Efficient image signatures and similarities using tensor products of local descriptors

David Picard, Philippe-Henri Gosselin

Efficient image signatures and similarities using tensor products of local descriptors

David Picard, Philippe-Henri Gosselin

ETIS, CNRS ENSEA Universite Cergy-Pontoise F-95000 Cergy-Pontoise

Abstract

In this paper, we introduce a novel image signature effective in both image retrieval and image classification. Our approach is based on the aggregation of tensor products of discriminant local features, named VLAT (vector of locally aggregated tensors). We also introduce techniques for the packing and the fast comparison of VLATs. We present connections between VLAT and methods like kernel on bags and Fisher vectors. Finally, we show the ability of our method to be effective for two different retrieval problems, thanks to experiments carried out on similarity search and classification datasets.

Keywords: Image retrieval, Image indexing, Image Similarity, Feature Vector

1. Introduction

Content Based Image Retrieval (CBIR) has been a main topic of interest at the the crossroad of many research communities for the last 20 years. These communities include (but are not limited to) Image Processing, Computer Vision, Machine Learning and Data Mining. During these years, interesting results were obtained on image collections designed on purpose by the community. With the tremendous amount of available images, CBIR is now even more challenging. For instance, Flickr repository contains more than 6 billion images as of 2011. These amounts also increased the number of searched concepts, and raised the need for computational efficiency. In [1], Smeulders et al. identify several main areas for CBIR, depending on what the user intends, like image search and image classification tasks.

In the search task, we are interested in the fast retrieval of a single image and its near duplicates within a very large collection [2]. For instance, let us consider a collection of images taken from Paris streets. A typical search query could be to retrieve all images of the front side of the *Sacré Cœur*. The query can be instantiated by presenting an example image to the system, or by typing a keyword. In such system, the collection is huge (up to several millions of images), and the resulting sets are small (usually less than ten images). The main challenges are threefold. The first one is to obtain a good recall, *i.e.* to retrieve almost all targeted images. The second challenge is to perform the retrieval with low computational complexity with respect to the collection size. In many cases, sub linear complexity is targeted [3, 4, 5], but linear complexity can also be relevant if indexes are small and/or similarity can be computed very quickly [6]. The third challenge is to lower the storage cost of the indexing method [7].

In the classification task, the goal is to label a collection of images with respect to a set of predetermined semantic categories. The categories are well defined objects (such as persons, cars, etc) or more complex semantic grouping of images (such as indoor, landscapes, urban areas, etc). Recent methods make a heavy use of machine learning techniques, such as SVM [8, 9, 10] or Boosting [11, 12]. A classifier is trained for each desired concept on a separate image set where corresponding labels are available. These classifiers are then used to label the collection. In such systems, the main challenges are twofold: firstly, to produce accurate classifiers, and secondly to perform the labeling in reasonable time.

In this paper, we introduce a novel image signature for image search and image classification tasks. We first present in the second section the characteristics of these tasks, and an overview of state of the art approaches in these scopes. Then we detail our propositions based on tensors aggregation in the third section, and techniques to save memory and computations. In the fourth section, we present results for similarity search and classification tasks on Holidays and VOC2007 datasets. In the last section, we discuss the relationships between our propositions and current methods from the literature.

2. State of the art image signatures and similarities

As stated in the introduction, CBIR offers several challenging tasks. Thus, current retrieval systems often differ depending on which tasks they address. However, we can identify from all systems a significant number of similar processing stages. The first one is the automatic extraction of a set of highly discriminant local features from the images [13, 14]. This set can be the result of specifically detected or uniformly sampled regions of interest [15]. In all cases, we assume in this paper that a set $\mathbf{B}_i = \{\mathbf{b}_{ri}\}_r$ of vector descriptors $\mathbf{b}_{ri} \in \mathbb{R}^D$ are extracted from image i.

The goal of the hereby studied step is to produce an image signature and the associated similarity measure. It that can ei-

Email addresses: picard@ensea.fr (David Picard), gosselin@ensea.fr (Philippe-Henri Gosselin)

ther be used for database indexing in the case of the search task, or be taken as input by a learning machine in the case of the classification task. In this scope, two main approaches can be highlighted. The first consists in keeping all descriptors from each image, and then in designing complex similarity measures on these sets. The second approach produces a single vector from the aggregation of all descriptors and uses standard vector based similarities.

2.1. Kernels on Bags approaches

Given two sets (named bags) $\mathbf{B}_i$ and $\mathbf{B}_j$ of local descriptors extracted from image i and image j, and a similarity function $k(\cdot,\cdot)$ on descriptors, a simple way of measuring the similarity $K(\cdot,\cdot)$ over the bags is to compute the sum of similarity of all possible pairing between elements of $\mathbf{B}_i$ and $\mathbf{B}_j$:

$$K(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{1}$$

This approach is known as the kernel on bags method and has been widely studied in the machine learning community [16]. Providing k is a kernel, it is easy to prove that K is also a kernel and thus can be used with any kernel learning algorithm.

However, the sum kernel often offers low discriminative output, since it is the average similarity between elements of the bags. It is even more the case when the number and variety of descriptors in bags increase. To tackle this problem, Lyu proposed a refined sum kernel by increasing the discriminative power of the minor kernel k [17]:

$$K_{lyu}(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p \tag{2}$$

Providing k has output in the range $[0, 1]$, power parameter $p \geq 1$ increases the discriminative ability of the minor kernel. Thus, only highly similar elements of the bags will be accounted for.

