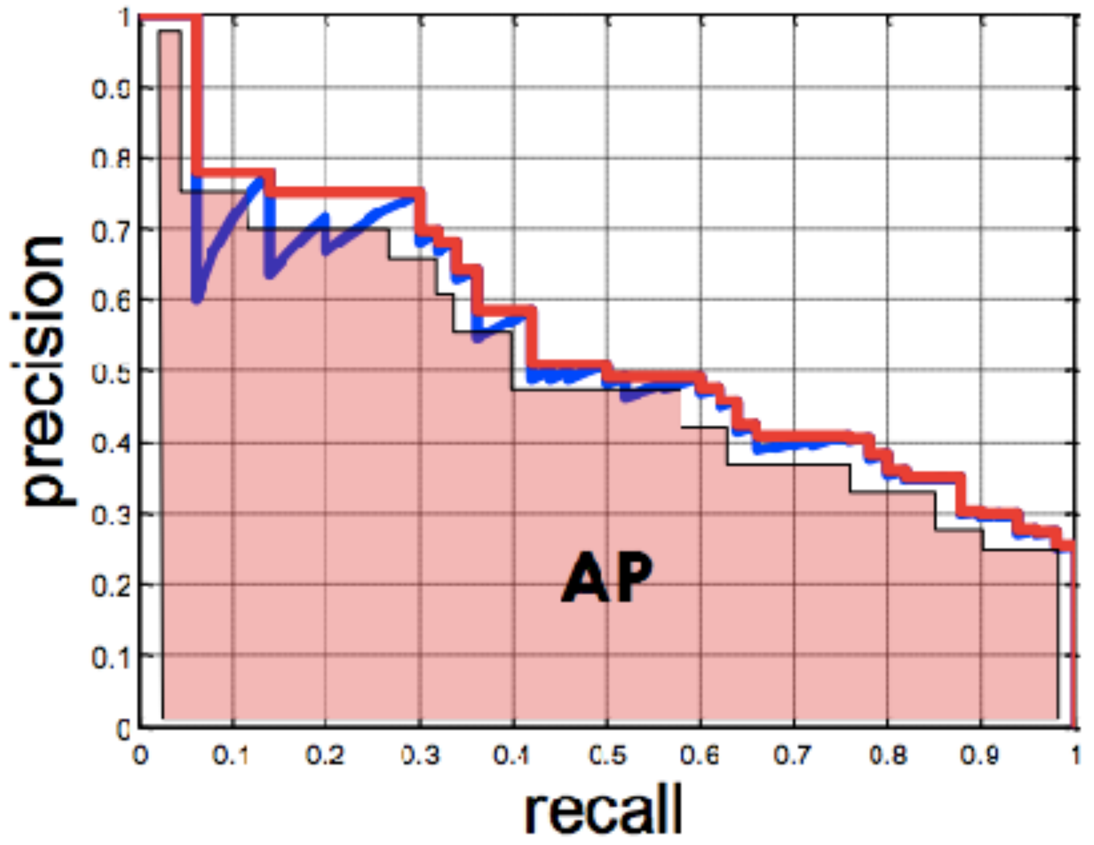


http://pascal.inrialpes.fr/data/human/

Detection evaluation



$$\text{overlap}(B_{\text{gt}}, B_{\text{p}}) = \frac{|B_{\text{gt}} \cap B_{\text{p}}|}{|B_{\text{gt}} \cup B_{\text{p}}|}$$



Assign each prediction to

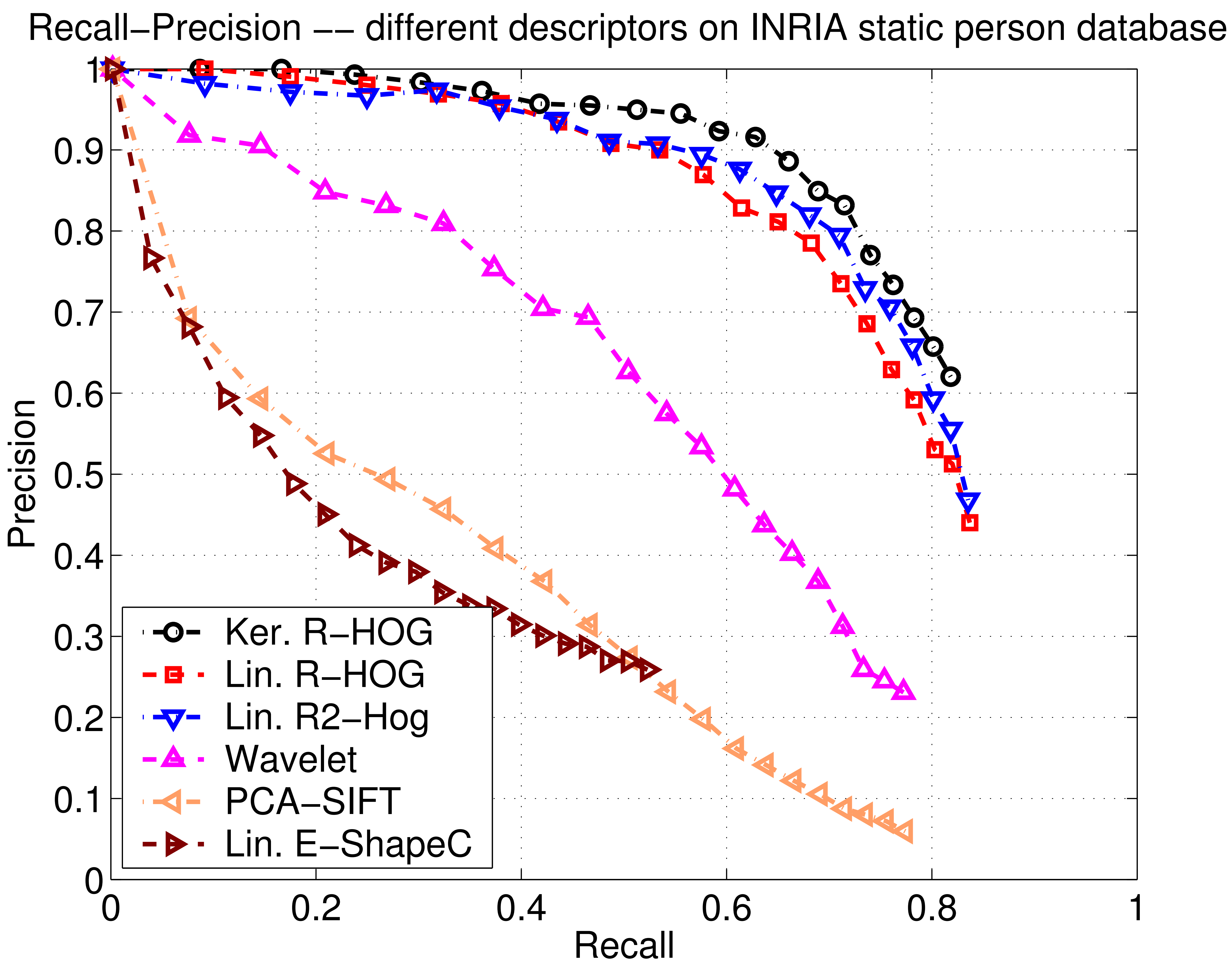
- true positive (TP) or false positive (FP)

Precision$_{@k}$ = #TP$_{@k}$ / (#TP$_{@k}$ + #FP$_{@k}$)

Recall$_{@k}$ = #TP$_{@k}$ / #TotalPositives

Average Precision (AP)

Pedestrian detection on INRIA dataset



AP = 0.75 with a linear SVM

Very good, right?

