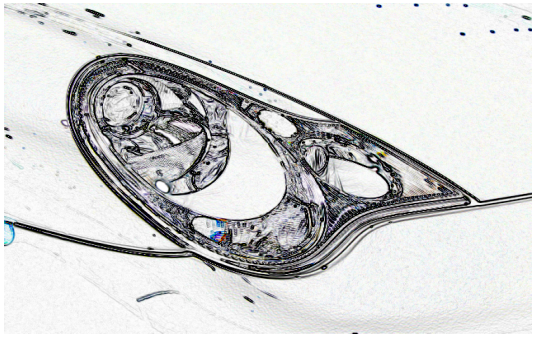
CSE578: Computer Vision

Spring 2015:

Pictorial Structures and Deformable Part Models


Anoop M. Namboodiri and P.J. Narayanan

Center for Visual Information Technology

IIIT Hyderabad, INDIA

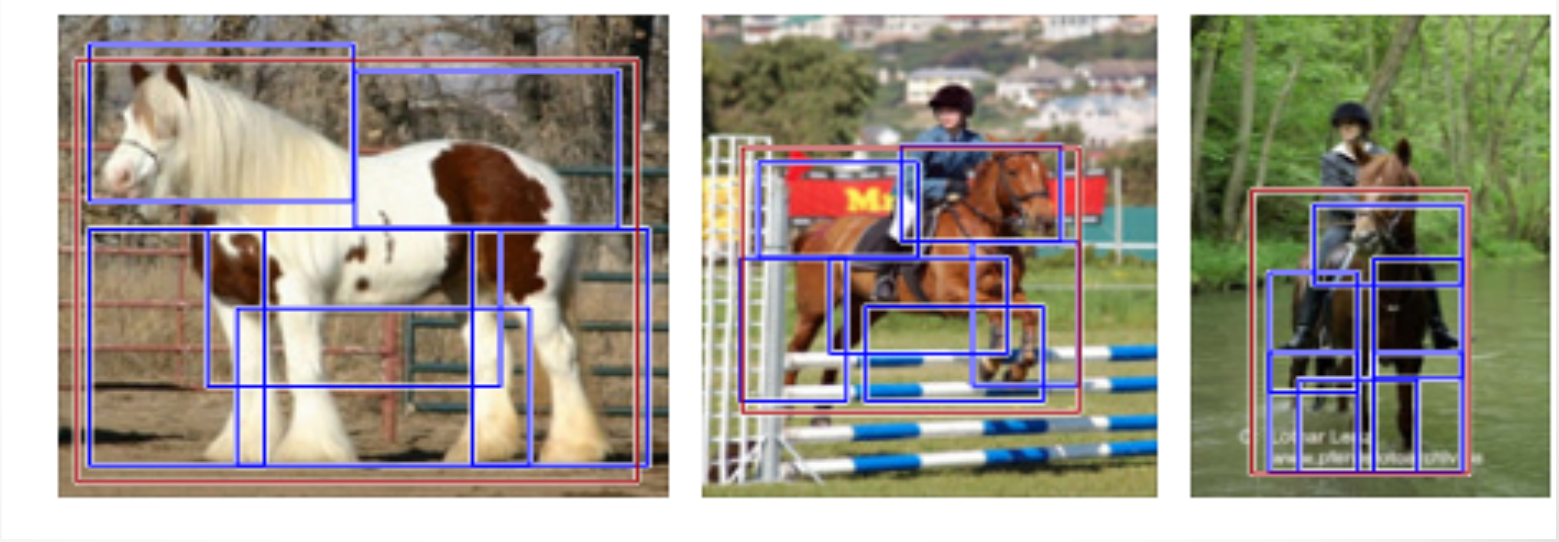
[Slides Generously Borrowed from Various Sources]

Motivation

- Object category detection:

  Detect all objects with the same category in an image

  For example horse detection:



Motivation

- Objects in rich categories exhibit significant variability
  - Viewpoint variation
  - Intra-class variability
    - bicycles of different types (e.g., mountain bikes, tandems...)
    - People wear different clothes and take different poses

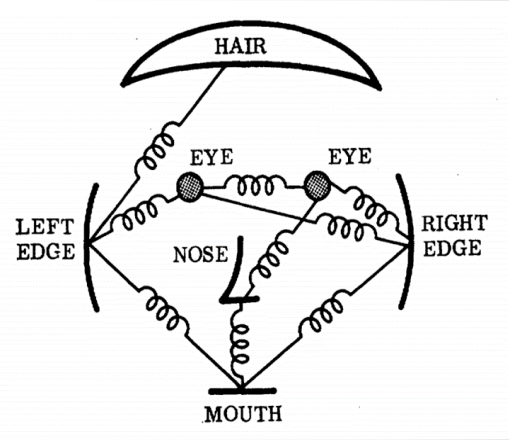
Solution Approaches

- Part Model
- Mixture Model
- Histogram of Gradient
- Feature Pyramid
- Support Vector Machine

- Part Model + Feature Pyramid
  - Pictorial Structures
- HOG + SVM
  - Human Detection: Dalal and Triggs
- All together
  - Deformable Part Model

Part-based Model

- Definition:
  - Root : Capture overall appearance of object
  - Part : Capture local appearance of parts
  - Spring : spatial connections between
- Displacement :
  - Using minimizing energy function to find the optimal displacement



[1] *Pictorial Structures for Object Recognition, Felzenszwalb, Huttenlocher, 2005*

Part-based representation

- K-fans model (D.Crandall, et.all, 2005)

