北京大學
PEKING UNIVERSITY

Object Detection

Yadong Mu
Machine Intelligence Lab
Institute of Computer Science & Technology
Peking University

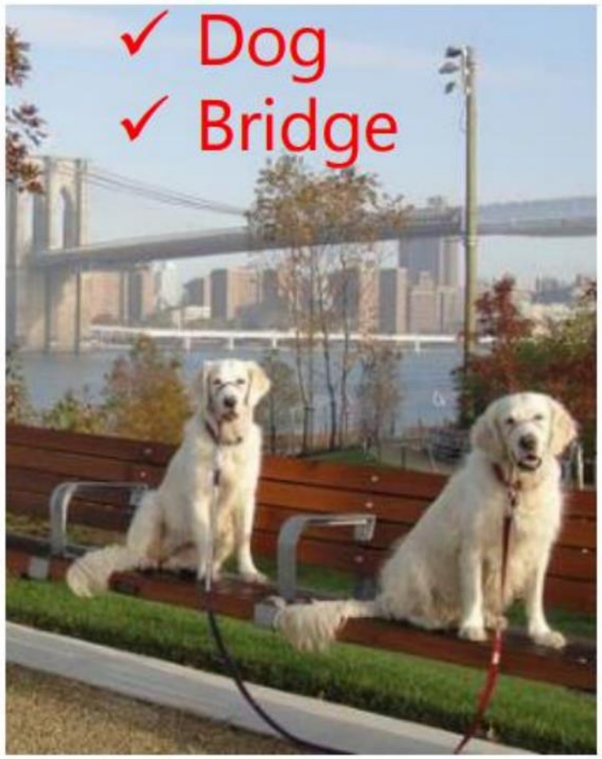
Email: myd@pku.edu.cn
Website: www.muyadong.com

Outline

- Object Detection: Task Specification
- Non-Deep Learning Methods:
  - Viola-Jones Face Detector
  - HOG + SVM
  - DPM
- Deep Learning Methods
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
  - Other Extensions

Image Classification v.s. Object Detection

Classification: WHAT



Detection: WHAT and WHERE

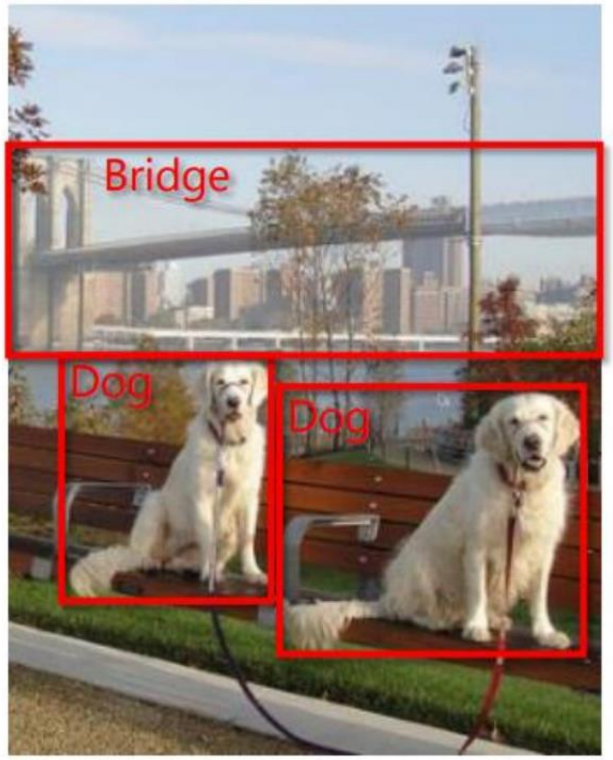
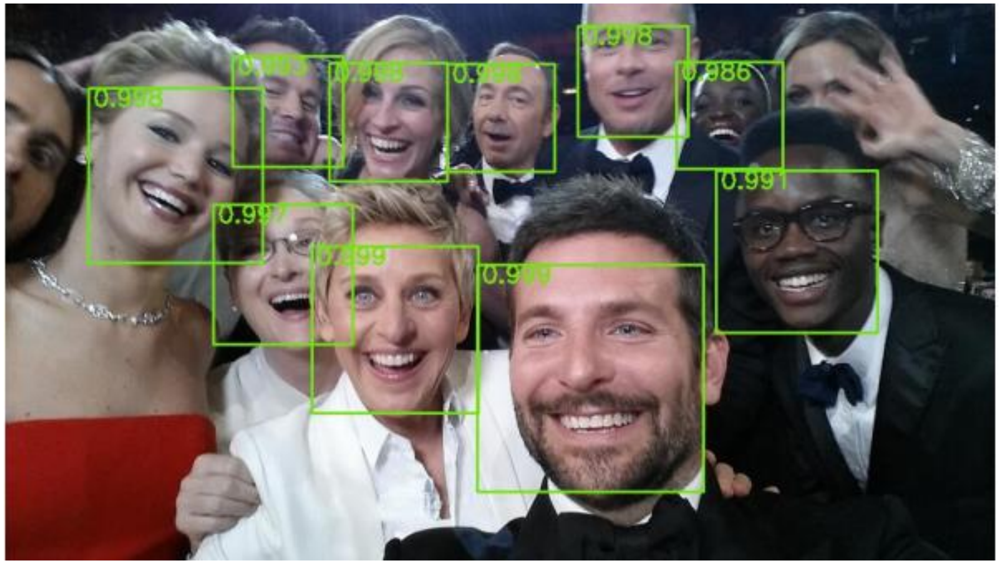


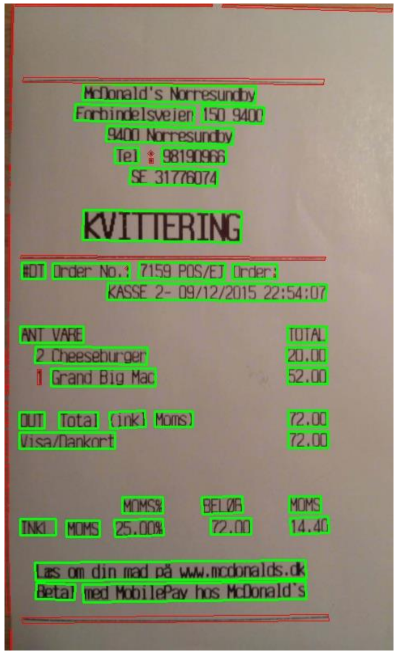
Image Classification v.s. Object Detection

- A more difficult problem than image classification
- Detection-by-classification is very popular and effective
- Key challenges
  - Image classification: discriminative feature representation
  - Object detection: balancing accuracy and efficacy

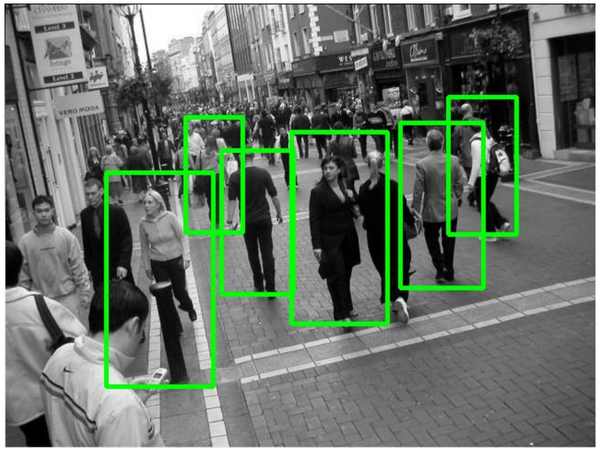
X Detection

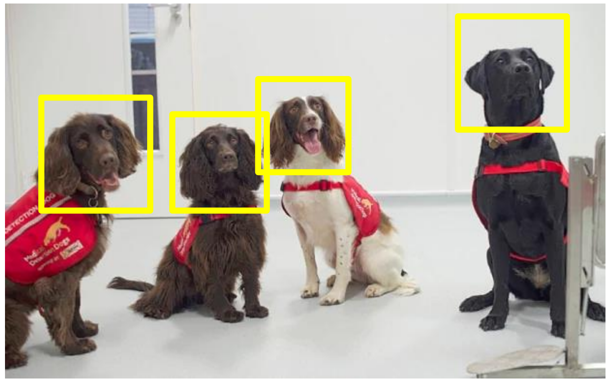
- Face, pedestrian, traffic sign, text etc.





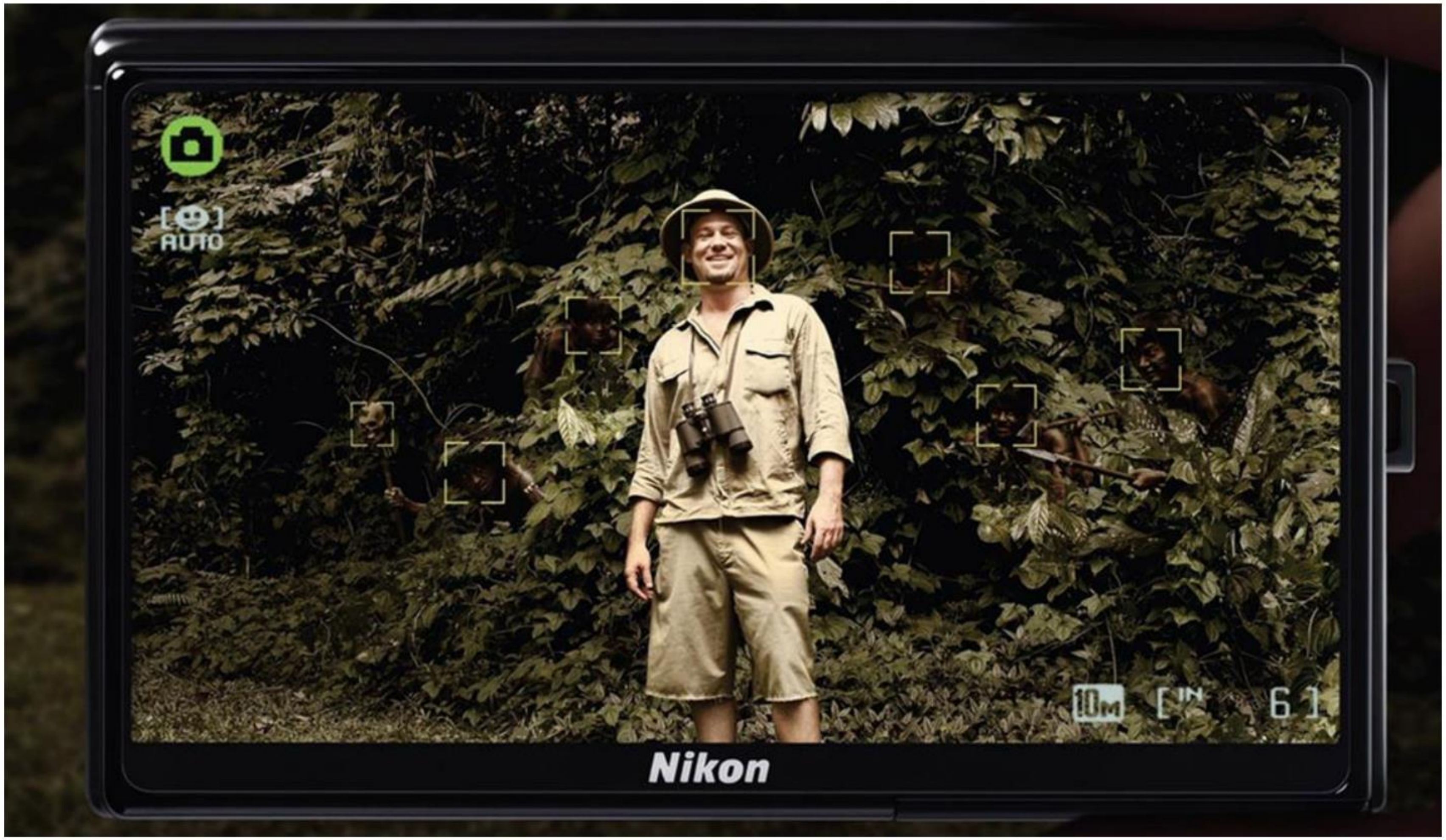
```
McDonald's Norresundby
Forbindelsveien 150 9400
9400 Norresundby
Tel : 98190966
SE 31776074

KVITTERING

#DT Order No.: 7159 POS/EJ Order:
KASSE 2- 09/12/2015 22:54:07

ANT VARE                          TOTAL
 2 Cheeseburger                   20.00
 1 Grand Big Mac                  52.00

OUT  Total (inkl Moms)            72.00
Visa/Dankort                      72.00

           MOMS%      BELØB       MOMS
INKL MOMS  25.00%      72.00      14.40

 Læs om din mad på www.mcdonalds.dk
 Betal med MobilePay hos McDonald's
```





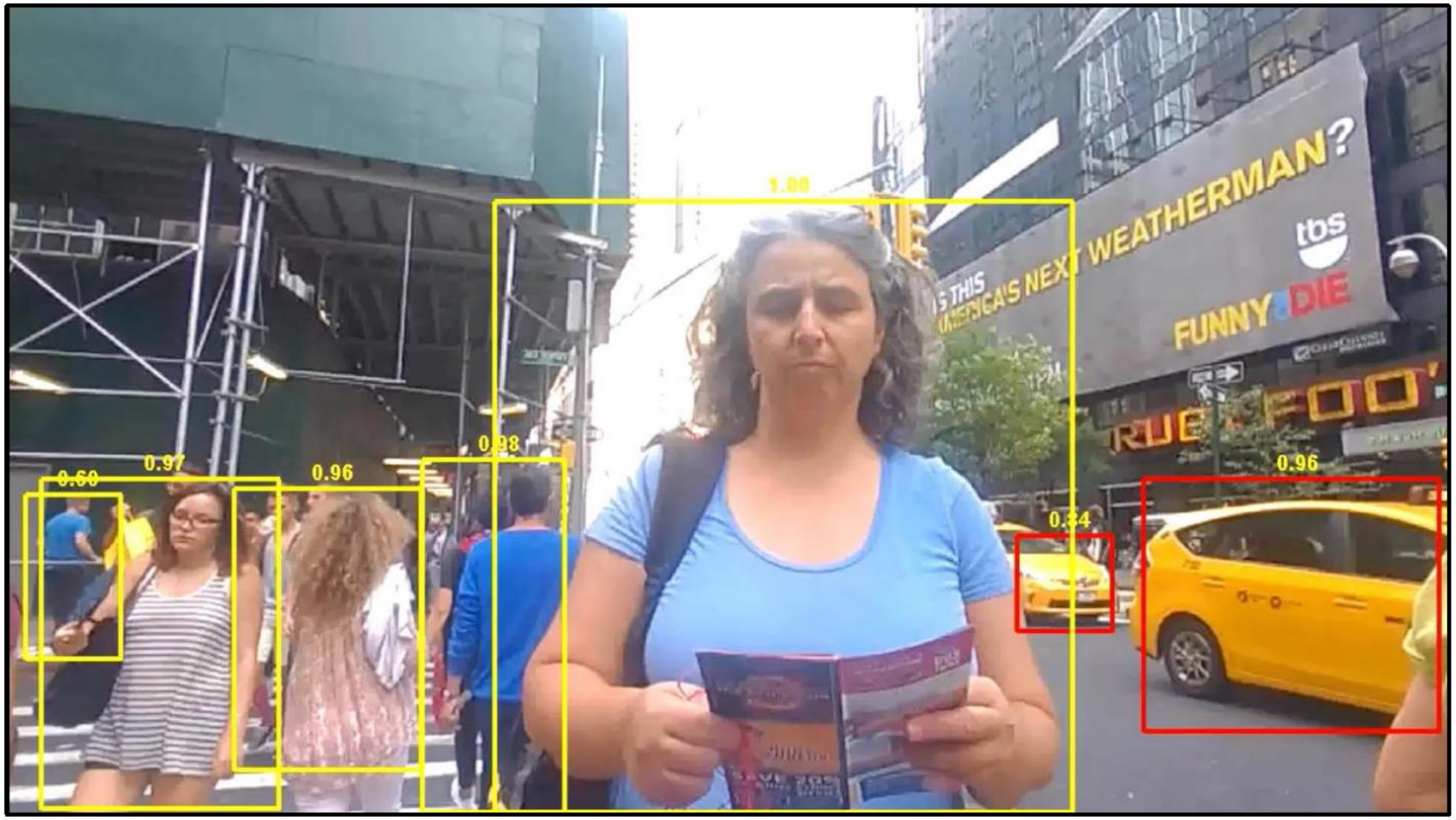
Applications

Funny Nikon ad: “"The Nikon S60 detects up to 12 faces."