PASCAL VOC Challenge

Localize & name *(detect)* 20 basic-level object categories

- Airplane, bicycle, motorbike, bus, boat, train, car, cat, bird, cow, dog, horse, person, sheep, bottle, sofa, monitor, chair, table, plant

Run from 2005 - 2012

11k training images with 500 to 8000 instances / category

Substantially more challenging images

Dalal and Triggs detector AP on ‘person’ category: **12%**

PASCAL examples

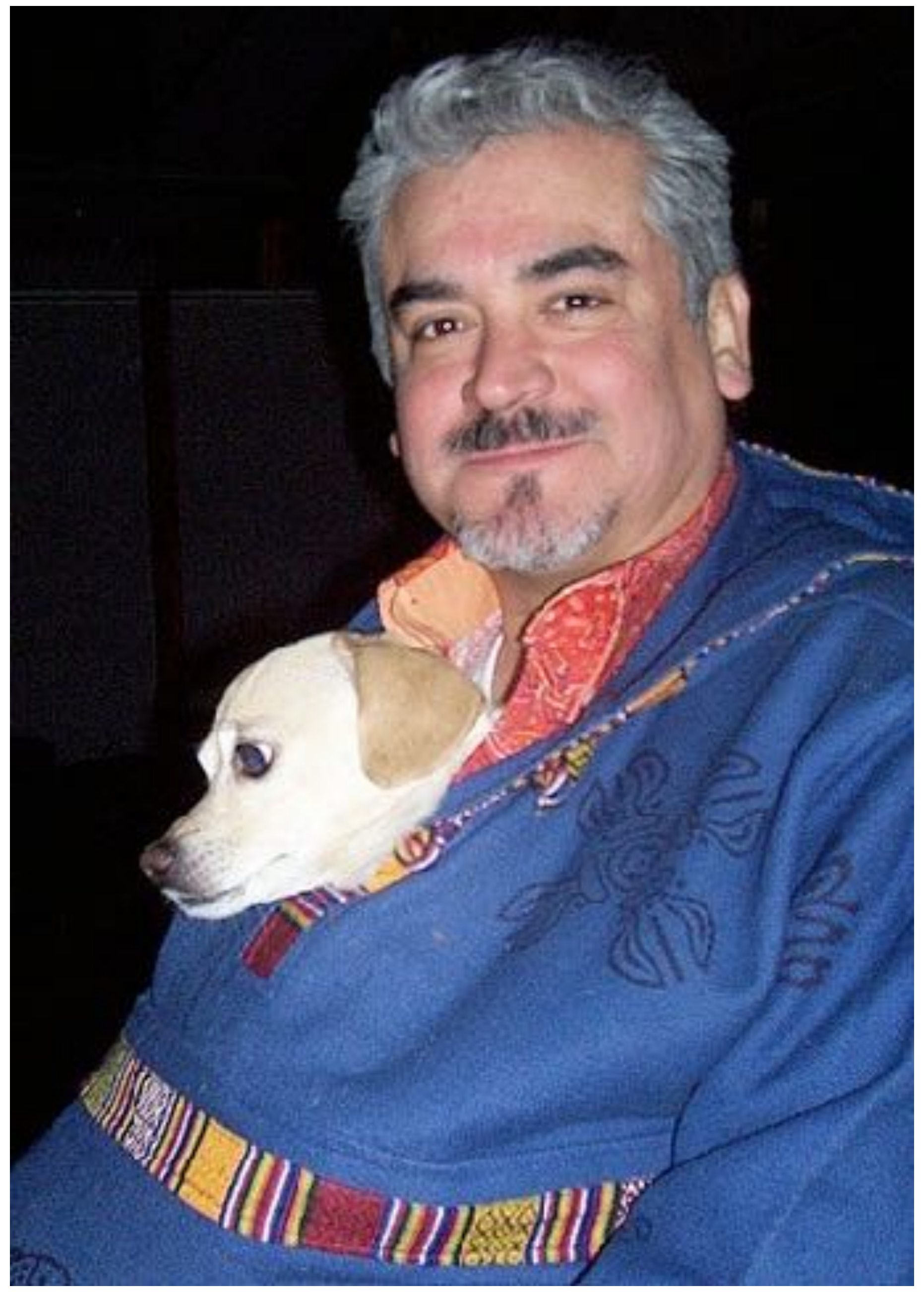


Image credits: PASCAL VOC

PASCAL examples

Viewpoint

Image credits: PASCAL VOC

PASCAL examples

Subcategory — “airplane” images

Image credits: PASCAL VOC

PASCAL examples

Subcategory — “car” images



Image credits: PASCAL VOC

Part-based models

A single template is does not capture the variability

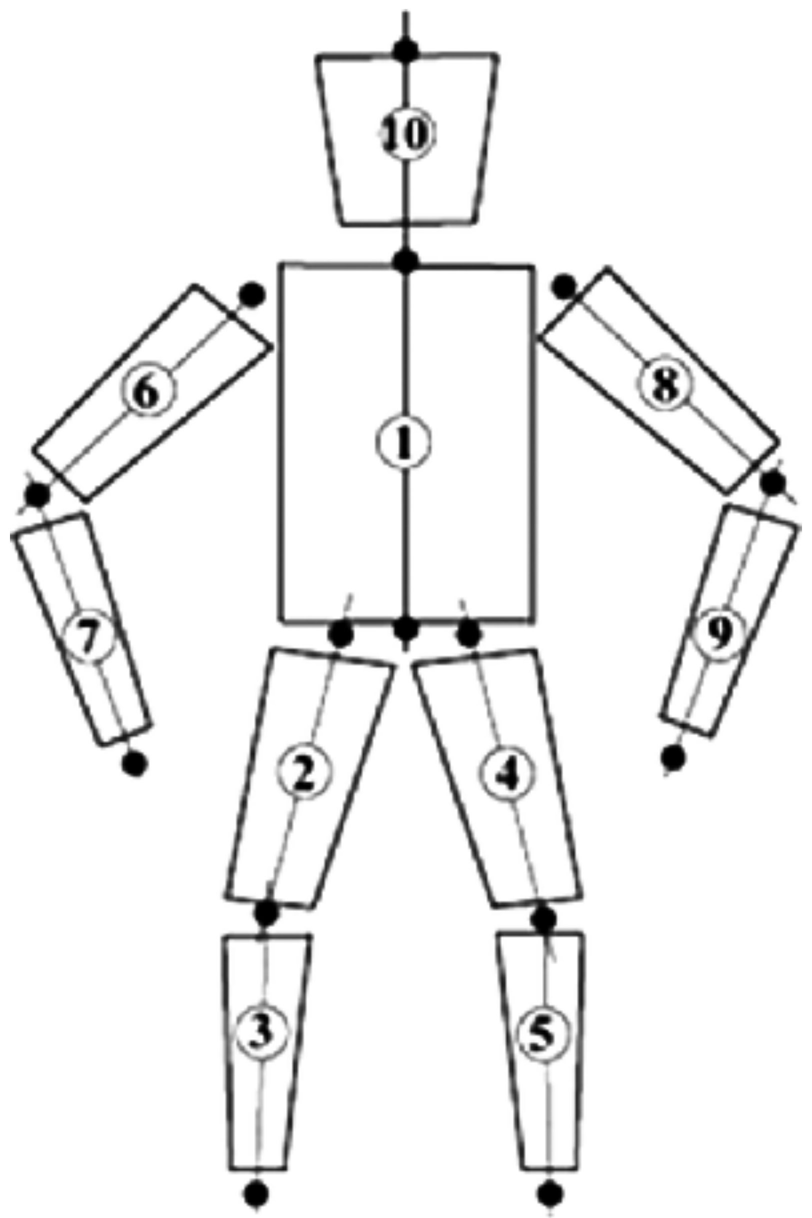
- Person detection AP = 12% using a single template

Lets focus on the person category

- How can we model the variability due to pose, articulation, viewpoint, etc.
- **Idea:** Detect parts and stitch them together
- But what should the parts be?

Parts based on human anatomy

pictorial structures



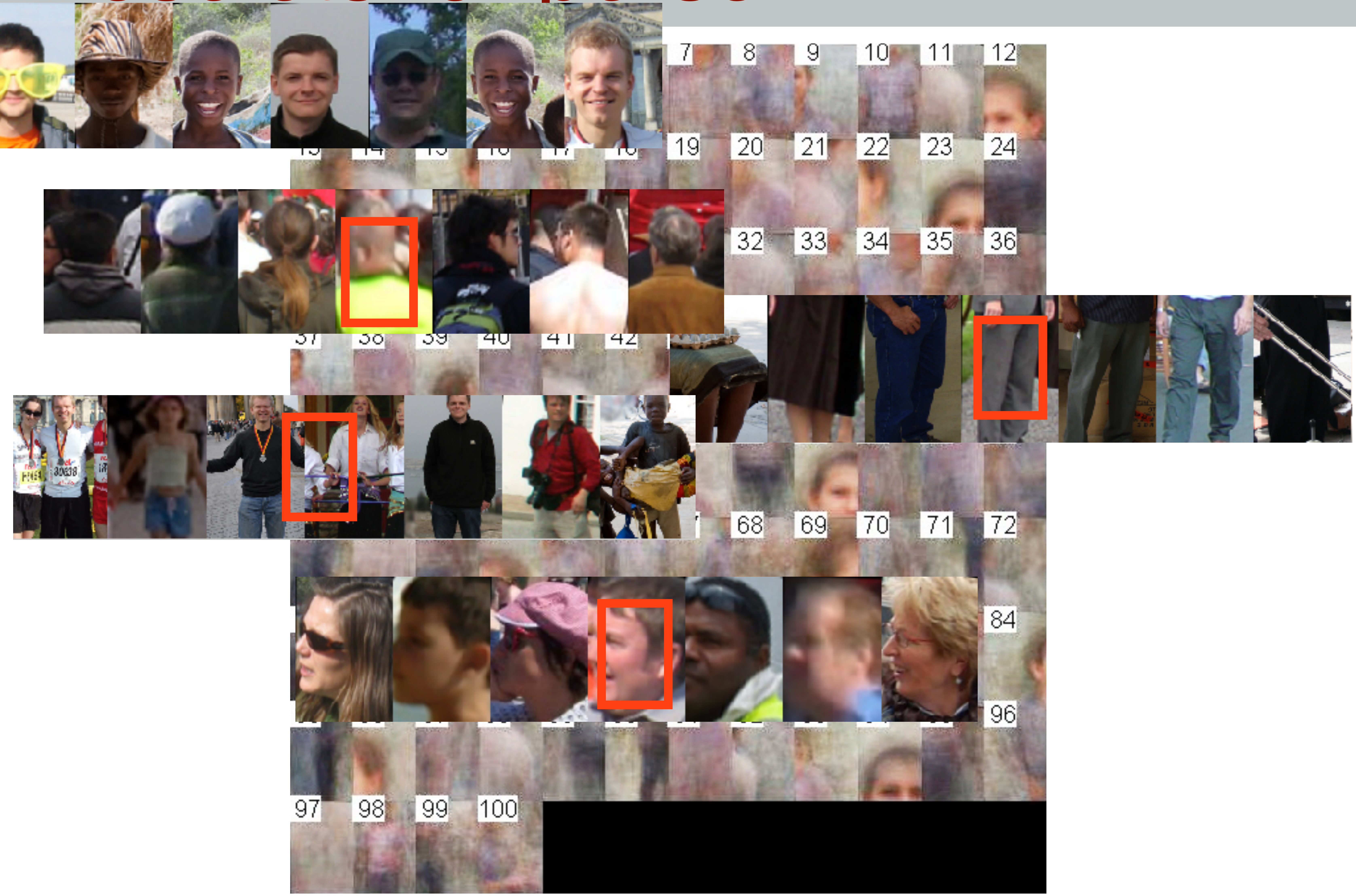
"stick-figure models"

