Noname manuscript No.
(will be inserted by the editor)

Aggregating Binary Local Descriptors for Image Retrieval

Giuseppe Amato · Fabrizio Falchi · Lucia Vadicamo

Received: date / Accepted: date

Abstract In this paper, we report an extensive comparison among state-of-the-art aggregation methods applied to binary features. Then, we mathematically formalize the application of Fisher Kernels to Bernoulli Mixture Models. Finally, we investigate the combination of the aggregated binary features with the emerging Convolutional Neural Network (CNN) features. Our results show that aggregation methods on binary features are effective and represent a worthwhile alternative to the direct matching. Moreover, the combination of the CNN with the Fisher Vector (FV) built upon binary features allowed us to obtain a relative improvement over the CNN results that is in line with that recently obtained using the combination of the CNN with the FV built upon SIFTs. The advantage is that the extraction process of binary features is about two order of magnitude faster than SIFTs.

Keywords Local binary feature · Fisher Vector · VLAD · Bag of Words · Convolutional Neural Network · Content-Based Image Retrieval

1 Introduction

Content-Based Image Retrieval (CBIR) is a relevant topic studied by many scientists in the last decades. CBIR refers to the possibility of organizing archives containing digital pictures, so that they can be searched and retrieved by using their visual content [18]. A specialization of the basic CBIR techniques include the techniques of object recognition [72], where visual content of images is analyzed so that objects contained in digital pictures are recognized, and/or images containing specific objects are retrieved. Techniques of CBIR and object recognition are becoming increasingly popular in many web search engines, where images can be searched by using their

G. Amato, F. Falchi, L. Vadicamo
Institute of Information Science and Technologies (ISTI) - CNR
Via Moruzzi 1, 56124 Pisa (Italy)
E-mail: {firstname}.{lastname}@isti.cnr.it

visual content [3, 1], and on smartphones apps, where information can be obtained by pointing the smartphone camera toward a monument, a painting, a logo [2].

During the last few years, local descriptors, as for instance SIFT [48], SURF [10], BRISK [44], ORB [59], to cite some, have been widely used to support effective CBIR and object recognition tasks. A local descriptor is generally a histogram representing statistics of the pixels in the neighborhood of an interest point (automatically) chosen in an image. Among the promising properties offered by local descriptors, we mention the possibility to help mitigating the so called *semantic gap* [66], that is the gap between the visual representation of images and the semantic content of images. In most cases visual similarity does not imply semantic similarity.

Executing image retrieval and object recognition tasks, relying on local features, is generally resource demanding. Each digital image, both queries and images in the digital archives, are typically described by thousands of local descriptors. In order to decide that two images match, since they contain the same or similar objects, local descriptors in the two images need to be compared, in order to identify matching patterns. This poses some problems when local descriptors are used on devices with low resources, as for instance smartphones, or when response time must be very fast even in presence of huge digital archives. On one hand, the cost for extracting local descriptors, storing all descriptors of all images, and performing feature matching between two images must be reduced to allow their interactive use on devices with limited resources. On the other hand, compact representation of local descriptors and ad hoc index structures for similarity matching [79] are needed to allow image retrieval to scale up with very large digital picture archives. These issues have been addressed by following two different directions.

To reduce the cost of extracting, representing, and matching local visual descriptors, researchers have investigated the use binary local descriptors, as of instance BRISK and ORB. Binary features are built from a set of pairwise intensity comparisons. Thus, each bit of the descriptors is the result of exactly one comparison. Binary descriptors are much faster to be extracted, are obviously more compact than non-binary ones, and can also be matched faster by using the Hamming distance [27] rather than the Euclidean distance. For example, in [59] it has been showed that ORB is an order of magnitude faster than SURF, and over two orders faster than SIFT. However, note that even if binary local descriptors are compact, each image is still associated with thousand local descriptors, making it difficult to scale up to very large digital archives.

Reduction of the cost of image matching on a very large scale has been obtained by using methods for quantizing and/or aggregating local features. On one hand, quantization methods, as for instance the Bag-of-Words approach (BoW) [65], define a finite vocabulary of “visual words”, that is a finite set of local descriptors to be used as representative. Every possible local descriptors is thus represented by its closest visual word, that is the closest element of the vocabulary. In this way images are described by a set (a bag) of identifiers of representatives, rather than a set of histograms. On the other hand, aggregation methods, as for instance Fisher Vectors (FV) [52] or Vectors of Locally Aggregated Descriptors (VLAD) [34], analyze the local descriptors contained in an image to create statistical summaries that still preserve the effectiveness power of local descriptors and allow treating them as global descriptors. In both cases index structures for approximate or similarity matching [79] can be used to guarantee scalability on very large datasets. However, given that these methods are basically defined on non-binary features, the cost of extracting local descriptors and to quantize/aggregate them on the fly, is still high, for some applications. Recently, some approaches that attempt to integrate the binary local descriptors with the quantization and aggregation methods have been proposed in literature. In these proposals, aggregation and quantization methods have been directly applied on top of binary local descriptors. The objective is to leverage on the advantages of both approaches, by reducing, or eliminating the disadvantages.

The contribution of this paper is providing an extensive comparisons and analysis of the aggregation and quantization methods applied to binary local descriptors also providing a novel formulation of Fisher Vectors built using the Bernoulli Mixture model (BMM). Moreover, we investigate the combination of FVs built upon binary features with the Convolutional Neural Network [58] features as other case of use of binary feature aggregations. We focus on cases where, for efficiency issues [59, 28], the binary features are extracted and used to represent images. Thus, we compare aggregations of binary features in order to find the most suitable techniques to avoid the direct matching. We expect this topic to be relevant for application that uses binary features on devices with low CPU and memory resources, as for instance mobile and wearable devices. In these cases the combination of aggregation methods with binary local features is very useful and led to scale up image search on large scale, where direct matching is not feasible.

This paper extends our early work on aggregations of binary features [5] by a) providing a formulation of the Fisher Vector (BMM-FV) built using the Bernoulli Mixture Model (BMM) which preserve the structure of the traditional FV built using a Gaussian Mixture Model (existing implementations of the FV can be easily adapted to work also with BMMs); b) comparison of the BMM-FV against the other state-of-the-art aggregation approaches on two standard benchmarks (INRIA Holidays[1] [31] and Oxford5k [56]); c) evaluation of the BMM-FV on the top of several binary local features (ORB [59], LATCH [45], AKAZE [4]) whose performances have not been yet reported on benchmark for image retrieval; d) evaluation of the combination of the BMM-FV with the emerging Convolutional Neural Network (CNN) features. The results of our experiments show that the use of aggregation and quantization methods with binary local descriptors is generally effective even if, as expected, retrieval performance is worse than that obtained applying the same aggregation and quantization methods directly to non-binary features. The BMM-FV approach provided us with performance results that are better than all the other aggregation methods on binary descriptors. In addition, our results show that some aggregation methods led to obtain very compact image representation with a retrieval performance comparable to the direct matching, which actually is the most used approch to evaluate the similarity of images described by binary local features. Moreover, we show that the combination of BMM-FV and CNN improve the latter retrieval performances and achieves effectiveness comparable with that obtained combining CNN and FV built upon SIFTs,

[1] Respect to the experimental setting used in our previous work[5], we improved the computation of the local features before the aggregation phase which allowed us to obtain better performances for BoW and VLAD on the INRIA Holidays dataset than that reported in [5].

previous proposed in [14]. The advantage is that BMM-FV relies on binary features whose extraction is noticeably faster than SIFT extraction.

The paper is organized as follows. Section 2 offers an overview of other articles in literature, related to local features, binary local features, and aggregation methods. Section 3 discusses how existing aggregation methods can be used with binary local features. It also contains our approach for applying Fisher Vectors on binary local features and how combining it with the CNN features. Section 4 discusses the evaluation experiments and the obtained results. Section 5 concludes.

2 Related Work

The research for effective representation of visual feature for images has received much attention over the last two decades. The use of *local features*, such as SIFT [48] and SURF [10], is at the core of many computer vision applications, since it allows systems to efficiently match local structures between images. To date, the most used and cited local feature is the Scale Invariant Feature Transformation (SIFT) [48]. The success of SIFT is due to its distinctiveness that enable to effectively find correct matches between images. However, the SIFTs extraction is costly due to the local image gradient computations. In [10] integral images were used to speed up the computation and the SURF feature was proposed as an efficient approximation of the SIFT. To further reduce the cost of extracting, representing, and matching local visual descriptors, researchers have investigated the *binary local descriptors*. These features have a compact binary representation that is not the result of a quantization, but rather is computed directly from pixel-intensity comparisons. One of the early studies in this direction was the Binary Robust Independent Elementary Features (BRIEF) [13]. In [59] Rublee et al. proposed a binary feature, called ORB (Oriented FAST and Rotated BRIEF), whose extraction process is an order of magnitude faster than SURF, and two orders faster than SIFT according to the experimental results reported in [59,50,28]. Recently, several other binary local features have been proposed, such as BRISK [44], AKAZE [4], and LATCH [45].

Local features have been widely used in literature and applications, however since each image is represented by thousands of local features there is a significant amount of memory consumption and time required to compare local features within large databases. The use of the information provided by each local feature is crucial for some tasks such as image stitching and 3D reconstruction. For other tasks such as image classification and retrieval high effectiveness have been achieved using the *aggregation techniques* which provide meaningful summarization of all the extracted feature of an image. One profitable outcome of using aggregation techniques is that each image is represented by a single descriptor rather than thousands descriptors. This reduces the cost of image comparison and leads to scale up the search to large database.

By far, the most popular aggregation method has been the Bag-of-Word (BoW) [65]. BoW was initially proposed for matching object in video and has been studied in many other papers, such as [17,56,32,34], for classification and CBIR tasks. BoW uses a visual vocabulary to quantize the local descriptors extracted from images; each image is then represented by a histogram of occurrences of visual words. The BoW approach used in computer vision is very similar to the BoW used in natural language processing and information retrieval [60], thus many text indexing techniques, such as inverted files [76], have been applied for image search. Search results obtained using BoW in CBIR have been improved by exploiting additional geometrical information [56, 51, 68, 81], applying re-ranking approaches [56, 31, 16, 70] or using better encoding techniques, such as the Hamming Embedding [31], soft/multiple-assignment [57, 23, 32], sparse coding [77, 12], locality-constrained linear coding [75] and spatial pyramids [41].

