Modeling Local Descriptors with Multivariate Gaussians for Object and Scene Recognition

Giuseppe Serra Costantino Grana Marco Manfredi
Rita Cucchiara
Università degli Studi di Modena e Reggio Emilia, Modena MO 41125, Italy

ABSTRACT

Common techniques represent images by quantizing local descriptors and summarizing their distribution in a histogram. In this paper we propose to employ a parametric description and compare its capabilities to histogram based approaches. We use the multivariate Gaussian distribution, applied over the SIFT descriptors, extracted with dense sampling on a spatial pyramid. Every distribution is converted to a high-dimensional descriptor, by concatenating the mean vector and the projection of the covariance matrix on the Euclidean space tangent to the Riemannian manifold. Experiments on Caltech-101 and ImageCLEF2011 are performed using the Stochastic Gradient Descent solver, which allows to deal with large scale datasets and high dimensional feature spaces.

Categories and Subject Descriptors

H.3.1 [**Information Systems Applications**]: Content Analysis and Indexing

1. INTRODUCTION

Image classification, including scene classification and object recognition, remains a major challenge to visual analysis that provides a solid basis for many applications related to multimedia, such as cultural heritage and video-surveillance. As summarized in [3], the typical object recognition pipeline is composed of three steps: the extraction of local image features (e.g. SIFT), encoding of the local features in an image descriptor (e.g. a histogram of the quantized local features), classification of the image descriptor (e.g. by a support vector machine).

While all elements are important, the encoding of the local features in global image statistics is definitely the step attracting most research efforts. The baseline method is to compute a histogram of visual word frequencies (quantized local features), commonly known as Bag of Words model [4]. Recent advances replace the hard quantization of features involved in this method with alternative encodings that retain more information about the original image features, such as soft quantization [6], local linear encoding [18], and Fisher encoding [3].

Historically many solutions started to describe local features with a short and compact descriptor, but later researchers realized that the summarization was too crude and reverted to enrich it with further information. Since the goal is to describe the descriptors distribution within an image (or windows on it), a reasonable solution has been to use histograms to provide a compact non parametric description. Instead, we propose to employ a parametric distribution and compare its capabilities to histogram based approaches.

A reasonable first choice is to assume that our data follows a Gaussian distribution, because it has useful mathematical properties, it was extensively used and studied, and its representation requires few parameters. The Gaussian distribution plays a crucial role in multivariate statistics in general, and in discrimination theory in particular [1].

A major aspect in statistical learning is to quantify the similarity/dissimilarity between two distributions. Many measures have been proposed in closed form expressions such as the Bhattacharyya divergence and the symmetric Kullback-Leibler (KL) divergence between two multivariate Gaussian densities [10]. By leveraging these dissimilarities, it is possible to build a non-linear kernel function, to run a learning task using the measured Gaussian parameters. Unfortunately this would require an enormous computational effort and would become soon prohibitive when moving to large scale problems with large feature vectors.

In this paper we propose to model the SIFT descriptors distribution as a multivariate Gaussian and to transform the mean/covariance couple in a high dimensional vector: we concatenate the mean vector and the projection of the covariance matrix on the Euclidean space tangent to the Riemannian manifold. With our representation, linear classification is now possible, opening the way to efficient and large scale image annotation. Differently from common techniques based on the Bag of Words model, our solution does not rely on the construction of a visual vocabulary, thus removing the dependence of the image descriptors on the specific dataset considered. We report results on two commonly used datasets (Caltech-101 and ImageCLEF2011), using both an off-the-shelf batch classifier and the Stochastic Gradient Descent on-line solver, which allows to deal with large scale datasets and high dimensional feature spaces.

