Covariance of Covariance Features for Image Classification

Giuseppe Serra, Costantino Grana, Marco Manfredi and Rita Cucchiara
Dipartimento di Ingegneria "Enzo Ferrari", Università degli Studi di Modena e Reggio Emilia
Via Vignolese, 905/b, Modena MO 41125, Italy
name.surname@unimore.it

ABSTRACT

In this paper we propose a novel image descriptor built by computing the covariance of pixel level features on densely sampled patches and encoding them using their covariance. Appropriate projections to the Euclidean space and feature normalizations are employed in order to provide a strong descriptor usable with linear classifiers. In order to remove border effects, we further enhance the Spatial Pyramid representation with bilinear interpolation. Experimental results conducted on two common datasets for object and texture classification show that the performance of our method is comparable with state of the art techniques, but removing any dataset specific dependency in the feature encoding step.

Categories and Subject Descriptors

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

Keywords

image retrieval, image classification, covariance features

1. INTRODUCTION

Image representation for object and scene recognition have been a major research direction in computer vision and multimedia retrieval. The basic component of all state of the art systems are local descriptors. The most famous and effective ones are SIFT [12]. Once a set of local descriptors has been extracted from an image, it is necessary to summarize these information in a fixed length image feature. The Bag Of Words technique [2] has been successfully applied to solve this problem.

A great amount of research has dealt with feature encoding and pooling in the last years, considerably improving on the original BoW approach. The Locality-constrained Linear Coding [21] projects each descriptor on the space formed by its k-nearest neighbors (k is small, e.g., $k = 5$). This procedure corresponds to performing the first two steps of the locally linear embedding algorithm [14], except that the neighbors are selected among the atoms of a dictionary rather than actual descriptors, and the weights are used as features instead of being mere tools to learn an embedding.

Fisher encoding [13], models the visual words with a Gaussian Mixture Model (GMM) and captures the average first and second order differences between the image descriptors and the centers of the GMM. In the Vector of Locally Aggregated Descriptors [6] (VLAD), 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.

The techniques discussed so far have all focused on improving the local descriptors encoding, relaying on training data for codewords generation. We observe that the codewords training step introduces a dependency on the dataset on which they are computed, thus producing an image representation that inherently lacks of generality. Torralba and Efros [15] highlighted the presence of a "bias" in every dataset, which the classifier could relay on to improve its performance. Here we want to point out that the same dependency holds for the encoding step. Introducing a quantization of the feature space ties dataset characteristics to the image representation, in the choice of both the position and the number of cluster centers to use. In fact, the quantization is learned from the training set (e.g. using k-means), therefore the cluster centers reflect the training data distribution. Furthermore, the optimal number of cluster centers can vary depending on the dataset. For example, in [1], the best accuracy using regular BoW is reached at 4k clusters for the Caltech-101 dataset, while in Pascal VOC 2007 it does not reach saturation even with 25k cluster centers.

Recently a new local feature has gained interest in the image representation community: the covariance of pixel-level features. Firstly introduced by [18] for pedestrian detection, it has been successfully employed for texture classification and object tracking [11].

In this paper we propose to densely compute this local descriptor over an image, and incorporating it in a Weighted Spatial Pyramid image representation computing its covariance. This approach has been named Covariance of Covariance Features. The main contributions of our work include a deep analysis of pixel-level features and normalization techniques suitable for image classification and an extension of the Spatial Pyramid representation, that exploits bilinear interpolation to overcome border effects artifacts. The method

ICMR '14, April 01 - 04 2014, Glasgow, United Kingdom

http://dx.doi.org/10.1145/2578726.2578781.

