Towards Effective Codebookless Model for Image Classification

Qilong Wang[a], Peihua Li[a], Lei Zhang[b], Wangmeng Zuo[c]

[a]*School of Information and Communications Engineering, Dalian University of Technology, Dalian 116024, China*
[b]*Department of Computing, Hong Kong Polytechnic University, Hong Kong, China*
[c]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*

Abstract

The bag-of-features (BoF) model for image classification has been thoroughly studied over the last decade. Different from the widely used BoF methods which model images with a pre-trained codebook, the alternative codebook-free image modeling method, which we call Codebookless Model (CLM), attracts little attention. In this paper, we present an effective CLM that represents an image with a single Gaussian for classification. By embedding Gaussian manifold into a vector space, we show that the simple incorporation of our CLM into a linear classifier achieves very competitive accuracy compared with state-of-the-art BoF methods (e.g., Fisher Vector). Since our CLM lies in a high-dimensional Riemannian manifold, we further propose a joint learning method of low-rank transformation with support vector machine (SVM) classifier on the Gaussian manifold, in order to reduce computational and storage cost. To study and alleviate the side effect of background clutter on our CLM, we also present a simple yet effective partial background removal method based on saliency detection. Experiments are extensively conducted on eight widely used databases to demonstrate the effectiveness and efficiency of our CLM method.

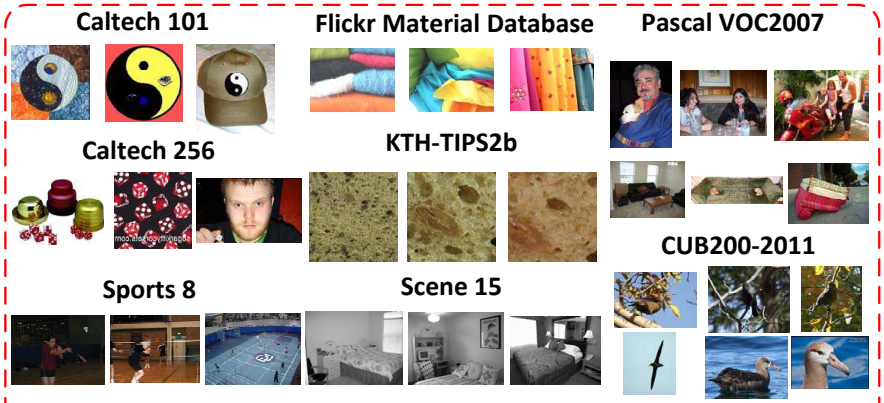
Keywords: Codebookless Model, image classification, bag-of-features, Riemannian manifold

Email address: peihuali@dlut.edu.cn (Peihua Li)

1. Introduction

Image classification has been attracting massive attentions in computer vision and pattern recognition communities in recent years. It is one of the most fundamental but challenging vision problems because images, as illustrated in Fig. 1, often suffer from significant scale, view or illumination variations (e.g., in texture classification [8] and material recognition [23]), and pose changes, background clutter, partial occlusion (e.g., in scene categorization [31, 32] and object recognition [17, 18, 22, 52]).

For a long time the bag-of-features (BoF) model [46] has been almost given priority to image classification. As shown in Fig. 2 (a), the BoF-based methods generally consist of five components: local features extraction, learning codebook with training data, coding local features with pre-trained codebook, pooling or aggregating codes over images, and finally, learning classifier (e.g., SVM) for classification. With this processing pipeline, the BoF-based methods can be seen as a hand-crafted five-layer hierarchical feed-forward network [49] with a pre-trained feature coding template (codebook) [7]. The learned codebook depicts the distribution of feature space, and makes coding of high dimensional features possible. This architecture has achieved very promising performance in a variety of image classification tasks.



	Caltech 101	Caltech 256	KTH-TIPS2b	FMD	VOC2007	CUB200-2011	Scene 15	Sports 8
FV	83.7	50.1	71.3	58.9	61.8	27.3	89.4	91.3
Our CLM	86.3	53.6	75.2	64.2	61.7	36.0	89.4	91.5

Figure 1: Some example images and accuracy comparison (in %) between Fisher vector (FV) and our codebookless model (CLM) on various image databases.

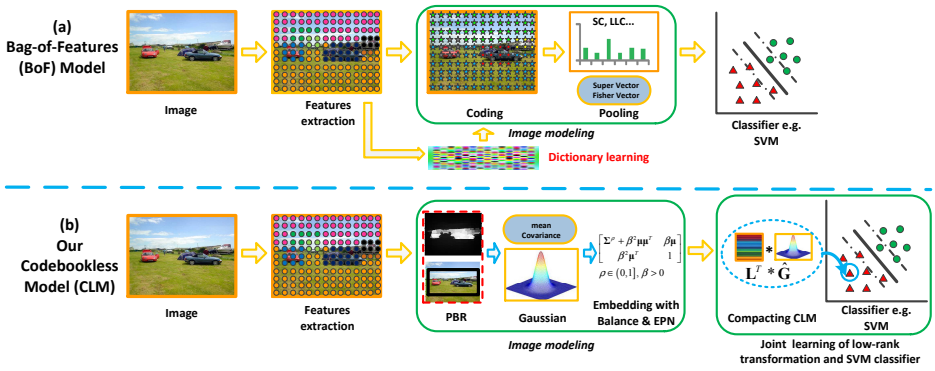


Figure 2: Comparison between (a) the BoF model and (b) our CLM. The major difference between them is that whether there is a pre-trained codebook & coding or not. Our CLM mainly consists of a Gaussian model for image representation and a joint low-rank learning with linear SVM classifier.

The codebook as a reference for feature coding serves as a bridge between local features and global image representation. However, it is well known that segmentation of feature space involved in building of codebook brings on quantization error [6], and leads to continuous striving for this side effect (e.g., soft coding methods [44, 19, 55] alleviate but cannot completely eliminate it). Though offline, training of codebook, particularly large size ones, is time consuming. In addition, in general the pre-trained codebook on one database cannot naturally adapt to other databases [58].

An alternative approach is to estimate the statistics directly on sets of local features from input images [10, 38, 50], as illustrated in Fig. 2 (b), which is called codebookless model (CLM) in this paper. It is clear from Fig. 2 that the major difference is that the BoF model learns a codebook to explore the statistical distribution of local features and then performs coding of descriptors, while the CLM represents images with descriptors directly, requiring no pre-trained codebook and the subsequent coding. Conceptually, the codebookless model has the potential to circumvent the aforementioned limitations of the BoF model, however, which has received little attention in image classification community. The main reasons may be that such methods have not yet shown competitive classification performance, and that they often need to utilize inefficient and unscalable kernel-based classifiers.