Fisher & Elchlager 73, Nevatia & Binford 77, Felzenszwalb et al. 05, Ren et al. 05, Andriluka et al. 09, Ferrari et al. 08, Ramanan 06

it is hard to detect limbs

Can we leverage the success of *face* and *pedestrian* detectors?

Poselets for person



Bourdev et al. 10

PASCAL VOC detection challenge

"person" category VOC 2010 test set

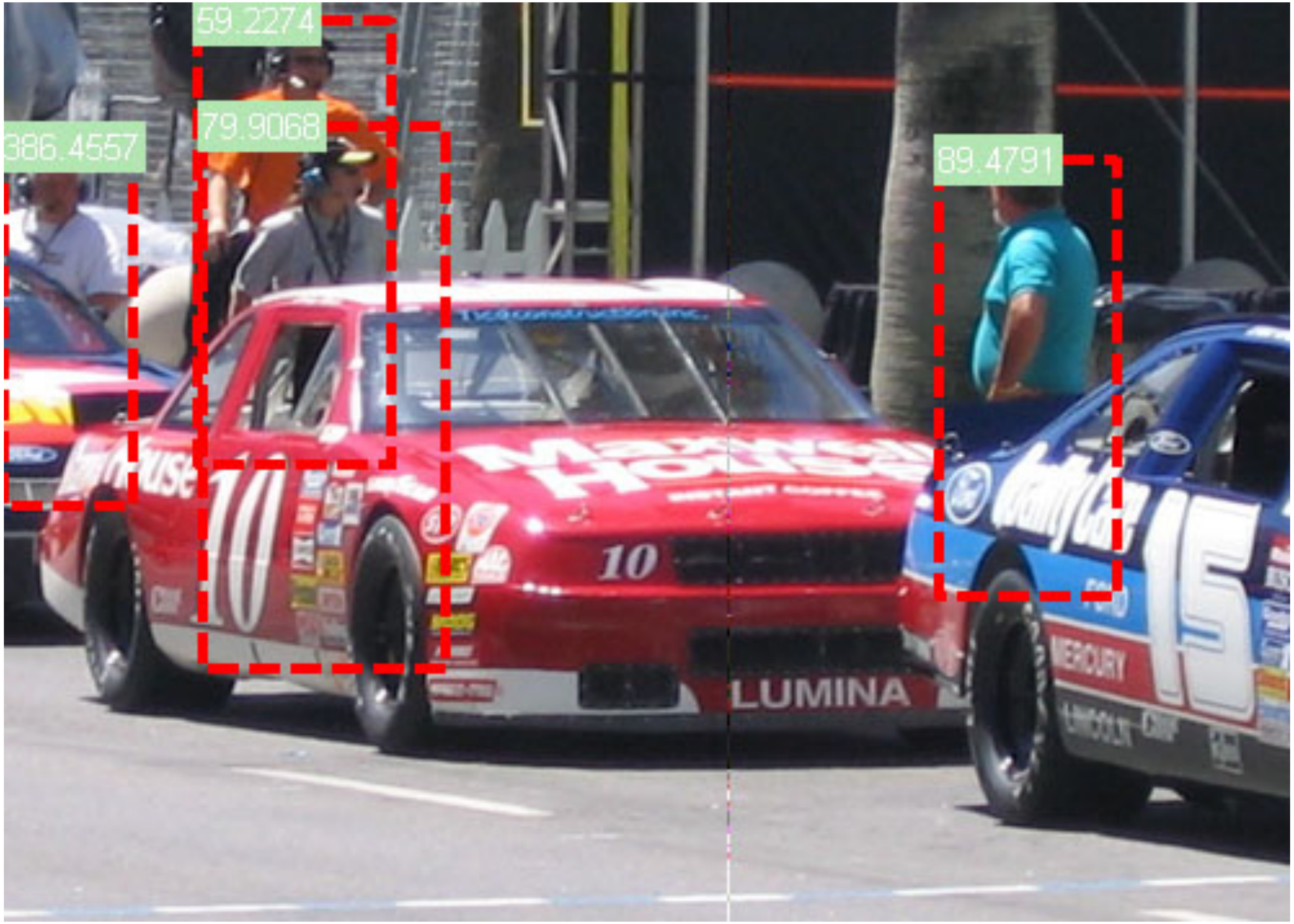
Method	Detection AP
Poselets	48.5%
Dalal & Triggs	12.0%
DPM (Girschik et al.)	43.3%

Poselet detector — same features, 100x templates

L. Bourdev, S. Maji, T. Brox, J. Malik, Detecting people using mutually consistent poselet activations, ECCV 2010

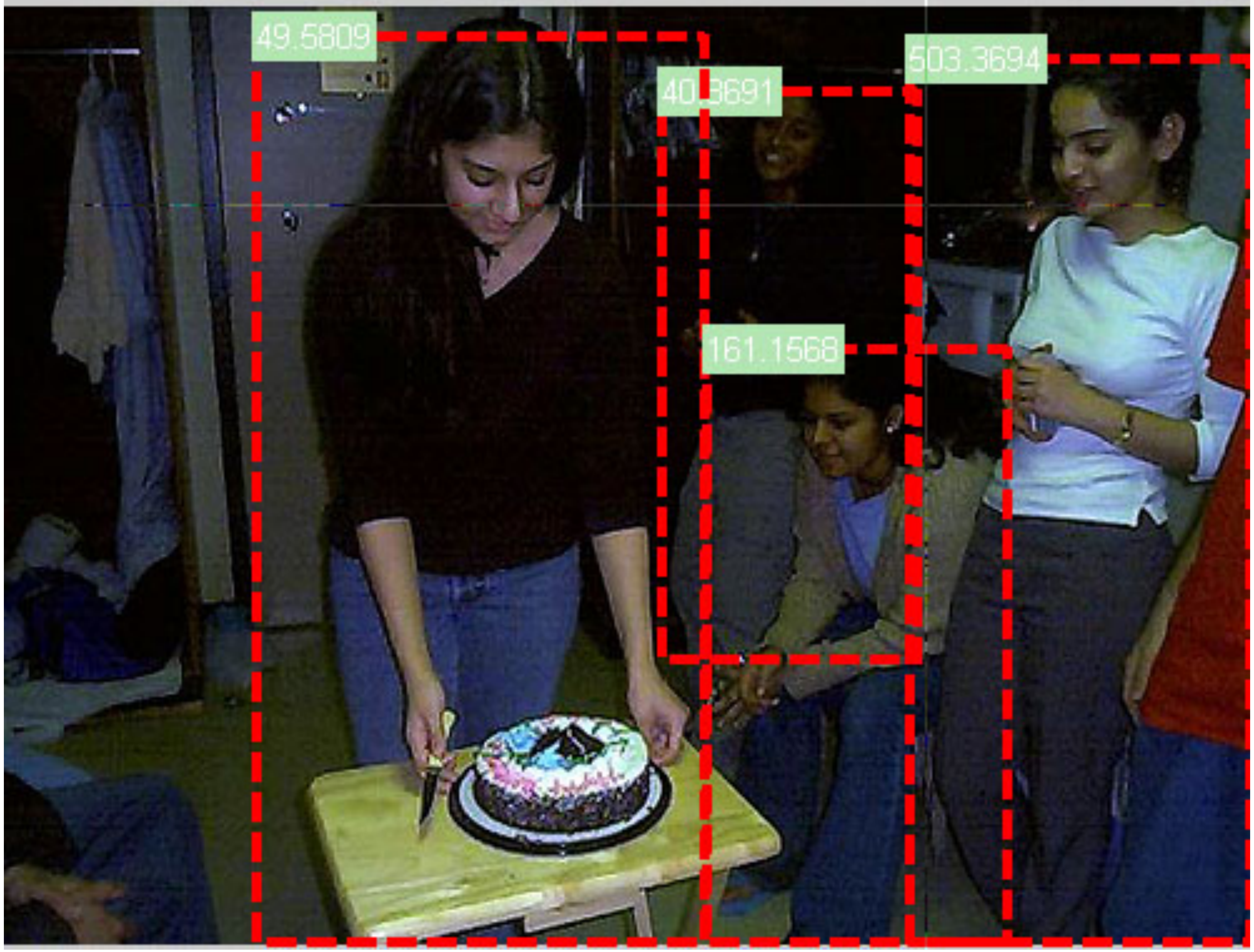
https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/poselets/

