# TRUNCATED ISOTROPIC PRINCIPAL COMPONENT CLASSIFIER FOR IMAGE CLASSIFICATION

*Alessandro Rozza*

Hyera Software
Research Team
Coccaglio BS, Italy
*alessandro.rozza@hyera.com*

*Giuseppe Serra, Costantino Grana*

Università di Modena e Reggio Emilia
Dipartimento di Ingegneria "Enzo Ferrari"
Modena MO, Italy
{*giuseppe.serra, costantino.grana*}*@unimore.it*

## ABSTRACT

This paper reports a novel approach to deal with the problem of Object and Scene recognition extending the traditional Bag of Words approach in two ways. Firstly, a dataset independent method of summarizing local features, based on multivariate Gaussian descriptors, is employed. Secondly, a recently proposed classification technique, particularly suited for high dimensional feature spaces without any dimensionality reduction step, allows to effectively exploit these features. Experiments are performed on two publicly available datasets and demonstrate the effectiveness of our approach when compared to state-of-the-art methods.

***Index Terms***— Truncated isotropic principal component classifier, image retrieval, image classification, multi-class classification.

## 1. INTRODUCTION

In the last decade a great deal of research has been devoted to Object and Scene recognition. In order to capture distinctive details of the images, most of the image representation techniques leverage local features, such as `SIFT` and `HOG` [1]. Anyway, the usage of local features as they are would require to solve an assignment problem between every image pair, thus making it unfeasible to use them in real world scenarios. For this reason, a common strategy to integrate the local features into a fixed length global representation is to use the Bag of Words approach. This technique is roughly composed of three steps: the local features extraction, the codebook generation and local features encoding, and the code pooling to generate the global image representation [2].

The codebook generation is a key step, because it provides a base to define a high-dimensional Bag of Words histogram. Typically a codebook is built by quantizing local feature descriptors extracted from training images. 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 encoding and pooling.

Despite their simplicity, these techniques introduce large quantization errors and limit 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 [3], and Fisher encoding [2].

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. In [4] the authors propose to represent `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, this solution does not rely on the construction of a visual vocabulary, thus removing the dependence of the image descriptors on the specific dataset. The experimental results reported in [4] confirm that these features are well suited for linear classifiers.

Considering the aforementioned approach, the obtained final descriptors are high dimensional vectors. It is important to note that when data are encoded in a high dimensional space, many techniques cannot be applied due to their high time and space complexity. Moreover, when the "small sample size problem" [5] occurs, that is when the dimensionality of the feature space is higher than the number of available training data, the underlying mathematical formulations of several learning algorithms could yield poor performance, since the amount of training data is not enough to compute reliable estimates of the employed mathematical entities. Besides, a related problem is overfitting, which could happen

when the number of available samples is too small compared to the space dimensionality and the model is complex enough. All these problems can reduce the classification performance.

For these reasons in this work we propose a novel classification framework that employs the multivariate Gaussian descriptors and exploits the peculiarities of a Fisher discriminant classifiers called Truncated Isotropic Principal Component Classifiers (T−IPCAC, [6]). This classifier is particularly suitable since it is developed to deal with high dimensional data without performing any dimensionality reduction step that might delete discriminative information as stated in [6]. Furthermore, this approach is based on the same assumption of Gaussianity of the employed features and this could improve the performance of the classification phase. The proposed framework performs remarkably better than state-of-the-art approaches on several benchmarks (the Caltech 101 dataset and the Fifteen Scene Categories Database) confirming the quality of our approach.

The paper is organized as follows: in Section 2 the employed features are described; in Section 3 T−IPCAC is summarized; in Section 4 the experimental results on standard benchmarks are presented; finally, conclusions and future works are summarized in Section 5.

