This is a pre print version of the following article:

GOLD: Gaussians of Local Descriptors for Image Representation / Serra, G., Grana, C., Manfredi, M., Cucchiara, R.. - In: COMPUTER VISION AND IMAGE UNDERSTANDING. - ISSN 1077-3142. - STAMPA. - 134:(2015), pp. 22-32. [10.1016/j.cviu.2015.01.005]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

09/11/2026 07:31

(Article begins on next page)

GOLD: Gaussians of Local Descriptors for Image Representation

Giuseppe Serra[a], Costantino Grana[a,*], Marco Manfredi[a], Rita Cucchiara[a]

[a]*Dipartimento di Ingegneria "Enzo Ferrari"*
Università degli Studi di Modena e Reggio Emilia, Modena MO 41125, Italy

Abstract

The Bag of Words paradigm has been the baseline from which several successful image classification solutions were developed in the last decade. These represent images by quantizing local descriptors and summarizing their distribution. The quantization step introduces a dependency on the dataset, that even if in some contexts significantly boosts the performance, severely limits its generalization capabilities. Differently, in this paper, we propose to model the local features distribution with a multivariate Gaussian, without any quantization. The full rank covariance matrix, which lies on a Riemannian manifold, is projected on the tangent Euclidean space and concatenated to the mean vector. The resulting representation, a Gaussian of local descriptors (GOLD), allows to use the dot product to closely approximate a distance between distributions without the need for expensive kernel computations. We describe an image by an improved spatial pyramid, which avoids boundary effects with soft assignment: local descriptors contribute to neighboring Gaussians, forming a weighted spatial pyramid of GOLD descriptors. In addition, we extend the model leveraging dataset characteristics in a mixture of Gaussian formulation further improving the classification accuracy. To deal with large scale datasets and high dimensional feature spaces the Stochastic Gradient Descent solver is adopted. Experimental results on several publicly available datasets show that the proposed method obtains state-of-the-art performance.

*Corresponding author
Email addresses: giuseppe.serra@unimore.it (Giuseppe Serra), costantino.grana@unimore.it (Costantino Grana), marco.manfredi@unimore.it (Marco Manfredi), rita.cucchiara@unimore.it (Rita Cucchiara)

Keywords: image classification, concept detection, Gaussian distribution, stochastic gradient descent

1. Introduction

Object and Scene Recognition have been a major research direction in computer vision, and, in particular, the task of automatically annotating images has received considerable attention. Systems extract some description from a training set of images, train a classifier and then can be used to perform their task on new images. The current "standard" approach for this task is some derivation of the Bag of Words (BoW) [1], and consists mainly of three steps: (i) extract local features, (ii) generate a codebook and then encode the local features into codes, (iii) pool all the codes together to generate a global image representation. In this approach a key step is the codebook generation, because it is the base to define a high-dimensional Bag Of Words histogram. Typically this is performed through clustering methods and the most common approach is the use of k-means clustering, because of its simplicity and convergence speed.

However, introducing a quantization of the feature space tightly ties dataset characteristics to the features representation, in the choice of both the position and the number of cluster centers to use. For the codewords positions, the quantization is learned from the training set, therefore the cluster centers reflect the training data distribution. The optimal number of cluster centers varies depending on the dataset. For example, in [2], the best accuracy using regular BoW is reached at 4k clusters for the Caltech-101 dataset, while, even if the improvement is progressively lower, in PASCAL VOC 2007 it does not reach saturation even with 25k cluster centers. Another example of this "hidden" dataset dependency inclusion may be found in many specializations of the BoW approach. [3] and [4] propose two different solutions to learn category specific codebooks and show how this is able to improve the descriptor ability to discriminate between similar categories.

The codebook generation step has been introduced in order to obtain a fixed length representation of the distribution of the local features of an image. This is not strictly necessary, since the descriptors distribution could be directly modeled with a parametric distribution [3, 5], and the parameters obtained on the single image may provide a summary of the local descriptors. In some contexts though, the information coming from the specific

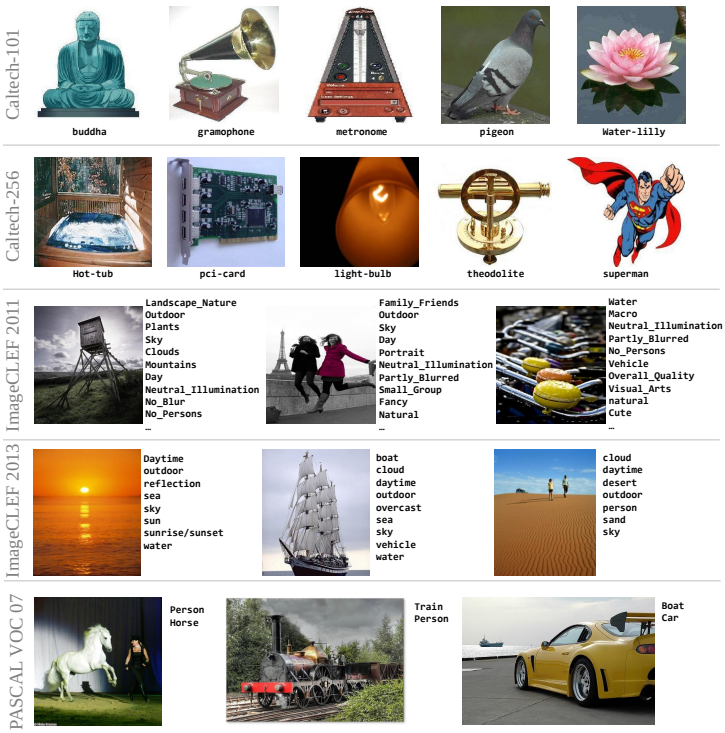

Figure 1: Sample images taken from the five datasets used in the experimental section. They pose different challenges both for object detection and multiple concepts annotation.

dataset characteristics is able to significantly boost the performance of the classification system.

Based on these considerations, in this paper we propose a solution to allow the descriptors to be obtained either in a dataset independent way or to leverage training information in their construction. Using a multivariate Gaussian distribution with full rank covariance matrix or a mixture of them it is possible to tune the system based on the context. We also show how to embed this descriptors in the Spatial Pyramid Representation [6] further removing border effects artifacts. The final image descriptor is then used both with an off-the-shelf batch classifier and with the Stochastic Gradient Descent on-line solver [7], which allows to deal with large scale datasets and high dimensional feature spaces.

We name our method Gaussian of Local Descriptors (GOLD) and demonstrate its effectiveness for automatic image annotation and object recognition. The main contributions of our work are:

- we provide a flexible local feature representation leveraging parametric probability density functions, that can be independent of the image archive (e.g. for collections that change dynamically) or specific to dataset characteristics;

- our method employs the projection of the full rank covariance matrix from the Riemmanian manifold to the tangent Euclidean space to obtain a fixed length descriptor suitable for linear classifiers based on dot product;

- we conduct experiments on several public databases (Caltech-101, Caltech-256, ImageCLEF2011, ImageCLEF2013, PASCAL VOC07). Some examples are reported in Fig. 1. The results demonstrate the effectiveness of utilizing our descriptor over different types of local features, both in dataset dependent and independent settings.

This paper is organized as follows. We introduce the state of the art on image descriptors focusing on encodings, normalizations and pooling strategies in Section 2. Then we elaborate the formulation of the GOLD descriptor in Section 3, and its combination with the spatial pyramid representation in Section 4. In Section 5 the extension to the mixture of Gaussian distributions is presented. We conduct extensive experiments in Section 6 to verify the advantage of our method for automatic image annotation and object recognition. Conclusions are drawn in Section 7.

2. Related Work

The basic component of all object recognition and scene understanding systems are local descriptors [8]. The most famous and effective ones are SIFT [9], and all their color variations [10].