Gosselin et al. [18] further refined the power sum kernel by approximating a soft max function over the similarities between elements in the bags:

$$K_{softmax}(\mathbf{B}_i, \mathbf{B}_j) = \left(\sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^p \right)^{\frac{1}{p}} \tag{3}$$

When p increases, k tends to the maximum similarity $k(\mathbf{b}_{ri}, \mathbf{b}_{sj})$ among possible pairs $(\mathbf{b}_{ri}, \mathbf{b}_{sj})$. It is however not known if this function is a Mercer kernel. In practice, it is observed to be semi-definite positive (*sdp*) on many datasets.

Further investigations on kernel on bags similarities have been proposed recently, which have been proven to be successful in many areas such as image retrieval, interactive search and graph matching [19, 20, 21].

The main drawback of kernel on bags methods is their high computational cost. The average cost is $O(T^2 D)$ where T is the average size of the bags and D the dimension of local features. For very large bags (several thousands of features), the cost becomes prohibitive, especially if the similarity is used intensively by the learning algorithm. Although acceleration schemes have been proposed [22], we believe this prohibitive cost is the main argument why kernels on bags have not been widely used in CBIR.

2.2. Aggregating approaches

A way of getting rid of the complexity of the bags is to find a mapping function encoding a set of descriptors into a single vector.

2.2.1. Bag of Words

Historical approaches for mapping the set of descriptors into a single vector compute a visual codebook composed of prototypes of descriptors, and then find the occurrences of these prototypes (denoted visual codewords or visual words) within an image [23, 24, 25]. The codebook is generally obtained by clustering a large set of descriptors. The approach is known as the "Bag of Words" (BoW) method, and has been adapted from text retrieval methods.

A simple way of mapping a set of descriptors into a vector is to compute a histogram of visual words. Further improvements consist in changing the value added to the related bin of the histogram. Gemert et al. [26] propose various schemes based on visual word ambiguity, where the assigned value is related to the similarity of the descriptor to the selected visual word, and can be normalized with respect to its similarity to other visual words.

2.2.2. Coding/pooling schemes

In [27], the authors propose to decompose the BoW approach into two distinct steps. The first step, called the coding step, involves the mapping of a single descriptor onto the codebook. In the second step, called the pooling step, all mapped descriptors are aggregated into a single vector.

In first BoW approaches, the values of a mapped descriptor are all zeros, except for the dimension corresponding to the closest visual word. The pooling step is simply the sum of all mapped vectors.

In the coding scheme of [27], a mapped vector $\boldsymbol{\alpha}^\star$ is the result of a reconstruction problem :

$$\boldsymbol{\alpha}^\star = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{b} - \mathbf{W}\boldsymbol{\alpha}\|^2 + \|\mathbf{d} \circ \boldsymbol{\alpha}\|^2 \tag{4}$$

With $\mathbf{b}$ being a descriptor, $\mathbf{W}$ the codebook matrix, $\boldsymbol{\alpha}$ the projection coefficients and $\mathbf{d}$ a locality constraint (*e.g.* farther visual words have a higher cost than closer words). The output vector $\boldsymbol{\alpha}^\star$ is thus optimized with respect to its 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. The max pooling step is shown to have great impact on the classification performances [28].

Other pooling steps include taking into account the spatial distribution of visual words. In [29], the authors propose a spatial pyramid matching scheme (SPM) where descriptors are aggregated in a subarea of the image depending on scale parameters, inspired by the Pyramid Matching Kernel of [30].

2.2.3. Fisher Vectors - VLAD

More recently, Perronnin et al. [31, 32] proposed a novel approach to map a set of descriptors to a single vector. In this method, the authors assume that the bag of descriptors $\mathbf{B}_i$ can be modeled by a Gaussian mixture, with a probability density function denoted as u_λ of parameters λ. The derivative of the log-likelihood of $\mathbf{B}_i$ with respect to the parameters describes each parameter contribution to the global likelihood:

$$G_\lambda^{\mathbf{B}_i} = \frac{1}{T}\nabla_\lambda \log u_\lambda(\mathbf{B}_i) \tag{5}$$

Using the Fisher information matrix, one can rewrite this derivative such that the dot product between rewritten vectors is an efficient kernel. The mapped vectors are thus the following for each Gaussian c:

$$\mathcal{G}_{\mu,c}^{\mathbf{B}_i} = \frac{1}{T\sqrt{\omega_c}}\sum_r \gamma_c(\mathbf{b}_{ri})\left(\frac{\mathbf{b}_{ri}-\boldsymbol{\mu}_c}{\boldsymbol{\sigma}_c}\right), \tag{6}$$

$$\mathcal{G}_{\sigma,c}^{\mathbf{B}_i} = \frac{1}{T\sqrt{\omega_c}}\sum_r \gamma_c(\mathbf{b}_{ri})\left[\frac{(\mathbf{b}_{ri}-\boldsymbol{\mu}_c)^2}{\boldsymbol{\sigma}_c^2}-1\right], \tag{7}$$

Where $(\omega_c, \boldsymbol{\mu}_c, \sigma_c)$ are the weight, mean and standard deviation of Gaussian c, and $\gamma_c(\mathbf{b}_{ri})$ the normalized likelihood of $\mathbf{b}_{ri}$ to Gaussian c. The output vector is obtained by concatenation, $\mathcal{G}_\mu = \bigcup_c \mathcal{G}_{\mu,c}^{\mathbf{B}_i}$ and $\mathcal{G}_\sigma = \bigcup_c \mathcal{G}_{\sigma,c}^{\mathbf{B}_i}$. The final feature vector is of size $2 \times D \times C$, where D is the dimension of input space and C the number of Gaussians.

To our knowledge, Fisher vectors and related methods obtain state of the art classification performances when using single SIFT descriptors with linear classifier and when properly normalized [33, 34]. These properties make them attractive for large scale classification challenges, as linear classifiers are almost costless compared to kernel based classifiers. Furthermore, the number of Gaussians needed to achieve good performances is far less than the typical size of the codebook in the BoW approaches. The clustering of high-dimensional spaces in many cluster being a difficult task, the Fisher vectors method is consequently more robust.

