Object Detection with Deformable Part Models (DPM)

Pedro F. Felzenszwalb
School of Engineering and Dept. of Computer Science

Spring 2012 Course

ENGN2520
Pattern Recognition and Machine Learning

Meeting: Tue/Thu 2:30-3:50
Instructor: Pedro Felzenszwalb

\We will consider applications in computer vision, signal processing, speech recognition and information retrieval.

Topics include: decision theory, parametric and non-parametric learning, dimensionality reduction, graphical models, exact and approximate inference, semi-supervised learning, generalization bounds and support vector machines.

Prerequisites: basic probability, linear algebra, calculus and some programming experience.

Object category detection

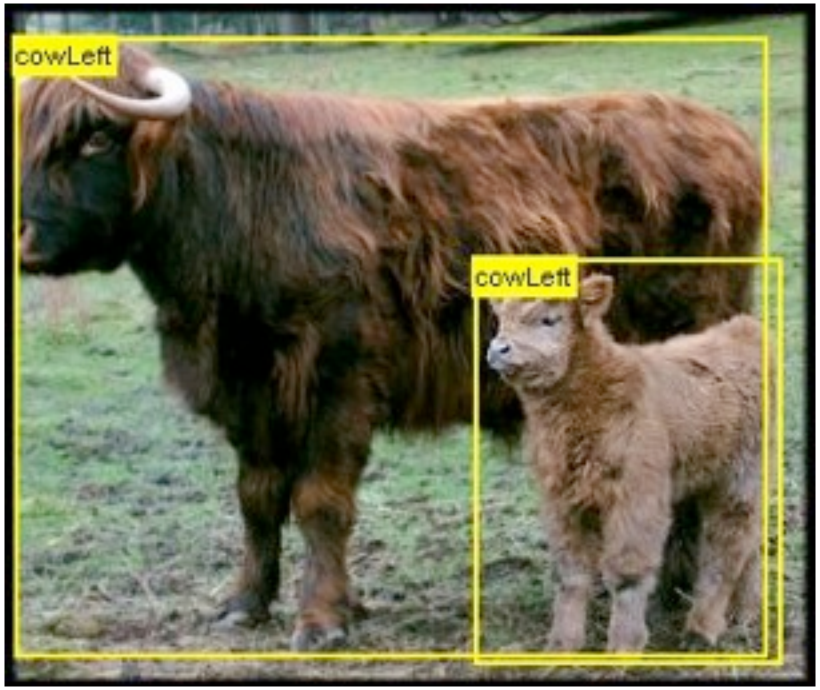
Goal: detect all pedestrians, cars, trees, squirrels, ...

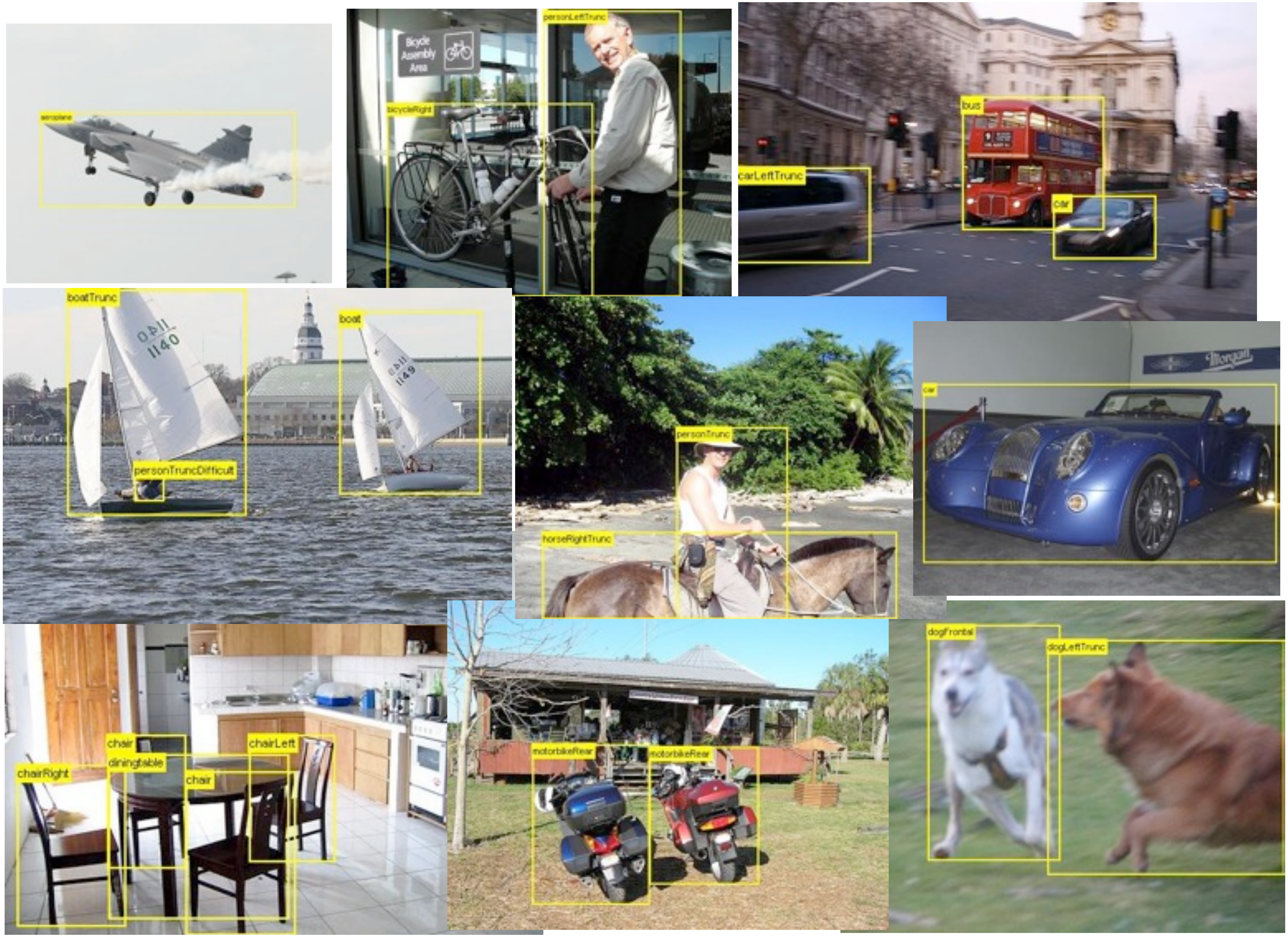
Why is it hard?

- Objects in a category have highly variable appearance
  - Photometric variation
  - Viewpoint variation
  - Intra-class variability
    - Cars come in a variety of shapes (sedan, minivan, etc)
    - People wear different clothes and take different poses

PASCAL Challenge

- Objects from 20 categories
  - person, car, bicycle, bus, airplane, sheep, cow, table, ...
- Objects are annotated with bounding boxes



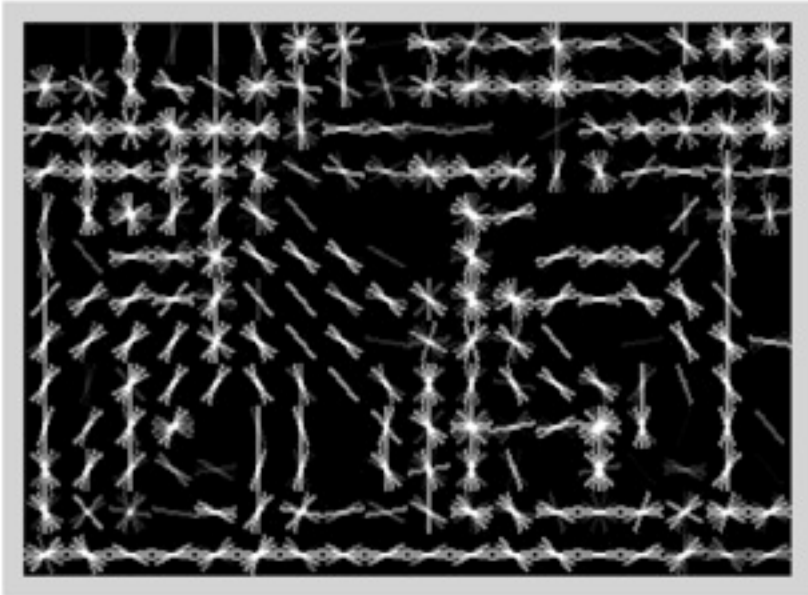

aeroplane
Bicycle Assembly Area
personLeftTrunc
bicycleRight
bus
carLeftTrunc
car
boatTrunc
boat
personTruncDifficult
personTrunc
horseRightTrunc
car
chair
diningtable
chairLeft
chairRight
chair
motorbikeRear
motorbikeRear
dogFrontal
dogLeftTrunc

Starting point: sliding window classifiers

Feature vector
$x = [\ldots, \ldots, \ldots, \ldots]$

- Detect objects by testing each subwindow
  - Reduces object detection to binary classification
  - Dalal & Triggs: HOG features + linear SVM

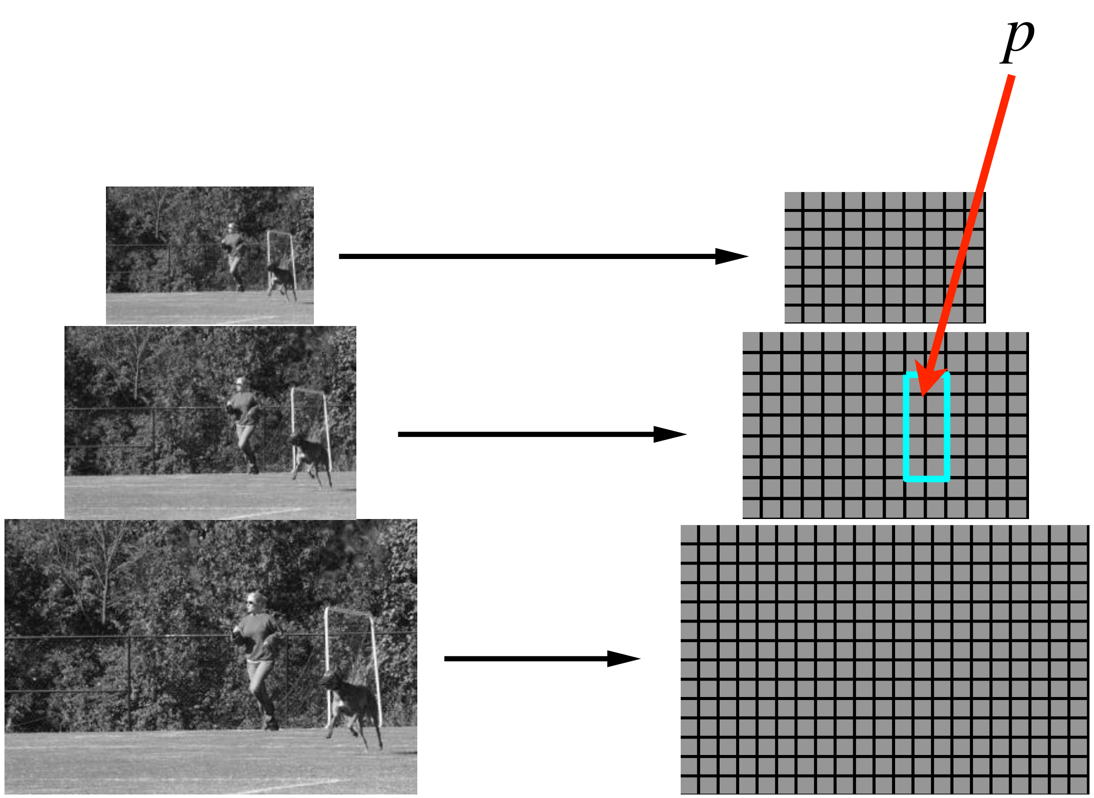
Histogram of Gradient (HOG) features



- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.