After describing images with unordered sets of local descriptors, we would like to directly compare them in order to get information on the images similarities. The problem could be tackled with solutions inspired by the assignment problem, but this would be infeasible as soon as we move away from tiny problems. For this reason, research has focused on finding a fixed length summary of local descriptors density distribution.

The original solution, named Bag of Words, consists in finding a set of *codewords* (obtained by the *k-means* algorithm) and assigning each local feature to a codeword. The final descriptor is given by a histogram counting

the number of local features assigned to every codeword (cluster center) [1]. This last strategy was later referred to as *hard-assignment*.

A histogram is obviously a crude representation of the local features continuous density profile, it introduces quantization errors and it is sensitive to noise and outliers [11]. Thus, it would appear that by improving this density representation to more accurately represent the input feature set the classifiers performance could be improved as well [3]. For example, in [12] the hard-assignment of features is replaced with soft-assignment, which distributes an appropriate amount of probability mass to all codewords, depending on the relative similarity with each of them. The Locality-constrained Linear Coding [13] projects each descriptor on the space formed by its k-nearest neighbors (with small k; they propose $k = 5$). This procedure corresponds to performing the first two steps of the locally linear embedding algorithm [14], except that the neighbors are selected among the codewords of a dictionary rather than actual descriptors, and the weights are used as features instead of being mere tools to learn an embedding.

In [15] two supervised nonnegative matrix factorizations are combined together to identify latent image bases, and represent the images in this bases space; in [16] the authors propose to combine structures of input features and output multiple tags into one regression framework for multitag image annotation.

Fisher encoding [17], models the codewords with a Gaussian Mixture Model (GMM), restricted to diagonal covariance matrices for each of the k components of the mixture. Then, they capture the average first and second order differences between the image descriptors and the centers of the GMM.

The Vector of Locally Aggregated Descriptors [18] (VLAD) can be seen as a simplification of the Fisher kernel. Each local descriptor is associated to its nearest visual word. The idea of the VLAD descriptor is to accumulate, for each visual word, the differences of the vectors assigned to it, thus characterizing the distribution of the vectors with respect to the center. As for Fisher encoding, the descriptors are pooled together with averaging. Recently a comprehensive study concerning feature coding methods that summarizes their main characteristics including motivations and mathematical representations has been presented in [19].

The techniques discussed so far have all focused on improving the local descriptors encoding, relaying on training data for codewords generation. Given that there are a great number of unlabeled images available, some works focused on semi-supervised learning in order to leverage unlabeled

data for large-scale image annotation [20].

In order to overcome the dataset dependency, some authors tried to build a codebook in a fully data-independent way. In [21] the feature space is directly discretized using a regular lattice. With four subdivisions for each dimension, the number of bins is in the order of 10^{77}, most of which are obviously empty. They thus employ a hash table and store only the non-empty bins. Constant time table lookup, i.e., independent of the size of the visual vocabulary, can then be guaranteed. In [22] it is shown that this fixed quantization method performs significantly worse then other techniques, probably due to the fact that it splits dense regions of the descriptor space arbitrarily along dimension axes, and the bins do not equally split the unit hypersphere which SIFT covers, resulting in a wildly uneven distribution of points. Moreover they further highlight on Oxford [23] and Paris [24] datasets that the performance on drop of quantization approaches when generating codewords from a dataset and using them on another. Similar conclusions were also found in [25]. In short, referring to a configuration as dataset1/dataset2 (meaning that codewords are generated by dataset1 and used them for retrieval on dataset2), the Oxford/Oxford combination provides a mAP value of 0.673, against a Paris /Oxford mAP of 0.494. In a recent work [26], to avoid to recompute codewords at every dataset change, a particularly effective solution for cluster center adaptation, applicable to VLAD descriptors, is proposed. This, combined with an appropriate normalization step, shows a remarkable improvement when the codewords are generated from a different dataset. It is significant to note that the more different the codeword generation dataset is, the worse the performance are. Although the proposed adaptation is particularly efficient, it still requires to apply a transformation to all VLAD descriptors of the dataset.

A different strategy was proposed in [3], in order to avoid codeword generation completely, and in this way intrinsically remove any dataset dependency. The idea is to first model each set of vectors by a probability density function (pdf) and then compare them with a kernel defined over the pdfs. The advantage of modeling each image's set of descriptors independently are that each image model is tailored to the specific descriptor set and hence should be more accurate. This solution received little attention, because of the need of using specific kernels for image comparison, again posing scalability issues. Recently Carreira et al. [5] proposed to use second-order analogues of the most common first-order pooling operators to describe arbitrary shaped regions in semantic segmentation contexts. In particular,

they focused on multiplicative second-order interactions (e.g. outer products), together with either the average or the max operators. Following the techniques used in [27], they managed to obtain a region descriptor suitable for linear classifiers. It can be noted that, when average-pooling is used, this is exactly the proposal of [3], when the choice for the pdf is a zero-mean Gaussian distribution, improved with the mapping which allows to avoid the kernel computation between pdfs.

We propose to follow this latter way of modeling local features distributions, by choosing a multivariate Gaussian distribution with mean and full covariance as the reference pdf. By employing the log-Euclidean projection of [27], detailed in the next section, we can transform the distribution to a vectorial representation which allows to use the dot product to closely approximate a distance between distributions. Thanks to the fact that this representation is indeed modeling a Gaussian distribution, we can further extend it by changing the pdf to a mixture model, still obtaining a linear space representation. The idea of computing a Gaussian mixture model on the training set and then adapt it to each individual image as a descriptor was introduced in [28]. Although we share similar intentions, the following points mark the differences with their proposal: (i) they run a full EM algorithm on every image, while we only use the posterior probability to weight each feature contribution to every component, (ii) they employ a global diagonal covariance matrix, while we use a full one, (iii) they assume that this covariance matrix is fixed throughout the whole corpus, i.e. they do not re-estimate the image specific covariance matrix, (iv) the final image descriptor is dependent only on adapted weights and means of the various components, each scaled by the globally estimated covariance matrices.

Another proposal is strictly related to our approach: the recently introduced Vector of Locally Aggregated Tensors (VLAT) [29]. Their approach extends the VLAD descriptor by aggregating tensor products of local descriptors. They first compute a visual codebook of visual words over a sample image set using k-means. To compute the signature of an image, for each cluster, they aggregate with summation the centered tensors of centered descriptors. Each aggregated tensor is flattened into a vector and concatenated for all clusters. Strong similarities can be observed with our proposal, but: (i) we theoretically motivate our proposal by modeling a set of descriptors with a multivariate Gaussian distribution, while the second order tensor used in VLAT is centered w.r.t. the cluster mean; (ii) they do not normalize the descriptor with respect to the cardinality of the feature set; (iii) the main dif-

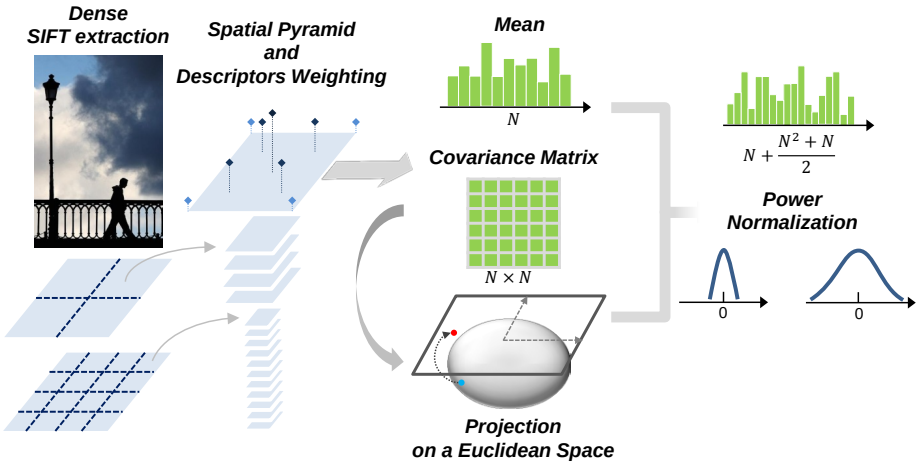

Figure 2: A schematization of the GOLD descriptor. An image is represented as a Weighted Pyramid of Gaussians of local descriptors. The covariance matrix is projected on the tangent space and concatenated to the mean to obtain the final region descriptor.

ference is that VLAT do not employ the log-Euclidean projection and simply simply vectorize the final tensors, assuming that these can be used with the Euclidean metric. Following our Gaussian motivation, we also include the mean to the final descriptor. The contributions of our choices are analyzed in Section 6.

