Experiments on Densely Sampling

Xin Sui
Three random sampling methods

Non-Overlapping

Overlapping

Random-Overlapping
Process Flow

Tools
1. Compute_Descriptor.In: Given region, compute the sift descriptor http://www.robots.ox.ac.uk/
2. LibSvm
   a convenient SVM tool
   http://www.csie.ntu.edu.tw/~cjlin/libsvm/

SVM: RBF Kernel
With gamma set to the median of the pairwise distances between the training descriptors
Sampling

- **Step 1:** Generating regions files
  - Ellipse Region
    - Format: 
      \[ a(x-u)(x-u)+2b(x-u)(y-v)+c(y-v)(y-v)=1 \]
  - Format:
    \[
    \begin{bmatrix}
    1.0 \\
    m \\
    u_1 v_1 a_1 b_1 c_1 \\
    \vdots \\
    u_m v_m a_m b_m c_m
    \end{bmatrix}
    \]
- **Step 2,** call `Compute_descriptor.in` to generate sift descriptor files
  - **Descriptor output**
    - Format: 
      \[
      \begin{bmatrix}
      N \\
      m \\
      u_1 v_1 a_1 b_1 c_1 d_{1,1} d_{1,2} \ldots d_{1,N} \\
      \vdots \\
      u_m v_m a_m b_m c_m d_{m,1} d_{m,2} \ldots d_{m,N}
      \end{bmatrix}
      \]
Dataset

- Pascal VOC 2005
  - Four Categories
  - Binary classification

<table>
<thead>
<tr>
<th></th>
<th>Motorbikes</th>
<th>Bicycles</th>
<th>People</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>214</td>
<td>114</td>
<td>84</td>
<td>272</td>
</tr>
<tr>
<td>Test(test 1)</td>
<td>216</td>
<td>114</td>
<td>84</td>
<td>275</td>
</tr>
</tbody>
</table>
## Results (200 words)

- **Accuracy (correct predictions/the total test examples)**

<table>
<thead>
<tr>
<th></th>
<th>Motorbikes</th>
<th>Bicycles</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Overlap</td>
<td>82.148% (566/689)</td>
<td>85.0508% (586/689)</td>
<td>80.4064% (554/689)</td>
</tr>
<tr>
<td>Overlap-rand</td>
<td>79.6807% (549/689)</td>
<td>Predict negative for almost all images</td>
<td>78.2293% (539/689)</td>
</tr>
<tr>
<td>Overlap-big</td>
<td>87.8084% (605/689)</td>
<td>88.5341% (610/689)</td>
<td>84.0348% (579/689)</td>
</tr>
</tbody>
</table>
ROC curves_motorbike

Random-Overlap

Overlap-big

Non-overlap
ROC curves_cars

Random-Overlap

Non-overlap

Overlap-big
ROC curves_bikes

Overlap_rand: predicting negative for most of all test images

![Non-overlap](image1)

![Overlap-big](image2)
Overlap_rand: predicting negative for most of all test images
Non-Overlap: predicting negative for most of all test images
Conclusion

• According to the above experiment, it seems:
  – Uniformly distributed patches with overlapping seems better
  – Uniformly distributed patches are better than random distributed patches