Object Detection

Vinay P. Namboodiri

- Slide credits to Navneet Dalal and Pedro Felzenszwalb

Overview of Methodology

Detection Phase

Scan image(s) at all scales and locations

↓

Extract features over windows

↓

Run linear SVM classifier on all locations

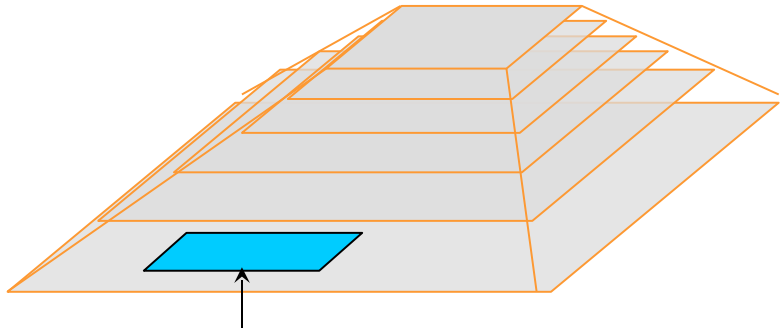
↓

Fuse multiple detections in 3-D position & scale space

↓

Object detections with bounding boxes

Scale-space pyramid

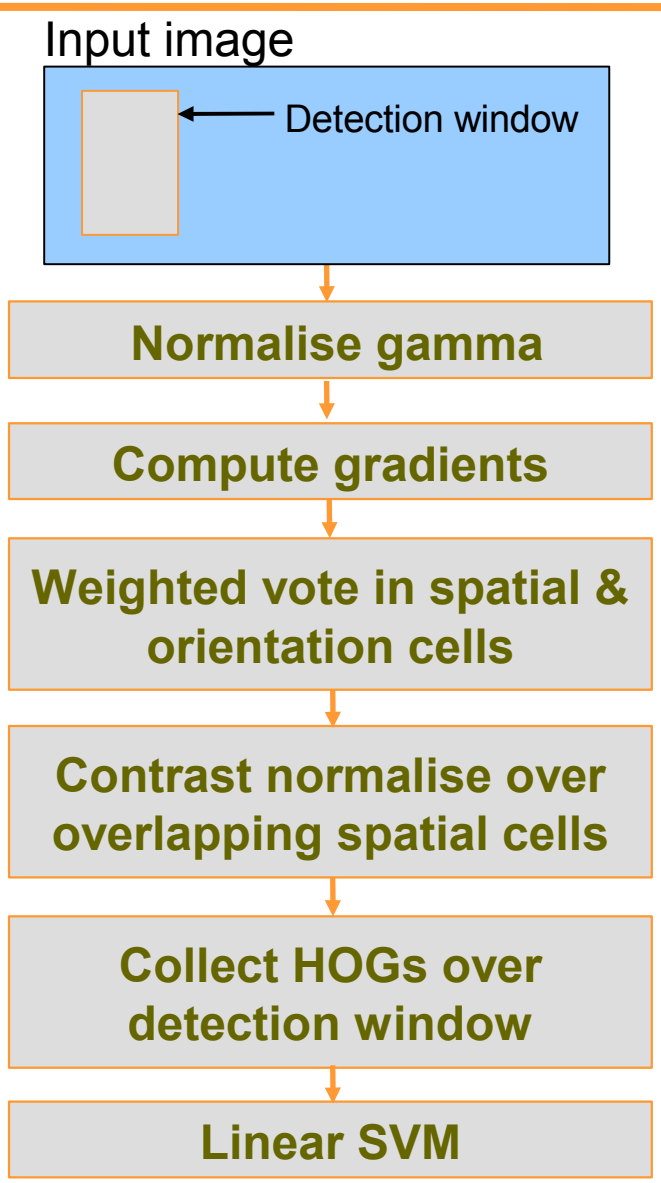


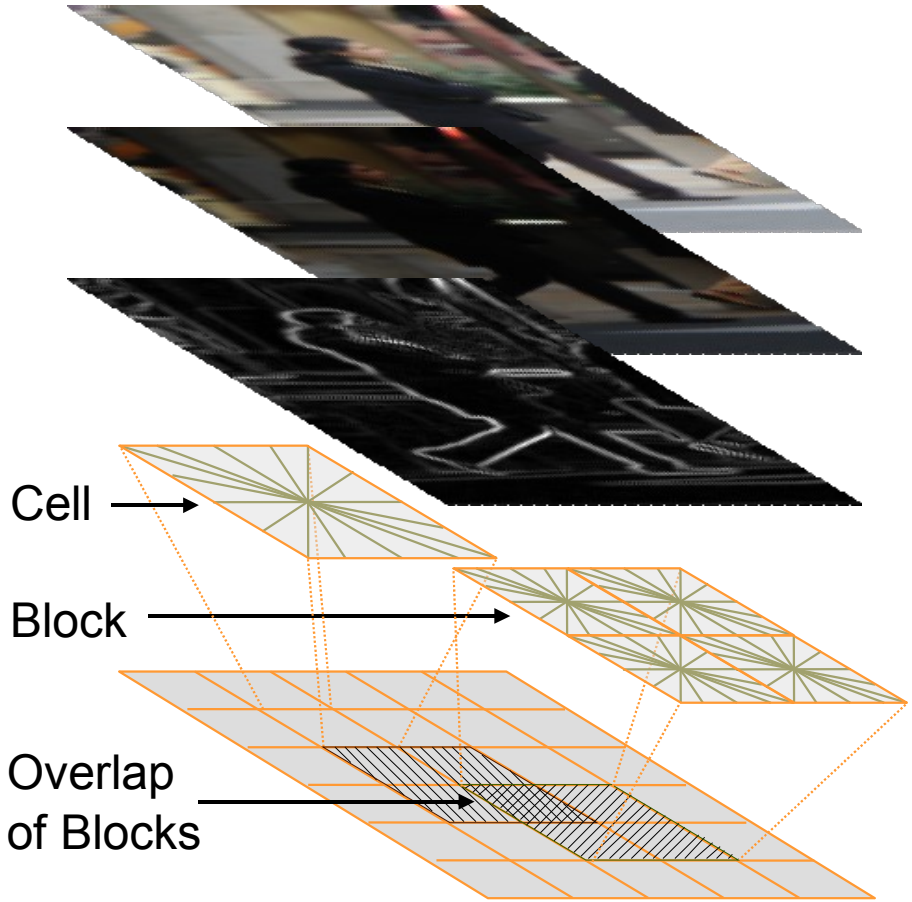
Detection window

Focus on building robust feature sets (static & motion)

Finding People in Images

Static Feature Extraction

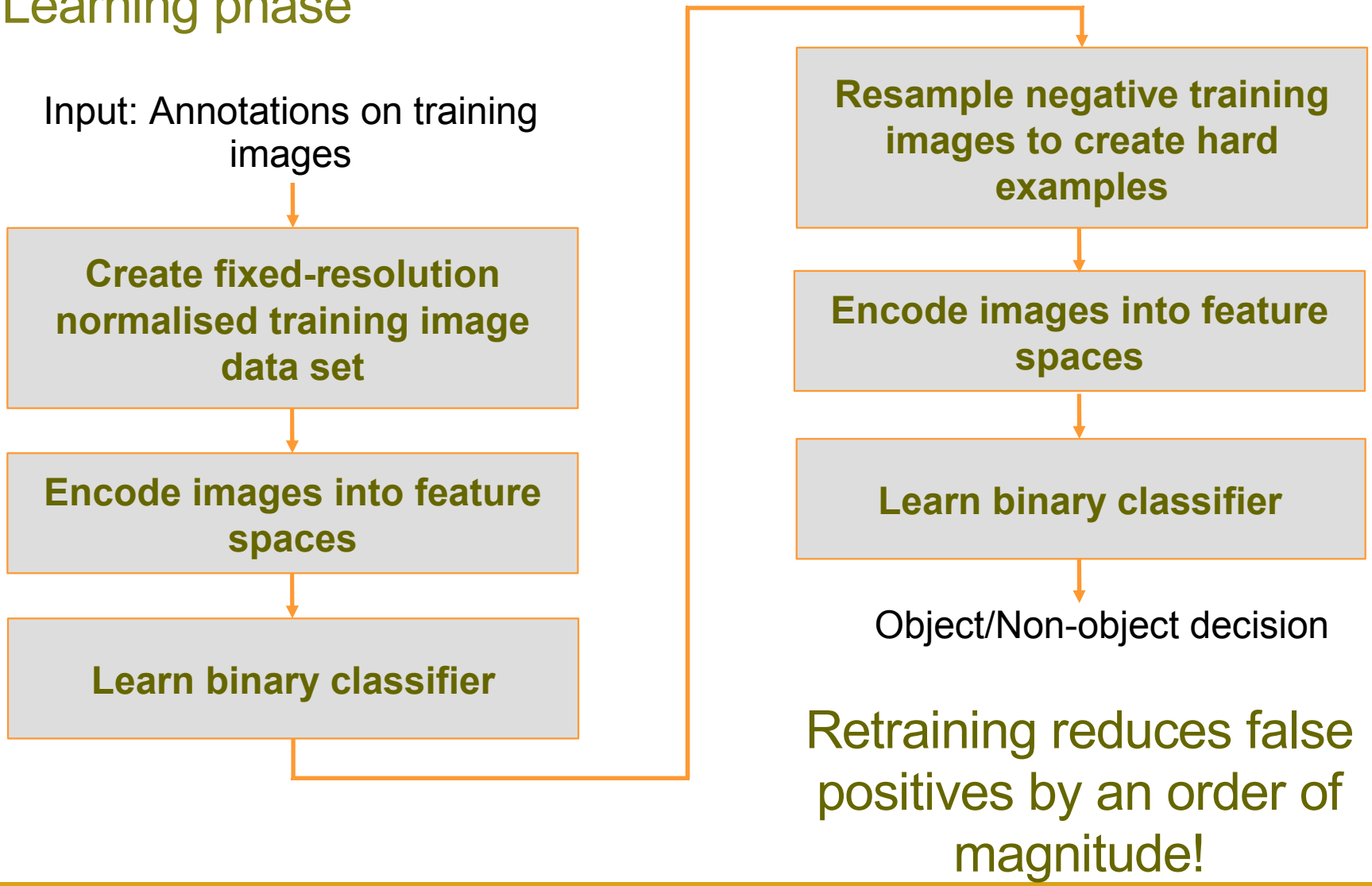




Feature vector f = [..., ..., ...]

Overview of Learning Phase

Learning phase



Retraining reduces false positives by an order of magnitude!

HOG Descriptors

Parameters

- Gradient scale
- Orientation bins
- Percentage of block overlap

Schemes

- RGB or Lab, colour/gray-space
- Block normalisation

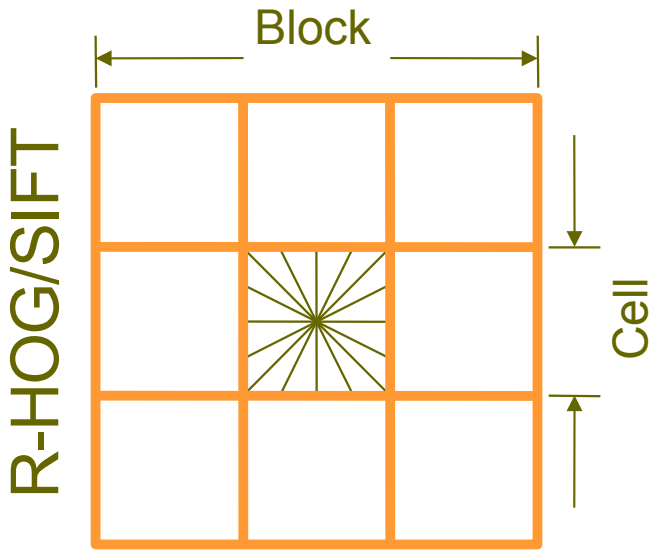
L2-norm,

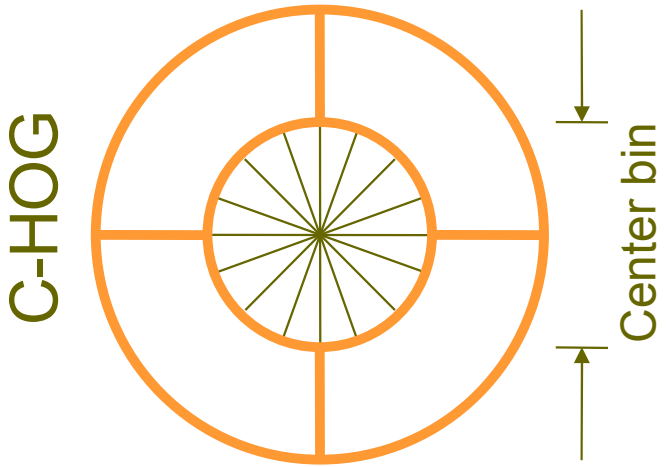
$$v \leftarrow v / \sqrt{\|v\|_2^2 + \varepsilon}$$

or

L1-norm,

$$v \leftarrow \sqrt{v / (\|v\|_1 + \varepsilon)}$$



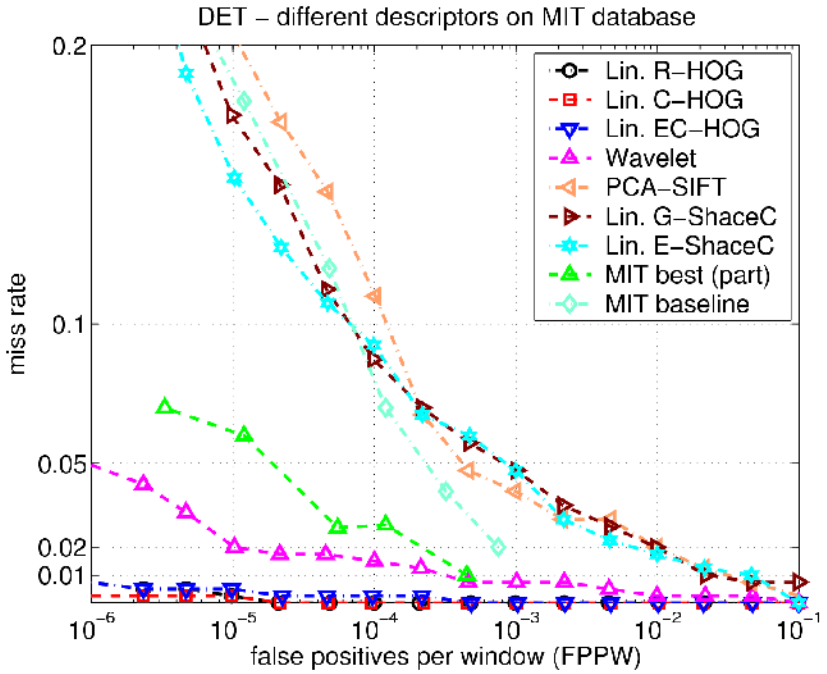


Evaluation Data Sets

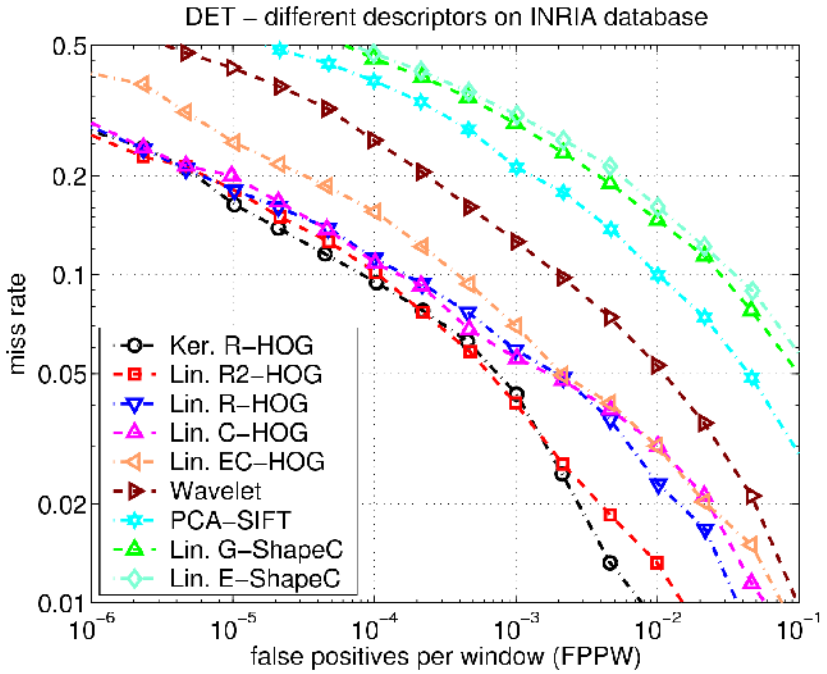
MIT pedestrian database		INRIA person database	
			