In [32], Jegou et al. propose an approximation of Fisher vectors by aggregating local descriptors (VLAD - Vectors of Locally Aggregated Descriptors). Given a codebook obtained by clustering a large set of descriptors, they compute the centered sum of all descriptors $\mathbf{B}_{ci} = \{\mathbf{b}_{rci}\}_r \subseteq \mathbf{B}_i$ from image i and cluster c:

$$\boldsymbol{\nu}_c = \sum_r \mathbf{b}_{rci} - \boldsymbol{\mu}_c \tag{8}$$

The output vector is obtained by concatenating $\boldsymbol{\nu}_c$ for all c, and is thus of size $D{\times}N$. Note that VLAD and FV are used with smaller codebook (from 16 to 4096) than BoW related techniques (from 4000 to millions). Consequently, the computation of the codebook is faster, and results are more stable from one codebook to another.

3. VLAT - A tensor based aggregation

Our approach extends VLAD by aggregating tensor products of local descriptors, hence the name Vector of Locally Aggregated Tensors (VLAT). Preliminary results were published in [35]. We first compute a visual codebook of C visual words over a sample image set S using k-means. Let us denote $\boldsymbol{\mu}_c$ the center of cluster c and $\mathcal{T}_c$ the mean tensor of centered descriptors belonging to cluster c:

$$\boldsymbol{\mu}_c = \frac{1}{|c|}\sum_i\sum_r \mathbf{b}_{rci} \tag{9}$$

$$\mathcal{T}_c = \frac{1}{|c|}\sum_i\sum_r (\mathbf{b}_{rci}-\boldsymbol{\mu}_c)(\mathbf{b}_{rci}-\boldsymbol{\mu}_c)^\top, \tag{10}$$

With $|c|$ being the total number of descriptors in cluster c for the whole image sample set S, and $\mathbf{b}_{rci}$ the descriptors of image i belonging to cluster c.

To compute the signature of an image i, and for each cluster c, we aggregate in $\mathcal{T}_{i,c}$ centered tensors of centered descriptor:

$$\mathcal{T}_{i,c} = \sum_r (\mathbf{b}_{rci}-\boldsymbol{\mu}_c)(\mathbf{b}_{rci}-\boldsymbol{\mu}_c)^\top - \mathcal{T}_c \tag{11}$$

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:

$$\mathbf{v}_i = (\mathbf{v}_{i,1} \dots \mathbf{v}_{i,C}) \tag{12}$$

The size of then VLAT is $C \times D \times D$, with C the number of clusters and D the dimension of descriptors. However, due to symmetry in the proposed tensors, half of the coefficients are discarded.

For each cluster c, $\mathbf{v}_{i,c}$ can be seen as the difference between the covariance matrix of descriptors in $\mathbf{B}_i$ belonging to cluster c and the covariance matrix $\mathcal{T}_c$ of cluster c. VLAT thus measures deviations of second order statistics of image i with respect to a reference cluster c, whereas VLAD measures deviations of first order statistics only. Theoretically, higher order of tensors can be considered, measuring deviations of higher order statistics in $\mathbf{B}_i$ with respect to cluster c.

As in [33], we can further improve VLAT by applying proper normalization steps. First we compute the power norm of $\mathbf{v}_i$ to produce vector $\mathbf{v}'_i$, which is simply raising each coordinate j of $\mathbf{v}_i$ to power α:

$$\forall j, \quad \mathbf{v}'_i[j] = \text{sign}(\mathbf{v}_i[j])|\mathbf{v}_i[j]|^\alpha, \tag{13}$$

With $\alpha = 0.5$ as a typical parameter value. Then, we compute the signature $\mathbf{x}_i$ of image i by applying ℓ_2-norm to $\mathbf{v}'_i$:

$$\mathbf{x}_i = \frac{\mathbf{v}'_i}{\|\mathbf{v}'_i\|^2} \tag{14}$$

It is worth noticing that VLAT is very easy to compute and does not require complex models of the feature space. Any method producing a clustering of the descriptors can be used to obtain the signatures. Moreover, VLAT is very fast to compute on current CPU and GPGPU, since most computations are performed with simple linear algebra.

3.1. Scalability issues - packing

The size in VLAT signature is quadratically increasing with descriptor space dimension D, and thus quickly leads to large feature vectors. A first and simple way of reducing this overhead is to apply principal component analysis (PCA) to visual descriptor space. Typically, SIFT descriptors can be reduced from $D = 128$ to $D' = 64$ dimension with negligible lost of information. Furthermore, PCA can be used to perform a whitening step before VLAT computations.

In order to save memory use, we propose a packing procedure to reduce the size of VLAT indexes. The idea is to store the tensor of cluster c for image i in a format close to the original expression, *i.e.* a bag of virtual descriptors $\{\mathbf{p}_{r'ci}\}_{r'ci}$. Similarly to Eq. (11), the approximate aggregated tensor is expressed using $\{\mathbf{p}_{r'ci}\}_{r'ci}$:

$$\mathcal{P}_{i,c} = \sum_{r'=1}^{n'_{ci}} \lambda_{r'ci}\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top} - n_{ci}\mathcal{T}_c \tag{15}$$

where we chose the parameters $\{(\lambda_{r'ci}, \mathbf{p}_{r'ci})\}_{r'}$ such as the error with the original tensor $\mathcal{T}_{ic}$ is minimized. We evaluate this error as the relative Euclidean distance between the two tensors:

$$Err(\mathcal{P}_{i,c}, \mathcal{T}_{i,c}) = \frac{\|\mathcal{T}_{i,c} - \mathcal{P}_{i,c}\|}{\|\mathcal{T}_{i,c}\|} \tag{16}$$

The parameters $\{(\lambda_{r'ci}, \mathbf{p}_{r'ci})\}_{r'}$ are computed using an eigendecomposition of image descriptors $\mathbf{B}_{ci}$ falling in cluster c:

$$\mathbf{B}_{ci} = \mathbf{P}_{ci}\Lambda_{ci}\mathbf{P}_{ci}^{\top} = \sum_{r=1}^{n_{ci}} \lambda_r \mathbf{p}_{rci}\mathbf{p}_{rci}^{\top} \tag{17}$$

where $\mathbf{P}_{ci}$ is an orthonormal matrix and Λ_{ci} a diagonal matrix.

