Sparse-Coded Features for Image Retrieval

Tiezheng Ge
getzh@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Qifa Ke
qke@microsoft.com
Microsoft Bing
Sunnyvale, CA, US

Jian Sun
jiansun@microsoft.com
Microsoft Research Asia
Beijing, China

The bag-of-features(BOF) image representation [7] is popular in large-scale image retrieval. With BOF, the memory to store the inverted index file and the search complexity are both approximately linearly increased with the number of images. To address the retrieval efficiency and the memory constraint problem, besides some improvement work based on BOF, there come alternative approaches which aggregate local descriptors in one image into a single vector using Fisher Vector [6] or Vector of Local Aggregated Descriptor (VLAD) [1]. It has been shown in [1] that with as few as 16 bytes to represent an image, the retrieval performance is still comparable to that of the BOF representation.

In this paper, we illustrate that Fisher Vector, VLAD and BOF can be uniformly derived in two steps: **i Encoding** – separately map each local descriptor into a code, and **ii Pooling** – aggregate all codes from one image into a single vector. Motivated by the success of these two-step approaches, we propose to use sparse coding(SC) framework to aggregate local feature for image retrieval. SC framework is firstly introduced by [10] for the task of image classification. It is a classical two-step approach:

Step 1: Encoding. Each local descriptor $\mathbf{x}$ from an image is encoded into an N-dimensional vector $\mathbf{u} = [u^1, u^2, ..., u^N]$ by fitting a linear model with sparsity (L_1) constraint:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{x} - \mathbf{u}\mathtt{V}\|_2^2 + \lambda\|\mathbf{u}\|_1, \\ \text{subject to} \quad & \mathbf{u} \succeq 0 \end{aligned} \tag{1}$$

Here $\mathtt{V} = [\mathbf{v}_1, ..., \mathbf{v}_N]^\top$ is the codebook, i.e., a set of over-complete bases. It's learned beforehand by alternative optimization [10]. The second term is the L_1 penalty term to enforce sparsity on the vector $\mathbf{u}$, and λ is the parameter to control the sparsity.

Step 2: Pooling. For a given image with T descriptors $\mathtt{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, after obtaining their corresponding sparse codes $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_T\}$ in Step 1, we now pool them into a single N-dimensional vector $\mathbf{y} = [y^1, ..., y^i, ...y^N]$. There are two often-used pooling methods—average pooling and max pooling:

$$\text{average pooling:} \quad y^i = \sum_{t=1}^{T} u_t^i \tag{2}$$

$$\text{max pooling:} \quad y^i = \max\{u_t^i \,|\, t = 1, ..., T\} \tag{3}$$

The pooled vector $\mathbf{y}$ is subsequently normalized by $\mathbf{y} := \mathbf{y}/\|\mathbf{y}\|_2$ to generate the final single-vector representation. In practice we adopt max pooling, for it generally works better in experiment.

Despite some implementation differences, the SC framework can be easily applied for image retrieval. And as expected, it works well in experiments–be slightly superior to Fisher Vector and VLAD given the same local features. We take it to be our first contribution.

As our second contribution, we propose the strategy to combine multiple local features, which can greatly improve search accuracy while not hurt efficiency. In particular, we propose the following *multiple feature pooling and compression*:

1. Extract multiple local descriptor sets from a given image (one set for each type of descriptors);
2. Apply sparse coding to each descriptor set to derive a single aggregated vector;
3. Jointly compress all sparse-coded aggregated vectors to derive the final representation in a compact vector.

More formally, if we extract K sets of local descriptors and apply sparse coding and max pooling on each set independently to obtain aggregated vectors $\{\mathbf{y}_1, ..., \mathbf{y}_K\}$. Then, the final image representation is a concatenated vector:

$$\mathbf{y}^* = [\mathbf{y}_1^\top, ..., \mathbf{y}_K^\top]^\top. \tag{4}$$

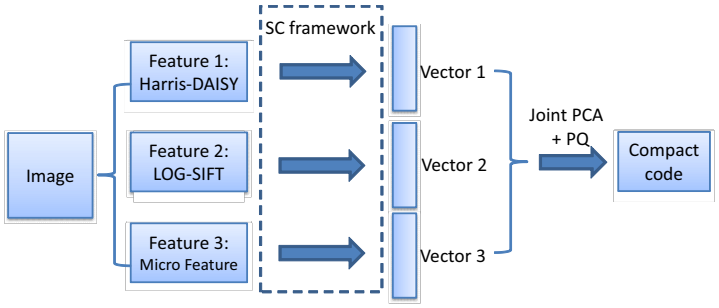

Figure 1: System pipeline

we found PCA performs well in reducing the dimension and achieve further compression (*e.g.*, to 20-40 bytes) with product quantization(PQ) [3].

To find effective and complementary local feature combination, we explored different local feature detectors [4](*e.g.*, LOG, Harris, MSER) associated with different types of descriptors(*e.g.*, SIFT, DAISY [8]). We tried various combinations of multiple detectors and descriptors, and empirically found that the best two-type combination was Harris-DAISY(HD) and LOG-SIFT(LS).

As our third contribution, we further propose a kind of novel color feature(called *micro feature*) which can well complement existing local invariant features. It is inspired by the work of bag-of-colors(BOC) [9] and can be seen as its extension.

The *micro feature* is quite simple: First, we densely sample the image on a grid with some predefined step size s. Then, at each sampled point, we extract a vector consisting of color values of a small $k \times k$ color patch in the CIE-Lab color space, resulting in a $3k^2$ dimensional vector, the *micro feature*. Finally, we feed all extracted vectors into the same sparse coding framework to produce our *sparse-coded micro features*.

The pipeline of complete system is depicted in Fig. 1.

We use two standard datasets (Holidays [2] and UKB [5]) to evaluate the performance of sparse-coded local invariant features, micro features, and their combinations.

[1] H. Jégou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[2] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[3] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *PAMI*, 33(1):117–128, 2011.

[4] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[5] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[6] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010.

[7] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[8] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR*, 2008.

[9] Christian Wengert, Matthijs Douze, and Herve Jégou. Bag of colors for improved image search. In *ACM Multimedia*, 2011.

[10] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classication. In *CVPR*, 2009.