In this paper, we propose an effective CLM scheme, and argue that the CLM can be a competitive alternative to the BoF methods for image classification. The

comparison between state-of-the-art BoF method, Fisher Vector (FV) [44], and our CLM on various image databases is shown in Fig. 1. First and foremost, we extract a set of local features (e.g., SIFT [37]) on a dense grid of image, and simply model them with a single Gaussian model to represent the input image. Then, we employ a two-step metric for matching Gaussian models. By using this metric, Gaussian models can be fed to a linear classifier for ensuring efficient and scalable classification while respecting the Riemannian geometry structure of Gaussian models. Moreover, we introduce two well-motivated parameters into the used metric. One is to balance the effect between mean and covariance of Gaussian, and another is for eigenvalue power normalization on covariance.

Our codebookless model usually is of high dimension, by incorporating low-rank learning with SVM, we propose a joint learning method to effectively compress Gaussian models while respecting their Riemannian geometry structure. It is mentionable that, to the best of our knowledge, we make the first attempt to perform joint learning of low-rank transformation and SVM on Gaussian manifold. Finally, to alleviate the side effect of background clutter, a saliency-based partial background removal method is proposed to enhance our CLM. The experimental results show that partial background removal is helpful to CLM when images are heavily cluttered (e.g., CUB200-2011 and Pascal VOC2007).

2. Related work

The codebookless model for directly modeling the statistics of local features has been studied in past decades. Rubner et al [43] introduced signatures for image representation, and proposed the Earth Mover's Distance for image matching which is robust but has high computational cost. Tuzel et al [50] for the first time used covariance matrices for representing regular image regions, and employed Affine-Riemannian metric which suffers from high computational cost [40]. Gaussian model as image descriptor has been used for visual tracking [20], in which Gaussian models are matched based on the Riemannian metric, involving expensive operations to solve generalized eigenvalue problem. Going beyond Gaussian, Gaussian mixture model (GMM) is more informative and is used in image classification and retrieval [3, 41]. However, GMM suffers from some limitations, such as high computational cost of matching methods and lacking of general criteria for model selection.

Our work is motivated by [9, 10] and [38]. Carreira et al [9, 10] modeled the free-form regions obtained by image segmentation with estimating the second-order moments. By using Log-Euclidean metric [2], the method in [9, 10] can be

combined with a linear classifier, which has shown competing recognition performance on images with less background clutter (e.g., Caltech101 [18]). Different from [9, 10], we employ a Gaussian model to represent the whole image. It is well-known that a covariance matrix can be seen as a Gaussian model with fixed mean vector. Compared to [9, 10], our CLM contains both the first-order (mean) and second-order (covariance) information. Note that the first-order statistics has proved to be important in image classification [26, 44]. Moreover, the manifold of Gaussian models and that of covariance matrices are quite different, and the embedding method in our CLM makes Gaussian models can be handled flexibly and conveniently.

Nakayama et al [38] also represented an image with a global Gaussian for scene categorization. However, they matched two Gaussian models by using the Kullback-Leibler (KL) divergence, and hence kernel-based classifiers have to be used. This method is not scalable and has high computational cost. In contrast to [38], our metric is decoupled which allows a linear classifier to be combined, which makes our method more efficient and scalable than the KL kernel based one in [38]. Moreover, compared with the ad-hoc linear kernel (Euclidean baseline) in [38], our method takes advantage of the geometry structure of Gaussian models and brings large performance improvement.

There is another line of research on codebookless model methods. Grauman et al [21] proposed a pyramid match kernel to map feature sets to multi-resolution histograms, and employed histogram intersection kernel for classification. Bo et al [5] presented efficient match kernels to map local features into a low dimensional space, and adopted a linear classifier. Boiman et al [6] developed an image-to-class distance between the sets of local features, and employed a nearest neighbor classifier. Yao et al [56] proposed a codebook-free approach by using a large number of randomly generated image templates for image representation, and developed a bagging-based classifier. Peng et al. [39] studied image representation from the discriminative-generative viewpoint, and suggested a deep boosting architecture for joint filter learning and feature selection in a layer-by-layer manner. Lin et al. [35] proposed to use a set of random image patches to represent an input image in object detection task and achieved promising results, where they decided locations of objects by employing location sensitive matching between image patch and reference patches (ε-balls). Those ε-balls are learned from a massive of training image patches with maximal information gain strategy. Different from the above methods, we exploit a single Gaussian model to represent image for classification.

3. Proposed method

We first introduce the image representation by a single Gaussian model. Then, we employ an effective and efficient two-step metric for matching Gaussian models, and propose two well-motivated parameters to improve the used distance metric. Finally, we present a joint learning method of low-rank transformation and SVM on Gaussian manifold.

3.1. Gaussian model for image representation

Given an input image, we extract a set of N local features $\{\mathbf{x}_i \in \mathbb{R}^{k\times 1}, i = 1,\ldots,N\}$ at a dense grid. By the maximum likelihood method, the image can be represented by the following Gaussian model:

$$\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{\exp\big(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})\big)}{\sqrt{(2\pi)^k \det(\boldsymbol{\Sigma})}},$$

where $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ and $\boldsymbol{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^T$ are mean vector and covariance matrix, and $\det(\cdot)$ denotes matrix determinant. Compared with histogram and covariance, Gaussian model is more informative. Meanwhile, unlike matching of signatures [43] or GMMs [3], matching of Gaussian models does not bring high computational cost.

3.2. Two-step metric between Gaussian models

To match Gaussian models, we exploit a two-step metric which has been proposed to compute the ground distance between Gaussian components of GMMs [34]. The first step is to embed Gaussian manifold into the space of SPD matrices [36], and then map the Lie group of SPD matrices into its corresponding Lie algebra, a linear space, by using the Log-Euclidean metric [2].

The space of k-dimensional Gaussian models is a Riemannian manifold. Let $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ be a Gaussian model with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Through a continuous function π, $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ is mapped to an affine matrix, an element in the affine group $\mathcal{A}_k^+ = \{(\boldsymbol{\mu},\mathbf{P})|\boldsymbol{\mu}\in\mathbb{R}^{k\times 1},\mathbf{P}\in\mathbb{R}^{k\times k},\det(\mathbf{P})>0\}$; that is,

$$\pi : \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \mapsto \mathbf{A} = \begin{bmatrix}\mathbf{P} & \boldsymbol{\mu}\\ \mathbf{0}^T & 1\end{bmatrix}, \tag{1}$$

where $\boldsymbol{\Sigma} = \mathbf{P}\mathbf{P}^T$ is the Cholesky factorization of $\boldsymbol{\Sigma}$. Further, through the function $\gamma : \mathbf{A} \mapsto \mathbf{S} = \mathbf{A}\mathbf{A}^T$, $\mathbf{A}$ is mapped to an SPD matrix $\mathbf{S}$. So far, by the successive

functions π and γ, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is uniquely designated as an $(k+1) \times (k+1)$ SPD matrix

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathbf{S} = \begin{bmatrix} \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}. \tag{2}$$

Please refer to [36] for details on the embedding process.