Train	507 positive windows Negative data unavailable	Train	1208 positive windows 1218 negative images
Test	200 positive windows Negative data unavailable	Test	566 positive windows 453 negative images
Overall 709 annotations+ reflections		Overall 1774 annotations+ reflections	

Overall Performance

MIT pedestrian database



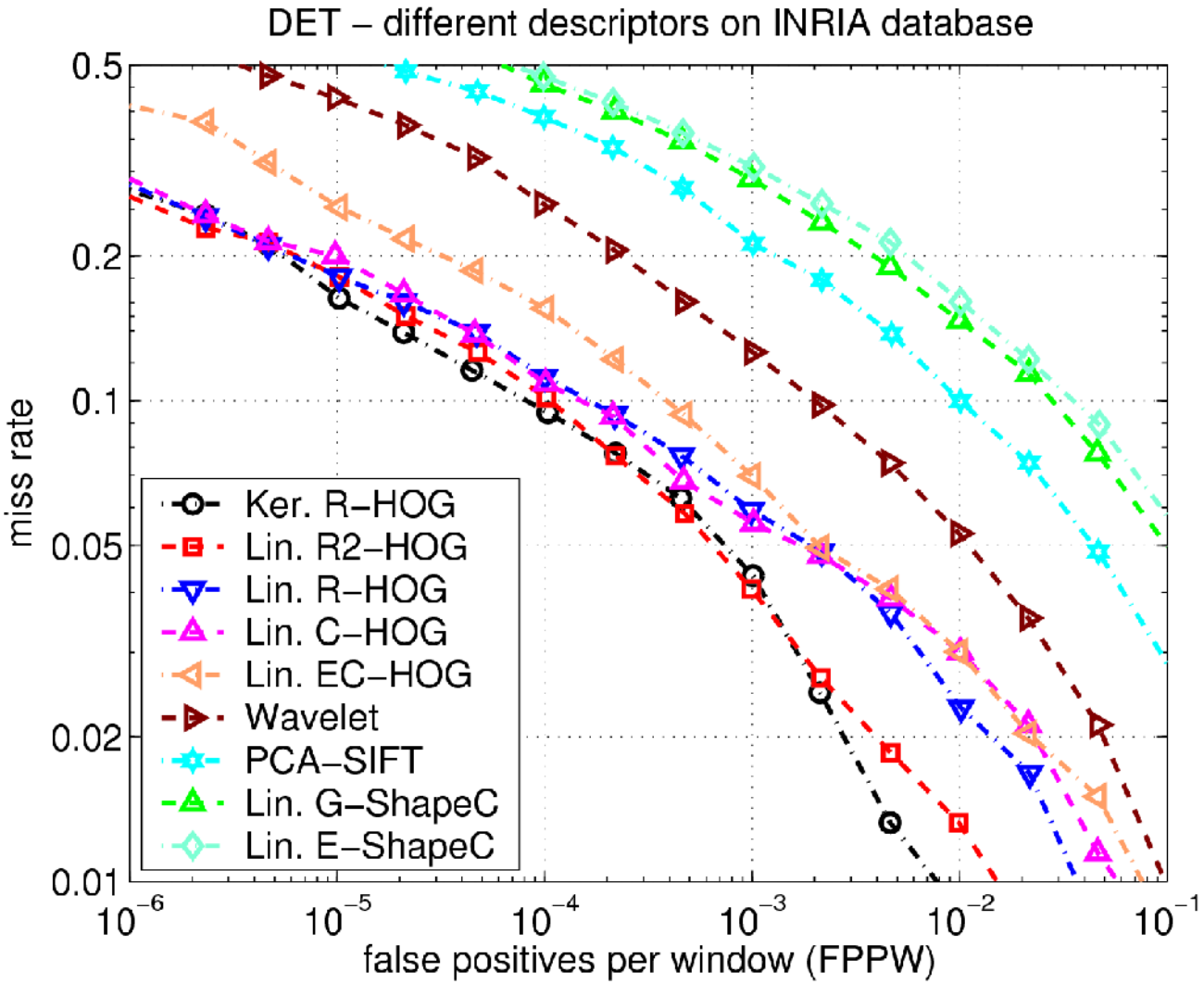
INRIA person database



R/C-HOG give near perfect separation on MIT database

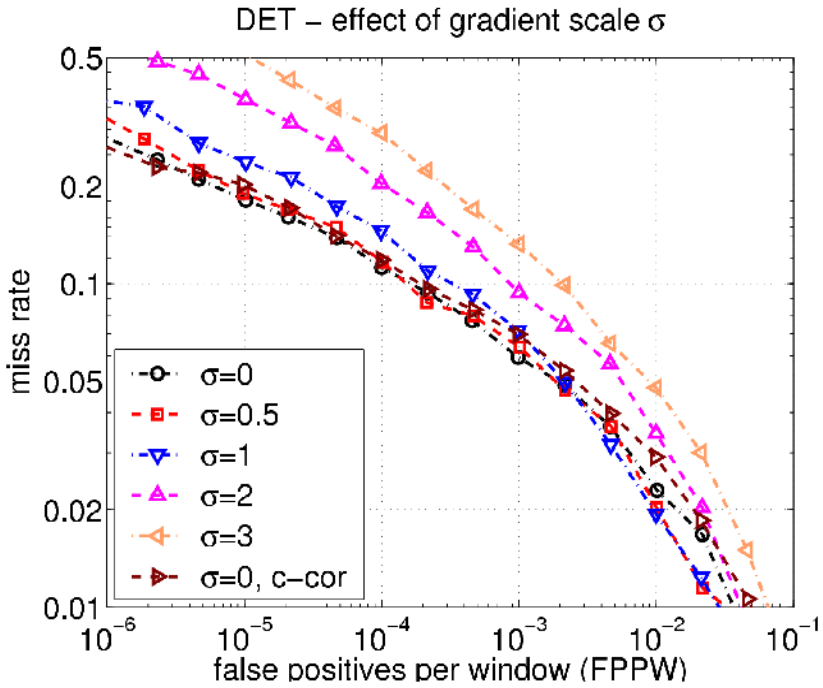
Have 1-2 order lower false positives than other descriptors

Performance on INRIA Database



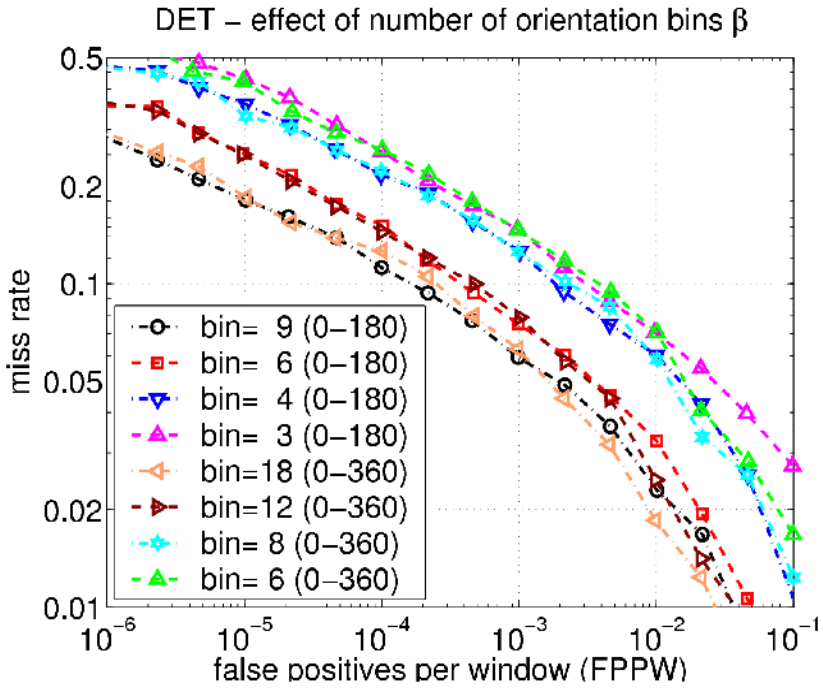
Effect of Parameters

Gradient smoothing, σ



Reducing gradient scale from 3 to 0 decreases false positives by 10 times

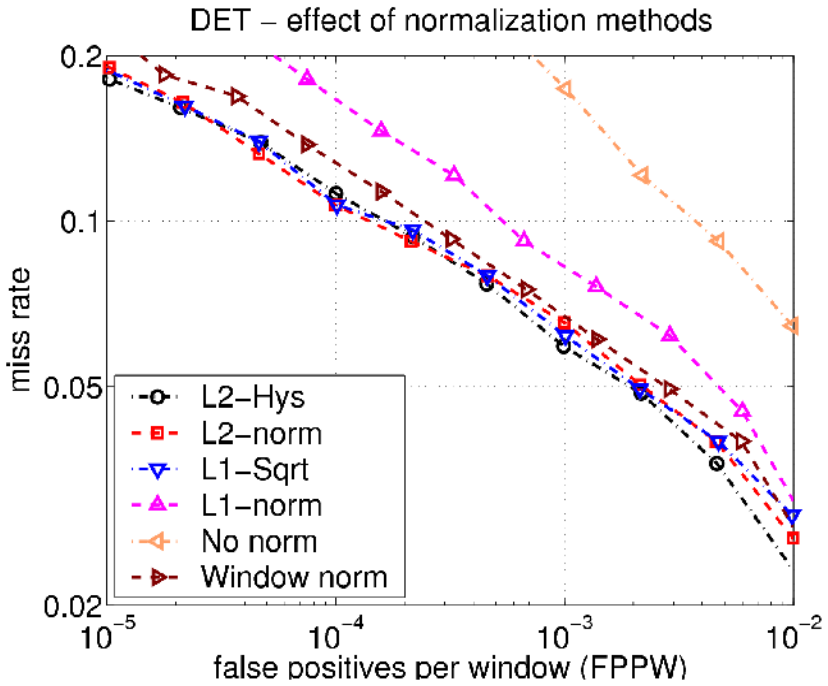
Orientation bins, β



Increasing orientation bins from 4 to 9 decreases false positives by 10 times

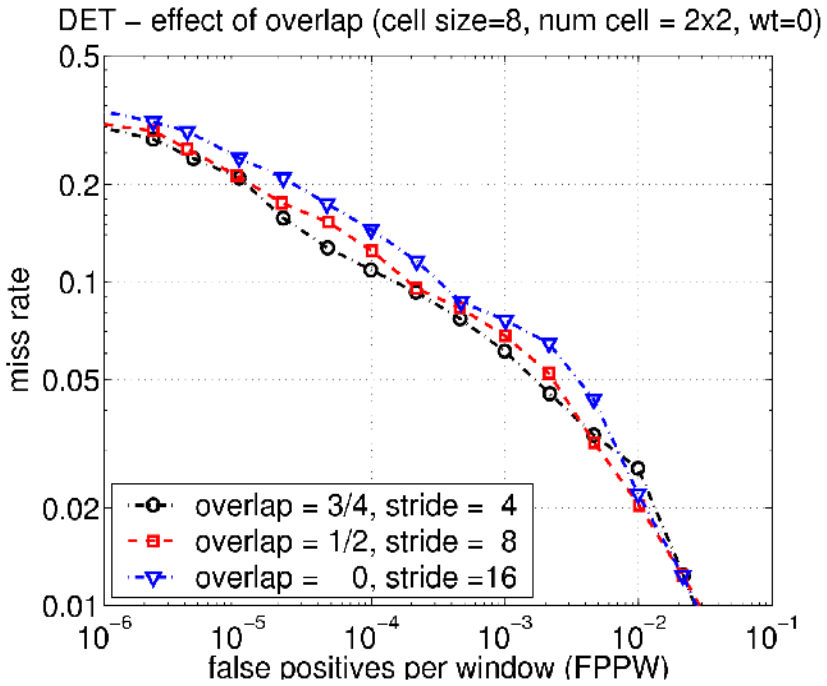
Normalisation Method & Block Overlap

Normalisation method



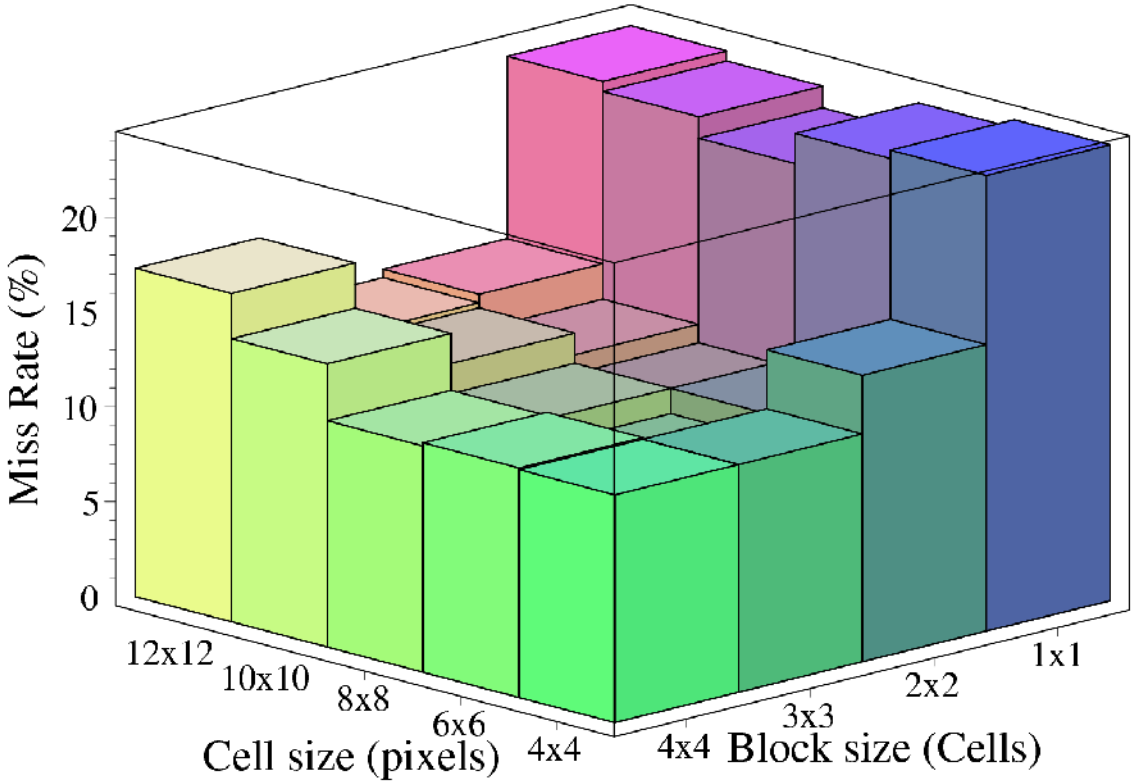
Strong local normalisation is essential

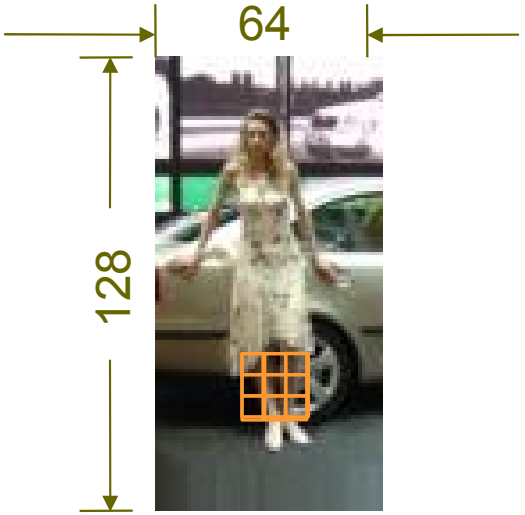
Block overlap



Overlapping blocks improve performance, but descriptor size increases

Effect of Block and Cell Size

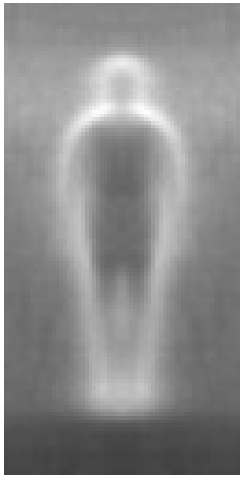




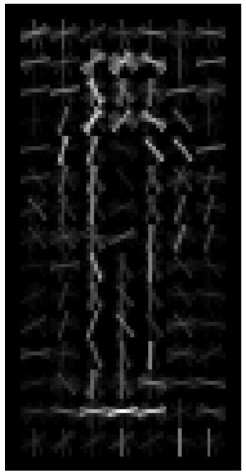
Trade off between need for local spatial invariance and need for finer spatial resolution