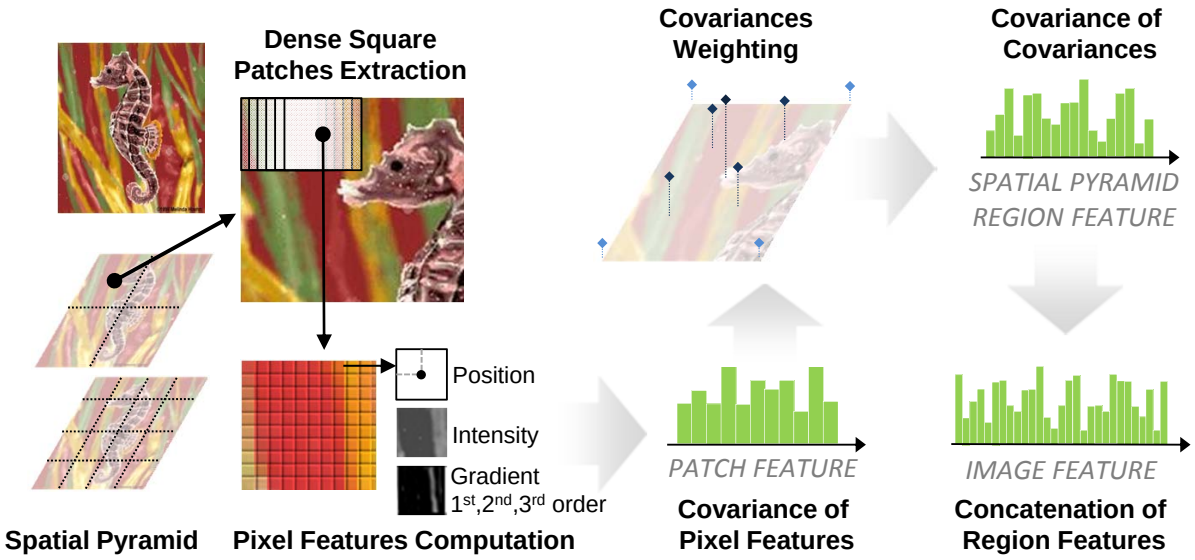

Figure 1: A schematization of the proposed approach.

has been tested on two public datasets of object and texture classification.

2. COVARIANCE OF COVARIANCES

To describe an image we follow a hierarchical approach that starts with pixel features, that are summarized to obtain *patch* features (small square image regions), that are in turn summarized to obtain *region* descriptors. The summarization is in both steps realized through covariance matrices.

Let $\mathbf{I}$ be a gray-level square image patch of size $r \times r$ and $\mathbf{f}(x, y)$ be a d-dimensional feature vector extracted at position $(x, y) \in \mathbf{L} = \{1..r, 1..r\}$; $\mathbf{f}(x, y)$ can include pixel position, intensity value and gradient information. To obtain a representation of patch $\mathbf{I}$, we summarize pixel features by computing their covariance matrix $\mathbf{C}$:

$$\mathbf{C} = \frac{1}{r^2 - 1} \sum_{(x,y)\in\mathbf{L}} (\mathbf{f}(x, y) - \bar{\mathbf{f}})(\mathbf{f}(x, y) - \bar{\mathbf{f}})^T, \qquad (1)$$

where $\bar{\mathbf{f}}$ is the mean pixel feature vector inside patch $\mathbf{I}$. The estimated covariance matrix encodes information about the variance of the features and their correlation, and provides a good insight on $\mathbf{I}$.

Since covariances belong to the Riemannian manifold of symmetric positive semi-definite matrices, Euclidean operations cannot be computed among them. In [18], a projection from the Riemannian manifold to a Euclidean tangent space is proposed, allowing, for example, the computation of the algebraic mean between the projected covariances. The tangency point is denoted as matrix $\mathbf{T}$. The aforementioned projection is obtained by:

$$\mathbf{C}' = \log_{\mathbf{T}}(\mathbf{C}) \triangleq \mathbf{T}^{\frac{1}{2}} \log\left(\mathbf{T}^{-\frac{1}{2}}\mathbf{C}\mathbf{T}^{-\frac{1}{2}}\right)\mathbf{T}^{\frac{1}{2}}, \qquad (2)$$

where $\log(\cdot)$ is the matrix logarithm. As observed in [16] the identity matrix is a suitable projection point, further simplifying the projection step which reduces to a simple matrix logarithm.

