Local Decorrelation for Improved Pedestrian Detection

Overview
- **PROBLEM:** heavily correlated features; orthogonal (axis-aligned) trees generalize poorly
- **OBSERVATION:** oblique trees generalize well on correlated features; however, they are slow
- **IDEA:** replace the oblique trees by orthogonal trees over *locally decorrelated* features

Boosted Trees + Correlated Data

Orthogonal/Oblique Trees on Correlated Data:
- (a) Original
- (b) Orthogonal
- (c) Oblique
- (d) LDA
- (e) PCA-whitened
- (f) ZCA-whitened

Orthogonal Trees on Transformed Data:
- (a) Original
- (b) Decorrelated
- (c) PCA-whitened
- (d) ZCA-whitened

Baseline Detector (ACF)
- compute channels
- aggregate
- vectorize
- apply boosted trees

Detection + Oblique Splits (ACF-LDA)
- channels
- 5x5 patches
- vectorize
- LDA
- apply boosted trees

Results

Detection + Oblique Splits (ACF-LDA)

Baseline Detector (ACF)
- boosted trees over aggregated channel features
- 10 Channels: LUV (3), grad magnitude (1), oriented grad (6)
- state-of-the-art detection performance; real time (30 fps)

Alternate Filtering Strategies

<table>
<thead>
<tr>
<th>Description</th>
<th># Channels</th>
<th>Miss Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ACF</td>
<td></td>
<td>17.3 ± .33</td>
</tr>
<tr>
<td>2. LDCAF small λ, k</td>
<td>10k</td>
<td>19.1 ± .32</td>
</tr>
<tr>
<td>3. LDCAF random, w, k</td>
<td>10k</td>
<td>19.6 ± .34</td>
</tr>
<tr>
<td>4. LDCAF LUV, 3k + 7</td>
<td>16.2 ± .37</td>
<td></td>
</tr>
<tr>
<td>5. LDCAF grad, 3 + 7k</td>
<td>14.9 ± .29</td>
<td></td>
</tr>
<tr>
<td>6. LDCAF constant, 3 + 7k</td>
<td>14.2 ± .34</td>
<td></td>
</tr>
<tr>
<td>7. LDCAF proposed approach</td>
<td>13.7 ± .35</td>
<td></td>
</tr>
</tbody>
</table>

Learning Decorrelation Filters
- training data (unsupervised)
- autoregression
- eigenvectors
- decorrelation filters

Locally Decorrelated Channel Features (LDCF)

Baseline Detector (ACF)
- (a) Orthogonal
- (b) Orthogonal
- (c) Orthogonal
- (d) Oblique
- (e) LDA
- (f) PCA-whitened
- (g) ZCA-whitened

Baseline Detector (ACF)
- compute channels
- aggregate
- vectorize
- apply boosted trees

Baseline Detector (ACF)
- boosted trees over aggregated channel features
- 10 Channels: LUV (3), grad magnitude (1), oriented grad (6)
- state-of-the-art detection performance; real time (30 fps)

Alternate Filtering Strategies

<table>
<thead>
<tr>
<th>Description</th>
<th># Channels</th>
<th>Miss Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ACF</td>
<td></td>
<td>17.3 ± .33</td>
</tr>
<tr>
<td>2. LDCAF small λ, k</td>
<td>10k</td>
<td>19.1 ± .32</td>
</tr>
<tr>
<td>3. LDCAF random, w, k</td>
<td>10k</td>
<td>19.6 ± .34</td>
</tr>
<tr>
<td>4. LDCAF LUV, 3k + 7</td>
<td>16.2 ± .37</td>
<td></td>
</tr>
<tr>
<td>5. LDCAF grad, 3 + 7k</td>
<td>14.9 ± .29</td>
<td></td>
</tr>
<tr>
<td>6. LDCAF constant, 3 + 7k</td>
<td>14.2 ± .34</td>
<td></td>
</tr>
<tr>
<td>7. LDCAF proposed approach</td>
<td>13.7 ± .35</td>
<td></td>
</tr>
</tbody>
</table>

Summary
- **ACURATE:** 10-fold reduction in # of FPs
- **FAST:** few minutes for learning filters
- **GENERAL:** applicable to any channel type
- **TRAINABLE:** no extra supervision required
- **CODE:** https://github.com/pdollard/toolbox/