2. MULTIVARIATE GAUSSIAN OF LOCAL DESCRIPTORS

The multivariate Gaussian distribution of a set of d-di-

MM '13, Barcelona, Catalunya, Spain

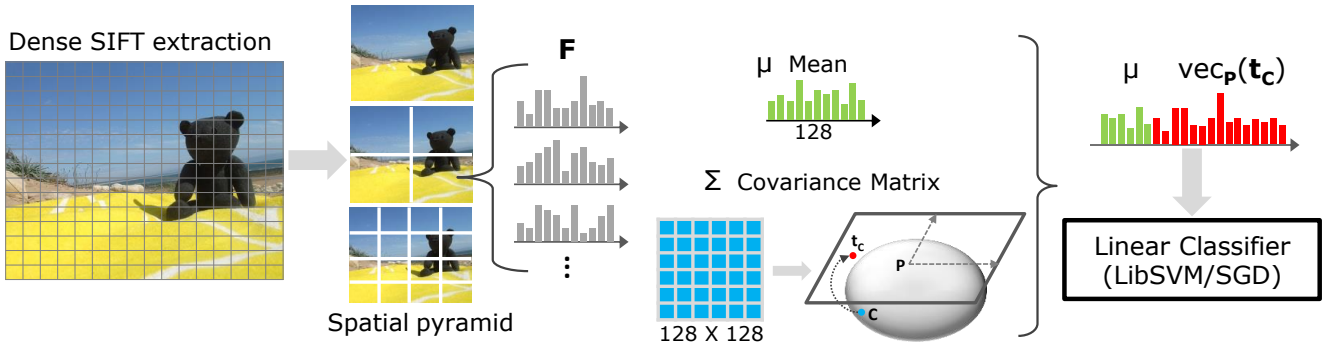

Figure 1: A schematization of the proposed approach. An image (or a window) is represented as a multivariate Gaussian distribution of the extracted local descriptors. The covariance matrix is projected on the tangent space and concatenated to the mean to obtain the final region descriptor.

mensional vectors F is given by

$$\mathcal{N}(\mathbf{f};\mu,\Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} exp\{-\frac{1}{2}(\mathbf{f}-\mu)^T\Sigma^{-1}(\mathbf{f}-\mu)\}, \quad (1)$$

where $|\cdot|$ is the determinant, μ is the mean vector and Σ is the covariance matrix ($\mathbf{f}, \mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{S}_{++}^{d\times d}$, with $\mathbb{S}_{++}^{d\times d}$ the space of real symmetric positive semi-definite matrices).

Let $F = \{\mathbf{f_1} \dots \mathbf{f_N}\}$ be a set of local features (e.g. SIFT descriptors, where $d = 128$) in an image (or a window) W, we suppose that they are normally distributed, and can thus be described with their distribution, that is by their mean and covariance. These can be estimated as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i, \quad (2)$$

$$\mathbf{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{f}_i-\mu)(\mathbf{f}_i-\mu)^T. \quad (3)$$

Although the covariance matrix is a very informative element of the distribution, which encode information about the variance of the features, it does not form a vector space. For example, the space is not closed under multiplication with negative scalars. Most of the common machine learning algorithms assume that the data points form a vector space, therefore a suitable transformation is required prior to their use. In particular we can observe that they are symmetric positive definite, and as such they can be formulated as a connected Riemannian manifold. 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.

In [16] an approach to map from Riemannian manifolds to Euclidean spaces is described. The first step is the projection of the covariance matrices on an Euclidean space tangent to the Riemannian manifold, on a specific tangency matrix $\mathbf{P}$. The second step is the extraction of the orthonormal coordinates of the projected vector.

The projected vector of a covariance matrix $\mathbf{C}$ is given by:

$$\mathbf{t_C} = \log_{\mathbf{P}}(\mathbf{C}) = \mathbf{P}^{\frac{1}{2}}\log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{C}\mathbf{P}^{-\frac{1}{2}}\right)\mathbf{P}^{\frac{1}{2}} \quad (4)$$

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

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:

$$\mathrm{vec}_{\mathbf{P}}(\mathbf{t_C}) = \mathrm{vec}_{\mathbf{I}}\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{t_C}\mathbf{P}^{-\frac{1}{2}}\right) \quad (5)$$

where $\mathbf{I}$ is the identity matrix, while the vector operator at identity is defined as:

$$\mathrm{vec}_{\mathbf{I}}(\mathbf{C}) = \left[c_{1,1}\ \sqrt{2}c_{1,2}\ \sqrt{2}c_{1,3}\dots c_{2,2}\ \sqrt{2}c_{2,3}\dots c_{d,d}\right] \quad (6)$$