Descriptor Cues

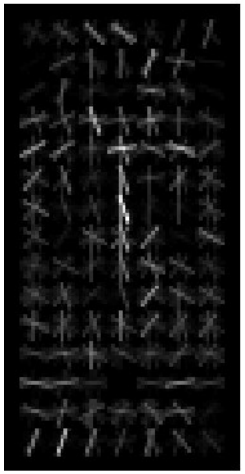
Input example



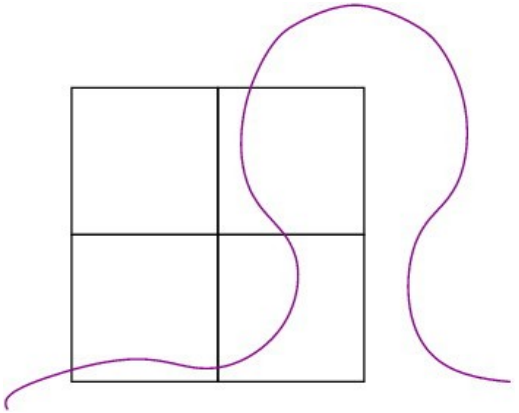
Average gradients



Weighted pos wts



Weighted neg wts



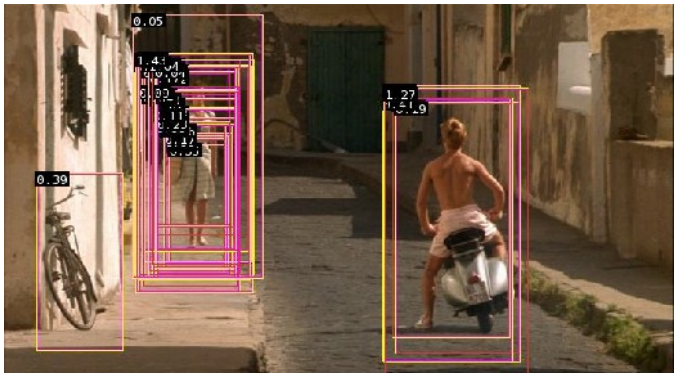
Outside-in weights

Most important cues are head, shoulder, leg silhouettes

Vertical gradients inside a person are counted as negative

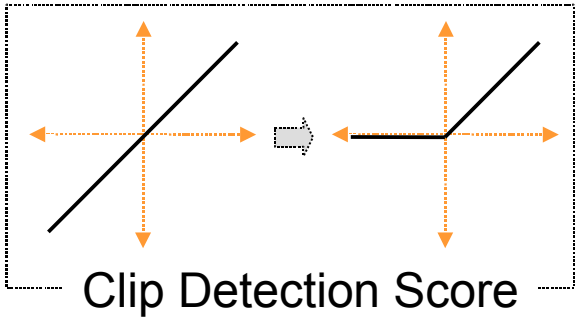
Overlapping blocks just outside the contour are most important

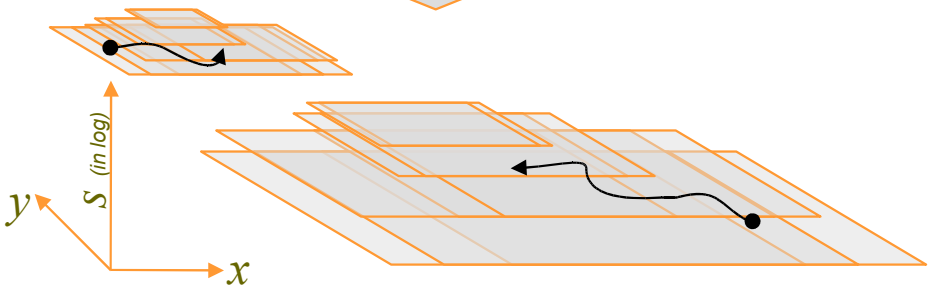
Multi-Scale Object Localisation

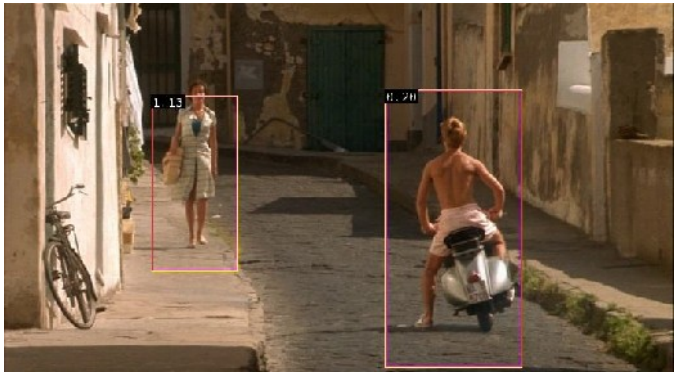


Multi-scale dense scan of detection window

Bias







Final detections

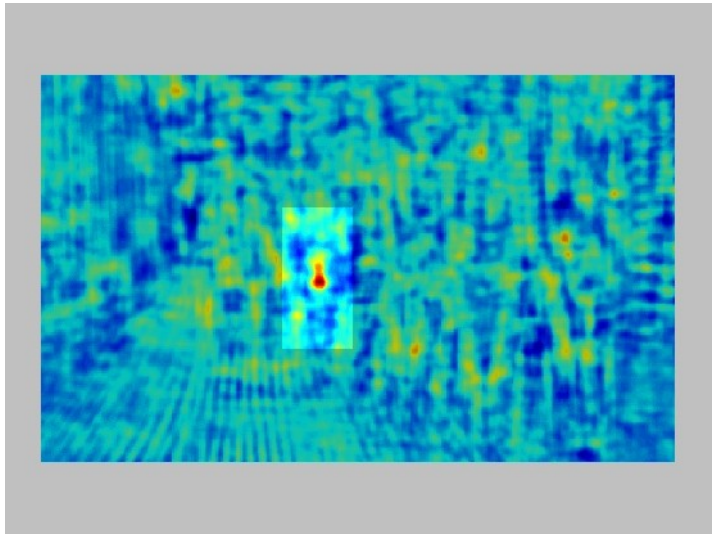
Threshold

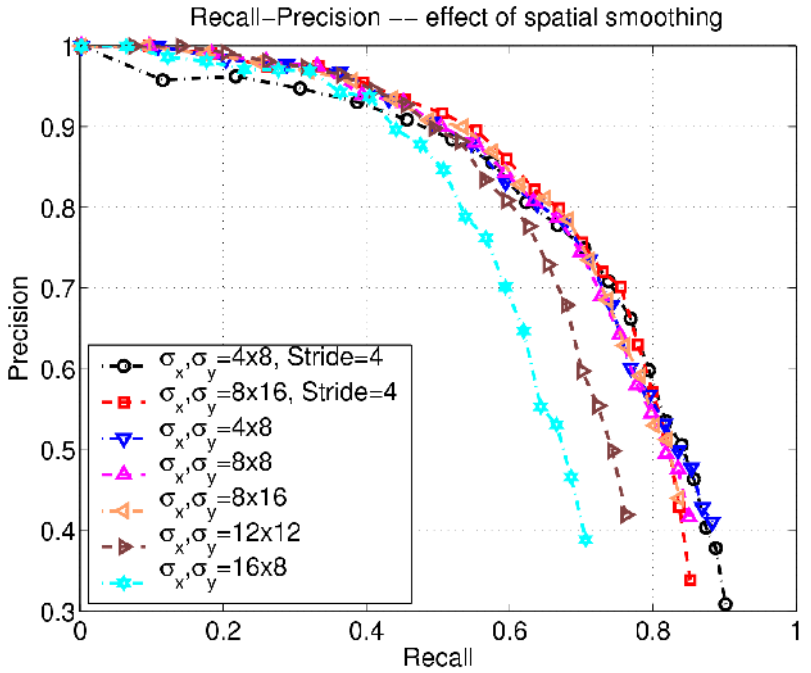
$$\mathbf{H}_i = [\exp(s_i)\sigma_x, \exp(s_i)\sigma_y, \sigma_s]$$

$$f(\mathbf{x}) = \sum_i^n w_i \exp\left(-\left\|(\mathbf{x}-\mathbf{x}_i)/\mathbf{H}_i^{-1}\right\|^2/2\right)$$

Apply robust mode detection, like mean shift

Effect of Spatial Smoothing



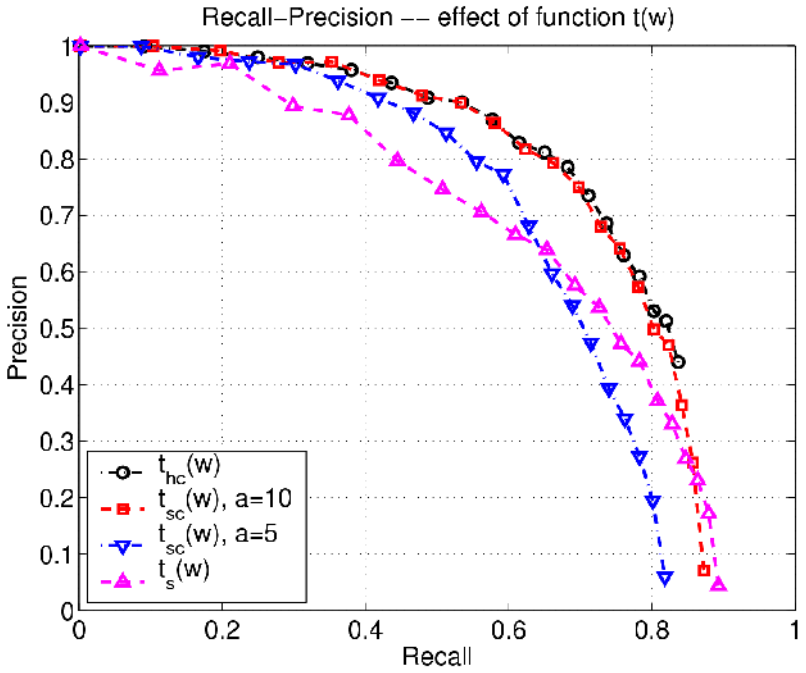


Spatial smoothing aspect ratio as per window shape, smallest sigma approx. equal to stride/cell size

Relatively independent of scale smoothing, sigma equal to 0.4 to 0.7 octaves gives good results

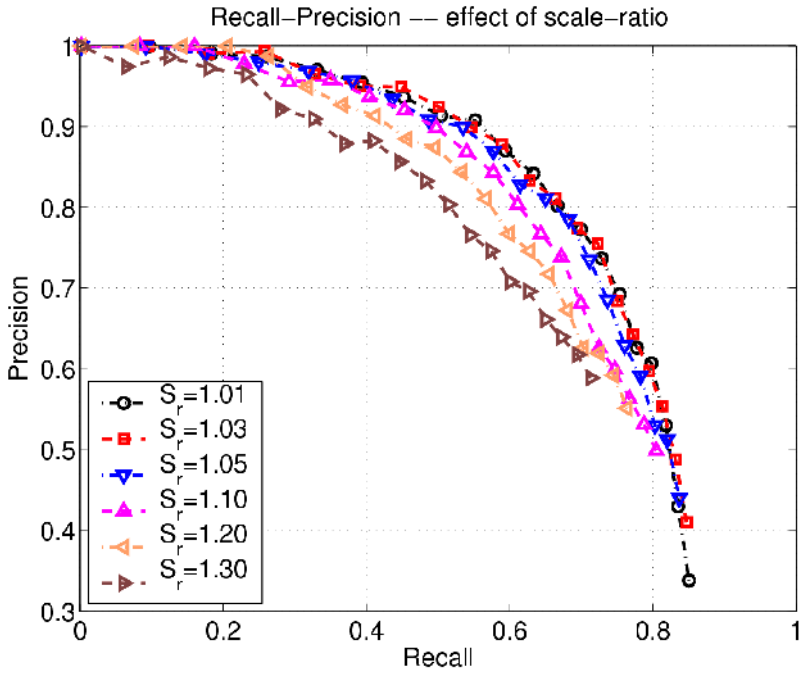
Effect of Other Parameters

Different mappings



Hard clipping of SVM scores gives the best results than simple probabilistic mapping of these scores

Effect of scale-ratio



Fine scale sampling helps improve recall

Results Using Static HOG

No temporal smoothing of detections

Conclusions for Static Case

Fine grained features improve performance

Rectify fine gradients then pool spatially

- No gradient smoothing, [1 0 -1] derivative mask
- Orientation voting into fine bins
- Spatial voting into coarser bins

Use gradient magnitude (no thresholding)

Strong local normalization

Use overlapping blocks

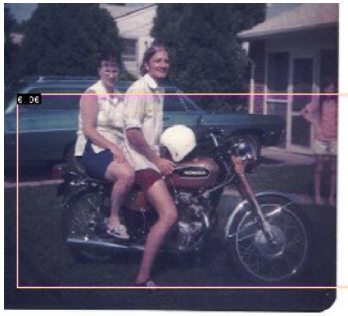
Robust non-maximum suppression

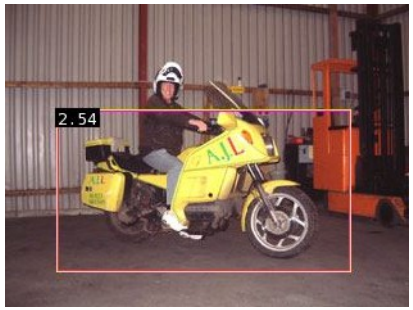
- Fine scale sampling, hard clipping & anisotropic kernel

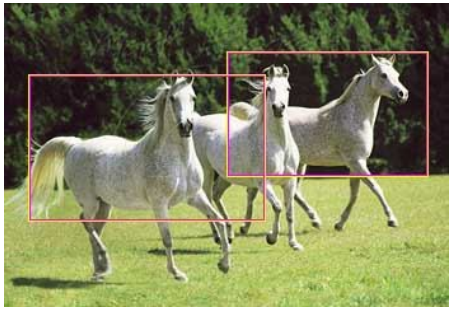
Human detection rate of 90% at 10^{-4} false positives per window

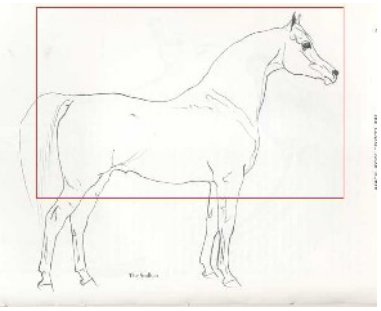
Slower than integral images of Viola & Jones, 2001

Applications to Other Classes









Parameter Settings

- Most HOG parameters are stable across different classes
- Parameters that change
  - Gamma compression
  - Normalisation methods
  - Signed/un-signed gradients

Results from Pascal VOC 2006

	Person	Car	Motorbike	Bicycle	Bus	Sheep	Horse	Cow	Cat	Dog
Cambridge	0.030	0.254	0.178	0.249	0.138	0.131	0.091	0.149	0.151	**0.118**
ENSMP	-	0.398	-	-	-	-	-	0.159	-	-
HOG	**0.164**	**0.444**	**0.390**	0.414	0.117	**0.251**	-	0.212	-	-
Laptev= HOG+ Ada-boost	0.114	-	0.318	**0.440**	-	-	**0.140**	0.224	-	-
TUD	0.074	-	0.153	-	-	-	-	-	-	-
TKK	0.039	0.222	0.265	0.303	**0.169**	0.227	0.137	**0.252**	**0.160**	0.113

HOG outperformed other methods for 4 out of 10 classes

Its adaBoost variant outperformed other methods for 2 out of 10 classes

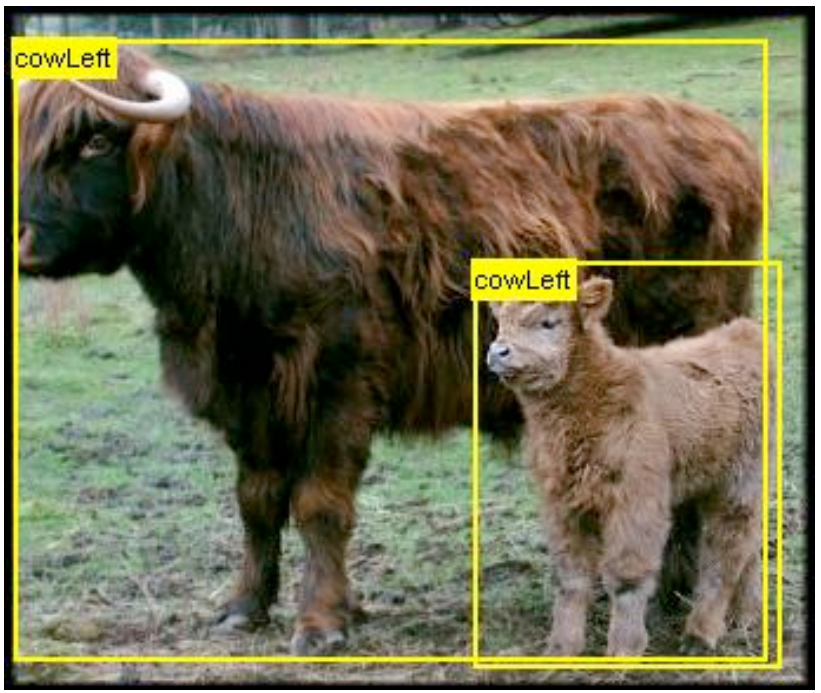
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb
Department of Computer Science
University of Chicago