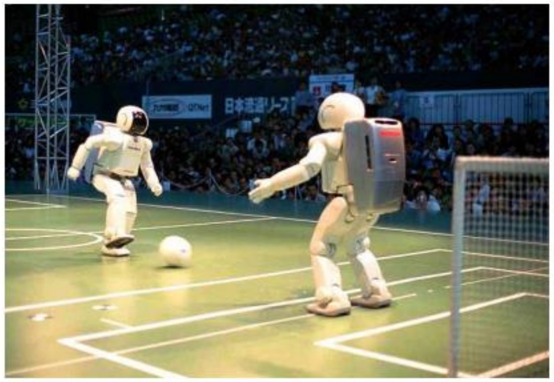
Applications

Lifelog video recording

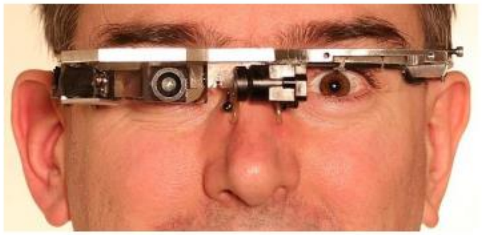


More Applications

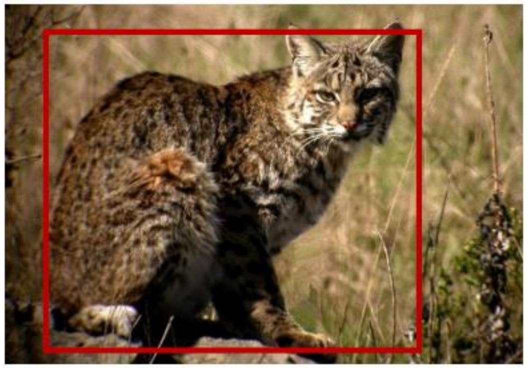
Robotics



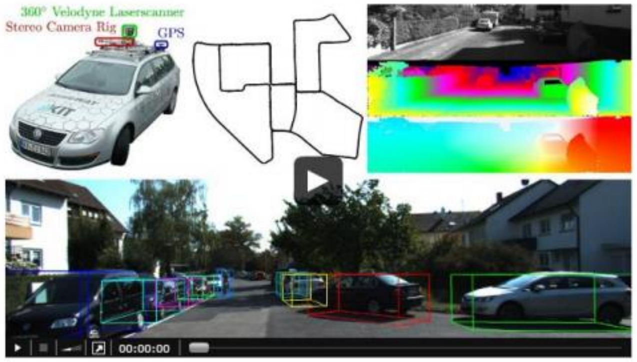
Augmented Reality



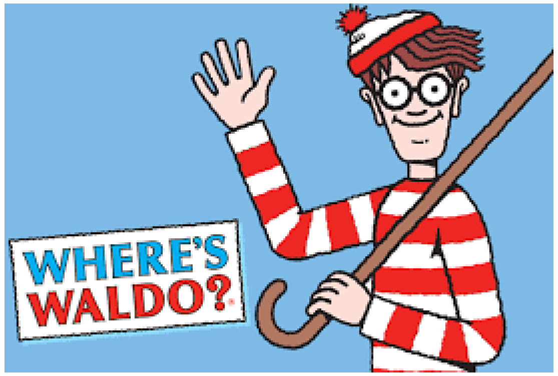
Wildlife Monitoring

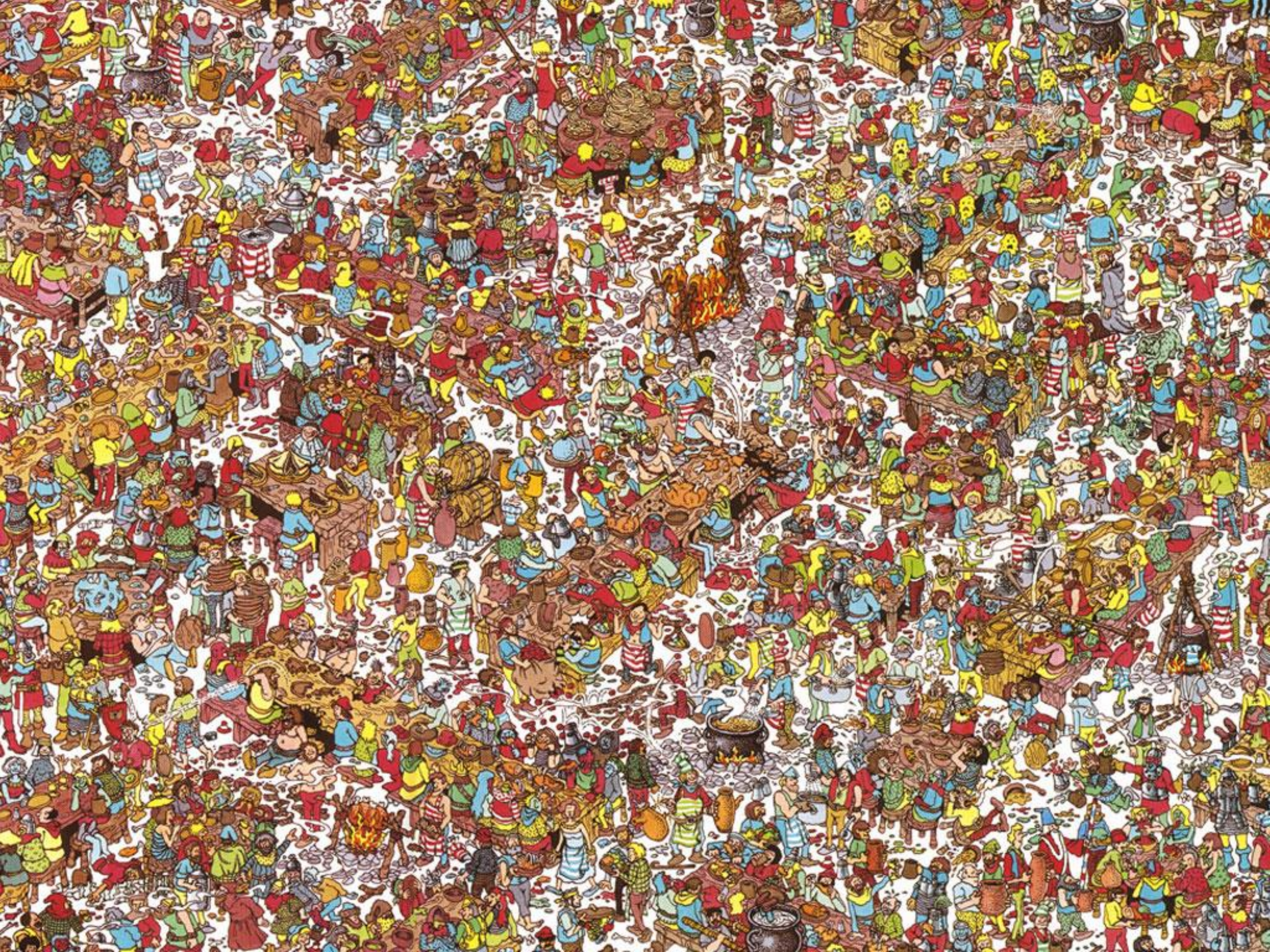


Self-Driving Cars



Find Waldo!



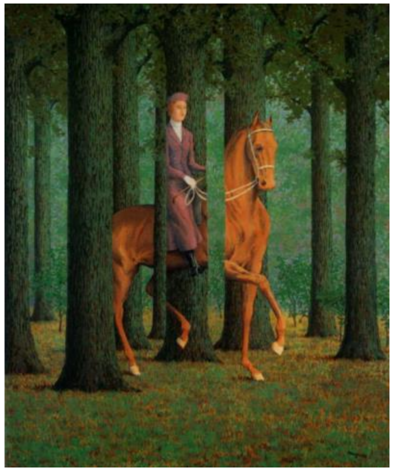


Challenges

view point variation

illumination

occlusion



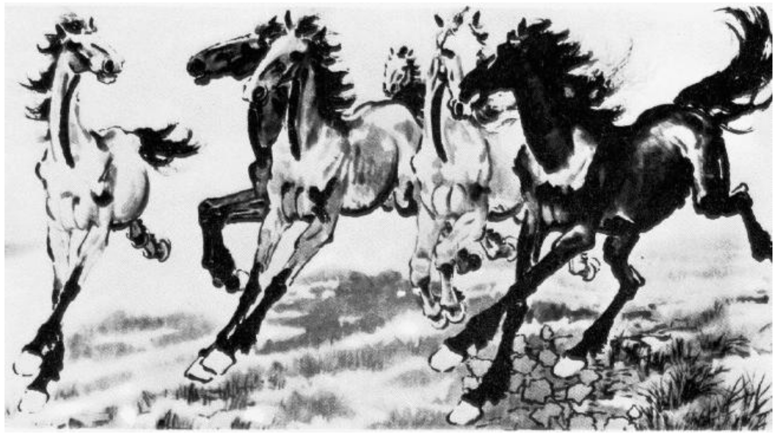
Magritte, 1957

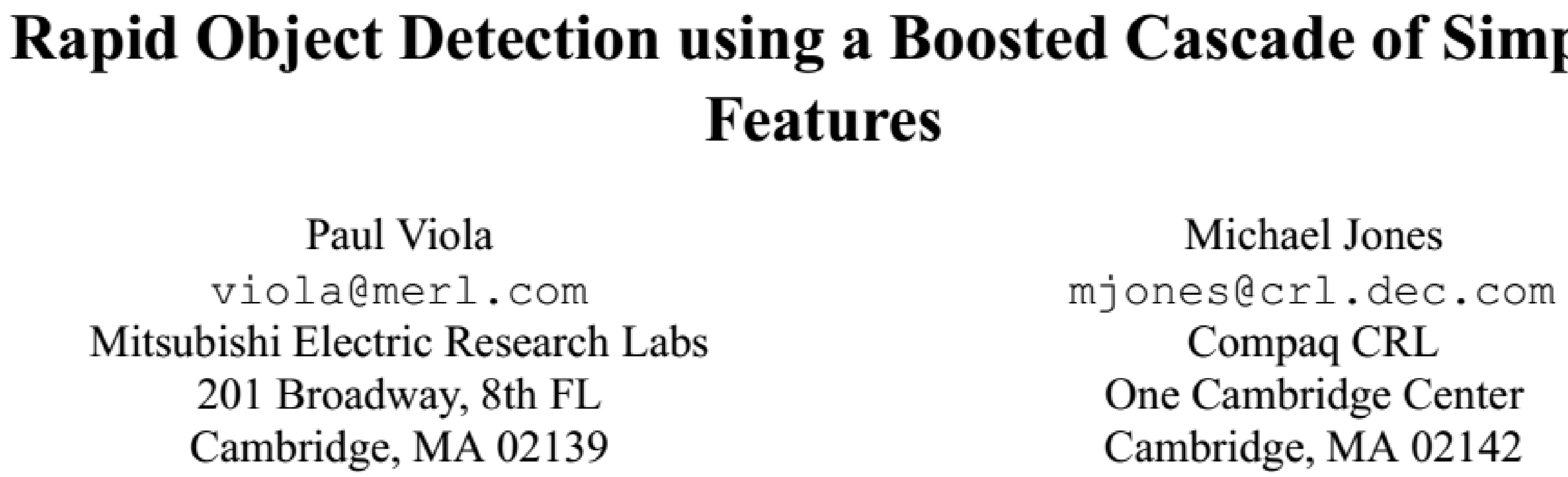
scale

Challenges

background clutter

deformation





Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

"…This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly. Operating on 384 by 288 pixel images, faces are detected at 15 frames per second on a conventional 700 MHz Intel Pentium III…"

Challenges of face detection

- Sliding window detector must evaluate tens of thousands of location/scale combinations
- Faces are rare: 0–10 per image
  - For computational efficiency, we should try to spend as little time as possible on the non-face windows
  - A megapixel image has ~10^6 pixels and a comparable number of candidate face locations
  - To avoid having a false positive in every image image, our false positive rate has to be less than 10^{-6}