Example detections

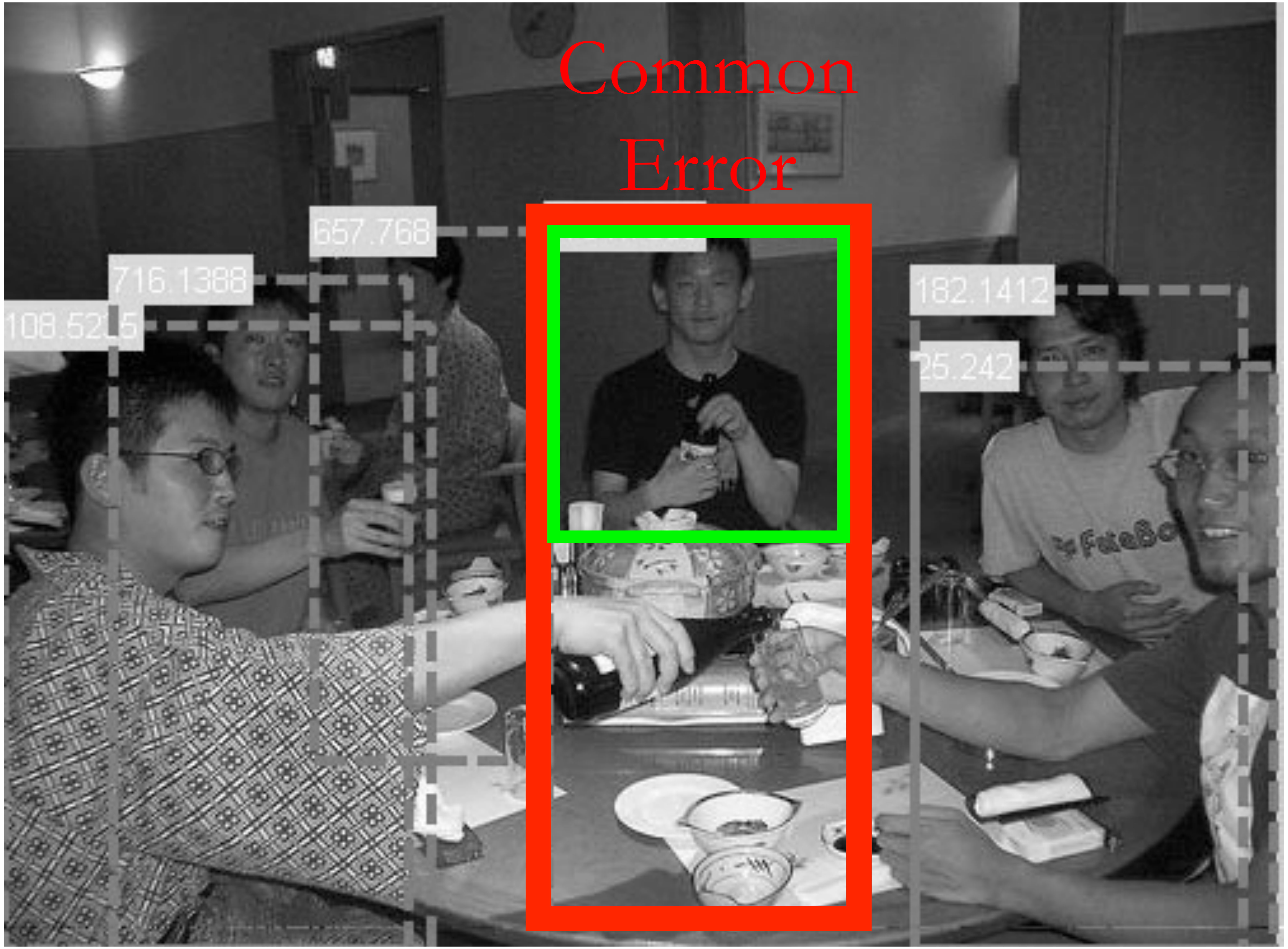


Example detections

Example detections



Example detections



Limitations of a sliding-window detector

Computationally expensive — there are too many windows

- Multiply by scales, aspect ratio (objects are not square)

Thus classifiers and features have to be *very* fast

- Linear classifiers and decision trees commonly used
- Features: simple pixel-based or gradient features used

But they also have to *accurate*!

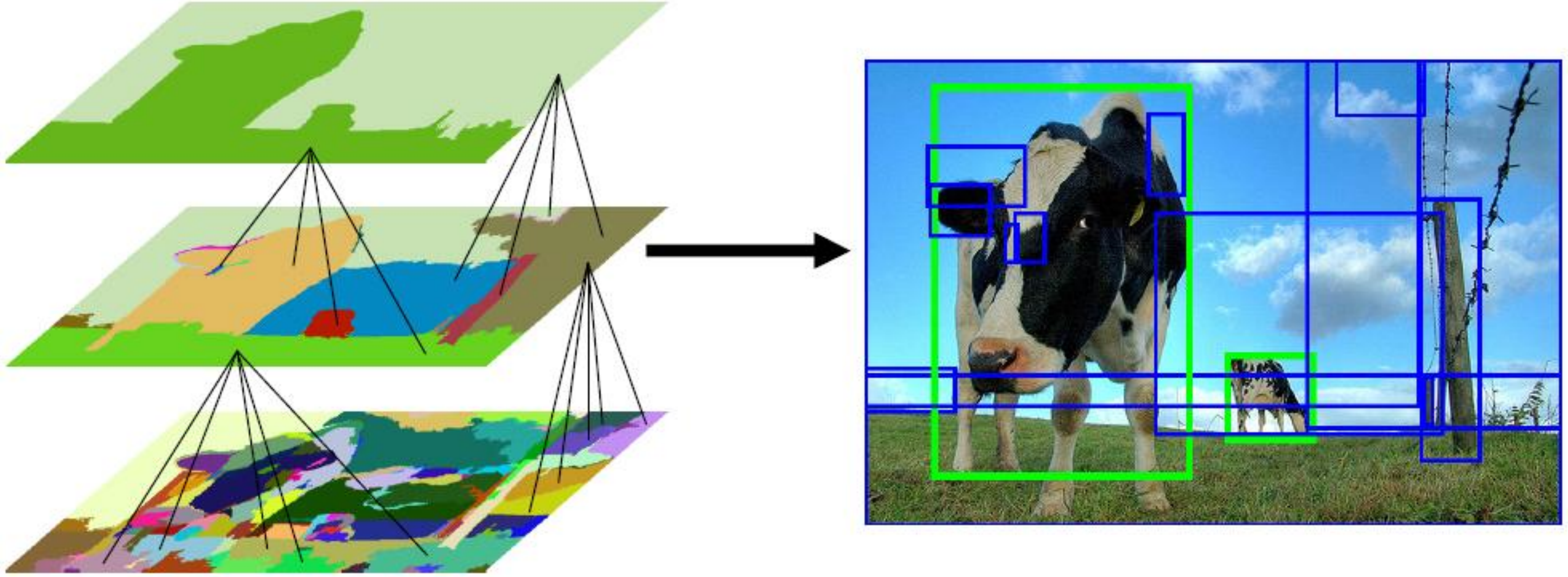
Alternate design: Region-based detectors

Choose a small number of regions to evaluate the classifier

- Number of regions (~10^3) << number of windows (10^6)
- We want high recall — no objects should be missed
- Should be category independent — to share the cost across categories
- Fast — shouldn't be slower than running the detector itself!