The space of $(k+1) \times (k+1)$ SPD matrices $\mathcal{S}^+_{k+1}$ is a Lie group that forms a Riemannian manifold. Two operations, namely the logarithmic multiplication and the scalar logarithmic multiplication, are defined in the Log-Euclidean metric [2], which equip $\mathcal{S}^+_{k+1}$ with structures of not only the Lie group but also vector space. Through the matrix logarithm, $\mathcal{S}^+_{k+1}$ is mapped into its Lie algebra $\mathcal{S}_{k+1}$, the vector space of $(k+1) \times (k+1)$ symmetric matrices. The matrix logarithm is a deffemorphism and an isomorphism so that operations over SPD matrices can be replaced by the Euclidean operations of their counterparts in the vector space. So, through the matrix logarithm, an SPD matrix $\mathbf{S}$ is one-to-one mapped to a symmetric matrices $\mathbf{G}$ which lies in a linear space, and the geodesic distance between SPD matrices $\mathbf{S}_i$ and $\mathbf{S}_j$ is defined by $dist_{\mathbf{S}_i,\mathbf{S}_j} = \|\mathbf{G}_i - \mathbf{G}_j\|_F$, where F is the Frobenius norm.

3.3. Two well-motivated parameters

In practice, we found that it is important to balance mean vector and covariance matrix in the embedding matrix (2), because their dimensions and order of magnitude of each dimension may vary considerably. Meanwhile, the effect of mean vector and covariance matrix may vary for different tasks. With these considerations, we introduce a parameter $\beta > 0$ in the function π (1):

$$\pi(\beta):\ \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \mapsto \mathbf{A} = \begin{bmatrix} \mathbf{P} & \beta\boldsymbol{\mu} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{3}$$

Accordingly, the embedding matrix has the following form:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathbf{S}(\beta) = \begin{bmatrix} \boldsymbol{\Sigma} + \beta^2\boldsymbol{\mu}\boldsymbol{\mu}^T & \beta\boldsymbol{\mu} \\ \beta\boldsymbol{\mu}^T & 1 \end{bmatrix}. \tag{4}$$

The embedding matrix (4) reduces to the covariance matrix when $\beta = 0$, and is equal to the original one when $\beta = 1$. Hence, the role of mean vector and covariance matrix can be adjusted by β.

The maximum likelihood estimator of the empirical covariance matrix is susceptible to interference of noise, especially for high dimension space [15]. Based

on observation that the maximum likelihood estimator of covariance ought to be improvable by eigenvalue shrinkage [48], we exploit power normalization on the eigenvalues of covariance matrix (EPN). Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian model estimated from a set of descriptors extracted from some image. The covariance matrix $\boldsymbol{\Sigma}$ has eigenvalue decomposition $\boldsymbol{\Sigma} = \mathbf{U}\mathrm{diag}(\lambda_i)\mathbf{U}^T$, where $\mathbf{U}$ is an orthornormal matrix whose i^{th} column is the eigenvector of $\boldsymbol{\Sigma}$ and $\lambda_i > 0$ is the corresponding eigenvalue, and $\mathrm{diag}(\cdot)$ denotes diagonal matrix. Then by introducing a parameter ρ, our normalization is defined as

$$\boldsymbol{\Sigma}^{\rho} = \mathbf{U}\mathrm{diag}(\lambda_i^{\rho})\mathbf{U}^T, \text{ with } 0 < \rho \leq 1. \tag{5}$$

With EPN, our final embedding matrix is:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathbf{S}(\beta, \rho) = \begin{bmatrix} \boldsymbol{\Sigma}^{\rho} + \beta^2 \boldsymbol{\mu}\boldsymbol{\mu}^T & \beta\boldsymbol{\mu} \\ \beta\boldsymbol{\mu}^T & 1 \end{bmatrix}. \tag{6}$$

It is easy to prove that the embedding matrix (6) is still positive definite as $\boldsymbol{\Sigma}^{\rho}$ being an SPD matrix. The eigenvalues power normalization has been proposed to measure distances between covariance matrices [16, 25] or tensor [30], namely, Power-Euclidean metric. Different from previous work, we use eigenvalues power normalization for robust estimation of covariance matrices in Gaussian setting for the case of high dimensional features, and compare Gaussians by using Gaussian embedding and the Log-Euclidean metric.

According to the Log-Euclidean framework, the matrix $\mathbf{S}(\beta, \rho)$ can be further embedded into a linear space by matrix logarithm:

$$\mathbf{G}(\beta, \rho) = \log(\mathbf{S}(\beta, \rho)). \tag{7}$$

Let $\mathcal{N}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\mathcal{N}_j = \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ be two Gaussian models and their corresponding symmetric matrices are $\mathbf{G}_i(\beta, \rho)$ and $\mathbf{G}_j(\beta, \rho)$. The distance between two Gaussian models is

$$dist_{\mathcal{N}_i, \mathcal{N}_j} = \left\| \mathbf{G}_i(\beta, \rho) - \mathbf{G}_j(\beta, \rho) \right\|_F. \tag{8}$$

It is easy to know that distance (8) is decoupled so that $\mathbf{G}_i(\beta, \rho)$ and $\mathbf{G}_j(\beta, \rho)$ can be computed separately and adopted in a linear classifier. For notational simplicity, we omit the parameters β and ρ in the distance measure (8).

3.4. Joint low-rank learning and SVM classifier

Our CLM usually is of high dimension ($> 10^4$). In order to suppress redundant and noisy information while reducing computational and storage cost, we propose a low-rank learning method to compact our CLM. The matrix $\mathbf{G}$ in geodesic distance (8) is a $(k+1)\times(k+1)$ symmetric matrix which lies in the Euclidean space. Due to its symmetry, we can unfold the upper triangular part of $\mathbf{G}$ to a vector of size $d=(k+1)\times(k+2)/2$. We can modify geodesic distance (8) by introducing a low-rank transformation matrix $\mathbf{L}\in\mathbb{R}^{d\times r}, r\ll d$:

$$dist_{\mathcal{N}_i,\mathcal{N}_j} = \|\mathbf{L}^T(\mathbf{f}_i-\mathbf{f}_j)\|_2, \tag{9}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the unfolding vectors of two Gaussian models $\mathcal{N}_i$ and $\mathcal{N}_j$, respectively.

Recent studies [27, 54] have shown that joint optimization of dimensionality reduction with classifier performs better than separate optimization of the two modules. Thus, given N training samples $\{\mathbf{f}_n, n\in[1,N]\}$, we optimize the low-rank learning jointly with a linear SVM (LRSVM):

$$\begin{aligned}
&\min_{\mathbf{L},\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\
&s.t. \quad y_n(\mathbf{w}^T\mathbf{L}^T\mathbf{f}_n + b) \geq 1-\xi_n, \forall\xi_n>0, n\in[1,N], \\
&\qquad \mathbf{L}^T\mathbf{L} = \mathbf{I},
\end{aligned} \tag{10}$$

where $\mathbf{w}, \boldsymbol{\xi}, b$ are parameters of SVM, and y_n is the label of $\mathbf{f}_n$. The dimensionality reduction for SPD matrices [24] has been studied with dimensionality reduction and classification separately performed, while our method is quite different in that we focus on Gaussian models and perform joint learning of low-rank transformation and SVM.