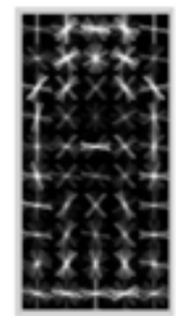
HOG Filters

- HOG filter is a template for HOG features
- Score is dot product of filter and feature vector



HOG pyramid H

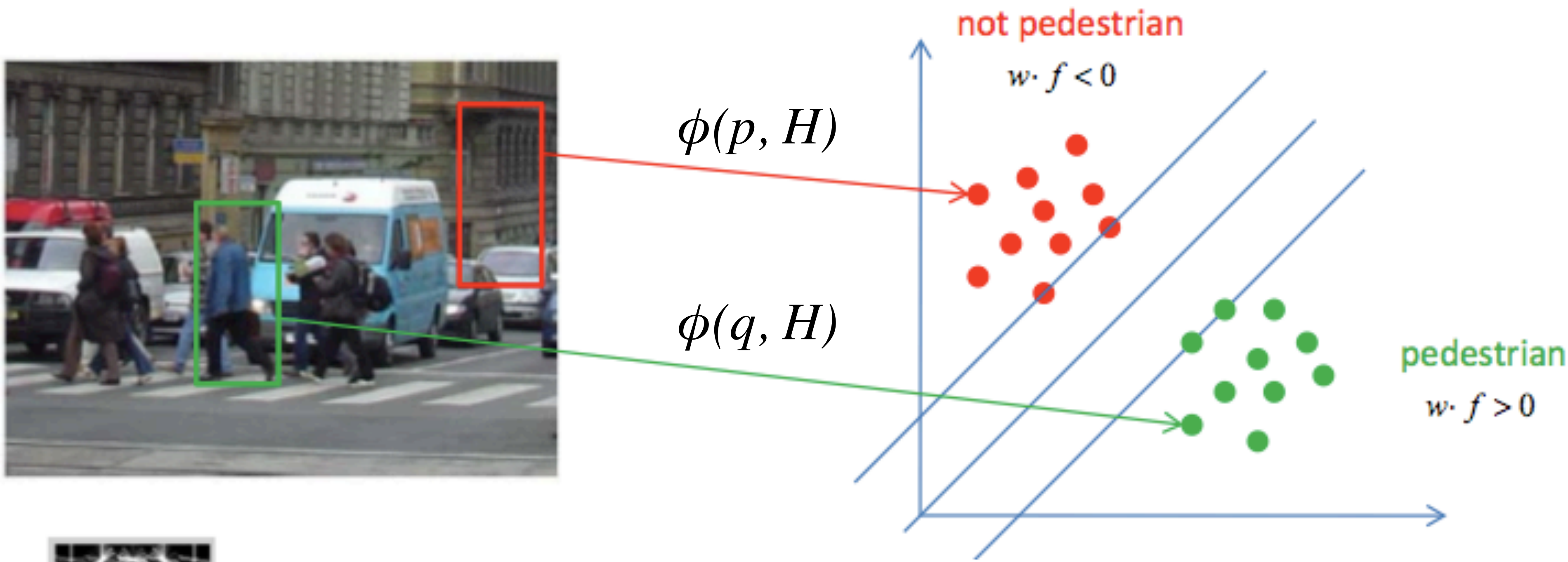
filter F



Score of F at position p is
$F \cdot \phi(p, H)$

$\phi(p, H)$ = HOG features in subwindow specified by p

Dalal & Triggs: HOG + linear SVMs



Typical form of a model

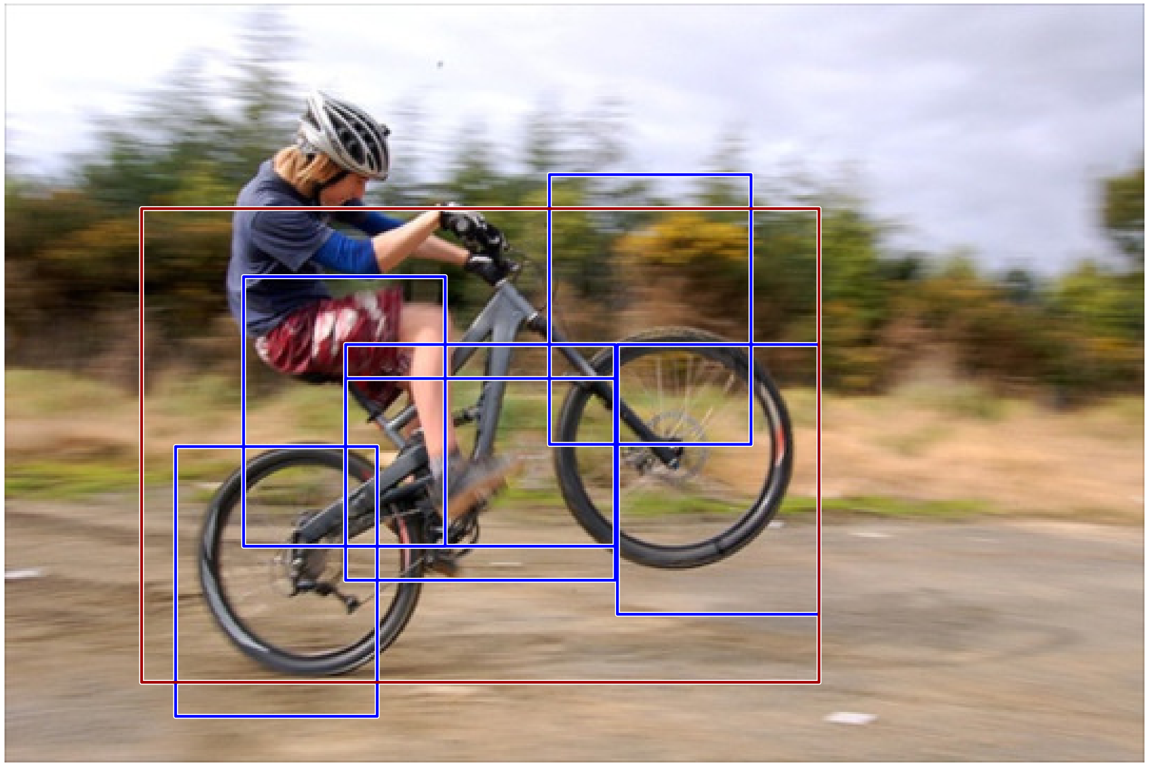
There is much more background than objects

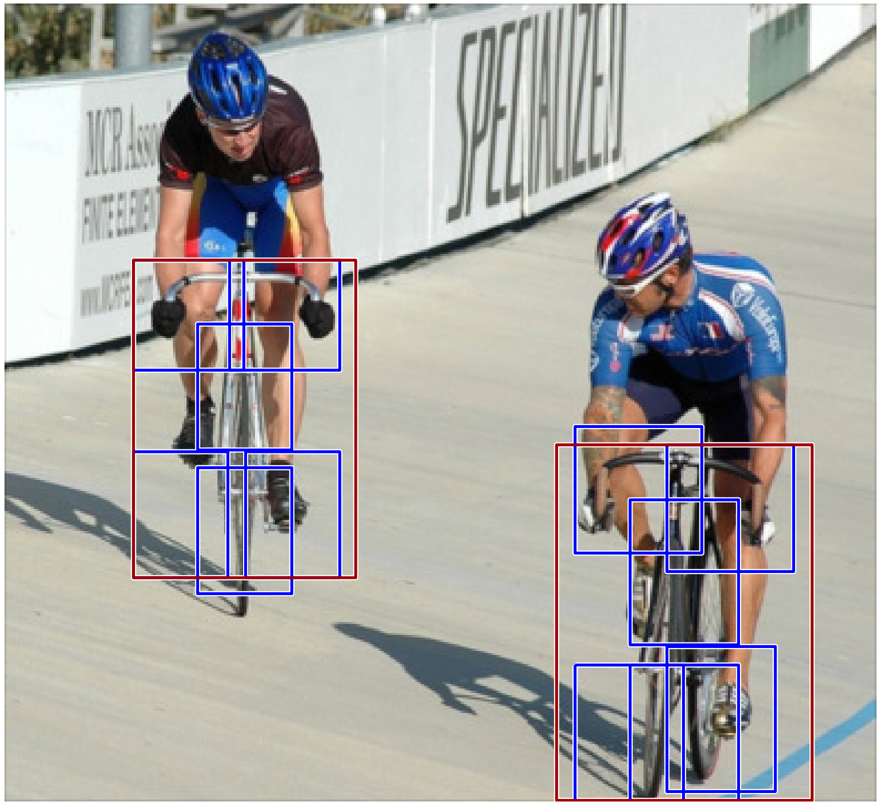
Start with random negatives and repeat:

1) Train a model

2) Harvest false positives to define "hard negatives"

