Image Classification with Multivariate Gaussian Descriptors

Costantino Grana, Giuseppe Serra, Marco Manfredi, and Rita Cucchiara

Università degli Studi di Modena e Reggio Emilia, Modena MO 41125, Italy

Abstract. Techniques based on Bag Of Words approach represent images by quantizing local descriptors and summarizing their distribution in a histogram. Differently, in this paper we describe an image as multivariate Gaussian distribution, estimated over the extracted local descriptors. The estimated distribution is mapped 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. To deal with large scale datasets and high dimensional feature spaces the Stochastic Gradient Descent solver is adopted. The experimental results on Caltech-101 and ImageCLEF2011 show that the method obtains competitive performance with state-of-the art approaches.

Keywords: image retrieval, image classification, multi-class, multi-label, stochastic gradient descent.

1 Introduction

Object and Scene Recognition have been a major research direction in computer vision. Recently, several local features, such as SIFT and HOG, are quite popular in representing images due to their ability to capture distinctive details of the images [14]. A common strategy to integrate the local features into a global representation is to use the the Bag Of Words approach, given its simplicity and effectiveness. It consists in three steps: extract local features, generate a codebook and then encode the local features into codes; pool all the codes together to generate the global image representation. The histogram is then fed to a classifier to predict the category [4].

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 a codebook is built by quantizing local feature descriptors extracted from training images. In recent years, there have been numerous vector quantization approaches to build visual codebooks, such as k-means clustering, or vocabulary trees [15]. However, generated codebooks are not sufficiently flexible to model heterogeneous kinds of new datasets. This is an underlying problem of the Bag Of Words approach, because every time the dataset (or more generally the context) changes the feature vector of an image must be recomputed. Other elements that have attracted research efforts are the encoding and pooling. The simplest encoding in the literature assigns a local feature to the closest visual word and computes a histogram

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 111–120, 2013.

of visual word frequencies [5]. A recent approach replaces the hard quantization of features with soft-assignment in which each local feature is assigned to multiple visual words [7]. In spite of their simplicity, these often introduce large quantization error and limits in the classification performance. To alleviate this problem, several authors have proposed alternative encodings that retain more information about the original image features, such as local linear encoding [22], and Fisher encoding [4]. Historically several solutions started to describe local features with a compact descriptor, but later researchers realized that the summarization was too crude and reverted to enrich it with further information. In addition, since the goal is to describe the descriptors distribution within an image, a reasonable solution has been to use histograms to provide a compact non parametric description.

Instead, we propose to use a parametric distribution and compare its capabilities and proprieties 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 [2]. In statistical learning a main aspect is to define function to measure similarity/dissimilarity between two distributions. Several measures in closed form expressions between two multivariate Gaussian densities have been proposed, such as the Bhattacharyya divergence and the symmetric Kullback-Leibler (KL) divergence [12]. These measures of divergence are positive, symmetric, and violate the triangle inequality [12], but recently a novel proposal meets the three metric axioms [1]. Based on these dissimilarities, it is possible to build a non-linear kernel function, which can be used in the classification process. However, this would require an enormous computational effort and would soon become prohibitive when moving to large scale classification problem with high-dimensional feature vectors.

In this paper we propose to represent the SIFT local features, extracted form an image, as a multivariate Gaussian, obtaining a mean vector and a covariance matrix. The covariance matrix, that lies on a Riemannian manifold, is projected on the Euclidean space tangent to the manifold and concatenated to the mean to obtain the final descriptor. 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. With linear classifiers, the proposed representation performs remarkably better than state-of-the-art approaches on several benchmarks (Caltech-101 and ImageCLEF2011), opening the way to efficient and large scale image classification. Results are obtained using both an off-the-shelf batch classifier and the Stochastic Gradient Descent (SGD) online solver, which allows to deal with large scale datasets and high dimensional feature spaces.