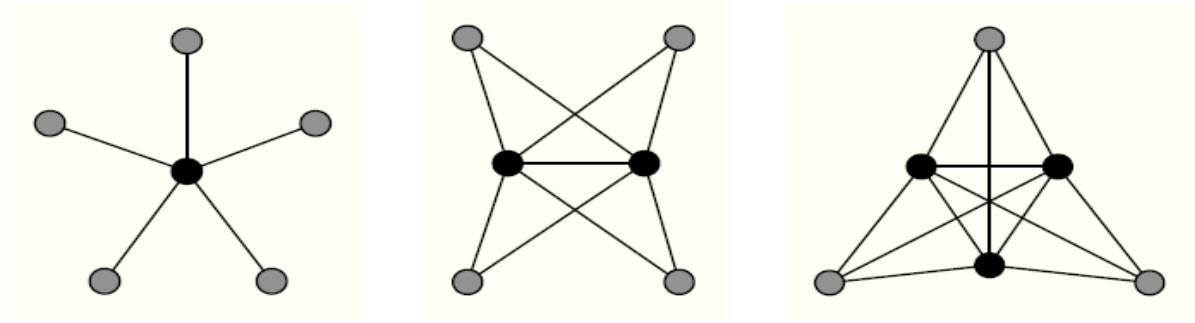
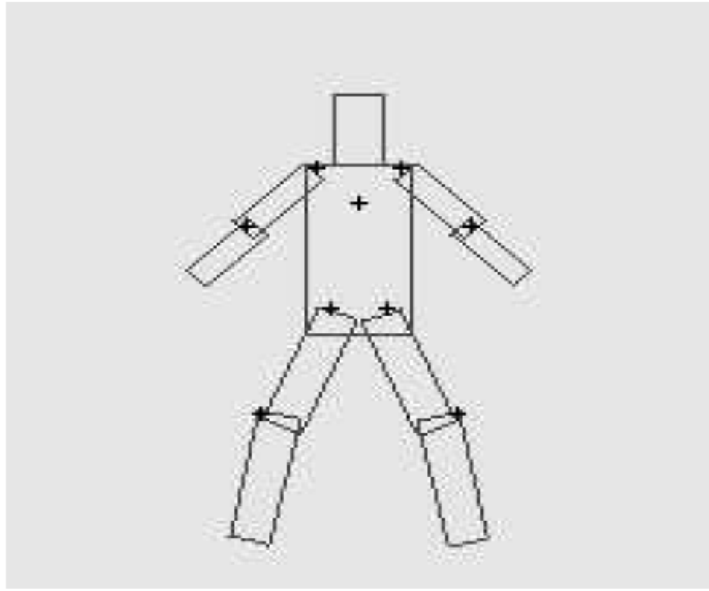
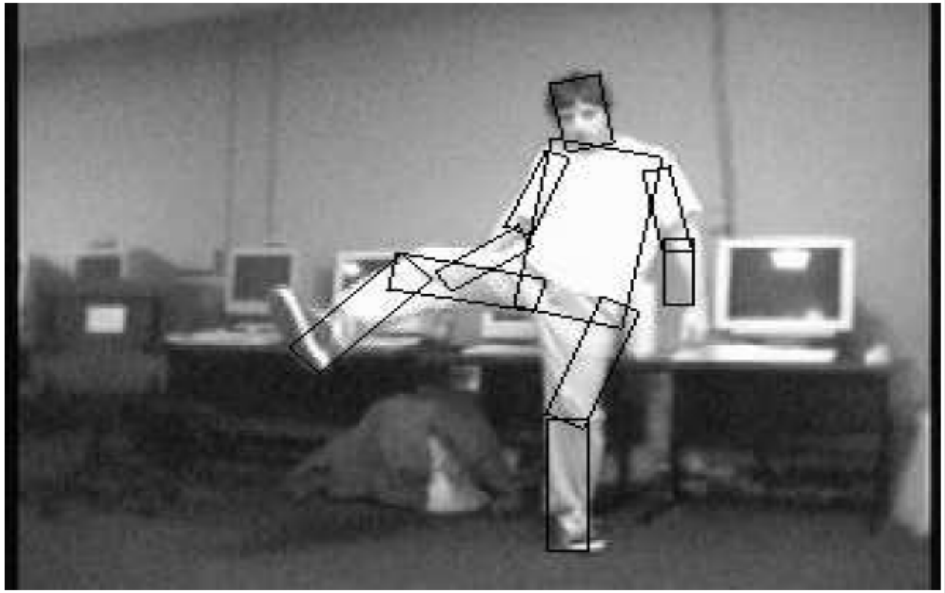


Figure 1. Some k-fans on 6 nodes. The reference nodes are shown in black while the regular nodes are shown in gray.

Part-based representation

- Tree model ➔ Efficient inference by dynamic programming





Pictorial Structure

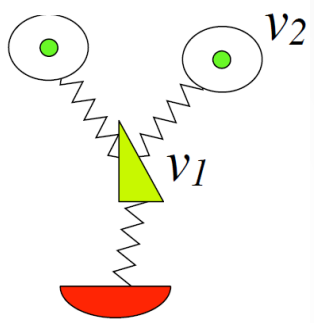
- Matching = Local part evidence + Global constraint

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- $m_i(l_i)$: matching cost for part I
- $d_{ij}(l_i,l_j)$: deformable cost for connected pairs of parts
- (v_i,v_j): connection between part i and j

Matching on tree structure

$$E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j)$$

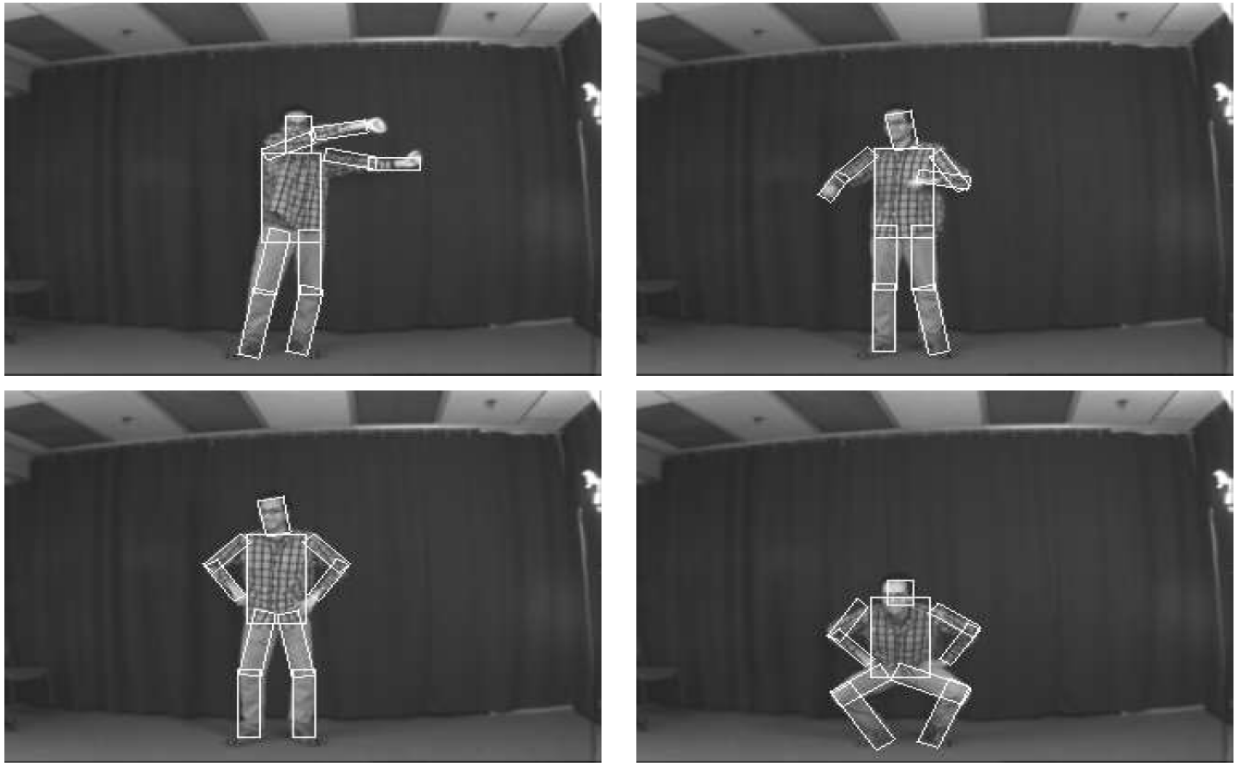


- For each l_1, find best l_2:

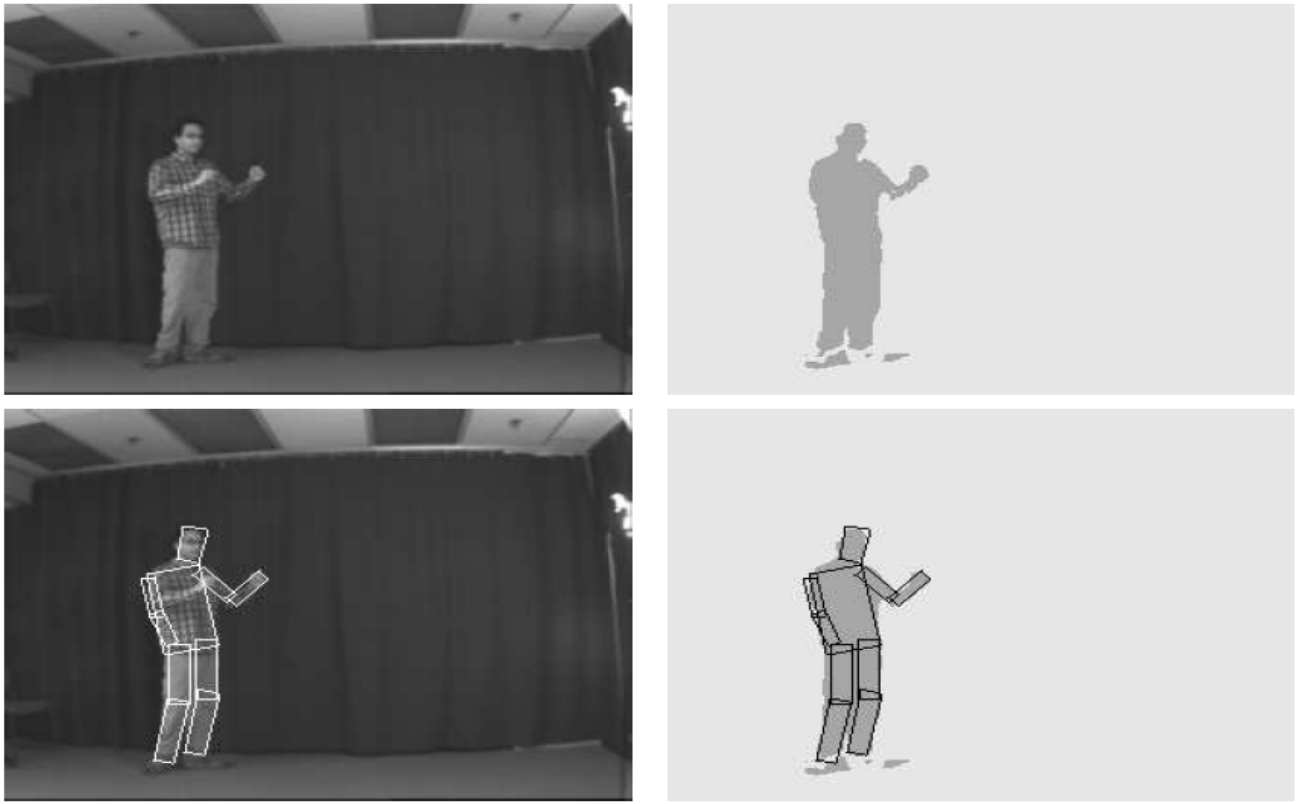
$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1,l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- Complexity: $O(nk^2)$: n parts, k locations per part