Next we select the n'_{ci} largest eigenvalues such as a target error is reached. Let us note that the error can be easily computed since the $\mathbf{p}_{rci}$ vectors are orthonormal:

$$\begin{aligned}
\|\mathcal{T}_{i,c} - \mathcal{P}_{i,c}\|^2 &= \|\sum_{r=1}^{n_{ci}} \lambda_r \mathbf{p}_{rci}\mathbf{p}_{rci}^{\top} - \sum_{r'=1}^{n'_{ci}} \lambda_{r'}\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top}\|^2 \\
&= \|\sum_{r=n'_{ci}+1}^{n_{ci}} \lambda_r \mathbf{p}_{rci}\mathbf{p}_{rci}^{\top}\|^2 \\
&= \sum_{r=n'_{ci}+1}^{n_{ci}} \lambda_r^2
\end{aligned}$$

A first use of this packing procedure for the comparison of two images i, j is to reconstruct VLAT estimates $\hat{\mathcal{T}}_{i,c}$ and $\hat{\mathcal{T}}_{j,c}$ using the packed VLATs $\mathcal{P}_{i,c}$ and $\mathcal{P}_{j,c}$, and then compute the dot product $\langle\hat{\mathcal{T}}_{i,c}, \hat{\mathcal{T}}_{j,c}\rangle$. This is relevant for small codebooks, but with larger ones, it is much more efficient to compute a straight comparison of packed VLATs:

$$\begin{aligned}
\langle\mathcal{P}_{i,c}, \mathcal{P}_{j,c}\rangle &= \left\langle \sum_{r'=1}^{n'_{ci}} \lambda_{r'ci}\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top} - n_{ci}\mathcal{T}_c, \sum_{s'=1}^{n'_{cj}} \lambda_{s'cj}\mathbf{p}_{s'cj}\mathbf{p}_{s'cj}^{\top} - n_{cj}\mathcal{T}_c \right\rangle \\
&= \sum_{r'=1}^{n'_{ci}}\sum_{s'=1}^{n'_{cj}} \lambda_{r'ci}\lambda_{s'cj}\langle\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top}, \mathbf{p}_{s'cj}\mathbf{p}_{s'cj}^{\top}\rangle + n_{ci}n_{cj}\|\mathcal{T}_c\|^2 \\
&\quad -n_{cj}\sum_{r'=1}^{n'_{ci}} \lambda_{r'ci}\langle\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top}, \mathcal{T}_c\rangle - n_{ci}\sum_{s'=1}^{n'_{cj}} \lambda_{s'cj}\langle\mathbf{p}_{s'cj}\mathbf{p}_{s'cj}^{\top}, \mathcal{T}_c\rangle \\
&= \sum_{r'=1}^{n'_{ci}}\sum_{s'=1}^{n'_{cj}} \lambda_{r'ci}\lambda_{s'cj}\langle\mathbf{p}_{r'ci}, \mathbf{p}_{s'cj}\rangle^2 + n_{ci}n_{cj}\|\mathcal{T}_c\|^2 - n_{cj}f_{ci} - n_{c}
\end{aligned} \tag{18}$$

with $f_{ci} = \sum_{r'=1}^{n'_{ci}} \lambda_{r'ci}\langle\mathbf{p}_{r'ci}\mathbf{p}_{r'ci}^{\top}, \mathcal{T}_c\rangle$.

Thanks to some index overhead to store f_{ci} values, the computational complexity is $O(C \times D \times n'_{ci} \times n'_{cj})$. With larger codebooks, image descriptors are more and more spread into different clusters, and $n'_{ci} \times n'_{cj}$ tends to be negligible when compared to descriptors dimensions D. This context leads a complexity of $O(C \times D)$, which is far less than the $O(C \times D \times D)$ complexity of unpacked VLATs.

Packed VLATs significantly reduce memory footprint, especially with larger codebooks. For instance, on the Holidays dataset [36] with 1024 clusters, standard VLATs needs about 8MB per image, whereas the packed VLATs with no reconstruction error need less than 1.5MB per image. This size can be reduced at the cost of some loss of performance. For instance with 20% reconstruction error, memory footprint drops to 0.8MB. This size can be compared to usual BoW signatures with 200.000 values, which also needs 0.8MB of storage using 32-bit float values. Furthermore, the size of packed VLATs is bounded by the number of extracted descriptors, whereas the size of standard VLAT always grows with the size of the codebook.

Let us note that this packing technique can be combined with other techniques. One can reduce the descriptors dimension though PCA, like we mentioned earlier. For instance, with the same example of 1024 clusters and descriptors reduced in size to 64 dimensions, the size of the standard VLAT is then about 2MB, and with a 20% reconstruction error, the size of a Packed VLAT is around 0.2MB. Size can be further reduced using post-processing techniques, like Kernel PCA and/or Product Quantizer [6]. In the former case, Packed VLAT can be used to speed up the processing, and signature of a few kB can be created. In the later case, very small signatures can be created (a few bytes).

