Fast Low Rank Representation based Spatial Pyramid Matching for Image Classification

Xi Peng[a], Rui Yan[b], Bo Zhao[a], Huajin Tang[b,*], Zhang Yi[b]

[a] *Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR) Singapore 138632*

[b] *College of Computer Science, Sichuan University, Chengdu, 610065, China.*

Abstract

Spatial Pyramid Matching (SPM) and its variants have achieved a lot of success in image classification. The main difference among them is their encoding schemes. For example, ScSPM incorporates Sparse Code (SC) instead of Vector Quantization (VQ) into the framework of SPM. Although the methods achieve a higher recognition rate than the traditional SPM, they consume more time to encode the local descriptors extracted from the image. In this paper, we propose using Low Rank Representation (LRR) to encode the descriptors under the framework of SPM. Different from SC, LRR considers the group effect among data points instead of sparsity. Benefiting from this property, the proposed method (i.e., LrrSPM) can offer a better performance. To further improve the generalizability and robustness, we reformulate the rank-minimization problem as a truncated projection problem. Extensive experimental studies show that LrrSPM is more efficient than its counterparts (e.g., ScSPM) while achieving competitive recognition rates on nine image data sets.

Keywords:
Closed-form Solution, Efficiency, Image Classification, Thresholding Ridge Regression, ℓ_2-regularization

[*] Corresponding author

Email addresses: pangsaai@gmail.com (Xi Peng), ryan@i2r.a-star.edu.sg (Rui Yan), zhaob@i2r.a-star.edu.sg (Bo Zhao), htang@i2r.a-star.edu.sg (Huajin Tang), zhangyi@scu.edu.cn (Zhang Yi)

Preprint submitted to Knowledge based Systems *October 27, 2015*

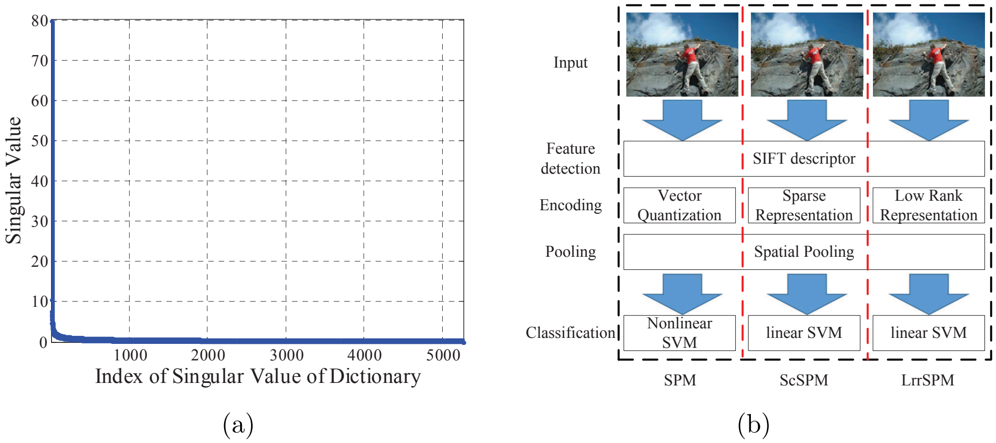
1. Introduction

Image classification system automatically assigns an unknown image to a category according to its visual content, which has been a major research direction in computer vision and pattern recognition. Image classification has two major challenges. First, each image may contain multiple objects with similar low level features, it is thus hard to accurately categorize the image using the global statistical information such as color or texture histograms. Second, a medium-sized grayscale image (e.g., 1024×800) corresponds to a vector with dimensionality of $819,200$, this brings up the scalability issue with image classification techniques.

To address these problems, numerous impressive approaches [1, 2, 3, 4, 5] have been proposed in the past decade, among which one of the most popular methods is Bag-of-Features (BOF) or called Bag-of-Words (BOW). BOW originates from document analysis [6, 7]. It models each document as the joint probability distribution of a collection of words. [8, 9, 10] incorporated the insights of BOW into image analysis by treating each image as a collection of unordered appearance descriptors extracted from local patches. Each descriptor is quantized into a discrete "visual words" corresponding to a given codebook (i.e., dictionary), and then the compact histogram representation is calculated for semantic image classification.