We will look at this approach

Segmentation as Selective Search for Object Recognition, K. Van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2013



Winner of the PASCAL VOC challenge 2010-12

Lets start with segmentations

“Efficient graph-based image segmentation”
Felzenszwalb and Huttenlocher, IJCV 2004

Apply some clustering approach using color information (e.g., k-means, graph-based clustering)
Often big objects are broken into multiple regions

How can we fix this?

How to obtain high recall?

Images are intrinsically hierarchical

Regions of a single size are not enough

- Lets merge regions to produce a hierarchy

Hierarchical clustering

1. Merge two most similar regions based on S
2. Update similarities between the new region and its neighbors
3. Go back to step 1 until the whole image is a single regions

Hierarchical clustering

Compute similarity measure between all adjacent region pairs *a* and *b* as:

$$S(a,b) = S_{size}(a,b) + S_{texture}(a,b)$$

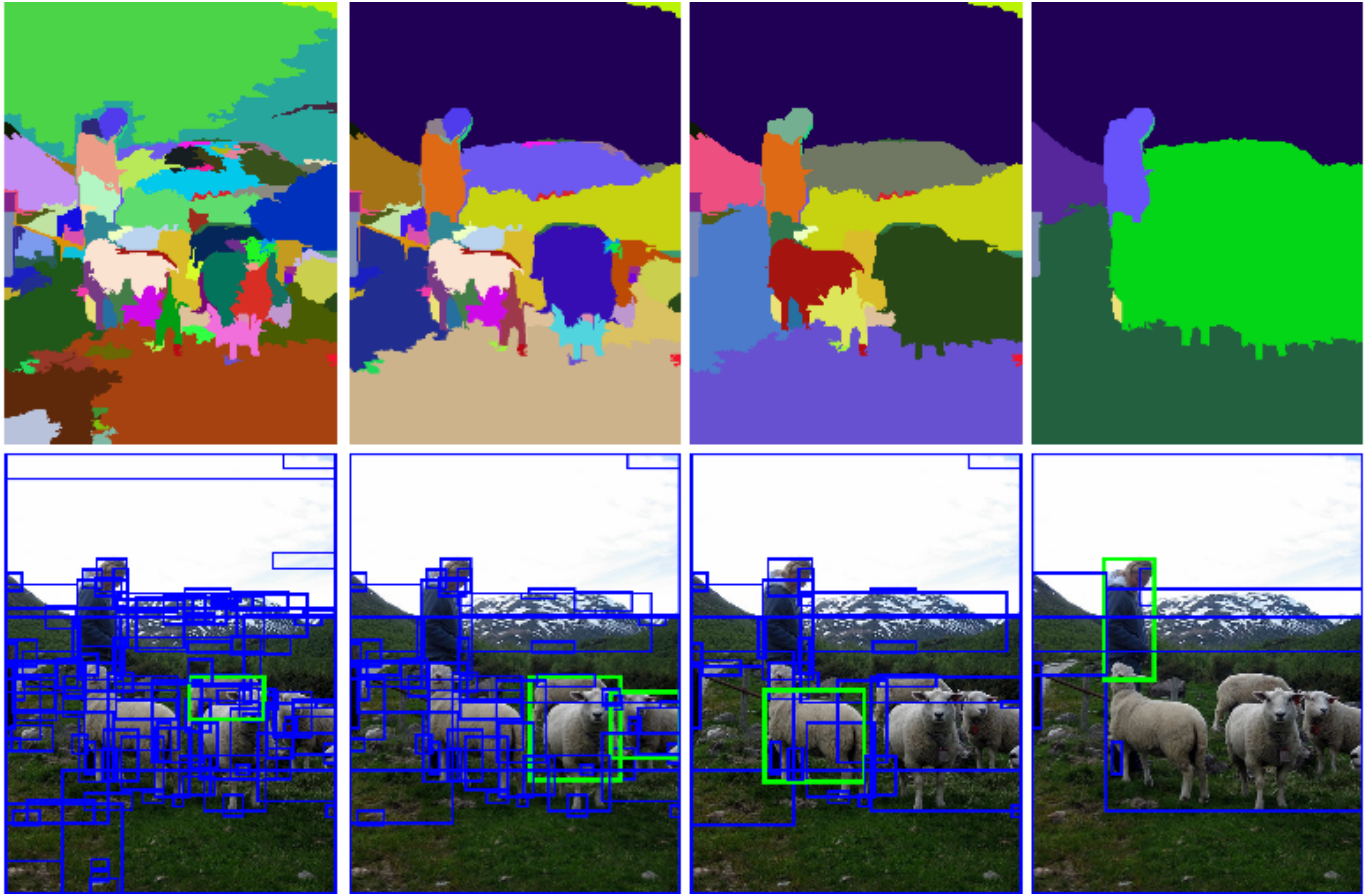
Proportion of the image area that a and b jointly occupy

Histogram intersection of 8-bin gradient direction histogram computed in each color channel

$S_{size}(a,b)$ → Encourages small regions to merge early and prevents single region from gobbling up all others one by one.

$S_{texture}(a,b)$ → Encourages regions with similar texture (and color) to be grouped early.

Example proposals



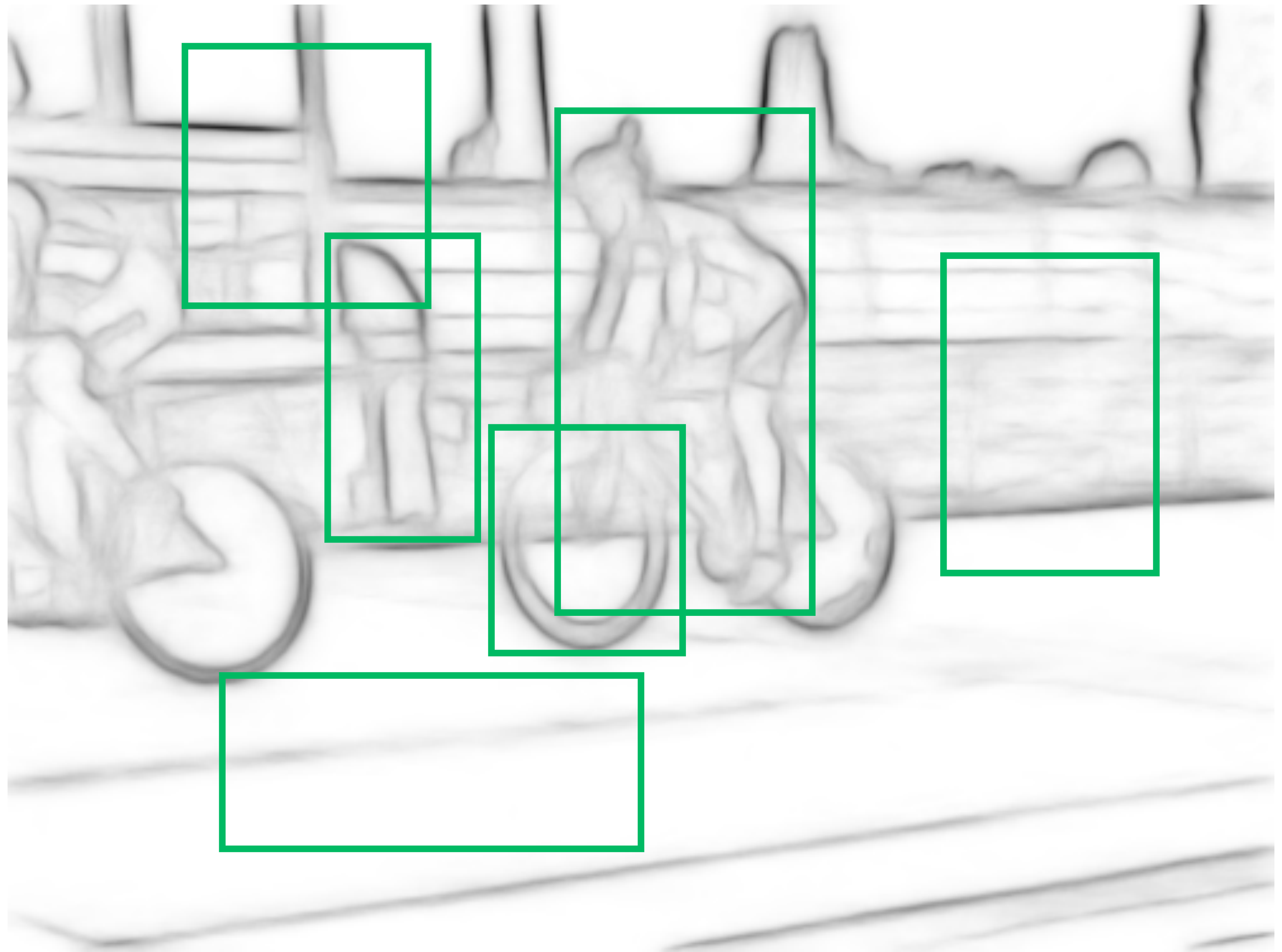
Example proposals

Another approach: “Objectness"

