Compact Tensor Based Image Representation for Similarity Search

Romain Negrel, David Picard, Philippe-Henri Gosselin

COMPACT TENSOR BASED IMAGE REPRESENTATION FOR SIMILARITY SEARCH

Romain Negrel, David Picard and Philippe-Henri Gosselin

ETIS/ENSEA - University of Cergy-Pontoise - CNRS, UMR 8051
6, avenue du Ponceau, BP44, F95014 Cergy-Pontoise, France
{romain.negrel,picard,gosselin}@ensea.fr

ABSTRACT

Within the Content Based Image Retrieval (CBIR) framework, one of the main challenges is to tackle the scalability issues. We propose a new compact signature for similarity search. We use an original method to perform a high compression of signatures while retraining their effectiveness. We propose an embedding method that maps large signatures into a low-dimensional hilbert space. We evaluated the method on Holidays database and compared the results with methods of state-of-the-art.

***Index Terms*—** Image retrieval, Image databases, Machine learning

1. INTRODUCTION

Content Based Image Retrieval (CBIR) is now an as well established field in the image processing community. In search by similarity tasks, the main problem is to compute image signatures that reduce as much as possible the semantic gap. Efficient systems have been proposed thanks to the introduction of highly discriminative local descriptors [1] and powerful aggregation schemes to produce the signatures [2, 3, 4]. However, these methods are not compact and require a lot of storage and computational resources. Scalability issues are then the next challenge of image search by similarity.

In this paper, we propose a new method for fast similarity search using compact signatures with high discriminative capacity. Our approach is as effective as current ones, but with a much smaller computational cost and data size. Our signatures are obtained by compressing a tensor based aggregation of local descriptors.

This paper is organized as follows: the next section describes recent related work on image representation. Section 3 introduces our novel approach and gives insight on its soundness. We then present successful experiences on Holidays dataset, before we conclude.

2. IMAGE VECTOR REPRESENTATION

In the similarity search and automatic indexing popular techniques are based on bag of features. Among these ones, two main methods: kernels based technique [5] and explicit embedding methods. In this article we will focus on mapping functions. These methods consist in mapping a set of descriptors into a single vector.

2.1. Bag of Words

A first method to map the set of descriptors into a single vector is called "Bag of Words" (BoW) [4]. This method involves a visual codebook composed of prototype descriptors (visual words), and count the occurrences of these prototypes within an image.

For some training set of images, the Visual codebook is usually computed using a clustering of descriptors (usually using K-means).

This method has many advantages: its implementation is very simple. The size of the signature generated by this method is small, as it depends only on the number of visual words in the visual codebook. However, nonlinear methods of similarity (e.g. Gaussian kernels with ℓ_2-distance or $\chi 2$-distance) are required to reach good performances.

2.2. Coding/pooling schemes

The Coding/pooling method is a generalisation of BoW. It divides the mapping in two steps. The first step is to map each descriptor of the image on the dictionary ("coding" step). The second step is to aggregate the mapped descriptors into a single signature ("pooling" step).

To compute BoW with these schemes, 1 is assigned to the closest visual word during the coding step (0 otherwise). The pooling step is the sum of all mapped vectors.

In the coding scheme of [2], a mapped vector $\boldsymbol{\alpha}^\star$ is the result of a reconstruction problem :

$$\boldsymbol{\alpha}^\star = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\mathbf{b} - \mathbf{W}\boldsymbol{\alpha}\|^2 + \|\mathbf{d} \circ \boldsymbol{\alpha}\|^2, \qquad (1)$$

with $\mathbf{b}$ being a descriptor, $\mathbf{W}$ the codebook matrix, $\boldsymbol{\alpha}$ the projection coeficients and $\mathbf{d}$ a locality constraint. The output vector $\boldsymbol{\alpha}^\star$ is thus optimized with respect to the reconstruction error and constrained to the projection on few nearby codewords. They also propose an alternative to the sum pooling

step by computing the maximum coefficient among descriptors of the set.

Unlike BoW, coding/pooling schemes give good performance with linear similarities, while retaining a small size. However this method is more complex to implement and it requires a fine tuning of the parameters to obtain good results.

2.3. Fisher Vectors - VLAD

Perronnin et al. [6, 3] have recently proposed a novel method called Fisher Vectors to map a set of descriptors in a single signature. Their model computes the deviation of each descriptor to a Gaussian Mixture Model of the distribution of all descriptors in the database. Then they sum these deviations for all descriptors of the image.