As an additional improvement, we apply a spatial soft assignment over the spatial pyramid representation. A schematization of the proposed approach is presented in Fig. 2.

3. GOLD: Gaussian of Local Descriptors

In order to provide a tractable description of the inherently unknown pdf of an unordered set of feature vectors, we employ the most classical parametric distribution, that is the multivariate Gaussian distribution. Let $F = \{\mathbf{f}_1 \dots \mathbf{f_N}\}$ be the set of d-dimensional local features and suppose that they are independent and identically distributed samples from a multivariate Gaussian distribution, defined as

$$\mathcal{N}(\mathbf{f}; \mathbf{m}, \mathbf{C}) = \frac{1}{|2\pi\mathbf{C}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{f}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{f}-\mathbf{m})}, \tag{1}$$

where $|\cdot|$ is the determinant, $\mathbf{m}$ is the mean vector and $\mathbf{C}$ is the covariance matrix; $\mathbf{f}, \mathbf{m} \in \mathbb{R}^d$ and $\mathbf{C} \in \mathbb{S}_{++}^{d\times d}$, and $\mathbb{S}_{++}^{d\times d}$ is the space of real symmetric positive semi-definite matrices. The mean and covariance parameters are estimated from F as follows:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i, \tag{2}$$

$$\mathbf{C} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{f}_i - \mathbf{m})(\mathbf{f}_i - \mathbf{m})^T. \tag{3}$$

The estimated covariance matrix encodes information about the variance of the features and their correlation, and, together with the mean, provides a good insight on the set of features F. The space of covariance matrices can be formulated as a differentiable manifold, but not as a vector space (e.g. the covariance space is not closed under multiplication with a negative scalar). Unfortunately, many efficient machine learning algorithms assume that the data points form a vector space where dot product is defined, therefore they cannot readily work with covariance matrices.

It is important to consider that a manifold is a topological space that is locally similar to a Euclidean space. In particular a Riemannian manifold is a differentiable manifold in which each tangent space has an inner product, which varies smoothly from point to point [27].

Recently, it has been shown by Pennec et al. [30] that it is possible to endow the space of covariance matrices with an affine-invariant Riemannian metric (thus defining a Riemannian manifold), which allows to map covariance matrices to points in the Euclidean space.

The first step is the projection of the covariance matrices on an Euclidean space tangent to the Riemannian manifold, at a specific tangency matrix $\mathbf{P}$. The second step is the extraction of the orthonormal coordinates of the projected vector. In the following, matrices (points in the Riemannian manifold) will be denoted by bold uppercase letters, while vectors (points in the Euclidean space) by bold lowercase ones.

More formally, the projected vector of a covariance matrix $\mathbf{C}$ is given by:

$$\mathbf{t}_{\mathbf{C}} = \log_{\mathbf{P}}(\mathbf{C}) \triangleq \mathbf{P}^{\frac{1}{2}} \log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{C}\mathbf{P}^{-\frac{1}{2}}\right)\mathbf{P}^{\frac{1}{2}} \tag{4}$$

where log is the matrix logarithm operator and $\log_{\mathbf{P}}$ is the manifold specific logarithm operator, dependent on the point $\mathbf{P}$ to which the projection hy-

perplane is tangent. The matrix logarithm operators of a matrix C can be computed by eigenvalue decomposition ($\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{U}^T$); it is given by:

$$log(\mathbf{C}) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}(\mathbf{C} - \mathbf{I})^k = \mathbf{U} log(D)\mathbf{U}^T. \tag{5}$$

The orthonormal coordinates of the projected vector $\mathbf{t_C}$ in the tangent space at point $\mathbf{P}$ are then given by the vector operator:

$$\mathrm{vec}_{\mathbf{P}}(\mathbf{t_C}) = \mathrm{vec}_{\mathbf{I}}\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{t_C}\mathbf{P}^{-\frac{1}{2}}\right) \tag{6}$$

where $\mathbf{I}$ is the identity matrix, while the vector operator on the tangent space at identity of a symmetric matrix $\mathbf{Y}$ is defined as:

$$\mathrm{vec}_{\mathbf{I}}(\mathbf{Y}) = \left[y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3} \ldots y_{2,2}\ \sqrt{2}y_{2,3} \ldots y_{d,d}\right]. \tag{7}$$

Substituting $\mathbf{t_C}$ from Eq. 4 in Eq. 6, the projection of $\mathbf{C}$ on the hyperplane tangent to $\mathbf{P}$ becomes

$$\mathbf{c} = \mathrm{vec}_{\mathbf{I}}\left(\log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{C}\mathbf{P}^{-\frac{1}{2}}\right)\right). \tag{8}$$

Thus, after selecting an appropriate projection origin, every covariance matrix is projected to an Euclidean space. Since $\mathbf{c}$ is a symmetric matrix of size $d \times d$ a $(d^2 + d)/2$-dimensional feature vector is obtained.

As observed in [31], by computing the sectional curvature of the Riemmanian manifold [32], i.e., the natural generalization of the classical Gaussian curvature for surfaces, it is possible to show that this space is almost flat. This means that the neighborhood relation between the points on the manifold remain unchanged, wherever the projection point $\mathbf{P}$ is located. Therefore, from a computational point of view, the best choice for $\mathbf{P}$ is the identity matrix, which simply translates the mapping into applying the $\mathrm{vec}_{\mathbf{I}}$ operator to the standard matrix logarithm. This also frees us from the problem of optimizing the projection point for the specific data under consideration, leading to a generally applicable descriptor.

Finally, the unordered set of feature vectors F can be described by a Gaussian of local descriptors (GOLD), that is the concatenation of the mean and the orthonormal projection of the covariance matrix.

3.1. Normalization

In image classification systems, feature normalization techniques have the potential to greatly decrease the error rate of the classification, and thus increase the overall performance. When dealing with classifiers relaying on dot-product (such as linear SVMs) there is some recent convergence on the combined use of power normalization and unit length normalization using a L_2 metric [17, 2].

Power normalization consists in applying, to each dimension of the descriptor, the function:

$$f(x) = sign(x)|x|^{\alpha} \quad \text{with } 0 < \alpha < 1. \tag{9}$$

Perronnin et al. [17] justify the use of power normalization with the empirical observation that it has the ability of "unsparsifying" the representation, making it suitable for dot-product similarity. A different interpretation is provided in [33] where it is shown that applying the square root (a special case of the power normalization with $\alpha = 0.5$) is equivalent to employ the Hellinger's kernel (Bhattacharyya's coefficient). Moreover Safadi and Quénot [34] tested different normalization approaches and distance measures on several image descriptors, and observed that power normalization consistently leads to better performance. Moreover they optimized the α parameter for every descriptor and distance combination, and concluded that the optimal value when using dot product is approximately 0.5.