Recently, alternative encodings schemes, like the Fisher Vectors (FVs) [52] and the Vector of Locally Aggregated Descriptors (VLAD) [34], have attracted much attention because of their effectiveness in both image classification and large-scale image search. The FV uses the Fisher Kernel framework [30] to transform an incoming set of descriptors into a fixed-size vector representation. The basic idea is to characterize how a sample of descriptors deviates from an average distribution that is modeled by a parametric generative model. The Gaussian Mixture Model (GMM) [49] is typically used as generative model and might be understood as a "probabilistic visual vocabulary". While BoW counts the occurrences of visual words and so takes in account just 0-order statistics, the FV offers a more complete representation by encoding higher order statistics (first, and optionally second order) related to the distribution of the descriptors. The FV results also in a more efficient representation, since fewer visual words are required in order to achieve a given performance. However, the vector representation obtained using BoW is typically quite sparse while that obtained using the Fisher Kernel is almost dense. This leads to some storage and input/output issues that have been addressed by using techniques of dimensionality reduction, such as Principal Component Analysis (PCA) [11], compression with product quantization [26, 33] and binary codes [54]. In [14] a fusion of FV and CNN features[58, 21] was proposed and other works [53, 63, 67] have started exploring the combination of FVs and CNNs by defining hybrid architectures.

The VLAD method, similarly to BoW, starts with the quantization of the local descriptors of an image by using a visual vocabulary learned by k-means. Differently from BoW, VLAD encodes the accumulated difference between the visual words and the associated descriptors, rather than just the number of descriptors assigned to each visual word. Thus, VLAD exploits more aspects of the distribution of the descriptors assigned to a visual word. As highlighted in [35], VLAD might be viewed as a simplified non-probabilistic version of the FV. In the original scheme [34], as for the FV, VLAD was L_2-normalized. Subsequently a power normalization step was introduced for both VLAD and FV [35, 54]. Furthermore, PCA dimensionality reduction and product quantization were applied and several enhancements to the basic VLAD were proposed [8, 15, 19, 81].

The aggregation methods have been defined and used almost exclusively in conjunction with local features that have a real-valued representation, such as SIFT and SURF. Few articles have addressed the problem of modifying the state-of-the-art aggregation methods to work with the emerging binary local features. In [22, 80, 25, 43], the use of ORB descriptors was integrated into the BoW model by using different clustering algorithms. In [22] the visual vocabulary was calculated by binarizing the centroids obtained using the standard k-means. In [80, 25, 43] the k-means clustering was modified to fit the binary features by replacing the Euclidean distance with the Hamming distance, and by replacing the mean operation with the median operation. In [74] the VLAD image signature was adapted to work with binary descriptors: k-means is used for learning the visual vocabulary and the VLAD vectors are computed in conjunction with an intra-normalization and a final binarization step. Recently, also the FV scheme has been adapted for the use with binary descriptors: Uchida et al. [71] derived a FV where the Bernoulli Mixture Model (BMM) was used instead of the GMM to model binary descriptors, while Sanchez and Redolfi [62] generalized the FV formalism to a broader family of distributions, known as the exponential family, that encompasses the Bernoulli distribution as well as the Gaussian one.

3 Image Representations

In order to decide if two images contain the same object or have a similar visual content, one needs an appropriate mathematical description of each image. In this section, we describe some of the most prominent approaches to transform an input image into a numerical descriptor. First we describe the principal aggregation techniques and the application of them to binary local features. Then, the emerging CNN features are presented.

3.1 Aggregation of local features

In the following we review how quantization and aggregation methods have been adapted to cope with binary features. Specifically we present the BoW [65], the VLAD [34] and the FV [52] approaches.

3.1.1 Bag-of-Words

The Bag of (Visual) Words (BoW) [65] uses a visual vocabulary to group together the local descriptors of an image and represent each image as a set (bag) of visual words. The visual vocabulary is built by clustering the local descriptors of a dataset, e.g. by using k-means [47]. The cluster centers, named *centroids*, act as the *visual words* of the vocabulary and they are used to quantize the local descriptors extracted from the images. Specifically, each local descriptor of an image is assigned to its closest centroid and the image is represented by a histogram of occurrences of the visual words. The retrieval phase is performed using text retrieval techniques, where visual words are used in place of text word and considering a query image as disjunctive term-query. Typically, the cosine similarity measure in conjunction with a term weighting scheme, e.g. term frequency-inverse document frequency (tf-idf), is adopted for evaluating the similarity between any two images.

BoW and Binary Local Features In order to extend the BoW scheme to deal with binary features we need a cluster algorithm able to deal with binary strings and Hamming distance. The *k-medoids* [37] are suitable for this scope, but they requires a computational effort to calculate a full distance matrix between the elements of each cluster. In [25] it was proposed to use a voting scheme, named *k-majority*, to process a collection of binary vectors and seek for a set of good centroids, that will become the visual words of the BoW model. An equivalent representation is given also in [80, 43], where the BoW model and the *k*-means clustering have been modified to fit the binary features by replacing the Euclidean distance with the Hamming distance, and by replacing the mean operation with the median operation.

3.1.2 Vector of Locally Aggregated Descriptors

The Vector of Locally Aggregated Descriptors (VLAD) was initially proposed in [34]. As for the BoW, a visual vocabulary $\{\mu_1, \ldots, \mu_K\}$ is first learned using a clustering algorithm (e.g. *k*-means). Then each local descriptor x_t of a given image is associated with its nearest visual word $NN(x_t)$ in the vocabulary and for each centroid μ_i the differences $x_t - \mu_i$ of the vectors x_t assigned to μ_i are accumulated: $v_i = \sum_{x_t : NN(x_t)=\mu_i} x_t - \mu_i$. The VLAD is the concatenation of the residual vectors v_i, i.e. $V = [v_1^\top \ldots v_K^\top]$. All the residuals have the same size D which is equal to the size of the used local features. Thus the dimensionality of the whole vector V is fixed too and it is equal to DK.

VLAD and Binary Local Features A naive way to apply the VLAD scheme to binary local descriptors is treating binary vectors as a particular case of real-valued vectors. In this way, the *k-means* algorithm can be used to build the visual vocabulary and the difference between the centroids and the descriptors can be accumulated as usual. This approach has also been used in [74], where a variation to the VLAD image signature, called BVLAD, has been defined to work with binary features. Specifically, the BVLAD is the binarization (by thresholding) of a VLAD obtained using power-law, intra-normalization, L_2 normalization and multiple PCA. Thereafter we have not evaluated the performance of the BVLAD because the binarization of the final image signature is out of the scope of this paper.

Similarly to BoW, various binary-cluster algorithms (e.g. *k-medoids* and *k-majority*) and the Hamming distance can be used to build the visual vocabulary and associate each binary descriptor to its nearest visual word. However, as we will see, the use of binary centroid may provide less discriminant information during the computation of the residual vectors.

3.1.3 Fisher Vector

The Fisher Kernel [30] is a powerful framework adopted in the context of image classification in [52] as efficient tool to encode image local descriptors into a fixed-size vector representation. The main idea is to derive a kernel function to measure the similarity between two sets of data, such as the sets of local descriptors extracted from two images. The similarity of two sample sets X and Y is measured by analyzing the difference between the statistical properties of X and Y, rather than comparing directly X and Y. To this scope a probability distribution $p(\cdot|\lambda)$ with some parameters $\lambda \in \mathbb{R}^m$ is first estimated on a training set and it is used as generative model over the the space of all the possible data observations. Then each set X of observations is represented by a vector, named *Fisher Vector*, that indicates the direction in which the parameter λ of the probability distribution $p(\cdot|\lambda)$ should be modified to best fit the data in X. In this way, two samples are considered similar if the directions given by their respective Fisher Vectors are similar. Specifically, as proposed in [30], the similarity between two sample sets X and Y is measured using the *Fisher Kernel*, defined as $K(X,Y) = (G_\lambda^X)^\top F_\lambda^{-1} G_\lambda^Y$, where F_λ is the *Fisher Information Matrix* (FIM) and $G_\lambda^X = \nabla_\lambda \log p(X|\lambda)$ is referred to as the *score function*.

The computation of the Fisher Kernel is costly due the multiplication by the inverse of the FIM. However, by using the Cholesky decomposition $F_\lambda^{-1} = L_\lambda^\top L_\lambda$, it is possible to re-written the Fisher Kernel as an Euclidean dot-product, i.e. $K(X,Y) = (\mathscr{G}_\lambda^X)^\top \mathscr{G}_\lambda^Y$, where $\mathscr{G}_\lambda^X = L_\lambda G_\lambda^X$ is the *Fisher Vector* (FV) of X [54].

Note that the FV is a fixed size vector whose dimensionality only depends on the dimensionality m of the parameter λ. The FV is further divided by $|X|$ in order to avoid the dependence on the sample size [61] and L_2-normalized because, as proved in [55,61], this is a way to cancel-out the fact that different images contain different amounts of image-specific information (e.g. the same object at different scales).

The distribution $p(\cdot|\lambda)$, which models the generative process in the space of the data observation, can be chosen in various way. The Gaussian Mixture Model (GMM) is typically used to model the distribution of non-binary features considering that, as pointed in [49], any continuous distribution can be approximated arbitrarily well by an appropriate finite Gaussian mixture. Since the Bernoulli distribution models an experiment that has only two possible outcomes (0 and 1), a reasonable alternative to characterize the distribution of a set of binary features is to use a Bernoulli Mixture Model (BMM).

FV and Binary Local Features In this work we derive and test an extension of the FV built using BMM, called *BMM-FV*, to encode binary features. Specifically, we chose $p(\cdot|\lambda)$ to be multivariate Bernoulli mixture with K components and parameters $\lambda = \{w_k, \mu_{kd}, k = 1,\dots,K, d = 1,\dots,D\}$:

$$p(x_t|\lambda) = \sum_{k=1}^{K} w_k p_k(x_t) \tag{1}$$

where

$$p_k(x_t) = \prod_{d=1}^{D} \mu_{kd}^{x_{td}} (1-\mu_{kd})^{1-x_{td}} \tag{2}$$

and

$$\sum_{k=1}^{K} w_k = 1, \quad w_k > 0 \qquad \forall k = 1,\dots,K. \tag{3}$$

To avoid enforcing explicitly the constraints in (3), we used the soft-max formalism [38,61] for the weight parameters: $w_k = \exp(\alpha_k)/\sum_{i=1}^{K}\exp(\alpha_i)$.