## 2. MULTIVARIATE GAUSSIAN DESCRIPTOR

For an image $\boldsymbol{Z}$, we first extract local features through densely sampling in a regular grid. Based on these features $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_1 \dots \boldsymbol{x}_N\}$ (e.g. SIFT descriptors, $\mathbf{x}_i \in \Re^d$ where $d = 128$) we describe them with a multivariate Gaussian distribution supposing that they are normally distributed. The Multivariate Gaussian distribution of a set $\boldsymbol{\mathcal{X}}$of $d$-dimensional vectors is given by:

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}, \tag{1}$$

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

The covariance matrix encodes information about the variance of the features and their correlation thus enhancing the discriminative capabilities of individual features. However it does not lie in a vector space. In fact, the covariance space is a Riemannian manifold and is not closed under multiplication with a negative scalar. 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 observe that the covariance matrix is symmetric positive definite therefore we can 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 [7] 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 $\boldsymbol{T}$. 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. The projection vector of a covariance matrix $\boldsymbol{\Sigma}$ on the tangency matrix $\boldsymbol{T}$ is given by:

$$\boldsymbol{c} = \text{vec}_{\boldsymbol{I}}\left(\log\left(\boldsymbol{T}^{-\frac{1}{2}}\boldsymbol{\Sigma}\boldsymbol{T}^{-\frac{1}{2}}\right)\right), \tag{2}$$

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

$$\text{vec}_{\boldsymbol{I}}(\boldsymbol{Y}) = \left[y_{1,1}\ \sqrt{2}y_{1,2}\ \dots \sqrt{2}y_{1,d} \dots\ y_{d,d}\right]^T. \tag{3}$$

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

As observed in [8], the tangency matrix $\boldsymbol{T}$ 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.

The image signature representation is the concatenation of the mean vector and the projected vector $\boldsymbol{c}$ thus obtaining, in the case of SIFT descriptor, a point $\boldsymbol{p} \in \Re^{8384}$. Finally, it is empirically possible to observe that most of the values in the concatenated descriptor are low, while few are high. In order to distribute the values more evenly, the power normalization method proposed by Perronnin et al. [9] is adopted.

## 3. TRUNCATED ISOTROPIC PRINCIPAL COMPONENT CLASSIFIER

In this section we describe the employed binary classification method and how we extend this classifier to the multi-class classification.

The first version of T−IPCAC, called Isotropic Principal Component Classifier (IPCAC), has been proposed in [10]. It is a binary classifier that exploits theoretical results presented by Brubaker and Vempala [11] to efficiently estimate the Fisher subspace (Fs). Precisely, in [11] it is shown that, given a set of $N$ clustered points sampled from an isotropic Mixture of Gaussians, Fs corresponds to the span of the class means; as a consequence, when a binary classification problem is considered, Fs is spanned by unit vector $\boldsymbol{f} = \frac{\boldsymbol{\mu}_A - \boldsymbol{\mu}_B}{||\boldsymbol{\mu}_A - \boldsymbol{\mu}_B||}$, being $A$ and $B$ the two classes, and $\boldsymbol{\mu}_{A/B}$ the class means.

IPCAC exploits this result by whitening[1] the training set $\boldsymbol{\mathcal{P}}_{Train} = \{\boldsymbol{p}_1 \dots \boldsymbol{p}_N\}$, computing $\boldsymbol{f}$, and classifying a new point $\boldsymbol{p}$ as follows:

$$\theta((\boldsymbol{W}_D^T \boldsymbol{f}) \cdot \boldsymbol{p} - \gamma) = \theta(\boldsymbol{w} \cdot \boldsymbol{p} - \gamma); \quad \gamma = \left\langle \underset{\bar{\gamma} \in \{\boldsymbol{w} \cdot \boldsymbol{p}_i\}_{i=1}^N}{\arg\max} \; Score(\bar{\gamma}) \right\rangle \tag{4}$$

where $\theta(x) = A$ if $x \geq 0$, $\theta(x) = B$ if $x < 0$, the matrix $\boldsymbol{W}_D$ represents the whitening transformation estimated on the $N$ training points, $Score(\bar{\gamma})$ computes the number of correctly classified training points when $\bar{\gamma}$ is used as threshold, and $\langle \cdot \rangle$ represents the average operator (we may have multiple $\bar{\gamma}$ corresponding to the maximum of the $Score$ function).

Unfortunately, the high computational complexity of classifiers based on the estimation of Fs prevents their application to high dimensional datasets. Moreover, this kind of technique often fails when the training-set cardinality is equal or lower than the space dimensionality. To overcame these limitations, T-IPCAC [6] improves IPCAC, and reduces the computational complexity, by replacing the first step of data whitening by a 'partial whitening' process; if the points to be classified belong to a $D$ dimensional space, this method whitens the data in the linear subspace $\pi_d = \mathbf{Span}\langle \boldsymbol{v}_1, \cdots, \boldsymbol{v}_d \rangle$, spanned by the first $d \ll D$ principal components, while maintaining unaltered the information related to the orthogonal subspace $(\pi_d)^\perp = \mathbf{Span}\langle \boldsymbol{v}_{d+1}, \cdots, \boldsymbol{v}_D \rangle$.