Motivated by these results, we apply power normalization to the GOLD vector, with $\alpha = 0.5$. While α optimization could slightly improve the performance, it would lead to a dataset-dependent tuning, again in contrast with our purposes.

4. Weighted Spatial Pyramid of GOLD

A standard way of introducing weak geometry in a Bag of Words representation is the use of spatial pyramids [6]. A spatial pyramid is a collection of feature histograms computed over subregions defined by a multilevel recursive image decomposition. At level zero, the decomposition consists of just a single region, and the representation is equivalent to the feature histogram of the entire image. At level one, the image is subdivided into four quadrants, yielding four feature histograms, and so on. The concept has been extended to several image representations by stacking the descriptors of every spatial region in a single vector.

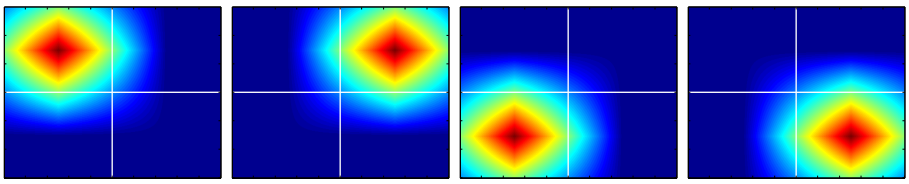

Figure 3: Bilinear interpolation applied to the spatial pyramid. The images depict the weights assigned to the local descriptors based on their positions with respect to the center of the spatial region under consideration, in the case of a 2×2 regions (level one of the spatial pyramid). The weights range from 1 (red) to 0 (blue). This means that a SIFT descriptor placed on the border between two spatial regions will be equally considered for both region descriptors.

However in this kind of representation the local features are hard-assigned to only one subregion, making the representation sensitive to border effects. For this reason, we follow an approach similar to [35], and apply a bilinear interpolation to spatial pyramids. We compute the GOLD vector of each region R, centered in (c_x, c_y) and with dimensions $w \times h$, on the local features that fall in the neighborhood R' with dimensions $2w \times 2h$, again centered at (c_x, c_y). A local feature $\mathbf{f}$, computed at (x, y), is then weighted, with respect to R, by

$$w(\mathbf{f}, R) = \left(1 - \frac{x - c_x}{w}\right)\left(1 - \frac{y - c_y}{h}\right). \tag{10}$$

A visual representation of the weights assigned to different positions in the case of a 2×2 regions is provided in Fig. 3.

In the original spatial pyramid formulation [6], histogram intersection was the kernel chosen to compare unnormalized BoW descriptors. This allowed to identify matches at different levels, and remove matches at finer levels (highly significant) from those at coarser ones (less significant). This led to the usually adopted per-level weights of 0.25, 0.25, 0.5, from coarse to fine, in a three levels pyramid. Later works tried to improve over the original proposal by learning the level weights [36], or the single regions weights [37, 38]. Again, these solutions are tailored for a specific dataset and lack of generality. A different strategy is instead followed in HOG descriptors, and later employed on the spatial pyramid by Harzallah et al. [39], that is independent L_2 normalization per region before constructing the final descriptor. This solution was later confirmed as the best choice in [17] and in [2].

GOLD descriptors are extracted from the weighted set of local features of every region, then they are power normalized. Finally, L_2 normalization

is employed, in order to avoid any learning step.

5. Mixture of GOLD

A possible extension of GOLD is to improve the model describing the local descriptors probability distribution. A natural choice would be to employ a Gaussian Mixture Model (GMM) instead of a single Gaussian. Unfortunately this is not as straightforward as it might seem, since the comparison of two GMMs would require the use of a complex kernel: the main problem for comparing GMMs is how to choose which component should be compared to whom, that is solving an assignment problem. The main advantage of GOLD was exactly the ability of avoiding kernel computations for efficient learning. We need a solution to perform a similar mapping leveraging a mixture of Gaussians.

We propose to start from a K-components GMM, learned from the training set with the EM algorithm:

$$p(\mathbf{f}|\Theta) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{f}; \mu_k, \Sigma_k) \tag{11}$$

where $\Theta = \{\omega_1, \mu_1, \Sigma_1, \ldots, \omega_K, \mu_K, \Sigma_K\}$. Similarly to what is done in soft quantization schemes [12], we can partially assign features to the k-th GMM component, according to the posterior probability for it:

$$\Pr(k|\mathbf{f}, \Theta) = \frac{\omega_k \mathcal{N}(\mathbf{f}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \omega_j \mathcal{N}(\mathbf{f}; \mu_j, \Sigma_j)} \tag{12}$$

It is now possible to build K multivariate Gaussian distributions from all the image descriptors, weighting them with the posterior probability of the k-th component. As in the maximization step of the EM algorithm, we estimate the Gaussian distribution parameters of the k-th component with the following equations:

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} \mathbf{f}_i \Pr(k|\mathbf{f}_i, \Theta)}{\sum_{i=1}^{N} \Pr(k|\mathbf{f}_i, \Theta)}, \tag{13}$$

$$\mathbf{C}_k = \frac{\sum_{i=1}^{N} (\mathbf{f}_i - \mathbf{m}_k)(\mathbf{f}_i - \mathbf{m}_k)^T \Pr(k|\mathbf{f}_i, \Theta)}{\sum_{i=1}^{N} \Pr(k|\mathbf{f}_i, \Theta)}. \tag{14}$$

The newly obtained Gaussian distributions are related to the GMM components originally estimated on the training set, but adapted to the specific set of local features. Their parameters can thus be used as descriptors for the local features distribution. As in Section 3 each Gaussian distribution can be mapped to a GOLD descriptor, obtaining a tuple of K GOLD vectors. These are then concatenated following the index of the corresponding GMM component. This allows us to directly compare images using a dot product operation, removing the need for non-linear kernel computations.

The concatenation of the K GOLD vectors is now our adapted projection of the original mixture. We will refer to this extension as *Mixture-GOLD*.

It is important to note that while this allows to have a highly informative descriptor for the feature space, it is based on a reference distribution (the GMM), whose parameters have been estimated on a training set.

The proposed technique is thus able to easily move from a codebook independent image description to a codebook based one, making it adaptable to different contexts and usage scenarios.

6. Experimental results

In order to analyze the proposed approach in different scenarios, we perform the experiments on five datasets: Caltech-101, Caltech-256, ImageCLEF 2011, ImageCLEF 2013 and PASCAL VOC07 (Fig. 1). Caltech datasets permit a wide comparison with a large number of techniques, while the ImageCLEF and PASCAL VOC07 datasets allow analyzing our proposal in less constrained and large-scale collections. In these two scenarios all the reported experiments are obtained with the dataset independent GOLD descriptor (single Gaussian) and the *Mixture*-GOLD descriptor showing the flexibility of our solution. In all experiments, SIFT feature descriptors and their color variations are extracted at four scales, defined by setting the width of the spatial bins to 4, 6, 8, and 10 pixels over a dense regular grid with a spacing of 3 pixels. We use the function `vl_phow` provided by the `vl_feat` library [40] with default settings.