Given a set $X = \{x_t, t = 1, \dots, T\}$ of D-dimensional binary vectors $x_t \in \{0,1\}^D$ and assuming that the samples are independent we have that the score vector G^X_λ with respect to the parameter $\lambda = \{\alpha_k, \mu_{kd}, k = 1, \dots, K, d = 1, \dots, D\}$ is calculated (see Appendix A) as the concatenation of

$$G^X_{\alpha_k} = \sum_{t=1}^{T} \frac{\partial \log p(x_t|\lambda)}{\partial \alpha_k} = \sum_{t=1}^{T} (\gamma_t(k) - w_k)$$
$$G^X_{\mu_{kd}} = \sum_{t=1}^{T} \frac{\partial \log p(x_t|\lambda)}{\partial \mu_{kd}} = \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_{td} - \mu_{kd}}{\mu_{kd}(1-\mu_{kd})} \right)$$

where $\gamma_t(k) = p(k|x_t, \lambda)$ is the *occupancy probability* (or posterior probability). The occupancy probability $\gamma_t(k)$ represents the probability for the observation x_t to be generated by the k-th Bernoulli and it is calculated as $\gamma_t(k) = w_k p_k(x_t) / \sum_{j=1}^{K} w_j p_j(x_t)$.

The FV of X is then obtained by normalizing the score G^X_λ by the matrix L_λ, which is the square root of the inverse of the FIM, and by the sample size T. In the Appendix B we provide an approximation of FIM under the assumption that the occupancy probability $\gamma_t(k)$ is sharply peaked on a single value of k for each descriptor x_t, obtained following an approach very similar to that used in [61] for the GMM case. By using our FIM approximation, we got the following normalized gradient:

$$\mathscr{G}^X_{\alpha_k} = \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} (\gamma_t(k) - w_k)$$
$$\mathscr{G}^X_{\mu_{kd}} = \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_{td} - \mu_{kd}}{\sqrt{\mu_{kd}(1-\mu_{kd})}} \right)$$

The final BMM-FV is the concatenation of $\mathscr{G}^X_{\alpha_k}$ and $\mathscr{G}^X_{\mu_{kd}}$ for $k = 1, \dots, K$, $d = 1, \dots, D$ and is therefore of dimension $K(D+1)$.

An extension of the FV by using the BMM has been also carried in [71,62]. Our approach differs from the one proposed in [71] in the approximation of the square root of the inverse of the FIM (i.e., L_λ) . It is worth noting that our formalization preserves the structure of the traditional FV derived by using the GMM, where Gaussian means and variances are replaced by Bernoulli means μ_{kd} and variances $\mu_{kd}(1-\mu_{kd})$ (see Table 1).

In [62], the FV formalism was generalized to a broaden family of distributions knows as *exponential family* that encompasses the Bernoulli distribution as well as the Gaussian one. However, [62] lacks in an explicit definition of the FV and of the FIM approximation in the case of BMM which was out of the scope of their work. Our formulation differs from that of [62] in the choice of the parameters used in the gradient computation of the score function [2]. A similar difference holds also for the FV computed on the GMM, given that in [62] the score function is computed w.r.t. the natural parameters of the Gaussian distribution rather than the mean and the variance

[2] A Bernoulli distribution $p(x) = \mu^x(1-\mu)^{1-x}$ of parameter μ can be written as exponential distribution $p(x) = exp(\eta x - log(1+e^\eta))$ where $\eta = \log(\frac{\mu}{1-\mu})$ is the *natural parameter*. In [62] the score function is computed considering the gradient w.r.t. the natural parameters η while in this paper we used the gradient w.r.t. the standard parameter μ of the Bernoulli (as also done in [71]).

Table 1 Comparison of the structure of the FVs derived using BMM with that derived using GMM. Parameters for BMM are $\lambda^B = \{w_k^B, \mu_{kd}^B, k = 1, \dots, K, d = 1, \dots, D\}$ and for GMM are $\lambda^G = \{w_k^G, \mu_{kd}^G, \Sigma_k^G = \mathrm{diag}(\sigma_{k1}^G, \dots, \sigma_{kD}^G,), k = 1, \dots, K, d = 1, \dots, D\}$, where w_k^B, μ_k^B are the mixture weight and the mean vector of the k-th Bernoulli and w_k^G μ_k^G, Σ_k^G are respectively the mixture weight, mean vector and covariance matrix of Gaussian k.

GMM-FV [61]
$\mathscr{G}_{\alpha_k^G}^X = \frac{1}{T\sqrt{w_k^G}} \sum_{t=1}^{T} \left(\gamma_t^G(k) - w_k^G \right)$ $\mathscr{G}_{\mu_{kd}^G}^X = \frac{1}{T\sqrt{w_k^G}} \sum_{t=1}^{T} \gamma_t^G(k) \frac{x_{td} - \mu_{kd}^G}{\sigma_{kd}^G}$ $\mathscr{G}_{\sigma_{kd}^G}^X = \frac{1}{T\sqrt{w_k^G}} \sum_{t=1}^{T} \gamma_t^G(k) \frac{1}{\sqrt{2}} \left[\frac{(x_{td} - \mu_{kd}^G)^2}{(\sigma_{kd}^G)^2} - 1 \right]$
BMM-FV (our formalization)
$\mathscr{G}_{\alpha_k^B}^X = \frac{1}{T\sqrt{w_k^B}} \sum_{t=1}^{T} \left(\gamma_t^B(k) - w_k^B \right)$ $\mathscr{G}_{\mu_{kd}^B}^X = \frac{1}{T\sqrt{w_k^B}} \sum_{t=1}^{T} \gamma_t^B(k) \frac{x_{td} - \mu_{kd}^B}{\sqrt{\mu_{kd}^B(1 - \mu_{kd}^B)}}$
BMM-FV (Uchida et. al [71])
($\mathscr{G}_{\alpha_k^B}^X$ not explicitly derived in [71]) $\mathscr{G}_{\mu_{kd}^B}^X = \frac{\sum_{t=1}^{T} \gamma_t(k) \frac{(-1)^{1-x_{td}}}{(\mu_{kd}^B)^{x_{td}} \left(1 - \mu_{kd}^B\right)^{1-x_{td}}}}{T\sqrt{Tw_k^B \left(\frac{\sum_{i=1}^{K} w_i^B \mu_{id}^B}{(\mu_{kd}^B)^2} + \frac{\sum_{i=1}^{K} w_i^B (1 - \mu_{id}^B)}{(1 - \mu_{kd}^B)^2} \right)}}$

parameters which are typically used in literature for the FV representation[54, 52, 61]. Unfortunately, the authors of [62] didn't experimentally compare the FVs obtained using or not the natural parameters.

Sànchez [61] highlights that the FV derived from GMM can be computed in terms of the following 0-order and 1-order statistics: $S_k^0 = \sum_{t=1}^{T} \gamma_t(k) \in \mathbb{R}$, $S_k^1 = \sum_{t=1}^{T} \gamma_t(k) x_t \in \mathbb{R}^D$. Our BMM-FV can be also written in terms of these statistics as

$$
\begin{aligned}
\mathscr{G}_{\alpha_k}^X &= \frac{1}{T\sqrt{w_k}} (S_k^0 - T w_k) \\
\mathscr{G}_{\mu_{kd}}^X &= \frac{S_{kd}^1 - \mu_{kd} S_k^0}{T\sqrt{w_k \mu_{kd}(1 - \mu_{kd})}}.
\end{aligned}
$$

We finally used power-law and L_2 normalization to improve the effectiveness of the BMM-FV approach.

3.2 Combination of BMM-FV and Convolutional Neural Network Features

Convolutional neural networks (CNNs)[42] have brought breakthroughs in the computer vision area by improving the state-of-the-art in several domains, such as image retrieval, image classification, object recognition, and action recognition. Depp CNN allows a machine to automatically learn representations of data with multiple levels of abstraction which can be used for detection or classification tasks. CNNs are neural networks specialized for data that has a grid-like topology as image data. The applied discrete convolution operation results in a multiplication by a matrix which has several entries constrained to be equal to other entries. Three important ideas are behind the success CNNs: sparse connectivity, parameter sharing, and equivariant representations [24].

In image retrieval, the activations produced by an image within the top layers of the CNN have been successfully used as a high-level descriptors of the visual content of the image [21]. The results reported in [58] shows that these CNN features, compared by using the Euclidean distance, achieve state-of-the-art quality in terms of mAP. Most of the papers reporting results obtained using the CNN features maintain the Rectified Linear Unit (ReLU) transform [21,58,14], i.e., negative activations values are discarded replacing them with 0. Values are typically L_2 normalized [9, 58,14] and we did the same in this work. In Section 4.2 we describe the CNN model used in our experiments.

Recently, in [14] it has been shown that the information provided by the FV built upon SIFT helps to further improve the retrieval performance of the CNN features and a combination of FV and CNN features has been used as well [14,6]. However, the benefits of such combinations are clouded by the cost of extracting SIFTs that can be considered to high with respect to the cost of computing the CNN features (see Table 2). Since the extraction of binary local features is up two times faster than SIFT, in this work we also investigate the combination of CNN features with the BMM-FV built upon binary local features.

We combined BMM-FV and CNN using the following approach. Each image was represented by a couple (c, f), where c and f were respectively the CNN descriptor and the BMM-FV of the image. Then, we evaluated the distance d between two couples (c_1, f_1) and (c_2, f_2) as the convex combination between the L_2 distances of the CNN descriptors (i.e. $\|c_1 - c_2\|_2$) and the BMM-FV descriptors (i.e. $\|f_1 - f_2\|_2$). In other words, we defined the distance between two couples (c_1, f_1) and (c_2, f_2) as

$$d\big((c_1, f_1),(c_2, f_2)\big) = \alpha \, \|c_1 - c_2\|_2 + (1-\alpha)\, \|f_1 - f_2\|_2 \tag{4}$$

with $0 \leq \alpha \leq 1$. Choosing $\alpha = 0$ corresponds to use only FV approach, while $\alpha = 1$ correspond to use only CNN features. Please note that in our case both the FV and the CNN features are L_2 normalized.

4 Experiments

In this section we evaluate and compare the performance of the techniques described in this paper to aggregate binary local descriptors. Specifically, in the Subsection 4.3

Table 2 Average time costs for computing various image representations using a CPU implementation. The cost of computing the CNN feature of an image was estimated using pre-learned AlexNet model and the Caffe framework [36]. The values related to the FV refers only to the cost of aggregating the local descriptors of an image into a single vector and do not encompass the cost of extracting the local features, neither the learning of the Gaussian or the Bernoulli Mixture Model which is calculated off-line. The cost of computing FV varies with the number K of mixtures of Gaussian/Bernoulli; we reported the approximate cost for $K = 64$ and $K = 256$. The cost of SIFT/ORB local feature extraction was estimated according to [28] by considering about $2,000$ features per image.