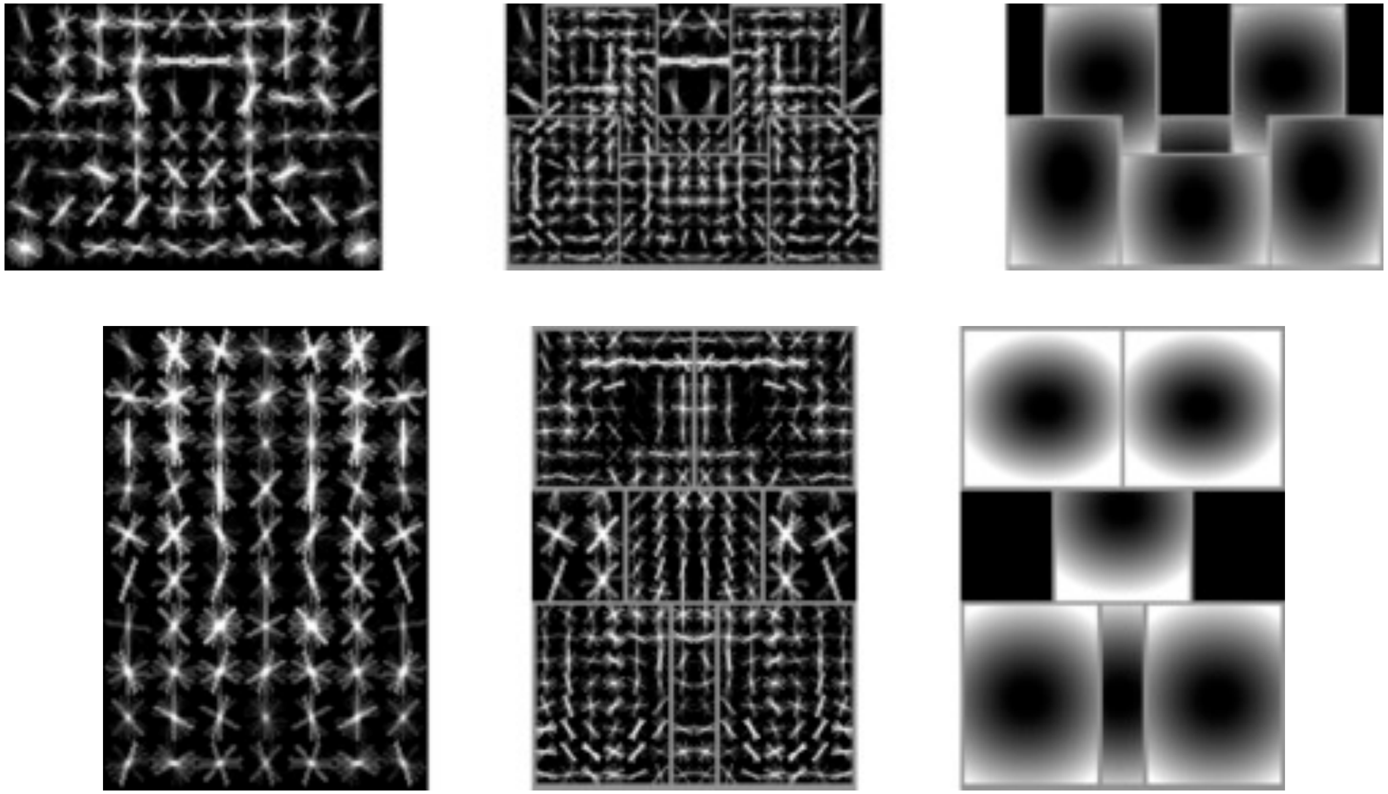
Deformable part models





- Collection of templates arranged in a deformable configuration
- Each model has global template + part templates
- Fully trained from bounding boxes alone

2 component bicycle model



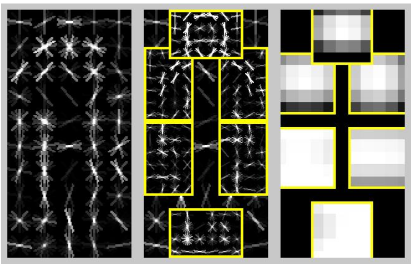
root filters
coarse resolution

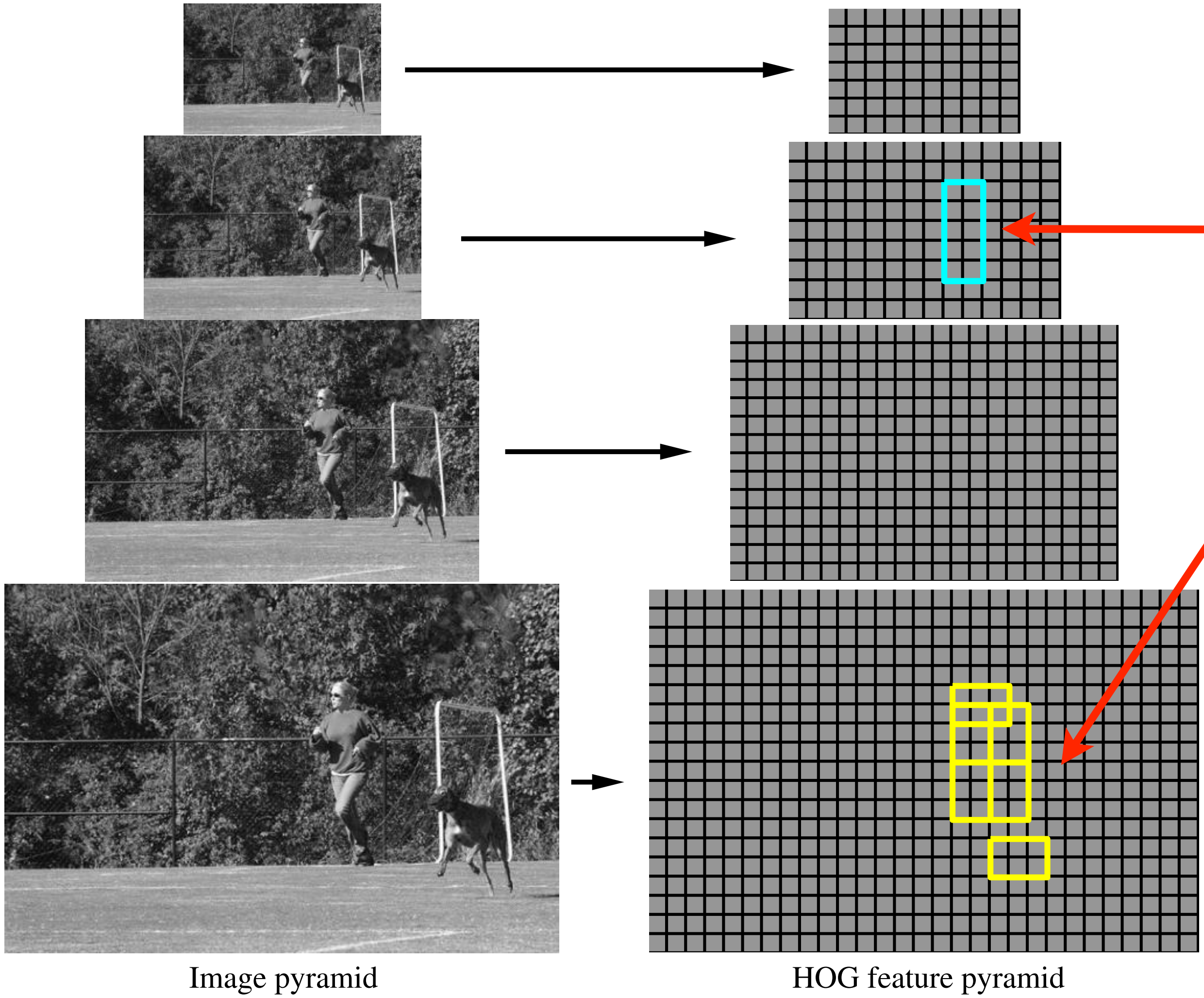
part filters
finer resolution

deformation
models

Each component has a root filter F_0
and n part models (F_i, v_i, d_i)

Object hypothesis





$z = (p_0, \ldots, p_n)$

p_0 : location of root

$p_1, \ldots, p_n$: location of parts

Score is sum of filter scores minus deformation costs

Multiscale model captures features at two-resolutions

Score of a hypothesis

“data term”

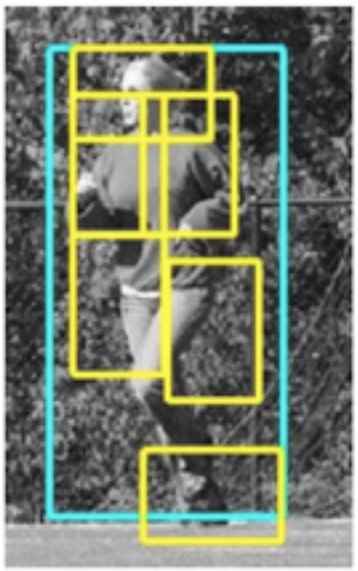
“spatial prior”

$$\mathrm{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)$$

filters

deformation parameters

displacements



$$\mathrm{score}(z) = \beta \cdot \Psi(H, z)$$

concatenation filters and deformation parameters

concatenation of HOG features and part displacement features

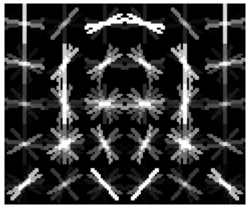
Matching

- Define an overall score for each root location
  - Based on best placement of parts

$$\text{score}(p_0) = \max_{p_1,\ldots,p_n} \text{score}(p_0,\ldots,p_n).$$

- High scoring root locations define detections
  - "sliding window approach"
- Efficient computation
  - Dyna
  - Generalized distance transforms (max-convolution)

For each part, pick location with high score near ideal location relative to root