Precisely, the linear transformation $\boldsymbol{W}_D$ representing the partial whitening operator is estimated as follows. The Truncated Singular Value Decomposition [12] is applied to estimate the first $d << D$ principal components[2], obtaining the low-rank factorization $\boldsymbol{P} \simeq \boldsymbol{U}_d \boldsymbol{Q}_d \boldsymbol{V}_d^T$ (where $\boldsymbol{P}$ is the matrix representing the training set $\boldsymbol{\mathcal{P}}_{Train}$ since it contains the training vectors). The $d$ largest singular values on the diagonal of $\boldsymbol{Q}_d$, and the associated left singular vectors, are employed to project on the subspace $\boldsymbol{\mathcal{SP}}_d$, spanned by the columns of $\boldsymbol{U}_d$, and to perform the whitening on the points contained in $\boldsymbol{P}$:

$$\bar{\boldsymbol{P}}_{\boldsymbol{W}_d} = q_d \boldsymbol{Q}_d^{-1} \boldsymbol{P}_{\perp SP_d} = q_d \boldsymbol{Q}_d^{-1} \boldsymbol{U}_d^T \boldsymbol{P} = \boldsymbol{W}_d \boldsymbol{P} \tag{5}$$

where $q_d$ is the smallest singular value of the points projected in $\boldsymbol{\mathcal{SP}}_d$. Note that, to obtain points whose covariance matrix best resembles a multiple of the identity, the value of the $d$ largest singular values is set to $q_d$ instead of 1, thus avoiding the gap between the $d$-th and the $(d+1)$-th singular value. The obtained matrix $\boldsymbol{W}_d$ projects and whitens the points in the linear subspace $\boldsymbol{\mathcal{SP}}_d$; however, dimensionality reduction during the whitening estimation might delete discriminative information, decreasing the classification performance. To avoid this information loss, this approach adds to the partially whitened data the residuals $\boldsymbol{R}$ of the points in $\boldsymbol{P}$ with respect to their projections on $\boldsymbol{\mathcal{SP}}_d$:

$$\boldsymbol{R} = \boldsymbol{P} - \boldsymbol{U}_d \boldsymbol{P}_{\perp \boldsymbol{\mathcal{SP}}_d} = \boldsymbol{P} - \boldsymbol{U}_d \boldsymbol{U}_d^T \boldsymbol{P} \tag{6}$$

$$\begin{aligned} \bar{\boldsymbol{P}}_{\boldsymbol{W}_D} &= \boldsymbol{U}_d \bar{\boldsymbol{P}}_{\boldsymbol{W}_d} + \boldsymbol{R} \\ &= \left( q_d \boldsymbol{U}_d \boldsymbol{Q}_d^{-1} \boldsymbol{U}_d^T + \boldsymbol{I} - \boldsymbol{U}_d \boldsymbol{U}_d^T \right) \boldsymbol{P} \\ &= \boldsymbol{W}_D \boldsymbol{P} \end{aligned} \tag{7}$$

where $\boldsymbol{W}_D \in \Re^{D \times D}$ represents the linear transformation that whitens the data along the first $d$ principal components, while keeping unaltered the information along the remaining ones.

Fs is estimated by exploiting the whitened class means, $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$, obtained by the class means estimated in the original space $\hat{\boldsymbol{\mu}}_A$ and $\hat{\boldsymbol{\mu}}_B$ as follows:

$$\begin{aligned} \boldsymbol{\mu}_A =& \boldsymbol{W}_D \hat{\boldsymbol{\mu}}_A \\ =& \left( q_d \boldsymbol{U}_d \boldsymbol{Q}_d^{-1} \boldsymbol{U}_d^T + \boldsymbol{I} - \boldsymbol{U}_d \boldsymbol{U}_d^T \right) \hat{\boldsymbol{\mu}}_A \\ =& q_d \boldsymbol{U}_d \boldsymbol{Q}_d^{-1} \boldsymbol{U}_d^T \hat{\boldsymbol{\mu}}_A + \hat{\boldsymbol{\mu}}_A - \boldsymbol{U}_d \boldsymbol{U}_d^T \hat{\boldsymbol{\mu}}_A \end{aligned} \tag{8}$$