Jegou et al. [7] proposed a method to perform an approximation of Fisher vectors, called Vectors of Locally Aggregated Descriptors (VLAD). This method has two steps: first, a small codebook is generated by clustering. Then, the sum of all centered descriptors from image i and cluster c is computed:

$$\boldsymbol{\nu}_{c,i} = \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)\,; \tag{2}$$

where $\mathbf{b}_{rci}$ are the descriptors for image i in cluster c and $\boldsymbol{\mu}_c$ is the center of the cluster. The final signature $\boldsymbol{\nu}_i$ is obtained by concatenating $\boldsymbol{\nu}_{c,i}$ for all c, and is thus of size $C \times D$, with C the size of codebook and D the size of features.

Like the coding/pooling, this approach gives good performance with linear metric. The dimension of VLAD vectors is large and depends on the size of descriptors, however it can be significantly reduced using techniques like PCA[3].

2.4. VLAT - A tensor based aggregation

The Vector of Locally Aggregated Tensors (VLAT) is an improvement of VLAD recently presented by Picard et al [8]. In this method, the authors propose to aggregate tensor products of local descriptors to produce a unique signature.

First, they compute a visual codebook of C visual words over a sample image set using k-means.

Then, they compute the mean descriptor $\boldsymbol{\mu}_c$ and mean centered tensor $\mathcal{T}_c$ of each cluster c:

$$\boldsymbol{\mu}_c = \frac{1}{|c|} \sum_i \sum_r \mathbf{b}_{rci}, \tag{3}$$

$$\mathcal{T}_c = \frac{1}{|c|} \sum_i \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top, \tag{4}$$

with $|c|$ the number of descriptors in cluster c and $\mathbf{b}_{rci}$ the descriptors of image i belonging to cluster c.

For each image and each cluster, a signature $\mathcal{T}_{i,c}$ is computed by aggregating the centered tensors of centered descriptors:

$$\mathcal{T}_{i,c} = \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c. \tag{5}$$

Each $\mathcal{T}_{i,c}$ is flattened into a vector $\mathbf{v}_{i,c}$.

The VLAT signature $\mathbf{v}_i$ for image i consists in the concatenation of $\mathbf{v}_{i,c}$ for all clusters c:

$$\mathbf{v}_i = (\mathbf{v}_{i,1} \dots \mathbf{v}_{i,C})\,. \tag{6}$$

The VLAT gives very good results in similarity search and automatic indexing of images with linear metric, but leads to large feature vectors. The size of the VLAT descriptor is $C \times D \times D$, with C the number of clusters and D the size of descriptors.

3. COMPACT VLAT

We propose to drastically reduce the size of VLAT features, while keeping their discriminative power. Our method uses the following scheme : first we preprocess the VLAT by performing a normalization step. Then, we compute the Gram matrix of normalized VLAT for some training set of images. We perform a low rank approximation of the Gram matrix and compute the set of projection vectors associated with the generated subspace. The compact VLAT signatures are the projections of normalized VLAT using the obtained projection vectors.

3.1. Normalization

First, we compute the set of VLAT signatures of the image database. Then, we perform a two steps normalization. we compute the power norm of $\mathbf{v}_i$ to produce $\mathbf{v}'_i$:

$$\forall j, \quad \mathbf{v}'_i[j] = \mathbf{v}_i[j]^\alpha, \tag{7}$$

With α typically set to 0.5. Then, we normalize the vector $\mathbf{v}'_i$ with the ℓ_2-norm:

$$\mathbf{x}_i = \frac{\mathbf{v}'_i}{\|\mathbf{v}'_i\|} \tag{8}$$

$\mathbf{x}_i$ is the normalized VLAT signature. We use the performance of normalized VLAT signatures for comparison with performance of the compact VLAT signatures. Normalization is necessary to obtain a Gram matrix with good properties (unitary diagonal, well-conditioned, etc).

3.2. Low rank approximation

We compute the Gram matrix $\mathbf{G}$ of centered normalized VLAT signatures:

$$\mathbf{G}_{i,j} = (\mathbf{x}_i - \mathbf{m}_\mathbf{x})^\top (\mathbf{x}_j - \mathbf{m}_\mathbf{x}), \tag{9}$$

with $\mathbf{m}_\mathbf{x}$ the mean of $\mathbf{x}_i$. Then, we compute the eigenvalues and eigenvectors of the Gram matrix:

$$(\lambda_1 \dots \lambda_l \dots \lambda_N) = eigval(\mathbf{G}), \tag{10}$$

$$(\mathbf{u}_1 \dots \mathbf{u}_l \dots \mathbf{u}_N) = eigvect(\mathbf{G}), \tag{11}$$

with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ and $\mathbf{u}_l$ the eigenvector associated the eigenvalue λ_l. We compute a low rank approximation of the Gram matrix. We denote by $\mathbf{L}_t$ the matrix with the t largest eigenvalues on the diagonal:

$$\mathbf{L}_t = diag(\lambda_1 \dots \lambda_t), \tag{12}$$

and we denote by $\mathbf{U}_t$ the matrix of the first t eigenvectors:

$$\mathbf{U}_t = [\mathbf{u}_1 \dots \mathbf{u}_t]\,. \tag{13}$$

We can then define $\mathbf{G}_t$ as an approximation of $\mathbf{G}$:

$$\mathbf{G}_t = \mathbf{U}_t \mathbf{L}_t \mathbf{U}_t^\top. \tag{14}$$

Then, we compute the projectors of VLAT signatures in approximated subspace:

$$\mathbf{P}_t = \mathbf{X}\mathbf{U}_t \mathbf{L}_t^{-1/2}, \tag{15}$$

with $\mathbf{X} = [\mathbf{x}_1 - \mathbf{m}_\mathbf{x} \dots \mathbf{x}_N - \mathbf{m}_\mathbf{x}]$ is the matrix of centred VLAT signatures. This method is analog to Kernel-PCA using a dot product Kernel.

3.3. Projection vectors

For each image, we compute the projection of VLAT signature in the approximated space:

$$\mathbf{y}_i = \mathbf{L}_t^{-1/2} \mathbf{P}_t^\top \mathbf{x}_i. \tag{16}$$

$\mathbf{y}_i$ contains an approximate and compressed version of $\mathbf{x}_i$. This method keeps the most informative directions of VLAT feature space.

Moreover, the normalization by $\mathbf{L}_t^{-1/2}$ produces axes of equal variance in the projected subspace. Let us compute the covariance matrix of projections $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N]$:

$$\begin{aligned} \mathbf{Y}\mathbf{Y}^\top &= \mathbf{L}_t^{-1/2} \mathbf{P}_t^\top \mathbf{X}\mathbf{X}^\top \mathbf{P}_t \mathbf{L}_t^{-1/2} && (17)\\ &= \mathbf{L}_t^{-1} \mathbf{U}_t^\top \mathbf{G}\mathbf{G}\mathbf{U}_t \mathbf{L}_t^{-1}. && (18) \end{aligned}$$

Let us denote by $\bar{\mathbf{U}}$ the matrix of remaining eigenvectors such that $\mathbf{G} = \mathbf{U}_t \mathbf{L}_t \mathbf{U}_t^\top + \bar{\mathbf{U}}\bar{\mathbf{L}}\bar{\mathbf{U}}^\top$, then $\mathbf{Y}\mathbf{Y}^\top$ simplifies to:

$$\begin{aligned} \mathbf{Y}\mathbf{Y}^\top &= \mathbf{L}_t^{-1} \mathbf{U}_t^\top (\mathbf{U}_t \mathbf{L}_t \mathbf{U}_t^\top + \bar{\mathbf{U}}^\top \bar{\mathbf{L}} \bar{\mathbf{U}})^2 \mathbf{U}_t \mathbf{L}_t^{-1} && (19)\\ &= \mathbf{L}_t^{-1} \mathbf{U}_t^\top \mathbf{U}_t \mathbf{L}_t \mathbf{U}_t^\top \mathbf{U}_t \mathbf{L}_t \mathbf{U}_t^\top \mathbf{U}_t \mathbf{L}_t^{-1} && (20)\\ &= \mathbf{I}, && (21) \end{aligned}$$

since $\mathbf{U}_t^\top \mathbf{U}_t = \mathbf{I}$ and $\mathbf{U}_t^\top \bar{\mathbf{U}} = 0$.

The size of $\mathbf{y}_i$ is very small compared to the size of $\mathbf{x}_i$, as it directly depends on t (varying from 1 to N). The final step is a ℓ_2-normalization of $\mathbf{y}_i$:

$$\mathbf{z}_i = \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|}. \tag{22}$$

	Compact VLAT						
t	16	32	64	128	256	512	1024
VLAT	29.6	38.6	46.1	**49.2**	48.7	44.1	22.2
Normalized VLAT	45.0	54.2	63.1	68.5	**72.3**	71.1	54.8
VLAD	35.7	43.5	49.0	**51.2**	50.3	47.1	21.6
Normalized VLAD	44.3	50.7	57.5	58.8	**59.6**	58.1	44.6

Table 1. Comparison of mAP(%) between normalized or not normalized VLAD and VLAT descriptor.

	Compact VLAT							Standard
D. Size	16	32	64	128	256	512	1024	VLAT
64	45.0	54.2	63.1	68.5	**72.3**	71.1	54.8	66.4
256	44.6	53.6	62.4	68.2	**74.6**	74.2	56.7	71.8
1024	40.1	52.0	62.3	69.2	74.2	**76.0**	57.1	75.7

Table 2. Comparison of mAP(%) between compact VLAT and VLAT descriptor.