In practice, we extend the objective function (10) to multi-class problem under the spatial pyramid matching (SPM) framework [31]. Given an image I_n, we can obtain its SPM representation $\mathbf{F}_n = [(\mathbf{f}_n^1)^T,\ldots,(\mathbf{f}_n^B)^T]^T$, where B is the number of blocks in SPM, which is fed to a one vs. all SVM for solving the M classes problem. As suggested in [27], we optimize the dual problem of the objective

function (10) under the SPM framework:

$$\begin{aligned}
&\min_{\widehat{\mathbf{L}}} \max_{\boldsymbol{\alpha}_m} \sum_{m=1}^{M} \Big(\sum_{n=1}^{N} \alpha_m^n - \frac{1}{2} (\boldsymbol{\alpha}_m^T \mathbf{Y}_m \mathbf{F} \mathbf{H} \mathbf{F}^T \mathbf{Y}_m \boldsymbol{\alpha}_m) \Big) \\
&s.t. \quad \sum_{n=1}^{N} y_m^n \alpha_m^n = 0, 0 \leq \boldsymbol{\alpha}_m \leq C, \forall m \\
&\qquad \widehat{\mathbf{L}}^T \widehat{\mathbf{L}} = \mathbf{I},\ \widehat{\mathbf{L}}^T = \mathrm{Diag}(\mathbf{L}_1^T, \ldots, \mathbf{L}_B^T),\ \mathbf{H} = \widehat{\mathbf{L}} \widehat{\mathbf{L}}^T,
\end{aligned} \tag{11}$$

where $\mathbf{F} = [\mathbf{F}_1, \ldots, \mathbf{F}_N]^T$ indicates all training features, and $\mathbf{Y}_m$ is the diagonal label matrix of the mth class with diagonal element $\mathbf{Y}_m(n,n) = y_m^n$.

The problem (11) is non-convex and can be optimized by a two-step alternating method: *Step One*, fixing $\widehat{\mathbf{L}}$, we can optimize the Lagrange parameters $\boldsymbol{\alpha}_m$ with off-the-shelf SVM; *Step Two*, for fixed $\boldsymbol{\alpha}_m$, we solve the following trace maximization problem:

$$\begin{aligned}
&\max_{\widehat{\mathbf{L}}} \mathrm{tr} \left(\widehat{\mathbf{L}}^T \mathbf{F}^T \sum_{m=1}^{M} (\mathbf{Y}_m \boldsymbol{\alpha}_m \boldsymbol{\alpha}_m^T \mathbf{Y}_m^T) \mathbf{F} \widehat{\mathbf{L}} \right) \\
&s.t. \quad \widehat{\mathbf{L}}^T \widehat{\mathbf{L}} = \mathbf{I},\ \widehat{\mathbf{L}}^T = \mathrm{Diag}(\mathbf{L}_1^T, \ldots, \mathbf{L}_B^T).
\end{aligned} \tag{12}$$

We optimize the problem (12) by independently solving each $\mathbf{L}_i^T,\ i = 1, \ldots, B$ with a closed-form solution [27] which is corresponding to the eigenvectors of matrix $(\mathbf{F}^i)^T \sum_{m=1}^{M} (\mathbf{Y}_m \boldsymbol{\alpha}_m \boldsymbol{\alpha}_m^T \mathbf{Y}_m^T)(\mathbf{F}^i)$ with larger eigenvalues, and $\mathbf{L}_i^T \mathbf{L}_i = \mathbf{I}$ is naturally satisfied. Due to the problem (11) being non-convex, initialization is nontrivial to reach a good local optimal solution and for fast convergence. In this paper, we use the basis of principal component analysis (PCA) as initialization, and we find that it can always achieve good performance and fast convergence.

4. Partial background removal (PBR)

We then present a simple yet effective method for analyzing and handling the side effect of background clutter based on unsupervised, bottom-to-up saliency detection. Our purpose here is to remove the interference of background, which is distinguished from the purpose of precise foreground localization in saliency detection community. Our method consists of two steps: coarse foreground detection and partial background removal. In the first step we localize in image the foreground based on saliency detection [28] and then determine the bounding-box

surrounding the foreground. Next, we adaptively expand bounding-box to accommodate some background regions based on size and intensity variance of the area inside the bounding-box. Then, the area outside bounding-box is removed for recognition. Our method is based on the considerations that accurate foreground detection is currently very difficult and neighboring regions of object can serve as the context and may be helpful for recognition. In our experiments, we adopt PBR to the two datasets with heavy background clutter: CUB200-2011 and VOC2007. Since PBR is designed for foreground objects with separable background clutter, we do not perform PBR on images with less background clutter and scene images where both foreground and background are valuable for scene understanding.

5. Implementation details

We extract multi-scale SIFT descriptors [37] (standard pipeline in the BoF model) with cell size 2^i, $i = 1, 2, \ldots$, and single scale pixel-wise covariance descriptor [33] via the dense sampling strategy with step-length 2. The dense covariance descriptors are computed with 17 dimensional raw features including intensity and four kinds of first-order and second-order gradients from [42]. We perform matrix logarithm on the covariance descriptors (LogCov), which are then vectorized. The SIFT features are calculated via the VLFeat library [51]. Moreover, following [9, 10], we also extract additional image cues, including color, location, scale, gradient and entropy to concatenate SIFT and LogCov. In order to ensure that there is sufficient data to estimate Gaussian models and covariance matrices are positive definite, we limit the minimum size of width or height of images to be larger than 64, and add 10^{-3} to the diagonal entries of covariance matrices, respectively. We employ the spatial pyramid strategy [31] which divides an image into some regular regions (e.g., 1×1, 2×2, 1×3, 4×4). For each region we compute a Gaussian model, and then concatenate them to represent the whole image. Each Gaussian is weighted by $\frac{1/N_l}{\sum_{l=1}^{L} 1/N_l}$, where L and N_l are the number of pyramid levels and regions in the l^{th} layer, respectively. We implement a one-vs-all SVM with LibSVM [11] and set parameter C to 0.01 on VOC2007 and 10 on all the other databases. All algorithms are written in Matlab, and run on a PC equipped with i7-4770k CPU and 32G RAM.

