Layered Object Detection for Multi-Class Image Segmentation

Yi Yang  Sam Hallman  Deva Ramanan  Charless Fowlkes

UC Irvine
Introduction

• PASCAL competition
• 20 object categories + “background”
Introduction
Introduction
Introduction

* We use part-based detectors from Felzenszwalb, Gorelick, McAllester, & Ramanan PAMI 09
Layered Representation

- Model each detection in “2.1D”: detections ordered by depth
Related work

• Related work combining segmentation and recognition
  – Biasing segmentation output based on object models [Yu et al. 03, Ramanan 06, Kumar et al. 05]
  – Iterating between bottom-up and top-down cues [Leibe et al. 04, Tu et al. 05]

• Related work on layers
  – Work in the video domain [Wang & Adelson 94, Jojic & Frey 01, Kumar et al. 04]
  – Some work in image domain [Gao et al. 07, Nitzberg et al. 93]
Issues

- Need detector thresholds
- Need comparable confidence scores
Detector calibration

Find optimal threshold for pixel labeling
Model

\( N = \# \text{ of detections} \)
\( x_i = \text{RGB values for pixel } i \)
\( z_i = \text{label for pixel } i \ (1..N \text{ or 0 for background}) \)
\( \theta_n = \text{color model for } n\text{th layer} \)
\( d_\pi = \text{set of detections with ordering } \pi \)

\[
P(z, x | \theta, d_\pi) = \prod_i P(z_i | d_\pi) P(x_i | \theta_{z_i})
\]
Inference: coordinate descent

$$P(z, x|\theta, d_\pi) = \prod_i P(z_i|d_\pi) P(x_i|\theta z_i)$$

$$f(z, \theta) = -\log P(z, x|\theta, d_\pi)$$

Step 1: \( \arg \min_z f(z, \theta) \) (per-pixel segmentation given shape prior & color likelihood)

Step 2: \( \arg \min_\theta f(z, \theta) \) (fit color models to segments and background)

Color models are learned on-the-fly for each detection instance (e.g., people may wear blue or red shirts)
Bottom-up segmentation

Use hierarchical segmentation of Arbelaez, Maire, Fowlkes, Malik (2009) at a threshold which generates ~200 segments per image.
Inference: coordinate descent

\[
P(z, x \mid \theta, d_\pi) = \prod_{i} P(z_i \mid d_\pi) P(x_i \mid \theta_{z_i})
\]

\[
f(z, \theta) = -\log P(z, x \mid \theta, d_\pi)
\]

Step 1: \(\arg\min_{z \in \mathcal{Z}} f(z, \theta)\) (per-pixel segmentation given shape prior & color likelihood)

Step 2: \(\arg\min_{\theta} f(z, \theta)\) (fit color models to segments and background)

\(\mathcal{Z} = \text{set of pixel labelings consistent with superpixel map}\)
Building $P(z_i | d_π)$

- Person detections
- True positives
- Ground truth segmentations
- Shape prior for the *person* class
Shape priors (cont’d)
Shape priors (cont’ d)
Shape priors (cont’d)
Bicycle part-based priors
Motorcycle part-based priors
Horse part-based prior
Bottle part-based priors
Building $P(z_i \mid d_\pi)$
Building $P(z_i | d_\pi)$

[ 1, 0, 0, 0, 0, 0, 0, 0 ]

pixel $i$
Building $P(z_i|d_\pi)$

\[
\begin{bmatrix}
1, 0, 0, 0, 0, 0 \\
0.1, 0.9, 0, 0, 0, 0
\end{bmatrix}
\]
Building $P(z_i | d_\pi)$

\[
\begin{bmatrix}
1, 0, 0, 0, 0, 0, 0 \\
0.1, 0.9, 0, 0, 0, 0 \\
0.099, 0.891, 0.01, 0, 0, 0 \\
\end{bmatrix}
\]
Building $P(z_i | d_\pi)$
Building $P(z_i | d_\pi)$
Building $P(z_i | d_\pi)$
Building $P(z_i \mid d_\pi)$
Building $P(z_i | d_\pi)$
Building $P(z_i | d_\pi)$
Building $P(z_i | d_\pi)$
Building $P(z_i \mid d_\pi)$
The algorithm

for each ordering $\pi$:

find $\arg \max_z P(z, x|\theta, d_\pi)$

output superpixel labels with most probable $\pi$

• There is no secret method for finding $\pi$...
  Just try all of them!