Since $\mathbf{C}'$ is also symmetric, its vectorized version, of size $d_{\mathbf{I}} = (d^2 + d)/2$, represents the patch feature:

$$\mathbf{F}_{\mathbf{I}} = vec(\mathbf{C}') = [\mathbf{C}'_{11}\mathbf{C}'_{12}\mathbf{C}'_{13}...\mathbf{C}'_{22}\mathbf{C}'_{23}...\mathbf{C}'_{dd}]. \qquad (3)$$

A rectangular region $\mathbf{R}$, that contains several overlapped patches, is described by computing the covariance of the patches features. The same problems encountered when computing patch features arise here, and the projection of the covariance matrices on a Euclidean space is therefore applied, along with the matrix vectorization. The feature vector thus obtained is $d_{\mathbf{R}} = (d_{\mathbf{I}}^2 + d_{\mathbf{I}})/2$ dimensional.

Several pixel features have been tested in our method, all of them exploiting only gray-level pixel information:

- pixel position $[x\ y]$;
- gray-level intensity $[I]$;
- gradient magnitude and orientation $[M\ O]$;
- first order derivatives $[|I_x|\ |I_y|]$;
- second order derivatives $[|I_{xx}|\ |I_{yy}|\ |I_{xy}|]$;
- third order derivatives $[|I_{xxx}|\ |I_{yyy}|\ |I_{xxy}|\ |I_{yyx}|\ ]$.

Operator $|\cdot|$ denotes the absolute value. Various combinations of pixel features have been investigated in the experimental section (Sec. 3).

Fig. 1 provides a summary of the proposed approach: local features are obtained by computing the covariance of pixel features on an image patch. Patch features contribute to build a region descriptor, weighted according to their distance from the region center. As a consequence, a patch is considered in more than one region (see Sec. 2.1). The features of the patches that contribute to a region are again encoded with their covariance. The final descriptor is given by the vector of all region covariances.

2.1 Spatial Pyramid Regions Weighting

Spatial Pyramid representation [10] is a fundamental technique in image classification, as it introduces spatial information in approaches like the Bag Of Words [2], that inherently lack geometric cues. The approach consists in recursively partitioning the image into non-overlapped regions, and concatenating regions features to form the image rep-

Table 1: Mean Recognition Rate per class using 15 training images on Caltech-101 with single patch size (16×16).

Feature vector	MRR
$[I]$	4.09
$[M\ O]$	19.13
$[\lvert I_x\rvert\ \lvert I_y\rvert]$	27.15
$[\lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert]$	38.97
$[\lvert I_{xxx}\rvert\ \lvert I_{yyy}\rvert\ \lvert I_{xxy}\rvert]$	37.73
$[x\ y\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert]$	55.32
$[x\ y\ M\ O\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert]$	60.38
$[x\ y\ M\ O\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert\ \lvert I_{xxx}\rvert\ \lvert I_{yyy}\rvert\ \lvert I_{xxy}\rvert]$	64.17
$[x\ y\ I\ M\ O\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert\ \lvert I_{xxx}\rvert\ \lvert I_{yyy}\rvert\ \lvert I_{xxy}\rvert\ \lvert I_{yyx}\rvert]$	65.20
$[x\ y\ I\ M\ O\ \lvert I_x\rvert\ \lvert I_y\rvert\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \lvert I_{xy}\rvert\ \lvert I_{xxx}\rvert\ \lvert I_{yyy}\rvert\ \lvert I_{xxy}\rvert\ \lvert I_{yyx}\rvert]$	**65.43**

resentation. Usually, the partition consists in dividing the image in four regions, and then dividing each region in four other sub-regions, leading to 21 regions in total (the entire image is also considered).