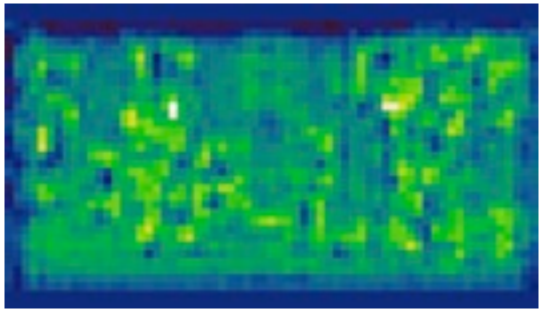
head filter

input image

Response of filter in l-th pyramid level

$$R_l(x, y) = F \cdot \phi(H, (x, y, l))$$

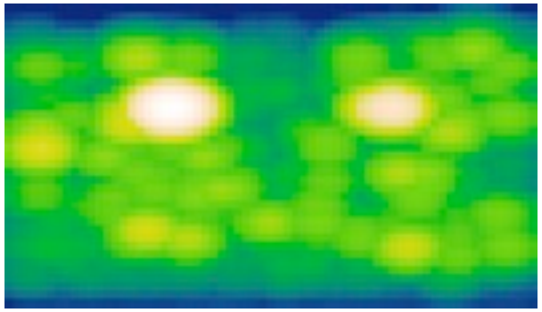
cross-correlation

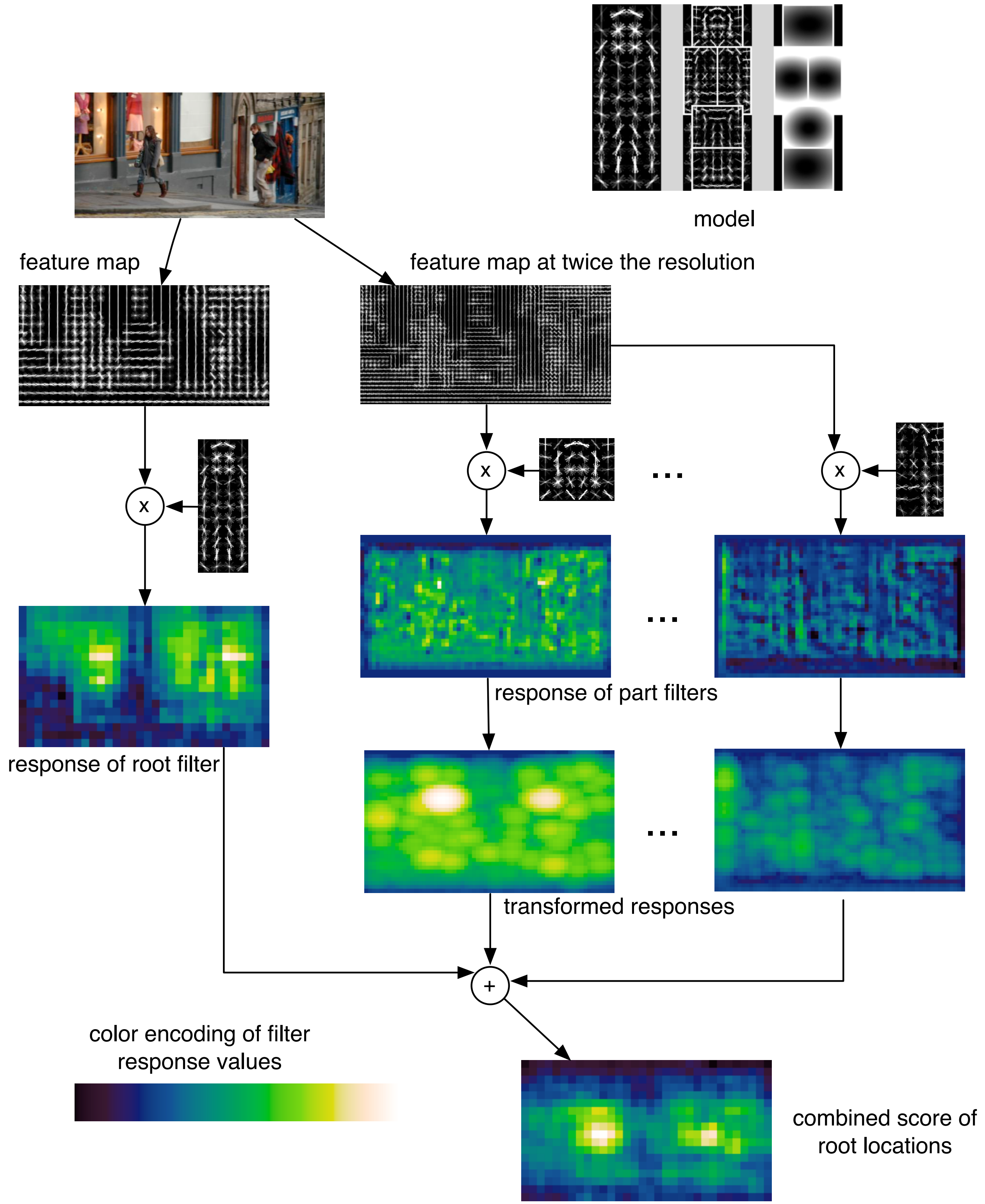


Transformed response

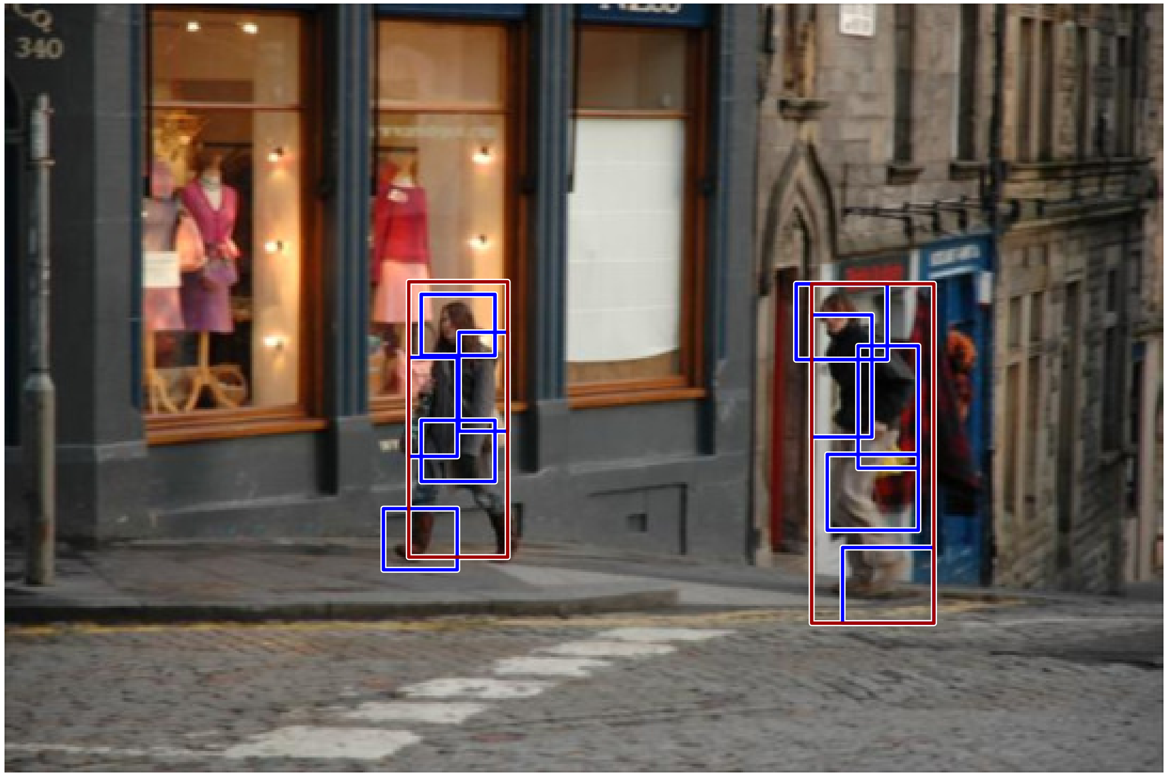
$$D_l(x, y) = \max_{dx, dy} \left(R_l(x + dx, y + dy) - d_i \cdot (dx^2, dy^2) \right)$$

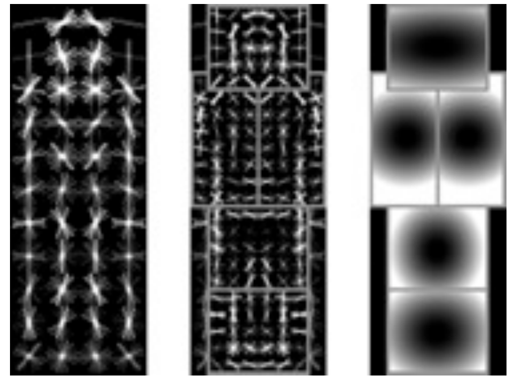
max-convolution, computed in linear time
(spreading, local max, etc)




model
feature map
feature map at twice the resolution
x
x
x
...
...
...
response of part filters
response of root filter
transformed responses
+
color encoding of filter response values
combined score of root locations

