More sliding window detection: Discriminative part-based models

Many slides based on P. Felzenszwalb
Challenge: Generic object detection
Pedestrian detection

- Features: Histograms of oriented gradients (HOG)
  - Partition image into 8x8 pixel blocks and compute histogram of gradient orientations in each block
- Learn a pedestrian template using a linear support vector machine
  - At test time, convolve feature map with template

N. Dalal and B. Triggs,
*Histograms of Oriented Gradients for Human Detection*, CVPR 2005
Discriminative part-based models

P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan,
Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010
Object hypothesis

- Multiscale model: the resolution of part filters is twice the resolution of the root.

\[ z = (p_0, \ldots, p_n) \]

- \( p_0 \): location of root
- \( p_1, \ldots, p_n \): location of parts

Score is sum of filter scores minus deformation costs.

Score of the filter: inner products between the filter and features.
Part-based representation

Objects are decomposed into parts and spatial relations among parts

E.g. Face model by Fischler and Elschlager ‘73
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 \( l \)
\( d_{ij}(l_i,l_j) \): 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
Scoring an object hypothesis

- The score of a hypothesis is the sum of filter scores minus the sum of deformation costs

\[
score(p_0, ..., p_n) = \sum_{i=0}^{n} F_i \cdot H(p_i) - \sum_{i=1}^{n} D_i \cdot (dx_i, dy_i, dx_i^2, dy_i^2)
\]
Scoring an object hypothesis

- The score of a hypothesis is the sum of filter scores minus the sum of deformation costs

\[
\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot H(p_i) - \sum_{i=1}^{n} D_i \cdot (dx_i, dy_i, dx_i^2, dy_i^2)
\]

- Recall: pictorial structures

\[
E(l_1, \ldots, l_n) = \sum_i m_i(l_i) + \sum_{i,j} d_{ij}(l_i, l_j)
\]
Scoring an object hypothesis

- The score of a hypothesis is the sum of filter scores minus the sum of deformation costs

\[
score(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot H(p_i) - \sum_{i=1}^{n} D_i \cdot (dx_i, dy_i, dx_i^2, dy_i^2)
\]

- Filters
- Deformation weights

\[
score(z) = w \cdot H(z)
\]

- Concatenation of filter and deformation weights
- Concatenation of subwindow features and displacements
Detection

- Define the score of each root filter location as the score given the best part placements:

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

• Define the score of each root filter location as the score given the best part placements:

\[ \text{score}(p_0) = \max_{p_1, \ldots, p_n} \text{score}(p_0, \ldots, p_n) \]

• Efficient computation: generalized distance transforms
  • For each “default” part location, find the best-scoring displacement

\[ R_i(x, y) = \max_{dx, dy} \left( F_i \cdot H(x + dx, y + dy) - D_i \cdot (dx, dy, dx^2, dy^2) \right) \]
Matching result
Training

- Training data consists of images with labeled bounding boxes
- Need to learn the filters and deformation parameters
Training

- The classifier has the form

\[ f(x) = \max_z w \cdot H(x, z) \]

- \( w \) are model parameters, \( z \) are \textit{latent} hypotheses

- **Latent SVM** training:
  - Initialize \( w \) and iterate:
    - Fix \( w \) and find the best \( z \) for each training example (detection)
    - Fix \( z \) and solve for \( w \) (standard SVM training)

- Issue: too many negative examples
  - Do “data mining” to find “hard” negatives
Car model

Component 1

Component 2
Car detections

high scoring true positives

high scoring false positives
Person model
Person detections

high scoring true positives

high scoring false positives
(not enough overlap)
Cat model
Cat detections

high scoring true positives

high scoring false positives (not enough overlap)
Bottle model
More detections

horse

sofa

bottle
Quantitative results (PASCAL 2008)

• 7 systems competed in the 2008 challenge
• Out of 20 classes, first place in 7 classes and second place in 8 classes
Summary

- Deformable model for object detection
  - Coarse root filter and finer part filter
  - Learn from weakly labeled data
  - Fast algorithm for matching
  - State-of-the-art results on PASCAL challenge