	CNN	FV Encoding	SIFT	ORB
Computing time per image	∼300 ms	∼40 ms [K=64] ∼160 ms [K=256]	∼1200 ms	∼26 ms

we compare the BoW, the VLAD, the FV based on the GMM, and the BMM-FV approach to aggregate ORB binary features. Since the BMM-FV achieved the best results over the other tested approaches, in the Subsection 4.4 we further evaluate the performance of the BMM-FVs using different binary features (ORB, LATCH, AKAZE) and combining them with the CNN features.

In the following, we first introduce the datasets used in the evaluations (Subsection 4.1) and we describe our experimental setup (Subsection 4.2). We then report results and their analysis.

4.1 Datasets

The experiments were conducted using two benchmark datasets, namely *INRIA Holidays* [31] and *Oxford5k* [56], that are publicly available and often used in the context of image retrieval [34, 81, 31, 7, 54, 35, 69].

INRIA Holidays [31] is a collection of 1 491 images which mainly contains personal holidays photos. The images are of high resolution and represent a large variety of scene type (natural, man-made, water, fire effects, etc). The dataset contains 500 queries, each of which represents a distinct scene or object. For each query a list of positive results is provided. As done by the authors of the dataset, we resized the images to a maximum of 786 432 pixels (768 pixels for the smaller dimension) before computing the local descriptors.

Oxford5k [56] consists of 5 062 images collected from Flickr. The dataset comprise 11 distinct Oxford buildings together with distractors. There are 55 query images: 5 queries for each building. The collection is provided with a comprehensive ground truth. For each query there are four image sets: *Good* (clear pictures of the object represented in the query), *OK* (images where more that 25% of the object is clearly visible), *Bad* (images where the object is not present) and *Junk* (images where less than 25% of the object is visible or images with high level of distortion).

As in many other articles, e.g. [34, 31, 57, 35], all the learning stages (clustering, etc.) were performed off-line using independent image collections. *Flickr60k* dataset [31] was used as training set for INRIA Holidays. It is composed of 67 714 images extracted randomly from Flickr. The experiments on Oxford5k were conducted performing the learning stages on *Paris6k* dataset [57], that contains 6 300 high resolution images obtained from Flickr by searching for famous Paris landmarks.

4.2 Experimental settings

In the following we report some details on how the features for the various approaches were extracted.

Local features. In the experiments we used ORB [59], LATCH [45], and AKAZE [4] binary local features that we extracted by using OpenCV (Open Source Computer Vision Library)[3]. We detected up to 2 000 local features per image.

Visual Vocabularies and Bernoulli/Gaussian Mixture Models. The visual vocabularies used for building the BoW and VLAD representations were computed using several clustering algorithms, i.e. k-medoids, k-majority and k-means. The k-means algorithm was applied to the binary features by treating the binary vectors as real-valued vectors. The parameters $\lambda^B = \{w_k^B, \mu_{kd}^B\}_{k=1,\dots,K,d=1,\dots,D}$ of the BMM and $\lambda^G = \{w_k^G, \mu_{kd}^G, \sigma_{kd}^G\}_{k=1,\dots,K,d=1,\dots,D}$ of the GMM (where K is the number of mixture components and D is the dimension of each local descriptor) were learned independently by optimizing a maximum-likelihood criterion with the Expectation Maximization (EM) algorithm [11]. EM is an iterative method that is deemed to have converged when the change in the likelihood function, or alternatively in the parameters λ, falls below some threshold ε. As stopping criterion we used the convergence in L_2-norm of the mean parameters, choosing $\varepsilon = 0.05$. As suggested in [11], the BMM/GMM parameters used in EM algorithm were initialized with: (a) $1/K$ for the mixing coefficients w_k^B and w_k^G; (b) random values chosen uniformly in the range $(0.25, 0.75)$, for the BMM means μ_{kd}^B; (c) centroids precomputed using k-means for the GMM means μ_{kd}^G; (d) mean variance of the clusters found using k-means for the diagonal elements σ_{kd}^G of the GMM covariance matrices.
All the learning stages, i.e. k-means, k-medoids, k-majority and the estimation of GMM/BMM, were performed using in order of 1M descriptors randomly selected from the local features extracted on the training sets (namely Flickr60k for INRIA Holidays and Paris6k for Oxford5k).

BoW, VLAD, FV. The various encodings of the local feature (as well as the visual vocabularies and the BMM/GMM) were computed by using our Visual Information Retrieval library that is publicly available on GitHub[4]. These representations are all parametrized by a single integer K. It corresponds to the number of centroids (visual words) used in BoW and VLAD, and to the number of mixture components of GMM/BMM used in FV representations.
For the FVs, we used only the components $\mathscr{G}_\mu$ associated with the mean vectors because, as happened in the non-binary case, we observed that the components related to the mixture weights do not improve the results.
As a common post-processing step [55,35], both the FVs and the VLADs were power-law normalized and subsequently L_2-normalized. The power-law normalization is parametrized by a constant β and it is defined as $x \rightarrow |x|^\beta \operatorname{sign}(x)$. In our experiments we used $\beta = 0.5$.

[3] `http://opencv.org/`
[4] https://github.com/ffalchi/it.cnr.isti.vir

We also applied PCA to reduce VLAD and FV dimensionality. The projection matrices were estimated on the training datasets.

CNN features. We used the pre-trained HybridNet [82] model, downloaded from the Caffe Model Zoo[5]. The architecture of HybridNet is the same as the BVLC Reference CaffeNet[6] which mimics the original AlexNet [39], with minor variations as described in [36]. It has 8 weight layers (5 convolutional + 3 fully-connected). The model has been trained on 1 183 categories (205 scene categories from Places Database [82] and 978 object categories from ImageNet [20]) with about 3.6 million images.
In the test phase we used Caffe and we extracted the output of the first fully-connected layer (*fc6*) after applying the Rectified Linear Unit (*ReLU*) transform. The resulting 4 096-dimensional descriptors were L_2 normalized.
As preprocessing step we warped the input images to the canonical resolution of 227×227 RGB (as also done in [21]).

Feature comparison and performance measure. The cosine similarity in conjunction with a term weighting scheme (e.g., tf-idf) is adopted for evaluating the similarity between BoW representations, while the Euclidean distance is used to compare VLAD, FV and CNN-based image signatures.
The image comparison based on the *direct matching* of the local features (i.e. without aggregation) was performed adopting the distance ratio criterion proposed in [48, 28]. Specifically, candidate matches to local features of the image query are identified by finding their nearest neighbors in the database of images. Matches are discarded if the ratio of the distances between the two closest neighbors is above the 0.8 threshold. Similarity between two images is computed as the percentage of matching pairs with respect to the total local features in the query image.
The retrieval performance of each method was measured by the *mean average precision* (mAP). In the experiments on INRIA Holidays, we computed the average precision after removing the query image from the ranking list. In the experiments on Oxford5k, we removed the *junk* images from the ranking before computing the average precision, as recommended in [56] and in the evaluation package provided with the dataset.

4.3 Comparison of Various Encodings of Binary Local Features

In Table 3 we summarize the retrieval performance of various aggregation methods applied to ORB features, i.e. the BoW, the VLAD, the FV based on the GMM, and the BMM-FV. In addition, in the last line of the table we reports the results obtained without any aggregation, that we refer to as the *direct matching* of local features, which was performed adopting the distance ratio criterion as previously described in the Subsection 4.2.

In our experiments the FV derived as in [71] obtained very similar performance to that of our BMM-FV, thus we have reported just the results obtained by using our

[5] `https://github.com/BVLC/caffe/wiki/Model-Zoo`
[6] `https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet`

Table 3 Performance evaluation of various aggregation methods applied on ORB binary features. *K* indicates the number of centroids (visual words) used in BoW and VLAD and the number of mixture components of GMM/BMM used in FV; *dim* is the number of components of each vector representation.

Method	Local Feature	Learning method	K	dim	mAP Holidays	mAP Oxford5k
BoW	ORB	*k*-means	20 000	20 000	44.9	22.2
BoW	ORB	*k*-majority	20 000	20 000	44.2	22.8
BoW	ORB	*k*-medoids	20 000	20 000	37.9	18.8
VLAD	ORB	*k*-means	64	16 384	47.8	23.6
				PCA→ 1 024	46.0	23.2
				PCA→ 128	30.9	19.3
VLAD	ORB	*k*-majority	64	16 384	32.4	16.6
VLAD	ORB	*k*-medoids	64	16 384	30.6	15.6
FV	ORB	GMM	64	16 384	42.0	20.4
				PCA→ 1 024	42.6	20.3
				PCA→ 128	35.5	19.6
FV		BMM	64	16 384	49.6	**24.3**
				PCA→ 1 024	**51.3**	23.4
				PCA→ 128	44.6	19.1
No-aggr.	ORB	-	-		38.1	31.7

Table 4 Aggregation methods on non-binary local features. Results are reported from [34, 35].

Method	Local Feature	Learning method	K	dim	mAP Holidays	mAP Oxford5k
BoW	SIFT	*k*-means	20 000	20 000	40.4	-
BoW	SIFT PCA 64	*k*-means	20 000	20 000	43.7	35.4
VLAD	SIFT	*k*-means	64	8 192	52.6	-
				PCA→ 128	51.0	-
VLAD	SIFT PCA 64	*k*-means	64	4 096	55.6	37.8
				PCA→ 128	55.7	28.7
FV	SIFT	GMM	64	8 192	49.5	-
				PCA→ 128	49.2	-
FV	SIFT PCA 64	GMM	64	4 096	59.5	41.8
				PCA→ 128	56.5	30.1

formulation. Furthermore, we have not experimentally evaluated the FVs computed using the gradient with respect to the *natural parameters* of a BMM or a GMM as described in [62], because the evaluation of the retrieval performance obtained using or not the natural parameters in the derivation of the score function is a more general topic which reserve to be further investigated outside the specific context of the encodings binary local features.

Among the various baseline aggregation methods (i.e. without using PCA), the BMM-FV approach achieves the best retrieval performance, that is a mAP of *49.6%* on Holidays and **24.3%** on Oxford. PCA dimensionality reduction from 16 384 to

1 024 components, applied on BMM-FV, marginally reduces the mAP on Oxford5k, while on Holiday allows us to get **51.3%** that is, for this dataset, the best result achieved between all the other aggregation techniques tested on ORB binary features.

Good results are also achieved using VLAD in conjunction with *k*-means, which obtains a mAP of *47.8%* on Holidays and *23.6%* on Oxford5k.

The BOW representation allows to get a mAP of *44.9%/44.2%/37.9%* on Holidays and *22.2%/22.8%/18.8%* on Oxford5k using respectively *k*-means/*k*-majority/*k*-medoids for the learning of a visual vocabulary of 20000 visual words.