The huge success of BOF has inspired a lot of works [11, 12]. In particular, Lazebnik et al. [13] proposed Spatial Pyramid Matching (SPM) which divides each image into $2^l \times 2^l$ blocks in different scales $l = 0, 1, 2$, then computes the histograms of local features inside each block, and finally concatenates all histograms to represent the image. Most state-of-the-art systems such as [14, 15, 16, 17, 18] are implemented under the framework of SPM and have achieved impressive performance on a range of image classification benchmarks like Columbia University Image Library-100 (COIL100) [19] and Caltech101 [20]. Moreover, SPM has been extensively studied for solving other image processing problems, e.g., image matching [21], fine-grained image categorization [22]. It has also been incorporated into deep learning to make deep convolutional neural networks (CNN) [23] handling arbitrary sized images possible. To obtain a good performance, SPM and its extensions have to pass the obtained representation to a Support Vector Machine classifier (SVM) with nonlinear Mercer kernels. This brings up the scalability issue with SPMs in practice.

Although SPM has achieved state-of-the-art recognition rates on a range



(a)

(b)

Figure 1: (a) Singular values of a given codebook. The codebook consists of 5,120 basis. It shows that most energy concentrates on the top singular values. (b) Schematic comparison of the original SPM, ScSPM and the proposed LrrSPM.

of databases, its computational complexity is very high. To speed up SPM, Yang et al. [24] proposed using Sparse Code (SC) instead of Vector Quantization (VQ) to encode each Scale-Invariant Feature Transform (SIFT) descriptor [25] over a codebook. Benefiting from the good performance of sparse code, Yang's method (namely ScSPM) with linear SVM obtains a higher classification accuracy, while using less time for training and testing.

The success of ScSPM could be attributed to that SC can capture the manifold structure of data sets. However, SC encodes each data point independently without considering the grouping effect among data points. Moreover, the computational complexity of SC is proportional to the cube of the size of codebook (denoted by n). Therefore, it is a daunting task to perform ScSPM when n is larger than $10,000$. To solve these two problems, this paper proposes using Low Rank Representation (LRR) rather than SC to hierarchically encode each SIFT descriptor.

To show our motivation, i.e., the collections of descriptor and representation are low rank, we carry out k-means clustering algorithm over the SIFT descriptors of the Caltech101 database [20] and obtain a codebook $\mathbf{D}$ consisting of 5,120 cluster centers. By performing Singular Value Decomposition (SVD) over the codebook shown in Figure 1(a), one can see that most energy (over 98%) concentrates on the top 2% singular values. In other words, the data space spanned by the codebook is low rank. For a testing data set $\mathbf{X} \in span(\mathbf{D})$, its representation can be calculated by $\mathbf{X} = \mathbf{DC}$. Since $\mathbf{X}$ and $\mathbf{D}$ are low rank, then $\mathbf{C}$ must be low rank. This observation motivates us to develop an novel SPM method, namely Low Rank Representation based Spatial Pyramid Matching (LrrSPM). Figure 1 illustrates a schematic comparison of the original SPM, ScSPM, and LrrSPM. It should be pointed out that, SPM, ScSPM, and LrrSPM are three basic models which do not incorporate the label information, kernel function learning, and multiple descriptors learning into their encoding schemes. The major difference among them is that both SPM and ScSPM perform encoding in the vector space, whereas LrrSPM calculates the representation in the matrix space.

The contributions of the paper are summarized as follows: 1) Different from the existing LRR methods [26, 27, 28], the proposed LrrSPM is a multiple-scale model which integrates more discriminative information compared to the traditional LRR. 2) Most existing LRR methods are proposed for clustering, which cannot be used for classification directly. In this paper, we fill this gap based on our new mathematical formulation. 3) Our LrrSPM has a closed form solution and can be calculated very fast. After the dictionary is learnt from the training data, LrrSPM computes the representation of testing data by simply projecting each testing datum into another space. Extensive experimental results show that LrrSPM achieves competitive results on nine image databases and is $25-50$ times faster than ScSPM.

