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

Viterbi algorithm

Main idea: determine optimal position (state) of predecessor, for each possible position of self. Then backtrack from best state for last vertex.

\[ E_{\text{total}} = E_1(v_1, v_2) + E_2(v_2, v_3) + ... + E_{n-1}(v_{n-1}, v_n) \]

Complexity: \( O(nm^2) \) vs. brute force search ____?

Example adapted from Y. Boyko
The Viterbi Algorithm

\[ V(i, k) = \begin{cases} 
\max_j V(j, k-1) P_i (q_j | q_i) P_e (x_k | q_i) & \text{if } k > 0, \\
P_i(q_i | q_0) P_e(x_0 | q_i) & \text{if } k = 0.
\end{cases} \]

\[ \phi_{\text{max}} = \arg \max \, \phi_{i,L-1} V(i, L-1) P_i(q_0 | q_i) \]

Viterbi: Traceback

\[ V(i, k) = \begin{cases} 
\max_j V(j, k-1) P_i (q_j | q_i) P_e (x_k | q_i) & \text{if } k > 0, \\
P_i(q_i | q_0) P_e(x_0 | q_i) & \text{if } k = 0.
\end{cases} \]

\[ T(i, k) = \begin{cases} 
\arg \max_j V(j, k-1) P_i (q_j | q_i) P_e (x_k | q_i) & \text{if } k > 0, \\
0 & \text{if } k = 0.
\end{cases} \]

\[ T( T( T( T( T(i, L-1), L-2), ..., 2), 1), 0) = 0 \]
Viterbi Algorithm in Pseudocode

```pseudocode
procedure viterbi(\alpha, \beta, \lambda, \delta, \gamma, \pi, \omega, \lambda_{\text{null}}, \lambda_{\text{null}})
1. for \kappa=0 up to |\Pi|-1 do
2. for i=0 up to |\Xi|-1 do
3. \lambda[(i, \kappa)] = \omega;
4. \tau[(i, \kappa)] = \text{NIL};
5. for i=1 up to |\Xi|-1 do
6. \lambda[(i+1, 0)] = \log(P_{\text{t}}(q_i, q_i)) + \log(P_{\text{e}}(\delta([\Xi][i],[\lambda])));
7. if \lambda[(i+1, 0)] > -\infty then \tau[(i+1, 0)] = \text{NIL};
8. for k=1 up to |\Pi|-1 do
9. for each q \in \lambda_{\text{null}} of \tau[(i, k-1)] do
10. \lambda[(i, k)] = \max_{q \in \lambda_{\text{null}} of \tau[(i, k-1)]} \lambda[(i, k-1)] + \log(P_{\text{t}}(q_i, q_i)) + \log(P_{\text{e}}(\gamma([\Pi][k], q_i)));
11. if \lambda[(i, k)] > \lambda[(i, k-1)] then
12. \tau[(i, k)] = \tau[(i, k-1)];
13. \gamma[(i, k)] = \text{NIL};
14. \gamma[(i, k)] = \text{NIL};
15. \gamma[(i, k)] = \text{NIL};
16. \gamma[(i, k)] = \text{NIL};
17. \gamma[(i, k)] = \text{NIL};
18. \gamma[(i, k)] = \text{NIL};
19. \gamma[(i, k)] = \text{NIL};
20. \gamma[(i, k)] = \text{NIL};
21. for i=0 down to 0 do
22. push \phi, 0;
23. \gamma[(i, k)] = \text{NIL};
24. \gamma[(i, k)] = \text{NIL};
25. return \phi;
```

Matching on tree structure

\[ E(L) = \sum_{i=1}^{n} m(l_i) + \sum_{(v, v') \in E} d_v(l_i, l_j) \]

For each \( l_1 \), find best \( l_2 \):

\[ \text{Best}_2(l_1) = \min_{l_2} \left[ m_2(l_2) + d_1(l_1, l_2) \right] \]