“What is an object?” Alexe et al., CVPR 2010

Learns to detect generic objects using simple color, texture, and edge features

Another approach: “Edge boxes”



Edge Boxes: Locating Object Proposals from Edges, Zitnick and Dollar, ECCV 2014

Number of contours that are fully contained (i.e., non-crossing) inside the box as the “objectness”

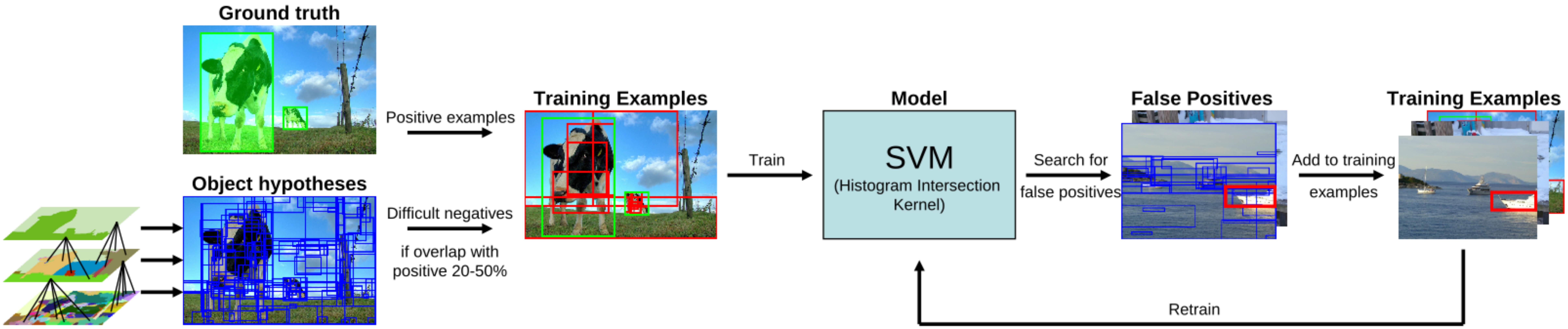
Very fast (0.25s per image on a CPU)

Detection using region proposals

Once again, detection = repeated classification

But we only classify object proposals

Training a classifier



Details of the features

HOG + linear classifiers were used in the DT detector for efficiency

But we can use complex features and better classifiers with regions

- In particular SIFT bag-of-words + non-linear SVMs
- Intersection Kernel SVMs (Maji, Berg & Malik, CVPR 2009)

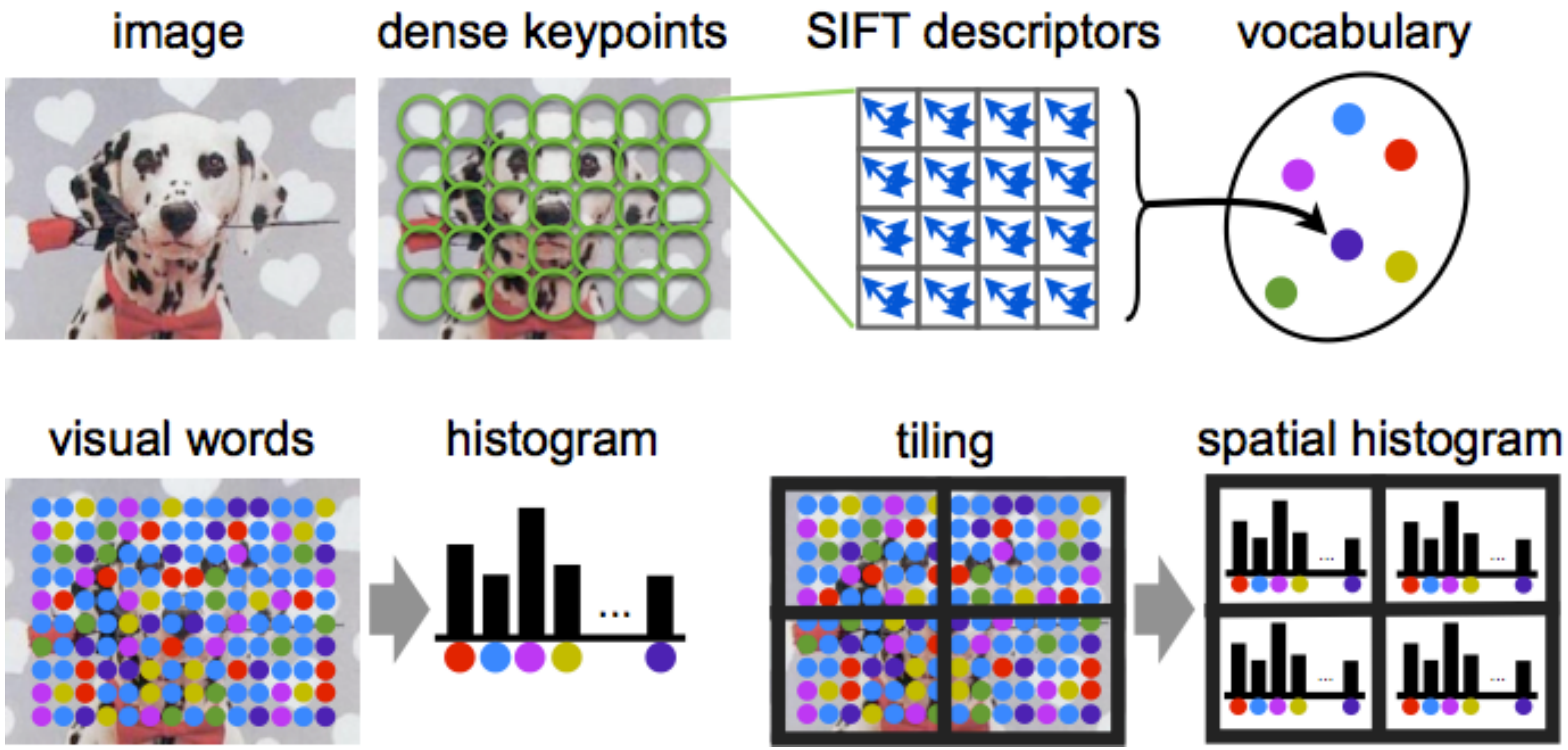
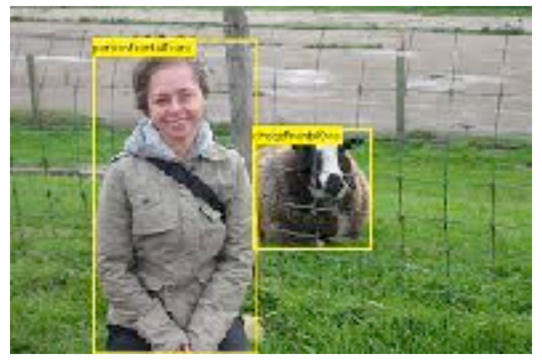
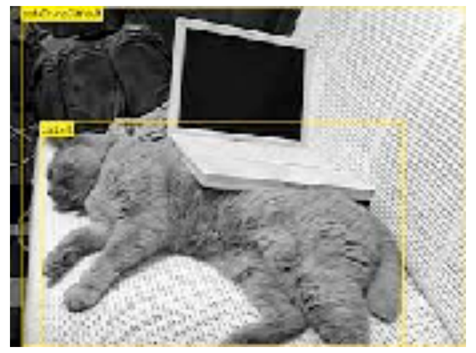
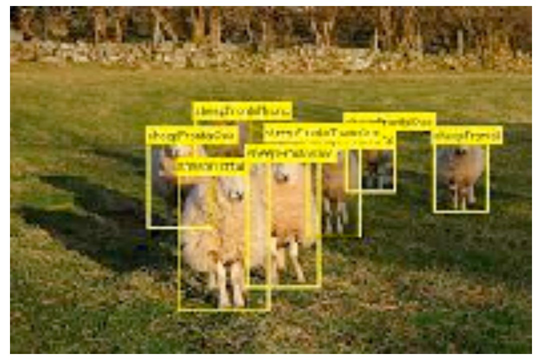


Image credit: Andrea Vedaldi

PASCAL VOC 2010 Detection

	“Shape”		“Texture”	
Method	**Person**	**Car**	**Cat**	**Sheep**
Poselets	**48.5%**	48.8%	22.2%	28.0%
DPM	43.3%	**49.1%**	31.1%	35.1%
Selective search	32.9%	36.8%	**46.1%**	**41.1%**