Joint with David Mcallester, Deva Ramanan, Ross Girshick

PASCAL Challenge

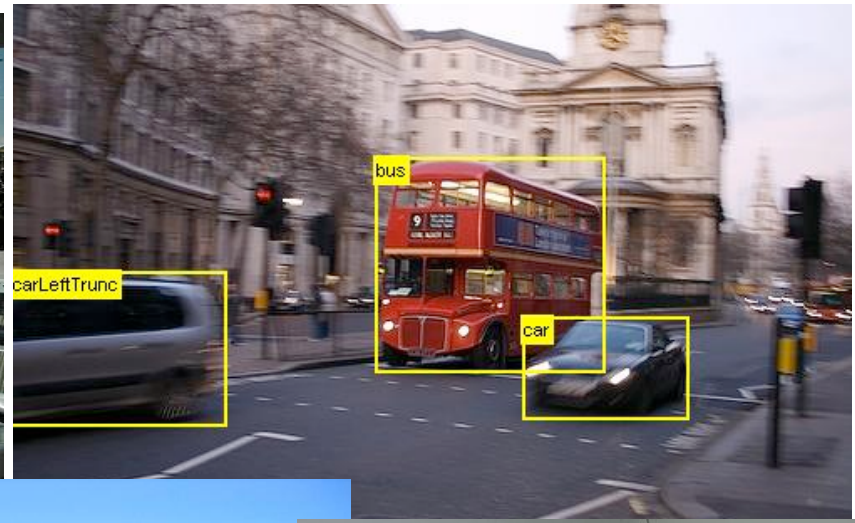
- ~10,000 images, with ~25,000 target objects
  - Objects from 20 categories (person, car, bicycle, cow, table...)
  - Objects are annotated with labeled bounding boxes

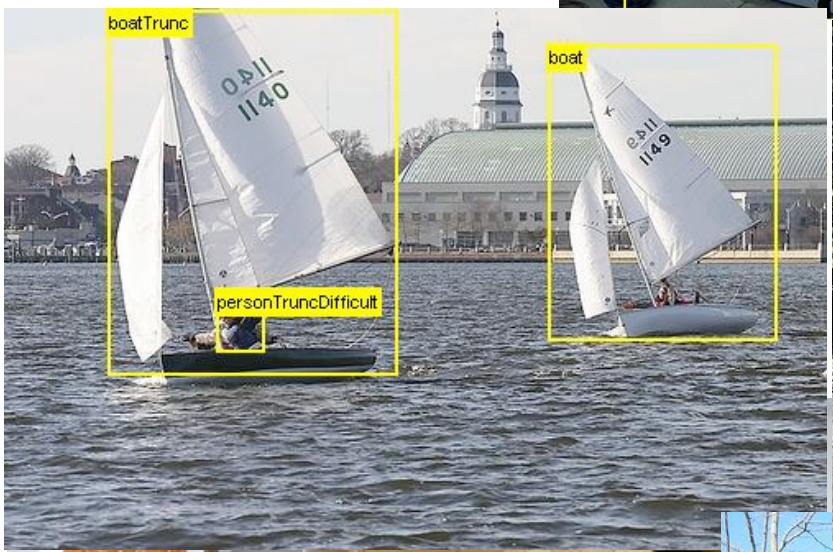


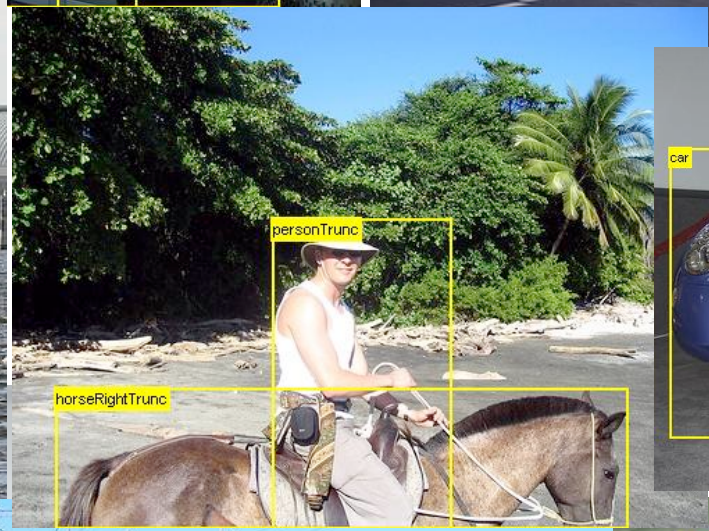

aeroplane


Bicycle
Assembly
Area
bicycleRight

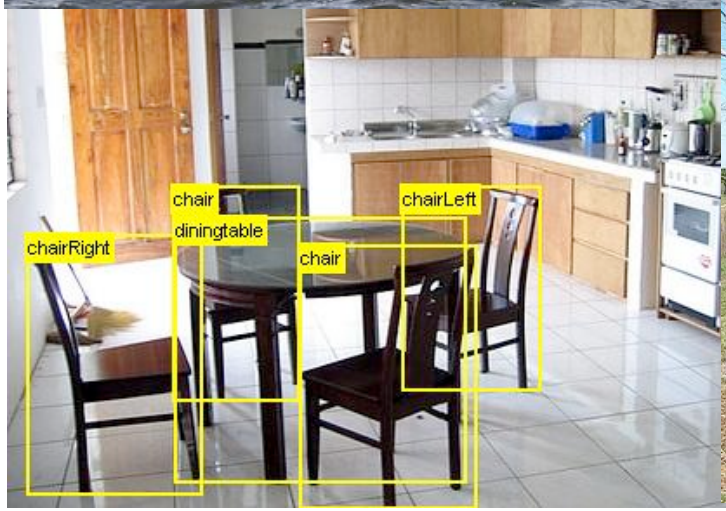

personLeftTrunc


bus
carLeftTrunc
car

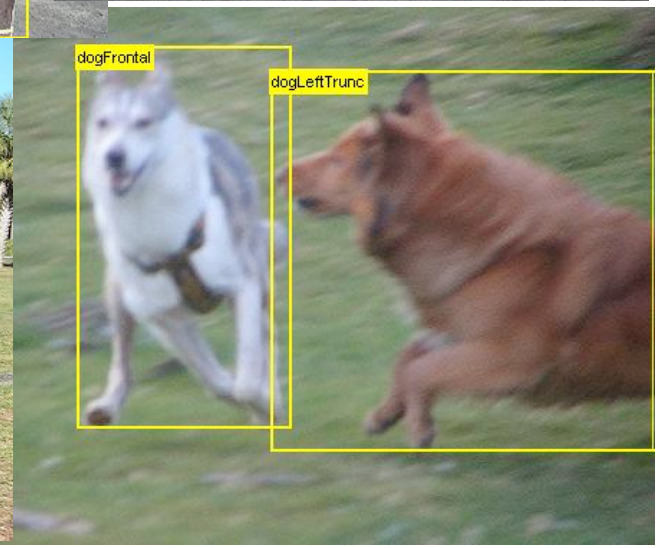

boatTrunc
personTruncDifficult
boat


personTrunc
horseRightTrunc


Morgan
car


chair
chairLeft
diningtable
chairRight
chair


motorbikeRear
motorbikeRear


dogFrontal
dogLeftTrunc

Why is it hard?

- Objects in rich categories exhibit significant variability
  - 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

We need rich object models
But this leads to difficult matching and training problems

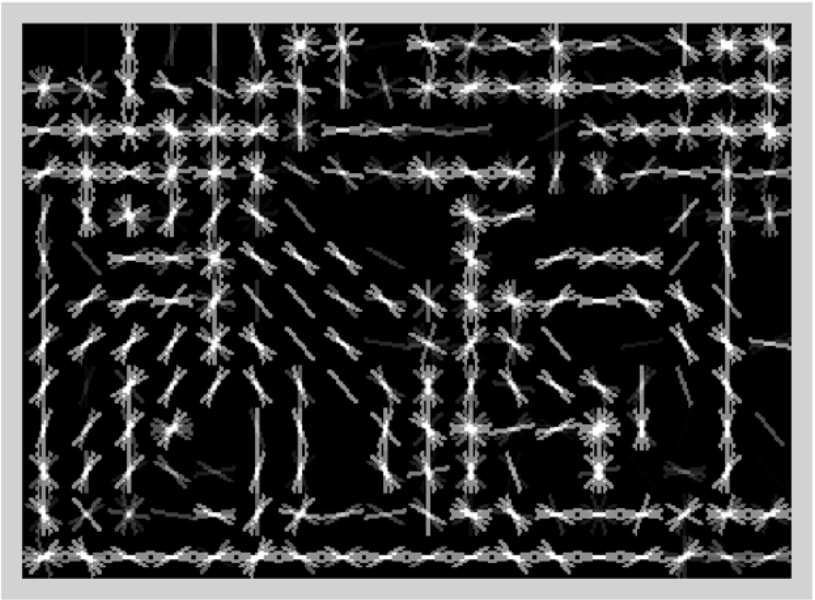
Starting point: sliding window classifiers

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 classifier
  - Previous state of the art for detecting people

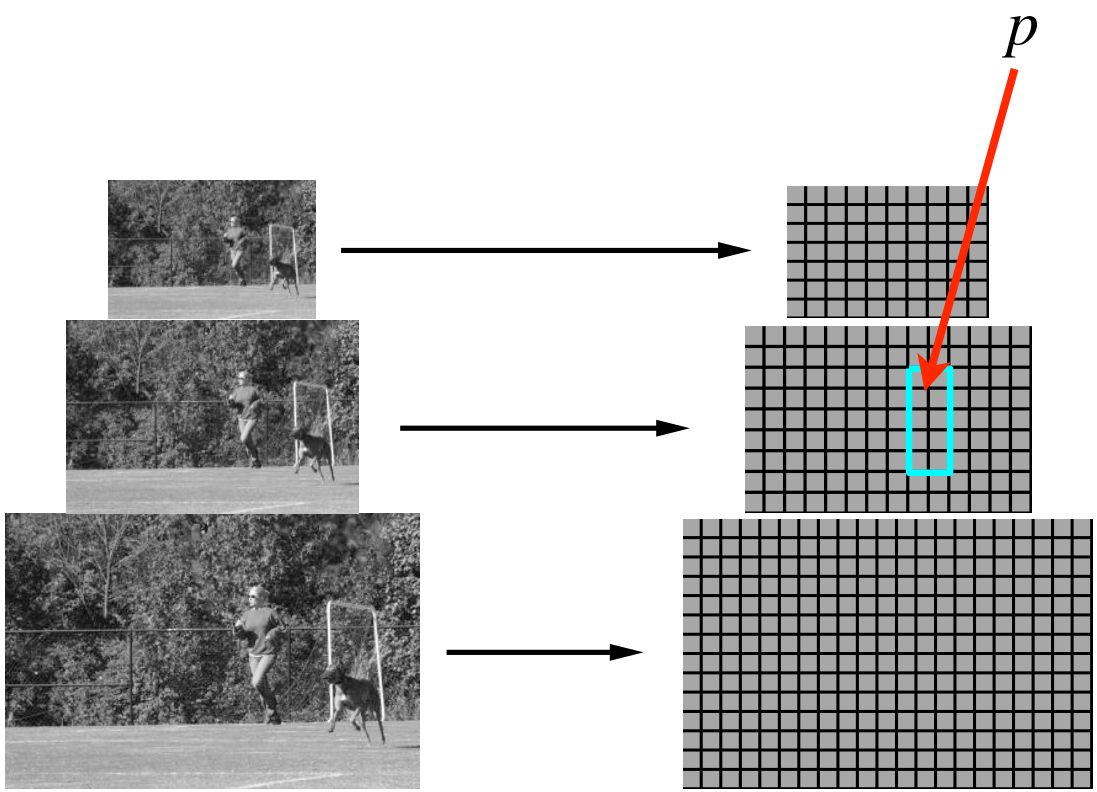
Histogram of Gradient (HOG) features



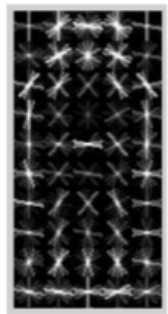
- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.
- Compute features at different resolutions (pyramid)

HOG Filters

- Array of weights for features in subwindow of HOG pyramid
- Score is dot product of filter and feature vector



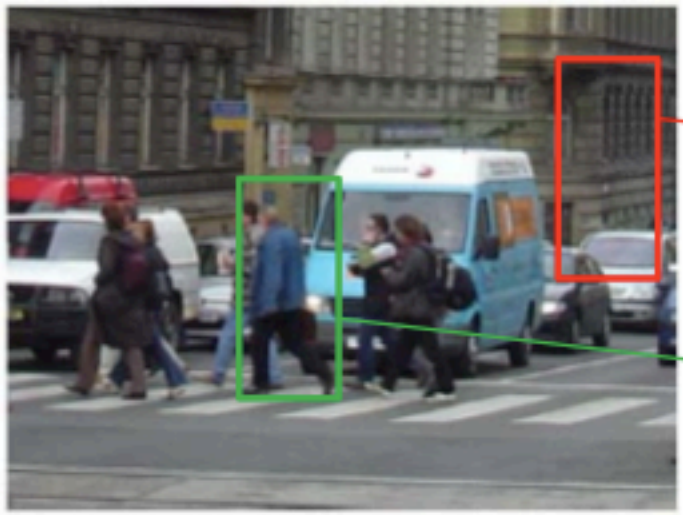
Filter F



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

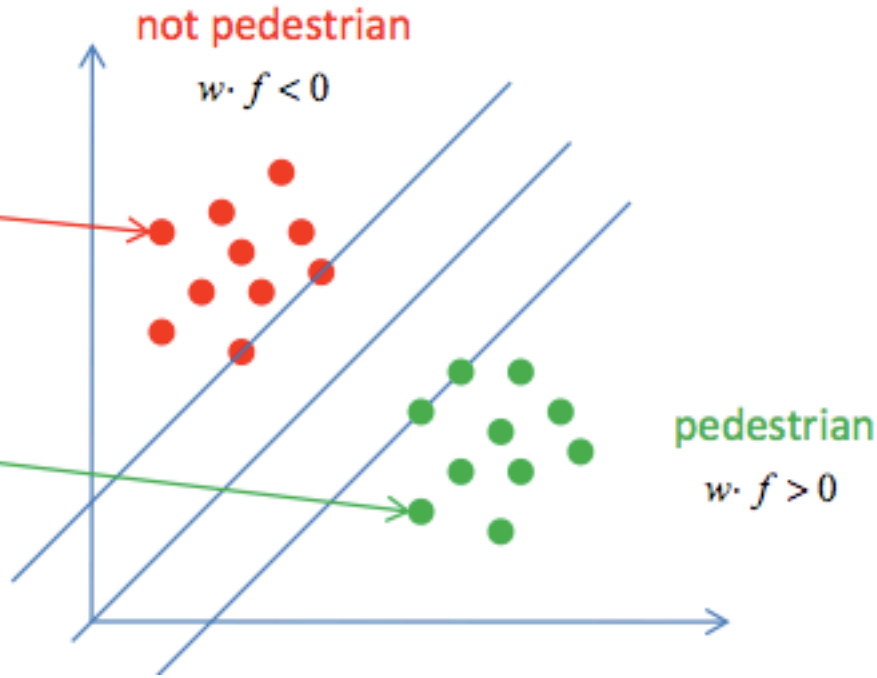
$\phi(p, H)$ = concatenation of HOG features from subwindow specified by p