• Luckily, the number of permutations to test is usually small.
Results!
PASCAL 2009 Segmentation Challenge

<table>
<thead>
<tr>
<th>Objects</th>
<th>Mean</th>
<th>Max</th>
<th>Us</th>
<th>Our rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>background</td>
<td>41.2</td>
<td>83.5</td>
<td>78.0</td>
<td>8</td>
</tr>
<tr>
<td>aeroplane</td>
<td>18.8</td>
<td>56.3</td>
<td>32.8</td>
<td>7</td>
</tr>
<tr>
<td>bicycle</td>
<td>10.4</td>
<td>26.6</td>
<td>29.4</td>
<td>1</td>
</tr>
<tr>
<td>bird</td>
<td>11.0</td>
<td>40.6</td>
<td>3.2</td>
<td>17</td>
</tr>
<tr>
<td>boat</td>
<td>11.5</td>
<td>36.1</td>
<td>5.0</td>
<td>16</td>
</tr>
<tr>
<td>bottle</td>
<td>18.2</td>
<td>46.1</td>
<td>33.1</td>
<td>3</td>
</tr>
<tr>
<td>bus</td>
<td>25.5</td>
<td>50.5</td>
<td>43.4</td>
<td>3</td>
</tr>
<tr>
<td>car</td>
<td>20.6</td>
<td>42.3</td>
<td>43.8</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>12.6</td>
<td>35.3</td>
<td>8.3</td>
<td>12</td>
</tr>
<tr>
<td>chair</td>
<td>4.2</td>
<td>9.1</td>
<td>5.1</td>
<td>9</td>
</tr>
<tr>
<td>cow</td>
<td>11.7</td>
<td>33.1</td>
<td>11.9</td>
<td>9</td>
</tr>
<tr>
<td>diningtable</td>
<td>9.1</td>
<td>27.0</td>
<td>8.2</td>
<td>11</td>
</tr>
<tr>
<td>dog</td>
<td>9.1</td>
<td>24.5</td>
<td>5.6</td>
<td>14</td>
</tr>
<tr>
<td>horse</td>
<td>17.5</td>
<td>42.7</td>
<td>21.0</td>
<td>7</td>
</tr>
<tr>
<td>motorbike</td>
<td>23.4</td>
<td>56.4</td>
<td>24.4</td>
<td>9</td>
</tr>
<tr>
<td>person</td>
<td>20.9</td>
<td>37.5</td>
<td>38.6</td>
<td>1</td>
</tr>
<tr>
<td>pottedplant</td>
<td>9.7</td>
<td>37.1</td>
<td>14.6</td>
<td>6</td>
</tr>
<tr>
<td>sheep</td>
<td>19.7</td>
<td>43.6</td>
<td>14.8</td>
<td>13</td>
</tr>
<tr>
<td>sofa</td>
<td>8.5</td>
<td>21.9</td>
<td>3.5</td>
<td>17</td>
</tr>
<tr>
<td>train</td>
<td>19.2</td>
<td>41.0</td>
<td>27.5</td>
<td>7</td>
</tr>
<tr>
<td>tv/monitor</td>
<td>22.3</td>
<td>47.8</td>
<td>45.7</td>
<td>2</td>
</tr>
<tr>
<td>average</td>
<td>16.4</td>
<td>36.2</td>
<td>23.7</td>
<td>7</td>
</tr>
</tbody>
</table>