The same calculation is done for $\boldsymbol{\mu}_B$. Using these quantities we estimate $\boldsymbol{f} = \frac{\boldsymbol{\mu}_A - \boldsymbol{\mu}_B}{\|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|}$. Then, we process an unknown point $\boldsymbol{p}$ by transforming it with $\boldsymbol{W}_D$, and projecting it on $\boldsymbol{f}$; both these steps are performed by the inner product $\boldsymbol{w} \cdot \boldsymbol{p}$, where:

$$\boldsymbol{w} = \boldsymbol{W}_D^T \boldsymbol{f} = q_d \boldsymbol{U}_d^T \boldsymbol{Q}_d^{-1} \boldsymbol{U}_d \boldsymbol{f} + \boldsymbol{f} - \boldsymbol{U}_d^T \boldsymbol{U}_d \boldsymbol{f} \tag{9}$$

Finally, given $\gamma$ as in Equation (4), $\boldsymbol{p}$ is assigned to class $A$ if $\boldsymbol{w} \cdot \boldsymbol{p} \geq \gamma$, to class $B$ otherwise.

### 3.1. One vs All Multi-Class Classification

Given an input dataset composed of points belonging to $G$ classes, and having selected the aforementioned binary classifying models (that is T-IPCAC) as the engine classifier, we train $G$ binary classifiers, each discriminating two classes in a *one-versus-all* fashion, and we combine their results. Precisely, the $G$ different binary classifiers are trained to distinguish the examples in a single class from the examples in all remaining classes. When it is desired to classify a new example, the $G$ trained classification models are run, thus obtaining a $G$-dimensional vector containing 0 (if the point belongs to the single class) or 1 (if the point belongs to the remaing classes). If we obtain a vector containing only a 0 in the g-th position of this vector (and the other elements have value 1), the point is assigned to the class $g$. Instead, if more elements of the vector have value 0 we select the single class with the small distance between the point taken into account and the computed thresholds (the $\gamma$s computed employing Equation (4)). Otherwise, if no vector elements have value 0 the distances between the single class means (computed at training time) and the point itself are computed, and the point is assigned to the class with the lower distance.

[1] We call "white" a dataset of points sampled from a probability distribution with $\boldsymbol{\mu} = \mathbf{0}$, and $\boldsymbol{\Sigma} = \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix.

[2] $d$ is a parameter to be set. Usually a good value is $d \simeq min(log_2^2 N, D)$

**Table 1**. Comparison with the state-of-the-art for Caltech101. In bold face the best results.

| | **15 Training** | **30 Training** |
|---|---|---|
| **Our method** ($d = 80$) | 72.52 | **80.48** |
| Grana et al. [4] | 71.62 | 79.68 |
| Grauman et al. [13] | 50.00 | 58.20 |
| Jia et al. [14] | - | 75.30 |
| Jiang et al. [15] | 67.50 | 75.30 |
| Liu et al. [16] | - | 74.21 |
| Tuytelaars et al. [17] | 69.20 | 75.20 |
| Wang et al. [3] | 65.43 | 73.40 |
| Yang et al. [18] | 67.00 | 73.20 |
| Chatfield et al. [2] | - | 77.78* |
| Duchenne et al. [19] | **75.30** | 80.30 |
| Huang et al. [20] | 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).

## 4. EXPERIMENTAL RESULTS

We have tested the performance of our approach on two well established datasets for object and scene classification: the Caltech 101 dataset and the Fifteen Scene Categories.

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. We have followed a common experimental setting: for training we have randomly selected 15 and 30 images respectively; for testing we have randomly selected at most 50 images for each category (this results in 3,060 images for training and 2,995 for testing).

The 15-scenes dataset consists of 4485 images spread over 15 indoor-outdoor catergories. Each category contains 200 to 400 images whose average size is $300 \times 250$ pixels. According to other papers in the literature, we train models on a randomly selected set of 1500 images (100 image per category) and test on the remaining images.

For both datasets, `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 [21] and, apart from the spacing step, the default options are used. Images are hierarchically partitioned into $1 \times 1$, $2 \times 2$ and $4 \times 4$ blocks on 3 levels respectively. We have reported 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.