Dalal & Triggs: HOG + linear SVMs



$\phi(p, H)$

$\phi(q, H)$



Typical form of a model

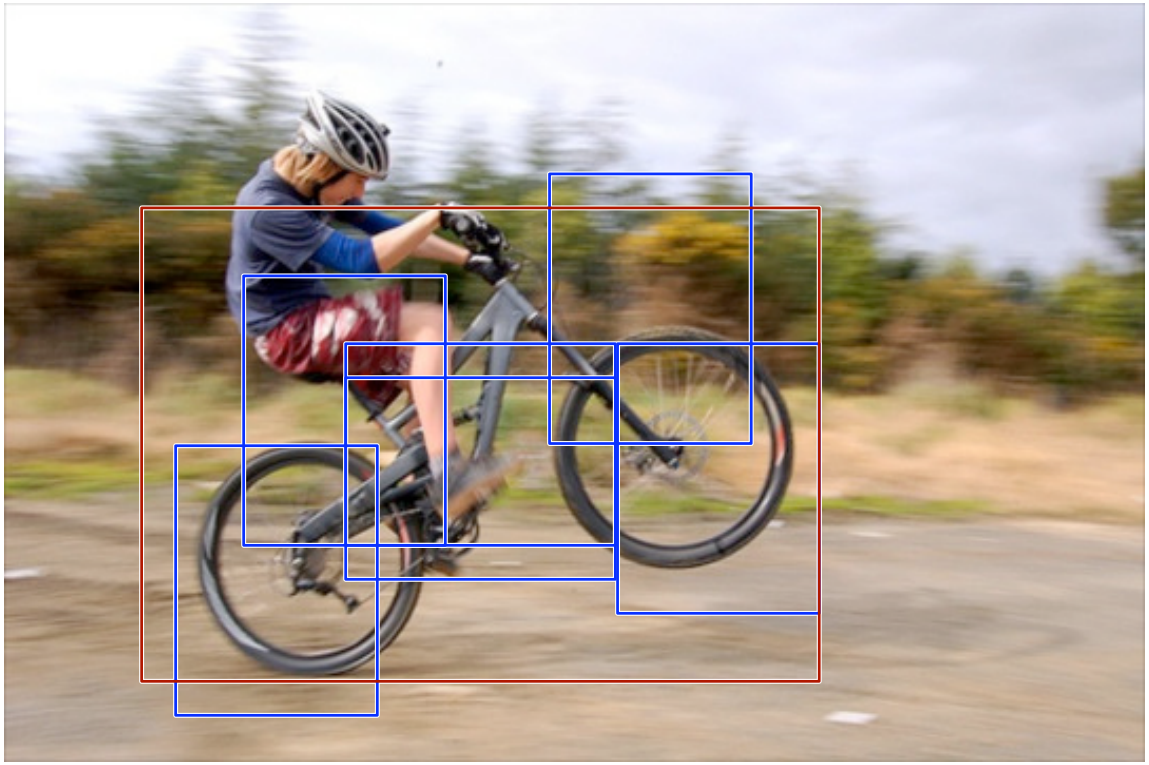
There is much more background than objects

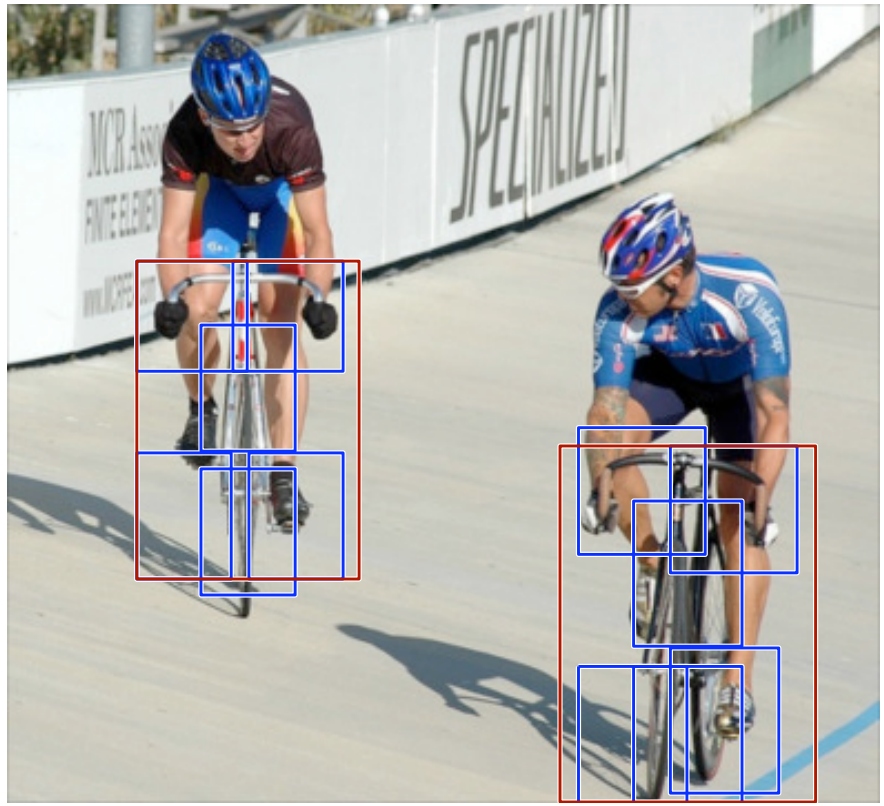
Start with random negatives and repeat:

1) Train a model

2) Harvest false positives to define “hard negatives”

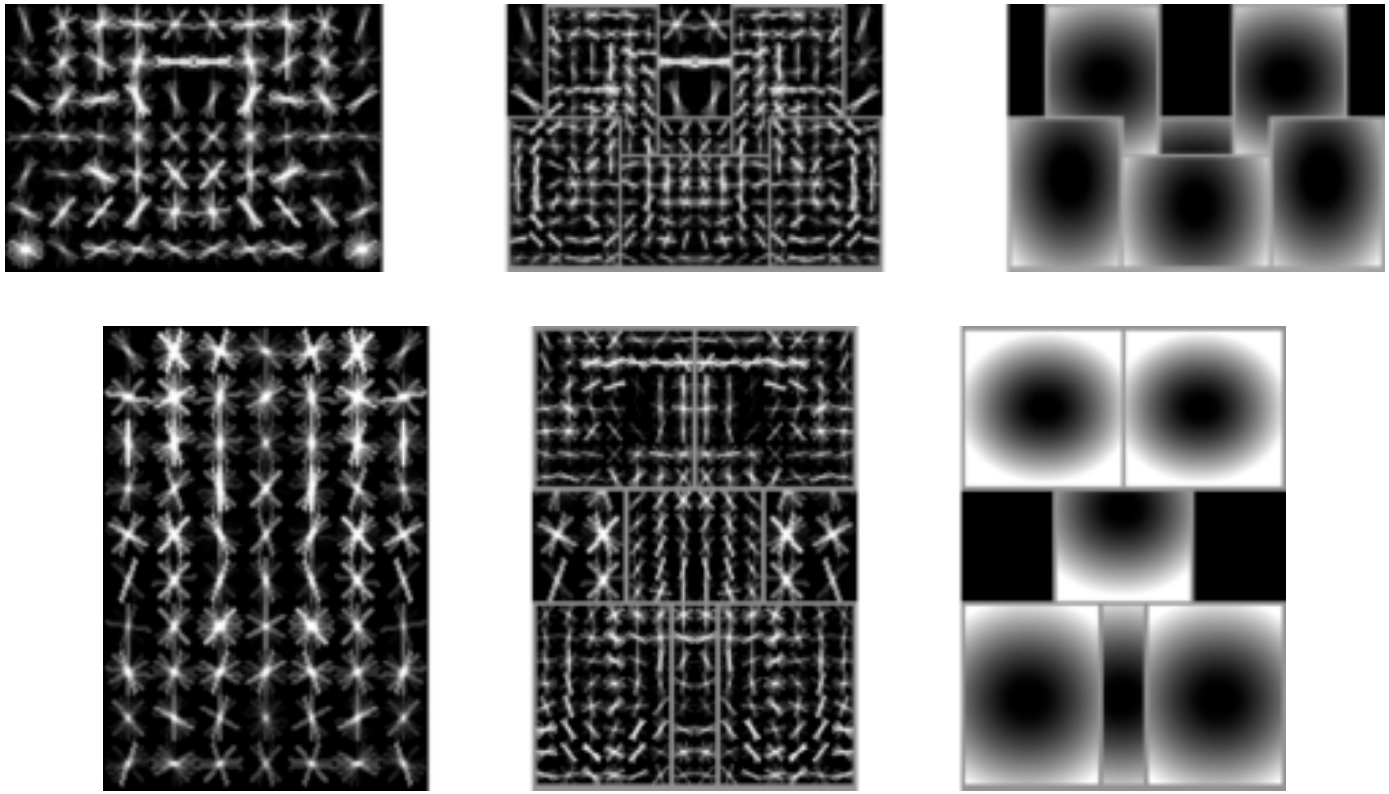
Overview of our models





- Mixture of deformable part models
- Each component has global template + deformable parts
- Fully trained from bounding boxes alone

2 component bicycle model



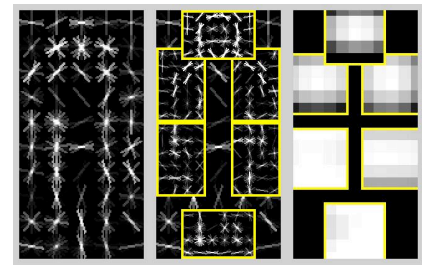
root filters
coarse resolution

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



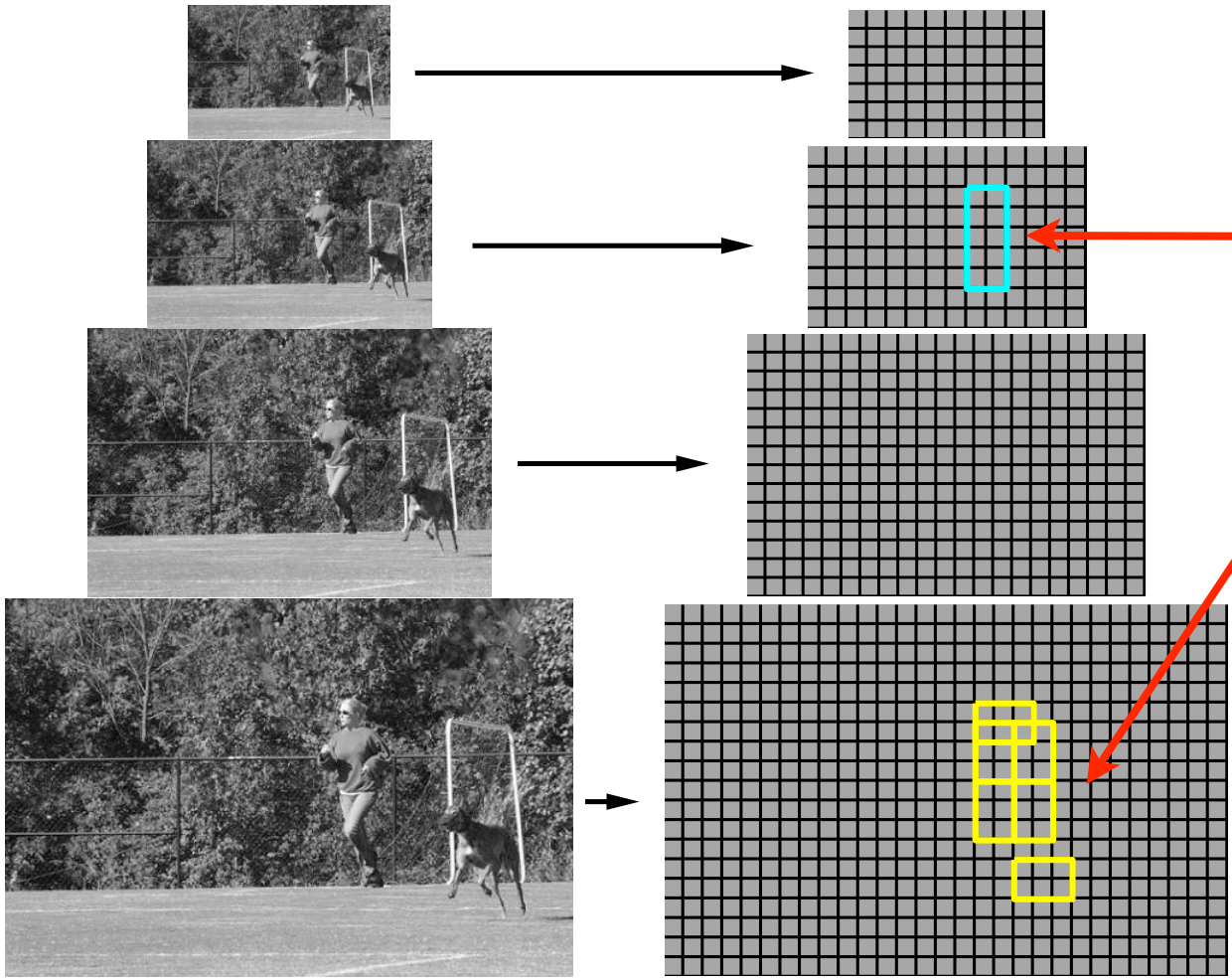


Image pyramid

HOG feature pyramid

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

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

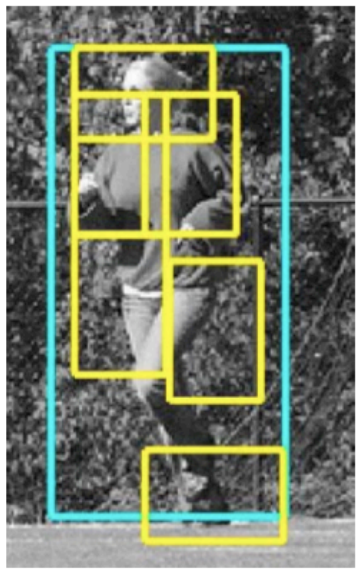
Score of a hypothesis

$$\text{score}(p_0, \ldots, p_n) = \underbrace{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}_{\text{“data term”}} - \underbrace{\sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)}_{\text{“spatial prior”}}$$

filters

displacements

deformation parameters



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

concatenation filters and deformation parameters

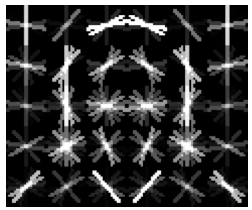
concatenation of HOG features and part displacement features

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: dynamic programming + generalized distance transforms (max-convolution)



