Hierarchical Learning for Object Detection

Long Zhu, Yuanhao Chen, William Freeman, Alan Yuille, Antonio Torralba
MIT and UCLA, 2010
Background I: Our prior work

- Our work for the Pascal Challenge is based on two recent publications from our group.
Background II: Related Work

• We build on previous work:
  • Learning part-based **structure** model
    – UoCTTI (Felzenszwalb PAMI 2009), Berkeley (Bourdev ICCV 2009), Caltech (ECCV 2008), Blaschko (ECCV 2008)
  • Learning **appearance** features
    – Oxford (Vedaldi ICCV 2009, Bosch CIVR 007)
  • Learning to include **contextual** information
    – UCI (Desai ICCV 2009), MIT (Choi et al. 2010), INRIA (Harzallah et al. ICCV 2009).
Overview of our approach

1. **Hierarchical part-based models** with three layers. 4-6 models for each object to allow for pose.

2. **Energy potential terms:** (a) HOGs for edges, (b) Histogram of Words (HOWs) for regional appearance, (c) shape features.

3. **Detect objects** by scanning sub-windows using dynamic programming (to detect positions of the parts).

4. **Learn the parameters** of the models by machine learning: a variant (iCCCP) of Latent SVM.
Hierarchical Part-Based Models: (1)

- Each hierarchy is a 3-layer tree.
- Each node represents a part.
- Total of 46 nodes: \((1+9+ 4 \times 9)\)

- Each node has a spatial position (parts can “move” or are “active”)
- Graph edges from parents to child – impose spatial constraints.
Hierarchical Part-Based Models: (2)

- The parts can move relative to each other. This allows the model to have spatial deformations.
- Constraints on these deformations are imposed by edges between parents and child (will be learnt)

Parent-Child spatial constraints  Parts: blue (1), yellow (9), purple (36)
Hierarchical Part-Based Models: (3)

• Each object is represented by 4 or 6 hierarchical models (mixture of models).
• These mixture components account for pose/viewpoint changes.
Hierarchical Part-Based Models: (4)

• The object model has variables:
  1. $p$ – represents the position of the parts.
  2. $V$ – specifies which mixture component (e.g. pose).
  3. $y$ – specifies whether the object is present or not.
  4. $\omega$ – model parameter (to be learnt).

• Note: during learning the part positions $p$ and the pose $V$ are unknown – so they are latent variables and will be expressed as $h = (V, p)$
Energy of the Model:

• The “energy” of the model is defined to be:
$$-\omega \cdot \Phi(x, y, h)$$ where $$x$$ is the image in the region.

• The object is detected by solving:
$$y^*, h^* = \text{arg max } \omega \cdot \Phi(x, y, h)$$

• If $$y^* = +1$$ then we have detected the object.

• If so, $$h^* = (p^*, V^*)$$ specifies the mixture component and the positions of the parts.
Energy of the Model:

• There are three types of potential terms \( \Phi(x, y, h) \)

  (1) Spatial terms \( \Phi_{\text{shape}}(y, h) \) which specify the distribution on the positions of the parts.

  (2) Data terms for the edges of the object
  \( \Phi_{\text{HOG}}(x, y, h) \) defined using HOG features.

  (3) Regional appearance data terms
  \( \Phi_{\text{HOW}}(x, y, h) \) defined by histograms of words (HOWs – using grey SIFT features and K-means).
Energy of the Model: HOGs and HOWs

- Edge-like: Histogram of Oriented Gradients (Upper row)
- Regional: Histogram Of Words (Bottom row)
- Dense sampling: 13950 HOGs + 27600 HOWs
Object Detection

• To detect an object requiring solving:

\[ y^*, h^* = \arg \max \omega \cdot \Phi(x, y, h) \]

for each image region.

