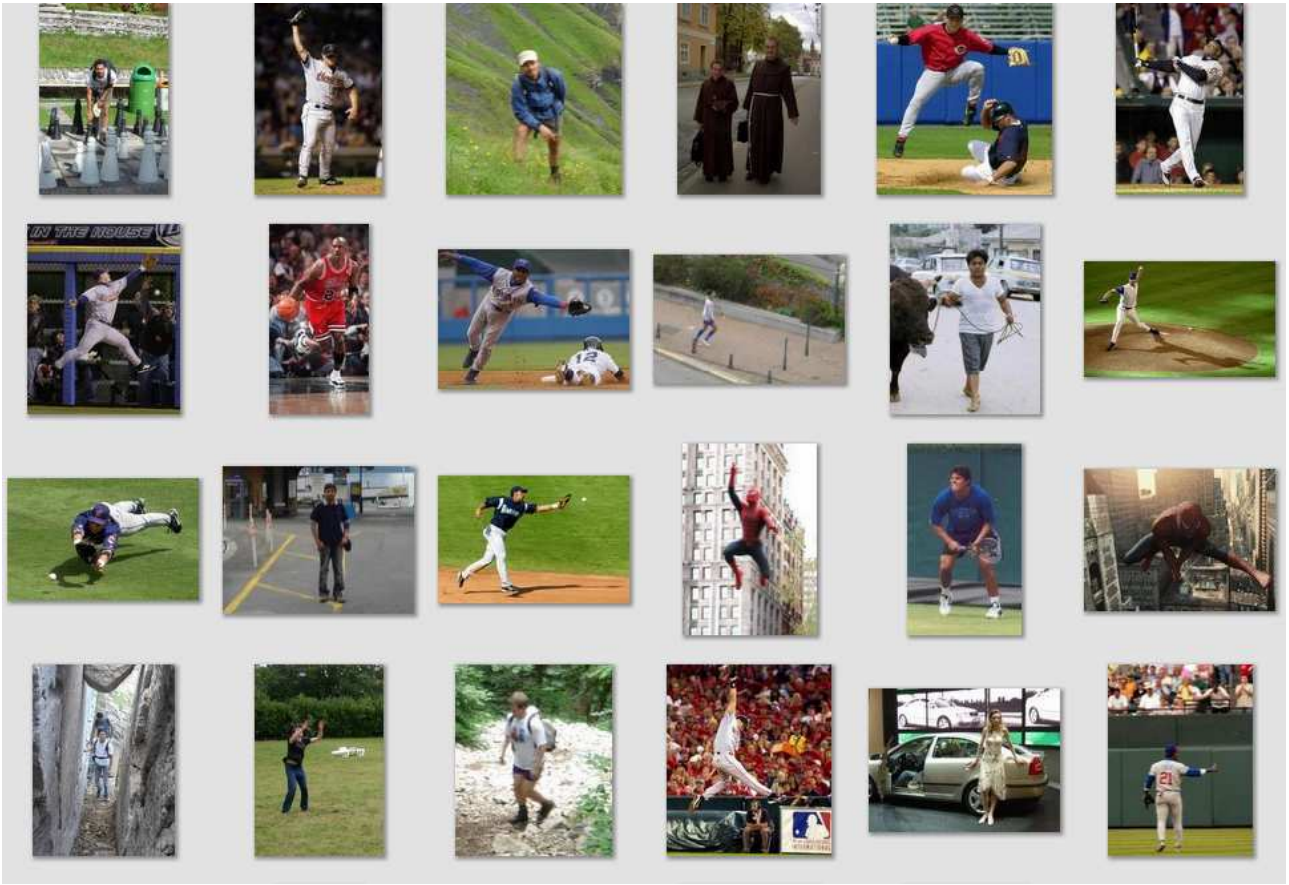
A Discriminatively Trained, Multiscale, Deformable Part Model