For larger datasets (Caltech-256, ImageCLEF 2011, ImageCLEF 2013), we used the stochastic gradient descent (SGD) algorithm [41], introduced for SVM classifiers training, because it is an online method and can be easily parallelized to simultaneously train several classifiers. We randomize the data on disk and we load the data in chunks which fit in memory. We then train the classifiers on further randomizations of the chunks, so that different

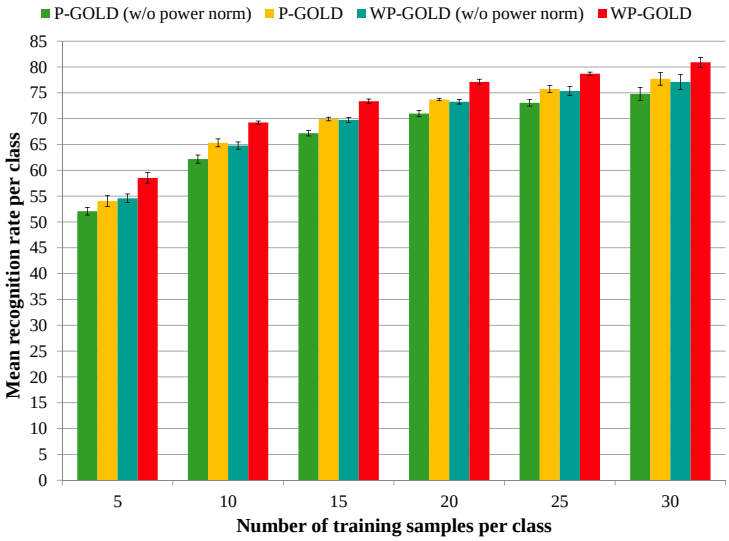

Figure 4: Performance of our approach on Caltech-101 for different settings, reported with different number of training samples. **P-Gold w/o power norm**: the Gaussian of local descriptors with the classical spatial pyramid procedure without the power normalization. **P-Gold**: the Gaussian of local descriptors with the classical spatial pyramid procedure with the power normalization. **WP-GOLD (w/o) power norm**: the Gaussian of weighted local descriptors of every spatial region without the power normalization. **WP-GOLD**: the Gaussian of weighted local descriptors of every spatial region with the power normalization.

epochs (one training epoch is defined as providing all training samples to the classifier once) will get the chunks data with different orderings.

The source code for the computation of our descriptors is publicly available for download to allow the community to reproduce our results[1].

6.1. Caltech-101 and Caltech-256

The Caltech-101 dataset is one of the most commonly used dataset for object recognition. It contains 9144 images from 101 object categories and one background category. The object categories can be very complex but a common viewpoint is chosen, with the object of interest at the center of the image at a uniform scale. The number of images per category varies from 31 to 800. The Caltech-256 dataset consists of 30,607 images divided

[1] `http://imagelab.ing.unimore.it/files/GOLD_image_classification.zip`

Table 1: Mean Recognition Rate per class on Caltech-101 when PCA is applied on SIFT descriptors. D is the number of principal components considered.

	PCA	**30 Training**
GOLD	D=128	80.92
GOLD	D=80	77.43
GOLD	D=64	77.13
GOLD	D=48	76.75
GOLD	D=32	74.70

Table 2: Mean Recognition Rate per class using 30 images training for five runs on Caltech-101.

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Average**
GOLD	80.53	82.38	79.90	80.45	81.33	80.92
Mixture-GOLD	80.61	82.43	79.95	80.56	81.34	80.98
VLAT_{ONE} [29]	76.39	77.38	74.61	75.08	76.56	76.00
VLAT [29]	78.58	80.24	78.50	78.31	78.77	78.88
HKM [33]	75.21	73.89	73.00	74.14	76.87	74.62

in 256 categories (with at least 80 images each). It presents a much higher variability in object size, location, and pose with respect to Caltech-101.

For both datasets we follow their respective common experimental settings: for Caltech-101 we randomly select 5, 10, 15, 20, 25, and 30 training images and at most 50 testing images for each category (this results in 3,060 images for training and 2,995 for testing in the 30 images test); for Caltech-256 we consider 30 and 60 training images and at most 50 for testing per class. We report the Mean Recognition Rate per class, i.e. the results are normalized based on the number of testing samples in that class and averaged over five independent runs. In all the experiment on Caltech datasets we extract SIFT descriptors, and images are analyzed with a 3 level pyramid, respectively partitioned in 1×1, 2×2 and 4×4 blocks.

The first experiment highlights the individual contribution of the mean and the projected covariance to the performance of the GOLD descriptor on Caltech-101 using 30 training images per class. The mean alone is obviously a very poor representation and therefore achieves a Mean Recognition Rate

Table 3: Comparison with the state-of-the-art for Caltech-101.

	15 Training	**30 Training**
GOLD	73.39	80.92
Mixture-GOLD	73.46	80.98
Bo et al. [42]	60.50	73.86
Grauman et al. [43]	50.00	58.20
Jia et al. [44]	-	75.30
Jiang et al. [45]	67.50	75.30
Liu et al. [46]	-	74.21
C. Zhang et al. [47]	69.58	75.68
Tuytelaars et al. [48]	69.20	75.20
Wang et al. [13]	65.43	73.40
Yang et al. [49]	67.00	73.20
Carreira et al. [5]	-	79.20
Lazebnik et al. [6]	56.40	64.60
Chatfield et al. [2]	-	77.78
Duchenne et al. [50]	75.30	80.30
Zeiler et al. [51]	83.80	86.50
He et al. [52]	-	93.42

of 30.19%, while the projected covariance obtains 80.83%. Concatenating the mean and the covariance, also due to a very high difference in dimensionality, slightly improves the performance, arriving to 80.92%.

Furthermore, to present the respective contributions of the power normalization and descriptors weighting steps, we report in Fig. 4 the performance gain given by these two procedures. Note that the Pyramidal-GOLD, i.e. the Gaussian of local descriptors with the classical spatial pyramid procedure, shows interesting results (P-GOLD w/o power norm), but the usage of the power normalization (P-GOLD) enhances the accuracy of about three percentage points. A similar improvement is obtained by including the weighting step (WP-GOLD w/o power norm). The combined use of both techniques (WP-GOLD), that weights the SIFT descriptors based on their spatial distribution and applies power normalization, further improves the accuracy of

about three percentage points. For simplicity, we will refer to this complete solution as GOLD.

Although the GOLD achieves a very good performance, the dimensionality of the final descriptor is quite large. For this reason, in Table 1 we present performance obtained by reducing the dimensionality of SIFT descriptors with PCA. After an initial drop, the performance slightly decreases until the dimensionality becomes 48, while for $D = 32$ we can observe a second important drop. These results motivate our choice of maintaining original (not reduced) SIFTs when using the single-Gaussian GOLD. However, when the Mixture-GOLD is employed, PCA becomes a necessary evil, in order to still have a tractable descriptors size.

As pointed out by Chatfield et al. [2] several works present results on the Caltech-101 dataset. However, missing details in the description of the methods or different tuning of the various components often make a fair comparison impossible. For this reason we firstly compare our method to VLAT [29] (that is the most similar approach) and the recently proposed approach by Vedaldi et al. [33], since they provide their code[2]. Results are shown in Table 2. For all of these methods we use the same experimental settings (same local features, same spatial pyramid and same classifier). For *Mixture*-GOLD and VLAT we use a codebook of 512 clusters (K=512) and SIFT are compressed to 48 dimensions using PCA, following [29]. We call VLAT_{ONE} the VLAT descriptor with K=1 and SIFT without PCA compression, that is directly comparable with our single-Gaussian GOLD. When using a single cluster, the VLAT_{ONE} descriptor describes the second order variation with respect to the training set mean, and this suffers from the lack of specificity with respect to the single image, but mostly from the lack of the projection on the tangent space. Rising the number of clusters definitely reduces the gap with respect to our proposal, but both GOLD and *Mixture*-GOLD show superior performance.

For completeness, Table 3 reports the results on Caltech-101 of several recent approaches that are quite comparable to our method. All of these use the same standard setting (15/30 samples for training, at most 50 for testing), and SIFT descriptors captured with dense sampling.

In addition, we include the results of Chatfield et al. [2] and Duchenne et al. [50] that rely on multiple features or test on a different number of images.

[2]http://www.vlfeat.org/applications/caltech-101-code.html

Table 4: Mean Recognition Rate per class for five runs on Caltech-101 using SGD algorithm.