The main contributions are twofold: the definition of a novel image feature based on local descriptors summarized as a multivariate Gaussian, that does not require any codebook generation, well suited for linear classifiers; an in-depth comparison between the SGD online solver and the standard LibSVM in two different settings.

2 Multivariate Gaussian Descriptor

For an image W, we first extract features through densely sampling in a regular grid or using an interest point detector. Let $F = \{\mathbf{f}_1 \dots \mathbf{f_N}\}$ be a set of local features (e.g. SIFT descriptors, where $d = 128$) in W (or a sub-region of W, when Spatial Pyramid Matching is used), we describe them with a multivariate Gaussian distribution supposing that they are normally distributed. The multivariate Gaussian distribution of a set of d-dimensional vectors F is given by

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

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 $\mathbf{C} \in \mathbb{S}_{++}^{d\times d}$, with $\mathbb{S}_{++}^{d\times d}$ the space of real symmetric positive semi-definite matrices) defined as follows:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}_i \qquad \mathbf{C} = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{f}_i - \mathbf{m})(\mathbf{f}_i - \mathbf{m})^T. \tag{2}$$

The covariance matrix encodes information about the variance of the features and their correlation. Although it is very informative, it does not lie in a vector space since the covariance space is not closed under multiplication with a negative scalar. In fact, it lies in a Riemannian manifold. 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. Since the covariance matrix is symmetric positive definite we adopt the Log-Euclidean metric. The basic idea of the Log-Euclidean metric is to construct an equivalent relationship between the Riemannian manifold and the vector space of the symmetric matrix.

In [19] 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. 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:

$$\mathbf{t}_{\mathbf{C}} = \log_{\mathbf{P}}(\mathbf{C}) = \mathbf{P}^{\frac{1}{2}} \log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{C}\mathbf{P}^{-\frac{1}{2}}\right)\mathbf{P}^{\frac{1}{2}} \tag{3}$$

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 hyperplane 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:

$$log(\mathbf{C}) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}(\mathbf{C} - \mathbf{I})^k = \mathbf{U} log(D) \mathbf{U}^T. \tag{4}$$

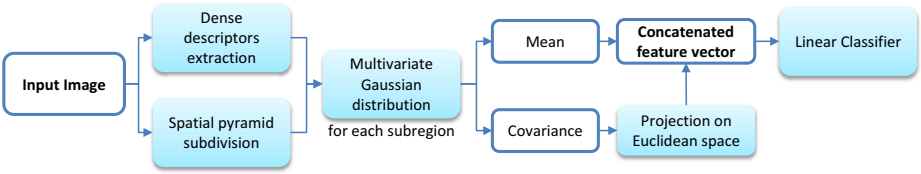

Fig. 1. Summary of the pipeline used in our proposal

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:

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

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

$$\text{vec}_{\mathbf{I}}(\mathbf{Y}) = \left[y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3}\ldots y_{2,2}\ \sqrt{2}y_{2,3}\ldots y_{d,d}\right]. \tag{6}$$

Substituting $\mathbf{t_C}$ from Eq. 3 in Eq. 5, the projection of $\mathbf{C}$ on the hyperplane tangent to $\mathbf{P}$ becomes

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

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.

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.

In short, our method is to extract local descriptors from an image and then collect them in a spatial pyramid; each sub-region is described by a multivariate Gaussian distribution. The covariance matrix is projected on a Euclidean space and concatenated to the mean vector to obtain the final descriptor (in the case of SIFT descriptors, the dimensionality becomes 8384 per sub-region). We empirically observe that most of the values in the concatenated descriptor are low, while few are high. In order to distribute the values more evenly, we adopt the power normalization method proposed by Perronnin et al. [16], i.e. to apply to each dimension the function $f(x) = sign(x)|x|^{\alpha}$ with $\alpha = 0.5$. Eventually, the concatenated descriptors are fed to a linear classifier. Fig. 1 summarizes the steps required by our proposal.