The GMM-FV method gives results slight worse than BoW: *42.0%* of mAP on Holidays and *20.4%* of mAP on Oxford5k. The use of PCA to reduce dimensions from 16384 to 1 024 lefts the results of GMM-FV on Oxford5k substantially unchanged while slightly improved the mAP on Holidays (*42.6%*).

Finally, the worst performance are that of VLAD in combination with vocabularies learned by *k*-majority (*32.4%* on Holidays and *16.6%* on Oxford) and *k*-medoids (*30.6%* on Holidays and *15.6%* on Oxford).

It is generally interesting to note that on INRIA Holidays, the VLAD with *k*-means, the BoW with *k*-means/*k*-majority, and the FVs are better than direct match. In fact, mAP of direct matching of ORB descriptors is *38.1%* while on Oxford5k the direct matching reached a mAP of *31.7%*.

In Table 5 we also report the performance of our derivation of the BMM-FV varying the number K of Bernoulli mixture components and investigating the impact of the PCA dimensionality reduction in the case of $K = 64$.

In Table (5(a)) we can see that with the Holidays dataset, the mAP grows from *32.0%* when using only 4 mixtures to **54.7%** when using $K = 512$. On Oxford5k, mAP varies from *14.3%* to **27.4%**, respectively, for $K = 4$ and $K = 512$.

Table (5(b)) shows that the best results are achieved when reducing the full size BMM-FV to 4 096 with a mAP of **52.6%** for Holidays and **25.1%** for Oxfrod5k.

Analysis of the results Summing up, the results show that in the context of binary local features the BMM-FV outperforms the compared aggregation methods, namely BoW, VLAD and GMM-FV. The performance of the BMM-FV is an increasing function of the number K of Benoulli mixtures. However, for large K, the improvement tends to be smaller and the dimensionality of the FV becomes very large (e.g. 65 536 dimensions using $K = 256$). Hence, for high values of K, the benefit of the improved accuracy is not worth the computational overhead (both for the BMM estimation and for the cost of storage/comparison of FVs).

The PCA reduction of BMM-FV is effective since it can provide a very compact image signature with just a slight loss in accuracy, as shown in the case of $K = 64$ (Table 5(b)). Dimension reduction does not necessarily reduce the accuracy. Conversely, limited reduction tend to improve the retrieval performance of the FV representations.

For the computation of VLAD, the *k*-means results are more effective than *k*-majority/*k*-medoids clustering, since the use of non-binary centroids gives more discriminant information during the computation of the residual vectors used in VLAD.

For the BoW approach, *k*-means and *k*-majority performs equally better than *k*-medoids. However, the *k*-majority is preferable in this case because the cost of the

Table 5 Retrieval performance of our BMM-FV on INRIA Holidays and Oxford5k. K is the number of BMM mixtures. *dim* is the number of components of the final vector representation.

(a) Performance evaluation for increasing number K of Bernullian mixture components

K	**dim**	**mAP** Holidays	**mAP** Oxford5k
4	1 024	32.0	14.3
8	2 048	38.2	17.4
16	4 096	41.9	19.4
32	8 192	45.9	21.3
64	16 384	49.6	24.3
128	32 768	52.3	26.4
256	65 536	53.0	27.3
512	131 072	**54.7**	**27.4**

(b) Results after dimensionality reduction when $K = 64$ Bernoulli are used

K	**dim**	**mAP** Holidays	**mAP** Oxford5k
64	16 384	49.6	24.3
64	PCA→ 4 096	**52.6**	**25.1**
64	PCA→ 2 048	51.8	24.3
64	PCA→ 1 024	51.3	23.4
64	PCA→ 512	48.2	21.7
64	PCA→ 256	45.9	20.3
64	PCA→ 128	44.6	19.1
64	PCA→ 64	42.9	17.2

quantization process is significantly reduced by using the Hamming distance, rather than Euclidean one, for the comparison between centroids and binary local features.

Both BMM-FV and VLAD, with only $K = 64$, outperform BoW. However, as happens for non-binary features (see Table 4), the loss in accuracy of BoW representation is comparatively lower when the variability of the images is limited, as for the Oxford5k dataset.

As expected, BMM-FV outperforms GMM-FV, since the probability distribution of binary local features is better described using mixtures of Bernoulli rather than mixtures of Gaussian. The results of our experiments also show that the use of BMM-FV is still effective even if compared with the direct matching strategy. In fact, the retrieval performance of BMM-FV on Oxford5k is just slightly worse than traditional direct matching of local feature, while on INRIA Holidays the BMM-FV even outperforms the direct matching result.

For completeness, in Table 4, we also report the results of the same base-line encodings approaches applied to non-binary features (both full-size SIFT and PCA-reduced to 64 components) taken from literature [34,35]. As expected, aggregation methods in general exhibit better performance in combination with SIFT/SIFTPCA then with ORB, expecially for the Oxford5k dataset. However, it is worth noting that

Table 6 Retrieval performance of various combinations of BMM-FV and HybridNet CNN feature. The BMM-FV representations were computed for three different binary local features (ORB, LATCH, and AKAZE) using $K = 64$ mixtures of Bernoulli. The CNN feature was computed as the output the HybriNet fc6 layer after applying the ReLU transform. *Dim* is the number of components of each vector representation. α is the parameter used in the combination of FV and CNN: $\alpha = 0$ corresponds to use only FV, while $\alpha = 1$ correspond to use only the HybridNet feature.

Method	**Dim**			α	**mAP**		
	ORB	LATCH	AKAZE		ORB	LATCH	AKAZE
BMM-FV (K=64)	16,384	16,384	32,768	0	49.6	46.3	43.7
Combination of *BMM-FV (K=64)* and *HybridNet fc6*	20,480	20,480	36,864	0.1	66.4	64.7	59.2
				0.2	74.8	73.8	68.7
				0.3	77.4	76.8	74.3
				0.4	79.1	77.5	77.3
				0.5	**79.2**	78.3	78.0
				0.6	79.0	**78.5**	**79.2**
				0.7	78.7	77.7	78.7
				0.8	77.8	76.7	77.5
				0.9	76.4	76.3	76.2
HybridNet fc6		4,096		1		75.5	

on the INRIA Holidays the BMM-FV outperforms the BoW on SIFT/SIFTPCA and reach similar performance of the FV built upon SIFTs.

The FV and VLAD get considerable benefit from performing PCA of SIFT local descriptors before the aggregation phase as the PCA rotation decorrelate the descriptors components. This suggest that techniques, such as VLAD with k-means and GMM-FV, which treat binary vectors as real-valued vectors, may also benefit from the use of PCA before the aggregation phase.

In conclusion, it is important to point-out that there are several applications where binary features need to be used to improve efficiency, at the cost of some effectiveness reduction [28]. We showed that in this case the use of the encodings techniques represent a valid alternative to the direct matching.

4.4 Combination of BMM-FVs and CNNs

In this section we evaluate the retrieval performance of the combination of BMM-FV and CNN features using the approach described in Section 3.2, We considered the INRIA Holidays dataset and we used the the output of the first fully-connected layer (*fc6*) of the HybridNet [82] model as CNN feature. In fact, in [14] several experiments on the INRIA Holidays have shown that *HybridNet fc6* achieve better mAP result than other outputs (e.g. *pool5, fc6, fc7, fc8*) of several pre-trained CNN models: OxfordNet [64], AlexNet [39], PlacesNet [82] and HybridNet itself.

In Table 6 we report the mAP obtained combining the HybridNet fc6 feature with the BMM-FV computed for three different kind of binary local features, namely ORB, LATCH and AKAZE, using $K = 64$ mixtures of Bernoulli. It is worth noting

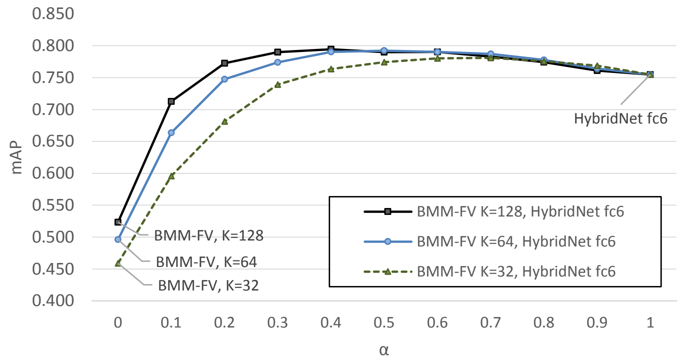

Fig. 1 Retrieval performance of the combinations of BMM-FV and HybridNet *fc6* for various number K of Bernoulli mixtures. The BMM-FVs were computed using ORB binary features. α is the parameter used in the combination: $\alpha = 0$ corresponds to use only FV, while $\alpha = 1$ correspond to use only the HybridNet feature.

that all the three BMM-FVs give a similar improvement when combined with the HybridNet fc6, although they have rather different mAP results (see first row of Table 6) which are substantially lower than that of CNN (last row of Table 6). The intuition is that the additional information provided by using a specific BMM-FV rather than using the CNN feature alone, do not depend very much on the used binary feature.

For each tested BMM-FV seems that exist an optimal α to be used in the convex combination (equation (4)). When ORB binary features were used, the optimal α was obtained around 0.5, which correspond to give the same importance to both FV and CNN feature. For the less effective BMM-FVs built upon LATCH and AKAZE, the optimal α was 0.6, which means that the CNN feature is used with slightly more importance than BMM-FV during the convex combination.

The use of ORB or AKAZE led to obtain the best performance that was **79.2%** of mAP. This results in a relative improvement of **4.9%** respect to the single use of the CNN feature, that in our case was **75.5%**. So we obtain the same relative improvement of [14] but using a less expensive FV representation. Indeed, in [14] the fusion of HybridNet *fc6* and a FV computed on 64-dimensional PCA-reduced SIFTs, using $K = 256$ mixtures of Gaussian, have led to obtain a relative improvement of 4.9% respect to the use of the CNN feature alone (see also Table 8).

Since as observed in [45,4] the ORB extraction is faster than LATCH and AKAZE, in the following we focus just on ORB binary feature. In figure 1 we show the results obtained by combining HybridNet *fc6* with the BMM-FVs obtained using $K = 32, 64, 128$. We observed that the performance of the CNN feature is improved also when it is combined with the less effective BMM-FV built using $K = 32$ Bernoulli. The BMM-FV with $K = 128$ achieve the best effectiveness (mAP of **79.5%**) for $\alpha = 0.4$. However, since the cost for computing and storing FV increase with the number K of Bernoulli, the improvement obtained using $K = 128$ respect to that of $K = 64$ doesn't worth the extra cost of using a bigger value of K.