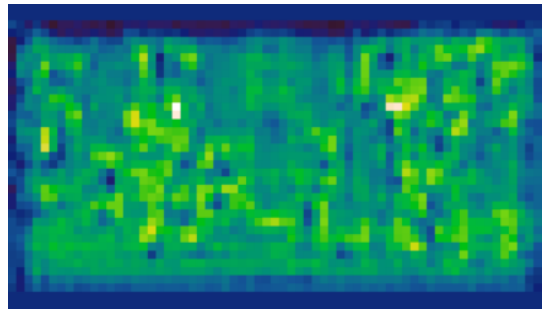
head filter

input image

Response of filter in l-th pyramid level

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

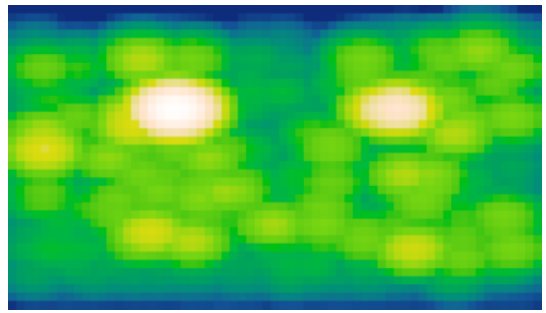
cross-correlation

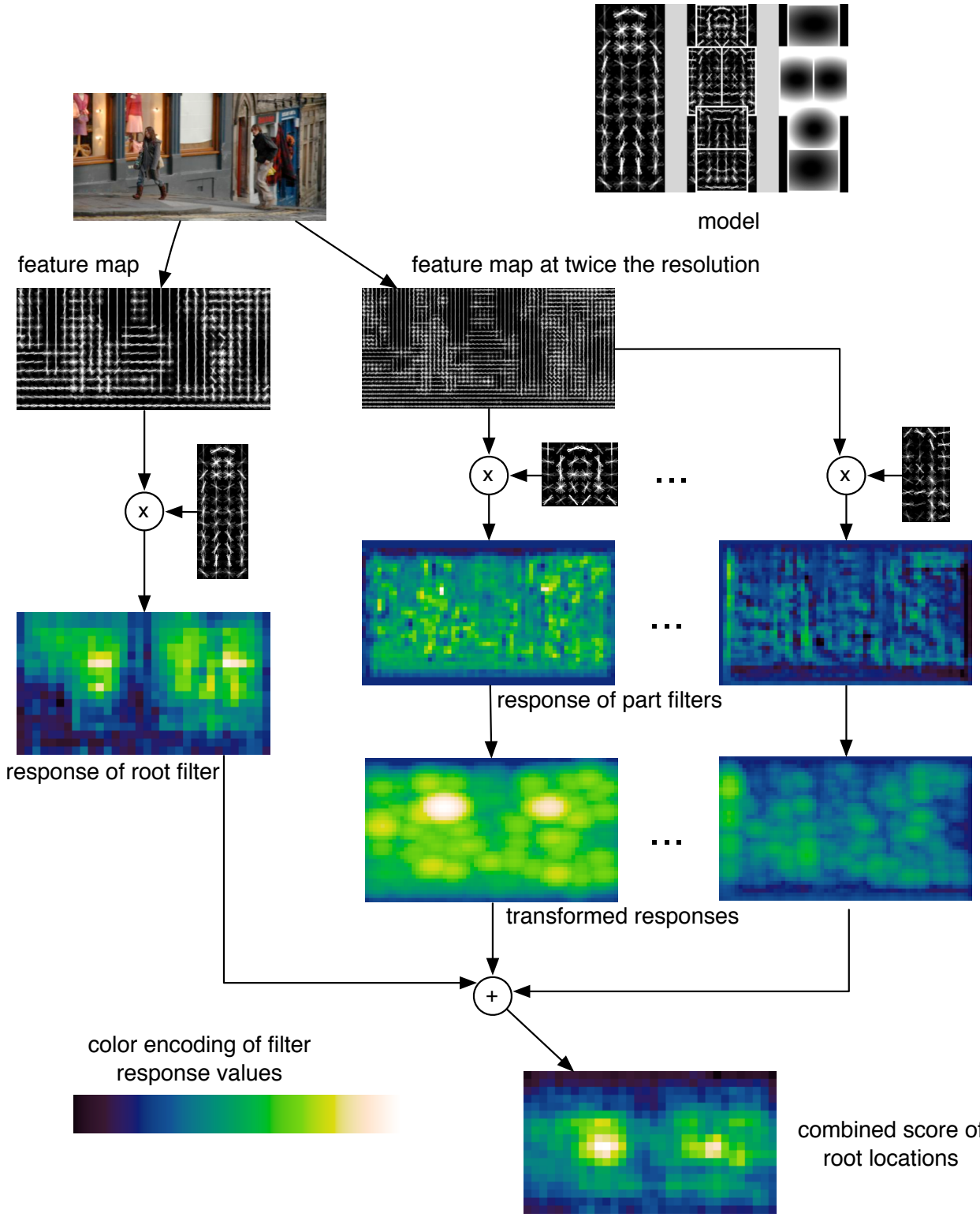


Transformed response

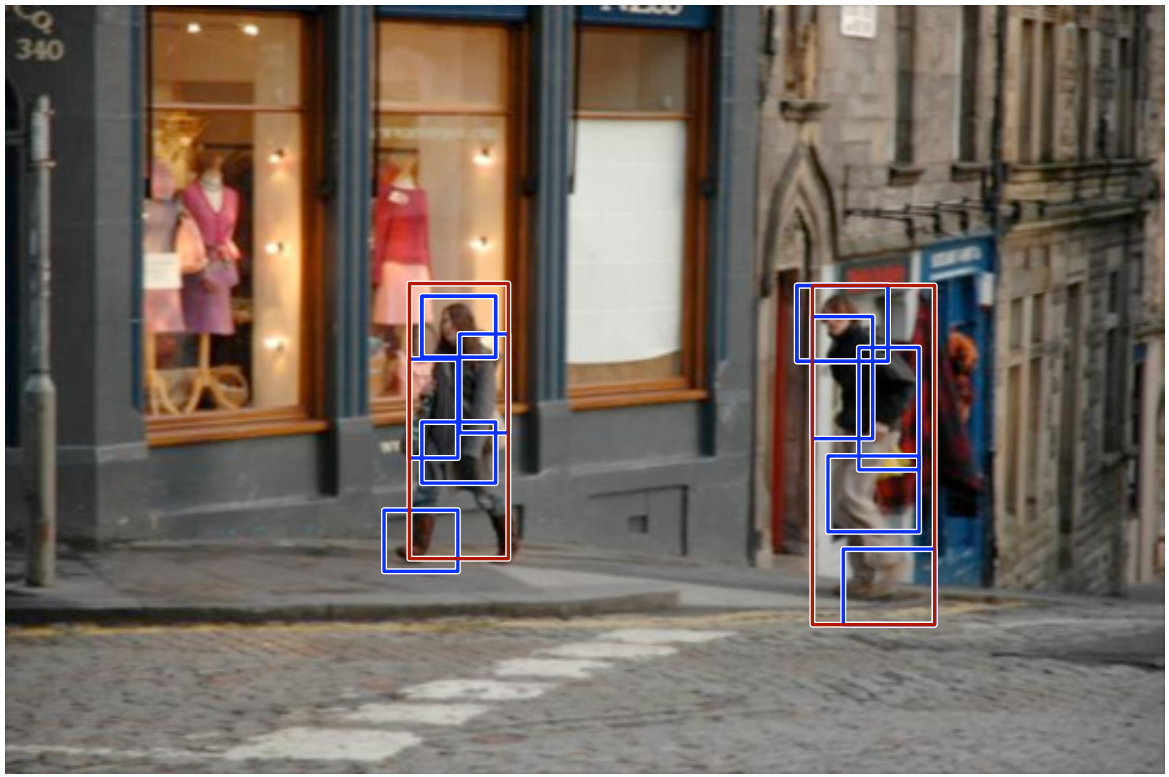
$$D_l(x, y) = \max_{dx,dy} \left(R_l(x + dx, y + dy) - d_i \cdot (dx^2, dy^2) \right)$$

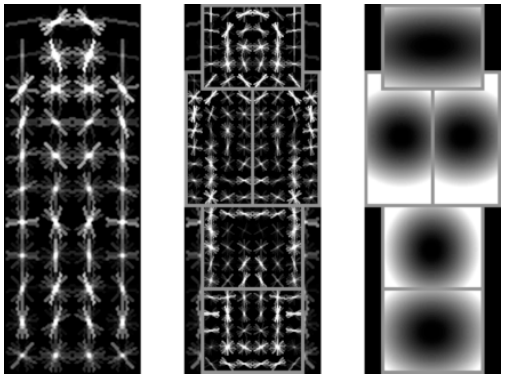
max-convolution, computed in linear time
(spreading, local max, etc)




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





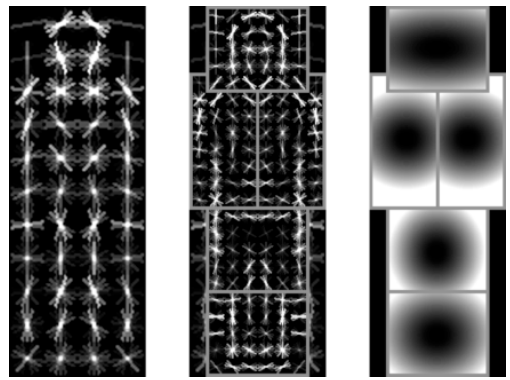
(after non-maximum suppression)

~1 second to search all scales

Training

- Training data consists of images with labeled bounding boxes.
- Need to learn the model structure, filters and deformation costs.

Training



Latent SVM (MI-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 $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle)$ $\quad y_i \in \{-1, 1\}$

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

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

- 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

Training Models

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

Background

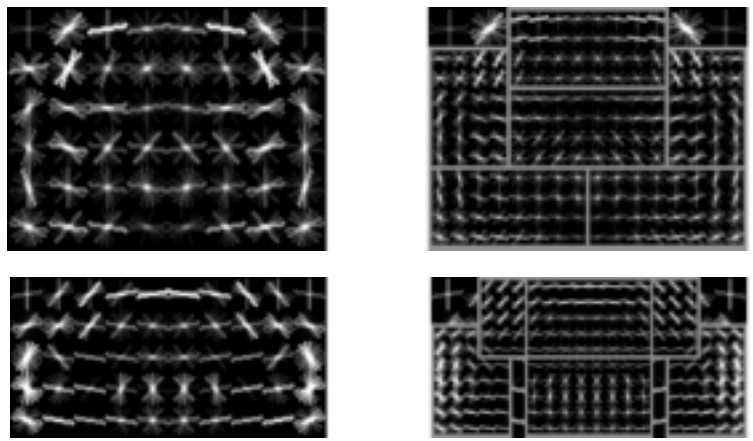
- Negative example specifies no z should have high score
- One negative example per root location in a background image
  - Huge number of negative examples
  - Consistent with requiring low false-positive rate

Training algorithm, nested iterations

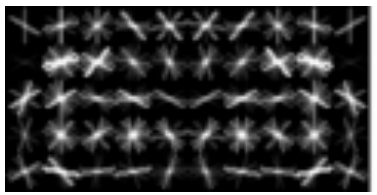
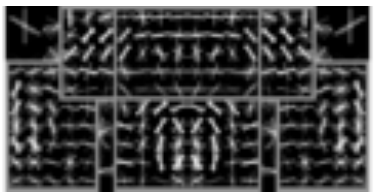
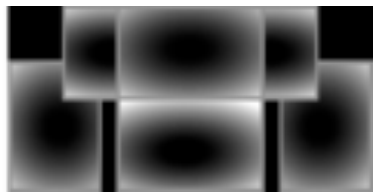
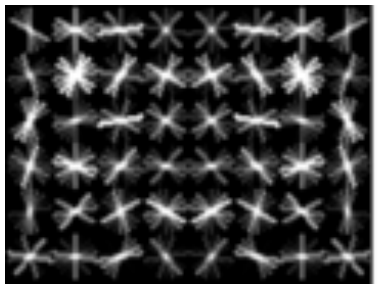
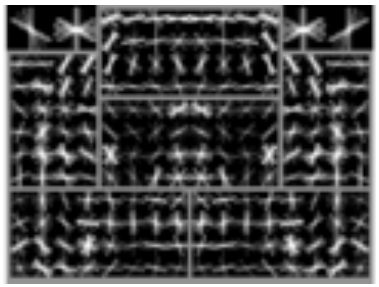
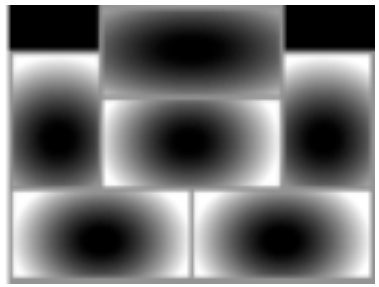
Fix “best” positive latent values for positives

- Harvest high scoring (x,z) pairs from background images
- Update model using gradient descent
- Trow away (x,z) pairs with low score

- Sequence of training rounds
  - Train root filters
  - Initialize parts from root
  - Train final model



Car model

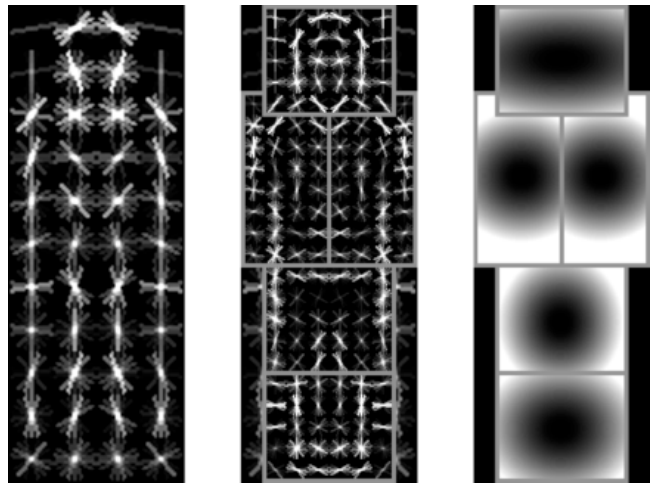







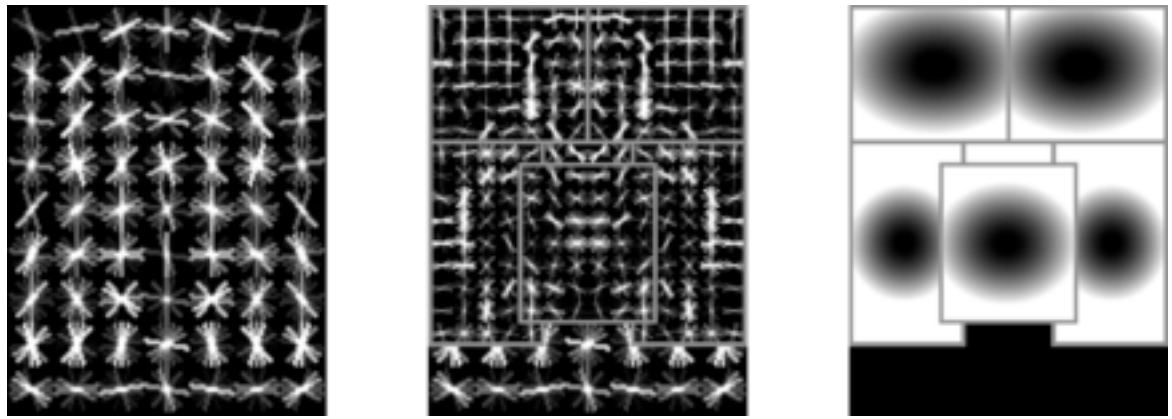
root filters
coarse resolution

part filters
finer resolution

deformation
models

Person model



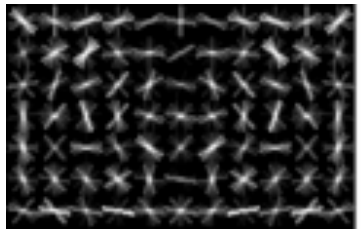
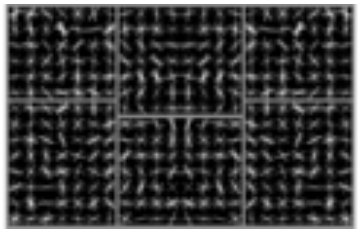
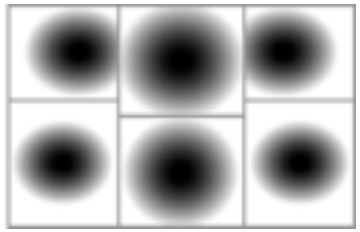
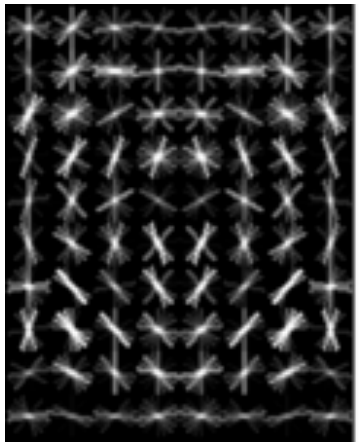
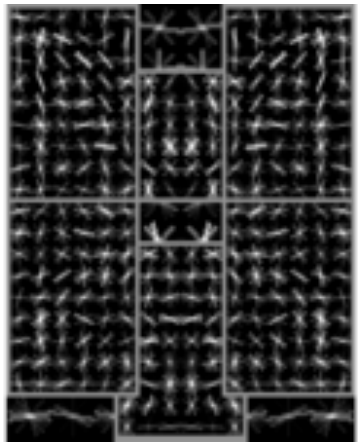
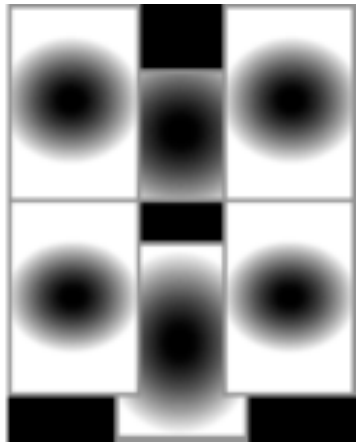