In our approach, we propose to enhance the spatial pyramid representation by introducing a bilinear interpolation. This consists in distributing each patch contribution between the neighboring regions, based on the patch distance from the region center. Given a patch $\mathbf{I}$ centered in $\mathbf{I}_c$ and a region $\mathbf{R}$ centered in $\mathbf{R}_c$ of size $w \times h$, the contribution of $\mathbf{I}$ for $\mathbf{R}$ is:

$$W_{\mathbf{R}}(\mathbf{I}) = \left(1 - \frac{\mathbf{I}_{cx} - \mathbf{R}_{cx}}{w}\right)\left(1 - \frac{\mathbf{I}_{cy} - \mathbf{R}_{cy}}{h}\right). \quad (4)$$

Considering that feature normalization plays an important role when describing images, we chose an appropriate normalization technique for each step of the algorithm: power normalization (applying a square root to each individual feature value, maintaining the sign) is applied to both patch features and to region features. After the concatenation of all the region features (to obtain the image representation), L2-normalization is applied. This last step of normalization has been proven to be appropriate when dealing with linear classifiers [1].

3. EXPERIMENTAL RESULTS

In this section, we test our approach in two different applications: object classification and texture recognition. First of all, we present a detailed analysis of the performance of the different pixel features on the well known Caltech-101 dataset. We demonstrate that our approach achieves performance comparable with state-of-the-art techniques, without requiring to build a codebook, thus overcoming any dataset dependency. Later we focus on KTH-TIPS dataset where we compare our approach with other state-of-the-art techniques based on covariance features. In all the experiments, a linear SVM was employed.

3.1 Object Classification

We use Caltech-101, since it represent a key benchmark for the object recognition community. 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.

Table 2: Comparison between different encondings and local features; several state-of-the-art solutions are also reported. All results are obtained with 15 training images on Caltech-101 with multiple patch sizes.

Feature	Encoding	MRR
Cov	BOW	37.29
Cov	Covariances	63.41
Cov	Covariances Weighting	67.07
Cov	Covariances Weighting + Power	**68.13**
SIFT	BOW	43.98
SIFT	BOW + Hellinger Kernel	46.23
SIFT	Homogeneous Kernel Map [20]	63.64
dHoG	LLC [21]	65.20
Lazebnik et al. [10]		56.40
Wang et al. [21]		65.43
Huang et al. [5]		66.88
Jiang et al. [7]		67.50
Tuytelaars et al. [17]		69.20
C. Zhang et al. [22]		69.58
Feng et al. [3]		70.34
Kong et al. [8]		75.10

As experimental protocol, we follow a common experimental setting: we randomly select 15 images for training and at most 50 images for testing for each category. Images are hierarchically partitioned into 1×1, 2×2 and 4×4 regions. We report the Mean Recognition Rate (MRR) 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 the first experiment we investigate how the different pixel features, employed in the covariance matrix, affect the overall performance. Each feature is extracted at a single scale, 16×16, over a dense regular grid with a spacing of three pixels. Table 1 presents the accuracy of each set of features, the best combination of two sets of features (6th Row), of three sets of features (7th Row) and so on. Results show that the best single features are the second order derivatives that obtain a MRR of 38.97%. In addition, enriching the feature vector with pixel coordinates leads to a large improvement of the performance: for example, the feature vector with second order derivatives and coordinates achieves 55.32%. In general, we observe that the more features we add, the better we get.

In the second experiment, we adopt the setting of other approaches, computing the patch features at four scales (16×16, 24×24, 32×32, 40×40) and use them with different embedding strategies. As shown in Table 2, the straightforward Bag of Words is not able to take advantage of the covariance feature properties and provides poor results. Instead, employing the covariance of covariance features strongly outperforms the Bag of Words approach. Our weighting strategy, based on bilinear interpolation, and the proper use of the power normalization further improve the results. In the middle part of the Table 2, we report a direct comparison with several techniques: Bag of Words approaches (with linear and nonlinear kernels) and very successful methods that publicly shared their code [20], [21]. 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

Table 3: Classification accuracy for the KTH-TIPS dataset.