Pedro Felzenszwalb, David McAllester
Deva Ramanan

Presented by: David Arnon

Slides credit: Pedro F. Felzenszwalb, Duan Tran

Deformable objects



Images from D. Ramanan's dataset

Images from Caltech-256

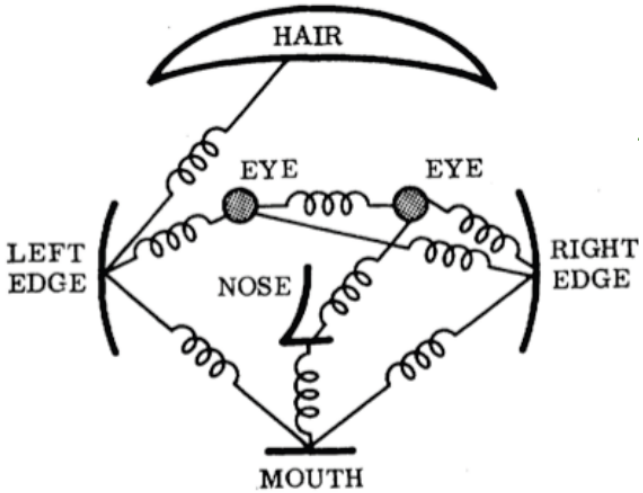
Challenges

- High intra-class variations
- Deformable

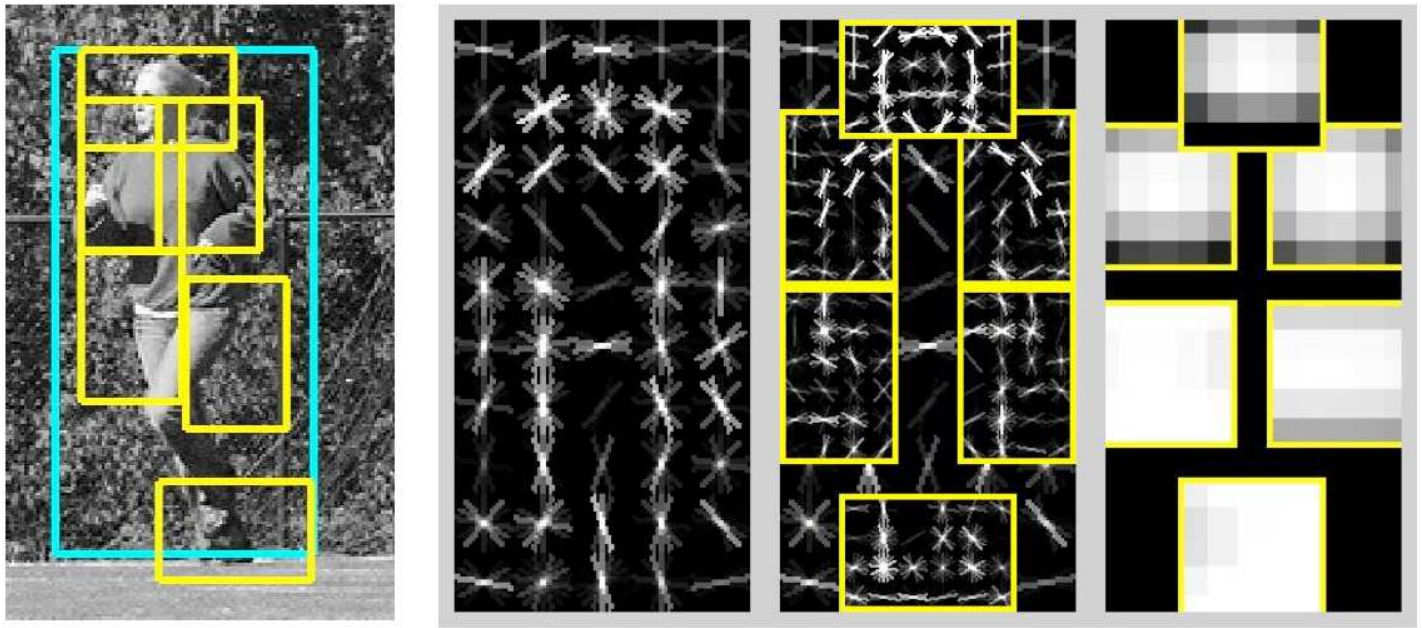
- Therefore…
  - Part-based model might be a better choice !

