Computationally Efficient Regression on a Dependency Graph for Human Pose Estimation

Kota Hara and Rama Chellappa
Center for Automation Research, University of Maryland, College Park, MD 20742
{kotahara, rama}@umiacs.umd.edu

Overview

- **Input**: single still image + upper body bounding box
- **Output**: positions of the body joints
- **Decompose a complex human pose estimation task into a set of local pose estimation tasks by introducing a dependency graph**
- **Address each local pose estimation task by multidimensional output regression**

Naïve Regression-based Approach

Train a single regressor which maps the image features to all the joint positions at once

- **Output dimensionality is too high (12 joints × 2 = 24)**
- **Size of the image patch which can cover all possible poses is large. As a result, the patch is dominated by the background, which makes regression very hard**

Multidimensional Output Boosted Regression Trees

**Regression Trees Training**: Recursively split the training data such that each subset of the training data becomes as coherent as possible in terms of the output

**Boosting Regression Trees**: Sequentially add regression trees to gradually reduce the empirical loss

Results

- **First non-pictorial structure method that achieves state-of-the-art results on Buffy Stickmen and PASCAL stickmen datasets [5]**
- **Computationally efficient (23 m sec/image with Boosted Regression Trees)**
- **Boosted Regression Trees provide a good balance between accuracy and speed**

Dependence graph

- Each node corresponds to one of the joint positions
- Root node corresponds to the center of the upper body bounding box (given by the upper body detector)
- Position of a child node is dependent on the position of its parent node and local image information

Training

- For each non-leaf node, a regressor is trained to estimate the relative positions of its child nodes
- **Regressor input**: HOG features extracted from a local patch around the position of the node

Testing

- Extract HOG features from a local patch around the center of the given upper body bounding box
- **Estimate the positions of 2, 3, 4, 5, 6 and 10 using the corresponding regressor**
- Extract HOG features from a local patch around the estimated position of 6 and then estimate the positions of 7 and 8 using the corresponding regressor
- Do the same for the other nodes until all the joint positions are estimated

Example results from PASCAL

Example results from Buffy

Table 1: PCCs on Buffy

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>P</th>
<th>R</th>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoDG-Boost</td>
<td>0.94</td>
<td>0.96</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>HOG-FLS</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>HOG-MVMU</td>
<td>0.94</td>
<td>0.96</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>RoDG-LSASSO</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 2: PCCs on PASCAL

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>P</th>
<th>R</th>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoDG-Boost</td>
<td>0.94</td>
<td>0.96</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>HOG-FLS</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>HOG-MVMU</td>
<td>0.94</td>
<td>0.96</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>RoDG-LSASSO</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
</tr>
</tbody>
</table>

References


Example results from Buffy Stickmen dataset [5]