Table 1 reports the results on Caltech 101 of several 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 method, based on Multivariate Gaussian Descriptor and `T-IPCAC`s, achieves performance definitely competitive with state-of-the art results. In particular `T-IPCAC`, since it is developed to deal where the dimensionality of the feature space is higher than the number of available training data, reduces the related problem of overfitting. Note that, differently from the common setting, the Fisher Kernel results reported in Chatfield et al. [2] 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. [19] and Huang et al. [20] 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. Nevertheless, also compared with these methods our approach obtains comparable or better results.

**Table 2**. Performance comparison with the state-of-the-art for 15-scene dataset. In bold face the best results.

| | **100 Training** |
|---|---|
| **Our method** ($d = 90$) | **85.71** |
| Grana et al. [4] | 84.77 |
| Niu et al. [23] | 82.50 |
| Liu et al. [16] | 82.70 |
| Tuytelaars et al. [17] | 85.00 |
| Yang et al. [18] | 80.28 |
| Boureau et al. [22] | 84.30 |

The performance comparison on 15-scene dataset is reported in Table 2. The proposed approach achieves better results than several related approaches, thus confirming the quality of our framework. Note that although our performance is only slightly higher than [17] and [22], our image signature representation does not require to build a codebook, that must be trained on every specific dataset, allowing to use the same feature vectors in different scenarios.

## 5. CONCLUSIONS AND FUTURE WORKS

`T-IPCAC` is a classifier developed to deal with high dimensional data without performing any dimensionality reduction step. The use of this technique combined with high dimensional multivariate Gaussian descriptors has shown impressive results in the field of image classification. This suggests that its application to large scale (high cardinality) datasets could even improve the current results. To achieve this goal we will extend current work by employing and improving the on-line version of `T-IPCAC` [6], thus allowing to tackle also large scale datasets together with the high dimensionality of the exploited image descriptor.

## 6. REFERENCES


[1] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE T. Pattern Anal.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[2] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011.

[3] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[4] Costantino Grana, Giuseppe Serra, Marco Manfredi, and Rita Cucchiara, "Image classification with multivariate gaussian descriptors," in *ICIAP*, 2013.

[5] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 30, pp. 1713–1726, 2000.

[6] A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli, "Novel fisher discriminant classifiers," *Pattern Recogn.*, vol. 45, no. 10, pp. 3725–3737, Oct. 2012.

[7] Oncel Tuzel, Fatih Porikli, and Peter Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE T. Pattern Anal.*, vol. 30, no. 10, pp. 1713–1727, 2008.

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

[9] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[10] A. Rozza, G. Lombardi, and E. Casiraghi, "Novel ipca-based classifiers and their application to spam filtering," *Proceedings of International Conference of Intelligent Systems Design and Applications (ISDA)*, pp. 797–802, 2009.

[11] S.C. Brubaker and S. Vempala, "Isotropic pca and affine-invariant clustering," *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pp. 551–560, 2008.

[12] Per C. Hansen, "The truncated SVD as a method for regularization," *BIT Numerical Mathematics*, vol. 27, no. 4, pp. 534–553, Dec. 1987.

[13] Kristen Grauman and Trevor Darrell, "The pyramid match kernel: Efficient learning with sets of features," *J. Mach. Learn. Res.*, vol. 8, pp. 725–760, 2007.

[14] Yangqing Jia, Chang Huang, and Trevor Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *CVPR*, 2012.

[15] Zhuolin Jiang, Guangxiao Zhang, and Larry S. Davis, "Submodular dictionary learning for sparse coding," in *CVPR*, 2012.

[16] Lingqiao Liu, Lei Wang, and Xinwang Liu, "In defense of soft-assignment coding," in *ICCV*, 2011.

[17] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, "The nbnn kernel," in *ICCV*, 2011.

[18] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[19] Olivier Duchenne, Armand Joulin, and Jean Ponce, "A graph-matching kernel for object categorization," in *Proc. of ICCV*, 2011.

[20] Yongzhen Huang, Kaiqi Huang, Chong Wang, and Tieniu Tan, "Exploring relations of visual codes for image classification," in *Proc. of CVPR*, 2011.

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

[22] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2559–2566.

[23] Zhenxing Niu, Gang Hua, Xinbo Gao, and Qi Tian, "Context aware topic model for scene recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2743–2750.