Part-based representation

- Objects are decomposed into parts and spatial relations among parts
- E.g. Face model by Fischler and Elschlager ‘73

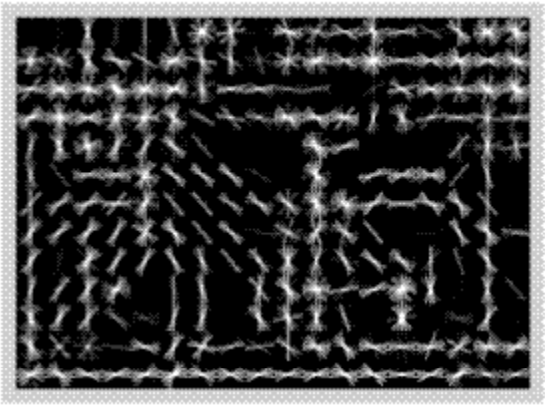


Overview



detection root filter part filters deformation models

HOG features



- Dalal & Triggs:
  - Histogram gradient orientations in 8x8 pixel blocks (9 bins)
  - Normalize with respect to 4 different neighborhoods and truncate
  - 9 orientations * 4 normalizations = 36 features per block
- **Repeat for all levels of an image pyramid at multiple scales**

Filters

- Filters are rectangular templates defining weights for features

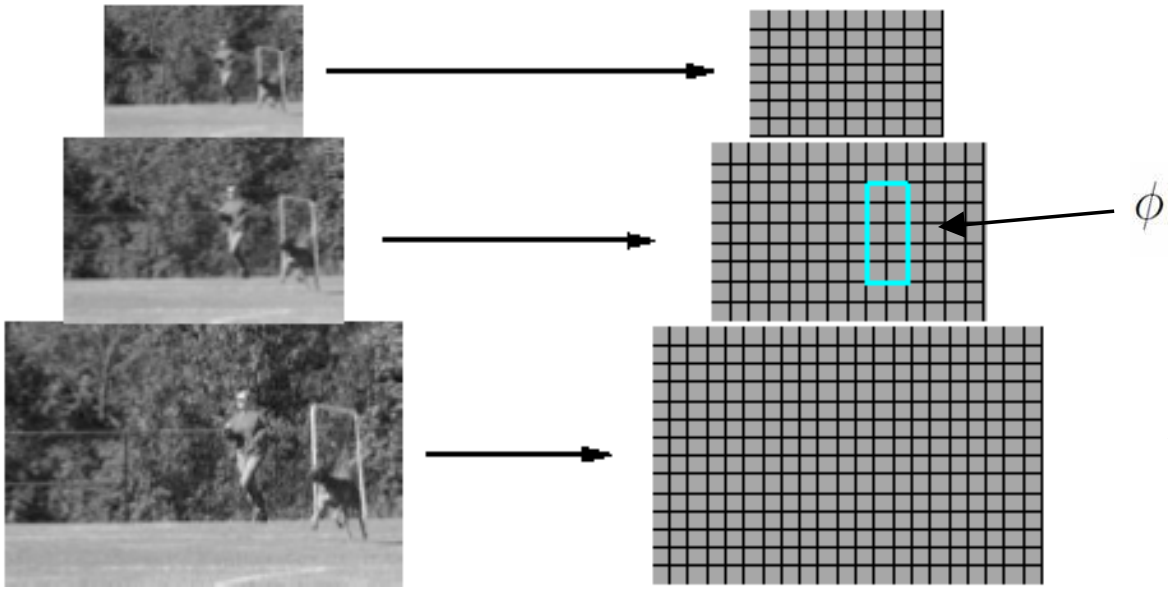


Image pyramid

HOG pyramid $\boldsymbol{H}$

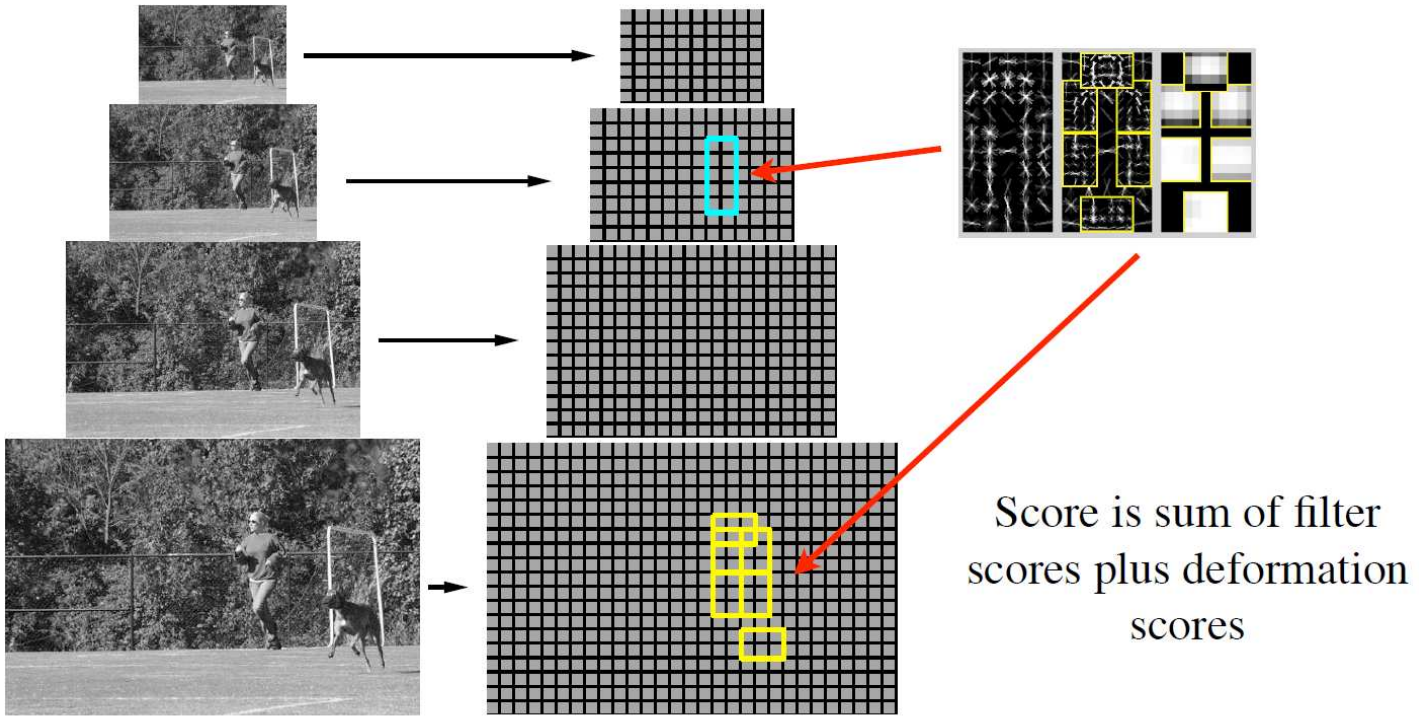
$\boldsymbol{F}$

Application of filter $\boldsymbol{F}$:

$$F \cdot \phi(H, p_i)$$

Part Filters

- Coarser details for the root filter (whole object) and finer details for part filters



Model

- A model consists of a root filter F_0 and part model $(P_1,\ldots,P_6)$, $P_i = (F_i, v_i, s_i, a_i, b_i)$
  - F_i : filter
  - (v_i,s_i) : location and size of part positions box
  - (a_i,b_i) : deformation cost parameters

Placement

- A placement of a model in a HOG pyramid:

$$z = (p_0, p_1, \ldots, p_6)$$

  - Part location should be in double resolution level
  - Given a model, part location must a legal one in that model

Placement Score

$$\sum_{i=0}^{n} F_i \cdot \phi(H, p_i) + \sum_{i=1}^{n} a_i \cdot (\tilde{x}_i, \tilde{y}_i) + b_i \cdot (\tilde{x}_i^2, \tilde{y}_i^2)$$

data term

deformation cost

- $\tilde{x}_i, \tilde{y}_i \in$ [-1,1] are the relative placement of the part inside it's box
- Using dynamic programming to find best placement

