Aggregating Deep Convolutional Features for Image Retrieval

A.Babenko, V.Lempitsky
ICCV 2015

Presenter: Taewook Ha
Contents

- Introduction
- Background
- Main approach
- Experiments
- References
Introduction

- Local deep convolutional features describe particular image regions

- Aggregating local features produces a global image descriptor for image retrieval
  - Some existing aggregation methods for SIFT features are used
    - i.e. VLAD [1], Fisher vector [2], Triangular embedding [3]
  - Need simple aggregation method for deep convolutional features

Descriptor aggregation

- An image $I$ is represented by a set of features $\{x_1, x_2, ..., x_n\} \subset \mathbb{R}^d$
- The goal is to combine features into global representation $\psi(I)$
- The common way to produce a representation $\psi(I)$ includes two steps: embedding and aggregation
  - Embedding maps individual feature $x$ into higher dimensional vector $\phi(x)$
  - Aggregation integrates vectors $\phi(x)$ into global representation $\psi(I)$

Ex) VLAD

- Precompute a codebook with $K$ centroids $\{c_1, ..., c_K\}$
- Map $x$ to vector $\phi_{VL}(x) = [0 \ 0 \ ... \ (x - c_k) \ ... \ 0]$ with $k$ closest centroids

- i.e. Simple summation $\psi(I) = \sum \phi(x)$
Background

- Descriptor aggregation
  - Ex) VLAD embedding and aggregation for SIFT features

\[ V(:, k) = \sum_{i=1}^{N} a_k(x_i)(x_i - c_k) \]

0/1 assignment of desc. \( i \) to cluster \( k \)

Sum over all \( N \) descriptors in the image

\[ V = [\vdots, \vdots, \vdots, \vdots, \ldots] \]

Background

- Deep descriptors for retrieval
  - Instead of traditional SIFT-like features, use deep convolutional features to aggregation methods
  - Ex) VLAD-embedding for deep convolutional features [4]

Main approach

- SPoC descriptor
  - Based on the aggregation of deep convolutional features **without embedding**
  - Each deep convolutional feature $f$ is computed from image $I$ with the spatial coordinates $(x,y)$
Main approach

- SPoC descriptor
  - Sum pooling

  \[
  \psi_1(I) = \sum_{y=1}^{H} \sum_{x=1}^{W} f(x,y)
  \]

- Similarity measure
  - Similarity measure uses scalar product of two descriptors

  \[
  \text{sim}(I_1, I_2) = \langle \psi(I_1), \psi(I_2) \rangle = \sum_{f_i \in I_1} \sum_{f_j \in I_2} \langle f_i, f_j \rangle
  \]
Main approach

- **SPoC descriptor**
  - **Centering prior**
    - Assign larger weights to the features from the center of the feature map
    
    $$\psi_2(I) = \sum_{y=1}^{H} \sum_{x=1}^{W} \alpha(x,y) f(x,y)$$
  
  - **Coefficients** $\alpha(x,y)$
    - Depends on the spatial coordinates $x$ and $y$
    - Uses the Gaussian weighting scheme
    
    $$\alpha(x,y) = \exp\left\{ - \frac{(y - \frac{H}{2})^2 + (x - \frac{W}{2})^2}{2\sigma^2} \right\}$$
Main approach

- SPoC descriptor
  - **Post-processing**
    - The obtained representation $\psi(I)$ is subsequently $l_2$-normalized, then PCA compression and whitening are performed
    - PCA compression

\[
\psi_3(I) = \text{diag}(s_1, s_2, \ldots, s_N)^{-1} M_{\text{PCA}} \psi_2(I)
\]

- $l_2$-normalization

\[
\psi_{SPoC}(I) = \frac{\psi_3(I)}{\|\psi_3(I)\|_2}
\]
Experiments

- Comparison among three types of features
  - Deep convolutional features, original SIFT features, and Fisher Vector-embedded SIFT features
Experiments

- Comparison of feature aggregation methods for deep convolutional features

<table>
<thead>
<tr>
<th>Method</th>
<th>Holidays</th>
<th>Oxford5K (full)</th>
<th>Oxford105K (full)</th>
<th>UKB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher vector, k=16</td>
<td>0.704</td>
<td>0.490</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Fisher vector, k=256</td>
<td>0.672</td>
<td>0.466</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Triangulation embedding, k=1</td>
<td>0.775</td>
<td>0.539</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Triangulation embedding, k=16</td>
<td>0.732</td>
<td>0.486</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Max pooling</td>
<td>0.711</td>
<td>0.524</td>
<td>0.522</td>
<td>3.57</td>
</tr>
<tr>
<td>Sum pooling (SPoC w/o center prior)</td>
<td>0.802</td>
<td>0.589</td>
<td>0.578</td>
<td>3.65</td>
</tr>
<tr>
<td>SPoC (with center prior)</td>
<td>0.784</td>
<td>0.657</td>
<td>0.642</td>
<td>3.66</td>
</tr>
</tbody>
</table>
Experiments

- Retrieved examples using SPoC descriptor on the Oxford Building dataset
Experiments

- Comparison of overfitting effect arose from PCA matrix learning for SPoC and other methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Holidays</th>
<th>Oxford5K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher vector, k=16</td>
<td>0.704</td>
<td>0.490</td>
</tr>
<tr>
<td>Fisher vector, PCA on test, k=16</td>
<td>0.747</td>
<td>0.540</td>
</tr>
<tr>
<td>Fisher vector, k=256</td>
<td>0.672</td>
<td>0.466</td>
</tr>
<tr>
<td>Fisher vector, PCA on test, k=256</td>
<td>0.761</td>
<td>0.581</td>
</tr>
<tr>
<td>Triang. embedding, k=1</td>
<td>0.775</td>
<td>0.539</td>
</tr>
<tr>
<td>Triang. embedding, PCA on test, k=1</td>
<td>0.789</td>
<td>0.551</td>
</tr>
<tr>
<td>Triang. embedding, k=16</td>
<td>0.732</td>
<td>0.486</td>
</tr>
<tr>
<td>Triang. embedding, PCA on test, k=16</td>
<td>0.785</td>
<td>0.576</td>
</tr>
<tr>
<td>Max pooling</td>
<td>0.711</td>
<td>0.524</td>
</tr>
<tr>
<td>Max pooling, PCA on test</td>
<td>0.728</td>
<td>0.531</td>
</tr>
<tr>
<td>SPoC w/o center prior</td>
<td>0.802</td>
<td>0.589</td>
</tr>
<tr>
<td>SPoC w/o center prior, PCA on test</td>
<td>0.818</td>
<td>0.593</td>
</tr>
<tr>
<td>SPoC (with center prior)</td>
<td>0.784</td>
<td>0.657</td>
</tr>
<tr>
<td>SPoC (with center prior), PCA on test</td>
<td>0.797</td>
<td>0.651</td>
</tr>
</tbody>
</table>
References


Q & A
Quiz

1. SPoC descriptor is based on the aggregation of deep convolutional features without ( ).
   a. Aggregation
   b. Embedding
   c. Quantization
   d. Clustering

2. SPoC descriptor uses ( ) to assign larger weights to the features from the center of the feature map
   a. Centroid
   b. Triangulation
   c. Centering prior
   d. Fisher vector