Table 7 Comparison of the results obtained combining HybridNet fc6 feature with the full-sized and the PCA-reduced version of the BMM-FV. The BMM-FV was computed on ORB binary feature using $K = 64$ mixtures of Bernoulli. *Dim* is the number of components of each vector representation. α is the parameter used in the combination of FV and CNN.

Method	**Dim**		α	**mAP**	
	FV full dim	FV PCA-reduced		FV full dim	FV PCA-reduced
BMM-FV (K=64)	16,384	4,096	0	49.6	52.6
Combination of *BMM-FV (K=64)* and *HybridNet fc6*	20,480	8,192	0.1	66.4	66.3
			0.2	74.8	73.9
			0.3	77.4	77.3
			0.4	79.1	**78.5**
			0.5	**79.2**	78.4
			0.6	79.0	78.5
			0.7	78.7	78.1
			0.8	77.8	77.7
			0.9	76.4	76.4
HybridNet fc6	4,096		1	75.5	

The BMM-FV with $K = 64$ is still high dimensional, so to reduce the cost of storing and comparing FV, we also evaluated the combination after the PCA-dimensionality reduction. As already observed, limited dimensionality reduction tends to improve the accuracy of the single FV representation. In fact, the BMM-FV with $K = 64$ achieved a mAP of 52.6% when reduced from 16384 to 4096 dimensions. However, as shown in Table 7 and Table 8, when the PCA-reduced version of the BMM-FV was combined with HybriNet *fc6*, the overall relative improvement in mAP was 3.9%, which is less than that obtained using the full-sized BMM-FV. These result is not surprising given that after the dimensionality reduction we may have a loss of the additional information provided by the FV representation during the combination with the CNN feature.

Finally, in Table 8 we summarizes the relative improvement achieved by combining BMM-FV and HybriNet *fc6*, and we compare the obtained results with the relative improvement achieved in [14], where the more expensive FV built upon SIFTs was used. We observed that BMM-FV led to achieve similar or even better relative improvements with an evident advantage from the computational point of view, because it use binary local features and smaller number K of mixtures.

5 Conclusion

Motivated by recent results obtained on one hand with the use of aggregation methods applied to local descriptors and on the other with the definition of binary local features, this paper has performed an extensive comparisons of techniques that mix the two approaches by using aggregation methods on binary local features. The use of aggregation methods on binary local features is justified by the need for increasing

Table 8 Relative mAP improvement obtained after combining FV with HybridNet *fc6*. Each relative improvements was computed respect to the use of the CNN feature alone, that is: $\left(\text{mAP}_{\text{after combination}} - \text{mAP}_{\text{HybridNet fc6}}\right)/\text{mAP}_{\text{HybridNet fc6}}$. The relative improvements obtained using the FV computed on 64-dimensional PCA-reduced SIFTs (SIFTPCA64) was computed according to the results reported in [14].

FV method	Local Feature	K	dim	Relative improvement
BMM-FV	ORB	128	32,768	5.2
BMM-FV	ORB	64	16,384	4.9
BMM-FV	AKAZE	64	32,768	4.9
BMM-FV	LATCH	64	16,384	4.0
BMM-FV+ PCA	ORB	64	4,096	3.9
BMM-FV	ORB	32	8,192	3.5
FV[14]	SIFTPCA64	256	32,768	4.9

efficiency and reducing computing resources for image matching on a large scale, at the expense of some degradation in the accuracy of retrieval algorithms. Combining the two approaches lead to execute image retrieval on a very large scale and reduce the cost for feature extraction and representation. Thus we expect that the results of our empirical evaluation are useful for people working with binary local descriptors.

Moreover, we investigated how the Fisher Vector obtained by aggregating binary local features (BMM-FV) works in conjunction with the CNN pipeline in order to improve the latter retrieval performance. We showed that the BMM-FV built upon ORB binary features can be profitable use to this scope, even if a relative small number of Bernoulli is used. In fact, the relative improvement in the retrieval performance obtained using the BMM-FV is similar to that previously obtained in [14] using the more expensive FV built on SIFT.

It is also worth mentioning that the BMM-FV approach is very general and could be applied to any binary feature. Recent works based on CNNs suggest that binary features aggregation technique could be further applied to deep features. In fact, on one hand, local features based on CNNs, aggregated with VLAD and FV approaches, have been proposed to obtain robustness to geometric deformations [78, 73]. On the other hand, binarization of global CNN features have been also proposed in [46, 40]. Thus, as a future work, we plan to test the BMM-FV approach over binary deep local descriptors leveraging on the local and binary approaches mentioned above.

Acknowledgements This work was partially founded by: EAGLE, Europeana network of Ancient Greek and Latin Epigraphy, co-founded by the European Commission, CIP-ICT-PSP.2012.2.1 - Europeana and creativity, Grant Agreement n. 325122; and Smart News, Social sensing for breakingnews, co-founded by the Tuscany region under the FAR-FAS 2014 program, CUP CIPE D58C15000270008.

References

1. Bing images. Http://www.bing.com/images/
2. Google googles. Http://www.google.com/mobile/goggles/
3. Google images. Https://images.google.com/

4. Alcantarilla, P.F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: In British Machine Vision Conference (BMVC) (2013)
5. Amato, G., Falchi, F., Vadicamo, L.: How effective are aggregation methods on binary features? In: Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 4, pp. 566–573 (2016)
6. Amato, G., Falchi, F., Vadicamo, L.: Visual recognition of ancient inscriptions using convolutional neural network and fisher vector. Journal on Computing and Cultural Heritage (JOCCH) (2016). To Appear
7. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2911–2918 (2012)
8. Arandjelovic, R., Zisserman, A.: All about VLAD. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 1578–1585 (2013). DOI 10.1109/CVPR.2013.207
9. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Computer Vision–ECCV 2014, pp. 584–599. Springer (2014). DOI 10.1007/978-3-319-10590-1_38. URL `http://dx.doi.org/10.1007/978-3-319-10590-1_38`
10. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: A. Leonardis, H. Bischof, A. Pinz (eds.) Computer Vision - ECCV 2006, *Lecture Notes in Computer Science*, vol. 3951, pp. 404–417. Springer Berlin Heidelberg (2006). DOI 10.1007/11744023_32. URL `http://dx.doi.org/10.1007/11744023_32`
11. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer (2006)
12. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2559–2566 (2010)
13. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: K. Daniilidis, P. Maragos, N. Paragios (eds.) Computer Vision - ECCV 2010, *Lecture Notes in Computer Science*, vol. 6314, pp. 778–792. Springer Berlin Heidelberg (2010)
14. Chandrasekhar, V., Lin, J., Morère, O., Goh, H., Veillard, A.: A practical guide to cnns and fisher vectors for image instance retrieval. CoRR **abs/1508.02496** (2015). URL `http://arxiv.org/abs/1508.02496`
15. Chen, D., Tsai, S., Chandrasekhar, V., Takacs, G., Chen, H., Vedantham, R., Grzeszczuk, R., Girod, B.: Residual enhanced visual vectors for on-device image matching. In: Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on, pp. 850–854 (2011). DOI 10.1016/j.sigpro.2012.06.005. URL `http://dx.doi.org/10.1016/j.sigpro.2012.06.005`
16. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–8 (2007)
17. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. Workshop on statistical learning in computer vision, ECCV **1**(1-22), 1–2 (2004)
18. Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: Approaches and trends of the new age. In: Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR '05, pp. 253–262. ACM, New York, NY, USA (2005)
19. Delhumeau, J., Gosselin, P.H., Jégou, H., Pérez, P.: Revisiting the VLAD image representation. In: Proceedings of the 21st ACM International Conference on Multimedia, MM 2013, pp. 653–656. ACM, New York, NY, USA (2013). DOI 10.1145/2502081.2502171. URL `http://doi.acm.org/10.1145/2502081.2502171`
20. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248–255 (2009). DOI 10.1109/CVPR.2009.5206848
21. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. CoRR **abs/1310.1531** (2013). URL `http://arxiv.org/abs/1310.1531`
22. Galvez-Lopez, D., Tardos, J.: Real-time loop detection with bags of binary words. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 51–58 (2011)
23. van Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.: Kernel codebooks for scene categorization. In: D. Forsyth, P. Torr, A. Zisserman (eds.) Computer Vision - ECCV 2008, *Lecture Notes in Computer Science*, vol. 5304, pp. 696–709. Springer Berlin Heidelberg (2008)
24. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning (2016). URL `http://www.deeplearningbook.org`. Book in preparation for MIT Press

25. Grana, C., Borghesani, D., Manfredi, M., Cucchiara, R.: A fast approach for integrating ORB descriptors in the bag of words model. In: C.G.M. Snoek, L.S. Kennedy, R. Creutzburg, D. Akopian, D. Wüller, K.J. Matherson, T.G. Georgiev, A. Lumsdaine (eds.) IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics (2013)
26. Gray, R.M., Neuhoff, D.L.: Quantization. Information Theory, IEEE Transactions on **44**(6), 2325–2383 (1998). DOI 10.1109/18.720541. URL `http://dx.doi.org/10.1109/18.720541`
27. Hamming, R.W.: Error detecting and error correcting codes. The Bell System Technical Journal **29**(2), 147–160 (1950). DOI 10.1002/j.1538-7305.1950.tb00463.x
28. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Computer Vision - ECCV 2012, Lecture Notes in Computer Science, pp. 759–773. Springer Berlin Heidelberg (2012)
29. Householder, A.: The Theory of Matrices in Numerical Analysis. A Blaisdell book in pure and applied sciences: introduction to higher mathematics. Blaisdell Publishing Company (1964)
30. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: In Advances in Neural Information Processing Systems 11, pp. 487–493. MIT Press (1998). URL `http://dl.acm.org/citation.cfm?id=340534.340715`
31. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: D. Forsyth, P. Torr, A. Zisserman (eds.) European Conference on Computer Vision, *LNCS*, vol. I, pp. 304–317. Springer (2008)
32. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. International Journal of Computer Vision **87**(3), 316–336 (2010). DOI 10.1007/s11263-009-0285-2. URL `http://dx.doi.org/10.1007/s11263-009-0285-2`
33. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(1), 117–128 (2011). DOI 10.1109/TPAMI.2010.57
34. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: IEEE Conference on Computer Vision & Pattern Recognition (2010). DOI 10.1109/CVPR.2010.5540039
35. Jégou, H., Perronnin, F., Douze, M., Sànchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(9), 1704–1716 (2012). DOI 10.1109/TPAMI.2011.235
36. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014). DOI 10.1145/2647868.2654889. URL `http://doi.acm.org/10.1145/2647868.2654889`
37. Kaufman, L., Rousseeuw, P.: Clustering by means of medoids. In: Y. Dodge (ed.) An introduction to L1-norm based statistical data analysis, *Computational Statistics & Data Analysis*, vol. 5 (1987)
38. Krapac, J., Verbeek, J., Jurie, F.: Modeling Spatial Layout with Fisher Vectors for Image Categorization. In: ICCV 2011 - International Conference on Computer Vision, pp. 1487–1494. IEEE, Barcelona, Spain (2011)
39. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: F. Pereira, C. Burges, L. Bottou, K. Weinberger (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
40. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
41. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2 (2006)
42. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). DOI 10.1038/nature14539
43. Lee, S., Choi, S., Yang, H.: Bag-of-binary-features for fast image representation. Electronics Letters **51**(7), 555–557 (2015)
44. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2548–2555 (2011)
45. Levi, G., Hassner, T.: LATCH: learned arrangements of three patch codes. CoRR **abs/1501.03719** (2015)
46. Lin, K., Yang, H.F., Hsiao, J.H., Chen, C.S.: Deep learning of binary hash codes for fast image retrieval. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2015)