Placement Score

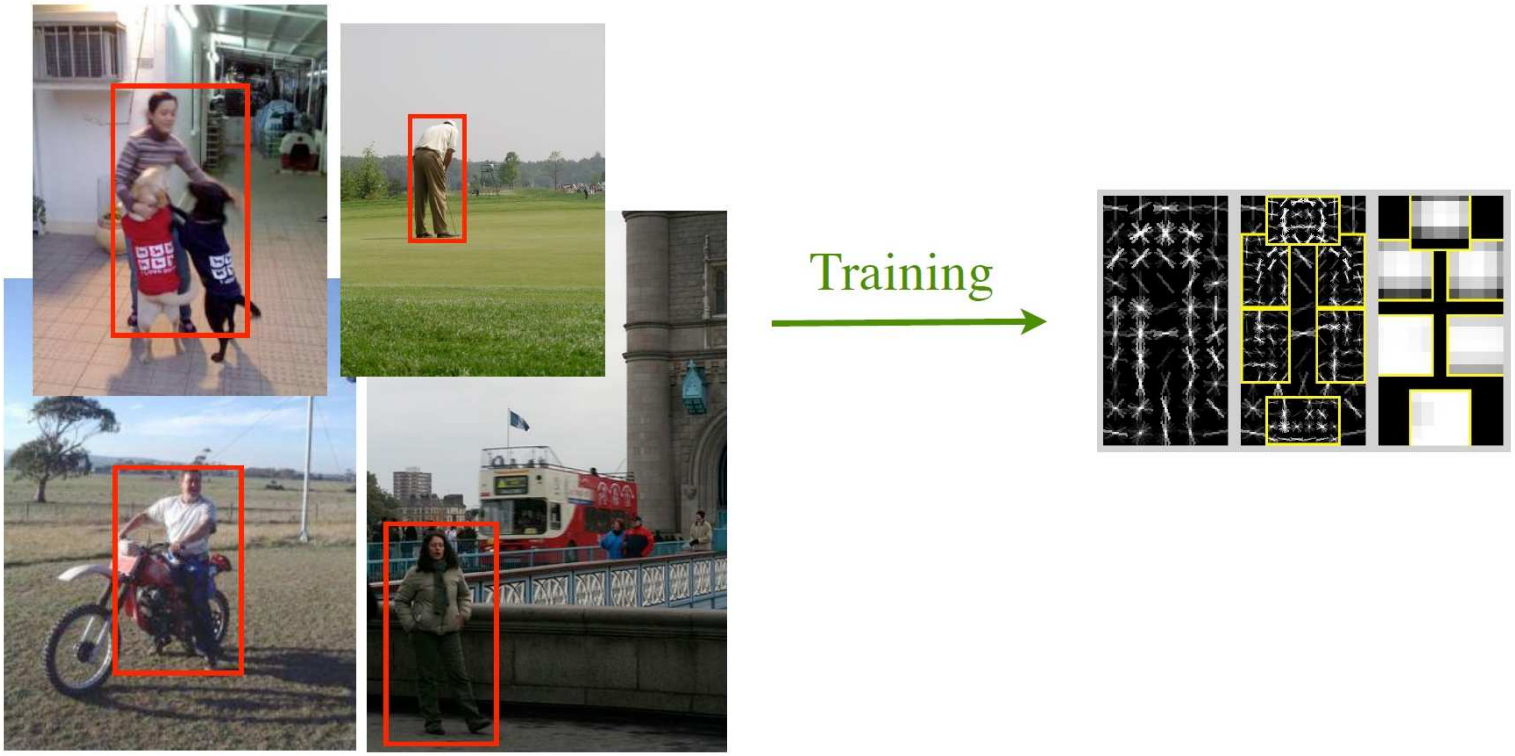
- The score can also be expressed as a dot product $\beta \cdot \Phi(x, z)$ where:

$$\beta = (F_0, \ldots, F_n, a_1, b_1 \ldots, a_n, b_n)$$

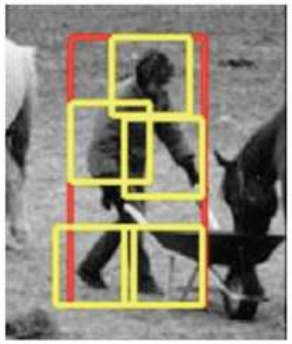
$$\Phi(x, z) = (\phi(H, p_0), \phi(H, p_1), \ldots \phi(H, p_n),$$
$$\tilde{x}_1, \tilde{y}_1, \tilde{x}_1^2, \tilde{y}_1^2, \ldots, \tilde{x}_n, \tilde{y}_n, \tilde{x}_n^2, \tilde{y}_n^2,)$$

Learning

- Training data consists of images with labeled bounding boxes
- We aim to learn the model structure
  - Filters, deformation costs



Latent SVM



- β : a model
- x : an image or detection window
- z : filter placements (Latent Variables)

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

concatenation of filters and deformation parameters

concatenation of features and part displacements

Latent SVM

- Find optimal β:

$$\beta^*(D) = \underset{\beta}{\operatorname{argmin}}\, \lambda||\beta||^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

- Problem: expression is not convex

Latent SVM : Try I

- Solve by coordinate descent:

1. Initialize model β by some heuristic
2. Given β, find the latent variables
   - calculating optimal positioning for each example
     $z_i = \text{argmax}_{z \in Z(x_i)} \beta \cdot \Phi(x, z)$.
3. Given a fixed z, solve the Linear SVM to find a new model β

Latent SVM: Semi Convexity

- Many examples (mostly negative ones) make the 2nd stage expensive

- Observation: Target function is convex for <u>negative examples</u> since f_β is convex in β

Latent SVM : Try II

- Solve by coordinate descent:

1. Initialize model β by some heuristic
2. Given β, find the latent variables
   - calculating optimal positioning for positive examples
3. Given a fixed z for positive examples, solve the convex optimization problem for a new model β

Model Initialization

- Root Filter
  - Select root filter size
    - Common ratio
    - 80%-tile size
  - Train root filter by linear SVM on stretched unoccluded examples.
  - Find optimal positioning on all examples and retrain.