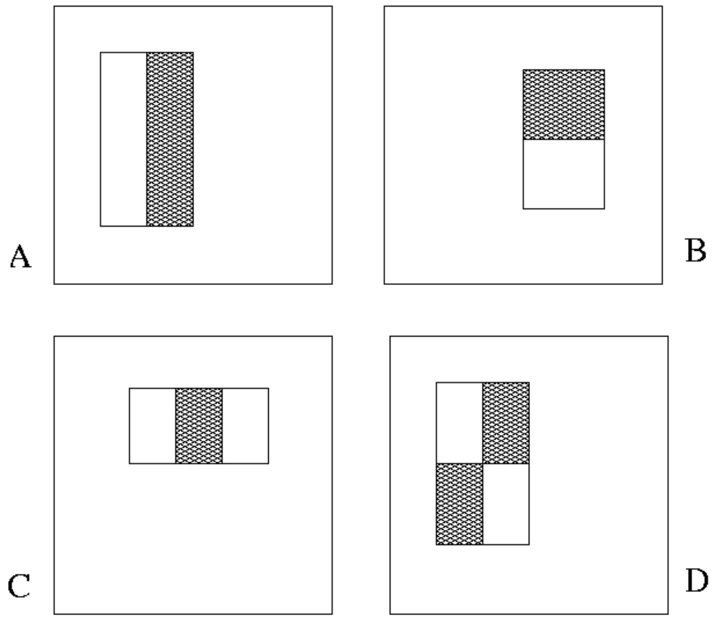
The Viola/Jones Face Detector

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* for fast rejection of non-face windows

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

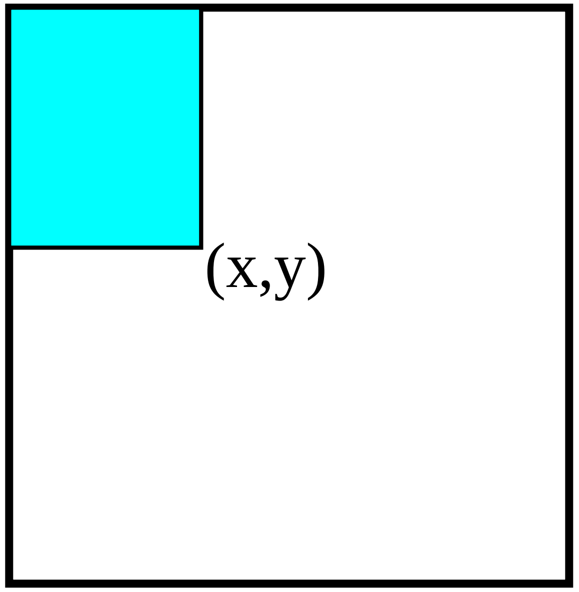
P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Image Features

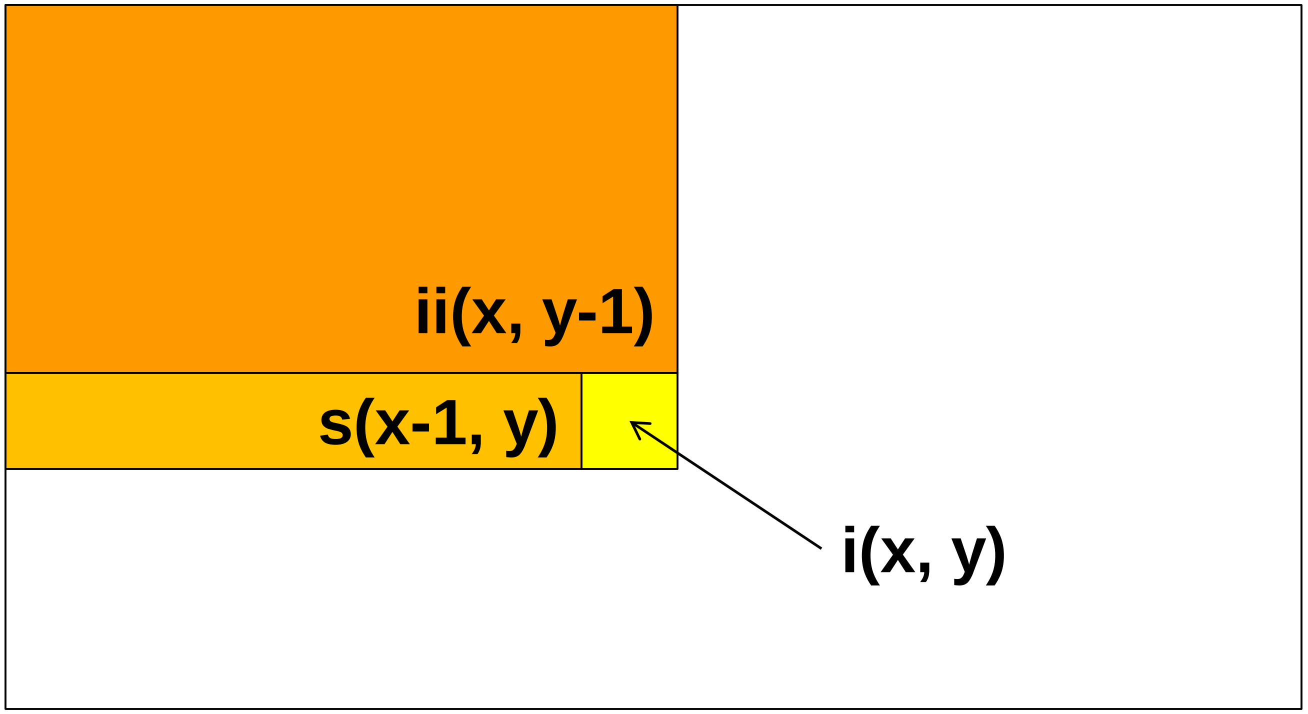


Fast computation with integral images

- The *integral image* computes a value at each pixel (*x*,*y*) that is the sum of the pixel values above and to the left of (*x*,*y*), inclusive
- This can quickly be computed in one pass through the image



Computing the integral image



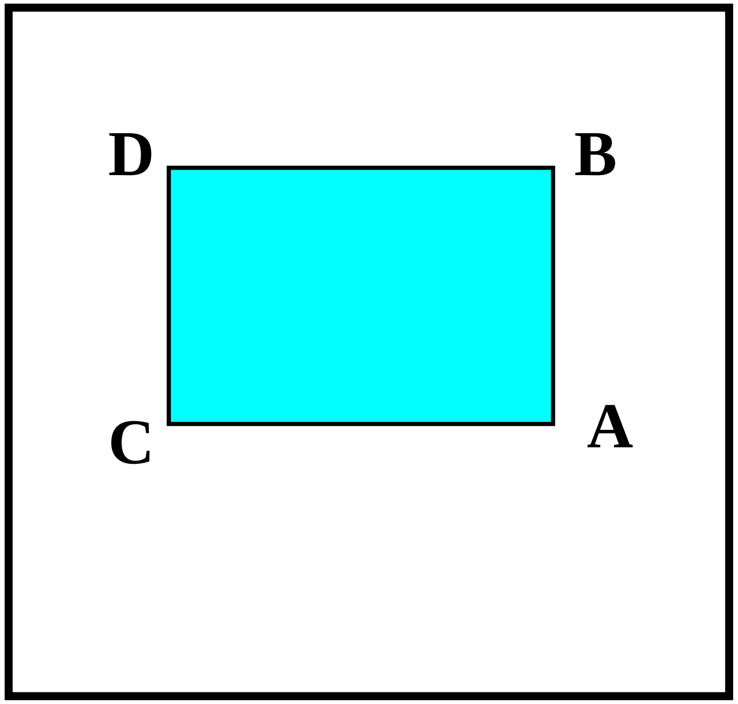
- Cumulative row sum: $s(x, y) = s(x-1, y) + i(x, y)$
- Integral image: $ii(x, y) = ii(x, y-1) + s(x, y)$

MATLAB: `ii = cumsum(cumsum(double(i)), 2);`

Computing sum within a rectangle

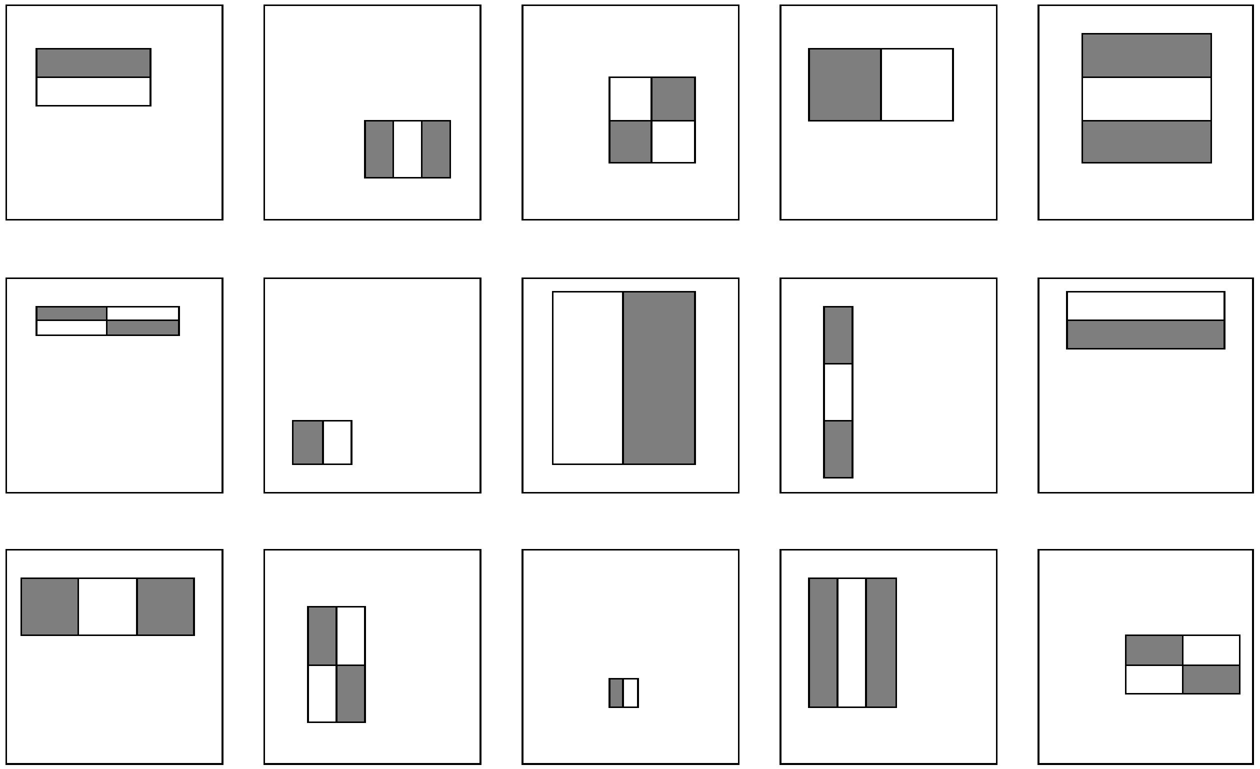
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

  sum = A – B – C + D
- Only 3 additions are required for any size of rectangle!



Feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!



Feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!
- At test time, it is impractical to evaluate the entire feature set
- Can we create a good classifier using just a small subset of all possible features?
- How to select such a subset?

Boosting

- Boosting is a classification scheme that works by combining *weak learners* into a more accurate ensemble classifier
  - A weak learner need only do better than chance
- Training consists of multiple *boosting rounds*
  - During each boosting round, we select a weak learner that does well on examples that were hard for the previous weak learners
  - "Hardness" is captured by weights attached to training examples

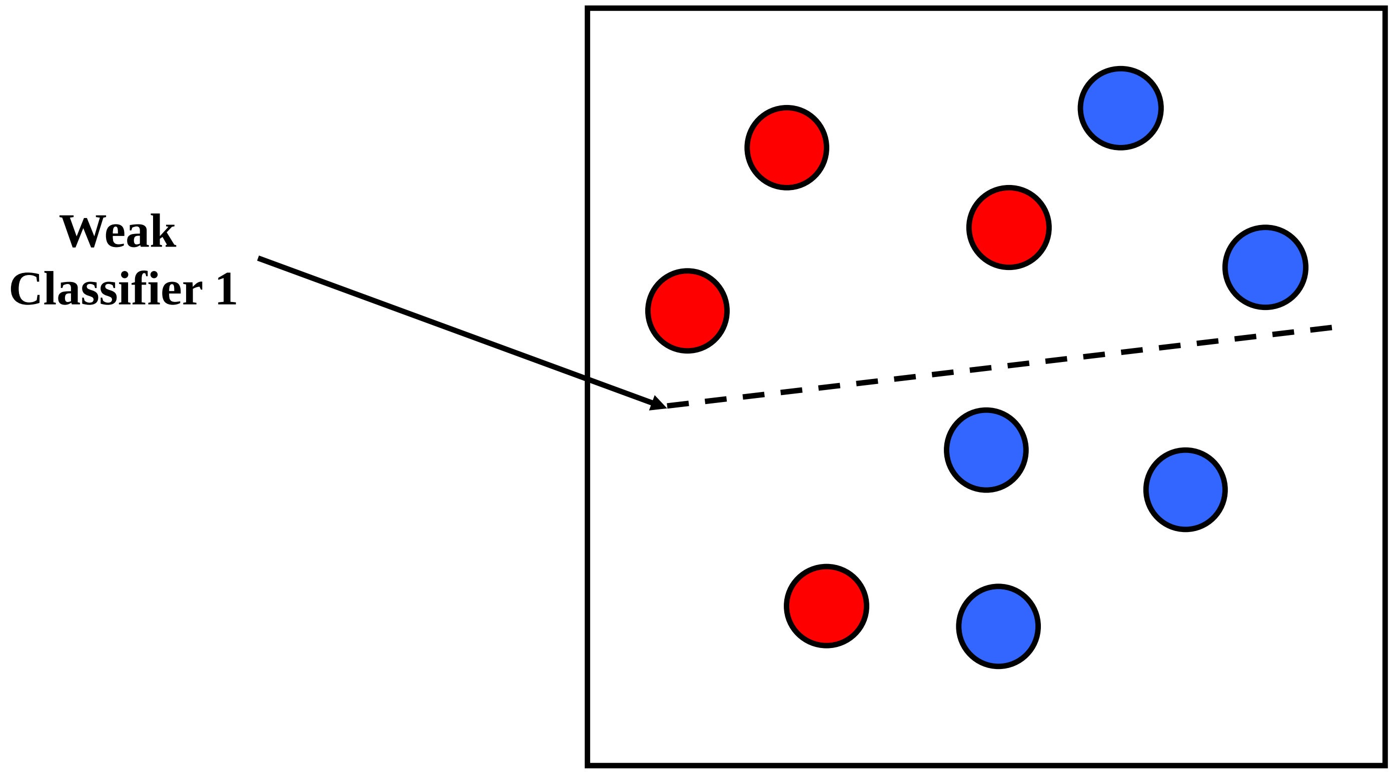
Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