Sample result on matching human

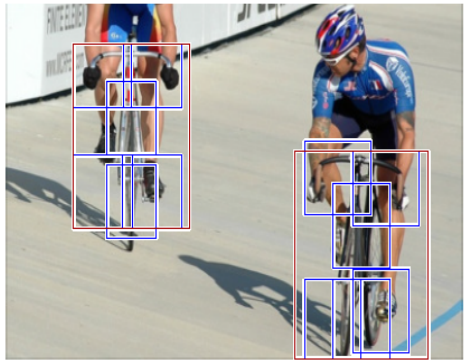


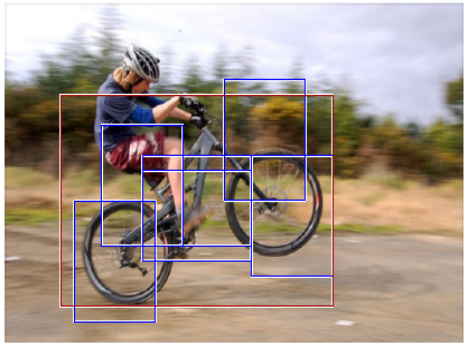
Sample result on matching human



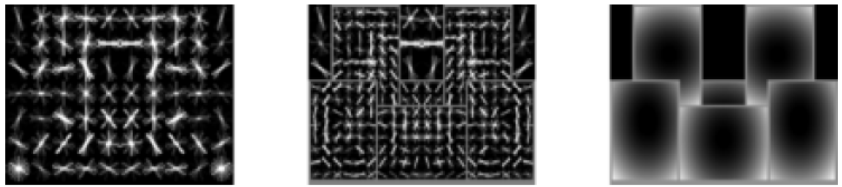
Mixture Model

Build a Mixture Model that includes different components of the same class

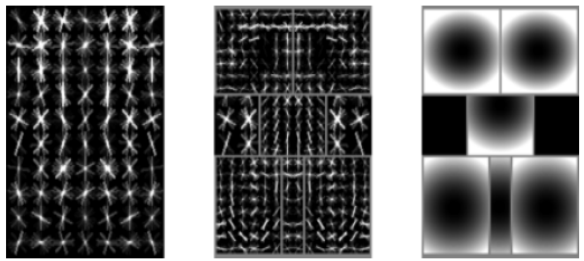




Component 1



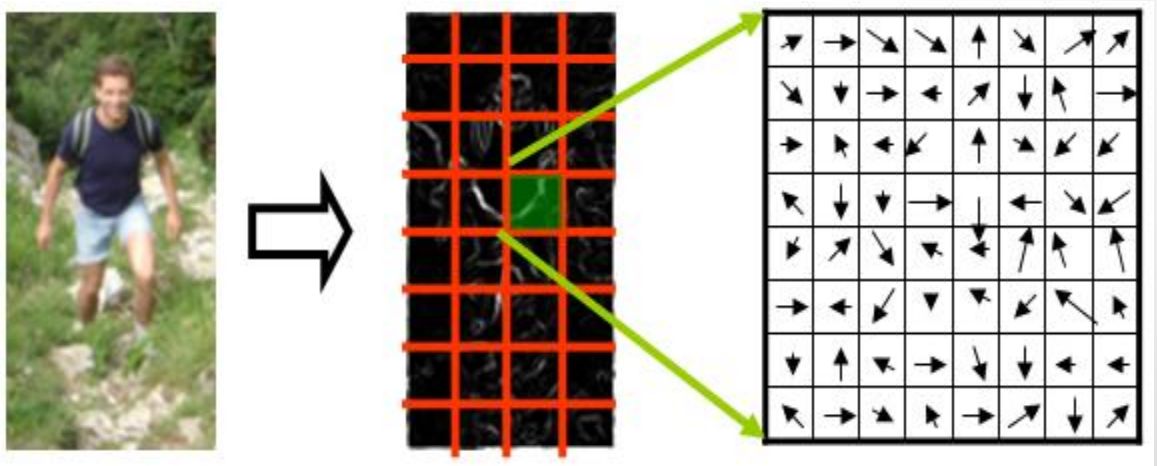
Component 2



Histogram of Gradient

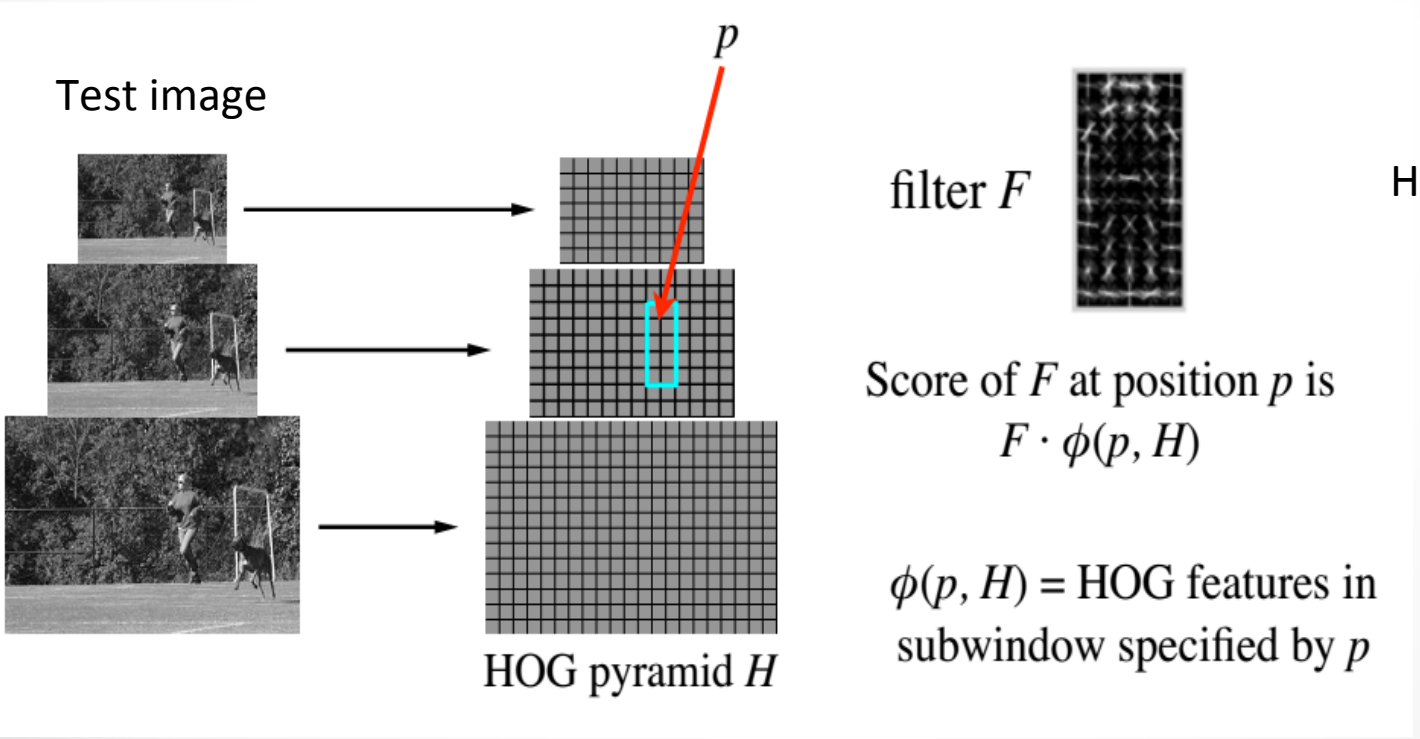
Step:

1. Compute gradient every pixel
2. Group 8×8 pixels into a cell, and 4×4 cells into a block and build histogram of each cells about 9 orientation bins (0^0~180^0)
3. 4×9 vector per block gives n×4×9 dimensional feature vector per window of n blocks
4. Train SVM



Feature Pyramid

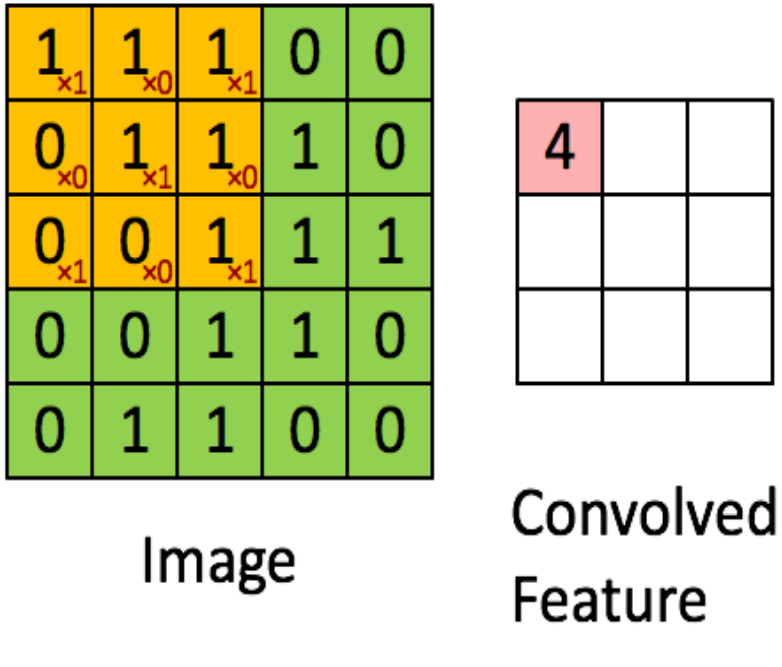
- Develop a representation to decompose images into multiple scales by smoothing and subsampling to extract features of interest and avoid noise



Feature Pyramid

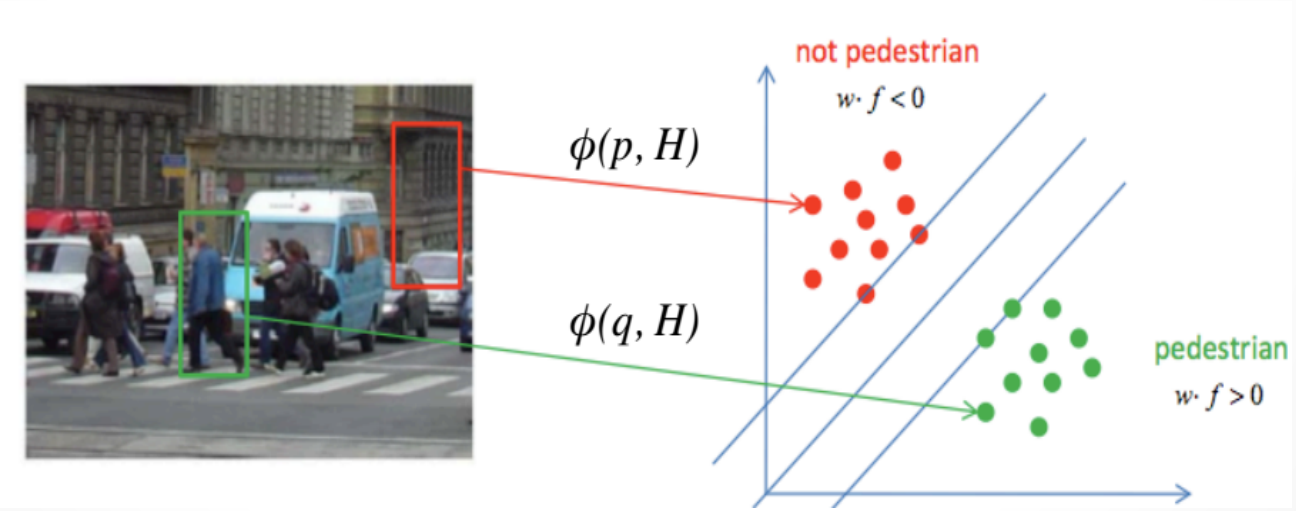
- Subsampling and smoothing:

Gaussian pyramid:



Support Vector Machines

- Build a hyper plane separate positive examples from negative
- In this case, positive is when a human exist in the bounding box

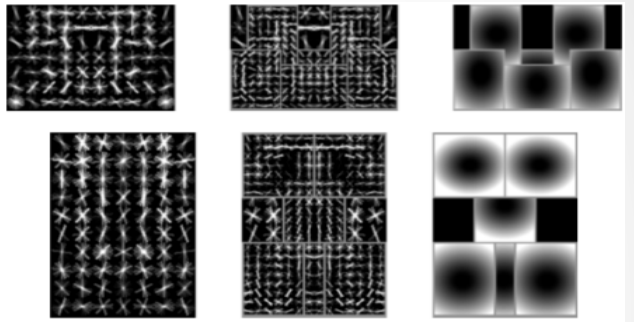


Combining D&T with PS

- Deformable Part Models
  - Build Models
  - Matching
  - Mixture Models
- Latent SVM
- Training Models

Deformable Part Model

- Build a model for an object with n parts :



root filters coarse res. | part filters finer res. | deformation models

$$(F_0, P_1, \ldots, P_n, b)$$

root filter

a model for the i-th part

real-valued bias term

(F_i, v_i, d_i)

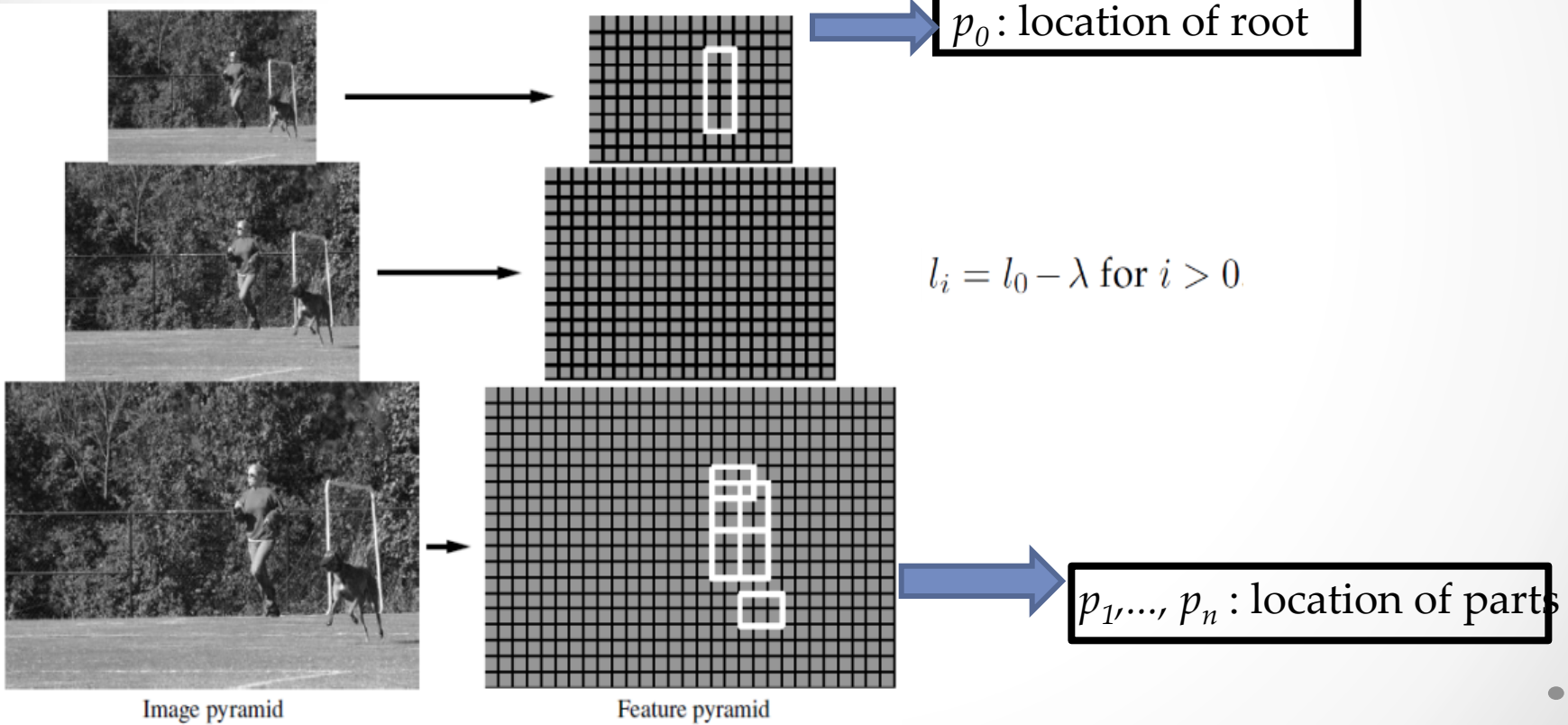
a filter for the i-th part

coefficients of a quadratic function

for part i relative to the root position

Deformable Part Model

Object hypothesis:



Deformable Part Model

- Part filters are placed at twice the spatial resolution of the placement of the root
- z specifies the location of each filter in feature pyramid

  p_i specifies the level and position of the ith filter

$$z = (p_0, \ldots, p_n) \quad p_i = (x_i, y_i, l_i)$$

Deformable Part Model

Score of hypothesis = filter scores - deformation costs

$$\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i' \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b,$$

filters

feature map

deformation parameters

displacements

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$$

Deformable Part Model

- Rewrite the equation:

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

$$\beta = (F'_0, \ldots, F'_n, d_1, \ldots, d_n, b).$$

$$\psi(H, z) = (\phi(H, p_0), \ldots \phi(H, p_n),\ -\phi_d(dx_1, dy_1), \ldots, -\phi_d(dx_n, dy_n), 1).$$