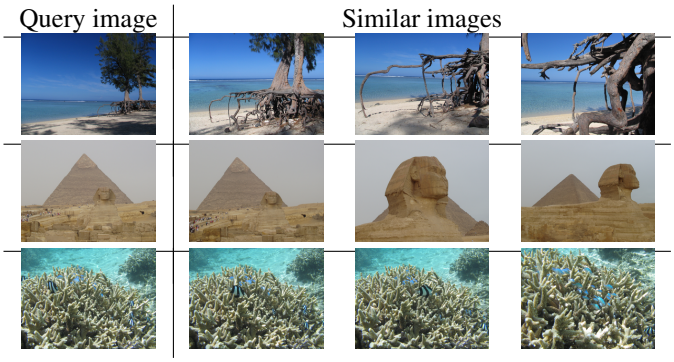


Figure 1: Images from Holidays database.

4. Experiments

We carried out experiments of both image search and image classification CBIR tasks on well known datasets.

4.1. Search task - Holidays dataset

The Holidays dataset [36] consists in 1491 images of holidays photographs (*cf.* Fig. 1). This set contains 500 categories, each one depicting a specific scene or object. The first image of each category is considered as a query, and the correct answers are the remaining images of the category. We used the same descriptors and experimental setup as in [37]. All parameters of the method (dictionaries and mean tensors) were trained using the 10,000 first images of the Flickr60k dataset. Images and SIFT descriptors from this dataset are also available on the holidays website.

The first results are displayed in Table 1, where we present the mean Average Precision (mAP) according to different codebooks and error rates using Packed VLATs and power normalization. Let us first focus on the last column, where results without any reconstruction error are presented. As we can see, we obtain very good performances, and with 1024 or more clusters we obtain results close to 75% of mAP. This is comparable to results of Hamming Embedding in [38] with the same descriptors. However, unlike [38], we use a simple dot product metric and no geometric constraint based re-ranking techniques. Now if we observe the effect of packing with reconstruction error, mAP decreases a bit with 10% and 20% error rates, and starts to drop with more than 30% reconstruction error.

We present in Figures 2(a) and 2(b) curves that show relations between the memory size of features and the corresponding performance. Both figures (a) and (b) show the mAP according to the average memory footprint of a Packed VLAT index for one image. The only difference between the two figures is the use of the power normalization (cf Eq. (13)). Each curve in these figures is with the same codebook, but with different error rates. The first point to the left of each curve is the result with 50% error rate, the next one with 40%, etc. The last point to the right of each curve is the result without any error.

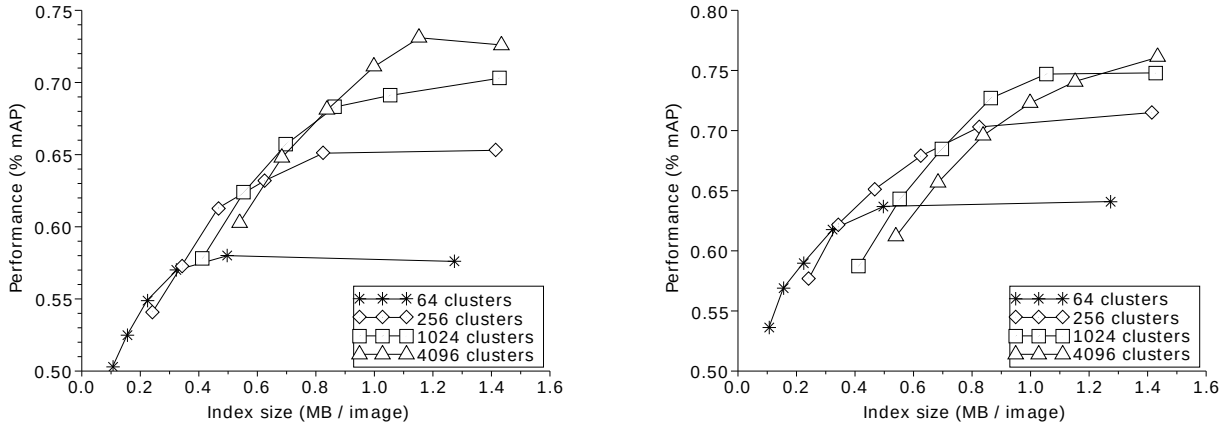
When error rate decreases, the size of indexes increases regularly before reaching a maximum. This maximum is related

	Error rate					
Dict. Size	50%	40%	30%	20%	10%	0%
64	53.6	56.9	59.0	61.8	63.7	64.1
256	57.7	62.2	65.1	67.9	70.3	71.5
1024	58.7	64.3	68.5	72.7	74.7	74.8
4096	61.2	65.7	69.6	72.3	74.1	**76.1**

Table 1: Mean Average Precision (%) using Packed Vectors of Locally Aggregated Tensors on Holidays database, with different codebook sizes and reconstruction error rates.

to the average size of extracted descriptors, which is around 1.35MB in these experiments (around 2,768 SIFT vectors per image using 32-bit float values). As we observed previously, there is a large mAP decrease with more than 30% of error (third dot from the left of each curve). We can also see that with 1024 clusters and higher, all parameters return close values: the mAP is about the same, the indexes size is near 1.45MB with no error and near 1MB with 10% error. If we consider power normalization, we observe a gain around 5%.