6. Experimental evaluation

In this section, we evaluate the classification performance of our CLM on eight benchmark databases. First of all, we make an analysis of local features,

	Local descriptors				Parameters		BR		
	ST	eST	LC	eLC	Beta	EPN	PBR	GT	Acc.
Cov.	✓								16.8
		✓							24.1
Gau.	✓								18.6
		✓							25.6
			✓						19.1
				✓					26.3
		✓			✓				26.5
		✓				✓			26.8
		✓					✓		33.3
		✓						✓	45.3
		✓			✓	✓			28.1
		✓			✓	✓	✓		36.0
		✓			✓	✓		✓	48.2

Table 1: Classification results (in %) of our CLM vs. various combinations of descriptors, parameters and background removal on CUB200-2011.

the parameters of our method, the proposed low-rank learning method and the partial background removal method on the challenging CUB200-2011 [52]. Then, we compare with state-of-the-art methods on Caltech101 [18], Caltech256 [22], KTH-TIPS2b [8], Flickr Material Database (FMD) [23], Pascal VOC2007 [17], Scene15 [31] and Sports8 [32]. Finally, we analyze the computational complexity of our CLM.

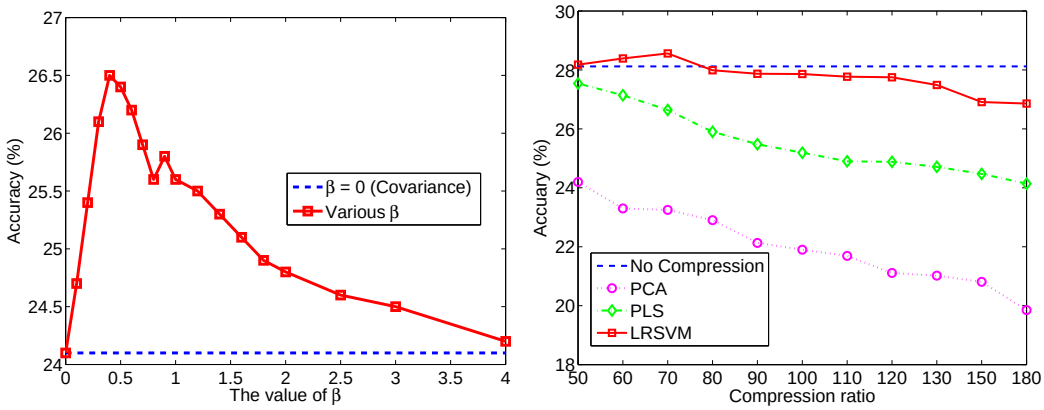


Figure 3: Effect of balance parameter β in Eq. (4) (left) and comparison of PCA, PLS and our LRSVM with various compression ratios on CUB200-2011 (right).

6.1. Parameters analysis

Local descriptors Four kinds of local descriptors, SIFT (ST) and its enrichment (eST), and LogCov (LC) and its enrichment (eLC), are evaluated in this section. The results of our CLM with various local descriptors on CUB200-2011 are shown in Table 1. We can see that the Gaussian model used in our method outperforms covariance matrix by 1.5% or higher with either SIFT or eSIFT, which, we believe, can indicate that the first-order (mean) information is non-trivial. We use eST to evaluate other parameters as follows.

Two well-motivated parameters The proposed EPN (5) is a generic method for robust estimation of covariance in high dimension space. We set parameter ρ in EPN (5) as 0.5 in all databases. From Table 1, we can see that EPN can bring 1.2% performance gain over the relevant method without EPN. The embedding parameter β (6) balances the effect of mean vector and covariance matrix. To test its effect, we determine the optimal value of β via cross validation. The performances of our CLM with various β are illustrated in Fig. 3 (left). Compared to $\beta = 0$ (covariance matrix only [9, 10]) and $\beta = 1$ (the embedding in [36]), appropriate balancing at $\beta = 0.4$ achieves 2.4% and 0.9% gains, respectively. The parameter β can balance the effect of covariance matrix and mean vector on classification, and the bigger β indicates mean vector has greater impact on classification. On CUB200-2011, the performance will continuously decrease with bigger value of β when it is bigger than 1, which indicates the effect of covariance matrix should be greater on CUB200-2011. The parameter β varies in different tasks or databases, which can be decided by cross validation. It is set to 0.4 on KTH-TIPS2b and CUB200-2011, 0.6 on Scene15 and FMD, 0.8 on Sports8 and Caltech101, 1.5 on Pascal VOC 2007 and Caltech 256, respectively.

LRSVM To evaluate the proposed LRSVM method, we compare LRSVM with unsupervised principal component analysis (PCA) and supervised partial least square (PLS) [1] under different compression ratios. The LRSVM is initialized by PCA, and the results on CUB200-2011 are illustrated in Fig. 3 (right). From it we can see that LRSVM always performs better than PLS, and is superior to PCA by a large margin. Different from PLS which exploits the least squares loss, LRSVM uses the hinge loss. We argue that the improvement owes to the joint learning of dimensionality reduction and classifier. Note that, with larger compression ratio, LRSVM achieves larger improvement over PCA and PLS. Meanwhile, the proposed LRSVM has insignificant performance loss (less than 1.5%) with large compression ratio (> 100). We also can see that LRSVM can slightly improve the performance of our CLM when compression rations are smaller (< 80), which

(a)

Database	Classes	Images in total	Training/Test	Measurement
CUB200-2011 [52]	200	11,788	Split in [52]	Acc. of split
Caltech101 [18]	102	9,144	30/remaining per class	Acc. of 5 runs
Caltech256 [22]	256	30,607	30/remaining per class	Acc. of 5 runs
Sports8 [32]	8	1,792	70/60 per class	Acc. of 5 runs
KTH-TIPS2b [8]	11	4,752	[13]	Acc. of splits
FMD [23]	10	1,000	50/50 per class	Acc. of 5 runs
VOC2007 [17]	20	9,963	Split in [17]	mAP of split
Scene15 [31]	15	4,485	100/remaining per class	Acc. of 5 runs

(b)

Database	Scale	View	Illumination	Pose	Bg Clutter	Occlusion
CUB200-2011 [52]	✓	✓	✓	✓	✓	✓
Caltech101 [18]	✓			✓		
Caltech256 [22]	✓			✓		✓
Sports8 [32]	✓	✓	✓	✓		
KTH-TIPS2b [8]	✓	✓	✓			
FMD [23]	✓	✓	✓			
VOC2007 [17]	✓	✓	✓	✓	✓	✓
Scene15 [31]	✓	✓		✓		

Table 2: Descriptions and experimental setup on eight widely used benchmarks.

we owe to that LRSVM can suppress some noisy information. In general, we set compression ratio as $80 \sim 100$ to balance the efficiency and effectiveness.

Impact of PBR We apply PBR to CUB200-2011 and the results are presented in Table 1. We can see that the method using PBR achieves great gains (more than 7.5%) over the one without PBR. Note that we achieve about 1% gain in VOC2007 by using PBR. It shows that our PBR is a general method to handle background for CLM. The gains achieved by using ground truth (GT) bounding box indicate more advanced background removal methods have further ability to improve the recognition performance of our CLM. Compared with the improvement in CUB200-2011, the gains in VOC2007 are relative small. The reasons are mainly that the saliency-based methods fail to locate precisely the foregrounds in the challenging databases, and CUB200-2011 only contains one object per image while one image may contain multiple objects in VOC2007. PBR can not segment image into multiple objects so that multi-object images will heavily influence the performance of CLM.