The quest for better features …

Rapid progress for a while followed by a plateaued

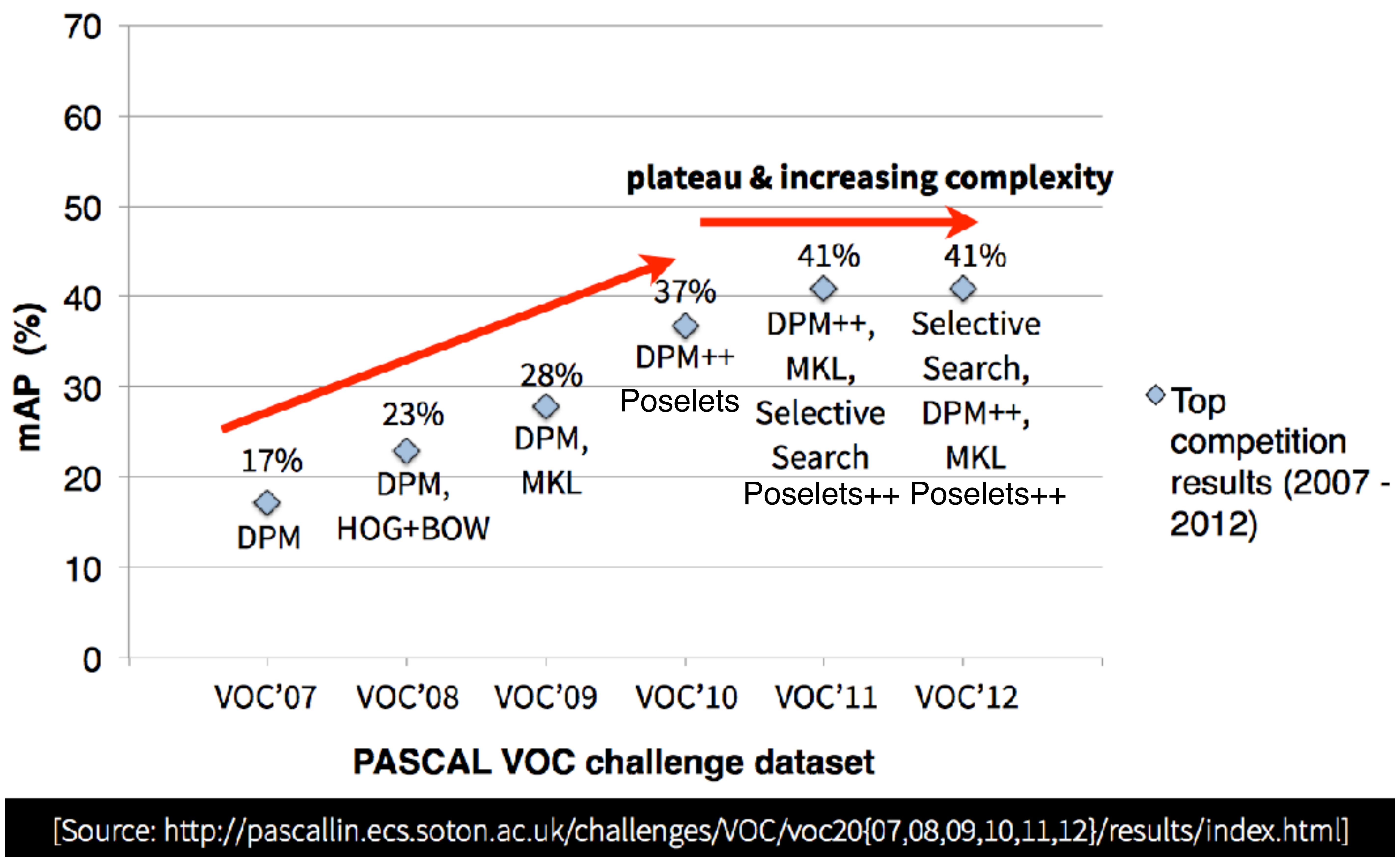
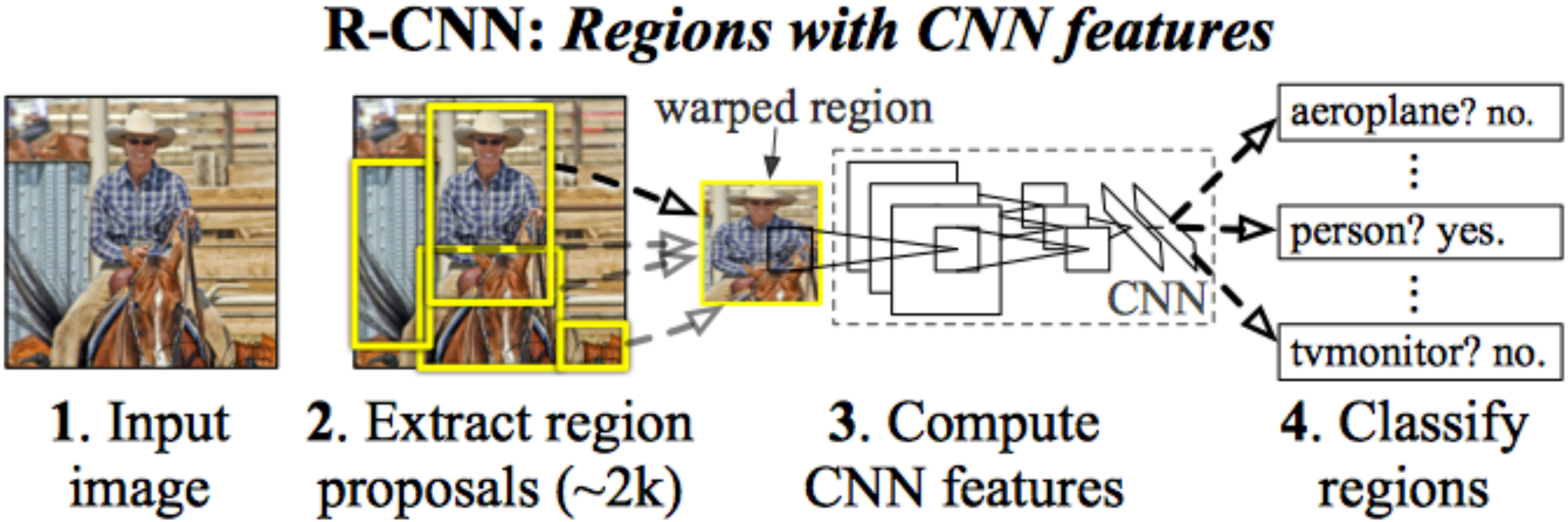


Figure by Ross Girshick

Breakthrough in object detection

R-CNNs (Girshick et al., CVPR 14) — regions with CNN features



Use ImageNet pre-trained CNNs to extract features!

R-CNN on PASCAL VOC

	VOC 2007	VOC 2010
DPM (Girshick et al. 2011)	33.7%	29.6%
UVA selective search (Uijlings et.al. 2013)		35.1%
R-CNN (Girshick et al. 2014)	54.2%	50.2%

average MAP across 20 categories

Slide credit: Ross Girshick

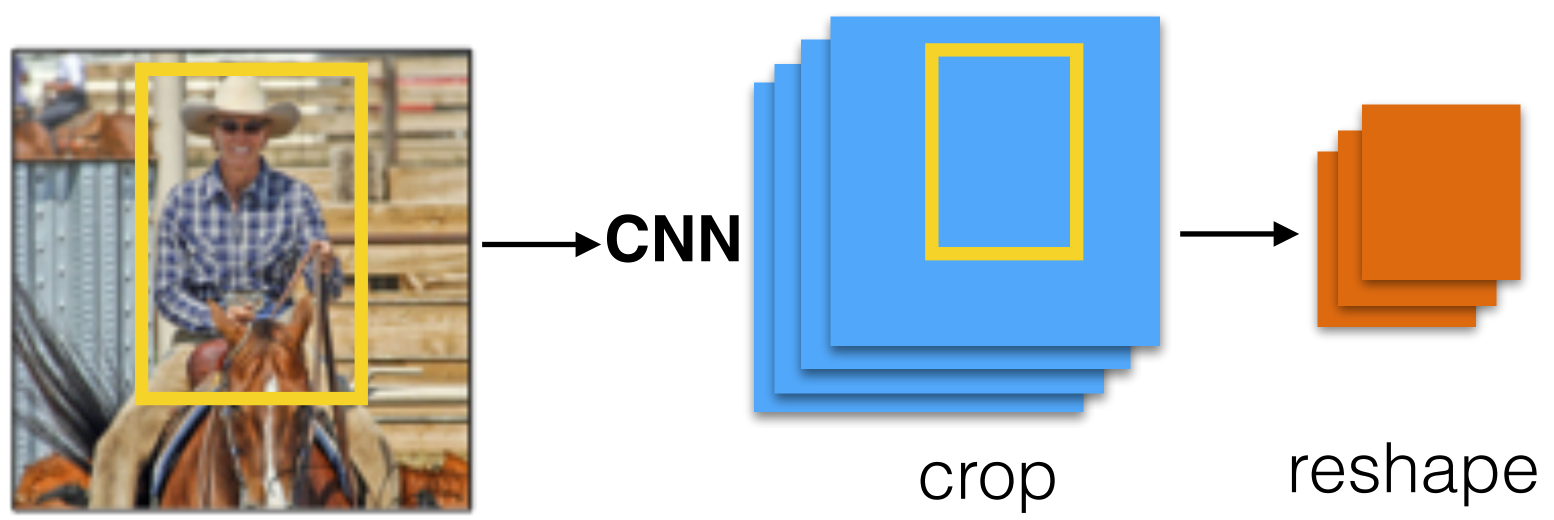
Current state of the art

Fast R-CNN [Girshick et al., 15]