Matching results



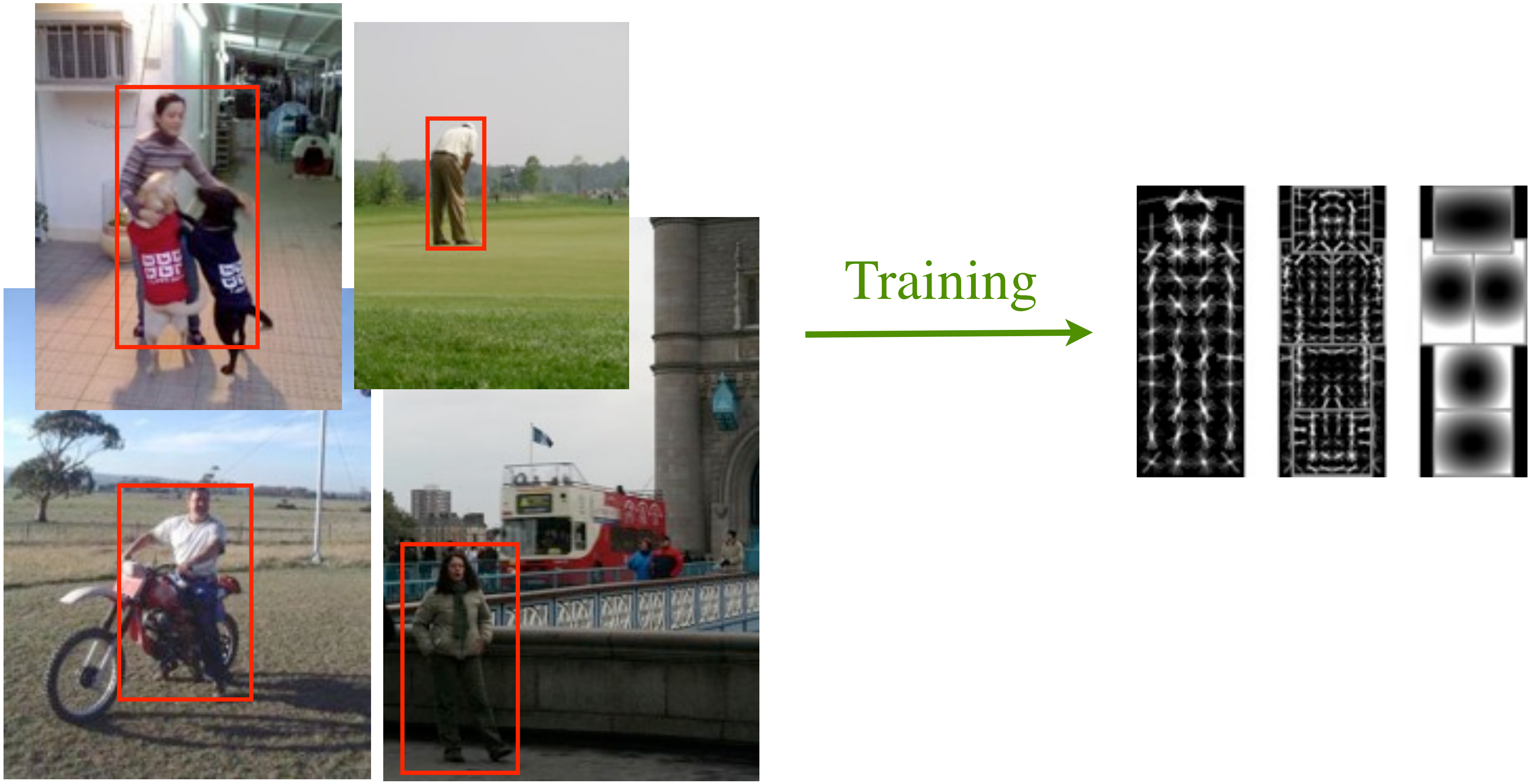


(after non-maximum suppression)

~1 second to search all scales on a multi-core computer

Learning

- Training data: images with bounding boxes
- Need to learn the model structure, filters and deformation costs



Latent SVM

Classifiers that score an example *x* using

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

β are model parameters
z are latent values

Training data $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle) \qquad y_i \in \{-1, 1\}$

We would like to find β such that: $y_i f_\beta(x_i) > 0$

Minimize

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

Semi-convexity

- Maximum of convex functions is convex
- $f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$ is convex in β
- $\max(0, 1 - y_i f_\beta(x_i))$ is convex for negative examples

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

Convex if latent values for positive examples are fixed

Latent SVM training

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

- Convex if we fix z for positive examples
- Optimization:
  - Initialize β and iterate:
    - Pick best z for each positive example
    - Optimize β via gradient descent with data-mining

Learning models from bounding boxes

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

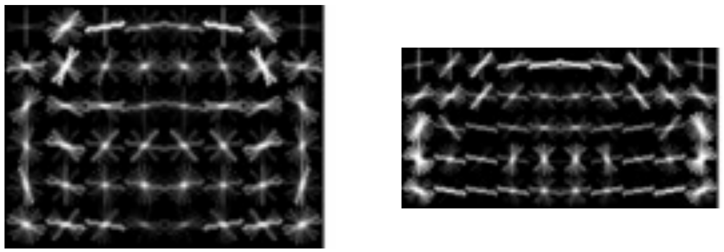
- Reduce to Latent SVM training problem
- Positive example: some z should have high score
- Bounding box defines range of root locations
  - Parts can be anywhere
  - This defines $Z(x)$

Background

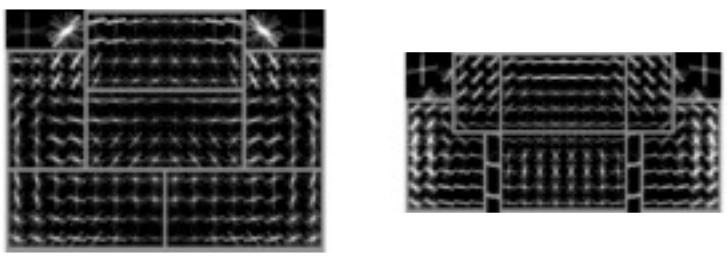
$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

- Negative example specifies no z should have high score
- One negative example per root location in a "background" image
  - Huge number of negative examples
  - Consistent with requiring low false-positive rate

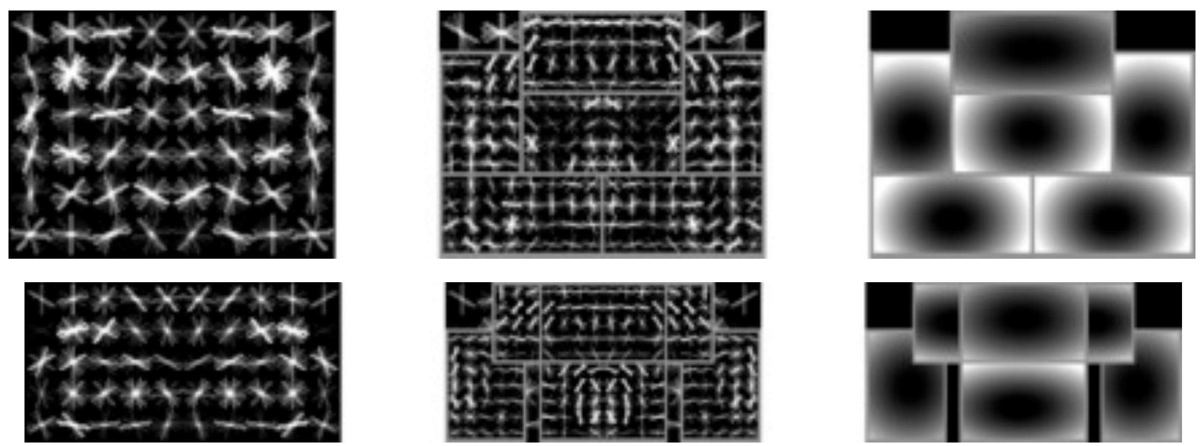
- Sequence of training rounds
  - Separate examples based on bounding box aspect ratio (“pose”)
  - Train multiple root filters