• We solve this by scanning over the subwindows of the image, use dynamic programming to estimate the part positions \( p \) and do exhaustive search over the \( y & V \)
Learning by Latent SVM

• The input to learning is a set of labeled image regions. \( \{(x_i, y_i) : i = 1, ..., N\} \)

• Learning require us to estimate the parameters \( \omega \)

• While simultaneously estimating the hidden variables \( h = (p, V) \)
Latent SVM Learning

• We use Yu and Joachim’s (2009) formulation of latent SVM.

• This specifies a non-convex criterion to be minimized. This can be re-expressed in terms of a convex plus a concave part.

\[
\min_w \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \left[ \max_{y,h} [w \cdot \Phi(x_i, y, h) + L(y_i, y, h)] - \max_h [w \cdot \Phi(x_i, y_i, h)] \right]
\]

\[
\iff \min_w \left[ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \max_{y,h} [w \cdot \Phi(x_i, y, h) + L(y_i, y, h)] \right]
\]

\[
- \left[ C \sum_{i=1}^{N} \max_h [w \cdot \Phi(x_i, y_i, h)] \right]
\]
Latent SVM Learning

• Yu and Joachims (2009) propose the CCCP algorithm (Yuille and Rangarajan 2001) to minimize this criterion.

• This iterates between estimating the hidden variables and the parameters (like the EM algorithm).

• We propose a variant – incremental CCCP – which is faster.

• Result: our method works well for learning the parameters without complex initialization.
Learning Algorithm: Incremental CCCP

• Iterative Algorithm:
  – Step 1: fill in the latent positions with best score (DP)
  – Step 2: solve the structural SVM problem using partial negative training set (incrementally enlarge).

• Initialization:
  – No pretraining (no clustering).
  – No displacement of all nodes (no deformation).
  – Pose assignment: maximum overlapping

• Simultaneous multi-layer learning
Kernels

• We use a quasi-linear kernel for the HOW features, linear kernels of the HOGs and for the spatial terms.
• We use:
  (i) equal weights for HOGs and HOWs
  (ii) equal weights for all nodes at all layers
  (iii) same weights for all object categories.
• Note: tuning the weights for different categories may improve the performance.
Post-processing: Context Modeling

- **Post-processing:**
  - Rescoring the detection results

- **Context modeling:** SVM+ contextual features
  - best detection scores of 20 classes, locations, recognition scores of 20 classes

- **Recognition scores** (Lazebnik CVPR06, Van de Sande PAMI 2010, Bosch CIVR07)
  - SVM + spatial pyramid + HOWs (no latent position variable)
Experiments on Pascal 2010

- 4 or 6 mixture components/poses.
- All other parameter settings (C, the relative weights of appearance features, the number of visual words, etc.) are identical for all categories.
- 300 visual words: one round of K-means.
Detection Results on PASCAL 2010: Cat
Horse
Car
Bus
Comparisons on PASCAL 2010

- Mean Average Precision (mAP) is reported.
- Note: the calculations of AP used in 2010 and 2009 are different.

<table>
<thead>
<tr>
<th>Methods (trained on 2010)</th>
<th>MIT-UCLA</th>
<th>NLPR</th>
<th>NUS</th>
<th>UoCTTI</th>
<th>UVA</th>
<th>UCI incomplete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test on 2010</td>
<td>35.99</td>
<td>36.79</td>
<td>34.18</td>
<td>33.75</td>
<td>32.87</td>
<td>32.52</td>
</tr>
<tr>
<td>Test on 2009</td>
<td>36.72</td>
<td>37.65</td>
<td>35.53</td>
<td>34.57</td>
<td>34.47</td>
<td>33.63</td>
</tr>
</tbody>
</table>
Conclusion

• Objects are represented by mixture of Hierarchical Models of parts.
• The energy for the model contains spatial terms, edge-like terms (HOGs), and regional appearance terms (HOWs).
• We learn the model parameter by a variant of latent SVM -- incremental CCCP – which only requires simple initialization.
• The code will be available soon.
• Current and future work
  – Increase the number of components/poses
  – Part sharing