6.2. Comparison with state-of-the-art methods

We compare our CLM with more than ten state-of-the-art methods on eight widely used benchmarks. The descriptions and experimental setup on these bench-

(a) CUB200-2011

Methods	Acc.
BoF-hard [31]	18.6
FV [44]	25.8
FV + eSIFT	27.3
FV + eSIFT + PBR	33.2
Kobayashi2014 [29]	27.3
PPK [57]	28.2
CLM (SIFT)	18.6
CLM (eSIFT)	28.1
CLM (LogCov)	19.1
CLM (eLogCov)	28.6
CLM (eSIFT) + PBR	36.0

(b) Caltech101

Methods	Acc. (Tr. = 30)
FV+SIFT [44]	80.8 ± 0.3
FV+eSIFT	83.7 ± 0.3
DeCAF [14]	86.9 ± 0.7
O2P+eSIFT [10]	80.8
SQ-O2P+SIFT [7]	79.5
M-GOLD [45]	81.0
CLM (SIFT)	84.9 ± 0.1
CLM (eSIFT)	86.3 ± 0.3
CLM (LogCov)	82.5 ± 0.3
CLM (eLogCov)	84.7 ± 0.2

(c) Caltech256

Methods	Acc. (Tr. = 30)
FV+SIFT [44]	47.4 ± 0.1
FV+eSIFT	50.1 ± 0.3
Kobayashi2014 [29]	49.8 ± 0.1
NBNN [6]	43
M-GOLD [45]	44.2
M-HMP [6]	50.7
CLM (SIFT)	48.9 ± 0.2
CLM (eSIFT)	53.6 ± 0.2
CLM (LogCov)	48.6 ± 0.3
CLM (eLogCov)	53.2 ± 0.1

(d) Sports8

Methods	Acc.
FV+SIFT [44]	91.3 ± 1.3
FV+eSIFT	90.4 ± 1.2
Kobayashi2014 [29]	92.6 ± 0.7
GG (ad-linear) [38]	80.2
GG (ct-linear) [38]	82.9 ± 1.0
GG + KL Div. [38]	84.4 ± 1.4
CLM (SIFT)	88.8 ± 1.0
CLM (eSIFT)	91.5 ± 1.2
CLM (LogCov)	88.3 ± 1.3
CLM (eLogCov)	90.7 ± 0.7

(e) KTH-TIPS2b

Methods	Acc.
BoF-LLC [53]	57.6 ± 2.3
VLAD [26]	63.1 ± 1.0
FV+SIFT [44]	69.3 ± 1.0
FV+eSIFT	71.3 ± 3.1
DeCAF [14]	70.7 ± 1.7
Attributes [13]	73.8 ± 1.3
CLM (SIFT)	71.8 ± 3.1
CLM (eSIFT)	75.2 ± 2.6
CLM (LogCov)	72.2 ± 3.3
CLM (eLogCov)	73.6 ± 2.6

(f) FMD

Methods	Acc.
VLAD [26]	52.6 ± 1.5
FV+SIFT [44]	58.3 ± 1.0
FV+eSIFT	58.9 ± 1.7
Kobayashi2014 [29]	57.3 ± 0.9
DeCAF [14]	60.7 ± 2.1
Attributes [13]	61.1 ± 1.4
CLM (SIFT)	51.6 ± 1.2
CLM (eSIFT)	57.7 ± 1.6
CLM (LogCov)	62.4 ± 1.5
CLM (eLogCov)	64.2 ± 1.0

(g) VOC2007

Methods	mAP.
BoF-LLC [53]	57.4
SV [59]	58.2
SQ-O2P+SIFT [7]	51.0
M-GOLD [45]	61.1
FV+SIFT [44]	61.8
FV+eSIFT	60.8
CLM (SIFT)	55.8
CLM (eSIFT)	60.4
CLM (LogCov)	56.6
CLM (eLogCov)	61.7

(h) Scene15

Methods	Acc.
SV [59]	85.0
FV+SIFT [44]	88.1 ± 0.2
FV+eSIFT	89.4 ± 0.2
GG (ad-linear) [38]	79.8
GG (ct-linear) [38]	82.3 ± 0.4
GG + KL Div. [38]	86.1 ± 0.5
CLM (SIFT)	88.1 ± 0.4
CLM (eSIFT)	89.4 ± 0.4
CLM (LogCov)	88.3 ± 0.6
CLM (eLogCov)	89.2 ± 0.5

Table 3: Comparison (in %) with state-of-the-art methods on eight widely used benchmark datasets

marks are listed in Table 2. We report the results in Table 3, and discuss the experimental results as follows.

Comparison of various local descriptors We combine our CLM with four kinds of local descriptors, and assess them on all databases. From Table 3 we can see that SIFT and LogCov achieve comparable results. For object recognition, LogCov is superior to SIFT on CUB200-2011 and VOC2007 while SIFT outperforms LogCov on Caltech101 and Caltech256. On scene categorization, SIFT and LogCov obtain similar performances on both Sports8 and Sence15. For texture and material classification, SIFT achieves gains over LogCov on KTH-TIPS2b while LogCov is superior to SIFT by a large margin on FMD. The eSIFT and eLogCov perform with the similar rule as SIFT and LogCov, respectively. The enrichment on SIFT and LogCov can considerably boost the performance of our CLM, which encourages us to utilize more informative descriptors for further improvement.

Comparison with counterparts Here, we compare our CLM with its counterparts, O2P [10], Global Gaussian (GG) [38], mixture of GOLD (M-GOLD) [45] and NBNN [6]. As shown in Tables 1 & 3, our CLM significantly outperforms O2P [10] on CUB200-2011 and Caltech101, and is also superior to its variant with sparse quantization (SQ-O2P) [7] on Caltech101 and VOC2007 by a large margin, which are mainly due to the appropriate use of mean information and EPN. Moreover, our CLM performs much better than GG methods [38] with ad-hoc linear kernel (ad-linear), center tangent linear kernel (ct-linear) and KL divergence on Sports8 and Sence15. The ad-linear can be seen as a baseline in Euclidean space. It is mentionable that the methods in [38] exploit probabilistic discriminant analysis (PDA) as a classifier. If SVM is used, their results will drop to 71.7%, 78.8% and 81.4% on Sports8, and 74.3%, 80.7% and 83.1% on Scene15, respectively. In addition, our CLM outperforms mixture of GOLD which modeled image with Gaussian or Gaussian mixture model, and then mapped covariance of Gaussian into Euclidean space with concatenating to the mean vector for matching Gaussian models. We attribute the gains of our CLM over [38, 45] to the use of two-step metric with the proposed well-motivated parameters. We also compare our CLM with NBNN [6]. It is easy to see that our CLM performs much better than NBNN on Caltech101 and Caltech256. The main differences between our CLM and NBNN are that our CLM employs an effective model-to-model distance and SVM classifier.