Substituting $\mathbf{t_C}$ from Eq. 4 in Eq. 5, the projection of $\mathbf{C}$ on the hyperplane tangent to $\mathbf{P}$ becomes

$$\mathbf{c} = \mathrm{vec}_{\mathbf{I}}\left(\log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{C}\mathbf{P}^{-\frac{1}{2}}\right)\right) \quad (7)$$

In this way, after selecting an appropriate projection origin, every covariance matrix of size $d \times d$ gets projected to a $(d^2+d)/2$-dimensional feature vector on an Euclidean space.

As observed in [13], the projection point $\mathbf{P}$ is arbitrary and, even if it could influence the performance (distortion) of the projection, from a computational point of view, the best choice is the identity matrix, which simply translates the mapping into a standard matrix logarithm, followed by unrolling.

Summarizing, the proposed approach is to extract some descriptors from an image and then collect them in a spatial pyramid (Fig. 1); each sub-region is described by the estimated parameters of a multivariate Gaussian distribution. The covariance matrix is projected on a Euclidean space and concatenated to the mean vector to obtain the final descriptor. If SIFT descriptors are used, the dimensionality becomes 128 + (128*128+128)/2 = 8384 per sub-region. Eventually, all the sub-region descriptors are fed to a linear classifier.

3. MOVING TO LARGE SCALE DATA

Although there exist many off-the-shelf SVM solvers, such as SVMlight, SVMperf or LibSVM/LIBLINEAR, they are not feasible for training large volumes of data. This is because most of them are batch methods, which require to go through all data to compute gradient in each iteration and often need many iterations to reach a reasonable solution. Even worse, most off-the-shelf batch type SVM solvers require to pre-load training data into memory, which is impossible when the size of the training data explodes. Indeed,

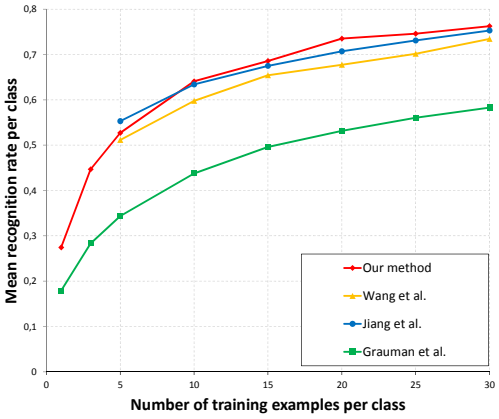

Figure 2: Comparison of results on the Caltech-101 dataset, reported with different number of training samples.

LIBLINEAR released an extended version that explicitly considered the memory issue, but in a recent test [11] it was shown that the performance dropped considerably and even on 80GB of training data it could not provide useful results. Therefore, a better solution may be provided by the stochastic gradient descent (SGD) algorithm recently introduced for SVM classifiers training.

We have training data that consists of T feature-label pairs, denoted as $\{\mathbf{x}_t, y_t\}_{t=1}^{T}$, where $\mathbf{x}_t$ is a $s \times 1$ feature vector representing an image and $y_t \in \{-1, +1\}$ is the label of the image. Then, the cost function for binary SVM classification can be written as

$$L = \sum_{t=1}^{T} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \max\left[0, 1 - y_t(\mathbf{w}^T \mathbf{x}_t + b)\right], \qquad (8)$$

where $\mathbf{w}$ is $s \times 1$ SVM weight vector, λ (nonnegative scalar) is a regularization parameter, and b (scalar) is a bias term. In the SGD algorithm, training data are fed to the system one by one, and the update rule for $\mathbf{w}$ and b respectively are

$$\begin{aligned} \mathbf{w}_t &= (1 - \lambda\eta)\mathbf{w}_{t-1} + \eta y_t \mathbf{x}_t \\ b_t &= b_{t-1} + \eta y_t \end{aligned} \qquad (9)$$

if margin $\Delta_t = y_t(\mathbf{w}^T \mathbf{x}_t + b)$ is less than 1; otherwise, $\mathbf{w}_t = (1 - \lambda\eta)\mathbf{w}_{t-1}$ and $b_t = b_{t-1}$. The parameter η is the step size. We set $\eta = (1 + \lambda t)^{-1}$, following the `vl_pegasos` implementation [17].

