A Discriminatively Trained, Multiscale, Deformable Part Model

by Pedro Felzenszwalb, David McAllester, and Deva Ramanan

CS381V Visual Recognition - Paper Presentation
Deformable objects

Images from D. Ramanan’s dataset
Deformable objects

Images from Caltech-256
Deformable Part Model

- Introduced by Fischler and Elschlager in 1973
- Part-based models:
  - Each part represents local visual properties
  - “Springs” capture spatial relationships

Matching model to image involves joint optimization of part locations “stretch and fit”
Deformable Part Model

Step 1: HOG Pyramid
Step 1: HOG Pyramid

- Normalize w.r.t. the sum of histogram values in each 2 x 2 block containing the cell.

4 x 9 descriptor:
Step 1: HOG Pyramid

Filters

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

Filter $H$ is a $w \times h \times 4 \times 9$ vector.
Object Hypothesis

Multiscale model captures features at two-resolutions

Score is sum of filter scores plus deformation scores
Training

- Training data consists of images with labeled bounding boxes
- Need to learn the model structure, filters and deformation costs
Step 2: Initialize Root Filter

Step 2: Train Root Filter

Unscaled Image

Find best placement in HOG pyramid.
At least 50% overlap w/ ground truth.

Step 2 Summary

1. Set filter size based on statistics in the data.
2. Train on unoccluded examples with SVM.
   ○ Scale each example to match the filter size.
   ○ Random subwindows of negative images give negative examples.
3. Find the best filter placement in the HOG pyramid for each training image.
   ○ Un-scaled training images.
   ○ At least 50% overlap.
   ○ Same negatives as before.
   ○ Iterate twice.
Step 3: Initialize Part Filters

- Train latent SVM on the full model:
  - $\beta = (F_0, F_1, ..., F_6, a_1, b_1, ..., a_6, b_6)$ are model parameters.

Step 3: Train Object Model

SVM Objective: 

$$\beta^*(D) = \arg\min_{\beta} \lambda \|eta\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

Labeled training data \((x_i, y_i)\).

Score of placement \(z\).

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$
Step 3 Summary

1. Initialize 6 parts.
   - Position in areas of highest energy of root filter.

2. Train latent SVM on the full model:
   - $\beta = (F_0, F_1, ..., F_6, a_1, b_1, ..., a_6, b_6)$ are model parameters.
   - For each positive example, find best overall placement $z$.
   - Use high-scoring regions in negative images as hard negatives.
   - Iterate 10 times. Each time cache as many hard negatives as can fit into memory.
     - Remove no-longer hard negatives.

SVM Objective:

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

Labeled training data $(x_i, y_i)$.

Score of placement $z$.

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

Results

● Decent performance:
  ○ PASCAL 2007 challenge.
  ○ First place in 10/20 classes.
  ○ Second place in 6/20.

● Fast:
  ○ 3-4 hours training.
  ○ ~2s evaluation.


Car Model
Component Analysis

Best overall results with all three components.
Conclusion

- HOG pyramid representation.
- Root filter + part filters + latent placement variables.
  - Train with latent SVM.
- Hard negative mining.
- **Possible extensions?**
  - Deeper part hierarchies (parts of parts).
  - Multiple viewpoint models (front, side, back, etc.).
  - 3D pose estimation.
  - Visual words for parts: multi-class detection.

Object Hypothesis Computation