root filters
coarse resolution

part filters
finer resolution

deformation
models

Cat model

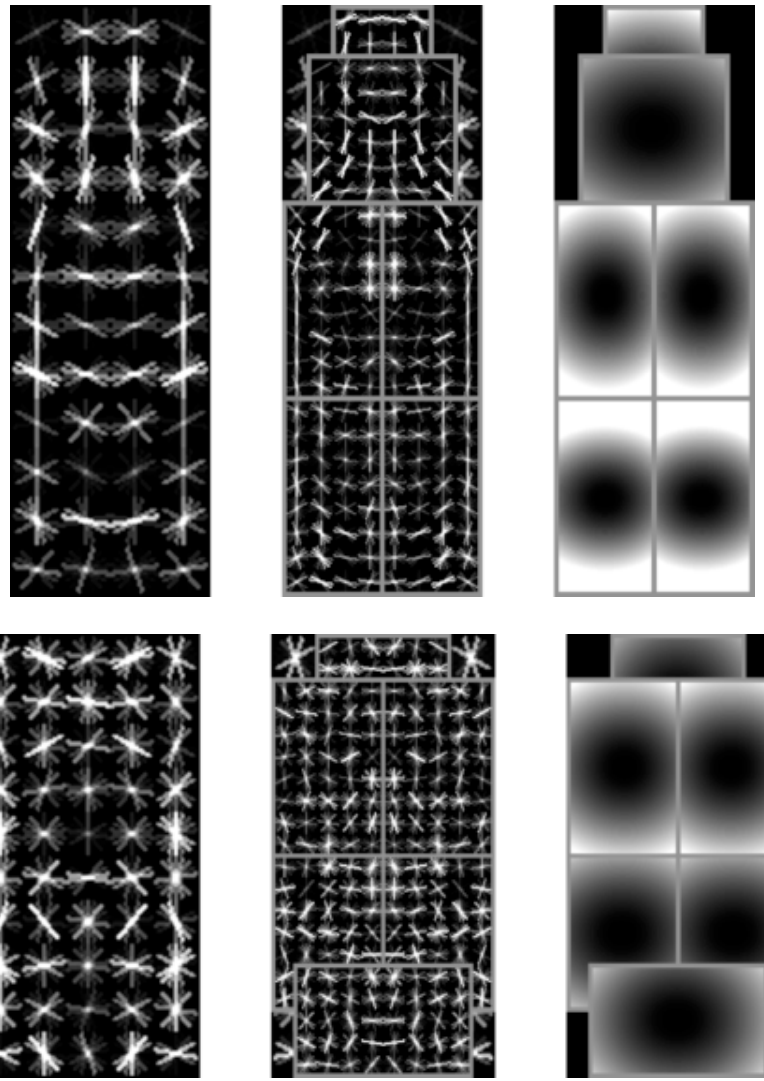







root filters
coarse resolution

part filters
finer resolution

deformation
models

Bottle model



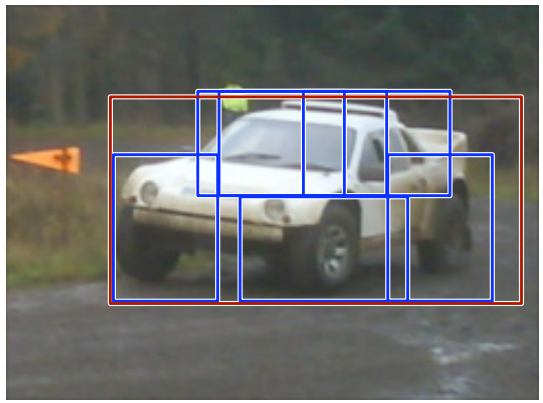
root filters
coarse resolution

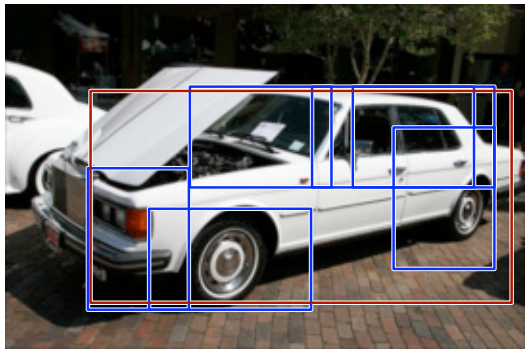
part filters
finer resolution

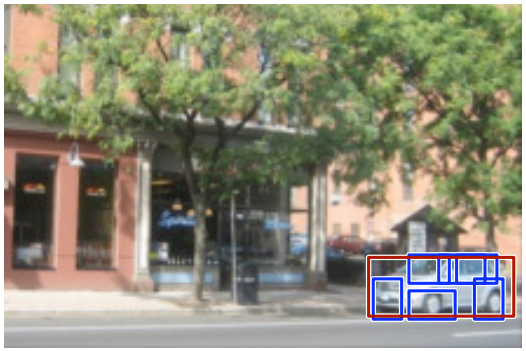
deformation
models

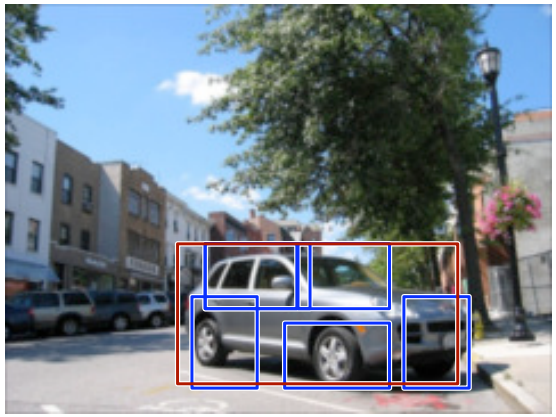
Car detections

high scoring true positives

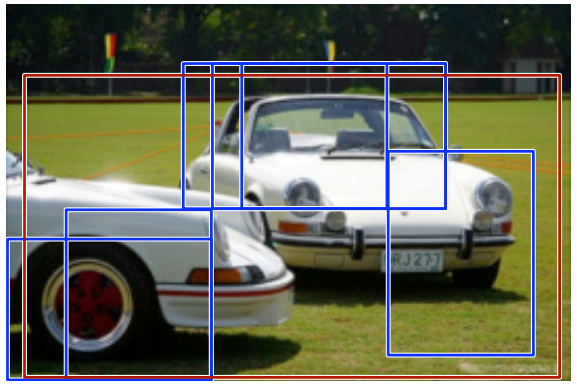


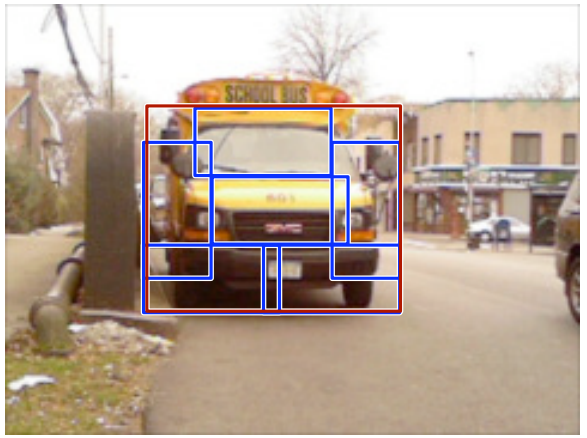






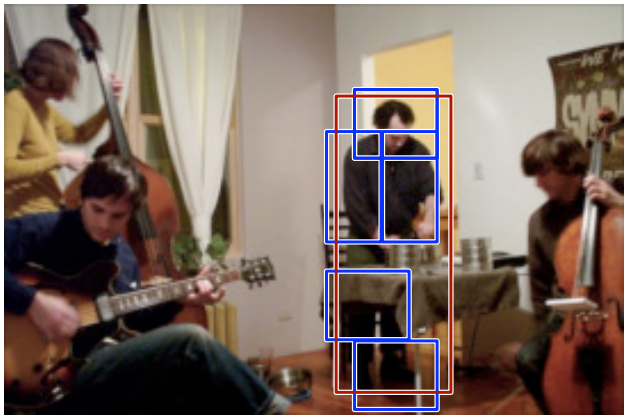
high scoring false positives

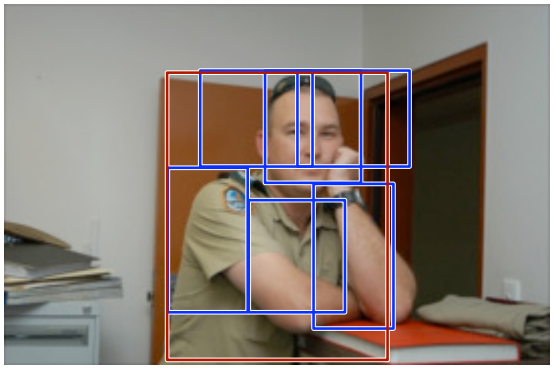


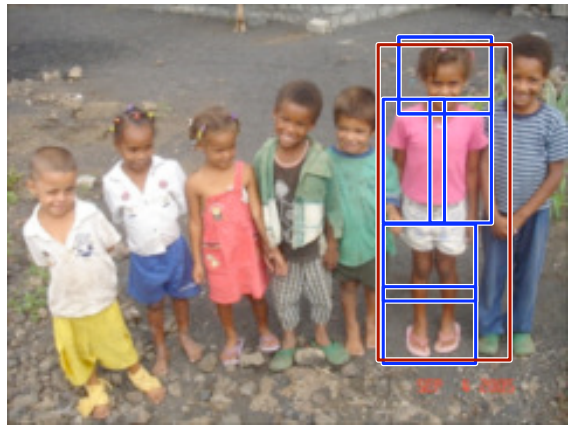


Person detections

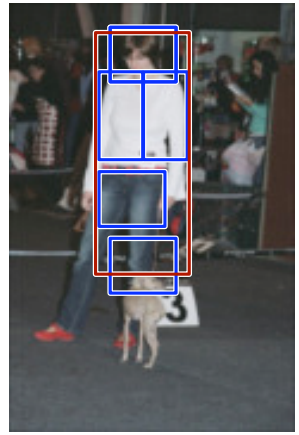
high scoring true positives





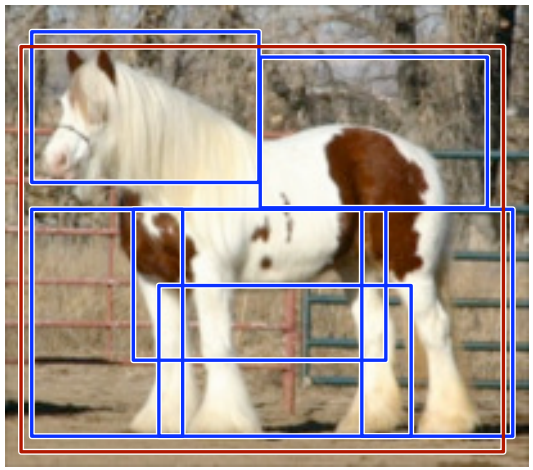


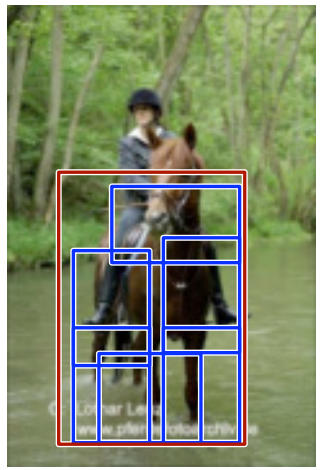
high scoring false positives
(not enough overlap)

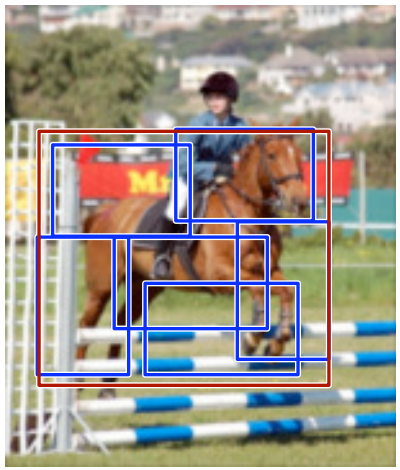


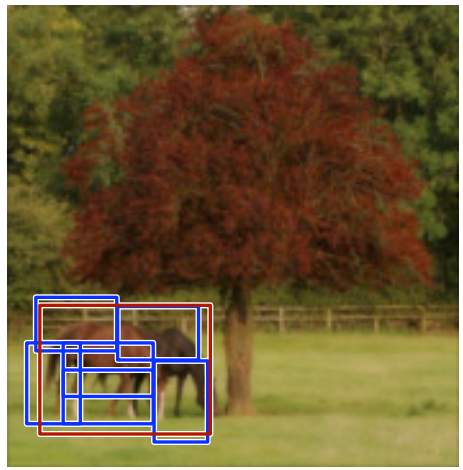
Horse detections

high scoring true positives

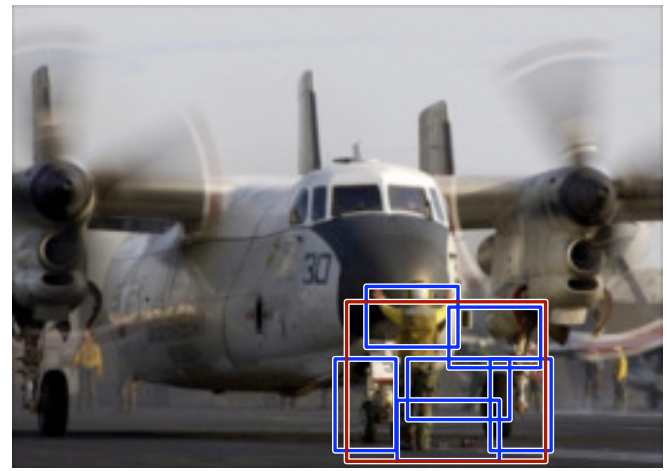


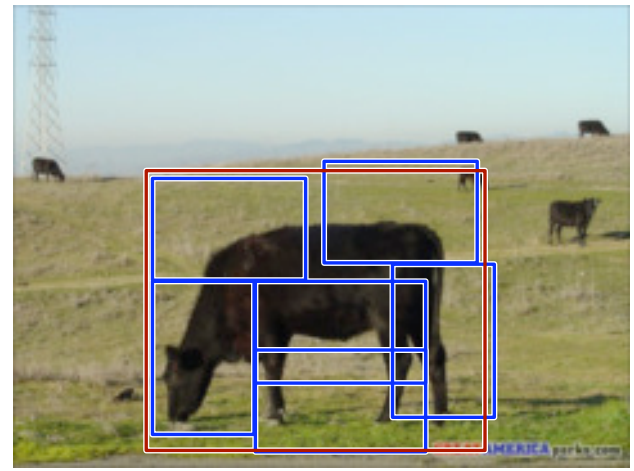






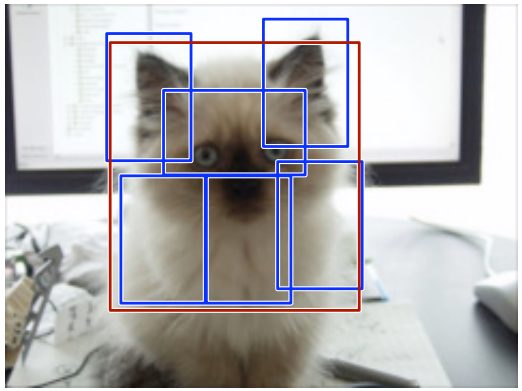
high scoring false positives

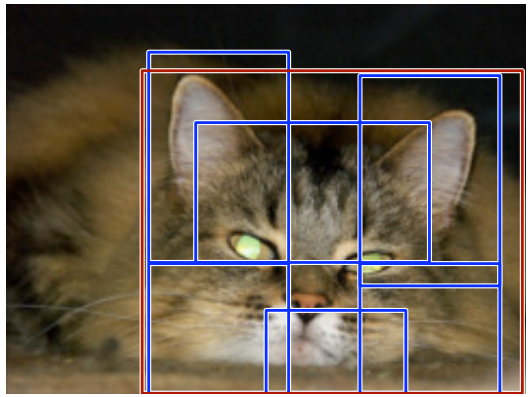


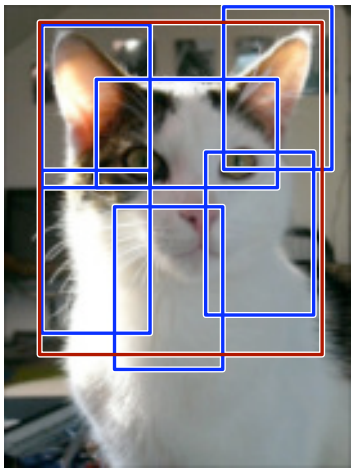


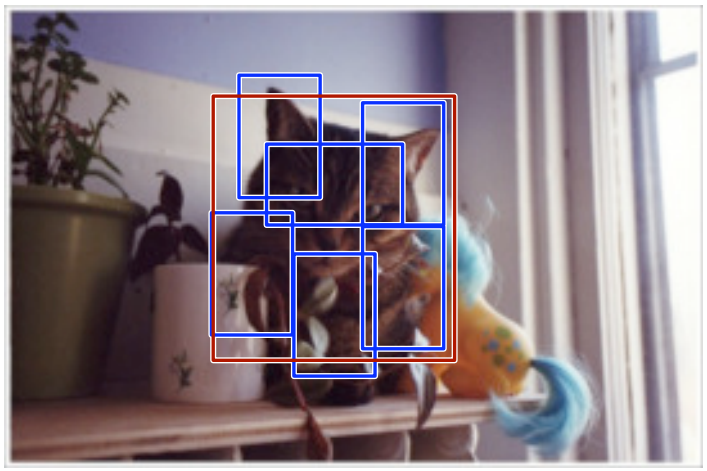
Cat detections

high scoring true positives

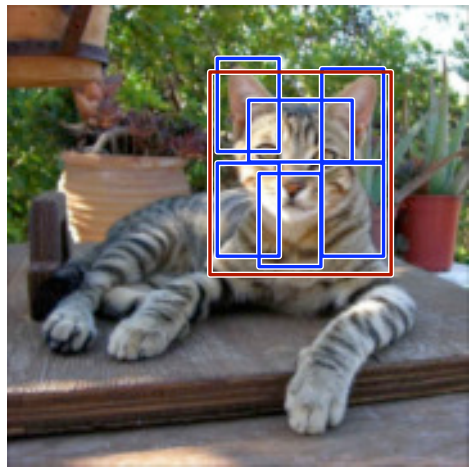


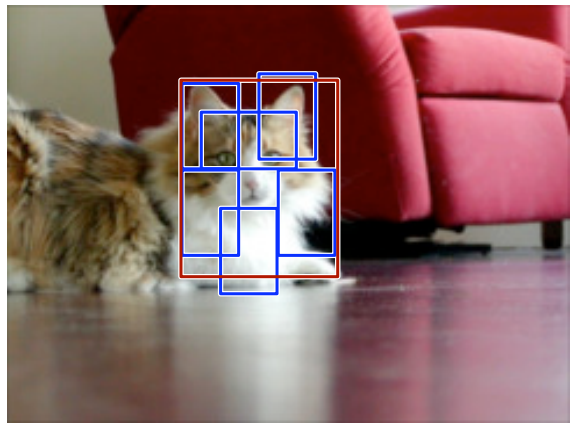






high scoring false positives
(not enough overlap)