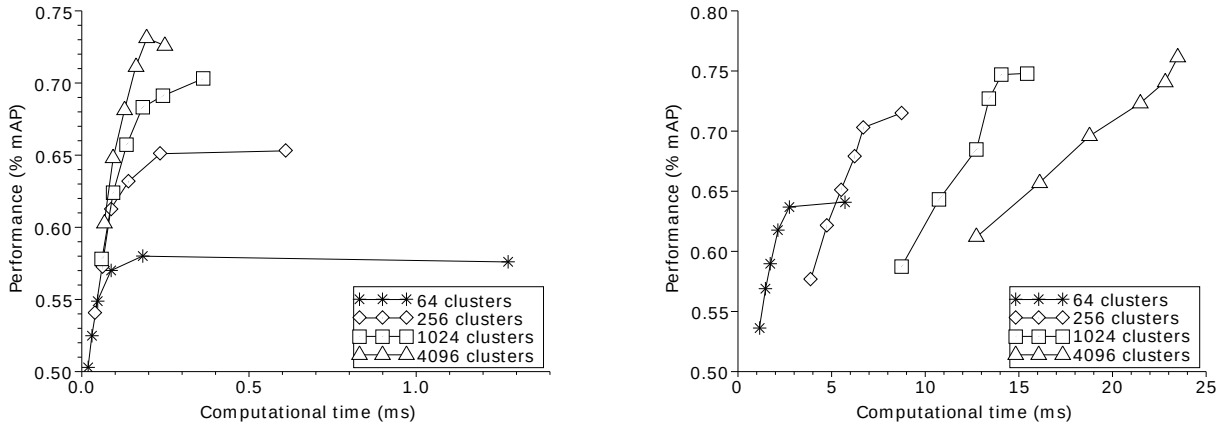
However, as one can see in Figures 3(c) and 3(d), the computational time is very different. These figures are similar to the previous ones, with the difference that mAP is presented according to the average computation time for the comparison of two image features in milliseconds. Without power normalization, we can use the computational trick we presented in section 3.1, Eq. (18) which saves a lot of computations, especially if we use large codebooks. However, with power normalization, we have to reconstruct the VLAT vector during image comparison, which leads to more computations. Let us note that these computational times are still very low when compared to kernels on bag computational times. For instance, the average computational time for the comparison of two images using the bag kernel of Eq. (1) with a Gaussian minor kernel is about 1000 ms.



(a) Without power normalization

(b) With power normalization

Figure 2: Mean Average Precision (%) using Packed Vectors of Locally Aggregated Tensors on Holidays database, according to the average size of image indexes.



(c) Without power normalization

(d) With power normalization

Figure 3: Mean Average Precision (%) using Packed Vectors of Locally Aggregated Tensors on Holidays database, according to the average computational time for the comparison of two image features.

Figure 4: Images from PASCAL Visual Object Classes Challenge 2007. *Categories where our method gives the best results.

4.2. Classification task - VOC 2007 database

The PASCAL-VOC challenge is one of the most used benchmark for the image classification task. In its 2007 revision, it consists in about 10,000 images and 20 categories [39], as shown on figure 4. The parameters of the method (dictionaries and mean tensors) are trained using the canonical *train* subset. These parameters are the same for all categories. For each category, a classifier is trained on the canonical *trainval* subset, and evaluated on the *test* subset. The standart performance measurement for this dataset is mean Average Precision (mAP) over the 20 classes.

4.2.1. Experimental setup

We extracted dense simplified SIFT features on a 2 pixels grid. Our implementation is close to that of the software provided by [34]. We used 4 different scales of SIFT with 4, 6, 8 and 10 pixels cell size respectively. We used a simple k-means clustering in order to obtain a codebook of 512 clusters. Prior to clustering, SIFT are compressed to 48 dimensions using PCA.

VLAT vectors are normalized using the power norm with $\alpha = 0.5$, then further ℓ_2 normalization is applied. The classification was done using linear SVM trained with a fast stochastic gradient descent algorithm provided by [40], for which the hyper-parameters (C and the number e of epochs) are set to default values ($C = 10$ and $e = 100$), regardless of the category.

4.2.2. Results

In table 2, we report results of Fisher Vectors (FK and IFK) taken from [33], linear local coding (LLC), supervectors (SV) and improved Fisher vectors (FK) taken from [34], and the winning method of VOC 2007 campain (VQ) along with our VLAT implementation. We also report in this table the presence of significant characteristics of compared methods, denoted by a $\times$ symbol. Namely, VQ and IFK from [33] used multiple features (*e.g.* sift and colors) to achieve these performances, whereas methods from [34] (FK, LLC and SV) use an ultra-dense SIFT sampling strategy (every 2 or 3 pixels) and a spatial pyramid scheme.

We achieve 60.8% mAP, which is less than FK, but better than recent methods like LLC and SV, which sounds promising for VLAT. In table 3, we report detailed comparison of VLAT with the bests methods of [34]. The VLAT approach manages to obtain the best results on categories *airplane*, *bicycle*, *bird*, *car*, *cat*, *dog*, *horse*, *person* and *train*.

5. Discussion

In the hereby section we discuss the relationship of VLAT to well known approaches.

5.1. Relation to Kernels on Bags

Using the dot product as minor kernel for the power kernel of Lyu et al., and with parameter $p = 2$, leads to the following similarity function:

$$K(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s \langle \mathbf{b}_{ri}, \mathbf{b}_{sj} \rangle^2 \tag{19}$$

This can be factorized as follows:

$$\begin{aligned} K(\mathbf{B}_i, \mathbf{B}_j) &= \sum_r \sum_s (\mathbf{b}_{ri}^\top \mathbf{b}_{sj})^2 && (20)\\ &= \sum_r \sum_s (\mathbf{b}_{ri} \mathbf{b}_{ri}^\top)^\top \mathbf{b}_{sj} \mathbf{b}_{sj}^\top && (21)\\ &= \left(\sum_r \mathbf{b}_{ri} \mathbf{b}_{ri}^\top \right)^\top \left(\sum_s \mathbf{b}_{sj} \mathbf{b}_{sj}^\top \right) && (22) \end{aligned}$$

This factorization is indeed the dot product applied to the sum of tensors products of all $\{\mathbf{b}_{ri}\}_r$. Hence, our method seems to be closely related to kernels on bags.