  - Initialize parts from root

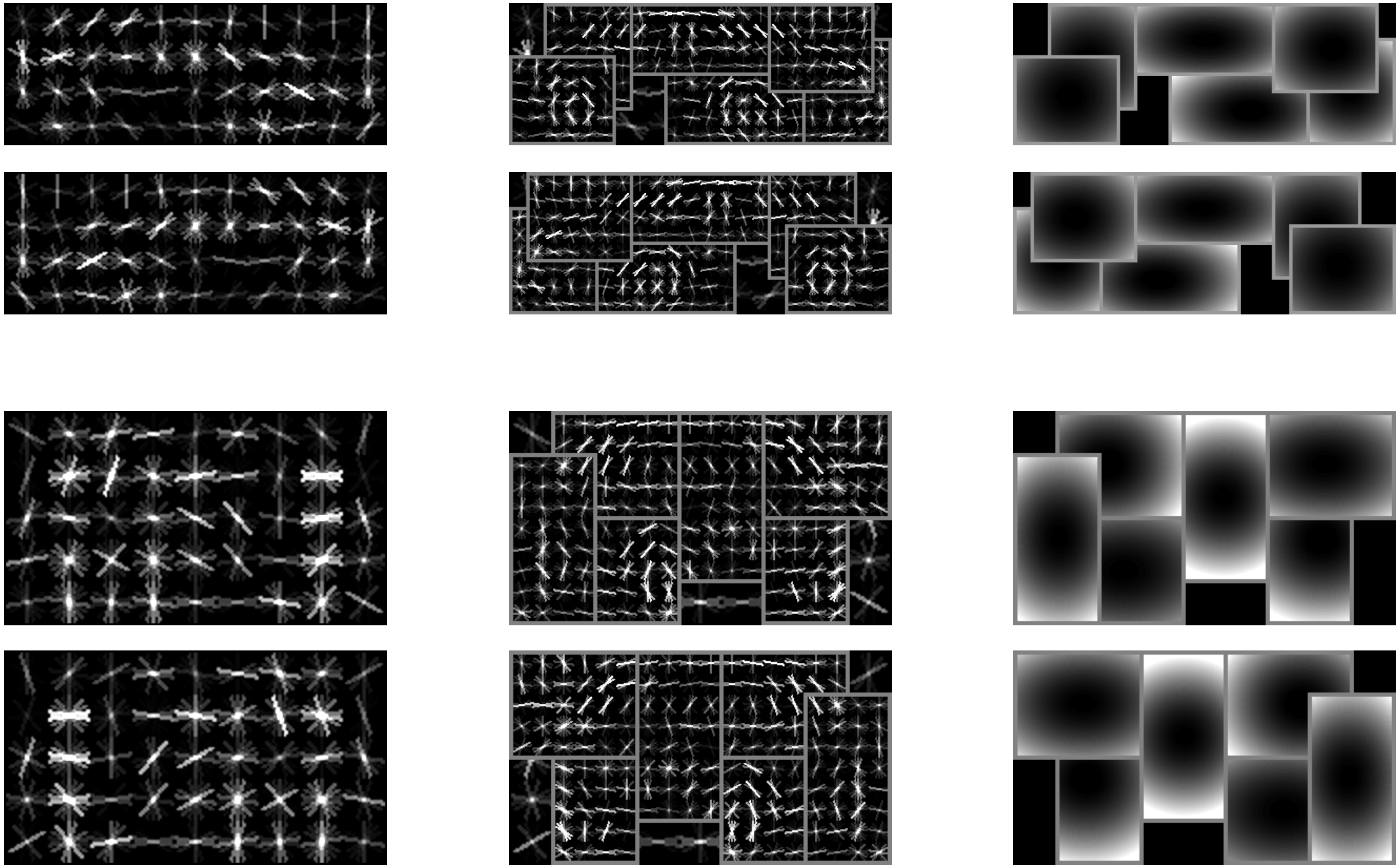


  - Merge into a mixture
  - Train final model



6 component car model

2 of 3 symmetric pairs shown



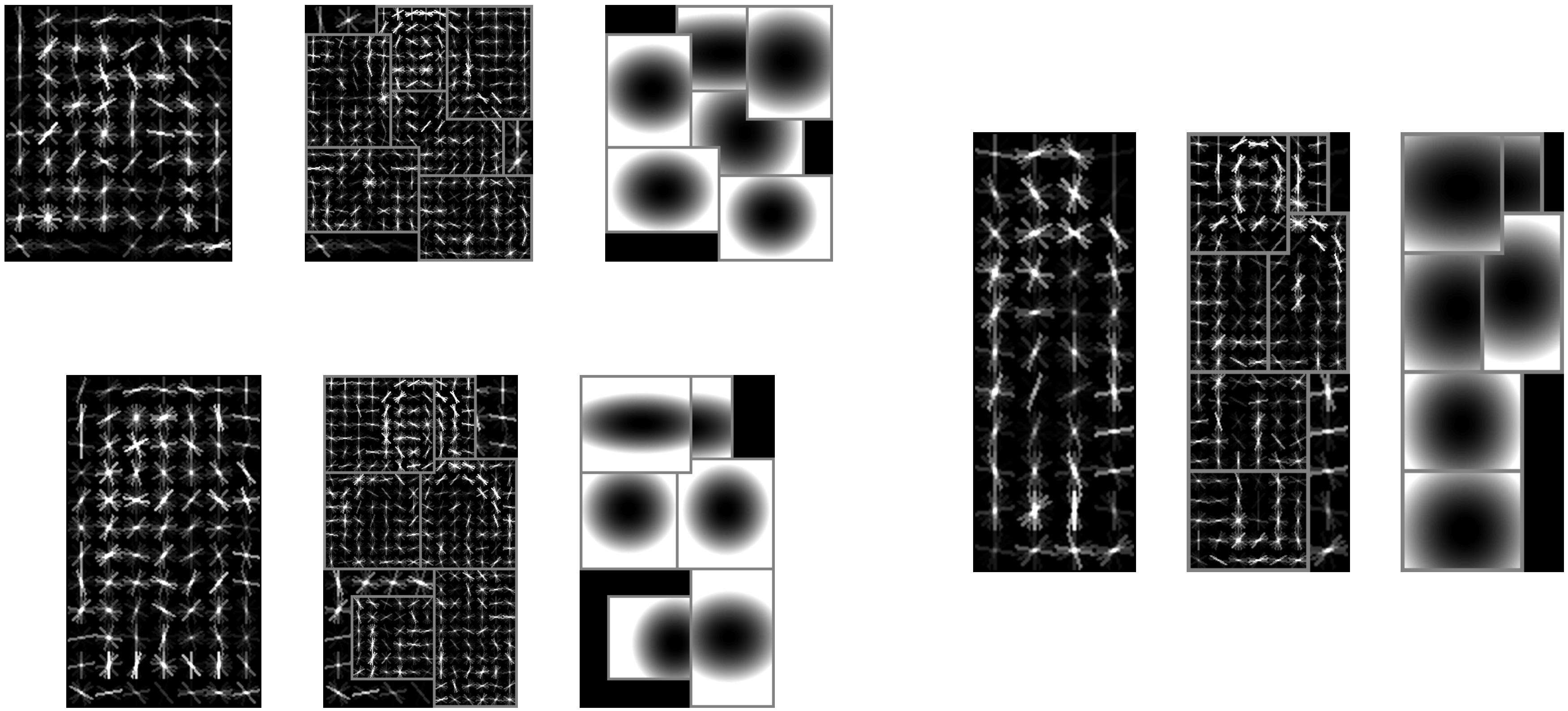
root filters
coarse resolution

part filters
finer resolution

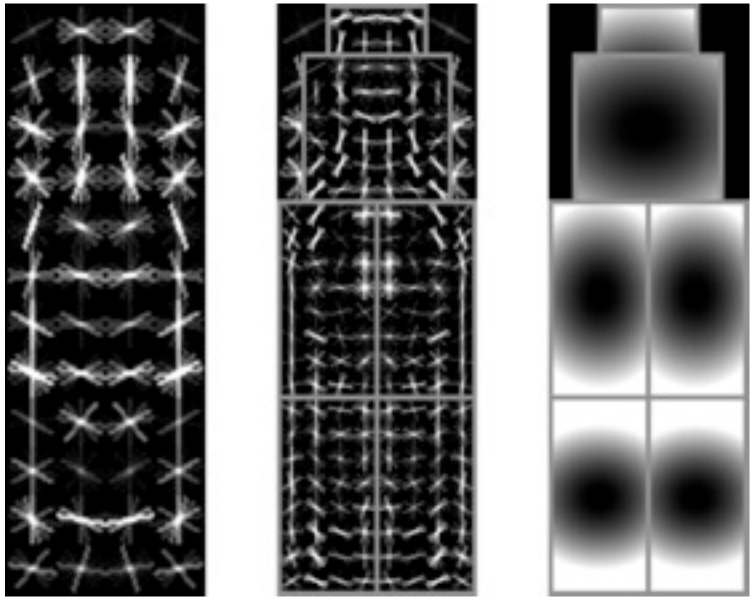
deformation
models

6 component person model

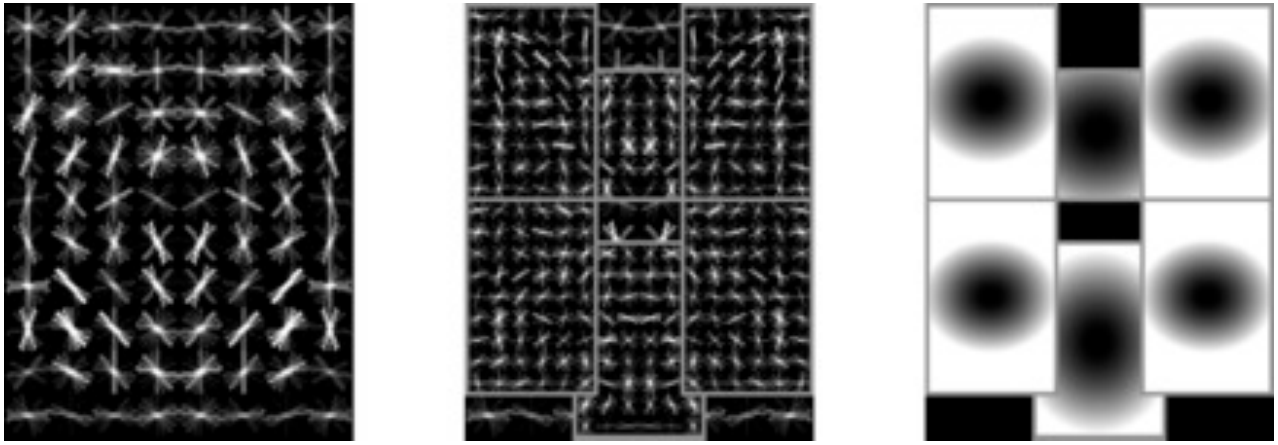
1 component from of each symmetric pair



Bottle