Training procedure

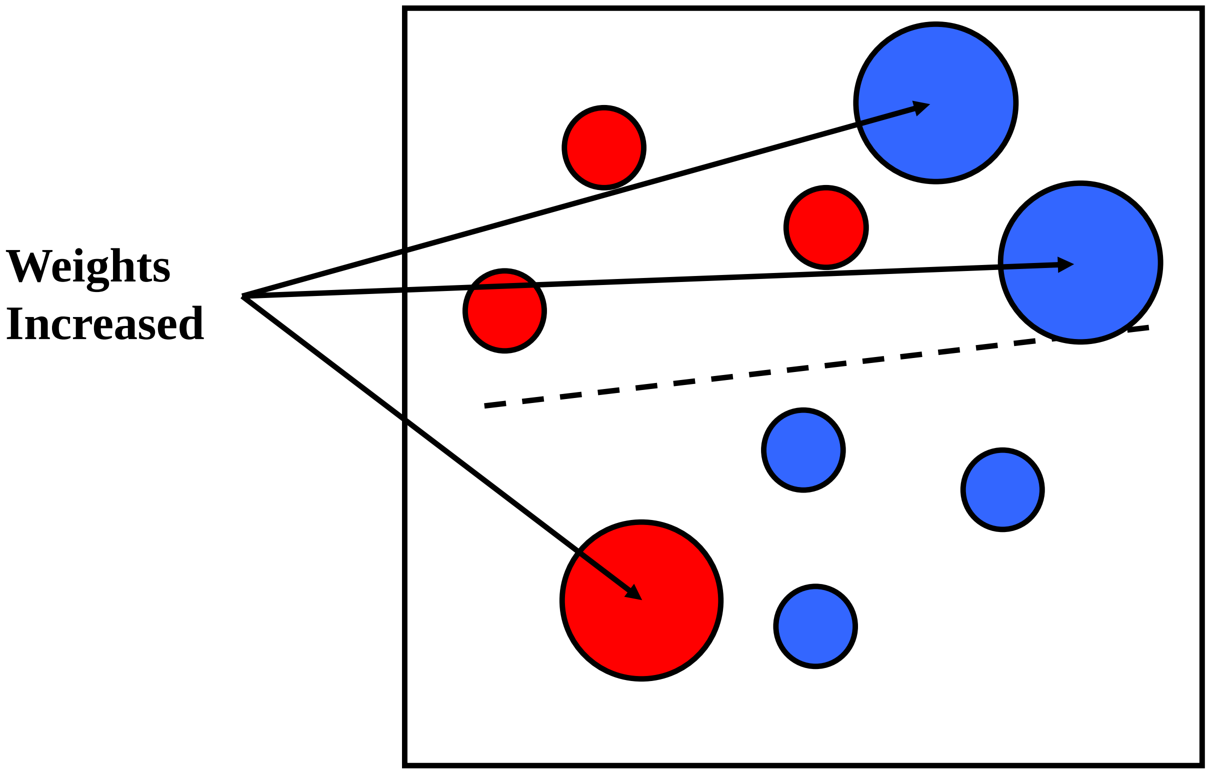
- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise the weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

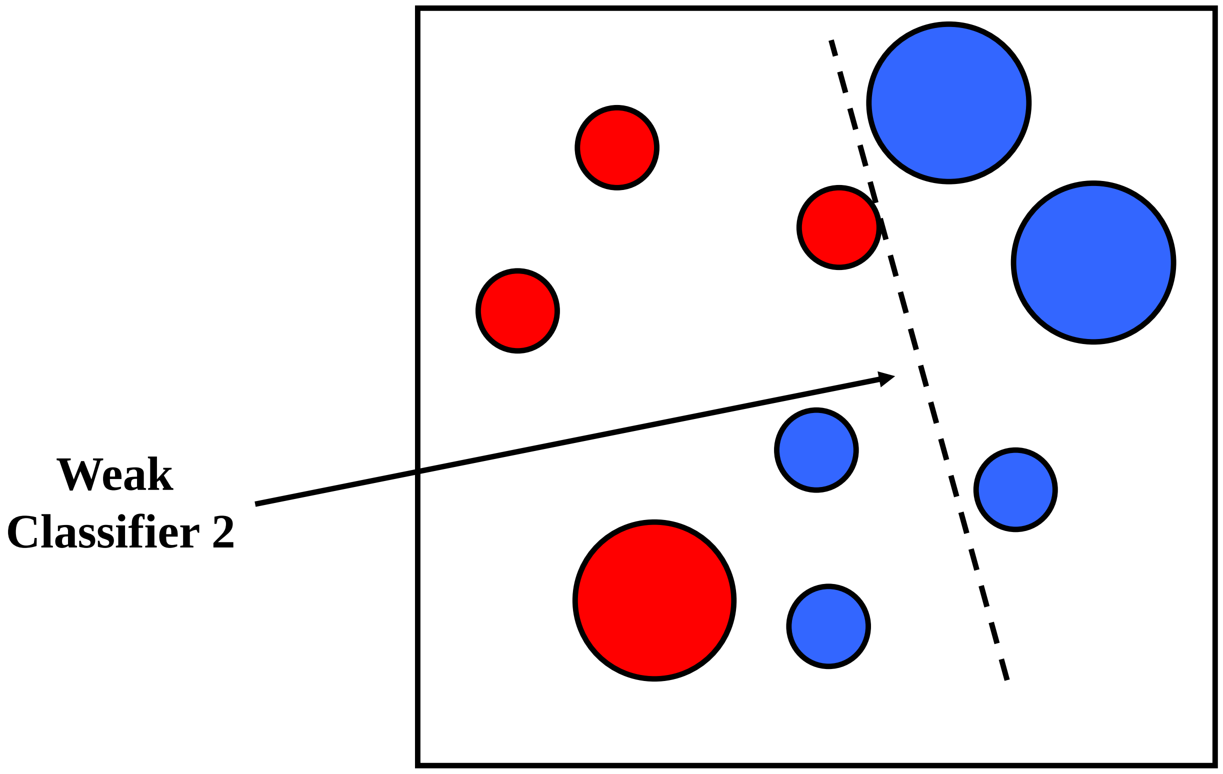
Boosting illustration



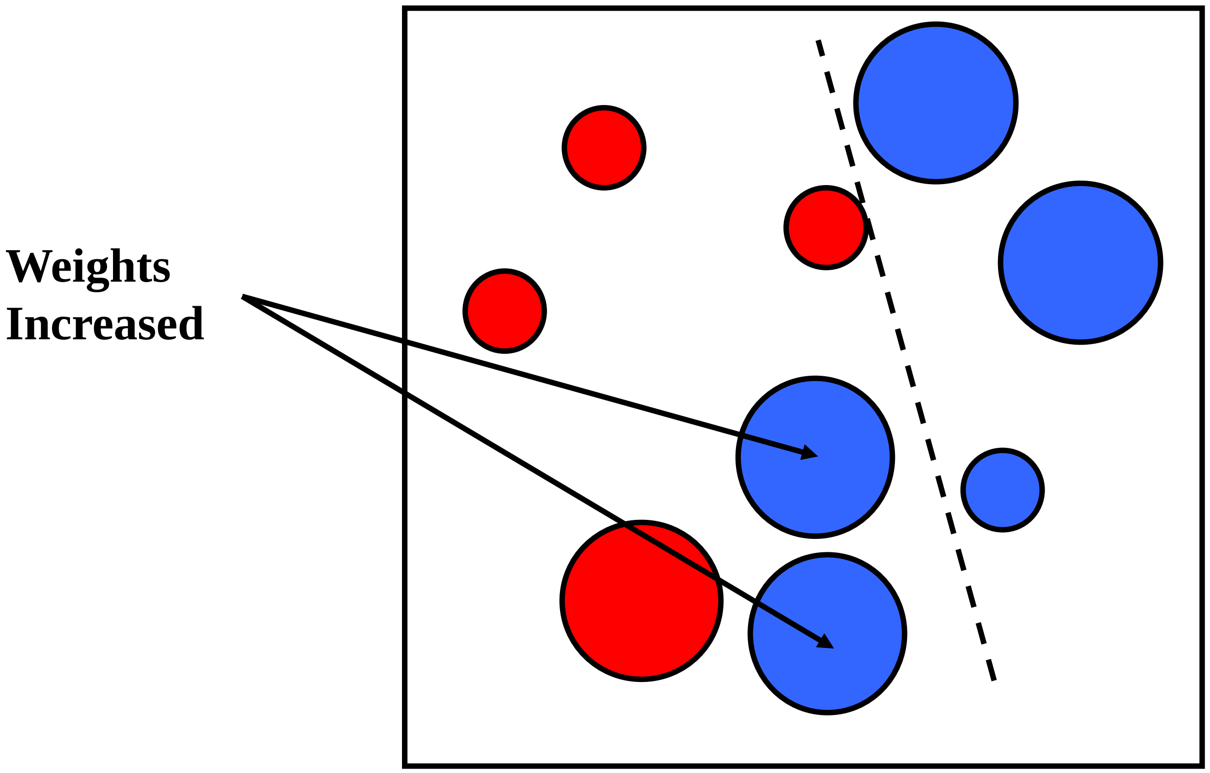
Boosting illustration



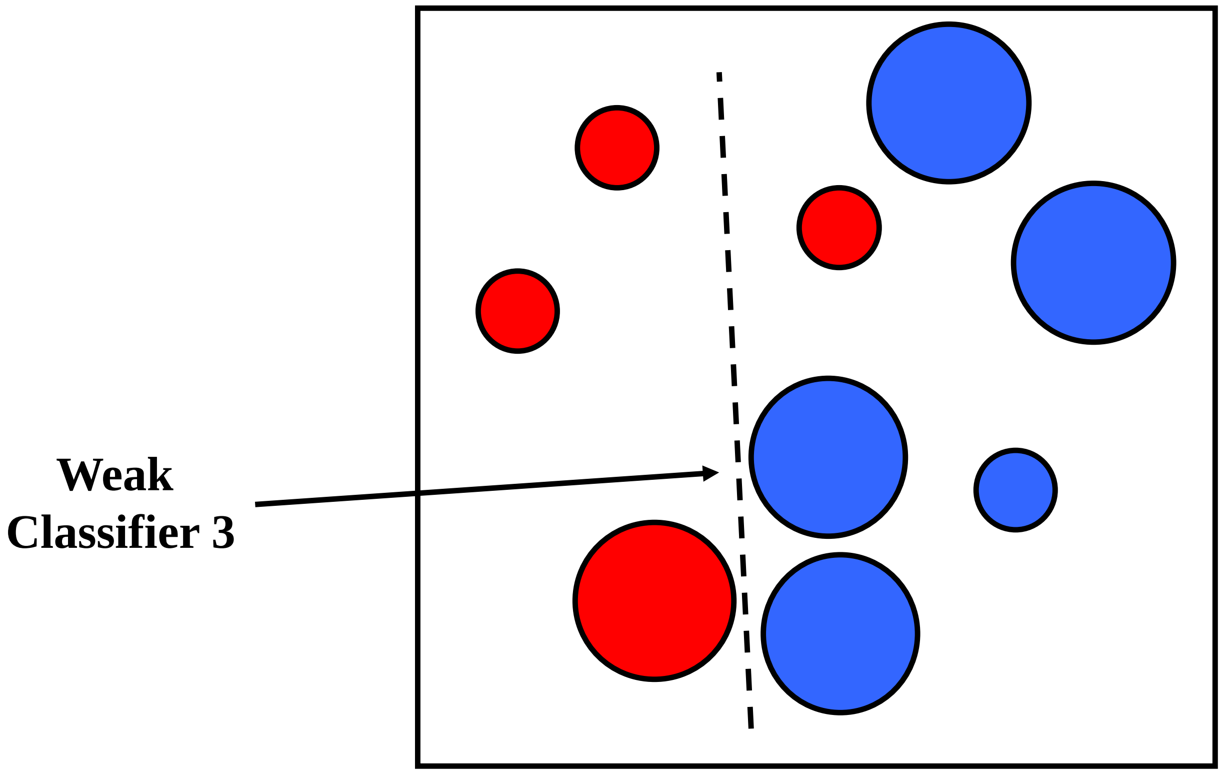
Boosting illustration



Boosting illustration

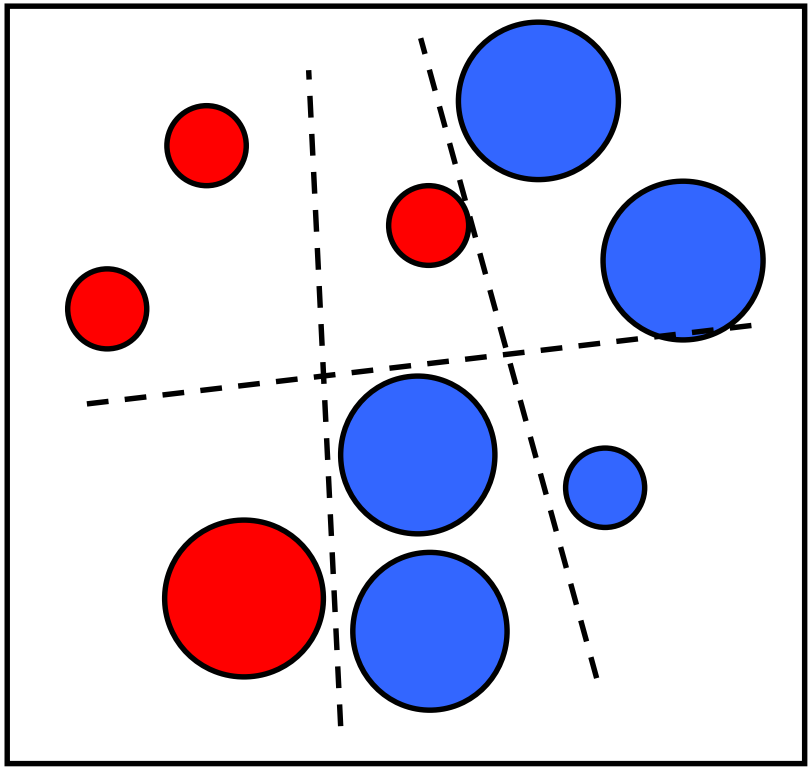


Boosting illustration



Boosting illustration

Final classifier is
a combination of weak
classifiers



Boosting vs. SVM

- Advantages of boosting
    - Integrates classification with feature selection
    - Complexity of training is linear instead of quadratic in the number of training examples
    - Flexibility in the choice of weak learners, boosting scheme
    - Testing is fast
    - Easy to implement
- Disadvantages
    - Needs many training examples
    - Often doesn’t work as well as SVM (especially for many-class problems)