To parallelize SVMs training, we randomize the data on disk. We load the data in chunks which fit in memory, then train the different classifiers in parallel threads on further randomizations of the chunks, so that different epochs will get the chunks data with different orderings.

4. EXPERIMENTAL RESULTS

We tested the performance of our method on the Caltech-101 and ImageCLEF 2011. Caltech-101 is one of the most commonly used datasets for object recognition. It contains 9144 images from 101 object categories plus one background category. The number of images per category varies from 31 to 800. The images are with high shape variation, but objects are all centered and have no viewpoint diversity. ImageCLEF 2011 Annotation Task dataset is composed of a training set of 8000 images and the test set is 10,000 images large. The ImageCLEF photo corpus is a challenging concept detection dataset (multiple labels per image) due to its heterogeneity of classes. There are 99 concepts, which are concrete objects such as 'flowers', 'vehicles' as well as more abstractly defined classes like 'scary' or 'technical'.

Table 1: Comparison with the state-of-the-art for Caltech-101.

	15 Training	30 Training
Our method	**68.57**	**76.25**
Grauman et al. [7]	50.00	58.20
Jia et al. [8]	-	75.30
Jiang et al. [9]	67.50	75.30
Liu et al. [12]	-	74.21
Tuytelaars et al. [15]	69.20	75.20
Wang et al. [18]	65.43	73.40
Yang et al. [19]	67.00	73.20
Chatfield et al. [3]	-	77.78 *

* Note that Chatfield et al. tested on a slightly different setting (30 test images per class, in contrast to the standard 50)

For Caltech-101 we compute SIFT descriptors at multiple scales over a dense regular grid with a spacing of 4 pixels, using the function `vl_phow` provided by the `vl_feat` library [17]. For spatial pyramids we use 1×1, 2×2 and 4×4. We follow a common experimental setting: for training we randomly select 1, 3, 5, 10, 15, 20, 25, 30 images; for testing we randomly select at most 50 images for each category. We report the Mean Recognition Rate per class, the results are normalized based on the number of testing samples in that class. The reported results are the average over five independent runs.

For ImageCLEF 2011 we follow the same procedure to extract the SIFT descriptors, and the Mean Average Precision (MAP) is used to evaluate the performance.

Table 1 reports a comparison of the results our method with those of significant recent approaches obtained on the Caltech-101 dataset at two common number of training images per class (15 and 30). Our performance is definitely competitive with state-of-the art results, and is surpassed only by the Fisher Kernel results reported in Chatfield et al. [3]. Note that, differently from the common setting, they limit the number of testing images to only 30 images, so the results are not entirely comparable. Moreover our solution does not require to build a codebook, that must be trained on every specific dataset. All the reported techniques use SIFT descriptors extracted with dense sampling. Usually the more features or kernels are taken into account, the more the results improve (e.g. in [5] a combination of 48 different kernels achieves 74.6% with only 15 training images).

In Fig. 2 we also show the behavior of our approach at increasing numbers of training samples. As expected, the accuracy steadily improves with higher numbers training samples, as other techniques reported in literature that presented their results at different training settings.

The results reported for the Caltech-101 dataset were obtained with LibSVM, since the feature data could fit in memory and it is a well known and effective software package. We also trained the SVM classifiers using the SGD algorithm, starting from the public implementation provided by Leon Bottou[1], but the training time is largely higher and the results are slightly lower (74.23%).