3 Online Learning for SVM Training

Off-the-shelf SVM solvers, such as LibSVM/LIBLINEAR or SVMlight are effective and well known solutions for train classifiers, however they are not fea-

sible for training large volumes of data. In fact, they 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 of them require to pre-load training data into memory, which is impossible when the size of the training data explodes. Therefore, we propose to use the stochastic gradient descent (SGD) algorithm, recently introduced for SVM classifiers training, because it is an online method and can be easily parallelized to simultaneously train several classifiers.

We have training data that consists of N feature-label pairs, denoted as $\{\mathbf{x}_t, y_t\}_{t=1}^{N}$, 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

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

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 randomized and fed to the system one by one, and the update rule for $\mathbf{w}$ and b respectively are:

$$\mathbf{w}_t = \begin{cases} (1 - \lambda\eta)\mathbf{w}_{t-1} + \eta y_t \mathbf{x}_t & \text{if } \Delta_t < 1 \\ (1 - \lambda\eta)\mathbf{w}_{t-1} & \text{otherwise} \end{cases}$$

$$b_t = \begin{cases} b_{t-1} + \eta y_t & \text{if } \Delta_t < 1 \\ b_{t-1} & \text{otherwise} \end{cases}$$

where $\Delta_t = y_t(\mathbf{w}^T \mathbf{x}_t + b)$. The parameter η is the step size and we adopt the following scheduling:

$$\eta = \eta_0 \frac{1}{(1 + \gamma n_0 t)^c} \tag{9}$$

where η_0, γ and c are some positive constants, and they are problem-dependent. Following the `vl_pegasos` implementation [20], we set $\eta_0 = c = 1$ and $\gamma = \lambda$.

To parallelize the computation for training SVM 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 epochs (one training epoch is defined as providing all training samples to the classifier once) will get the chunks data with different orderings. This last step of randomization turns out to be essential to make the SGD algorithm work properly.

4 Experimental Results

We perform the experiments on two different datasets: Caltech-101 and ImageCLEF 2011. Sample images of these datasets are shown in Fig. 2. Caltech-101 is

Fig. 2. Sample images taken from the Caltech-101 dataset (first row) and ImageCLEF2011 dataset (second row). The difference in terms of complexity is noticeable.

Table 1. 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
Our method	**79.40**	**81.39**	**78.44**	**78.70**	**80.47**	**79.68**
Vedaldi et al. [21]	75.21	73.89	73.00	74.14	76.87	74.62

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. ImageCLEF 2011 Annotation Task dataset is composed of a training set of 8000 images and a test set of 10000 images. 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 "church" or "trees" as well as more abstractly defined classes like "funny" or "unpleasant".

For Caltech-101, SIFT descriptors are extracted at four scales, defined by setting the width of the SIFT spatial bins to 4, 6, 8, and 10 pixels respectively, over a dense regular grid with a spacing of 3 pixels. We use the function `vl_phow` provided by the `vl_feat` library [20] and, apart from the spacing step, the defaults options are used. Images are hierarchically partitioned into 1×1, 2×2 and 4×4 blocks on 3 levels respectively. We follow a common experimental setting: for training we randomly select 15 and 30 images respectively; for testing we randomly select at most 50 images for each category (this results in 3,060 images for training and 2,995 for testing). 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[1].

[1] The lists of train/test images used for each run are public available at `http://imagelab.ing.unimore.it/files/caltech101_splits.zip`

Table 2. Comparison with the state-of-the-art for Caltech-101

	15 Training	**30 Training**
Our method	**71.62**	**79.68**
Grauman et al. [8]	50.00	58.20
Jia et al. [10]	-	75.30
Jiang et al. [11]	67.50	75.30
Liu et al. [13]	-	74.21
Tuytelaars et al. [18]	69.20	75.20
Wang et al. [22]	65.43	73.40
Yang et al. [23]	67.00	73.20
Chatfield et al. [4]	-	77.78*
Duchenne et al. [6]	75.30	80.30
Huang et al. [9]	66.88	74.25

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

As pointed out by Chatfield et al. [4] 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, as the first experiment, we compare our method to the recently proposed approach by Vedaldi et al. [21], since they provide their code[2], using, for feature extraction, the function `vl_phow` with the same parameters. We slightly modified their code in order to use exactly the same images for every defined run in both techniques. In fact, as reported in Table 1, the performance is closely linked to the choice of the training and testing images of each run and can vary of several percentage points. For example our accuracy is 81.39 in the second and 78.44 in the third run. Note that our method significantly outperforms the other method of about four percentage points in every run.