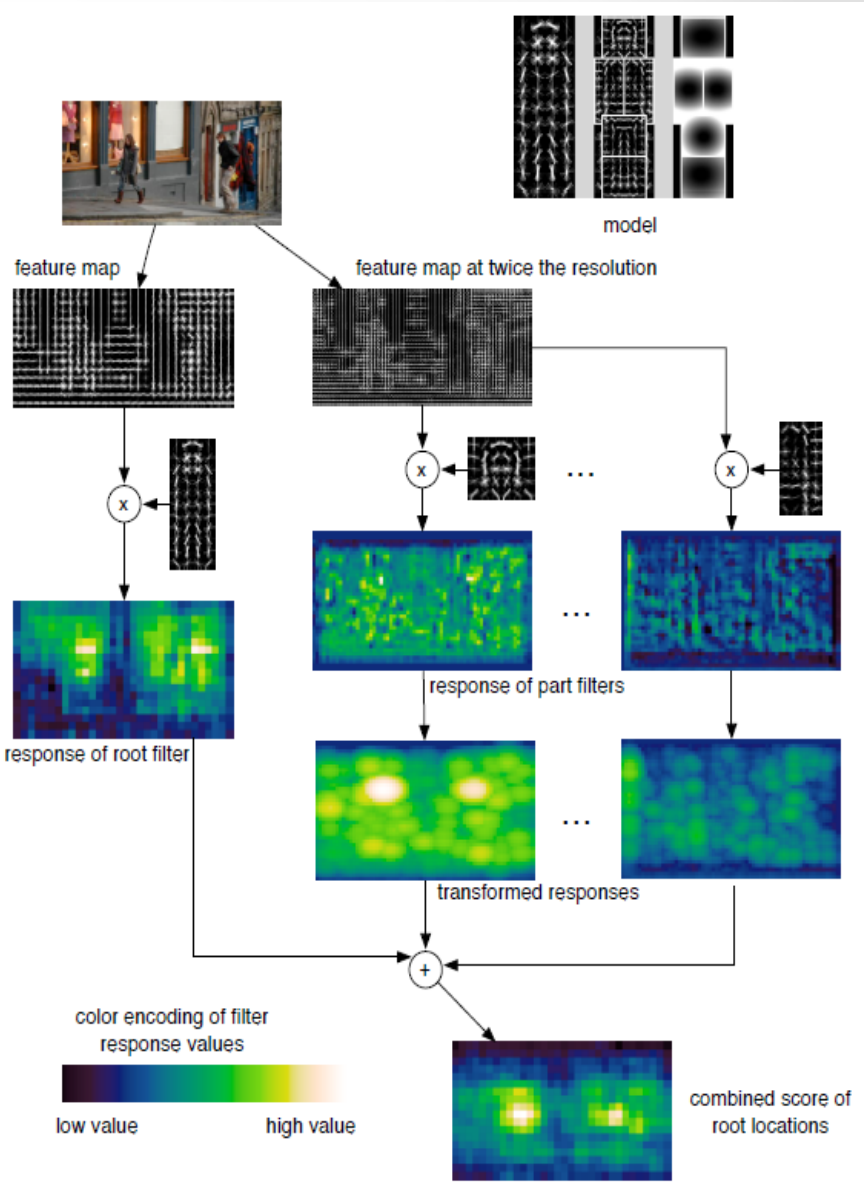
Deformable Part Model

- Given a root position find the best placement of parts:

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

- Using sliding window approach, high score of root score define detections

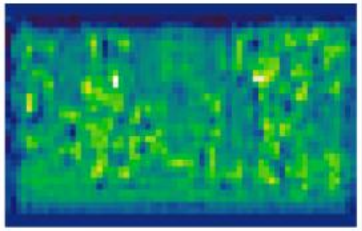
Matching Process



Matching

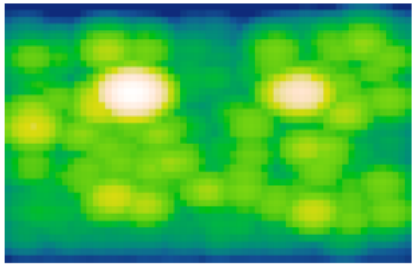
- Response of filter in *l*-th pyramid level

$$R_{i,l}(x,y) = F_i' \cdot \phi(H,(x,y,l))$$



- Transformed response: finding best part displacement relate to root

$$D_{i,l}(x,y) = \max_{dx,dy} \left(R_{i,l}(x+dx,y+dy) - d_i \cdot \phi_d(dx,dy)\right).$$



Matching

Overall root scores :

$$\text{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b$$

Mixture Models

- A mixture model with m components $M = (M_1, \ldots, M_m)$
- 1<=c<=m

$$z = (c, p_0, \ldots, p_{n_c}) \qquad z' = (p_0, \ldots, p_{n_c})$$

$$\beta = (\beta_1, \ldots, \beta_m)$$

$$\psi(H, z) = (0, \ldots, 0, \psi(H, z'), 0, \ldots, 0)$$

$$\beta \cdot \psi(H, z) = \beta_c \cdot \psi(H, z')$$

Mixture Models

- Detect objects using a mixture model, we use matching algorithm to find root positions independently for each component

Latent SVM (LSVM)

- Classifiers that score an example x (bounding box):

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

  - β are model parameters, z are latent values

We want $f_B(x)>0$ when positive

$f_B(x)<0$ when negative

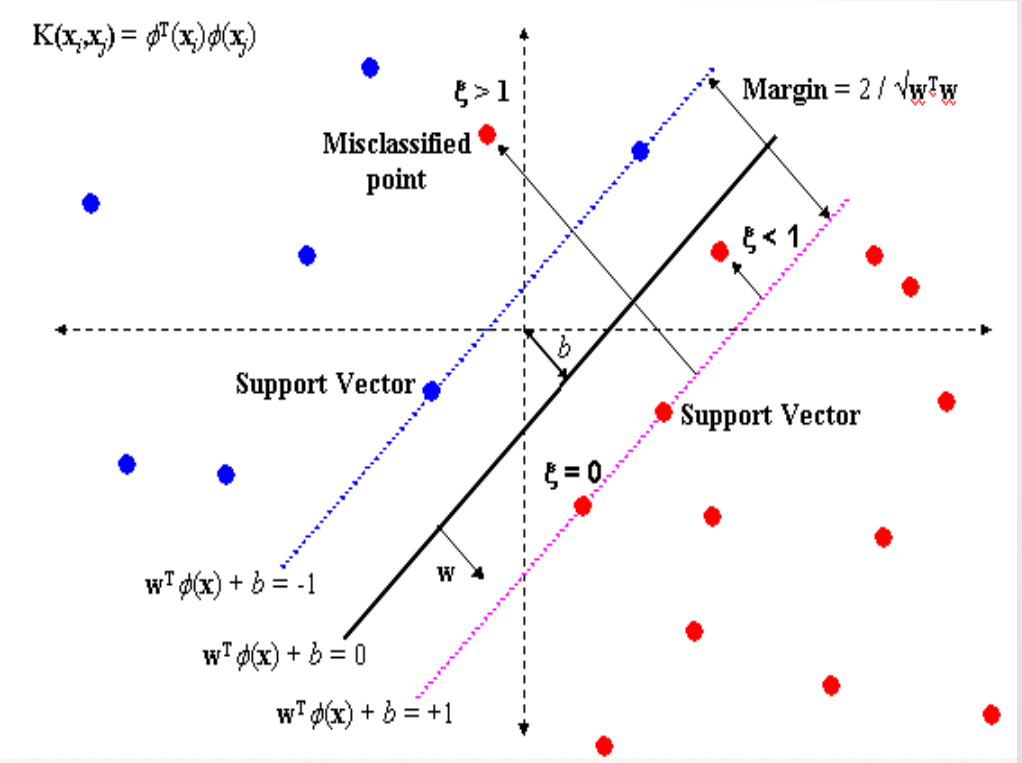
Latent SVM (LSVM)

Minimize :

Hinge loss

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

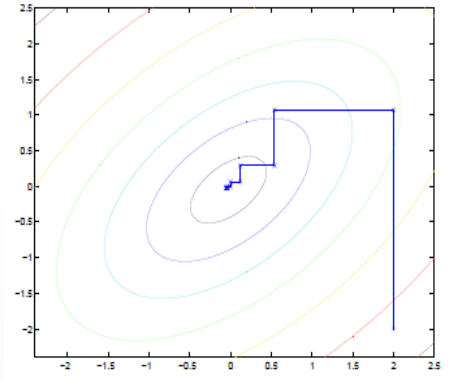
$$y_i \in \{-1, 1\}$$



Latent SVM (LSVM)

- Semi-convex:
  - is convex for negative examples
  - for positive examples, convex if latent values fixed
- Solution fixed latent values by coordinate decent:

1) *Relabel positive examples:* Optimize $L_D(\beta, Z_p)$ over Z_p by selecting the highest scoring latent value for each positive example,
   $z_i = \text{argmax}_{z \in Z(x_i)} \beta \cdot \Phi(x_i, z)$.
2) *Optimize beta:* Optimize $L_D(\beta, Z_p)$ over β by solving the convex optimization problem defined by $L_{D(Z_p)}(\beta)$.