47. Lloyd, S.: Least squares quantization in pcm. Information Theory, IEEE Transactions on **28**(2), 129–137 (1982). DOI 10.1109/TIT.1982.1056489. URL http://dx.doi.org/10.1109/TIT.1982.1056489
48. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004). DOI 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94
49. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley series in probability and statistics. Wiley (2000)
50. Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 2681–2684 (2012)
51. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 9–16 (2009)
52. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pp. 1–8 (2007). DOI 10.1109/CVPR.2007.383266
53. Perronnin, F., Larlus, D.: Fisher Vectors Meet Neural Networks: A Hybrid Classification Architecture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3743–3752 (2015)
54. Perronnin, F., Liu, Y., Sànchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3384–3391 (2010). DOI 10.1109/CVPR.2010.5540009
55. Perronnin, F., Sànchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Computer Vision - ECCV 2010, *Lecture Notes in Computer Science*, vol. 6314, pp. 143–156. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-15561-1_11. URL http://dx.doi.org/10.1007/978-3-642-15561-1_11
56. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on, pp. 1–8 (2007). DOI 10.1109/CVPR.2007.383172
57. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8 (2008). DOI 10.1109/CVPR.2008.4587635
58. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, pp. 512–519. IEEE (2014). DOI 10.1109/CVPRW.2014.131
59. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2564–2571 (2011)
60. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
61. Sànchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International Journal of Computer Vision **105**(3), 222–245 (2013). DOI 10.1007/s11263-013-0636-x. URL http://dx.doi.org/10.1007/s11263-013-0636-x
62. Sànchez, J., Redolfi, J.: Exponential family fisher vector for image classification. Pattern Recognition Letters **59**, 26 – 32 (2015). DOI http://dx.doi.org/10.1016/j.patrec.2015.03.010
63. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep fisher networks for large-scale image classification. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 163–171. Curran Associates, Inc. (2013)
64. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014). URL http://arxiv.org/abs/1409.1556
65. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, *ICCV '03*, vol. 2, pp. 1470–1477. IEEE Computer Society (2003). DOI 10.1109/ICCV.2003.1238663
66. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1349–1380 (2000)
67. Sydorov, V., Sakurada, M., Lampert, C.H.: Deep fisher kernels - end to end learning of the fisher kernel gmm parameters. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
68. Tolias, G., Avrithis, Y.: Speeded-up, relaxed spatial matching. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1653–1660 (2011). DOI 10.1109/ICCV.2011.6126427

69. Tolias, G., Furon, T., Jégou, H.: Orientation covariant aggregation of local descriptors with embeddings. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds.) Computer Vision - ECCV 2014, *Lecture Notes in Computer Science*, vol. 8694, pp. 382–397. Springer International Publishing (2014)
70. Tolias, G., Jégou, H.: Local visual query expansion: Exploiting an image collection to refine local descriptors. Research Report RR-8325 (2013). URL `https://hal.inria.fr/hal-00840721`
71. Uchida, Y., Sakazawa, S.: Image retrieval with fisher vectors of binary features. In: Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on, pp. 23–28 (2013)
72. Ullman, S.: High-Level Vision - Object Recognition and Visual Cognition. MIT Press (1996)
73. Uricchio, T., Bertini, M., Seidenari, L., Del Bimbo, A.: Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2015)
74. Van Opdenbosch, D., Schroth, G., Huitl, R., Hilsenbeck, S., Garcea, A., Steinbach, E.: Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In: IEEE International Conference on Image Processing (2014)
75. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3360–3367 (2010)
76. Witten, I.H., Moffat, A., Bell, T.C.: Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann (1999)
77. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1794–1801 (2009)
78. Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2015)
79. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach, *Advances in Database Systems*, vol. 32. Springer (2006)
80. Zhang, Y., Zhu, C., Bres, S., Chen, L.: Encoding local binary descriptors by bag-of-features with hamming distance for visual object categorization. In: P. Serdyukov, P. Braslavski, S. Kuznetsov, J. Kamps, S. Rger, E. Agichtein, I. Segalovich, E. Yilmaz (eds.) Advances in Information Retrieval, *Lecture Notes in Computer Science*, vol. 7814, pp. 630–641. Springer Berlin Heidelberg (2013)
81. Zhao, W., Jégou, H., Gravier, G.: Oriented pooling for dense and non-dense rotation-invariant features. In: BMVC - 24th British Machine Vision Conference (2013)
82. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (eds.) Advances in Neural Information Processing Systems 27, pp. 487–495. Curran Associates, Inc. (2014)

A Score vector computation

In the following, we have reported the computation of the score function G^X_λ, defined as the gradient of the log-likelihood of a data X with respect to the parameters λ of a Bernoulli Mixture Model. Throughout this appendix we have used $[\![\cdot]\!]$ notation to represent the Iverson bracket which equals one if the arguments is true, and zero otherwise.

Under the independence assumption, the Fisher score with respect to the generic parameter λ_k is expressed as: $G^X_{\lambda_k} = \sum_{t=1}^T \frac{\partial \log p(x_t|\lambda)}{\partial \lambda_k} = \sum_{t=1}^T \frac{1}{p(x_t|\lambda)} \frac{\partial}{\partial \lambda_k}\left[\sum_{i=1}^K w_i p_i(x_t)\right]$. To compute $\frac{\partial}{\partial \lambda_k}\left[\sum_{i=1}^K w_i p_i(x_t)\right]$, we first observe that

$$\begin{aligned}
\frac{\partial w_i}{\partial \alpha_k} &= \frac{\partial}{\partial \alpha_k}\left[\frac{\exp(\alpha_i)}{\sum_{j=1}^K \exp(\alpha_j)}\right] \\
&= \frac{\exp(\alpha_k)\left(\sum_{j=1}^K \exp(\alpha_j)\right)[\![i=k]\!] - \exp(\alpha_i)\exp(\alpha_k)}{\left(\sum_{j=1}^K \exp(\alpha_j)\right)^2} \\
&= w_k[\![i=k]\!] - w_k w_i
\end{aligned} \tag{5}$$

and

$$
\begin{aligned}
\frac{\partial p_i(x_t)}{\partial \mu_{kd}} &= \frac{\partial}{\partial \mu_{kd}} \left[\prod_{l=1}^{D} \mu_{kl}^{x_{tl}} \left(1-\mu_{kl}\right)^{1-x_{tl}} \right] [\![i=k]\!] \\
&= ([\![x_{td}=1]\!] - [\![x_{td}=0]\!]) \left(\prod_{\substack{l=1\\ l\neq d}}^{D} \mu_{kl}^{x_{tl}} \left(1-\mu_{kl}\right)^{1-x_{tl}} \right) [\![i=k]\!] \\
&= ([\![x_{td}=1]\!] - [\![x_{td}=0]\!]) \left(\frac{p_k(x_t)}{\mu_{kd}^{x_{td}} \left(1-\mu_{kd}\right)^{1-x_{td}}} \right) [\![i=k]\!] \\
&= p_k(x_t) \left(\frac{(1-\mu_{kd})[\![x_{td}=1]\!] - \mu_{kd}[\![x_{td}=0]\!]}{\mu_{kd}(1-\mu_{kd})} \right) [\![i=k]\!] \\
&= p_k(x_t) \left(\frac{x_{td}-\mu_{kd}}{\mu_{kd}(1-\mu_{kd})} \right) [\![i=k]\!].
\end{aligned}
\tag{6}
$$

Hence, the Fisher score with respect to the parameter α_k is obtained as

$$
\begin{aligned}
G_{\alpha_k}^X &= \sum_{t=1}^{T} \sum_{i=1}^{K} \frac{p_i(x_t)}{p(x_t|\lambda)} \frac{\partial w_i}{\partial \alpha_k} \overset{(5)}{=} \sum_{t=1}^{T} \sum_{i=1}^{K} \frac{p_i(x_t)}{p(x_t|\lambda)} w_k \left([\![i=k]\!] - w_i\right) \\
&= \sum_{t=1}^{T} \left(\frac{p_k(x_t)}{p(x_t|\lambda)} w_k - \sum_{i=1}^{K} \frac{p_i(x_t)}{p(x_t|\lambda)} w_k w_i \right) = \sum_{t=1}^{T} \left(\gamma_t(k) - w_k \sum_{i=1}^{K} \gamma_t(i) \right) \\
&= \sum_{t=1}^{T} \left(\gamma_t(k) - w_k\right)
\end{aligned}
\tag{7}
$$

and the Fisher score related to the parameter μ_{kd} is

$$
\begin{aligned}
G_{\mu_{kd}}^X &= \sum_{t=1}^{T} \frac{\partial \log p(x_t|\lambda)}{\partial \mu_{kd}} = \sum_{t=1}^{T} \frac{1}{p(x_t|\lambda)} \frac{\partial}{\partial \mu_{kd}} \left[\sum_{i=1}^{K} w_i p_i(x_t) \right] \\
&= \sum_{t=1}^{T} \frac{w_k}{p(x_t|\lambda)} \frac{\partial p_k(x_t)}{\partial \mu_{kd}} \overset{(6)}{=} \sum_{t=1}^{T} \frac{w_k p_k(x_t)}{p(x_t|\lambda)} \left(\frac{x_{td}-\mu_{kd}}{\mu_{kd}(1-\mu_{kd})} \right) \\
&= \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_{td}-\mu_{kd}}{\mu_{kd}(1-\mu_{kd})} \right).
\end{aligned}
\tag{8}
$$