For completeness, in Table 2 we report the results on Caltech-101 of several recent approaches, compared with 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. Our performance is definitely competitive with state-of-the art results. Moreover, our solution does not require to build a codebook, that must be trained on every specific dataset. Note that, differently from the common setting, the Fisher Kernel results reported in Chatfield et al. [4] limit the number of testing images to only 30 images, so the results are not entirely comparable. In addition, we include the results of Duchenne et al. [6] and Huang et al. [9] for reference, although their approaches are high-order ones which perform costly alignment steps in kernel computation and are thus not strictly comparable with our approach.

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

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

Table 3. 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	3.73	3.89	3.88	3.95	3.86	3.86
2	25.99	25.20	12.76	13.78	16.12	18.77
8	47.22	45.44	46.68	39.55	44.33	44.65
16	63.32	63.07	61.54	62.31	63.58	62.77
128	71.62	69.48	70.11	68.40	71.75	70.27
512	74.74	75.05	73.18	75.11	76.47	74.91
2048	78.27	80.59	77.66	77.49	78.28	78.46
4096	79.46	81.62	77.99	78.88	79.70	**79.53**

the limited size of the dataset. We also trained the SVM classifiers using the SGD algorithm, starting from the public implementation provided by Leon Bottou[3]. In Table 3 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 0.15%), proving the efficacy of the online solver.

For ImageCLEF 2011 we extract OpponentSIFT descriptors 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`. Since the OpponentSIFT descriptor is a 384-dimensional feature, the multivariate Gaussian descriptor of an image (or a sub-region) would become an extremely large vector. For this reason, we obtain the final image feature by concatenating the multivariate Gaussian descriptors computed for each color channel separately. For spatial pyramids we use 1×1, 2×2 and 1×3. The Mean Average Precision (MAP) is used to evaluate the performance.

With larger datasets such as ImageCLEF 2011, an online learning approach (in our case SGD) becomes the only possible choice on common PCs. Only loading the entire training set on memory (8000 samples) occupies about 6GB, requiring to split the data in chunks (see Sec. 3). Using a more powerful PC, we are able to run the LibSVM batch solver reaching a MAP of 0.302. 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 $\lambda = 10^{-6}$. Fig. 3 reports the results in term of Mean Average Precision (MAP) at different number of training epochs. Note that the performance increases until the 32th epoch obtaining a MAP of 0.341 , but thereafter the MAP tends to decrease, probably due to an over-fitting of the SVM on the training data. On the ImageCLEF2011 dataset the SGD algorithm performs significantly better than LibSVM (we did not investigate the default stopping criterion). The experiments show that is very difficult to

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

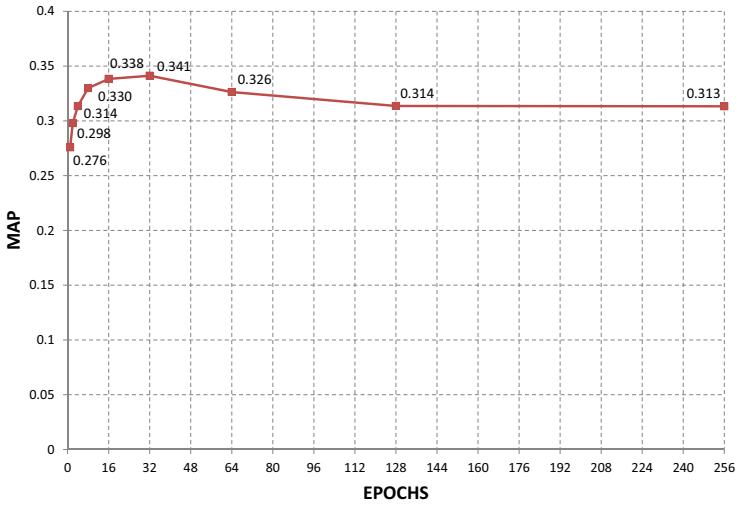

Fig. 3. Mean Average Precision on ImageCLEF 2011 at different number of training epochs

