Improving Image Similarity With Vectors of Locally Aggregated Tensors (VLAT)

David Picard and Philippe-Henri Gosselin

picard@ensea.fr, gosselin@ensea.fr
ETIS / ENSEA - University of Cergy-Pontoise - CNRS UMR 8051
F-95000 Cergy-Pontoise Cedex, France

12 September 2011
Visual Object Classification

Database

Classifier

Training

Training set
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
Descriptors

Bags of Descriptors

1 Extraction of Zones of Interest
   ⇒ Regions,
   ⇒ Keypoints,
   ⇒ ...

2 Description of Zones of Interest
   ⇒ Colors, Textures
   ⇒ HoG, SIFT,
   ⇒ ...

3 Bag of Descriptors $B_i = \{b_{ri}\}_r$
   ⇒ $b_{ri} \in D$ : descriptor $r$ of image $i$

Bag $B = \{b_r\} \in B$
Vector representations using codebook

1. Build a code book
   - K-Means,
   - Gaussian Mixture Models,
   - ...

2. Project Bags of Descriptors on the codebook
   - Hard assignment
   - Soft assignment
   - ...

3. Vector representation $\mathbf{w}_i \in \mathbb{R}^m$, with $m$ words
   - $w_{ri} \in \mathbb{R}$ : weight of word $r$ of image $i$
Vector representations using codebook

Advantages

- Small representation
  - Low memory consumption
  - Fast similarities
- Compatible with usual classifiers (kNN, SVM, ...)

Drawbacks

- Performance depends on the codebook
- A lot of words are required (from 1k to 100k)
- Codebook computation in high dimensional spaces is difficult and unstable
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
Kernels on Bags

Change initial space

- Work directly on Bags of Descriptors $B_i = \{b_{ri}\}_r$, $b_{ri} \in D$
- Perform matching computations in a new space $\mathcal{H}$
- Use a mapping function $\phi : \mathcal{D} \rightarrow \mathcal{H}$
  - $\mathcal{D}$: visual space (where descriptors are)
  - $\mathcal{H}$: induced space (where matching is performed)
- Vector representation $\phi(b_{ri})$ in the induced space $\mathcal{H}$
  - High dimension vectors (sometimes infinite)
  - Matching is easier in large spaces
- Implicit use of induced space $\mathcal{H}$ using a kernel function:

$$k(b_{ri}, b_{sj}) = \langle \phi(b_{ri}), \phi(b_{sj}) \rangle$$
Kernels on Bags

“Soft maximum” (Shawe-Taylor)

\[ K_{\text{softmax}}(B_i, B_j) = \sum_{b_{ri} \in B_i} \sum_{b_{sj} \in B_j} k(b_{ri}, b_{sj}) \]

- \( k \) kernel function \( \Rightarrow K_{\text{softmax}} \) kernel function
- Compare barycenters of bags:

\[ \Phi(B_i) = \sum_{b_{ri} \in B_i} \phi(b_{ri}) \]

\[ K_{\text{softmax}}(B_i, B_j) = \langle \Phi(B_i), \Phi(B_j) \rangle \]

\[ = \sum_{b_{ri} \in B_i} \sum_{b_{sj} \in B_j} \langle \phi(b_{ri}), \phi(b_{sj}) \rangle \]
Kernels on Bags

Advantages
- As efficient as vote-based systems
- No large codebook
  - Stable
  - Easy parameter tuning
- Compatible with usual classifiers (kNN, SVM, ...)

Drawbacks
- High computational complexity
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
First idea: Approximation of Gaussian Kernel on Bags

- Taylor expansion:

\[ K(B_i, B_j) = \sum_{r,s} e^{-\frac{2(b_{ri}, b_{sj})}{\sigma^2}} \]

\[ = \sum_{r,s} \sum_{p} \alpha_p (\langle b_{ri}, b_{sj} \rangle)^p \]

- Vectorization with Tensors:

\[ K(B_i, B_j) = \sum_{r,s} \sum_{p} \alpha_p \langle \otimes_p b_{ri}, \otimes_p b_{sj} \rangle \]

\[ = \sum_{p} \alpha_p \langle \sum_r \otimes_p b_{ri}, \sum_s \otimes_p b_{sj} \rangle \]
Second idea: Divide and center

Inspired from Vectors of Locally Aggregated Descriptors (VLAD) method:

- Divide descriptor space in a few partition (from 16 to 64)
  - Kernels on Bags are more efficient with smaller bags
  - Clustering in high dimension spaces with few words is easy and stable
- Center descriptors in each cluster
  - Relative comparisons are more accurate (cf Fisher kernels)
- Perform a matching in each cluster
- Sum up the matching results
Vectors of Locally Aggregated Tensors

Proposed method

Descriptor computation for order $p$:

1. Compute a small codebook and get centers $c_n$

2. For each image $i$
   - Compute tensor for each center $c_n$

$$\mathcal{T}_n^p(B_i) = \sum_{b_{ri}\text{ such that }NN(b_{ri})=c_n} \otimes_p (b_{ri} - c_n)$$