Model Initialization

- Part Filters
  - Select filter size
    - 6*area = 80% root area
  - sequentially choose area having high positive energy in the root filter
  - Initialize filter by interpolating matching root filter to double precision
- Deformation:

  a = (0,0) , b = (-1,-1)

Data Mining Hard Negatives

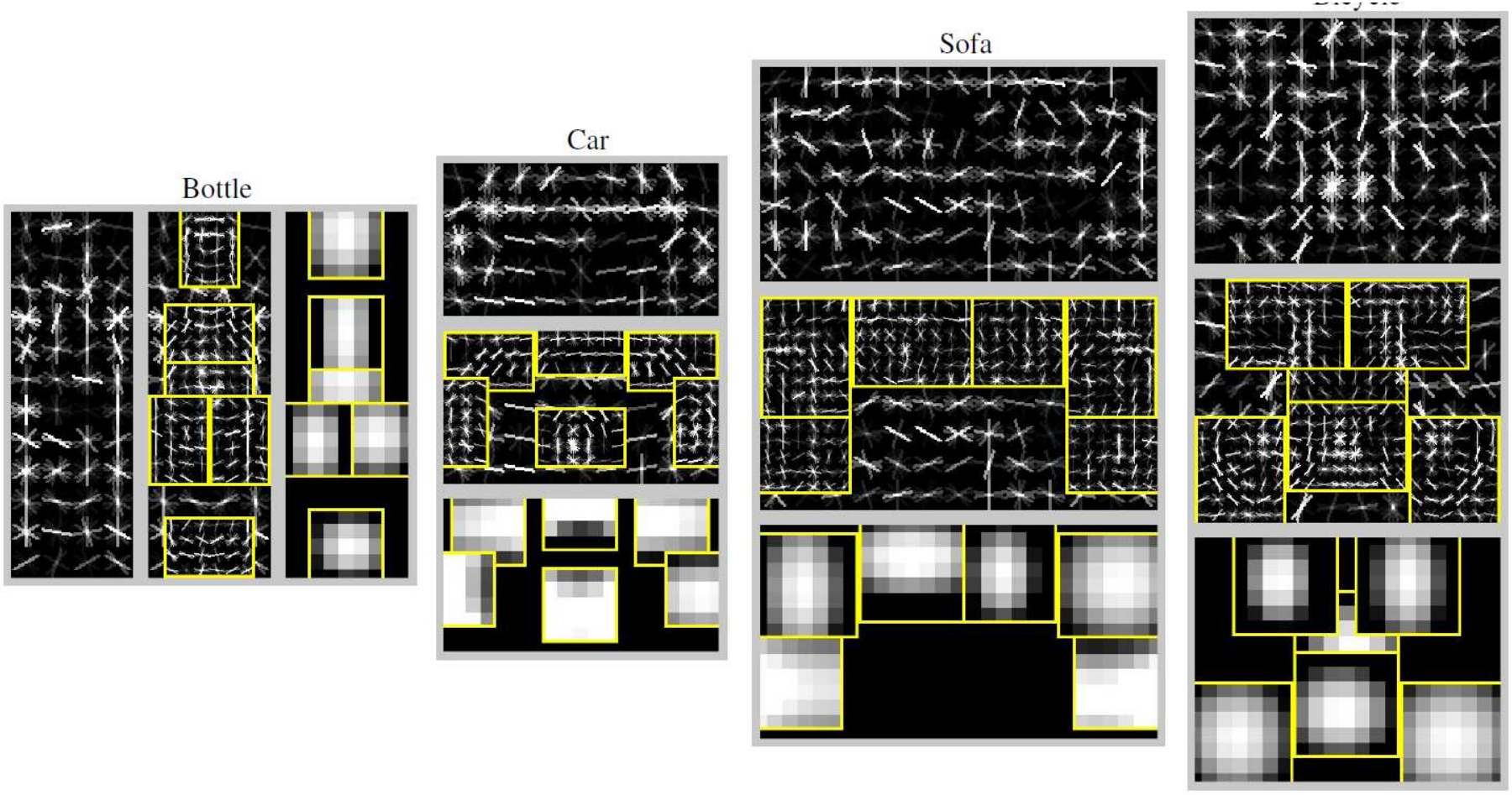
- Hard examples:

$$M(\beta, D) = \{\langle x, y\rangle \in D \mid y f_\beta(x) \leq 1\}$$

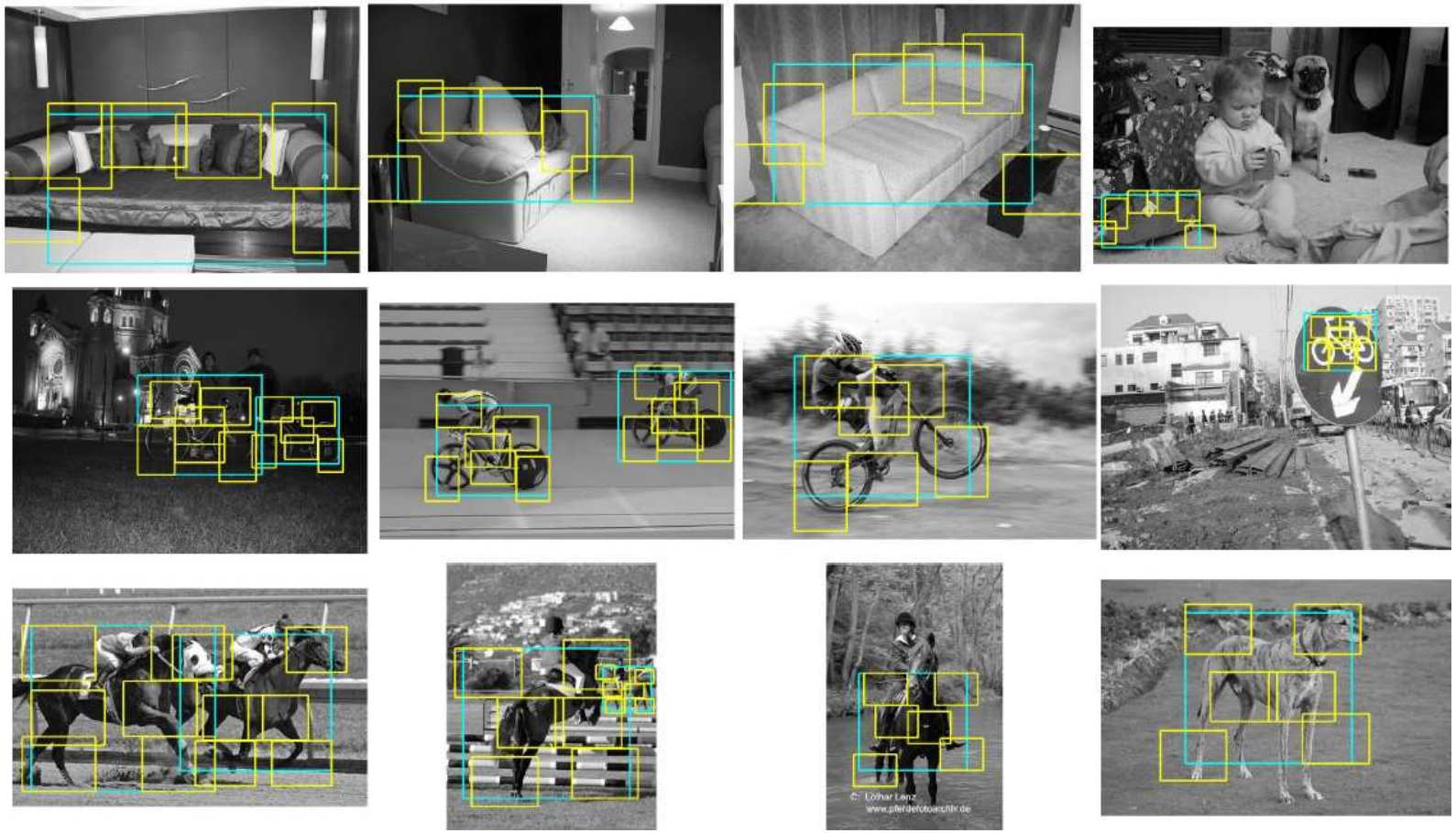
- Initialize $\boldsymbol{C}$ with positive examples and random negative examples.
- Iterate:

1. Let $\beta := \beta^*(C)$.
2. Shrink C by letting $C := M(\beta, C)$.
3. Grow C by adding examples from $M(\beta, D)$

Results: Models



Results: Detection



Pascal VOC Challenge results

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Our rank	3	1	2	1	1	2	2	4	1	1	1	4	2	2	1	1	2	1	4	1
Our score	.180	**.411**	.092	**.098**	**.249**	.349	.396	.110	**.155**	**.165**	**.110**	.062	.301	.337	**.267**	**.140**	.141	**.156**	.206	**.336**
Darmstadt							.301													
INRIA Normal	.092	.246	.012	.002	.068	.197	.265	.018	.097	.039	.017	.016	.225	.153	.121	.093	.002	.102	.157	.242
INRIA Plus	.136	.287	.041	.025	.077	.279	.294	.132	.106	.127	.067	.071	**.335**	.249	.092	.072	.011	.092	.242	.275
IRISA		.281					.318	.026	.097	.119			.289	.227	.221		**.175**			.253
MPI Center	.060	.110	.028	.031	.000	.164	.172	.208	.002	.044	.049	.141	.198	.170	.091	.004	.091	.034	.237	.051
MPI ESSOL	.152	.157	**.098**	.016	.001	.186	.120	**.240**	.007	.061	.098	**.162**	.034	.208	.117	.002	.046	.147	.110	.054
Oxford	**.262**	.409				**.393**	**.432**							**.375**					**.334**	
TKK	.186	.078	.043	.072	.002	.116	.184	.050	.028	.100	.086	.126	.186	.135	.061	.019	.036	.058	.067	.090

Table 1. PASCAL VOC 2007 results. Average precision scores of our system and other systems that entered the competition [7]. Empty boxes indicate that a method was not tested in the corresponding class. The best score in each class is shown in bold. Our current system ranks first in 10 out of 20 classes. A preliminary version of our system ranked first in 6 classes in the official competition.

Pascal2006 Person

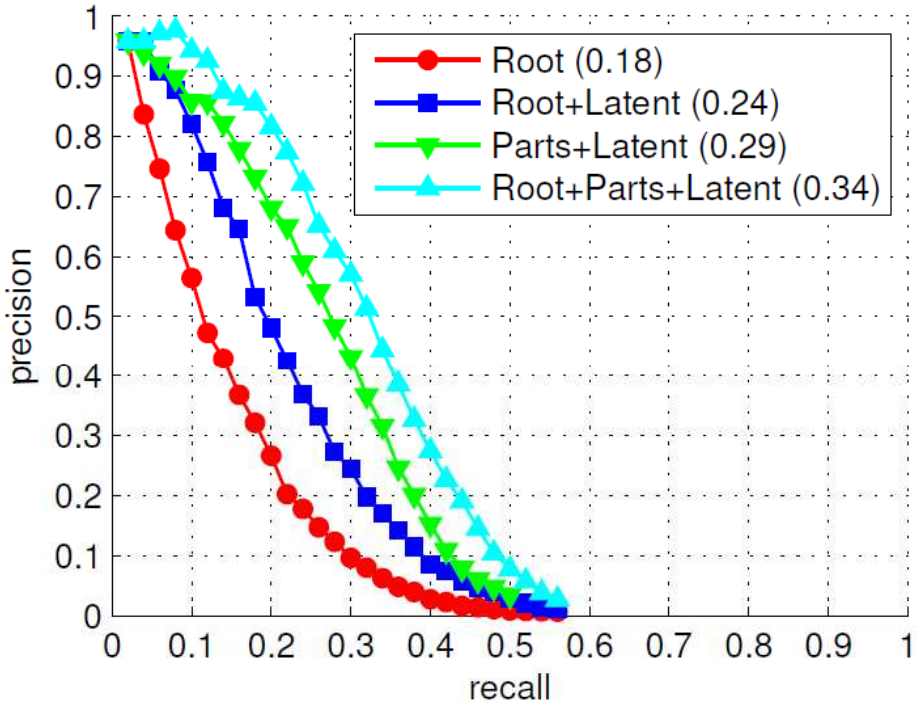


Figure 6. Evaluation of our system on the PASCAL VOC 2006 person dataset. *Root* uses only a root filter and no latent placement of the detection windows on positive examples. *Root+Latent* uses a root filter with latent placement of the detection windows. *Parts+Latent* is a part-based system with latent detection windows but no root filter. *Root+Parts+Latent* includes both root and part filters, and latent placement of the detection windows.