Comparison with FV We make a comprehensive comparison with one state-of-the-art BoF method, FV [44], throughout all databases, and also adopt enrichment SIFT (eSIFT) to FV. On all databases except for FMD, our CLM achieves better than or comparable performances with FV when SIFT or eSIFT is used. On

FMD, with SIFT or eSIFT, our CLM is inferior to FV, but with LogCov or eLogCov, our CLM is much better than FV. In our experiments, we find that LogCov or eLogCov is not very suitable for FV, so the relevant results are not reported. It is found that our CLM is more sensitive to local descriptors than FV, as eSIFT brings less or no gains on FV while our CLM greatly benefits from the enrichment on SIFT or LogCov. On CUB200-2011, we also adopt the proposed PBR to Fisher vector with eSIFT (FV+eSIFT+PBR). FV+eSIFT+PBR can achieve 33.2% accuracy, which improves FV+eSIFT but is inferior to our CLM+ PBR (36.0%) by 2.8%. Note that PBR is not essential to our proposed method, but it is a simple yet effective method to alleviate the effect of background clutter on our CLM, if necessary.

Comparison with other state-of-the-art methods Some recent results are also presented for comparison. On Caltech101, DeCAF [14] with 6 layers CNN and dropout strategy [47] slightly outperforms our CLM. Without dropout, the result of DeCAF drops to 84.8%. On Caltech256, our CLM outperforms the deep architecture Multipath Hierarchical Matching Pursuit (M-HMP) [4] by 2.9%. Cimpoi et al [13] achieved state-of-the-art results on KTH-TIPS2b and FMD with semantic attributes which are trained on the additional database by combining FV [44] and DeCAF [14]. Our CLM is superior to the method with attributes, FV and DeCAF. By combining attribute features, FV and DeCAF, Cimpoi et al [13] obtained 77.3% and 67.1% accuracy on KTH-TIPS2b and FMD. Kobayashi [29] proposed a histogram transformation method, and it achieves state-of-the-art results on Sports8 and VOC2007.

Summary In this paper, we assess our CLM on eight image benchmarks, as shown in Table 2, which contains various transformations or noisy factors. We claim that (1) the results on Caltech101 and Caltech256 show that our CLM can well deal with location and pose variations of objects; (2) the results on FMD and KTH-TIPS2b show that our CLM is robust to scale, viewpoint, illumination and appearance variation; (3) the results on Sports8 and Sence15 indicate our CLM can well classify scene images with certain background clutters; and (4) the results on CUB200-2011 and VOC2007 demonstrate our CLM also can handle images with complex surroundings, such as heavy background clutters and occlusion.

6.3. Computational complexity analysis

Our CLM for classification mainly consists of three components: extracting local descriptors, computing Gaussian models using Eq.(4) followed by EPN (5) and matrix logarithm in Eq.(8), and learning LRSVM for classification. Most of the computational costs of CLM lie in the eigenvalue decomposition produced by

EPN and matrix logarithm. Their computational complexity are $O(k^3)$ and $O((k+1)^3)$, respectively, where k is the dimension of local descriptors. During joint training of low-rank matrix and SVM classifier, optimizing the objective function (11) consists of alternating SVM minimization problem and trace minimization problem, whose complexity is $O(J(N^2D + D^3 + Bd^3))$, where N is the number of training samples of dimension $D = Bd$, and J is the number of iterations which is less than 3 in our experiments.

Here, we give empirical running time by taking KTH-TIPS2b and Caltech101 as examples. The time of computing image representation, which includes extraction of SIFT at multiple scales, and the time of computation of Gaussian models and embedding matrices, are 30 minutes on KTH-TIPS2b and 1.5 hours on Caltech101. The average time of modeling one image takes about 0.4 second and 0.6 second on relevant databases. For each trial, training (resp. test) of LRSVM takes 20s (resp. 2s) and 7min (resp. 40s) on KTH-TIPS2b and Caltech101, respectively.

7. Discussion and conclusion

The bag-of-features (BoF) is a popular method in classification and recognition fields, demonstrating convincing performance in many computer vision tasks in the last decades. It might seem that training codebook & descriptor coding are indispensable ingredients. However, the codebookless model (CLM) proposed in this work has proven to be an effective alternative method to the BoF methods for image classification. Below we give some discussions about why CLM shows such competitive performance.

Different from the BoF methods, our CLM leverages continuous functions for statistical modeling of local descriptors, which does not need codebook and thus has no quantization brought in. Recent studies [12] showed that high dimensionality can bring impressive performance. The state-of-the-art BoF methods such as SV/VLAD or FV have inherently high dimensionality, which, in our opinion, is the key for characterizing distinctness and discriminativess of individual images as well as image categories. Our CLM directly employs the first- and second-order statistics of high dimensional local descriptors, giving rise to informative image-level models of high dimensionality as well. In this respect, it is worthwhile to study more informative or high dimensional CLM. Moreover, as shown in [9, 10], the CLM is more efficient than the BoF methods for modeling images because learning codebook & coding are not necessary. In addition, the CLM may be more suitable for the tasks where the datasets will be regularly updated or

increased, and thus the codebook in the BoF model has to be regularly adjusted to fit the changing datasets.

The contributions of this paper are concluded as follows. (1) Our work has clearly shown that the CLM is a very competitive alternative to the mainstream BoF model. The above finding, to our best knowledge, has not yet appeared in pervious literatures. We hope our work can raise potential interests in the classification (or retrieval) community and pave a way to future research. (2) Our method enables Gaussian models to be successfully combined with linear SVM classifier, which makes our method scalable and efficient. The key is that we embed Gaussian models into a vector space which also allows us to perform joint low-rank learning and SVM on Gaussian manifold, which is different from pervious related work [38, 45]. Meanwhile, the proposed two well-motivated parameters further improve our CLM. (3) We performed extensive experiments, evaluating various aspects of our CLM and comparing with its counterparts as well as state-of-the-art methods. The comprehensive experiments demonstrated the promising performance of our CLM. In future work, we will extend our method with more effective local features (e.g., CNN features), and apply the proposed method to other vision tasks and practical applications, such as texture classification and segmentation, scene categorization and image retrieval.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (61471082, 61271093) and the Hong Kong RGC General Research Fund Grant (PolyU 5313/13E).

References

[1] J. Arenas-Garcła, K.B. Petersen, L.K. Hansen, Sparse kernel orthonormalized PLS for feature extraction in large data sets., in: Neural Information Processing Systems, 2006.

[2] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Fast and simple calculus on tensors in the Log-Euclidean framework, in: Medical Image Computing and Computer Assisted Intervention, 2005.

[3] C. Beecks, A.M. Zimmer, S. Kirchhoff, T. Seidl, Modeling image similarity by gaussian mixture models and the signature quadratic form distance, in: International Conference on Computer Vision, 2011.