B Approximation of the Fisher Information Matrix

Our derivation of the FIM is based on the assumption (see also [54,61]) that for each observation $x = (x_1,\ldots,x_D) \in \{0,1\}^D$ the distribution of the occupancy probability $\gamma(\cdot) = p(\cdot|x,\lambda)$ is sharply peaking, i.e. there is one Bernoulli index k such that $\gamma_x(k) \approx 1$ and $\forall i \neq k$, $\gamma_x(i) \approx 0$. This assumption implies that

$$
\begin{aligned}
&\gamma_x(k)\gamma_x(i) \approx 0 \quad \forall k,i = 1\ldots,K, i \neq k \\
&\gamma_x(k)^2 \approx \gamma_x(k) \quad \forall k = 1,\ldots,K
\end{aligned}
$$

and then

$$
\gamma_x(k)\gamma_x(i) \approx \gamma_x(k)[\![i=k]\!],
\tag{9}
$$

where $[\![\cdot]\!]$ is the Iverson bracket.
The elements of the FIM are defined as:

$$
[F_\lambda]_{i,j} = \mathbb{E}_{x\sim p(\cdot|\lambda)} \left[\left(\frac{\partial \log p(x|\lambda)}{\partial \lambda_i} \right) \left(\frac{\partial \log p(x|\lambda)}{\partial \lambda_j} \right) \right].
\tag{10}
$$

Hence, the FIM F_λ is symmetric and can be written as block matrix

$$F_\lambda = \begin{bmatrix} F_{\alpha,\alpha} & F_{\mu,\alpha} \\ F_{\mu,\alpha}^\top & F_{\mu,\mu} \end{bmatrix}.$$

By using the definition of the occupancy probability (i.e. $\gamma_x(k) = w_k p_k(x)/p(x|\lambda)$) and the fact that p_k is the distribution of a D-dimensional Bernoulli of mean μ_k, we have the following useful equalities:

$$\mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(k)\right] = \sum_{x\in\{0,1\}^D} \gamma_x(k)p(x|\lambda) = w_k \tag{11}$$

$$\mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(k)x_d\right] = w_k\mu_{kd} \tag{12}$$

$$\mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(k)x_d x_l\right] = w_k\mu_{kd}\left(\mu_{kl}[\![d\neq l]\!] + [\![d=l]\!]\right) \tag{13}$$

$$\mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\frac{\partial \log p(x|\lambda)}{\partial \alpha_k}\right] \overset{(7)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(k) - w_k\right] = 0 \tag{14}$$

$$\mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\frac{\partial \log p(x|\lambda)}{\partial \mu_{id}}\right] \overset{(8)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\frac{\gamma_x(k)(x_d-\mu_{kd})}{\mu_{kd}(1-\mu_{kd})}\right] = 0. \tag{15}$$

It follows that F_λ may approximated by a diagonal block matrix, because the mixing blocks F_{μ_{kd},α_i} are close to the zero matrix:

$$\begin{aligned}
F_{\mu_{kd},\alpha_i} &= \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\left(\frac{\partial \log p(x|\lambda)}{\partial \mu_{kd}}\right)\left(\frac{\partial \log p(x|\lambda)}{\partial \alpha_i}\right)\right] \\
&\overset{(7)-(8)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(k)\frac{(x_d-\mu_{kd})}{\mu_{kd}(1-\mu_{kd})}(\gamma_x(i)-w_i)\right] \\
&\overset{(9)}{\approx} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\frac{\gamma_x(k)(x_d-\mu_{kd})}{\mu_{kd}(1-\mu_{kd})}\right]([\![i=k]\!]-w_i) \\
&\overset{(15)}{=} 0.
\end{aligned}$$

The block $F_{\mu,\mu}$ can be written as $KD\times KD$ diagonal matrix, in fact:

$$\begin{aligned}
F_{\mu_{id},\mu_{kl}} &\overset{(10)}{=} \mathbb{E}\left[\left(\frac{\partial \log p(x|\lambda)}{\partial \mu_{id}}\right)\left(\frac{\partial \log p(x|\lambda)}{\partial \mu_{kl}}\right)\right] \\
&\overset{(8)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\gamma_x(i)\gamma_x(k)\frac{(x_d-\mu_{id})}{\mu_{id}(1-\mu_{id})}\frac{(x_l-\mu_{kl})}{\mu_{kl}(1-\mu_{kl})}\right] \\
&\overset{(9)}{\approx} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\frac{\gamma_x(k)(x_d-\mu_{kd})(x_l-\mu_{kl})}{\mu_{kd}\mu_{kl}(1-\mu_{kd})(1-\mu_{kl})}\right][\![i=k]\!] \\
&\overset{(11)-(13)}{=} \frac{w_k(\mu_{kd}\mu_{kl}[\![d\neq l]\!] + \mu_{kl}[\![d=l]\!] - \mu_{kd}\mu_{kl})}{\mu_{kd}\mu_{kl}(1-\mu_{kd})(1-\mu_{kl})}[\![i=k]\!] \\
&= \frac{w_k(\mu_{kd}[\![d\neq l]\!] + [\![d=l]\!] - \mu_{kd})}{\mu_{kd}(1-\mu_{kd})(1-\mu_{kl})}[\![i=k]\!] \\
&= \frac{w_k}{\mu_{kd}(1-\mu_{kd})}[\![i=k]\!][\![d=l]\!].
\end{aligned} \tag{16}$$

The relation (16) points that the diagonal elements of our FIM approximation are $w_k/\mu_{kd}(1-\mu_{kd})$ and the corresponding entries in L_λ (i.e. the square root of the inverse of FIM) equal $\sqrt{\mu_{kd}(1-\mu_{kd})/w_k}$. The block related to the α parameters is $F_{\alpha,\alpha} = (\mathrm{diag}(w) - ww^\top)$ where $w = [w_1,\ldots,w_K]^\top$, in fact:

$$\begin{aligned}
F_{\alpha_k,\alpha_i} &\overset{(10)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[\left(\frac{\partial \log p(x|\lambda)}{\partial \alpha_k}\right)\left(\frac{\partial \log p(x|\lambda)}{\partial \alpha_i}\right)\right] \\
&\overset{(7)}{=} \mathbb{E}_{x\sim p(\cdot|\lambda)}\left[(\gamma_x(k)-w_k)(\gamma_x(i)-w_i)\right] \\
&\overset{(9)}{\approx} \mathbb{E}_{p(\cdot|\lambda)}\left[\gamma_x(k)[\![i=k]\!] - \gamma_x(k)w_i - \gamma_x(i)w_k + w_i w_k\right] \\
&\overset{(11)-(12)}{=} \left(w_k[\![i=k]\!] - w_i w_k\right).
\end{aligned}$$

The matrix $F_{\alpha,\alpha}$ is not invertible (indeed $F_{\alpha,\alpha}\mathbf{e}=0$ where $\mathbf{e}=[1,\ldots,1]^\top$) due to the dependence of the mixing weights ($\sum_{i=1}^K \alpha_i = \sum_{i=1}^K w_i = 1$). Since there are only $K-1$ degrees of freedom in the mixing weight, as proposed in [61], we can fix α_K equal to a constant without loss of generality and work with a reduced set of $K-1$ parameters: $\tilde{\alpha}=[\alpha_1,\ldots,\alpha_{K-1}]^\top$.

Taking into account the Fisher score with respect to $\tilde{\alpha}$, i.e.

$$G_{\tilde{\alpha}}^X = \nabla_{\tilde{\alpha}} \log p(X|\lambda) = [G_{\alpha_1}^X,\ldots,G_{\alpha_{K-1}}^X]^\top = \widetilde{G_{\alpha}^X},$$

the corresponding block of the FIM is $F_{\tilde{\alpha},\tilde{\alpha}} = (\mathrm{diag}(\tilde{w}) - \tilde{w}\tilde{w}^\top)$, where $\tilde{w}=[w_1,\ldots,w_{K-1}]^\top$. The matrix $F_{\tilde{\alpha},\tilde{\alpha}}$ is invertible, indeed it can be decomposed into a product of an invertible diagonal matrix $D=\mathrm{diag}(\tilde{w})$ and an invertible elementary matrix [7] $E(\mathbf{e},\tilde{w},-1) = I-\mathbf{e}\tilde{w}^\top$; its inverse is

$$F_{\tilde{\alpha},\tilde{\alpha}}^{-1} = \mathrm{diag}(\tilde{w})^{-1}\left(I + \frac{1}{\sum_{i=1}^{K-1} w_i - 1}\mathbf{e}\tilde{w}^\top\right) = \left(\mathrm{diag}(\tilde{w})^{-1} + \frac{1}{w_K}\mathbf{e}\mathbf{e}^\top\right).$$

It follows that

$$K_{\tilde{\alpha}}(X,Y) = (G_{\tilde{\alpha}}^X)^\top F_{\tilde{\alpha},\tilde{\alpha}}^{-1} G_{\tilde{\alpha}}^Y = \left((G_{\tilde{\alpha}}^X)^\top \mathrm{diag}(\tilde{w})^{-1} G_{\tilde{\alpha}}^Y + \frac{1}{w_K}(\mathbf{e}^\top G_{\tilde{\alpha}}^X)(\mathbf{e}^\top G_{\tilde{\alpha}}^Y)\right) = \sum_{k=1}^K \frac{G_{\alpha_k}^X G_{\alpha_k}^Y}{w_k}$$

where we used $\mathbf{e}^\top G_{\tilde{\alpha}}^Z = \sum_{k=1}^{K-1}\sum_{z\in Z}(\gamma_z(k)-w_k) = -\sum_{z\in Z}(\gamma_z(K)-w_K) = -G_{\alpha_K}^Z$.
By defining $\mathscr{G}_{\alpha_k}^X = \frac{1}{\sqrt{w_k}}\sum_{x\in X}(\gamma_x(k)-w_k)$, we finally obtain $K_{\tilde{\alpha}}(X,Y) = \left(\mathscr{G}_{\alpha}^X\right)^\top \mathscr{G}_{\alpha}^Y$. Please note that we don't need to explicitly compute the Cholesky decomposition of the matrix $F_{\tilde{\alpha},\tilde{\alpha}}^{-1}$ because the Fisher Kernel $K_{\tilde{\alpha}}(X,Y)$ can be easily rewritten as dot product between the feature vector $\mathscr{G}_{\alpha}^X$ and $\mathscr{G}_{\alpha}^Y$.

[7] An elementary matrix $E(u,v,\sigma) = I-\sigma uv^H$ is non-singular if and only if $\sigma v^H u \neq 1$ and in this case the inverse is $E(u,v,\sigma)^{-1} = E(u,v,\tau)$ where $\tau = \sigma/(\sigma v^H u - 1)$. More details on this topic can be found in [29].