Remove \( v_2 \), and repeat with smaller tree, until only a single part

Complexity: \( O(nk^2) \): \( n \) parts, \( k \) locations per part

\[ B_j(l_i) = \min_{l_j} \left( m_j(l_j) + d_1(l_i, l_j) + \sum_{v \in C_j} B_v(l_j) \right) \]

For root no 2\textsuperscript{nd} term, for leaves no 3\textsuperscript{rd} term
Sample result on matching human

Pictorial Structures
We can efficiently solve the above optimization Problem using distance transform in linear $O(nk)$

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

Pictorial structures combine local appearance scores with global spatial constraints
Discriminatively trained part based models

Filters

Filters are rectangular templates defining weights for features

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

Coarser level for the root filter (whole object)
and higher level for part filters

Object hypothesis

• Multiscale model: the resolution of part filters is twice the resolution of the root

Score is sum of filter scores minus deformation costs

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

\[ p_0 : \text{location of root} \]

\[ p_1, \ldots, p_n : \text{location of parts} \]
Scoring an object hypothesis

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

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

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

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

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

Detection

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

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

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

Detection

[Diagram of detection process]
Matching result

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 (filters and deformation parameters, \( z \) are \textit{latent} hypotheses)
- \( x \) is detection window, \( z \) are features and filter placements
- **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

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

Quantitative results (PASCAL 2008)

- Bicycles
- Person
- Bird
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

Implicit shape models

• Combining the edge based Hough Transform style voting with appearance codebooks
• Visual codebook is used to index votes for object position

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
Implicit shape models

- Visual codebook is used to index votes for object position

Idea Implicit Shape Model

Faces rectangular templates – detection windows

Does not generalize to more complex object with different shapes
Initial Recognition Approach

First Step: Generate hypotheses from local features
Training: Agglomerative Clustering

- How to decide when to merge two clusters
- Average NCC of patches

$\text{similarity}(C_1, C_2) = \frac{\sum_{p \in C_1} \sum_{q \in C_2} \text{NCC}(p, q)}{|C_1| \times |C_2|} > r, \quad \text{NCC}(p, q) = \frac{\sum (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum (p_i - \bar{p})^2 \sum (q_i - \bar{q})^2}}$

- NCC between two patches

Codebook words - spatial information is lost
For each codebook entry store all positions it was activated in relative to object center (positions parametrized by $r$ and $\theta$)
Parts vote for object center

Lowe’s DoG Detector
Resize to $25 \times 25$
Find codebook patches
Learn Spatial Distribution

3 $\sigma \times 3 \sigma$ patches
Pedestrian Detection

1. Interleaved Object Categorization and Segmentation, BMVC’03

Pedestrian Detection

Many applications
Large variation in shape, appearance
Need to combine different representations
Basic Premise: “[Such a] problem is too difficult for any type of feature or model alone”
Probabilistic bottom up, top down segmentation
Open Question: How would you do pedestrian detection/segmentation?

Solution: Integrate as many cues as possible from many sources.

- Support of segmentation from local features
- Support of segmentation from global features (Chamfer Matching)

Goal: Localize AND count pedestrians in a given image.

Datasets:
- Training Set: 35 people walking parallel to the image plane
- Testing Set (Much harder!): 209 images of 595 annotated pedestrians
Initial Recognition Approach

First Step: Generate hypotheses from local features (Intrinsic Shape Models)

Testing:
Initial Hypothesis: Overall
Initial Recognition Approach

First Step: Generate hypotheses from local features (Intrinsic Shape Models)

Testing:
Initial Hypothesis: Overall

Second Step: Segmentation based Verification (Minimum Description Length)

Caveat: it leads to another set of problems

Or four legs and three arms
ISM doesn’t know a person doesn’t have three legs!

Global Cues are needed
Assimilation of Global Cues

Distance Transform, Chamfer Matching

$$D_{Chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} \min(DT_t(t), r)$$

get Feature Image by an edge detector
get DT image by computing distance to nearest feature point
Chamfer Distance between template and DT image

Assimilation of Global Cues (Attempt 1)

Distance Transform, Chamfer Matching

Initial hypothesis generated by local features
Use scale estimate to cut out surrounding region
Apply Canny detector and compute DT