Training Models

- We initial k component with a specific class, sort the bounding boxes by aspect ratio and intraclass variation then split into k group
- Initial root filters and use coordinate decent to update
- Initial part filters by greedily place parts to cover high energy regions of the root filter
- Training by SVM

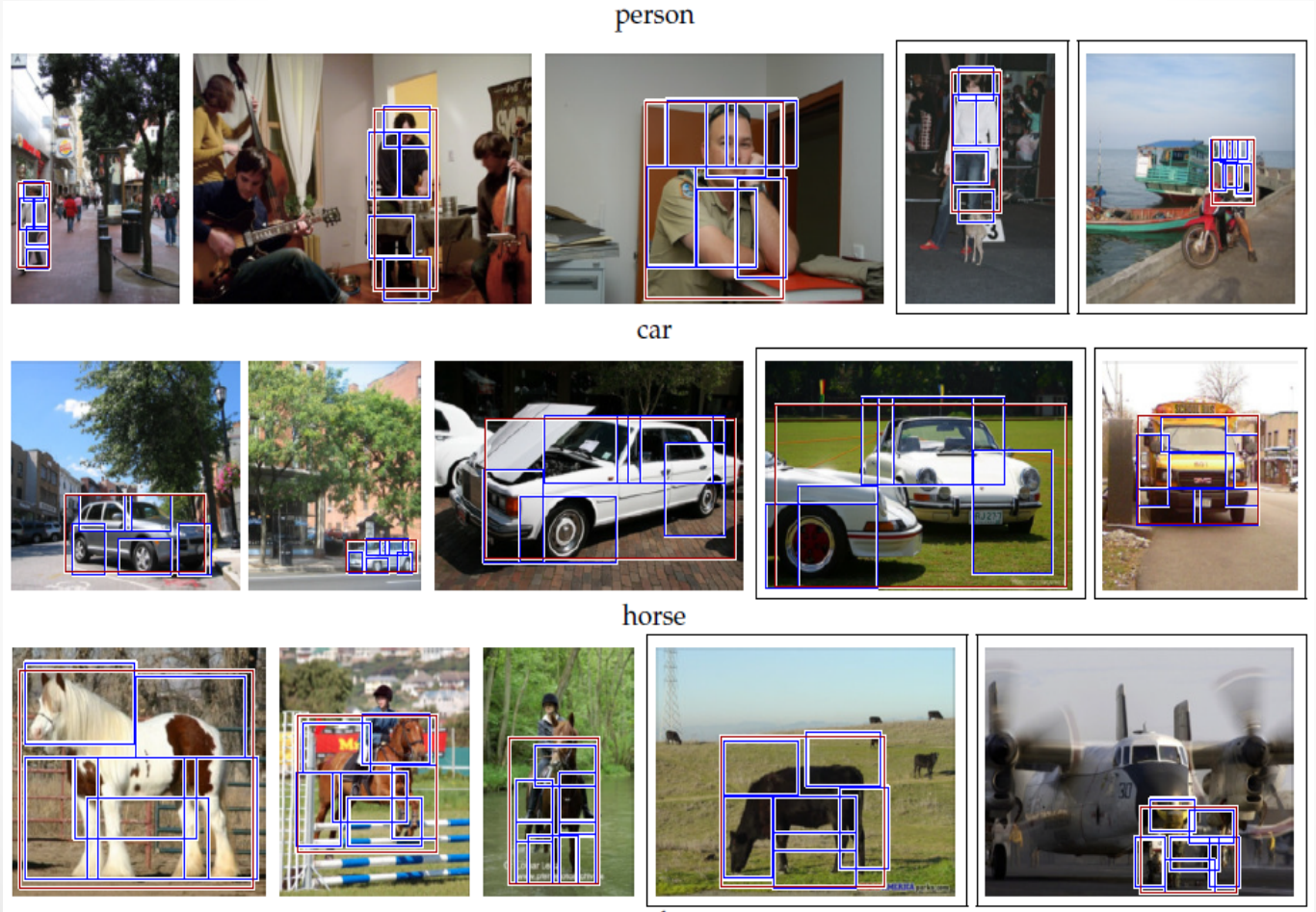
Experimental Results

- PASCAL VOC 2006,2007,2008 comp3 challenge datasets
- Some statistics:
  - It takes 2 seconds to evaluate a model in one image (4952 images in the test dataset)
  - It takes 4 hours to train a model
  - MUCH faster than most systems.
  - All of the experiments were done on a 2.8Ghz 8-core Intel Xeon Mac Pro computer running Mac OS X 10.5.