Boosting for face detection

- Define weak learners based on rectangle features

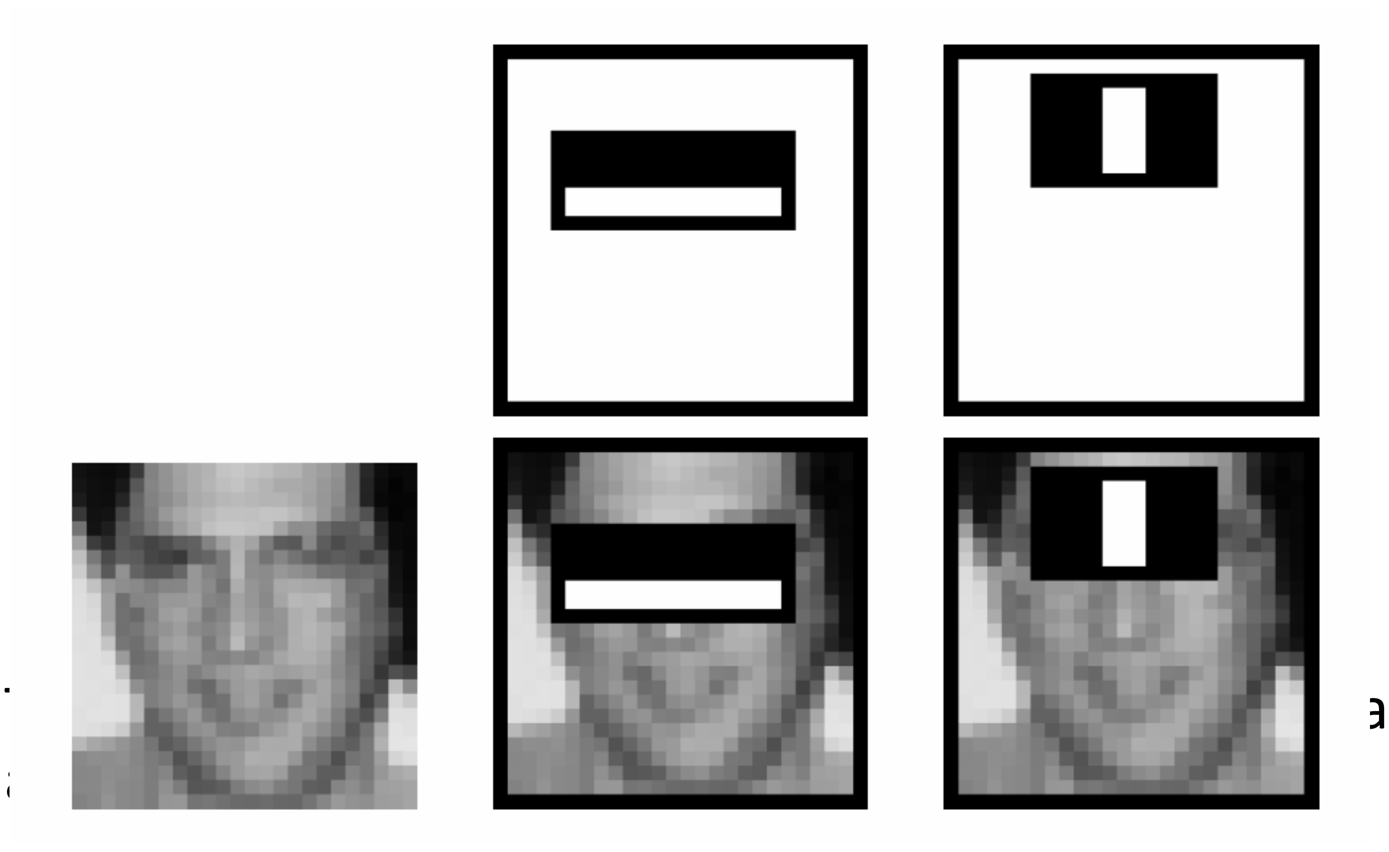
$$h_t(\underset{\uparrow \atop \text{window}}{x}) = \begin{cases} 1 & \text{if } \; p_t \overset{\text{value of rectangle feature} \atop \downarrow}{f_t}(x) > \underset{\uparrow \atop \text{parity}}{p_t} \underset{\nwarrow \atop \text{threshold}}{\theta_t} \\ 0 & \text{otherwise} \end{cases}$$

Boosting for face detection

- Define weak learners based on rectangle features
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Select best threshold for each filter
  - Select best filter/threshold combination
  - Reweight examples
- Computational complexity of learning: $O(MNK)$
  - M rounds, N examples, K features

Boosting for face detection

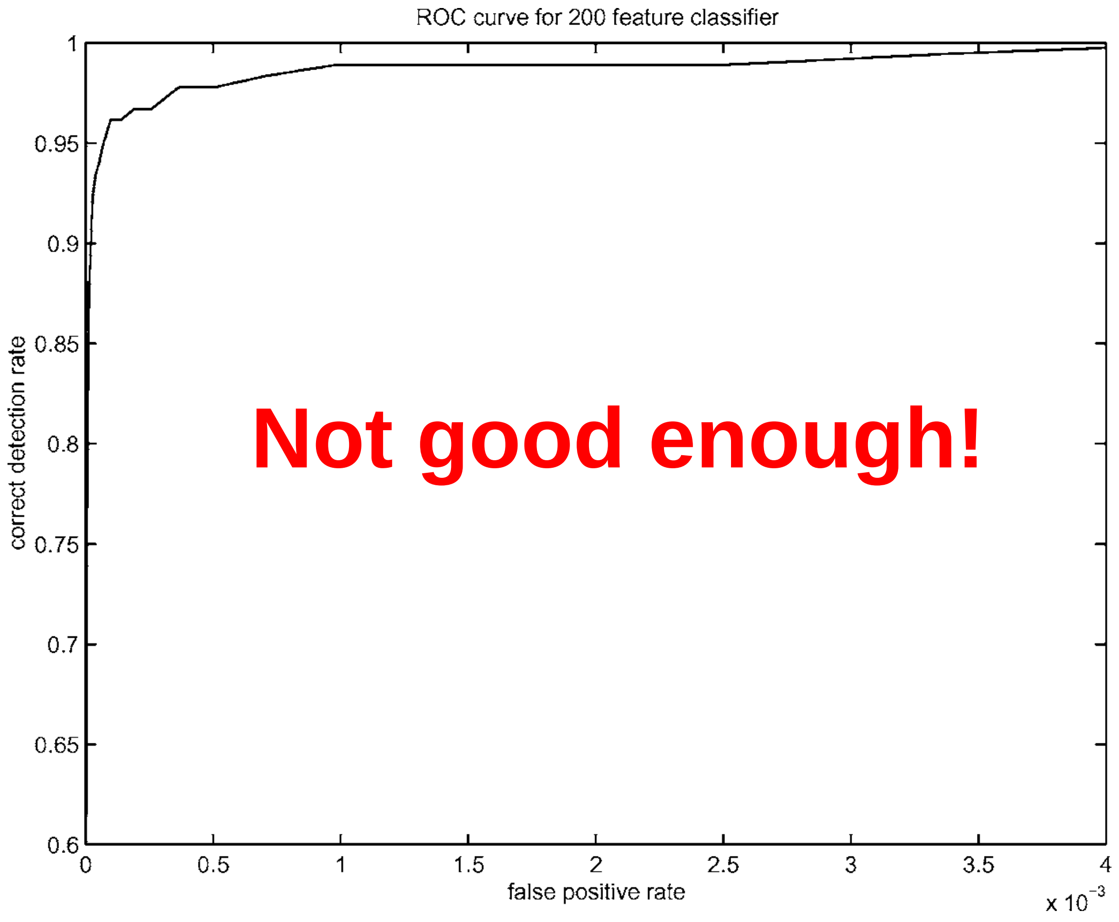
- First two features selected by boosting:



ate

Boosting for face detection

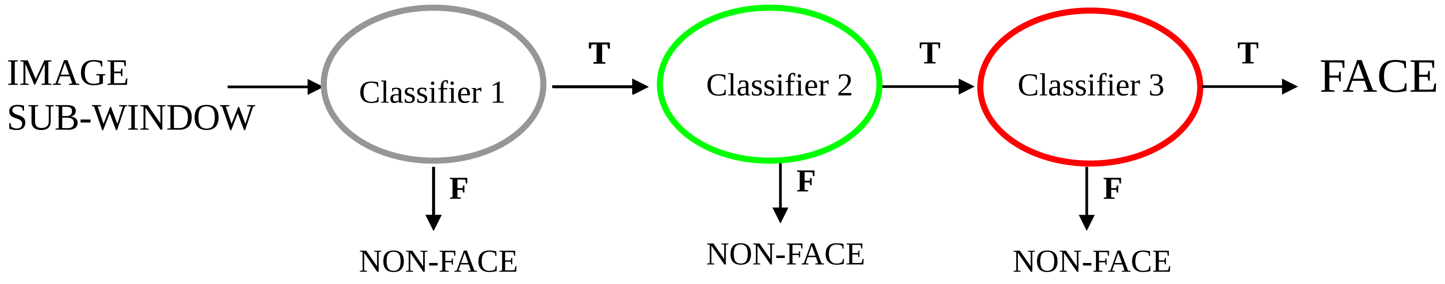
- A 200-feature classifier can yield 95% detection rate and a false positive rate of 1 in 14084



Receiver operating characteristic (ROC) curve

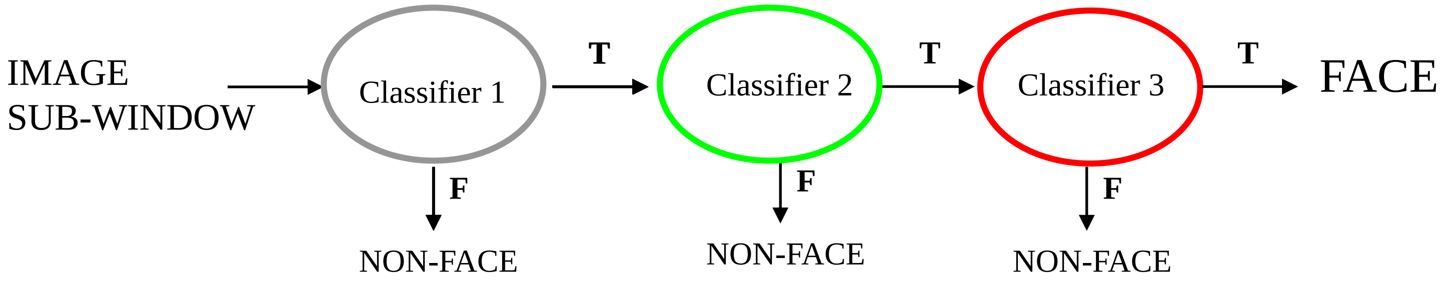
Attentional cascade

- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows
- Positive response from the first classifier triggers the evaluation of a second (more complex) classifier, and so on
- A negative outcome at any point leads to the immediate rejection of the sub-window



Attentional cascade

- The detection rate and the false positive rate of the cascade are found by multiplying the respective rates of the individual stages
- A detection rate of 0.9 and a false positive rate on the order of 10^{-6} can be achieved by a 10-stage cascade if each stage has a detection rate of 0.99 ($0.99^{10} \approx 0.9$) and a false positive rate of about 0.30 ($0.3^{10} \approx 6\times10^{-6}$)



Training the cascade

- Set target detection and false positive rates for each stage
- Keep adding features to the current stage until its target rates have been met
  - Need to lower AdaBoost threshold to maximize detection (as opposed to minimizing total classification error)
  - Test on a *validation set*
- If the overall false positive rate is not low enough, then add another stage
- Use false positives from current stage as the negative training examples for the next stage

The implemented system

- Training Data
  - 5000 faces
    - All frontal, rescaled to 24x24 pixels
  - 300 million non-faces
    - 9500 non-face images
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose

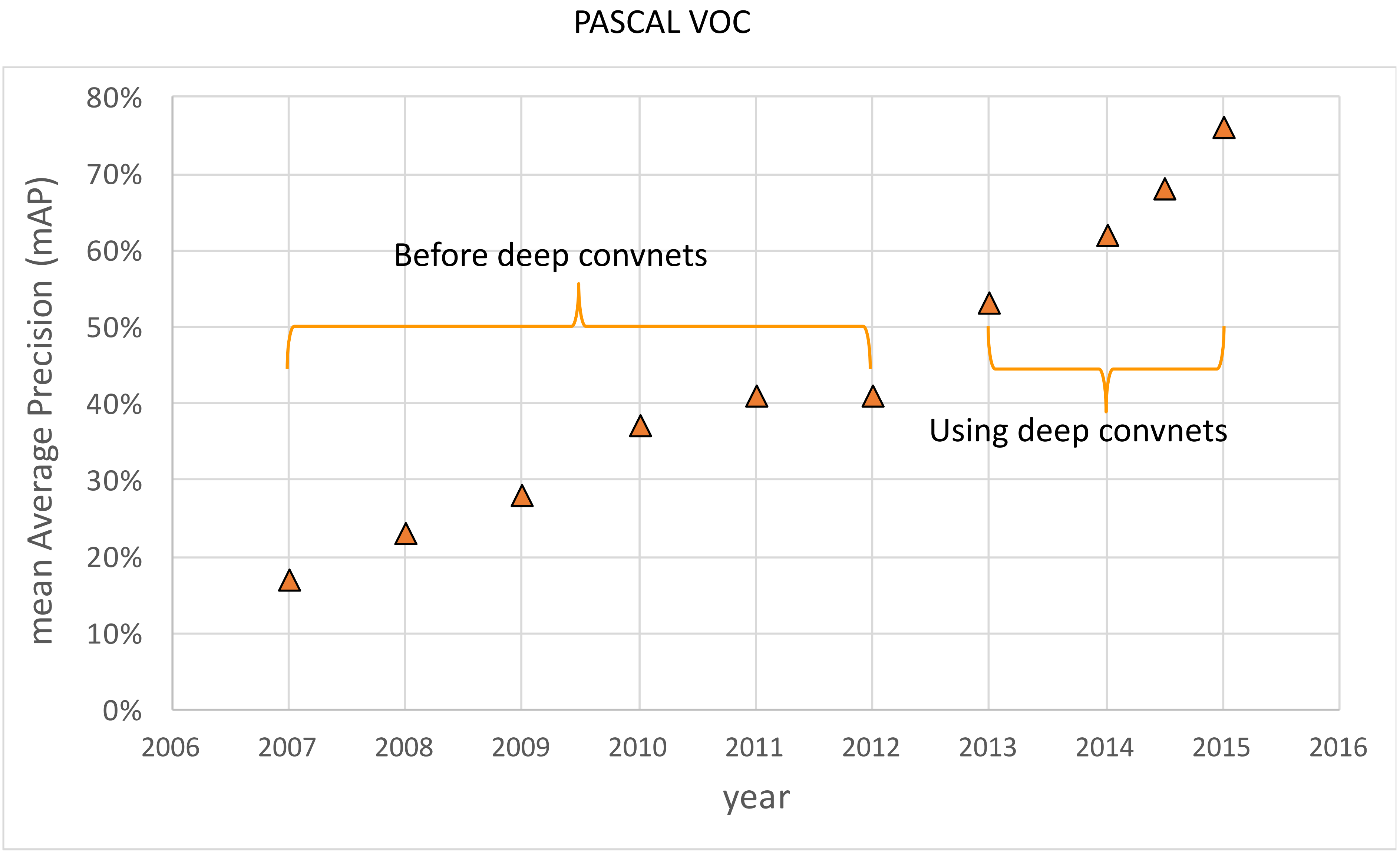
HOG + SVM

- See Sanja Fidler's slides
- http://www.cs.utoronto.ca/~fidler/CSC420.html

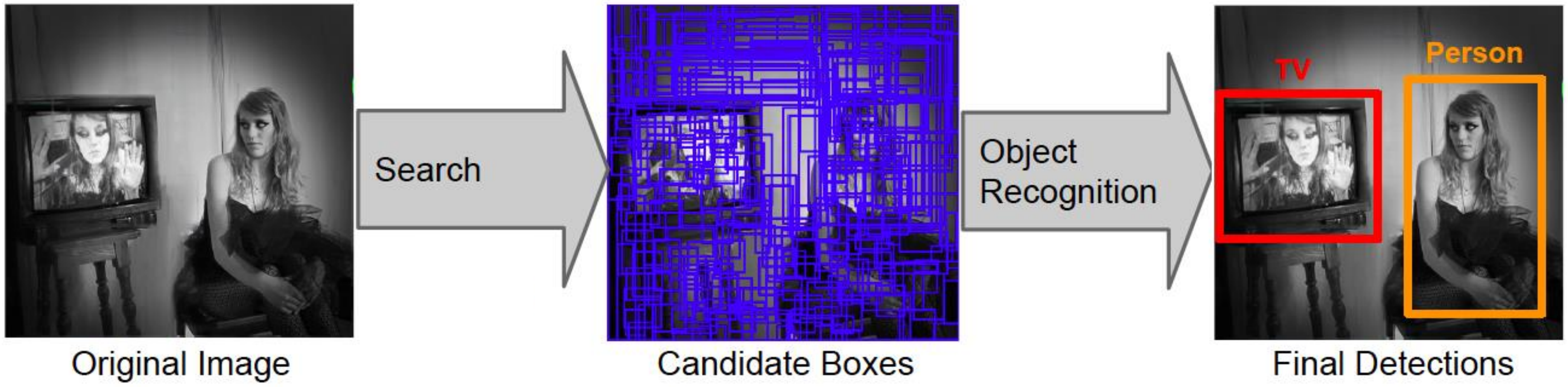
DPM

- See Sanja Fidler’s slides
- http://www.cs.utoronto.ca/~fidler/CSC420.html

Recent developments in object detection



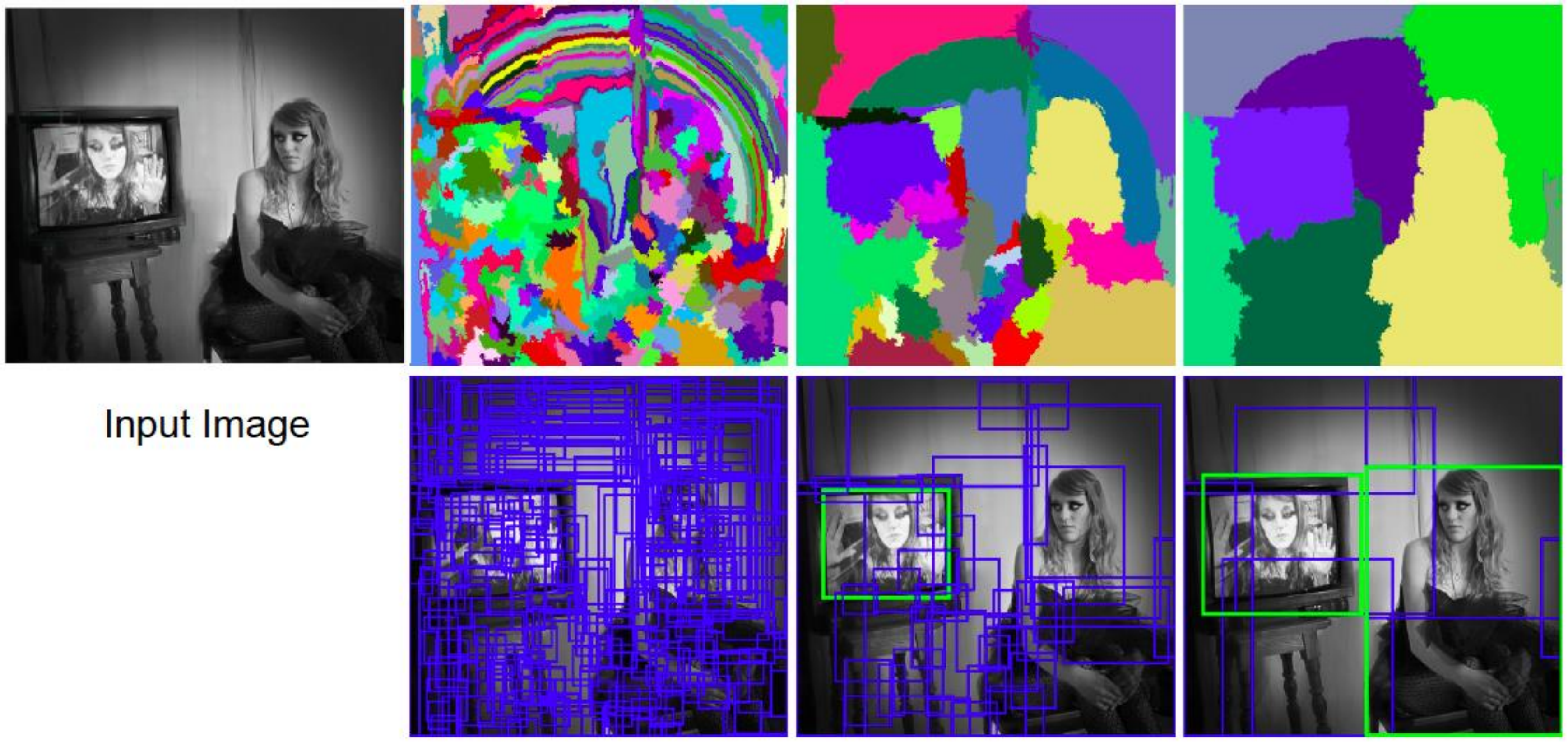
Beyond sliding windows: Region proposals



- Advantages:
  - Cuts down on number of regions detector must evaluate
  - Allows detector to use more powerful features and classifiers
  - Uses low-level *perceptual organization* cues
  - Proposal mechanism can be category-independent
  - Proposal mechanism can be trained

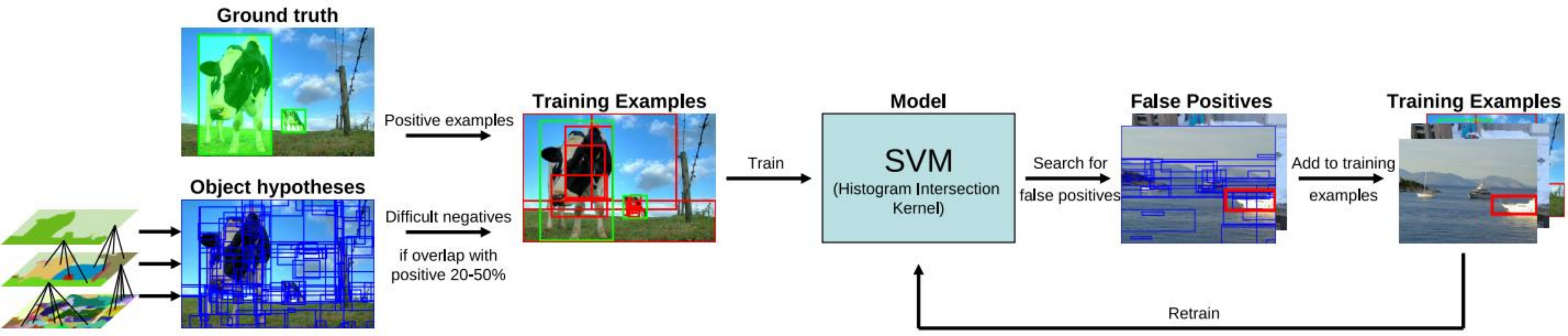
Selective search: Basic idea

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



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

Selective search detection pipeline

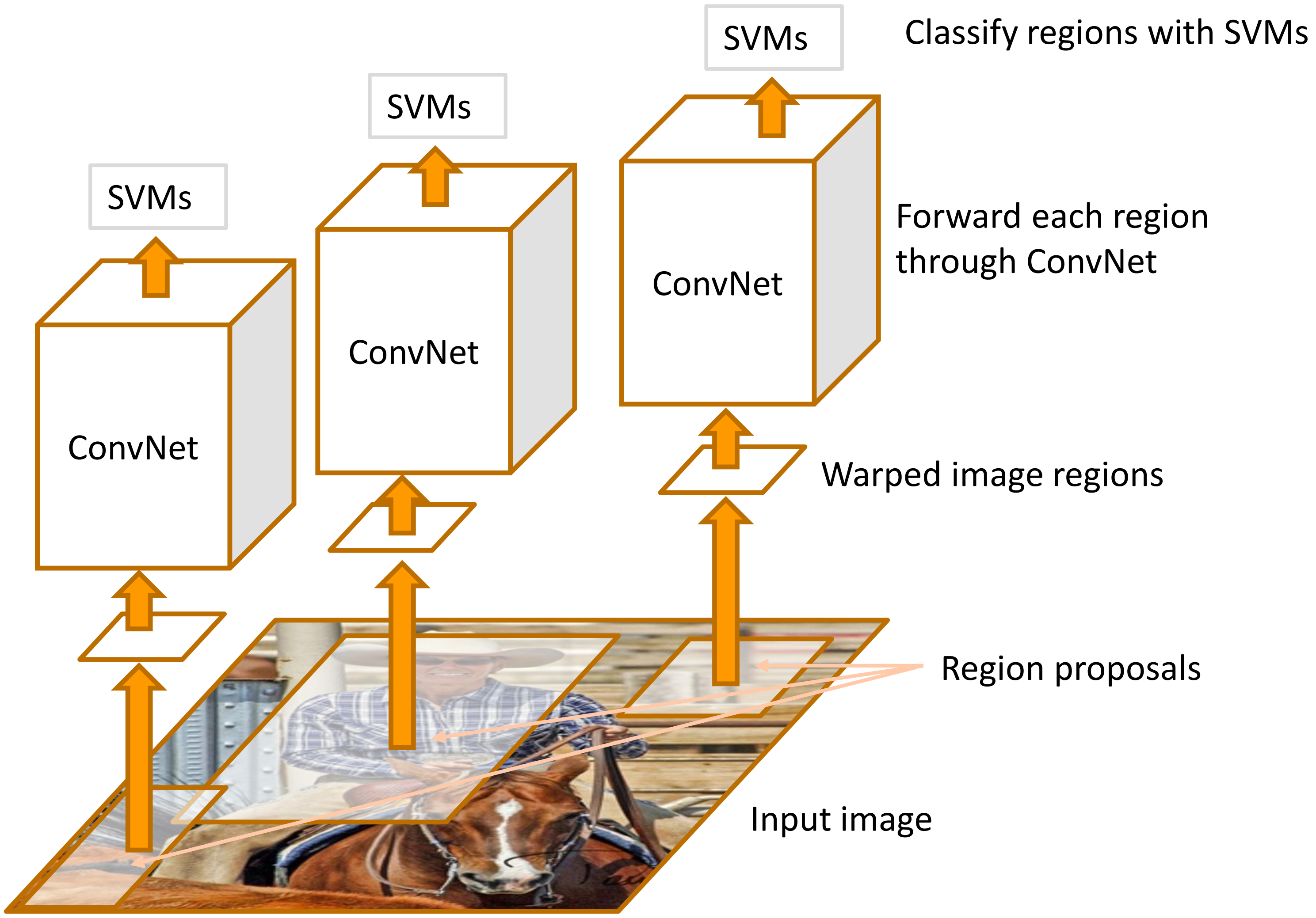


- Feature extraction: color SIFT, codebook of size 4K, spatial pyramid with four levels = 360K dimensions

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

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.