Epochs	Run 1	Run 2	Run 3	Run 4	Run 5	Average
1	1.59	1.71	1.81	1.36	1.36	1.57
2	35.90	36.89	35.31	33.46	38.39	35.99
8	66.14	60.79	64.17	61.95	65.56	63.72
16	73.95	74.14	72.64	72.14	73.17	73.21
128	79.31	81.11	78.95	79.47	80.36	79.84
512	80.35	81.84	79.65	80.34	81.26	80.69
2048	80.56	82.30	79.70	80.50	81.27	80.87
4096	80.59	82.32	79.75	80.52	81.27	**80.89**

Finally, we report the latest results obtained with deep convolutional neural networks presented in Zeiler et al. [51] and He et al. [52], which clearly outperform every other traditional method, including ours.

The results reported for the Caltech-101 dataset were obtained with LibSVM, a well known software package for batch SVMs solving. The adoption of a batch solver was appropriate because feature data could entirely fit in memory, due to the limited size of the dataset. In order to verify the applicability of on-line solvers, we also trained the SVM classifiers using the SGD algorithm, starting from the public implementation provided by Leon Bottou[3]. In Table 4 the Mean Recognition Rate over the five runs at different number of training epochs is reported. Note that the results at the first epoch are very low for all runs, but they rapidly increase after few epochs. After 2048 epochs the SGD algorithm achieves good results, but only at 4096 epochs the SGD achieves the MRR score obtained with LibSVM (with a gap of only 0.03%), proving the efficacy of the on-line solver.

Lastly, in this section we report the results obtained on Caltech-256, as shown in Table 5. Since this dataset is larger than Caltech-101, for this experiment we employed the SGD solver and similar to Caltech-101 we fix the epochs equal to 4096. Also on this more challenging dataset, our method shows very competitive performance with respect to several SIFT-based tech-

[3]http://leon.bottou.org/projects/sgd

Table 5: Comparison with the-state-of-the-art for Caltech-256.

	30 Training	**60 Training**
GOLD	43.89	49.41
Mixture-GOLD	44.21	50.11
Bo et al. [42]	30.50	37.60
Yang et al. [49]	34.00	40.10
van Gemert et al. [54]	-	27.20
Perronnin et al. [17]	40.80	47.90
Tuytelaars et al. [48]	37.00	-
Wang et al. [13]	41.19	47.68
Duchenne et al. [50]	38.10	-
Cao et al. [53]	38.74	45.43
Zeiler et al. [51]	70.60	74.20

niques. Note also that the proposed approach obtains significantly better result than more complex techniques such as [50, 53]. As observed for Caltech-101, the deep convolutional neural network approach [51] has a significant advantage over all reported methods.

In both Caltech-101 and Caltech-256, the improvement in performance given by the *Mixture*-GOLD over the GOLD is only of some decimal points. We think that this behavior can be partially explained analyzing the characteristics of the datasets: as firstly demonstrated by [6], in these datasets the spatial pyramid is really effective, due to the homogeneous location and size of the objects. A good description of the spatial regions is therefore crucial to obtain a high recognition rate. The smaller the region, the stronger is the assumption (on which GOLD is based) that local descriptors follow a (single) Gaussian distribution, reducing the advantage of the GMM model used in *Mixture*-GOLD.

6.2. ImageCLEF 2011

ImageCLEF 2011 Annotation Task dataset is composed of a training set of 8000 images and a test set of 10000 images. The ImageCLEF 2011 photo corpus is a challenging concept detection dataset (multiple labels per image) due to its high heterogeneity of classes (see samples in Fig. 1). There are 99

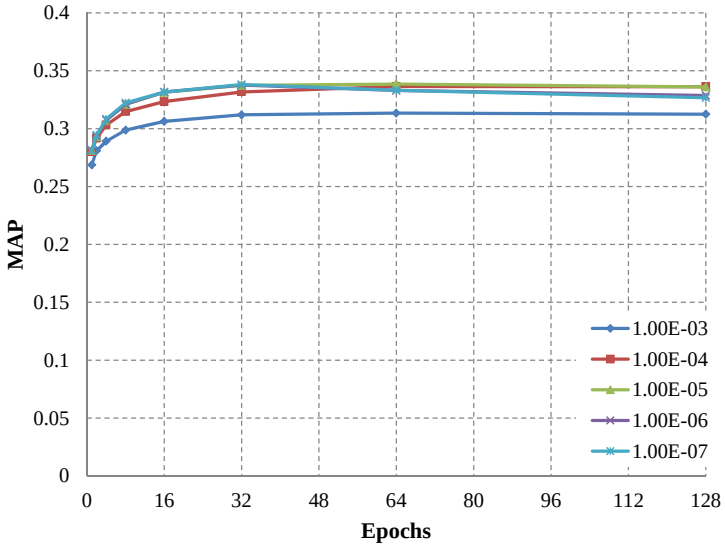

Figure 5: Cross validation results for choosing the parameter λ on the ImageCLEF 2011 dataset.

concepts, which are concrete objects such as "church" or "trees" as well as more abstractly defined classes like "funny" or "unpleasant".

On this dataset we extract RGBSIFT descriptors [10] at four scales (4, 6, 8, and 10 pixels respectively) over a dense regular grid with a spacing of 3 pixels and, even in this case, we use the function `vl_phow`. As spatial pyramids we use 1×1, 2×2 and 3×1. The Mean Average Precision (MAP) is used to evaluate the performance.

With larger datasets such as ImageCLEF 2011, an on-line learning approach (in our case SGD) becomes the only possible choice on common PCs. Only loading the entire training set in memory (8000 samples) occupies about 6GB, requiring to split the data in chunks.

To select an appropriate regularization parameter λ for the SGD solver, we randomly split the training set in two and run the SGD varying λ from 10^{-3} to 10^{-7} in power of 10 steps. Based on this preliminary experiments we fix $\lambda = 10^{-5}$ (see Fig. 5). Furthermore Fig. 6 reports the results in term of Mean Average Precision (MAP) at different number of training epochs. Note that the performance increases until the 64th epoch obtaining a MAP of 36.2, but thereafter the MAP tends to slightly decrease, probably due to an over-fitting of the SVM on the training data. The experiments show that is very difficult to predict the exact number of epochs necessary to reach the

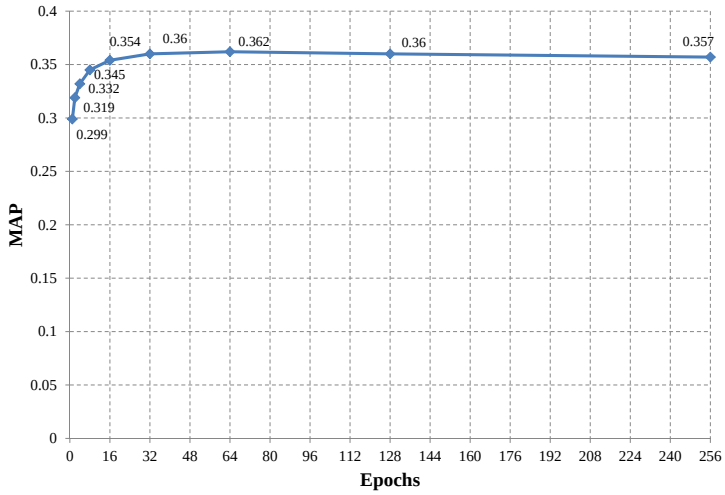

Figure 6: Mean Average Precision on ImageCLEF 2011 at different number of training epochs.

best results, and that even if there is a relation with the number of training samples and the size of the feature vectors, it is not a simple one. We found that the best practice is to run a k-fold cross-validation on the training set, which closely follow the final trend on the testing set.