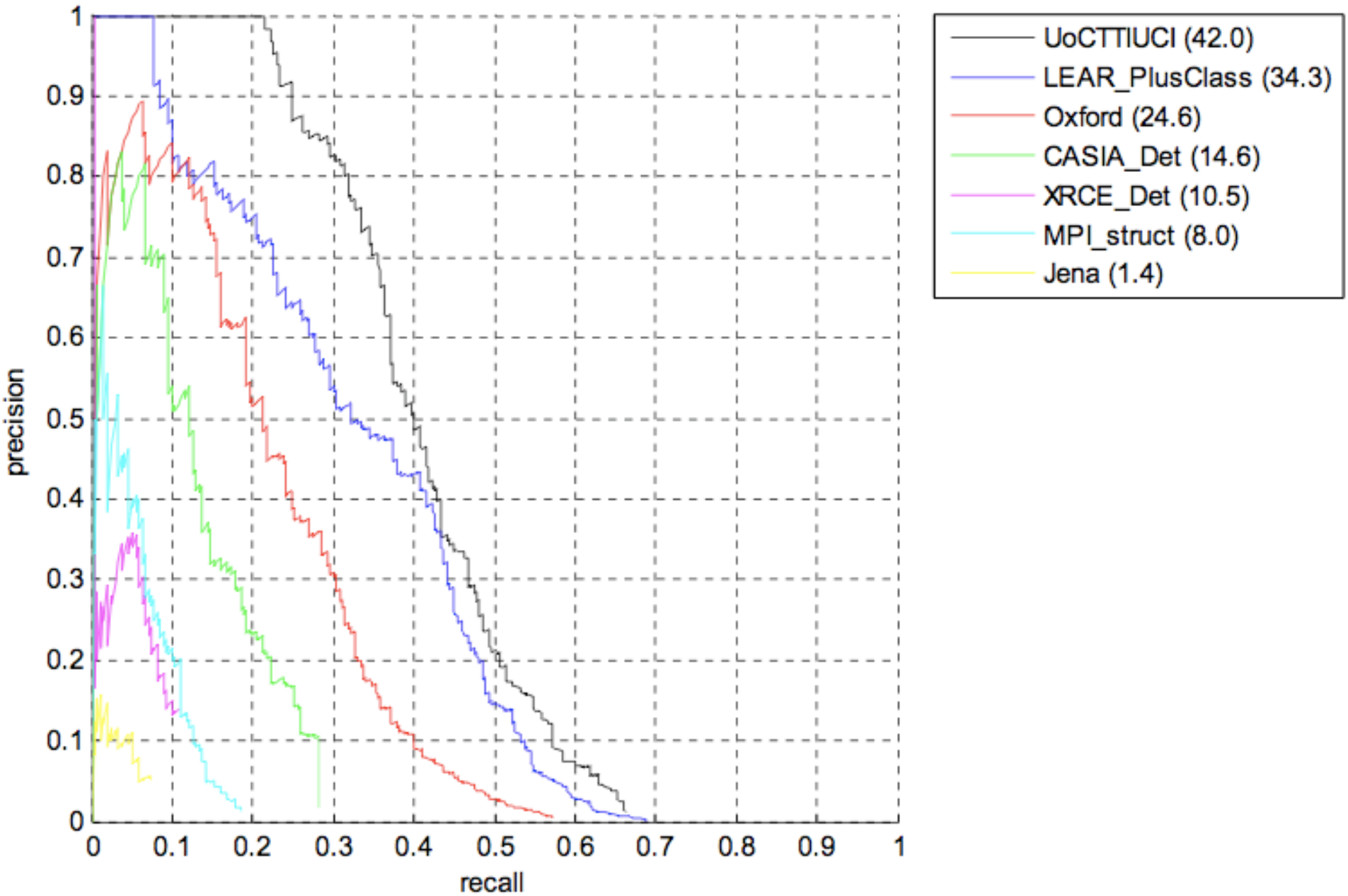




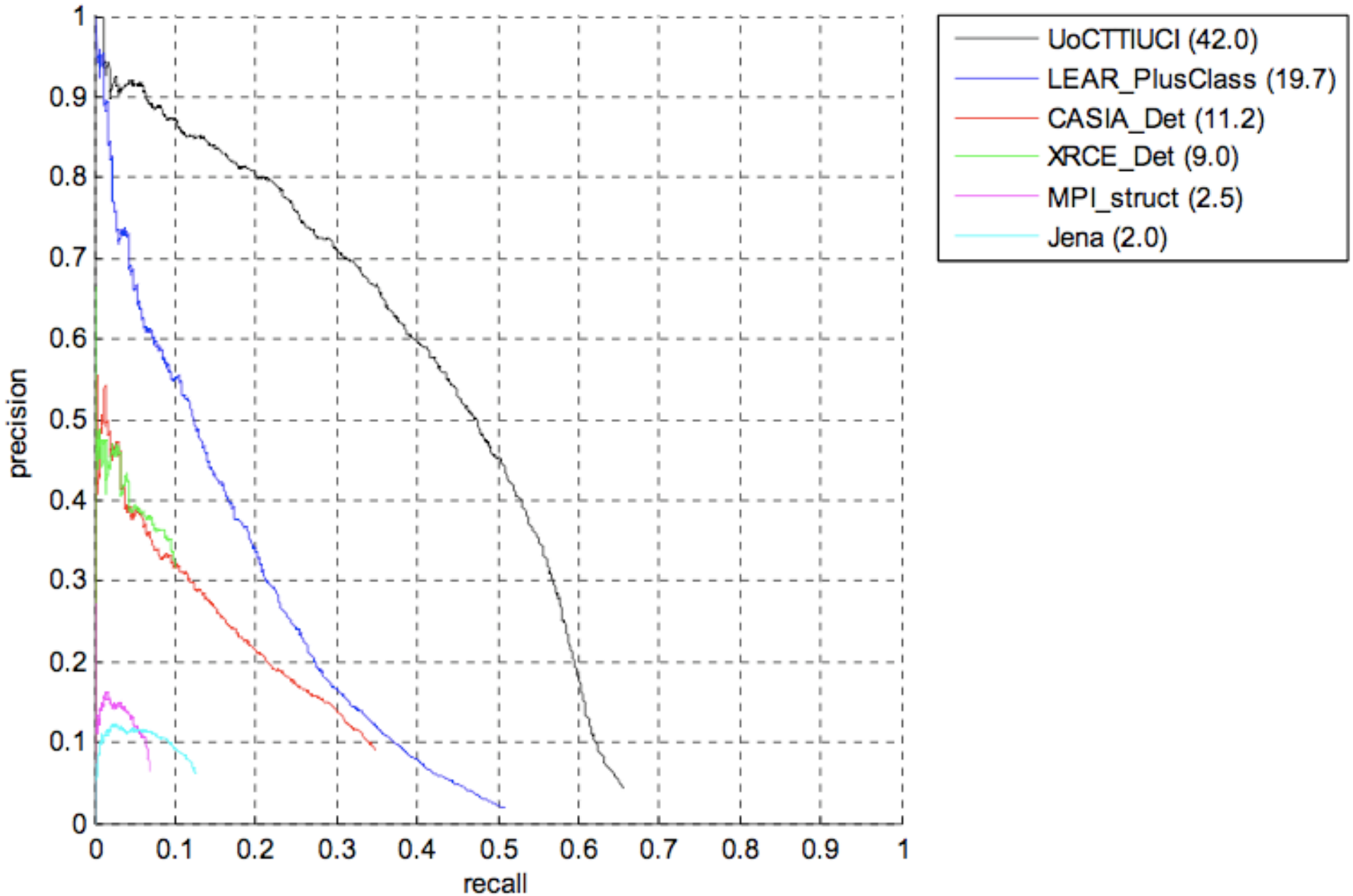
Quantitative results

- 7 systems competed in the 2008 challenge
- Out of 20 classes we got:
  - First place in 7 classes
  - Second place in 8 classes
- Some statistics:
  - It takes ~2 seconds to evaluate a model in one image
  - It takes ~4 hours to train a model
  - MUCH faster than most systems.

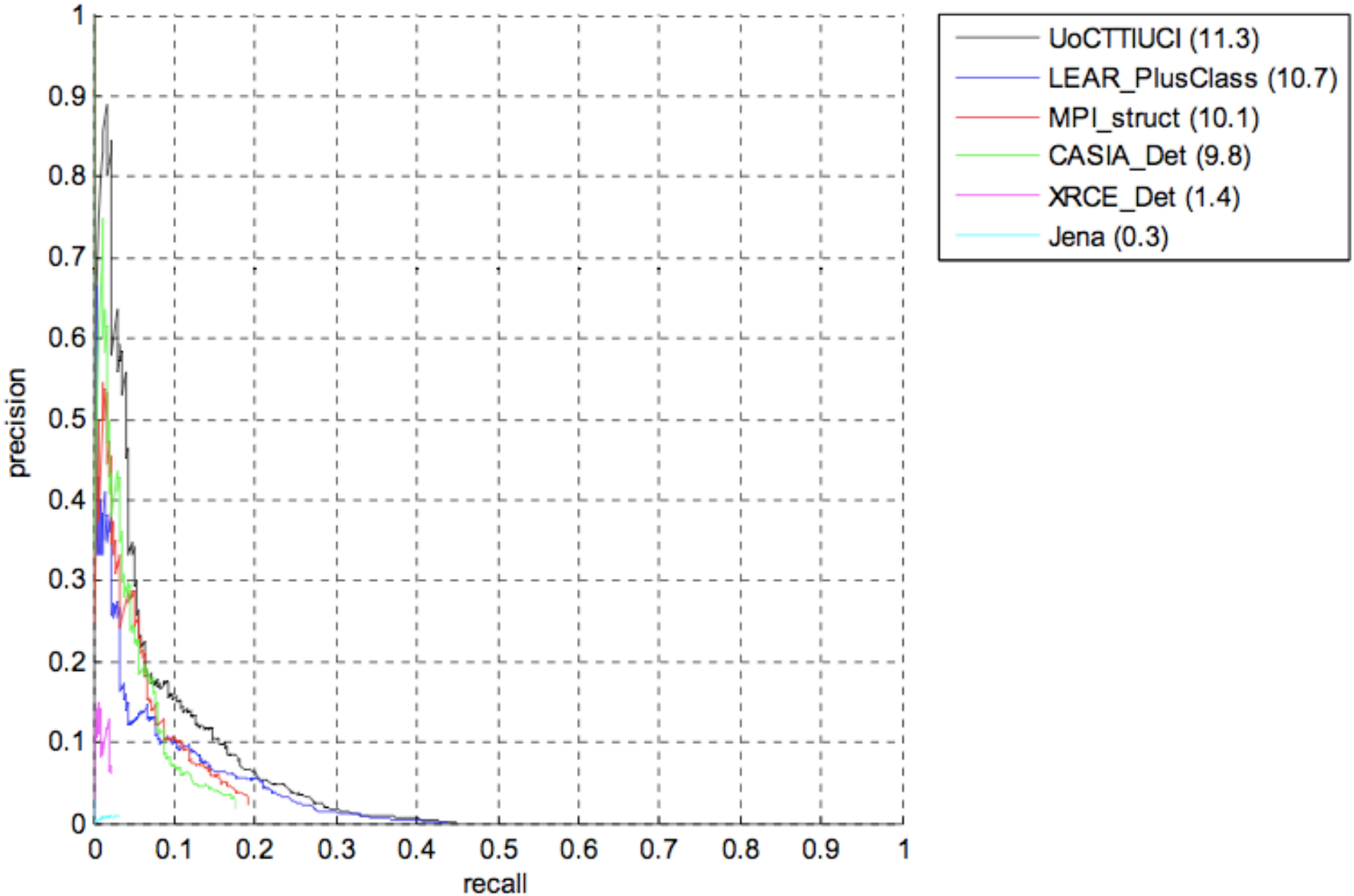
Precision/Recall results on Bicycles 2008



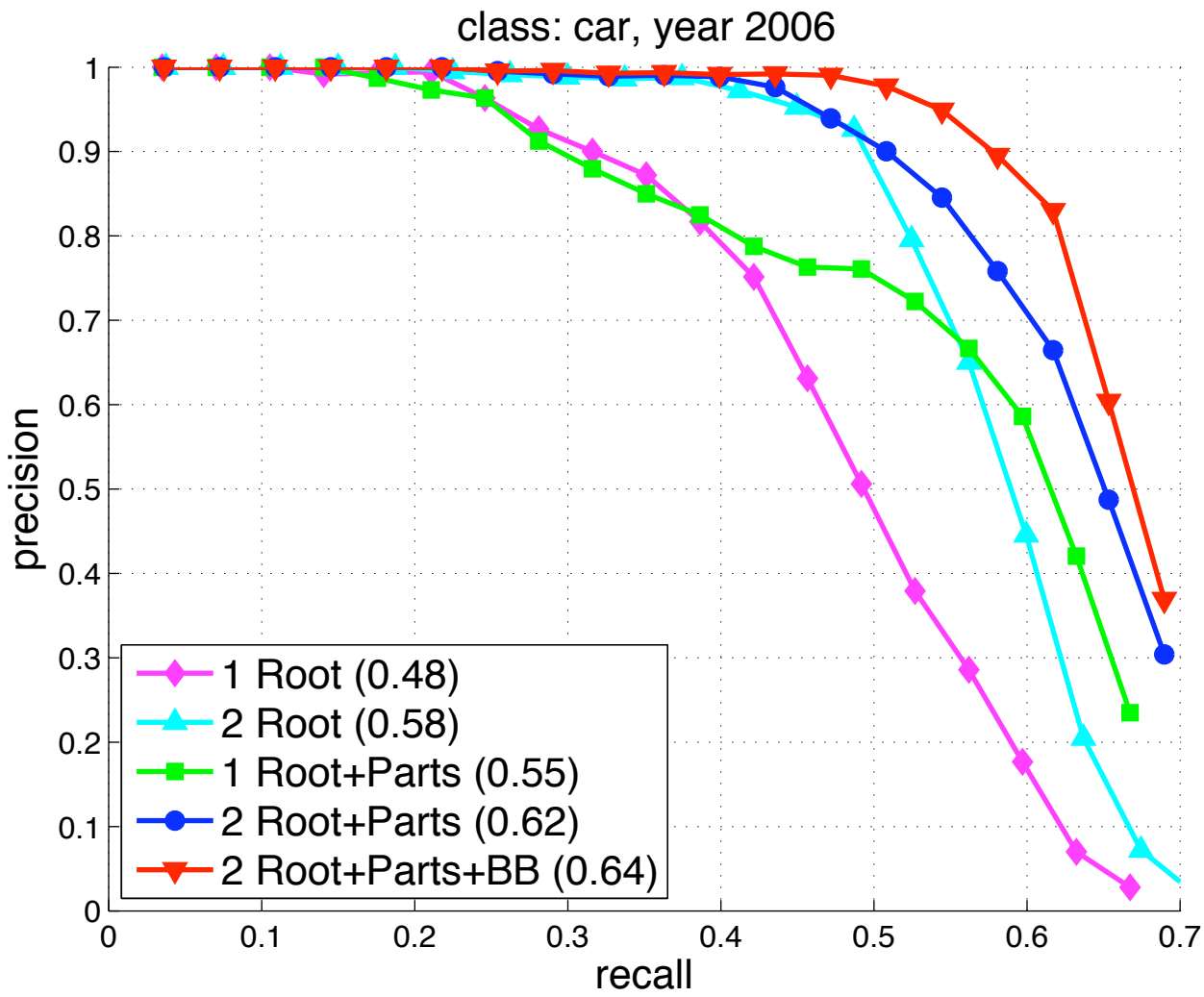
Precision/Recall results on Person 2008



Precision/Recall results on Bird 2008



Comparison of Car models on 2006 data



Summary

- Deformable models for object detection
  - Fast matching algorithms
  - Learning from weakly-labeled data
  - Leads to state-of-the-art results in PASCAL challenge
- Future work:
  - Hierarchical models
  - Visual grammars
  - AO* search (coarse-to-fine)