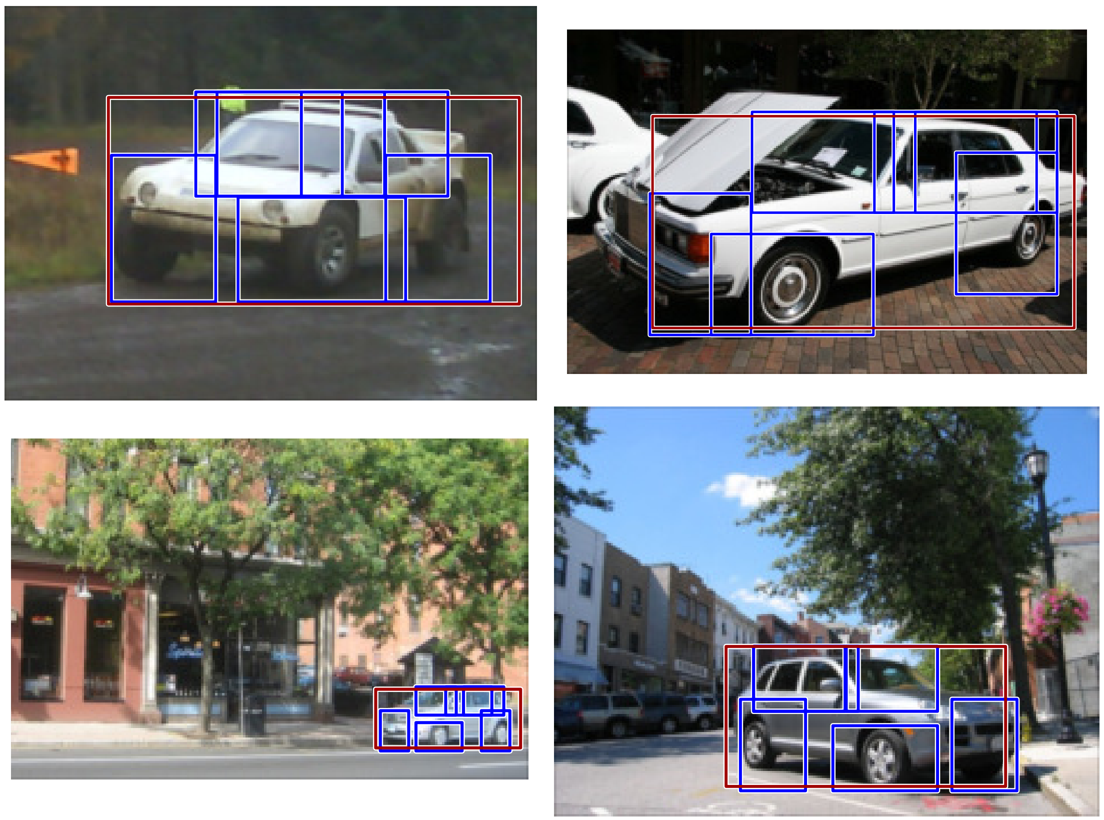


Cat

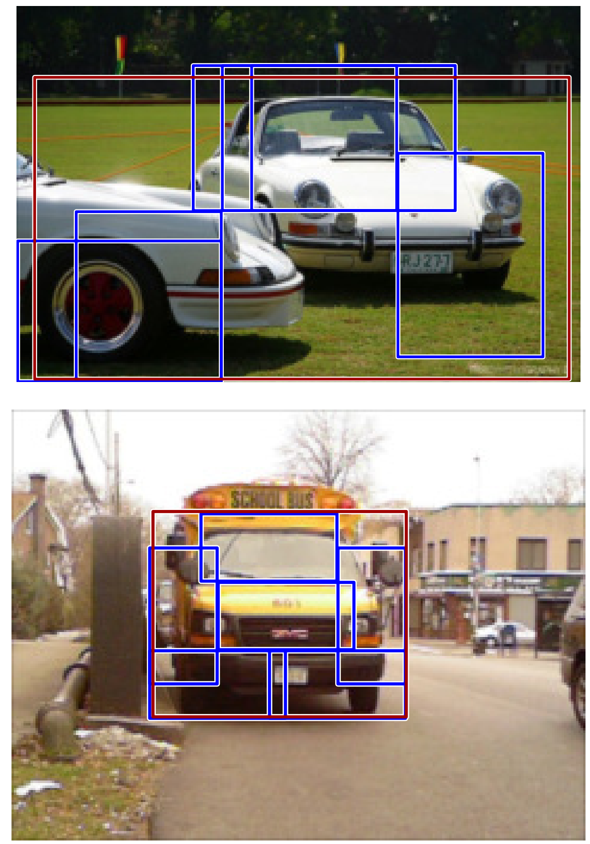


Car detections

high scoring true positives

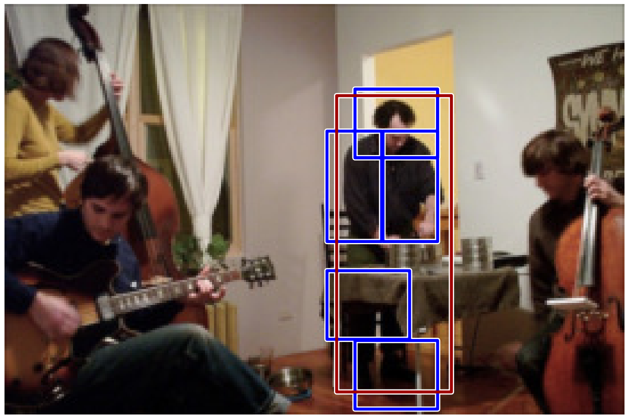


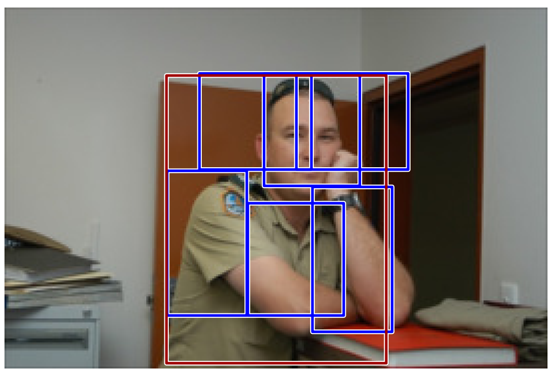
high scoring false positives

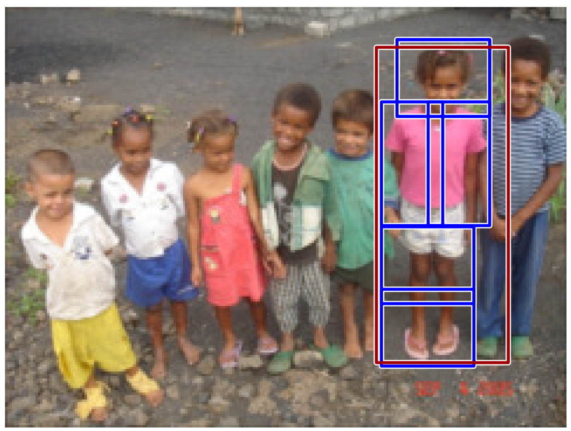


Person detections

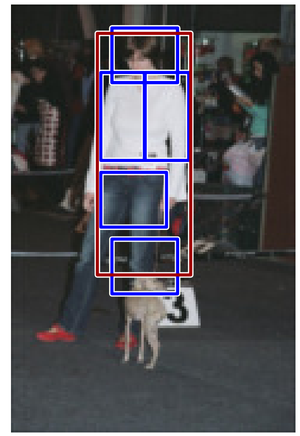
high scoring true positives





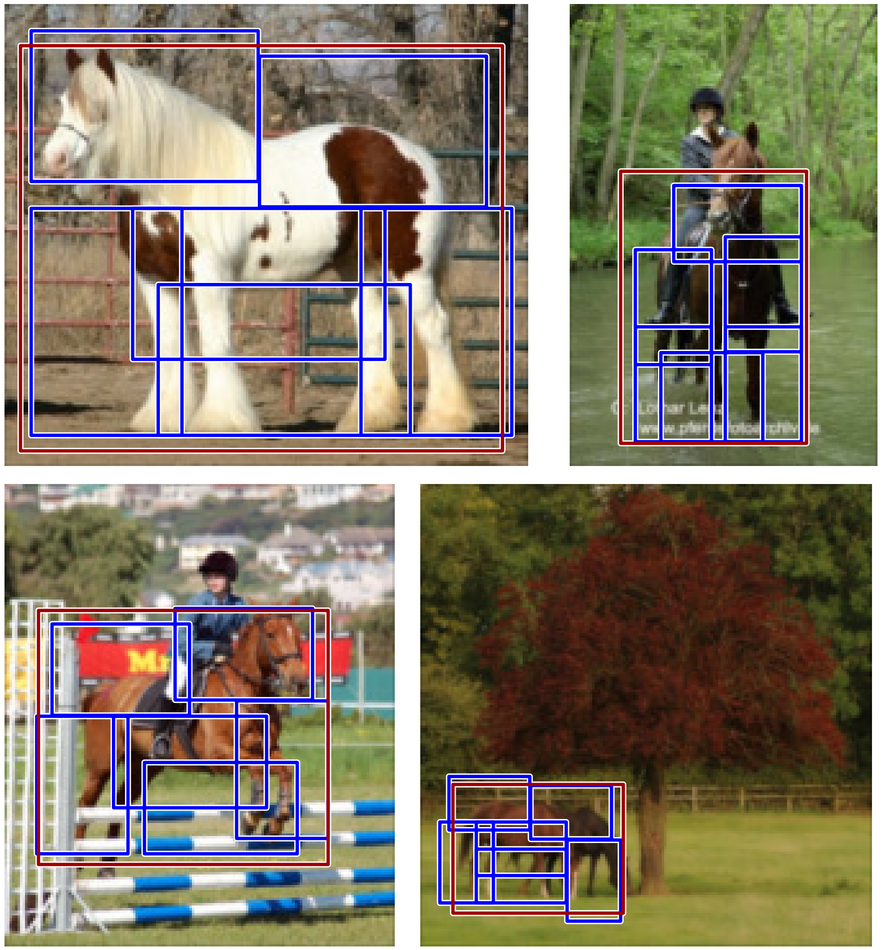


high scoring false positives
(not enough overlap)