Lastly we report in Table 6 a comparison with several techniques: in the upper part of the table we directly compare our approach with Bag of Words approaches (with linear and non-linear kernels) and very successful methods that publicly share their code [33, 13]. All of these methods use the same experimental settings (same local features, same spatial pyramid and same SGD classifier). For the Bag of Words approaches we use 4000 visual words since we observed that the performance tends to saturate at this codebook size, while, for the other techniques, we use the values suggested by the authors. In the table we also include the best run of the ImageCLEF workshop that obtained a MAP of 38.8 [55]. However these authors used three different color SIFT variations, different sampling strategies and improvements, and a Multiple Kernel Learning approach. Moverover, their computations required a cluster with 11,000 Core Units which had (according to cpubenchmark.net) a speed rank of 134 in August 2011. Our tests were performed on a 12 cores machine, which clearly limits the affordable computational effort. A more comparable approach, from a computational requirements point of view, was followed in [56], which used 7 color SIFT variations with both Harris and

Table 6: Comparison with the-state-of-the-art for ImageCLEF 2011.

	MAP
GOLD	36.20
Mixture-GOLD	37.65
BOW	25.06
BOW + Hellinger Kernel	33.87
Homogeneous Kernel Map [33]	34.72
Fisher Vectors	35.69
LLC [13]	34.12
Spyromitros-Xioufis et al. [56]	31.10
Binder et al. [55]	38.80

Dense sampling, leading to 14 separate classifiers per concept, combined with late fusion (averaging). They obtained a MAP of 31.1, clearly showing that the summarization properties of the GOLD and *Mixture*-GOLD representations, computed with only the basic RGBSIFT, are able to beat the description of the BoW approach.

6.3. ImageCLEF 2013

ImageCLEF 2013 Scalable Concept Image Annotation dataset is composed by 250,000 training images, obtained by querying popular image search engines (namely Google, Bing and Yahoo) when searching for words in the English dictionary. It includes various precomputed visual feature descriptors, extracted using the ColorDescriptor software [10], and textual features extracted from the websites in which the images appeared. It also provides a development and test sets of 1,000 and 2,000 images, respectively, both manually annotated for 95 and 116 concepts [57]. The competition objective is to develop systems that can easily change or scale the list of concepts used for image annotation.

Two possible strategies have been identified: i) finding images similar to the query, and from those extract the image concepts, leveraging the provided textual annotation; ii) directly using the textual annotation to roughly annotate the training set and then for every concept building a classifier applicable to the query. In the competition it has been shown that the second strategy largely outperforms the first one.

Table 7: Evaluation of our method with different local descriptors on ImageCLEF 2013.

Descriptor	**Baseline**	**GOLD**	***Mixture*-GOLD**
SIFT	28.32	36.02	38.43
RGBSIFT	29.50	38.53	40.12
OPPONENTSIFT	30.31	37.84	39.72

We tested this dataset for several reasons: it consists of a very large number of images; it is an unconstrained and challenging dataset, because it has a high heterogeneity of classes (mixed professional and user-generated content) and training images are not manually annotated.

In this experiment, following the second strategy, we compare our approach with SVM classifiers learned by the provided precomputed BoW [57]. Since the organizers computed the BoW features using a spatial pyramid of 1×1 and 2×2, we also used the same setting. In order to perform a fair comparison, all the techniques use the same textual annotation to select the image training set. Table 7 reports the performance in terms of MAP on the development set using three different local descriptors: SIFT, RGBSIFT and OPPONENTSIFT. It can be noted that our approach obtains superior MAP values with all of the three features.

In our best run at the ImageCLEF 2013 workshop [58], images are described using the GOLD descriptor computed on standard SIFT and on three different color SIFT variations, combined with a late fusion averaging approach. In this run, textual analysis on the web pages containing training images is also performed, to retrieve a relevant set of samples for learning each concept classifier based on WordNet lexical database. This run obtained the best result of the ImageCLEF 2013 workshop in terms of MAP: 45.6 (for more detailed results see [57][4]). Also in this dataset, the *Mixture*-GOLD is able to further improve the performance of the GOLD descriptor, of about 1.5 MAP points.

6.4. PASCAL VOC07

The PASCAL VOC07 dataset is a challenging archive for image classification with 9963 image divided in 20 classes of objects. Images are taken

[4] `http://imageclef.org/2013/photo/annotation/results`

from Flickr and have large variations in size, illumination, scale, and viewpoint. Classification accuracy is measured using Mean Average Precision (MAP) over the 20 classes following the common experimental protocol [2]. In this experiment we use the VLFeat library [40] that includes multiple encoding methods such as BOW, LLC, Super Vectors and Fisher Vectors. All the tested methods use densely extracted multi-scale SIFT descriptors, and images are partitioned with a 3 level pyramid: 1×1, 2×2 and 3×1. Following [2], for BOW and LLC the codebook size is set to 25000, for Super Vectors it is set to 1024, while Fisher vectors uses a GMM with $K = 256$ components, after reducing the dimensionality of the SIFT descriptor to 80 by using PCA. Similarly, for the *Mixture*-GOLD (M-GOLD) we used a GMM with $K = 16$ components and again a 80-dimensional PCA-SIFT.

Table 8 shows the performance of our method with respect to the other approaches. Although this dataset is very challenging, without specializing the GOLD image descriptor we are able to reach the performance of the BOW technique, which on the contrary requires to learn a very large and specific codebook. In order to achieve state of the art results, obtained by the Fisher Vector technique, introducing dataset dependency with GMM modeling is required. Moving from the GOLD to the *Mixture*-GOLD improves the performance of 6 MAP points getting results comparable with Fisher Vectors.

Therefore, our solution enables the user to choose between a descriptor which is effectively reusable when the image collection dynamically evolves, and one that provides better performance, thanks to the specific dataset characteristics.

7. Conclusions

In this paper we presented a new way to summarize local descriptors by means of multivariate Gaussian distributions. While still providing the possibility to include all the techniques which improve system performance, such as spatial pyramids and power normalization, this allows to obtain an image descriptor totally independent on the dataset. The experimental results show that the method achieves performance which are very competitive with state-of-the art approaches on several well-known datasets. This solution could be also employed in many different situations in which the dataset changes dynamically (for example in online services such as Flickr or Google Images), still allowing to use the same feature vectors in different scenarios.

Table 8: Comparison with the-state-of-the-art for PASCAL VOC07.

Class	GOLD	M-GOLD	BOW	LLC [13]	SV [59]	FV [17]
Aeroplane	76.45	77.58	67.29	71.35	74.32	78.97
Bicycle	58.26	65.57	55.22	62.65	63.79	67.43
Bird	41.14	51.75	36.58	46.12	47.02	51.94
Boat	70.51	76.39	64.42	68.98	69.44	70.92
Bottle	21.95	29.32	21.89	26.04	29.06	30.79
Bus	63.86	69.71	56.31	63.92	66.46	72.18
Car	75.02	78.16	72.90	76.98	77.31	79.97
Cat	61.02	63.12	52.11	59.71	60.18	61.35
Chair	52.09	54.12	51.51	53.96	50.19	55.98
Cow	36.96	47.70	38.23	46.34	46.46	49.61
Diningtable	48.51	58.35	46.50	52.10	51.86	58.40
Dog	36.33	46.27	34.99	42.39	44.07	44.77
Horse	78.01	79.98	74.62	77.17	77.85	78.84
Motorbike	65.19	69.63	60.71	67.15	67.12	70.81
Person	82.81	81.64	80.05	83.36	83.07	84.96
Pottedplant	19.75	30.28	18.79	23.11	27.56	31.72
Sheep	38.27	46.76	37.13	44.45	48.50	51.00
Sofa	47.75	59.41	50.22	52.12	51.10	56.41
Train	75.46	79.01	71.71	75.36	75.50	80.24
Tvmonitor	50.84	56.53	48.32	52.21	52.26	57.46
Mean	**55.01**	**61.06**	**51.97**	**57.27**	**58.16**	**61.69**

Furthermore an extension to a mixture of Gaussians is proposed, enhancing the image description considering context information. Its discriminative capability allows to boost classification results in specific scenarios.