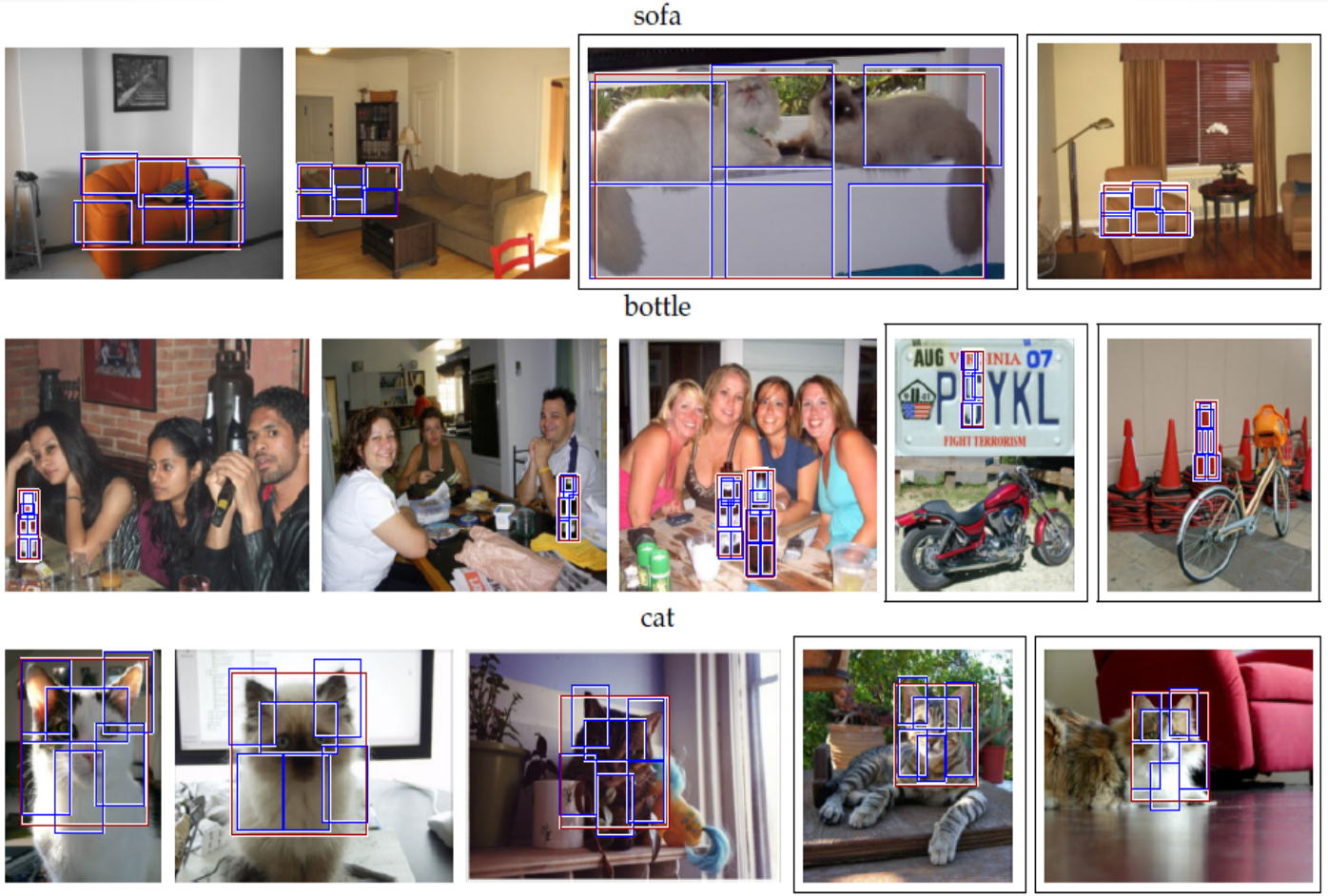
Experimental Results

Measurement: predicted bounding box is correct if it overlaps more than 50 percent with ground truth bounding box; otherwise, considered false positive

Experimental Results



Experimental Results

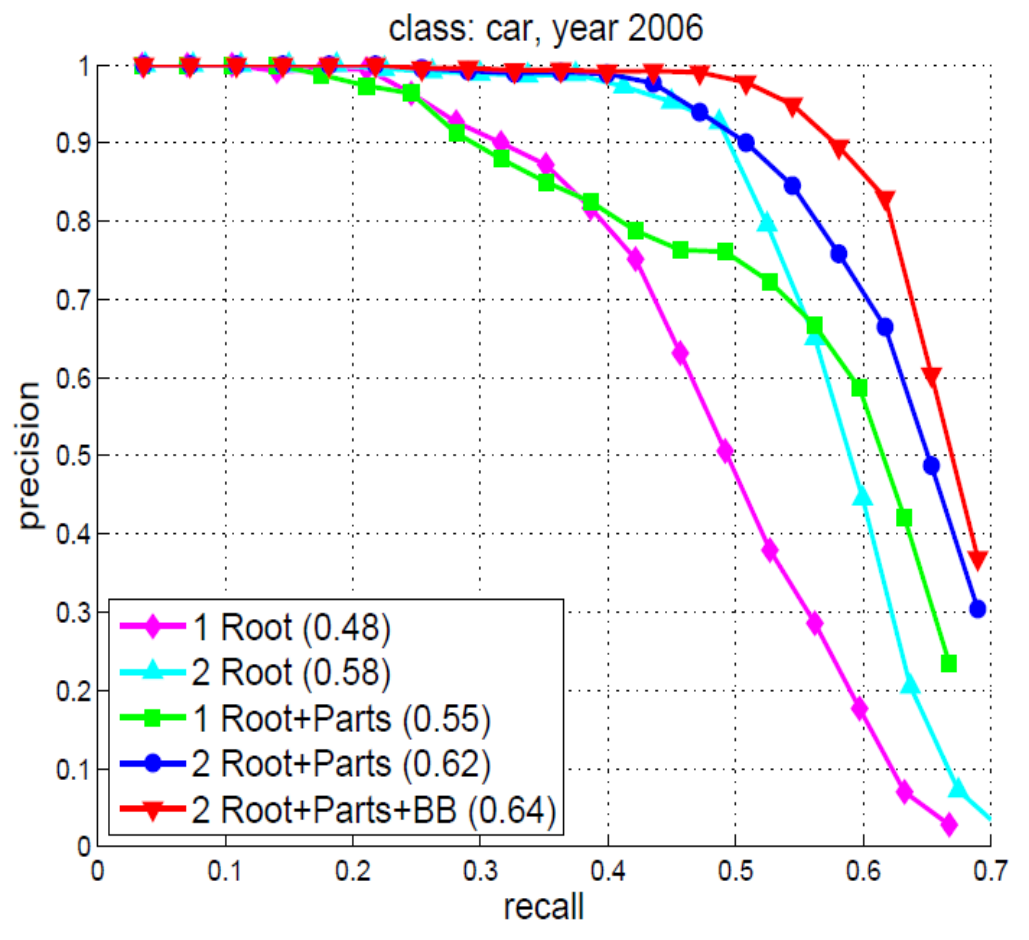


Experimental Results

- Best Average Precision score in 9 out of 20, second in 8

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	pers	plant	sheep	sofa	train	tv
a) base	.336	.371	.066	.099	.267	.229	.319	.143	.149	.124	.119	.064	.321	.353	.407	.107	.157	.136	.228	.324
b) BB	.339	.381	.067	.099	.278	.229	.331	.146	.153	.119	.124	.066	.322	.366	.423	.108	.157	.139	.234	.328
c) context	.351	.402	.117	.114	.284	.251	.334	.188	.166	.114	.087	.078	.347	.395	.431	.117	.181	.166	.256	.347
d) rank	2	1	1	1	1	1	2	2	1	2	4	5	2	2	1	1	2	2	3	1
(UofCTTIUCI)	.326	.420	.113	.110	.282	.232	.320	.179	.146	.111	.066	.102	.327	.386	.420	.126	.161	.136	.244	.371
CASIA Det	.252	.146	.098	.105	.063	.232	.176	.090	.096	.100	.130	.055	.140	.241	.112	.030	.028	.030	.282	.146
Jena	.048	.014	.003	.002	.001	.010	.013		.001	.047	.004	.019	.003	.031	.020	.003	.004	.022	.064	.137
LEAR PC	.365	.343	.107	.114	.221	.238	.366	.166	.111	.177	.151	.090	.361	.403	.197	.115	.194	.173	.296	.340
MPI struct	.259	.080	.101	.056	.001	.113	.106	.213	.003	.045	.101	.149	.166	.200	.025	.002	.093	.123	.236	.015
Oxford	.333	.246					.291			.125			.325	.349						
XRCE Det	.264	.105	.014	.045	.000	.108	.040	.076	.020	.018	.045	.105	.118	.136	.090	.015	.061	.018	.073	.068

Experimental Results



Conclusions

- Deformable Part Model
  - Fast matching algorithm
  - handle Viewpoint variation, and Intra-class variability problems
- Still have some problem need to solve:
  - Fixed box size
  - Fixed number of components
- Future Work
  - Build grammar based models that represent objects with variable hierarchical structures
  - Sharing part models between components