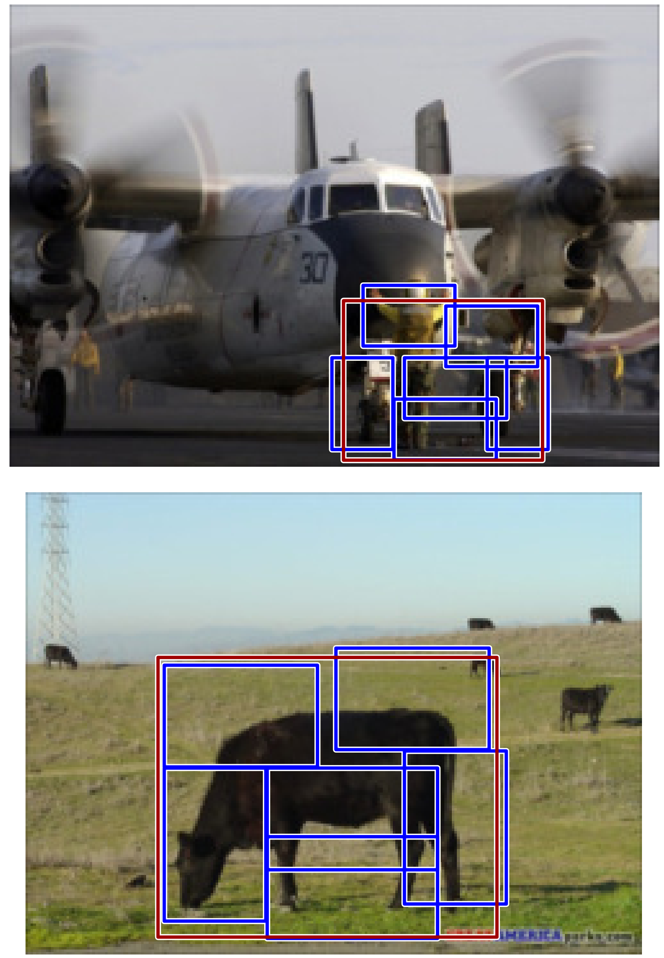


Horse detections

high scoring true positives

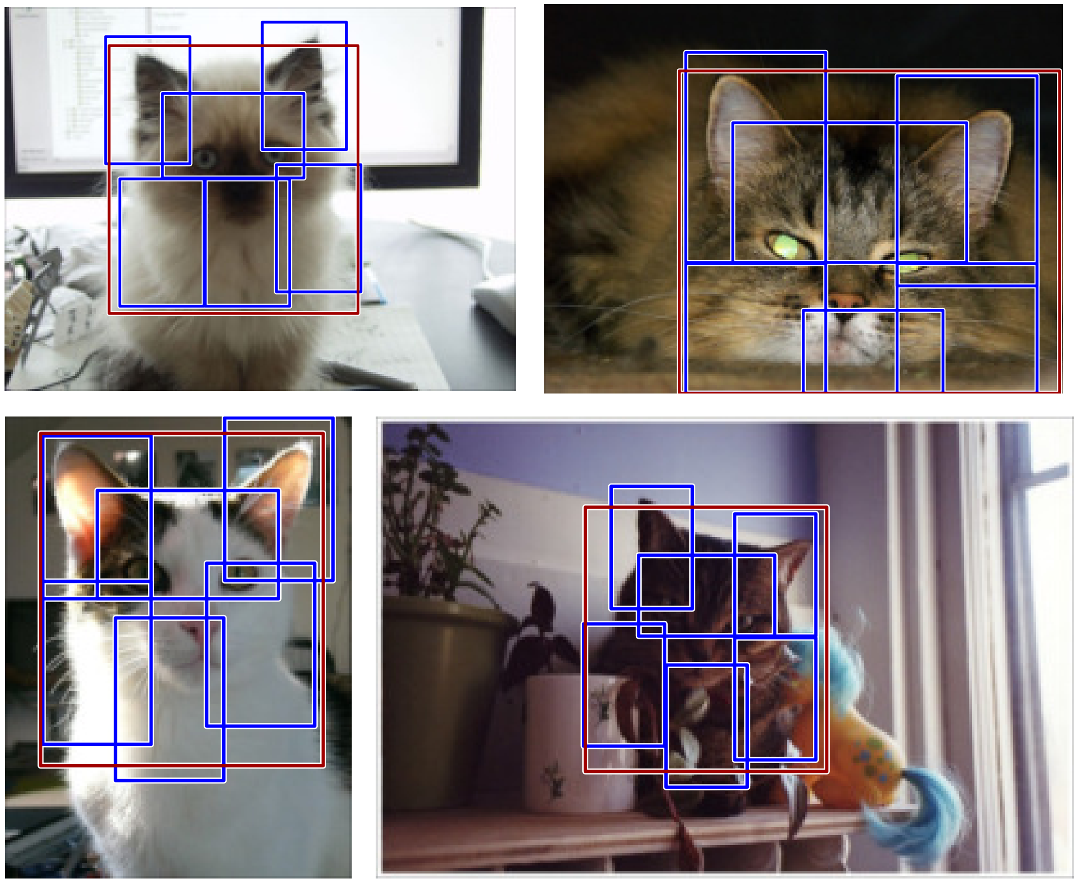


high scoring false positives

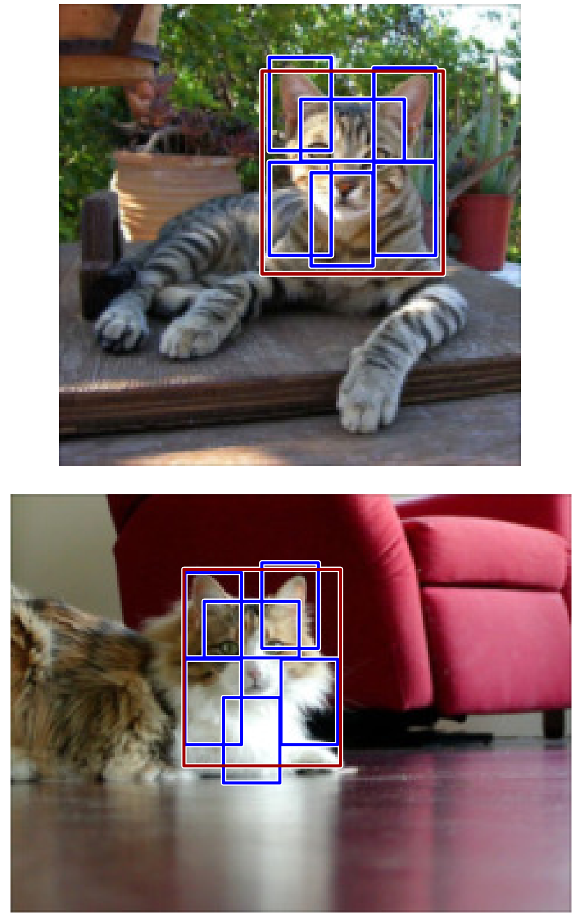


Cat detections

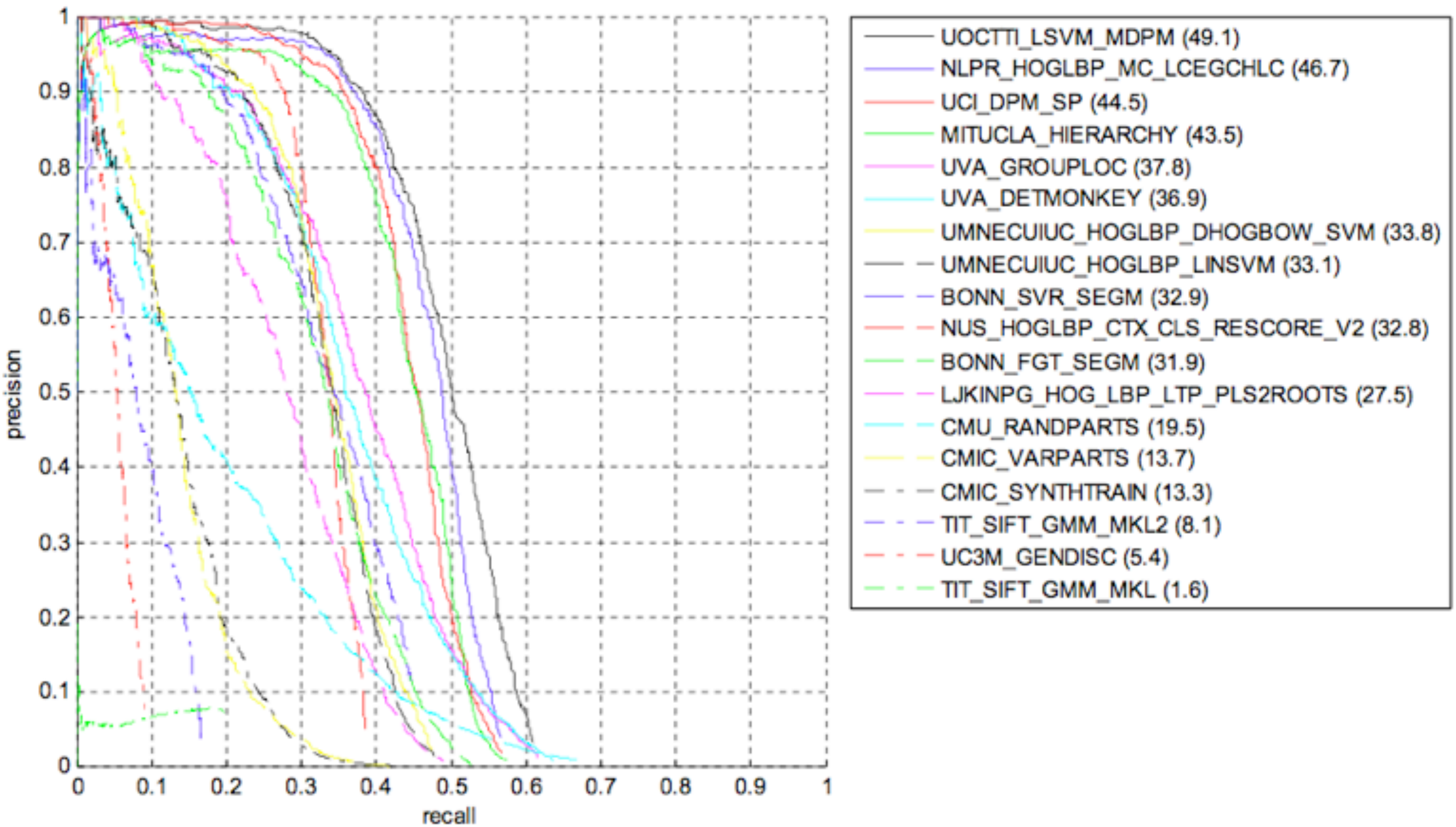
high scoring true positives



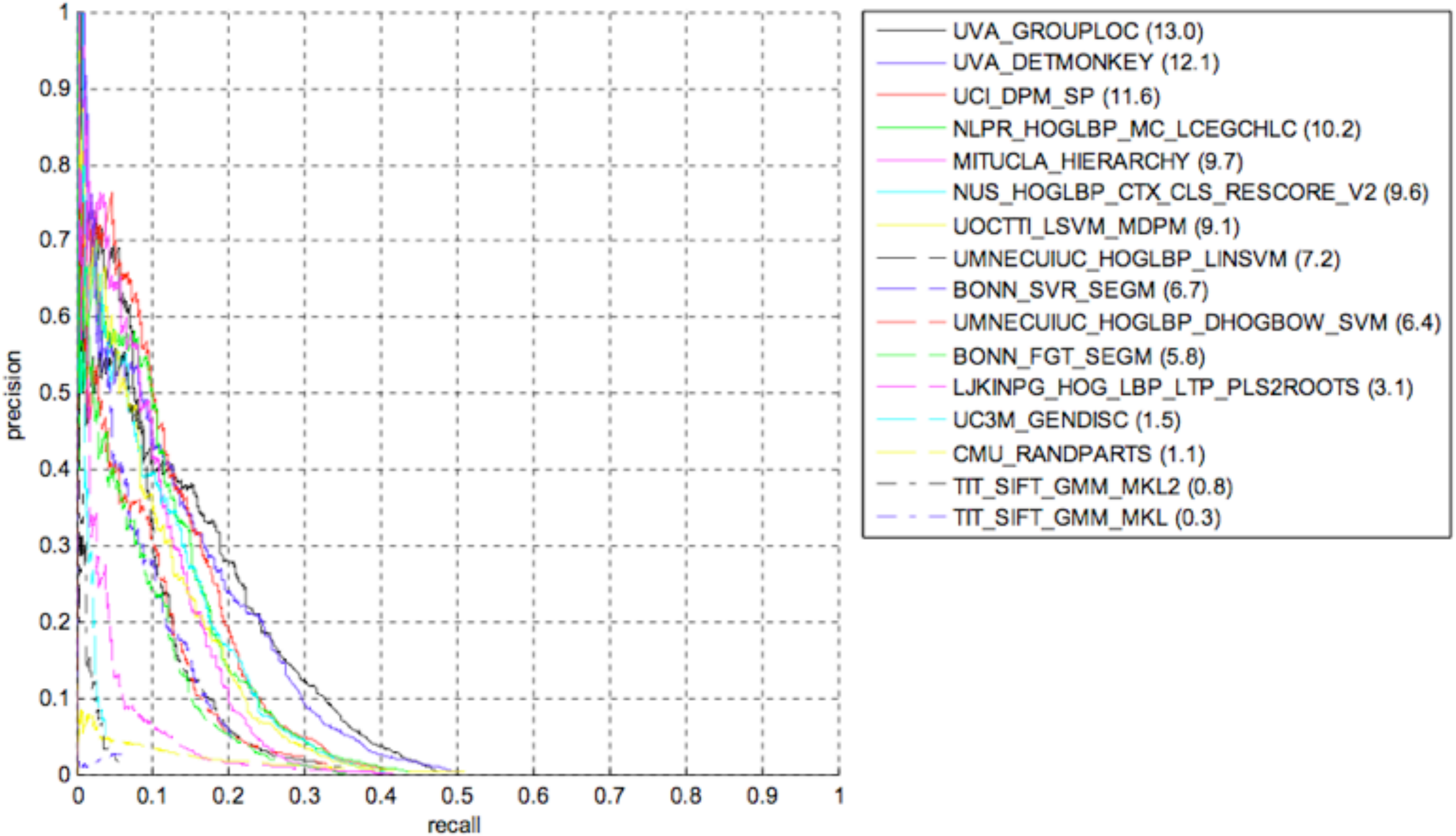
high scoring false positives
(not enough overlap)



Precision/Recall results on Cars 2010



Precision/Recall results on Plants 2010



Comparison of Car models