Approach	Acc.
Our Method	**98.62**
P. Li et al. [11]	98.12
J. Zhang et al. [23]	95.40
Hayman et al. [4]	91.30
Lazebnik et al. [9]	91.3
Varma et al. [19]	92.4

suggested by the authors. All of these methods use the same experimental settings (same patch feature sizes, same spatial pyramid and same classifier). Results clearly state the effectiveness of our solution. For completeness, we include in Table 2 several recent solutions that are quite comparable to our method. Note that our approach obtains competitive results, while constructing dataset independent image features, as none of the others do. It is clear that recent dictionary learning techniques may definitely outperform our proposal at the price of being strongly dependent on the specific dataset.

3.2 Texture Classification

For texture classification, we test our technique on KTH-TIPS Dataset [4]. Images are captured at nine scales spanning two octaves (the relative scale changes from 0.5 to 2), viewed under three different illumination directions and three different poses, thus giving a total of 9 images per scale, and 81 images per ten materials. The size of samples is 200×200 pixels.

Texture recognition is as classic application where covariance features achieve one of its best performances. For this reason, we present a comparative evaluation of our approach with five texture classification methods. Table 3 shows the classification accuracy using 40 training images and the remaining for test. Even if all the techniques achieve very high results, this experiment shows that our image representation is robust and useful for different applications.

4. CONCLUSION

In this paper we showed how the hierarchical application of the covariance matrix descriptor is able to provide a very effective feature vector, which can be employed in object and texture classification problems. Despite their simplicity and reduced dimensionality, pixel features extracted from gray-level images demonstrate their effectiveness when summarized with our proposal. The use of bilinear interpolation jointly with the spatial pyramid representation further enhances the system performance.

5. REFERENCES

[1] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, pages 76.1–76.12, 2011.

[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *SLCV Workshop*, pages 1–12, 2004.

[3] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric lp-norm feature pooling for image classification. In *CVPR*, pages 2697–2704, 2011.

[4] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. In *ECCV*, pages 253–266, 2004.

[5] Y. Huang, K. Huang, C. Wang, and T. Tan. Exploring relations of visual codes for image classification. In *CVPR*, pages 1649–1656, 2011.

[6] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[7] Z. Jiang, G. Zhang, and L. S. Davis. Submodular dictionary learning for sparse coding. In *CVPR*, pages 3418–3425, 2012.

[8] S. Kong and D. Wang. A dictionary learning approach for classification: Separating the particularity and the commonality. In *ECCV*, pages 186–199, 2012.

[9] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *TPAMI*, 27(8):1265–1278, 2005.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, pages 2169–2178, 2006.

[11] P. Li and Q. Wang. Local Log-Euclidean Covariance Matrix (L^2ECM) for Image Representation and Its Applications. In *ECCV*, pages 469–482, 2012.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005.

[13] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[14] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *JMLR*, 4:119–155, 2003.

[15] A. Torralba and A. A. Efros. Unbiased Look at Dataset Bias. In *CVPR*, pages 1521–1528, 2011.

[16] D. Tosato, M. Spera, M. Cristani, and V. Murino. Characterizing humans on riemannian manifolds. *TPAMI*, 35(8):1972–1984, 2013.

[17] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In *ICCV*, pages 1824–1831, 2011.

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

[19] M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *CVPR*, pages II – 691–8 vol.2, 2003.

[20] A. Vedaldi and A. Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *TPAMI*, 34(3):480–492, 2012.

[21] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained Linear Coding for image classification. In *CVPR*, pages 3360–3367, 2010.

[22] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *CVPR*, pages 1673–1680, 2011.

[23] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV*, 73(2):213–238, 2007.