However, we observe two differences with our VLAT approach. Firstly, our approach splits the bag according to a clustering, and sums the contribution of each split. The equivalent kernel on bags would be the following:

$$K(\mathbf{B}_i, \mathbf{B}_j) = \sum_c \sum_r \sum_s \langle \mathbf{b}_{cri}, \mathbf{b}_{csj} \rangle^2 \tag{23}$$

The second difference lies in the centering applied to the tensors. This centering can be viewed as a affine transform in mapped space. Thus, VLAT appears to be a kernel on bags with the following explicit mapping:

$$K(\mathbf{B}_i, \mathbf{B}_j) = \sum_c \sum_r \sum_s \langle \mathbf{b}_{cri} \mathbf{b}_{cri}^\top - \mathcal{T}_c \,,\, \mathbf{b}_{csj} \mathbf{b}_{csj}^\top - \mathcal{T}_c \rangle \tag{24}$$

VLAT is thus the explicit linear writing of a highly non-linear kernel on bags. Although implicit mappings have been a major advantage allowing efficient learning algorithm with non-linear similarities using the kernel trick, explicit mappings have a clear computational advantage. They transfer the highest part

	VQ[39]	IFK [33]	FK [34]	LLC[34]	SV[34]	VLAT
NLK	×					
MF	×	×				
SP	×	×	×	×	×	
UDSS			×	×	×	×
mAP	**59.3**	**60.3**	**61.7**	**57.3**	**58.2**	**60.8**

Table 2: Mean Average Precision (%) of different methods. Results are reported from respective papers. For the methods, VQ: vector quantization, IFK: improved fisher vectors, FK: fisher Vectors, LLC: Linear Local Coding, and SV: Supervectors. For the configurations, NLK: Non-linear Kernel, MF: Multiple Features, SP: Spatial Pyramid, and UDSS: Ultra-dense Sift sampling (*i.e.* every 2 or 3 pixels).

	airplane	bicycle	bird	boat	bottle	bus	car
FK	79.0	67.4	51.9	**70.9**	**30.8**	**72.2**	79.9
SV	74.3	63.8	47.0	69.4	29.1	66.5	77.3
VLAT	**81.7**	**72.6**	**53.1**	69.5	27.4	68.9	**82.5**
	cat	chair	cow	table	dog	horse	mbike
FK	61.3	**56.0**	**49.6**	**58.4**	44.8	78.8	**70.8**
SV	60.2	50.2	46.5	51.9	44.1	77.9	67.1
VLAT	**61.4**	52.8	46.3	56.1	**48.1**	**80.8**	70.0
	person	plant	sheep	sofa	train	monitor	mean
FK	85.0	**31.7**	**51.0**	**56.4**	80.2	**57.5**	**61.7**
SV	83.1	27.6	48.5	51.1	75.5	52.3	58.2
VLAT	**86.4**	**31.7**	36.9	52.3	**81.8**	55.7	60.8

Table 3: Mean Average Precision (%) for Fisher vectors (FK), Supervectors (SV) and VLAT on all categories of VOC2007 dataset. Results are reported from [34].

of the cost from the online similarity computation to the offline signature calculation.

This relationship can explain the excellent behavior of VLAT in image search task, where descriptor matching methods achieve the best performances.

5.2. Relation to Fisher Vectors

In order to relate VLAT to Fisher Vectors (FV), let us consider the diagonal terms in equation 11:

$$\begin{aligned} \mathcal{T}_{i,c}[t,t] &= \sum_r (\mathbf{b}_{rci}[t] - \boldsymbol{\mu}_c[t])^2 - \boldsymbol{\sigma}_c[t]^2 \\ &= \boldsymbol{\sigma}_c[t]^2 \sum_r \left[\frac{(\mathbf{b}_{rci}[t] - \boldsymbol{\mu}_c[t])^2}{\boldsymbol{\sigma}_c[t]^2} - 1 \right] \end{aligned} \quad (25)$$

With $\boldsymbol{\sigma}_c$ being the standard deviation vector of cluster c. In these diagonal only terms, the main appearing differences with Fisher Vectors are the following: contributions are summed for all descriptors in FV, whereas they are summed only for the descriptors of the corresponding cluster in VLAT. Moreover, FV weight the contribution by the likelihood of descriptors to the corresponding cluster.

FV are also normalizing the resulting sum by the number of descriptors in the bags and the weight of the clusters, which is consistent with assigning descriptors to all clusters. In VLAT, as the assignment is only with respect to the corresponding cluster, there is no need to such normalization schemes. In fact, VLAT diagonal can be seen as a hard assignment variant of FV.

This close relationship between VLAT and FV can explain their comparative performances in image classification.

6. Conclusion

In this paper, we introduced a new efficient image signature in two different tasks on content based image retrieval, *i.e.* image search and image classification. Our scheme is based on the aggregation of tensor products of local descriptors named VLAT. We showed our VLAT approach to be closely related to kernels on bags methods which are known to be efficient for image search tasks. We also found relation between VLAT and Fisher vectors, which are known to be efficient in image classification tasks. We also carried out experiments on well known image datasets for these two tasks (namely VOC2007 and Holidays), and achieved near state of the art performances. These results highlight the ability of our signatures to be effective in similarity search and in classification. Moreover, we provide an easy generalization of our approach to higher order of tensors. Future works will investigate the extra information added by these higher orders.

References

[1] A. W. M. Smeulders, S. Member, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (2000) 1349–1380.