References

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Statistical Learning in Computer Vision Workshop, 2004, pp. 1–12.

[2] 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, 2011, pp. 76.1–76.12.

[3] J. Farquhar, S. Szedmak, H. Meng, J. Shawe-Taylor, Improving "bag-of-keypoints" image categorisation: Generative Models and PDF-Kernels, Tech. rep., University of Southampton (2005).

[4] F. Perronnin, C. Dance, G. Csurka, M. Bressan, Adapted Vocabularies for Generic Visual Categorization, in: European Conference on Computer Vision, 2006, pp. 464–475.

[5] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic segmentation with second-order pooling, in: European Conference on Computer Vision, 2012, pp. 430–443.

[6] 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, 2006, pp. 2169–2178.

[7] L. Bottou, Large-Scale Machine Learning with Stochastic Gradient Descent, in: International Conference on Computational Statistics, 2010, pp. 177–186.

[8] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.

[9] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[10] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating Color Descriptors for Object and Scene Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596.

[11] B.-K. Bao, G. Zhu, J. Shen, S. Yan, Robust Image Analysis With Sparse Representation on Quantized Visual Features, IEEE Transactions on Image Processing 22 (3) (2013) 860–871.

[12] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, A. W. Smeulders, Kernel Codebooks for Scene Categorization, in: European Conference on Computer Vision, 2008, pp. 696–709.

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

[14] L. K. Saul, S. T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119–155.

[15] L. Jing, C. Zhang, M. Ng, SNMFCA: Supervised NMF-Based Image Classification and Annotation, IEEE Transactions on Image Processing 21 (11) (2012) 4508–4521.

[16] Y. Han, F. Wu, Q. Tian, Y. Zhuang, Image Annotation by Input-Output Structural Grouping Sparsity, IEEE Transactions on Image Processing 21 (6) (2012) 3066–3079.

[17] 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.

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

[19] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature Coding in Image Classification: A Comprehensive Study, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[20] Y. Yang, F. Wu, F. Nie, H. Shen, Y. Zhuang, A. Hauptmann, Web and Personal Image Annotation by Mining Label Correlation With Relaxed Visual Graph Embedding, IEEE Transactions on Image Processing 21 (3) (2012) 1339–1351.

[21] T. Tuytelaars, C. Schmid, Vector Quantizing Feature Space with a Regular Lattice, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[23] `http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/`.

[24] `http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/`.

[25] D. Nister, H. Stewenius, Scalable Recognition with a Vocabulary Tree, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.

[26] R. Arandjelović, A. Zisserman, All about VLAD, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 1578–1585.

[27] O. Tuzel, F. Porikli, P. Meer, Pedestrian Detection via Classification on Riemannian Manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1713–1727.

[28] X. Zhou, N. Cui, Z. Li, F. Liang, T. Huang, Hierarchical Gaussianization for image classification, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1971–1977.

[29] D. Picard, P.-H. Gosselin, Efficient image signatures and similarities using tensor products of local descriptors, Computer Vision and Image Understanding 117 (6) (2013) 680–687.

[30] X. Pennec, P. Fillard, N. Ayache, A Riemannian Framework for Tensor Computing, International Journal of Computer Vision 66 (1) (2006) 41–66.

[31] D. Tosato, M. Spera, M. Cristani, V. Murino, Characterizing Humans on Riemannian Manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1972–1984.

[32] I. Chavel, Riemannian Geometry: A Modern Introduction, Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2006.

[33] A. Vedaldi, A. Zisserman, Efficient Additive Kernels via Explicit Feature Maps, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 480–492.

[34] B. Safadi, G. Quenot, Descriptor optimization for multimedia indexing and retrieval, in: International Workshop on Content-Based Multimedia Indexing, 2013, pp. 65–71.

[35] V. Viitaniemi, J. Laaksonen, Spatial Extensions to Bag of Visual Words, in: ACM International Conference on Image and Video Retrieval, 2009, pp. 37:1–37:8.

[36] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, in: ACM International Conference on Image and Video Retrieval, 2007, pp. 401–408.

[37] J. He, S.-F. Chang, L. Xie, Fast kernel learning for spatial pyramid matching, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–7.

[38] T. Harada, Y. Ushiku, Y. Yamashita, Y. Kuniyoshi, Discriminative spatial pyramid, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2011, pp. 1617–1624.

[39] H. Harzallah, F. Jurie, C. Schmid, Combining efficient object localization and image classification, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 237–244.

[40] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, http://www.vlfeat.org/ (2008).

[41] L. Bottou, O. Bousquet, The tradeoffs of large scale learning, in: Neural Information Processing Systems, 2008, pp. 161–168.

[42] L. Bo, C. Sminchisescu, Efficient Match Kernel between Sets of Features for Visual Recognition, in: Neural Information Processing Systems, 2009, pp. 135–143.

[43] K. Grauman, T. Darrell, The Pyramid Match Kernel: Efficient Learning with Sets of Features, Journal of Machine Learning Research 8 (2007) 725–760.

[44] Y. Jia, C. Huang, T. Darrell, Beyond spatial pyramids: Receptive field learning for pooled image features, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3370–3377.

[45] Z. Jiang, G. Zhang, L. S. Davis, Submodular dictionary learning for sparse coding, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3418–3425.

[46] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: IEEE International Conference on Computer Vision, 2011, pp. 2486–2493.

[47] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, S. Ma, Image classification by non-negative sparse coding, low-rank and sparse decomposition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2011, pp. 1673–1680.

[48] T. Tuytelaars, M. Fritz, K. Saenko, T. Darrell, The NBNN kernel, in: IEEE International Conference on Computer Vision, 2011, pp. 1824–1831.

[49] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[50] O. Duchenne, A. Joulin, J. Ponce, A graph-matching kernel for object categorization, in: IEEE International Conference on Computer Vision, 2011, pp. 1792–1799.

[51] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, in: European Conference on Computer Vision, 2014, pp. 818–833.

[52] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, in: European Conference on Computer Vision, 2014, pp. 346–361.

[53] L. Cao, R. Ji, Y. Gao, Y. Yang, Q. Tian, Weakly supervised sparse coding with geometric consistency pooling, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3578–3585.

[54] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, J.-M. Geusebroek, Visual Word Ambiguity, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1271–1283.

[55] A. Binder, W. Samek, M. Kloft, C. Müller, K.-R. Müller, M. Kawanabe, The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task, in: CLEF 2011 working notes, 2011, pp. 1–9.

[56] E. Spyromitros-Xioufis, K. Sechidis, G. Tsoumakas, I. P. Vlahavas, MLKD's Participation at the CLEF 2011 Photo Annotation and Concept-Based Retrieval Tasks, in: CLEF 2011 working notes, 2011, pp. 1–15.

[57] M. Villegas, R. Paredes, B. Thomee, Overview of the ImageCLEF 2013 Scalable Concept Image Annotation Subtask, in: CLEF 2013 working notes, 2013, pp. 1–19.

[58] C. Grana, G. Serra, M. Manfredi, R. Cucchiara, R. Martoglia, F. Mandreoli, UNIMORE at ImageCLEF 2013: Scalable Concept Image Annotation, in: CLEF 2013 working notes, 2013, pp. 1–12.

[59] X. Zhou, K. Yu, T. Zhang, T. Huang, Image Classification Using Super-Vector Coding of Local Image Descriptors, in: European Conference on Computer Vision, 2010, pp. 141–154.