Bounding Box Regression

- Input: A set of N training pairs $\{(P^i, G^i)\}_{i=1,\ldots,N}$, where $P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$ and $G^i = (G_x^i, G_y^i, G_w^i, G_h^i)$. (Drop superscript i for simplicity)
- Output: Four functions $d_x(P), d_y(P),\ d_w(P), d_h(P)$

$$\hat{G}_x = P_w d_x(P) + P_x \tag{1}$$

$$\hat{G}_y = P_h d_y(P) + P_y \tag{2}$$

$$\hat{G}_w = P_w \exp(d_w(P)) \tag{3}$$

$$\hat{G}_h = P_h \exp(d_h(P)). \tag{4}$$

- Learn model parameters by optimizing the regularized least squares objective

$$\mathbf{w}_\star = \operatorname*{argmin}_{\hat{\mathbf{w}}_\star} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \left\|\hat{\mathbf{w}}_\star\right\|^2 . \tag{5}$$

Bounding Box Regression

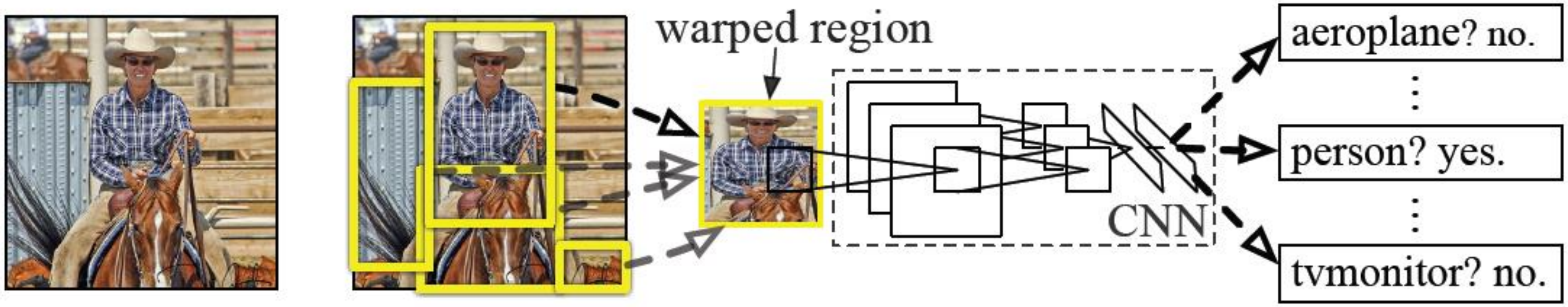
- Learn model parameters by optimizing the regularized least squares objective

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 . \quad (5)$$

- The regression targets t_* for the training pair (P, G) are defined as

$$\begin{aligned} t_x &= (G_x - P_x)/P_w && (6) \\ t_y &= (G_y - P_y)/P_h && (7) \\ t_w &= \log(G_w/P_w) && (8) \\ t_h &= \log(G_h/P_h). && (9) \end{aligned}$$

R-CNN details



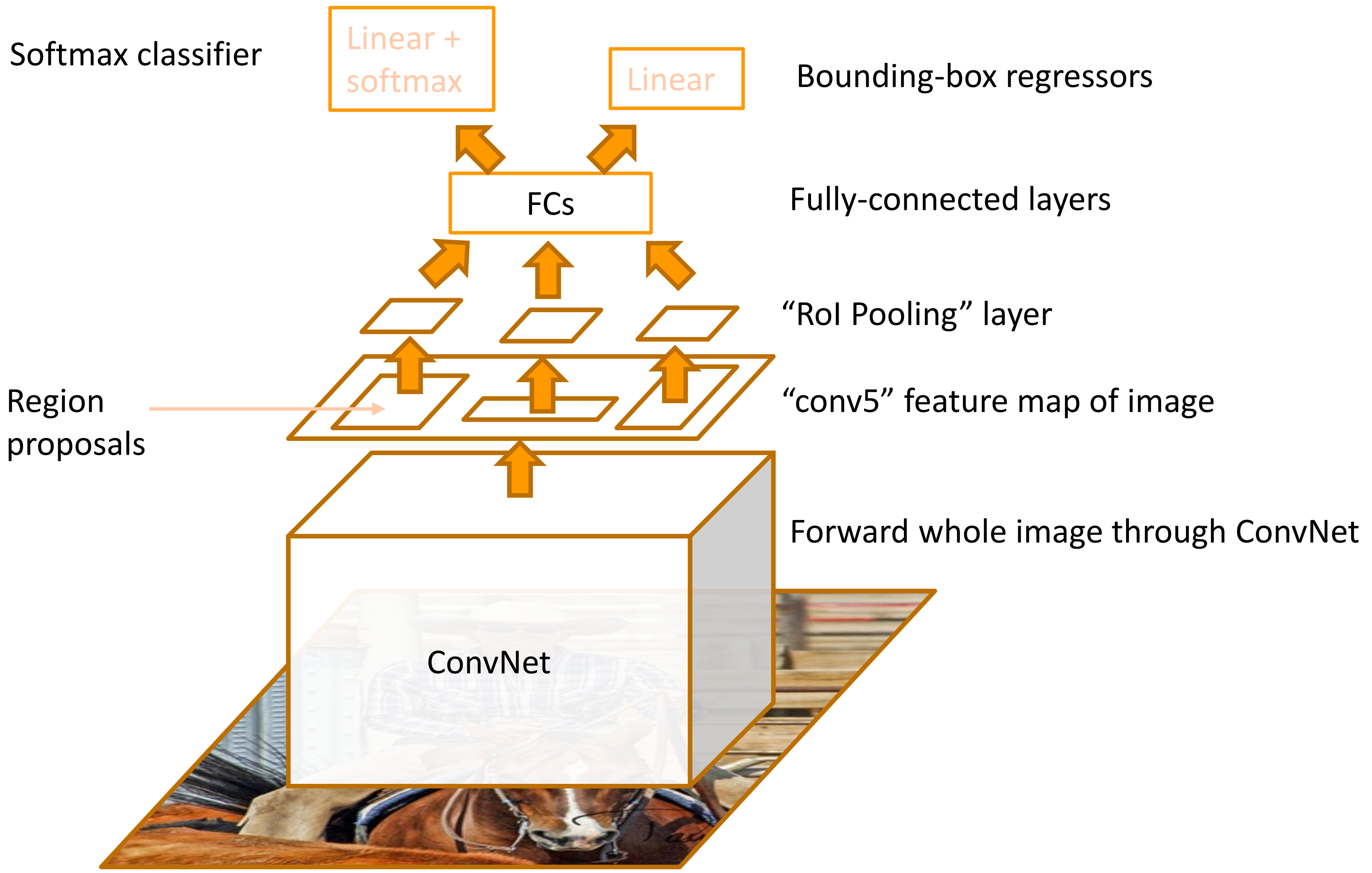
- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to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

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

R-CNN pros and cons

- Pros
  - Accurate!
  - Any deep architecture can immediately be “plugged in”
- Cons
  - Ad hoc training objectives
    - 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 convnet passes per image
  - Inference (detection) is slow (47s / image with VGG16)