[2] L. Amsaleg, P. Gros, Content-based retrieval using local descriptors: problems and issues from a database perspective, Pattern Analysis and Applications 4 (2/3) (2001) 108–124.

[3] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, in: ACM Symposium on Theory of Computing, 1998, pp. 604–613.

[4] F. Akune, E. Valle, R. da Silva Torres, Monorail: A disk-friendly index for huge descriptor databases, in: IAPR International Conference on Pattern Recognition, 2010, pp. 4145–4148.

[5] A. Ólafsson, B. Pór Jónsson, L. Amsaleg, H. Lejsek, Dynamic behavior of balanced nv-trees, Multimedia Systems 17 (2011) 83–100, 10.1007/s00530-010-0199-4.
URL `http://dx.doi.org/10.1007/s00530-010-0199-4`

[6] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (1) (2011) 117–128.
URL `http://lear.inrialpes.fr/pubs/2011/JDS11`

[7] H. Lejsek, B. . Jnsson, L. Amsaleg, Nv-tree: Nearest neighbors at the billion scale, in: ACM International Conference on Multimedia Retrieval, Trento, Italie, 2011.

[8] C. Cortes, V. Vapnik, Support-vector networks, in: Machine Learning, 1995, pp. 273–297.

[9] S. Tong, D. Koller, Support vector machine active learning with application to text classification, International Journal on Machine Learning Research 2 (2001) 45–66.

[10] P. Gosselin, M. Cord, S. Philipp-Foliguet, Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, Computer Vision and Image Understanding 110 (3) (2008) 403–417.

[11] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, in: International Conference on Machine Learning, Bari, Italy, 1996, pp. 148–156.

[12] K. Tieu, P. Viola, Boosting image retrieval, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2000, pp. 228–235.

[13] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 2 (60) (2004) 91–110.

[14] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, Surf: Speeded up robust features, Computer Vision and Image Understanding 110 (3) (2008) 346–359.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. V. Gool, A comparison of affine region detectors, International Journal of Computer Vision 65 (1–2) (2005) 43–72.

[16] J. Shawe-Taylor, N. Cristianini, Kernel methods for Pattern Analysis, Cambridge University Press, ISBN 0-521-81397-2, 2004.

[17] S. Lyu, Mercer kernels for object recognition with local features, in: IEEE International Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005, pp. 223–229.

[18] P.-H. Gosselin, M. Cord, S. Philipp-Foliguet, Kernel on bags for multi-object database retrieval, in: ACM International Conference on Image and Video Retrieval, Amsterdam, Netherlands, 2007, pp. 226–231.
URL http://publi-etis.ensea.fr/2007/GCP07a

[19] D. Gorisse, M. Cord, F. Precioso, S. Philipp-Foliguet, Fast approximate kernel-based similarity search for image retrieval task, in: IAPR International Conference on Pattern Recognition, IEEE, USF, IEEE, 2008, pp. 1–4.
URL http://publi-etis.ensea.fr/2008/GCPP08

[20] J. Lebrun, P.-H. Gosselin, S. Philipp-Foliguet, Inexact graph matching based on kernels for object retrieval in image databases, Image and Vision Computing 29 (2011) 716–729. doi:10.1016/j.imavis.2011.07.008.

[21] J.-E. Haugeard, S. Philipp-Foliguet, P.-H. Gosselin, Kernel on graphs based on dictionary of paths for image retrieval, in: IAPR International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 2965–2968.
URL http://publi-etis.ensea.fr/2010/HPG10

[22] D. Gorisse, M. Cord, F. Precioso, Scalable active learning strategy for object category retrieval, in: International Conference on Image Processing, 2010, pp. 1013–1016.
URL http://publi-etis.ensea.fr/2010/GCP10

[23] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: IEEE International Conference on Computer Vision, Vol. 2, 2003, pp. 1470–1477.

[24] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 604–610.

[25] T. de Campos, G. Csurka, F. Perronnin, Images as Sets of Locally Weighted Features, Computer Vision and Image Understanding-doi:10.1016/j.cviu.2011.07.011.
URL http://dx.doi.org/10.1016/j.cviu.2011.07.011

[26] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, J. M. Geusebroek, Visual word ambiguity, IEEE Trans. on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1271–1283.
URL http://www.science.uva.nl/research/publications/2010/vanGemertTPAMI2010

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[28] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566. doi:10.1109/CVPR.2010.5539963.

[29] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2006, pp. 2169–2178. doi:http://dx.doi.org/10.1109/CVPR.2006.68.

[30] K. Grauman, T. Darrell, The pyramid match kernel: Efficient learning with sets of features, International Journal on Machine Learning Research 8 (2007) 725–760.

[31] F. Perronnin, C. R. Dance, Fisher kernels on visual vocabularies for image categorization, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[32] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. on Pattern Analysis and Machine IntelligenceQUAERO.
URL http://hal.inria.fr/inria-00633013/en/

[33] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision, 2010, pp. 143–156.

[34] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: British Machine Vision Conference, Vol. 76, 2011, pp. 1–12.

[35] D. Picard, P.-H. Gosselin, Improving image similarity with vectors of locally aggregated tensors, in: International Conference on Image Processing, Brussels, Belgique, 2011.
URL http://hal.archives-ouvertes.fr/hal-00591993/en/

[36] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, Springer, 2008, pp. 304–317.

[37] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311.

[38] M. Jain, H. Jégou, P. Gros, Asymmetric hamming embedding, in: ACM Multimedia, Scottsdale, United States, 2011.

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (2007).

[40] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Y. Lechevallier, G. Saporta (Eds.), International Conference on Computational Statistics, Springer, Paris, France, 2010, pp. 177–187.
URL http://leon.bottou.org/papers/bottou-2010