Motivation

• Object category detection:
  – Detect all objects with the same category in an image
    • For example horse detection:
Classification Task

At least one bus

100%
**Detection Task**

Predicted bounding box should overlap by at least 50% with ground truth!!!
Detections “near misses”
Pascal VOC Challenge-The Object Classes

Images retrieved from flicker website.
Successful Detection Method  (P.Felzenszwalb et al.)

- Joint winner in 2009 Pascal VOC challenge with the Oxford Method.
- Award of "lifetime achievement" in 2010.
- Mixture of deformable part models.
- Each component has global template + deformable parts
  - HOG feature templates.
- Fully trained from bounding boxes alone.
Recap: Deformable Models

Challenge:
- handling variation of appearances within object classes
- non-rigid objects

Idea:
- Consider objects as a deformed version of a template!
What is DPM?

- Deformable Part is a discriminatively trained, multi-scale model for image training that aim at making possible the effective use of more latent information such as hierarchical (grammar) models and models involving latent three dimensional pose.
DPM

• Definition:
  – Root : Catch Roughly appearance of object
  – Part : Catch local appearance of object
  – Spring : spatial connections between parts

• Displacement :
  – Using minimizing energy function to find the optimal displacement

[1] *Pictorial Structures for Object Recognition, Daniel P. Huttenlocher*, 1973
The deformable model includes both a coarse global template covering an entire object and higher resolution part templates. The templates represent histogram of gradient features.
Modeling Spatial Relations with DPM

Spring-based models

[Fischler & Elschlager, ’73], [Ramanan et al,’ 07],
[Felszwenwalb et al,’ 05,’ 09], [Kumar et al, ‘09]
Parts Based Model

Goal: Assign model parts to image regions preserving both local appearance and spatial relationships
Matching Problem

- Model is represented by a graph $G = (V, E)$
  - $V = \{v_1, ..., v_n\}$ are the parts
  - $(v_i, v_j) \in E$ indicates a connection between parts
- $m_i(l_i)$ is a cost for placing part $i$ at location $l_i$
- $d_{ij}(l_i, l_j)$ is a deformation cost
- Optimal configuration for the object is $L = (l_1, ..., l_n)$ minimizing

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

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

- Assume \( n \) parts, \( k \) possible locations for each part
  - There are \( k^n \) configurations \( L \)
- If graph is a tree we can use dynamic programming
  - \( O(nk^2) \) algorithm
- If \( d_{ij}(l_i,l_j) = g(l_i-l_j) \) we can use min-convolutions
  - \( O(nk) \) algorithm
  - As fast as matching each part separately!
Inference problem: What is the best scoring assignment $f$?

\[ \arg \max_f \text{Score}[f] = \arg \max_f \left( \sum \theta_{a;f(a)} + \sum_{a,b} \theta_{a,b;f(a),f(b)} \right) \]

For trees can use **belief propagation** for exact solution in polytime

Inference is **NP-hard** for general graphs
Parts based models - Learning Problem

**Linear models:**

\[ \theta_a; f(a) = w_a^T \bar{\theta}_a; f(a) \]

\[ \theta_{a,b}; f(a), f(b) = w_{ab}^T \bar{\theta}_{ab}; f(a)f(b) \]

**Learning linear models:** Find weight vectors that best separate positive and negative examples. E.g.,

\[
\arg\min_w \frac{1}{2} \|w\|^2 + C \left( \sum_k \xi^{(k)} + \sum_\ell \xi^{(\ell)} \right) \\
\text{s.t. } w^T \bar{\theta}_+^{(k)} \geq 1 - \xi^{(k)}, \ \forall k \\
w^T \bar{\theta}_-^{(\ell),f} \leq 1 + \xi^{(\ell)}, \ \forall \ell, f \\
\xi^{(k)} \geq 0, \ \forall k, \xi^{(\ell)} \geq 0, \ \forall \ell
\]

[Kumar et al.,’ 09]
Finding Motorbike

Model with 6 parts:
- 2 wheels
- 2 headlights
- front & back of seat
Human Tracking

Ramanan, Forsyth, Zisserman, Tracking People by Learning their Appearance
IEEE Pattern Analysis and Machine Intelligence (PAMI). Jan 2007
Human Pose Estimation
HOG: Histogram of Gradients

- 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.
- We compute features at different resolutions (pyramid)
1. Compute gradient every pixel
2. Group 8*8 pixels into a cell, and 4*4 cells into a block.

Build histogram of each cells about 9 orientation bins (0°~180°)
Histogram of Gradient

3. A block contains a 4*9 feature vector for local information, and a bounding box which contains n’s blocks has n*4*9 feature vector.

4. Training by SVM
Filters

- Filters are rectangular templates defining weights for features
- Score is dot product of filter and subwindow of HOG pyramid

Score of $H$ at this location is $H \cdot W$
Object Hypothesis

Score is sum of filter scores plus deformation scores

Multiscale model captures features at two-resolutions
Training

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

Bottle

Car

Sofa

Bicycle
Training Models

• Latent SVM is used for learning the training images
Experiment Result

- 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.
Experiment Result

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

person

car

horse
Experiment Result

sofa

bottle

cat
Experiment Result

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

class: car, year 2006

precision vs. recall

How many selected items are relevant? Precision =

How many relevant items are selected? Recall =
Slice Credits and References

• Jonathan Huang, Tomasz Malisiewicz, 2009.