- Reshape features instead of image

R-CNN

Fast R-CNN



Faster R-CNN [Ren et al. 15]

- Use the CNN backbone to propose regions (no external region proposal scheme)

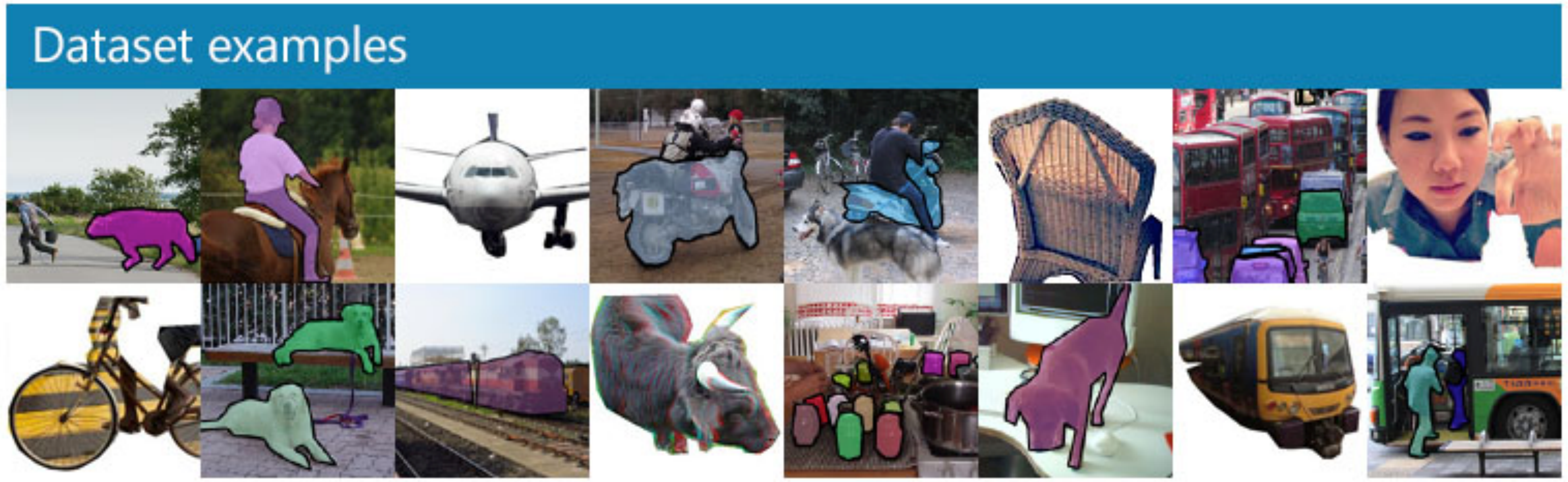
Single-Shot Detector (SSD) [Liu et al. 16]

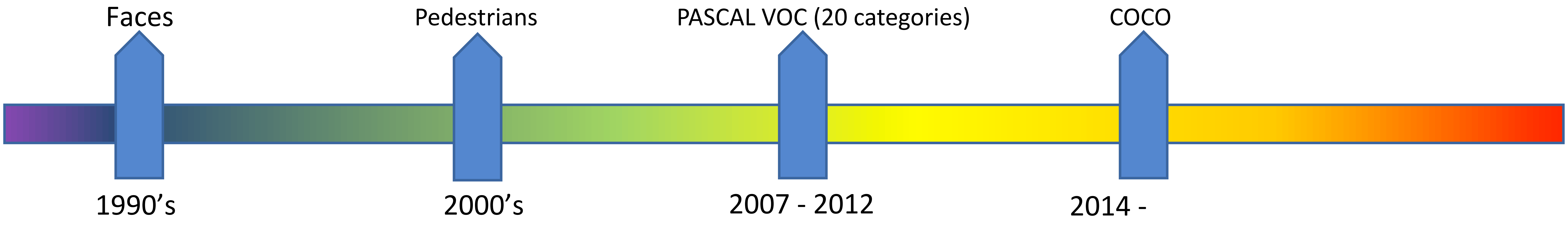
- Directly predict a list of bounding boxes and scores

Many other designs, including Transformers to replace CNNs

COCO dataset

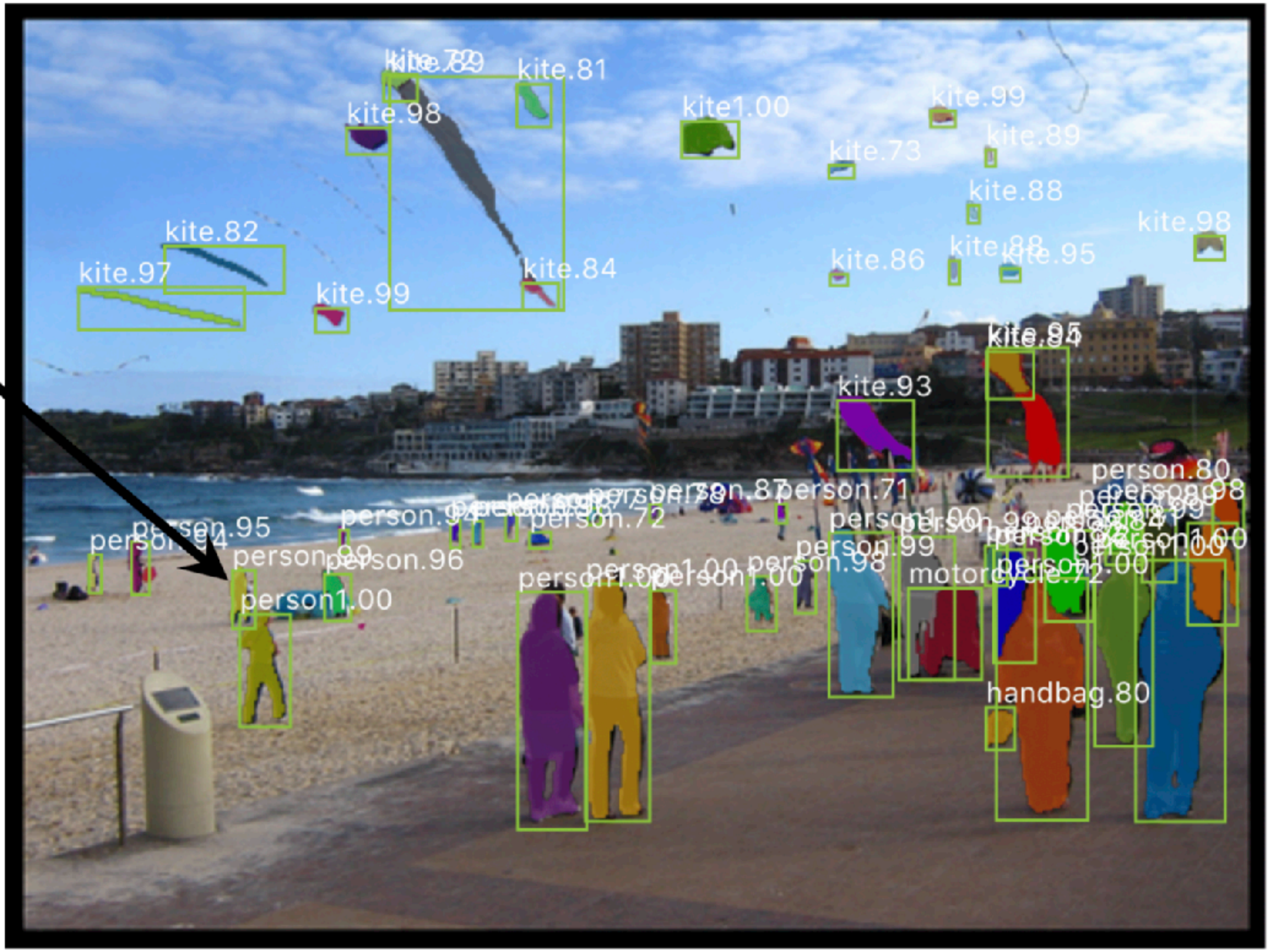
- 80 diverse categories
- 100K images
- Heavy occlusions, many objects per image, large scale variations





Mask R-CNN: Very Good Results!

small
objects



Mask R-CNN results on COCO

Slides credit

Some of the slides are by Ross Girshick, Andrea Vedaldi, Van de Sande, and others