predict the exact number of epochs necessary to reach the 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.

The best run of the ImageCLEF workshop obtained a MAP of 0.388. However the authors used three different color SIFT variations, different sampling strategies and improvements, and a Multiple Kernel Learning approach [3]. Moverover, their computations required a cluster with 11,000 Core Units. 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 [17], 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 OpponentSIFT, are able to beat the description of the bag of visual words approach.

5 Conclusions

In this paper we presented a new image classification method that describes the extracted local SIFT descriptors as a multivariate Gaussian distribution. The estimated mean vector and the projection of the covariance matrix on the Euclidean space tangent to the Riemannian manifold are concatenated to define a high-dimensional descriptor. The experimental results show that the method achieves very competitive performance with state-of-the art approaches in two different datasets, Caltech-101 and ImageCLEF 2011.

References

1. Abou–Moustafa, K.T., De La Torre, F., Ferrie, F.P.: Designing a metric for the difference between two gaussian densities. Adv. Intel. Soft Comput. 83, 57–70 (2010)
2. Ali, S.M., Silvey, S.D.: A general class of coefficients of divergence of one distribution from another. J. of the Royal Stat. Soc (B) 28(1), 131–142 (1966)
3. Binder, A., Samek, W., Kloft, M., Müller, C., Müller, K.R., Kawanabe, M.: The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the Image CLEF2011 Photo Annotation Task. In: CLEF Workshop (2011)
4. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC (2011)
5. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop Stat. Learn. Comput. Vision (2004)
6. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: Proc. of ICCV (2011)
7. van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Smeulders, A.W.M.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
8. Grauman, K., Darrell, T.: The pyramid match kernel: Efficient learning with sets of features. J. Mach. Learn. Res. 8, 725–760 (2007)
9. Huang, Y., Huang, K., Wang, C., Tan, T.: Exploring relations of visual codes for image classification. In: Proc. of CVPR (2011)
10. Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: Receptive field learning for pooled image features. In: CVPR (2012)
11. Jiang, Z., Zhang, G., Davis, L.S.: Submodular dictionary learning for sparse coding. In: CVPR (2012)
12. Kailath, T.: The divergence and Bhattacharyya distance measures in signal selection. IEEE T. Commun. Techn. 15(1), 52–60 (1967)
13. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV (2011)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE T. Pattern Anal. 27(10), 1615–1630 (2005)
15. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: IEEE International Conference on Computer Vision and Pattern Recognition (2006)
16. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
17. Spyromitros-Xioufis, E., Sechidis, K., Tsoumakas, G., Vlahavas, I.P.: MLKD's Participation at the CLEF 2011 Photo Annotation and Concept-Based Retrieval Tasks. In: CLEF Workshop (2011)
18. Tuytelaars, T., Fritz, M., Saenko, K., Darrell, T.: The nbnn kernel. In: ICCV (2011)
19. Tuzel, O., Porikli, F., Meer, P.: Pedestrian Detection via Classification on Riemannian Manifolds. IEEE T. Pattern Anal. 30(10), 1713–1727 (2008)
20. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org/`
21. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. IEEE T. Pattern Anal. 34(3), 480–492 (2012)
22. T., Wang, Y.G.J., Yang, J., Yu, K., Lv, F., Huang: Locality-constrained linear coding for image classification. In: CVPR (2010)
23. Yang, T.J., Yu, K., Gong, Y., Huang: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)