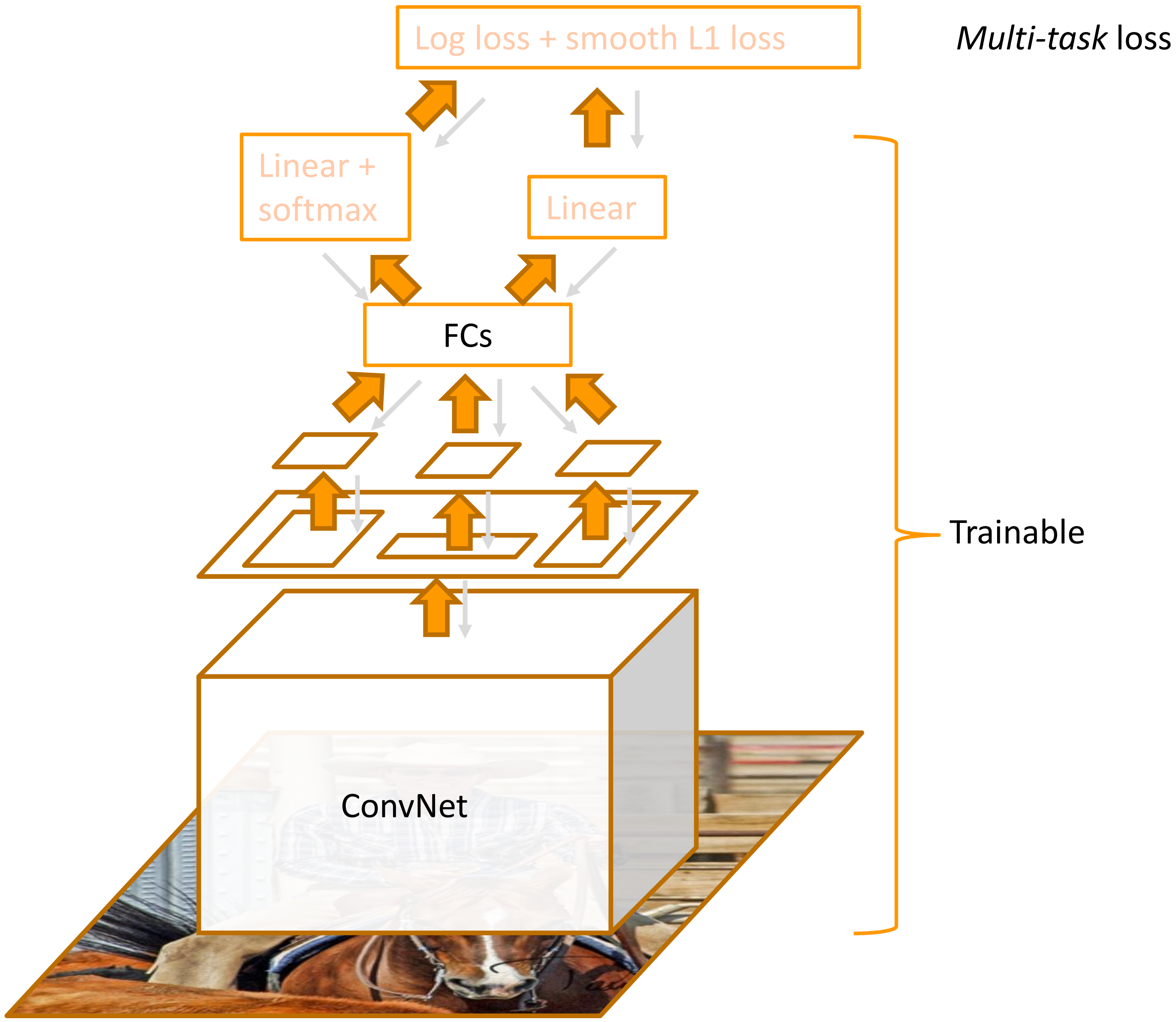
Fast R-CNN



Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015

Fast R-CNN training



Source: R. Girshick

R. Girshick, Fast R-CNN, ICCV 2015

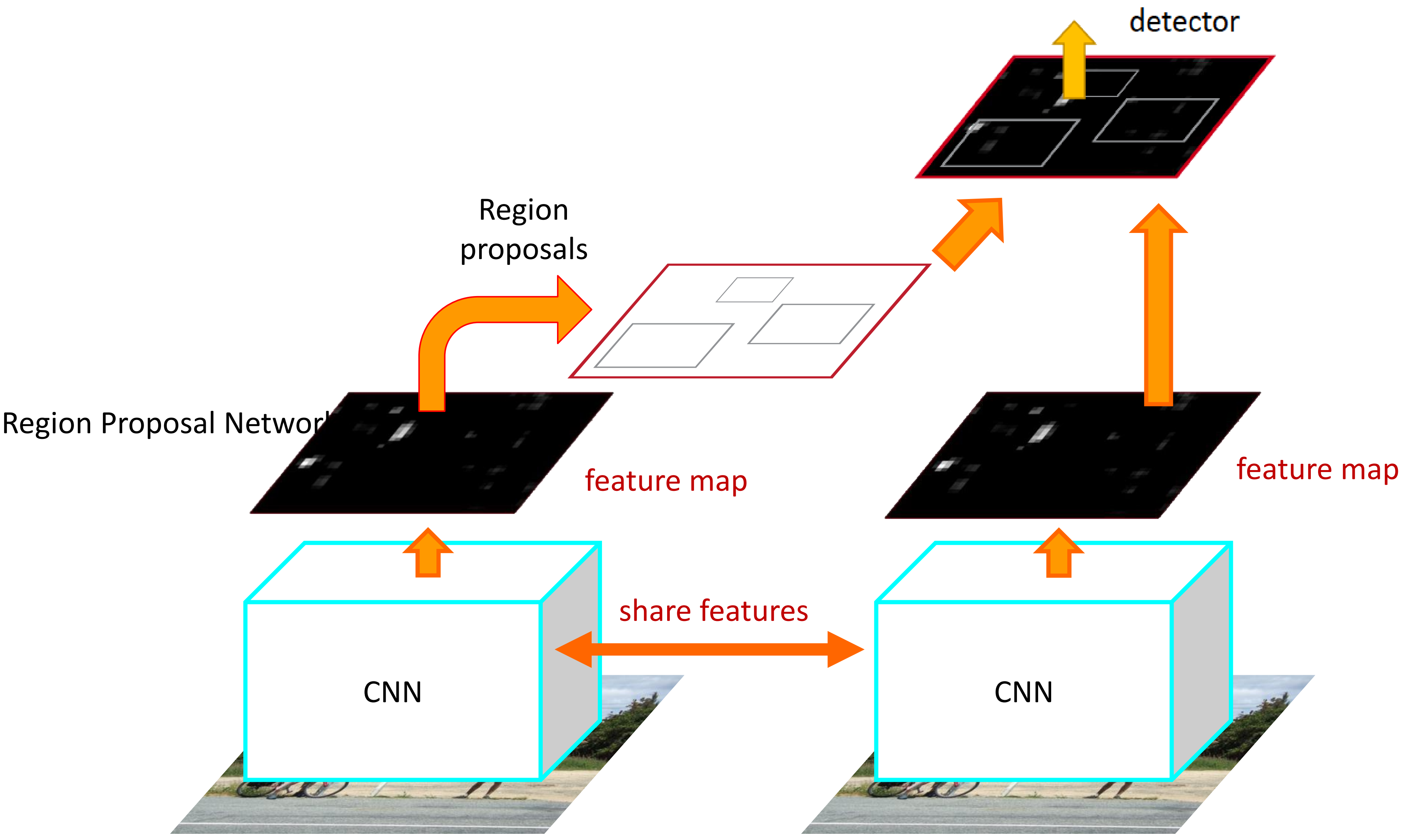
Fast R-CNN results

	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%

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

Source: R. Girshick

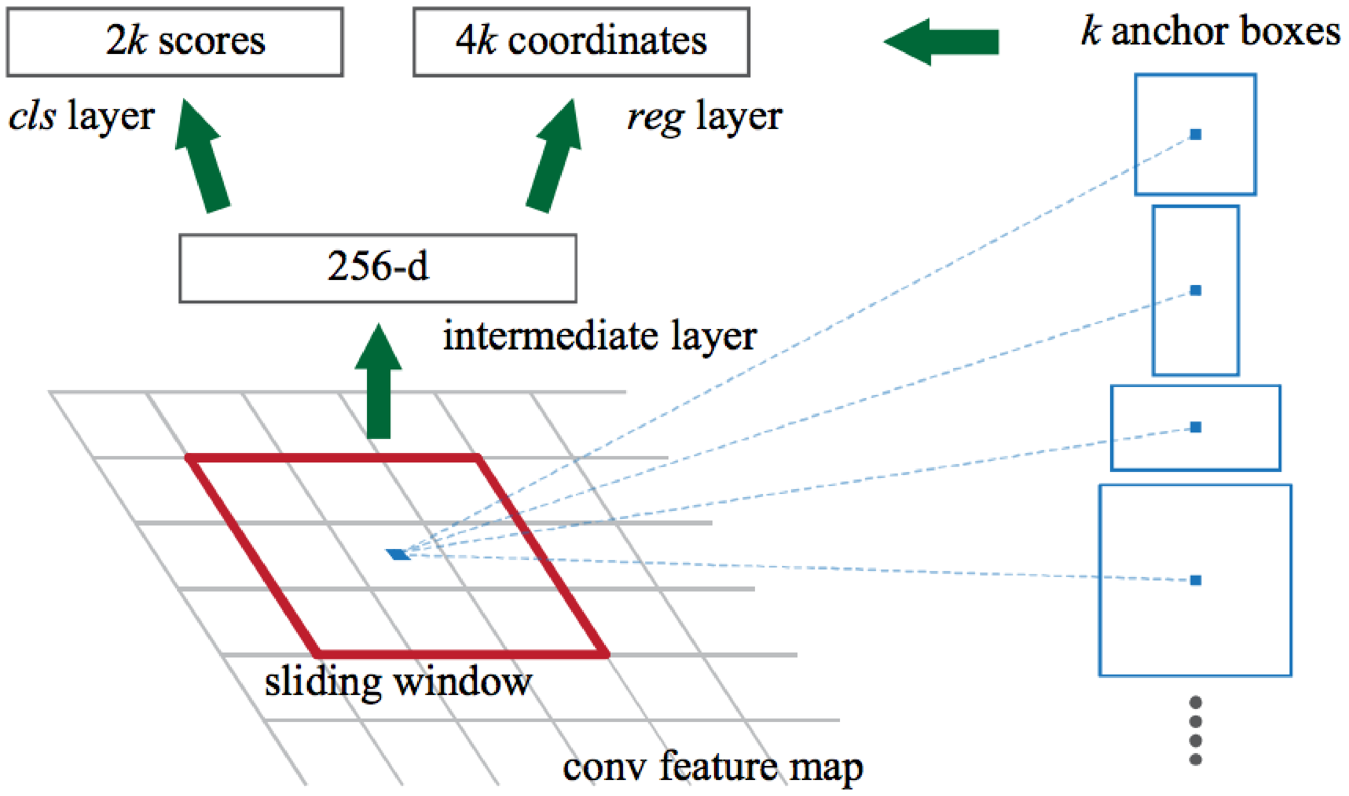
Faster R-CNN



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

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



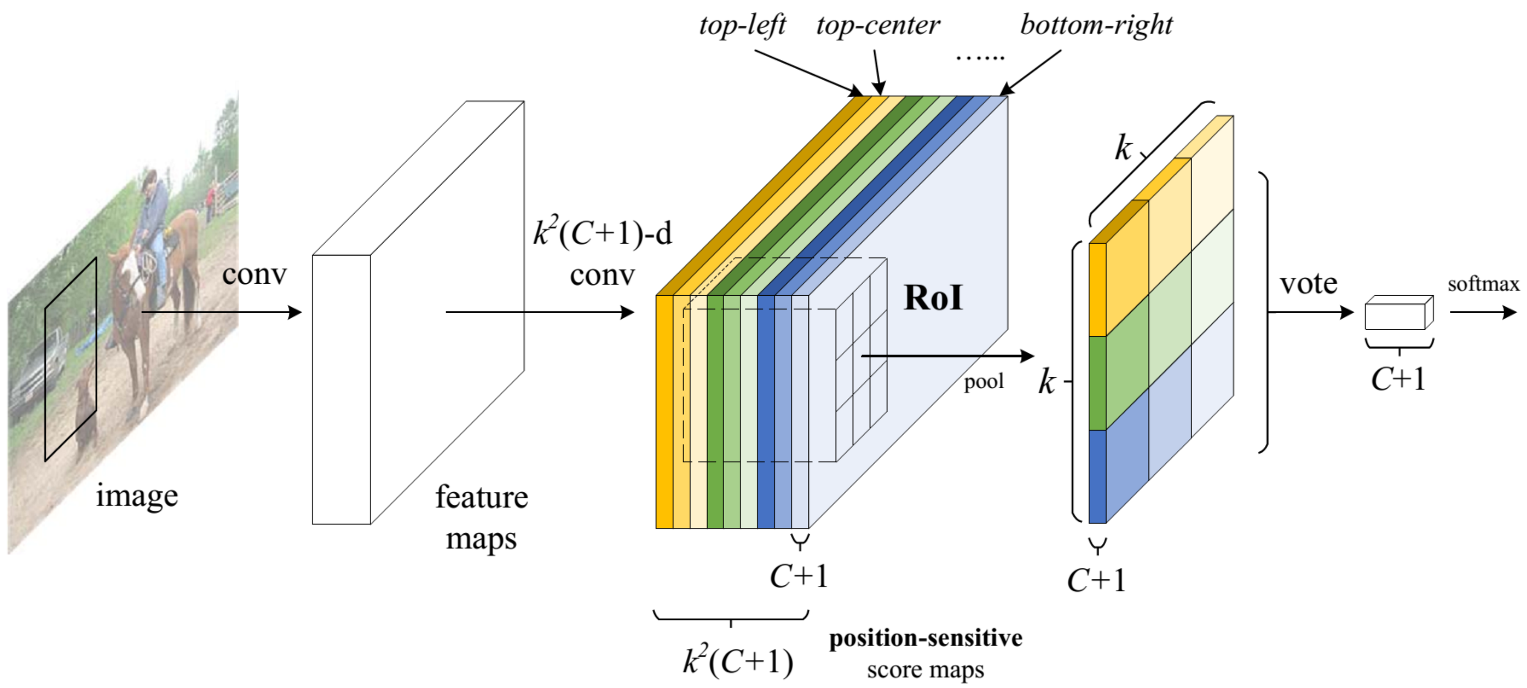
Faster R-CNN results

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

R-FCN

- Abandon FC layers. Use fully convolutional layers.



Jifeng Dai et al., R-FCN: Object Detection via Region-based Fully Convolutional Networks, NIPS 2016

R-FCN

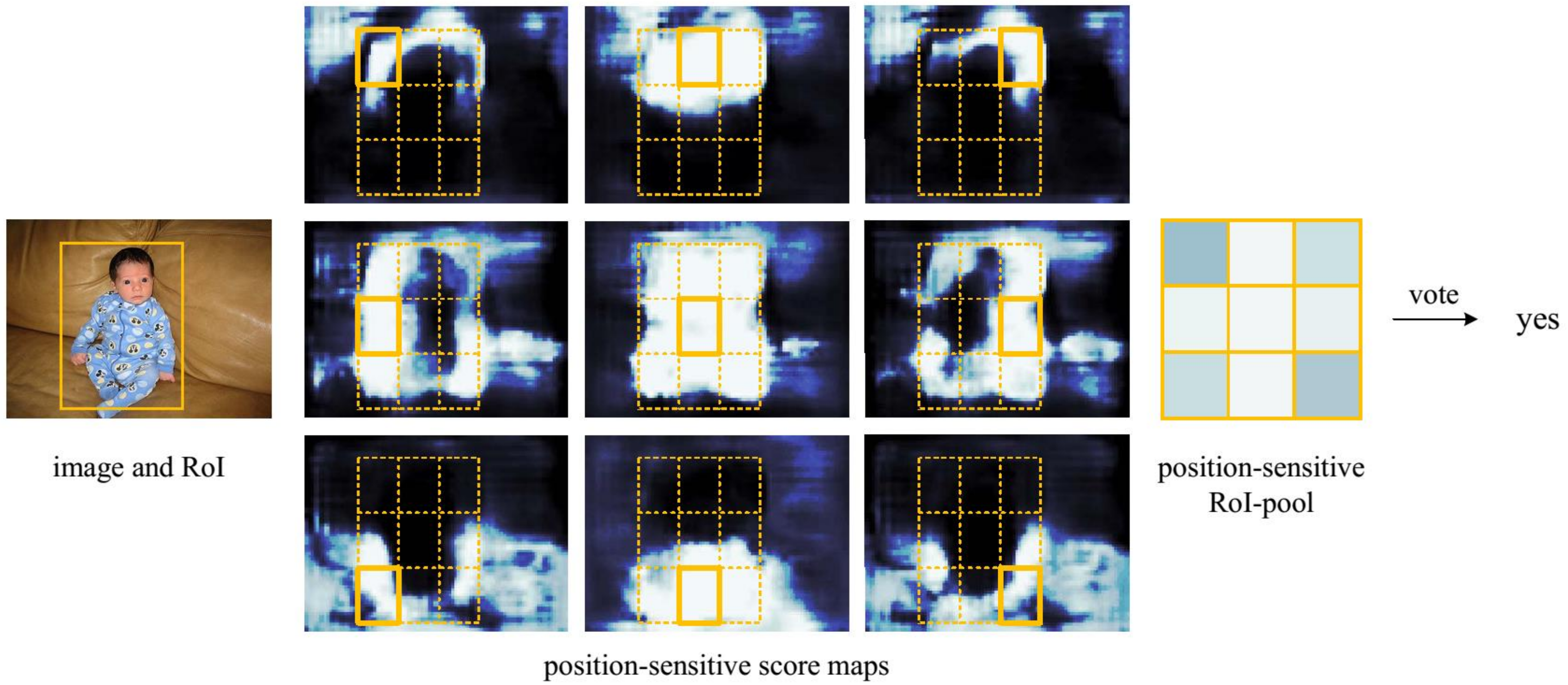
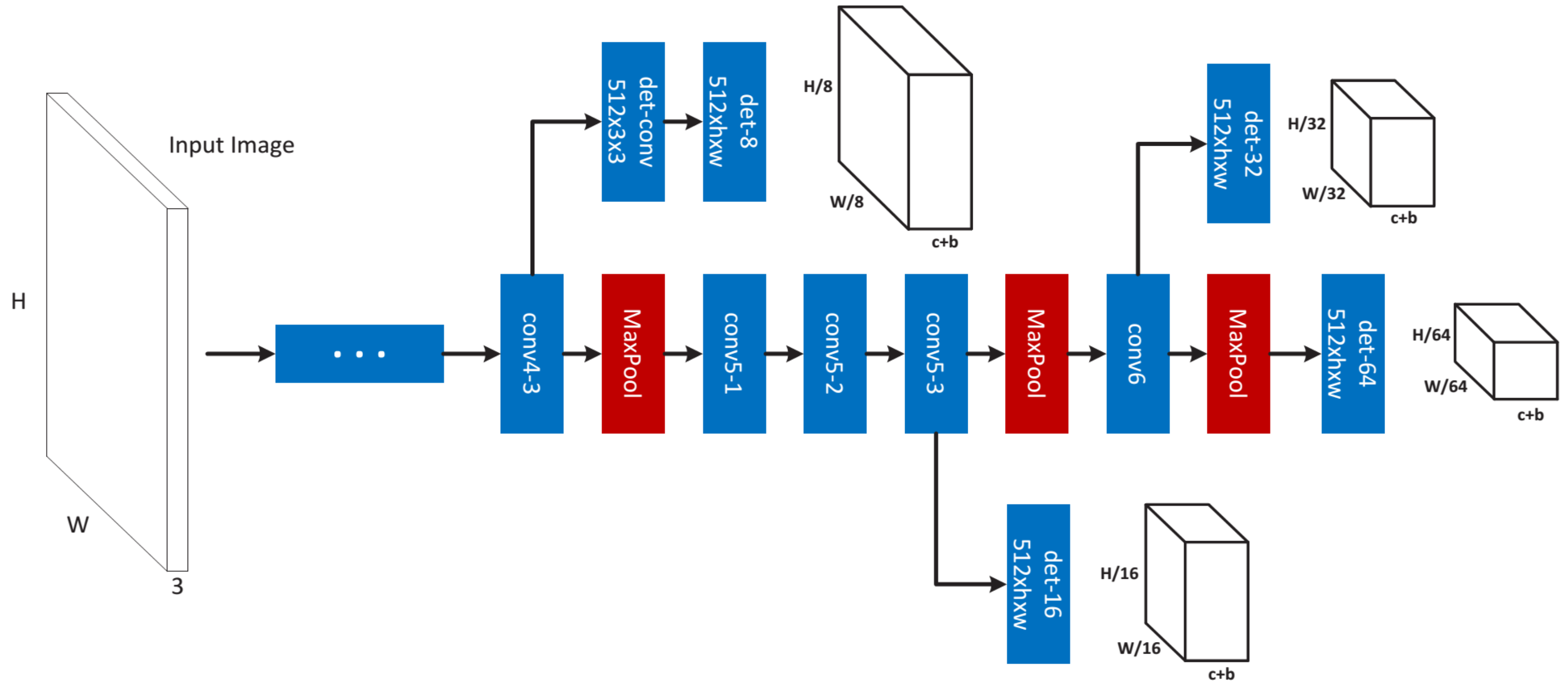


Figure 3: Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.

Multi-Scale R-CNN

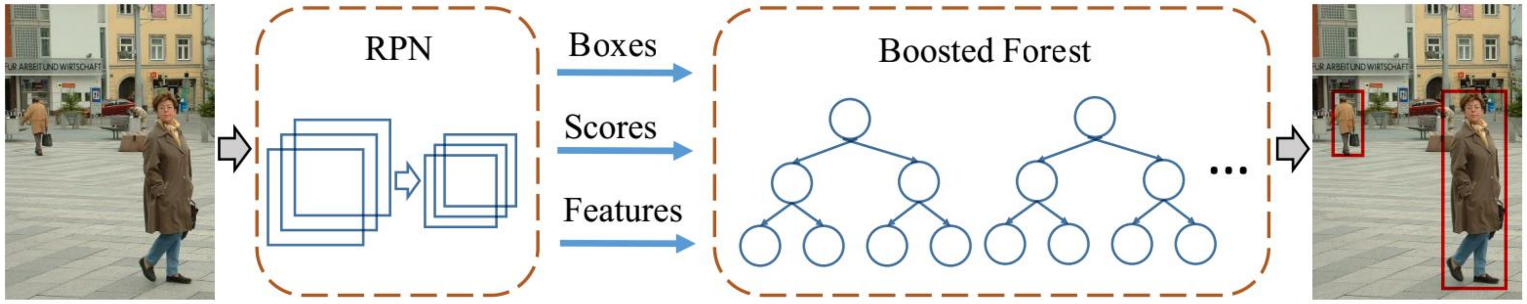
- Objects at different scales



Zhaowei Cai et al, A Unified Multi-scale Deep CNN for Fast Ob ject Detection, ECCV 2016

Remove Linear Classifiers

- FC layers are “weak” classifiers
- Replace with non-linear classifiers



Liliang Zhang et al., Is Faster R-CNN Doing Well for Pedestrian Detection? ECCV 2016

RPN v.s. Cascade

- RPN network is a two-stage cascade
- Can we remove RPN? - an open research problem

Questions?