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

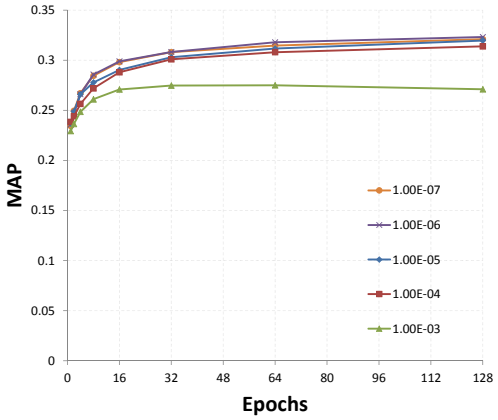

Figure 3: Cross validation results for choosing the parameter λ on the ImageCLEF2011 dataset.

Instead, with larger datasets such as ImageCLEF 2011, an on-line approach becomes the only viable choice, and its performance easily reach those of batch solvers. To select an appropriate regularization parameter λ, 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. Fig. 3 shows the MAP values at increasing number of training epochs (i.e. providing the algorithm all training samples in current split) for every λ value. While $\lambda = 10^{-3}$ leads to significantly worse results, all other values provide similar performance, with a slight improvement at $\lambda = 10^{-6}$, which we chose for our test. We then run the SGD algorithm on the whole 8000 training samples and tested on the testset, obtaining a MAP value of 0.332.

The best run of the ImageCLEF workshop obtained an impressive MAP of 0.388 using 4 different color SIFT variations, different sampling strategies and improvements and a Multiple Kernel Learning approach [2]. The feature and kernel computations required a cluster with 40 * 275 = 11,000 Core Units, which had (according to cpubenchmark.net) a speed rank of 134 in August 2011. Our tests were performed on a 2 * 6 cores machine, which clearly limits the affordable computational effort. A more comparable approach, from a computational requirements point of view, was followed in [14], which used 7 color SIFT variations with both Harris and Dense sampling, leading to 14 separate classifiers per concept, combined with late fusion (averaging). They obtained a MAP of 0.311, clearly showing that the summarization properties of our projected multivariate Gaussian descriptor, computed with only the basic gray-scale SIFT, are able to beat the description of the bag of visual words approach.

5. CONCLUSIONS

In this paper we proposed a new image representation based on the estimation of the multivariate Gaussian distribution of the SIFT descriptors, extracted with dense sampling on a spatial pyramid. Each distribution is converted to a high-dimensional descriptor, by concatenating the mean vector and the projection of the covariance matrix on the Euclidean space tangent to the Riemannian manifold. The reported results show promising performance, on par with state of the art approaches, without the need to build a codebook from training images.

6. REFERENCES

[1] S. Ali and S. Silvey. A general class of coefficients of divergence of one distribution from another. *J. of the Royal Stat. Soc. (B)*, 28(1):131–142, 1966.

[2] A. Binder, W. Samek, M. Kloft, C. Müller, K.-R. Müller, and M. Kawanabe. The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task. In *CLEF Workshop*, 2011.

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop Stat. Learn. Comput. Vision*, 2004.

[5] P. Gehler and S. Nowozin. On feature combination for multiclass object classication. In *ICCV*, 2009.

[6] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[7] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *J. Mach. Learn. Res.*, 8:725–760, 2007.

[8] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *CVPR*, 2012.

[9] Z. Jiang, G. Zhang, and L. S. Davis. Submodular dictionary learning for sparse coding. In *CVPR*, 2012.

[10] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE T. Commun. Techn.*, 15(1):52–60, 1967.

[11] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, and K. Yu. Large-scale image classification: Fast feature extraction and svm training. In *CVPR*, 2011.

[12] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011.

[13] S. Martelli, D. Tosato, M. Farenzena, M. Cristani, and V. Murino. An FPGA-based Classification Architecture on Riemannian Manifolds. In *DEXA Workshops*, 2010.

[14] E. Spyromitros-Xioufis, K. Sechidis, G. Tsoumakas, and I. P. Vlahavas. MLKD's Participation at the CLEF 2011 Photo Annotation and Concept-Based Retrieval Tasks. In *CLEF Workshop*, 2011.

[15] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The nbnn kernel. In *ICCV*, 2011.

[16] O. Tuzel, F. Porikli, and P. Meer. Pedestrian Detection via Classification on Riemannian Manifolds. *IEEE T. Pattern Anal.*, 30(10):1713–1727, 2008.

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

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

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