Performs well where detector works well (objects with well defined, fairly rigid shapes)
<table>
<thead>
<tr>
<th>Category</th>
<th>ordering</th>
<th>color</th>
<th>superpixel</th>
<th>parts</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>background</td>
<td>79.37</td>
<td>78.93</td>
<td>78.65</td>
<td>79.62</td>
<td>79.36</td>
</tr>
<tr>
<td>aeroplane</td>
<td>35.26</td>
<td>32.39</td>
<td>30.61</td>
<td>37.22</td>
<td>35.26</td>
</tr>
<tr>
<td>bicycle</td>
<td>25.46</td>
<td>23.12</td>
<td>20.7</td>
<td>24.58</td>
<td>25.45</td>
</tr>
<tr>
<td>bird</td>
<td>2.81</td>
<td>2.78</td>
<td>2.68</td>
<td>2.79</td>
<td>2.81</td>
</tr>
<tr>
<td>boat</td>
<td>9.87</td>
<td>9.16</td>
<td>9.64</td>
<td>9.14</td>
<td>9.87</td>
</tr>
<tr>
<td>bottle</td>
<td>41.44</td>
<td>39.73</td>
<td><strong>41.76</strong></td>
<td>40.19</td>
<td>41.29</td>
</tr>
<tr>
<td>bus</td>
<td>49.83</td>
<td>48.52</td>
<td>48.54</td>
<td>48.72</td>
<td>49.87</td>
</tr>
<tr>
<td>car</td>
<td>46.88</td>
<td>45.66</td>
<td>44.25</td>
<td>46.14</td>
<td>47.03</td>
</tr>
<tr>
<td>cat</td>
<td>18.4</td>
<td>17.68</td>
<td>16.81</td>
<td>15.06</td>
<td>18.4</td>
</tr>
<tr>
<td>chair</td>
<td>10.05</td>
<td>9.06</td>
<td>9.57</td>
<td>8.37</td>
<td>10</td>
</tr>
<tr>
<td>cow</td>
<td>17.74</td>
<td>16.83</td>
<td><strong>18.1</strong></td>
<td>15.91</td>
<td>17.77</td>
</tr>
<tr>
<td>diningtable</td>
<td>6.94</td>
<td>6.8</td>
<td>6.79</td>
<td>6.85</td>
<td>7.27</td>
</tr>
<tr>
<td>dog</td>
<td>11.53</td>
<td>10.55</td>
<td>11.18</td>
<td>10.91</td>
<td><strong>11.53</strong></td>
</tr>
<tr>
<td>horse</td>
<td>16.07</td>
<td>14.6</td>
<td>15.33</td>
<td>15.19</td>
<td><strong>16.21</strong></td>
</tr>
<tr>
<td>motorbike</td>
<td>25.72</td>
<td>24.38</td>
<td>24.46</td>
<td>24.88</td>
<td>25.62</td>
</tr>
<tr>
<td>person</td>
<td>36.88</td>
<td>34.98</td>
<td>35.3</td>
<td>32.4</td>
<td>36.81</td>
</tr>
<tr>
<td>pottedplant</td>
<td>15.55</td>
<td>14.92</td>
<td>15.17</td>
<td>14.32</td>
<td><strong>15.55</strong></td>
</tr>
<tr>
<td>sheep</td>
<td>21.09</td>
<td>18.77</td>
<td>20.33</td>
<td>17.77</td>
<td><strong>21.1</strong></td>
</tr>
<tr>
<td>sofa</td>
<td>12.63</td>
<td>12.2</td>
<td>12.14</td>
<td>12.05</td>
<td><strong>12.63</strong></td>
</tr>
<tr>
<td>train</td>
<td>28.6</td>
<td>27.43</td>
<td>27.88</td>
<td>27.86</td>
<td><strong>28.6</strong></td>
</tr>
<tr>
<td>tvmonitor</td>
<td><strong>46.41</strong></td>
<td>46.01</td>
<td>46.36</td>
<td>43.67</td>
<td>46.28</td>
</tr>
<tr>
<td>average</td>
<td><strong>26.6</strong></td>
<td>25.45</td>
<td><strong>25.53</strong></td>
<td>25.41</td>
<td><strong>26.6</strong></td>
</tr>
</tbody>
</table>

Superpixels really help bikes.

Color helps person segmentation.

Parts provide small but noticeable improvement.
Does ordering help?

• Default ordering based on detector score works fairly well (after calibration)

• PASCAL segmentation benchmark limitations:
  – images often only have a single object
  – scoring is per-class rather than per-instance
    • Ordering between same class detections doesn’t affect benchmark score

• We found ordering helped on a subset of PASCAL with overlapping objects
Conclusions

• A simple layered model for compiling multi-object detections into segmentations
  – Deformable shape prior built on part-based detector
  – Per-instance color model
• Provides a globally consistent 2.1D interpretation
• Good performance on PASCAL segmentation benchmark
Thanks!