4. EXPERIMENTS

We used the Holidays database to evaluate our compact VLAT signatures and compare them with VLAT and VLAD signatures. The Holidays dataset is a set of images drawn from personal holidays photos, created to test methods of similarity search. It contains 1491 images gathered in 500 subgroups, each of them represents a distinct scene or object (Figure 1). Images in this database are in high resolution color. The Holidays dataset includes set of SIFT descriptors.

4.1. Image Similarity

We used the same evaluation setup as Jegou et al. [9]. For all our experiments, we compute a set of codebook (64, 256, 1024 visual words) with all provided SIFT descriptors [1]. For each experiment, we evaluate the influence of parameter t (number of selected eigenvalue and eigenvector) as a function of the size of the codebook. We compare our results with those obtained with VLAD and Compact VLAD. We used our method to build compact VLAD. Note that for all method and experiments, we used the same codebooks.

First, we propose to examine the importance of normalization of VLAT and VLAD signatures. For this, we evaluate the compact signatures without the normalization step (section 3.1). We can see (Table 1) that the normalization step is essential to keep the performance after the of low rank approximation. Due to the large size of the VLAT descriptors, it is more sensitive to normalization. Table 2 is the comparison between the compact VLAT and standard VLAT descriptors. We see that, whatever the size of the codebook, there are values of t for which the compact VLAT signatures give better results than the standard VLAT signatures. When keeping fewer eigenvalues, noise is introduced by the normalization of the projection vectors in eq. 15. There is a optimal level of approximation where information is better represented in the compact VLAT than in the standard VLAT. We

[1] http://www.vlat.fr/

Fig. 1. Images from Holidays dataset.

	Compact VLAD							Standard
D. Size	16	32	64	128	256	512	1024	VLAD
64	44.3	50.7	57.5	58.8	**59.6**	58.1	44.6	55.2
256	45.6	54.3	60.6	65.1	**66.1**	64.0	50.7	59.6
1024	45.8	55.4	62.8	67.8	**70.8**	69.8	54.1	63.9

Table 3. Comparison of mAP(%) between compact VLAD and VLAD descriptor.

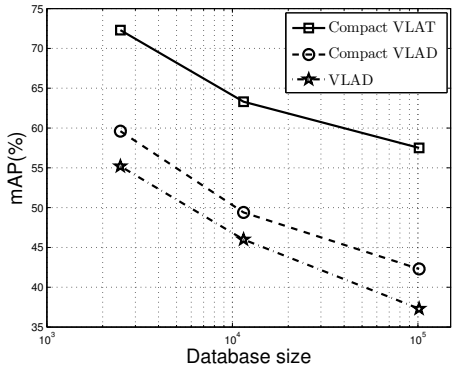

Fig. 2. Search mAP as a function of the database size (Codebook size : 64, t = 256).

see that the compact VLAD (Table 3) has the same behavior as the VLAT compact, there is a optimal level of approximation which maximizes results. Comparing the results of the Compact VLAT with those of standard VLAD and compact VLAD, we can see that our method generally gives better results. With a codebook of size 1024, we have **12.1%** of gain compared to standard VLAD.

4.2. Large Scale - Similarity search

In this section, we want to test the robustness of our method on large search sets. For our tests, we used the Holidays database artificially enlarged by subsets of different sizes (10k and 100k documents) from the Flickr10M database. We can then see how searching in the Holidays database is disrupted by the expansion of the database. Figure 2 shows the evolution of the mean Average Precision against the size of database, we see that our signatures have the same behavior as the VLAD feature. Our compact VLAT signatures always perform better than VLAD signatures. If we compare with the results of state-of-the-art [3] (on different codebook), we have a gain of about **11%** mAP for 100k images.

5. CONCLUSION

In this paper, we proposed a compact image signature for similarity search in large databases. Our method consists in the projection of tensors based aggregation of local descriptors on a low dimensional subspace. This subspace is obtained by a low rank approximation of the Gram matrix on a training set. We provided experiments on the well known Holydays dataset showing that our approach gives very good results while being several orders of magnitude more compact than uncompressed signatures. Experiments on a larger dataset (100k images) lead to promising results regarding the scalability of the method.

6. REFERENCES

[1] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[2] J. Wang, J., K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE CVPR*, 2010, pp. 3360–3367.

[3] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE PAMI*, 2012, QUAERO.

[4] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, vol. 2.

[5] P.H. Gosselin, M. Cord, and S. Philipp-Foliguet, "Kernel on bags of fuzzy regions for fast object retrieval," in *ICIP*, San Antonio, Texas, USA, September 2007.

[6] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE CVPR*, June 2007.

[7] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011, vol. 76, pp. 1–12.

[8] D. Picard and P.H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *ICIP*, Brussels, Belgique, September 2011.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE CVPR*, June 2010, pp. 3304–3311.