[4] L. Bo, X. Ren, D. Fox, Multipath sparse coding using hierarchical matching pursuit, in: Computer Vision and Pattern Recognition, 2013.

[5] L. Bo, C. Sminchisescu, Efficient match kernel between sets of features for visual recognition, in: Neural Information Processing Systems, 2009.

[6] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: Computer Vision and Pattern Recognition, 2011.

[7] X. Boix, G. Roig, S. Diether, L.V. Gool, Self-adaptable templates for feature coding, in: Neural Information Processing Systems, 2014.

[8] B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, in: International Conference on Computer Vision, 2005.

[9] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic Segmentation with Second-Order Pooling, in: European Conference on Computer Vision, 2012.

[10] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Free-Form Region Description with Second-Order Pooling, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (2014) 1.

[11] C.C. Chang, C.J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27.

[12] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification, in: Computer Vision and Pattern Recognition, 2013.

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Computer Vision and Pattern Recognition, 2014.

[14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International Conference on Machine Learning, 2014.

[15] D.L. Donoho, M. Gavish, I.M. Johnstone, Optimal shrinkage of eigenvalues in the spiked covariance model, arXiv 1311.0851 (2014).

[16] L. Dryden, A. Koloydenko, D. Zhou, Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging, Annals of Applied Statistics (2009).

[17] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) Challenge, International Journal of Computer Vision 88 (2010) 303–338.

[18] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 594–611.

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

[20] L. Gong, T. Wang, F. Liu, Shape of gaussians as feature descriptors, in: Computer Vision and Pattern Recognition, 2009.

[21] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, in: International Conference on Computer Vision, 2005.

[22] G. Griffin, A. Holub, P. Perona, The Caltech-256, Technical Report, California Institute of Technology, 2007.

[23] L. haran, R. Rosenholtz, E.H. Adelson, Material perception: What can you see in a brief glance?, Journal of Vision 9 (2009) 784.

[24] M.T. Harandi, M. Salzmann, R. Hartley, From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices, in: European Conference on Computer Vision, 2014.

[25] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, M. Harandi, Kernel methods on the riemannian manifold of symmetric positive definite matrices, in: Computer Vision and Pattern Recognition, 2013.

[26] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Computer Vision and Pattern Recognition, 2010.

[27] S. Ji, J. Ye, Linear dimensionality reduction for multi-label classification, in: International Joint Conference on Artificial Intelligence, 2009.

[28] B. Jiang, L. Zhang, H. Lu, C. Yang, M.H. Yang, Saliency detection via absorbing markov chain, in: International Conference on Computer Vision, 2013.

[29] T. Kobayashi, Dirichlet-based histogram feature transform for image classification, in: Computer Vision and Pattern Recognition, 2014.

[30] P. Koniusz, F. Yan, P.H. Gosselin, K. Mikolajczyk, Higher-order Occurrence Pooling on Mid- and Low-level Features: Visual Concept Detection, Technical Report, 2013.

[31] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition, 2006.

[32] L.J. Li, F.F. Li, What, where and who? classifying events by scene and object recognition., in: International Conference on Computer Vision, 2007.

[33] P. Li, Q. Wang, Local log-euclidean covariance matrix (l2ecm) for image representation and its applications, in: European Conference on Computer Vision, 2012.

[34] P. Li, Q. Wang, L. Zhang, A novel earth mover's distance methodology for image matching with gaussian mixture models, in: International Conference on Computer Vision, 2013.

[35] L. Lin, P. Luo, X. Chen, K. Zeng, Representing and recognizing objects with massive local image patches, Pattern Recognition 45 (2012) 231–240.

[36] M. Lovric, M. Min-Oo, E.A. Ruh, Multivariate normal distributions parametrized as a riemannian symmetric space, Journal of Multivariate Analysis 74 (2000) 36–48.

[37] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[38] H. Nakayama, T. Harada, Y. Kuniyoshi, Global gaussian approach for scene categorization using information geometry, in: Computer Vision and Pattern Recognition, 2010.

[39] Z. Peng, L. Lin, R. Zhang, J. Xu, Deep boosting: Layered feature mining for general image classification, Neurocomputing (2015).

[40] X. Pennec, P. Fillard, N. Ayache, A riemannian framework for tensor computing, International Journal of Computer Vision (2006) 41–66.

[41] H. Permuter, J. Francos, I. Jermyn, A study of gaussian mixture models of color and texture features for image classification and segmentation, Pattern Recognition 39 (2006) 695–706.

[42] W.K. Pratt, Digital Image Processing, 4th Edition, John Wiley & Sons, Inc., New York, NY, USA, 2007.

[43] Y. Rubner, C. Tomasi, L.J. Guibas, The Earth Mover's Distance as a metric for image retrieval, International Journal of Computer Vision 40 (2000) 99–121.

[44] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the Fisher vector: Theory and practice, International Journal of Computer Vision 105 (2013) 222–245.

[45] G. Serra, C. Grana, M. Manfredi, R. Cucchiara, GOLD: gaussians of local descriptors for image representation, Computer Vision and Image Understanding 134 (2015) 22–32.

[46] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: International Conference on Computer Vision, 2003.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.

[48] C. Stein, Lectures on the theory of estimation of many parameters, Journal of Mathematical Sciences 34 (1986) 1373–1403.

[49] V. Sydorov, M. Sakurada, C.H. Lampert, Deep fisher kernels - end to end learning of the Fisher kernel GMM parameters, in: Computer Vision and Pattern Recognition, 2014.

[50] O. Tuzel, F. Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: European Conference on Computer Vision, 2006.

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

[52] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Technical Report, 2011.

[53] J. Wang, J. Yang, K. Yu, F. Lv, T.S. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Computer Vision and Pattern Recognition, 2010.

[54] J. Weston, S. Bengio, N. Usunier, Wsabie: Scaling up to large vocabulary image annotation, in: International Joint Conference on Artificial Intelligence, 2011.

[55] Y.B. Yang, Q.H. Zhu, X.J. Mao, L.Y. Pan, Visual feature coding for image classification integrating dictionary structure, Pattern Recognition 48 (2015) 3067 – 3075.

[56] B. Yao, G. Bradski, L. Fei-Fei, A codebook-free and annotation-free approach for fine-grained image categorization, in: Computer Vision and Pattern Recognition, 2012.

[57] N. Zhang, R. Farrell, T. Darrell, Pose pooling kernels for sub-category recognition, in: Computer Vision and Pattern Recognition, 2012.

[58] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, Q. Tian, Towards codebook-free: Scalable cascaded hashing for mobile image search, IEEE Transactions on Multimedia 16 (2014) 601–611.

[59] X. Zhou, K. Yu, T. Zhang, T.S. Huang, Image classification using supervector coding of local image descriptors, in: European Conference on Computer Vision, 2010.