3. Concatenate: $\mathcal{T}^p(B_i) = (\mathcal{T}_1^p(B_i) \ldots \mathcal{T}_m^p(B_i))$

4. Normalize: $\hat{\mathcal{T}}^p(B_i) = \mathcal{T}^p(B_i) / \|\mathcal{T}^p(B_i)\|_{L_2}$
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
# Image classification experiments

## Visual Object Classes Challenge 2007

- Pascal Network Network of Excellence
- 9,963 images
- 20 categories
- SIFT descriptors
Image classification experiments

Evaluation Protocol

- SVM Classifier
- Triangle kernel with $L^2$ distance
- Training on official train+val sets (5011 images)
- Testing on official test set (4952 images)
- Performance evaluation with official tool (Average Precision)
## Experiments

<table>
<thead>
<tr>
<th>Category</th>
<th>BOF</th>
<th>VLAD</th>
<th>VLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>4k</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>aeroplane</td>
<td>65.3</td>
<td>60.2</td>
<td>61.0</td>
</tr>
<tr>
<td>bicycle</td>
<td>44.5</td>
<td>35.3</td>
<td>38.6</td>
</tr>
<tr>
<td>bird</td>
<td>38.7</td>
<td>33.7</td>
<td>34.6</td>
</tr>
<tr>
<td>boat</td>
<td>49.4</td>
<td>53.7</td>
<td>55.8</td>
</tr>
<tr>
<td>bottle</td>
<td>18.7</td>
<td>13.9</td>
<td>16.0</td>
</tr>
<tr>
<td>bus</td>
<td>37.1</td>
<td>41.0</td>
<td>44.8</td>
</tr>
<tr>
<td>car</td>
<td>63.3</td>
<td>65.8</td>
<td>68.3</td>
</tr>
<tr>
<td>cat</td>
<td>33.9</td>
<td>35.2</td>
<td>37.9</td>
</tr>
<tr>
<td>chair</td>
<td>39.0</td>
<td>38.2</td>
<td>40.2</td>
</tr>
<tr>
<td>cow</td>
<td>24.8</td>
<td>22.7</td>
<td>22.7</td>
</tr>
<tr>
<td>diningtable</td>
<td>22.3</td>
<td>23.3</td>
<td>25.9</td>
</tr>
<tr>
<td>dog</td>
<td>26.9</td>
<td>34.1</td>
<td>32.4</td>
</tr>
<tr>
<td>horse</td>
<td>58.4</td>
<td>65.8</td>
<td>65.2</td>
</tr>
<tr>
<td>motorbike</td>
<td>36.2</td>
<td>45.2</td>
<td>47.2</td>
</tr>
<tr>
<td>person</td>
<td>75.6</td>
<td>75.6</td>
<td>77.4</td>
</tr>
<tr>
<td>pottedplant</td>
<td>11.9</td>
<td>11.9</td>
<td>14.4</td>
</tr>
<tr>
<td>sheep</td>
<td>24.2</td>
<td>20.2</td>
<td>26.1</td>
</tr>
<tr>
<td>sofa</td>
<td>33.9</td>
<td>34.3</td>
<td>36.2</td>
</tr>
<tr>
<td>train</td>
<td>57.7</td>
<td>60.7</td>
<td>64.6</td>
</tr>
<tr>
<td>tvmonitor</td>
<td>39.8</td>
<td>37.3</td>
<td>34.9</td>
</tr>
<tr>
<td><strong>all</strong></td>
<td>40.1</td>
<td>40.4</td>
<td>42.2</td>
</tr>
</tbody>
</table>
### Image search experiments

#### Holidays database

<table>
<thead>
<tr>
<th>linear metric</th>
<th>BoW</th>
<th>HE</th>
<th>VLAD</th>
<th>VLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>256</td>
<td></td>
<td></td>
<td>54.9</td>
<td>65.1</td>
</tr>
<tr>
<td>1024</td>
<td></td>
<td></td>
<td>60.1</td>
<td>72.2</td>
</tr>
<tr>
<td>2048</td>
<td></td>
<td></td>
<td>62.7</td>
<td>75.1</td>
</tr>
<tr>
<td>4096</td>
<td></td>
<td></td>
<td>63.6</td>
<td>76.2</td>
</tr>
<tr>
<td>20.000</td>
<td>40.4</td>
<td></td>
<td>65.1</td>
<td>75.5</td>
</tr>
</tbody>
</table>

BoW = Bags of Words
HE = Hamming Embedding
VLAD = Vectors of Locally Aggregated Vectors
VLAT = Vectors of Locally Aggregated Tensors (proposed)
Plan

1. Vector representations using codebooks
2. Kernels on Bags
3. Vectors of Locally Aggregated Tensors
4. Experiments
5. Conclusion
Vectors of Locally Aggregated Tensors (VLAT)

- Vectorization of Kernels on Bags
  - Faster than Kernels on Bags
  - Easier tuning than Bag of Words
- State of the art performance
  - Object classification
  - Image search
- New framework, many more to come!