The rest of the paper is organized as follows: Section 2 provides a brief review on two classic image classification methods, i.e., SPM [13] and ScSPM [24]. Section 3 presents our method (i.e., LrrSPM) which uses multiple-scale low rank representation to represent each image. Section 4 carries out some experiments using nine image data sets and several popular approaches. Finally, Section 5 concludes this work.

Notations: Lower-case bold letters represent column vectors and upper-case bold ones denote matrices. $\mathbf{A}^T$ and $\mathbf{A}^{-1}$ denote the transpose and pseudo-inverse of the matrix $\mathbf{A}$, respectively. $\mathbf{I}$ denotes the identity matrix. Table 1 summarizes some notations used throughout the paper.

2. Related works

In this section, we mainly introduce SPM and ScSPM which employ two basic encoding schemes, i.e., vector quantization and sparse code. To the best of our knowledge, most of other SPM based methods can be regarded as

Table 1: Some used mathematic notations.

Notation	Definition
n	the number of descriptors (features)
l	the scale or resolution of a given image
m	the dimensionality of the descriptors
s	the number of subjects
k	the size of codebook
r	the rank of a given matrix
$\mathbf{y}$	an image
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$	a set of features
$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k]$	codebook
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$	the representation of $\mathbf{X}$ over $\mathbf{D}$

the extensions of them, e.g., the method proposed in [17] is a kernel version of ScSPM.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a collection of the descriptors and each column vector of $\mathbf{X}$ represents a feature vector $\mathbf{x}_i \in \mathbb{R}^m$, SPM) [13] applies VQ to encode $\mathbf{x}_i$ via

$$\min_{\mathbf{C},\mathbf{D}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|_2^2 \ \text{ s.t. } Card(\mathbf{c}_i) = 1, \tag{1}$$

where $\|\cdot\|_2$ denotes ℓ_2-norm, $\mathbf{c}_i \in \mathbb{R}^k$ is the representation or called the cluster assignment of $\mathbf{x}_i$, the constraint $Card(\mathbf{c}_i) = 1$ guarantees that only one entry of $\mathbf{c}_i$ is with value of one and the rest are zeroes, and $\mathbf{D} \in \mathbb{R}^{m \times k}$ denotes the codebook.

In the training phase, $\mathbf{D}$ and $\mathbf{C}$ are iteratively solved, and VQ is equivalent to the classic k-means clustering algorithm which aims to

$$\min_{\mathbf{D}} \sum_{i=1}^{n} \sum_{j=1}^{k} \min \|\mathbf{x}_i - \mathbf{d}_j\|_2^2, \tag{2}$$

where $\mathbf{D}$ consists of k cluster centers identified from $\mathbf{X}$.

In the testing phase, each $\mathbf{x}_i \in \mathbf{X}$ is actually assigned to the nearest $\mathbf{d}_j \in \mathbf{D}$. Since each $\mathbf{c}_i$ has only one nonzero element, it discards a lot of information for $\mathbf{x}_i$ (so-called hard coding problem). To solve this problem, Yang et al. [24] proposed ScSPM which uses sparse representation to represent $\mathbf{x}_i$ via

$$\min_{\mathbf{c}_i} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_1, \tag{3}$$

where $\|\cdot\|_1$ denotes ℓ_1-norm which sums the absolute values of a vector, and $\lambda > 0$ is the sparsity parameter.

The advantage of ScSPM is that the sparse representation $\mathbf{c}_i$ has a small number of nonzero entries and it can represent $\mathbf{x}_i$ better with less reconstruction errors. Extensive studies [15, 24] have shown that ScSPM with linear SVM is superior to the original SPM with nonlinear SVM. The disadvantage of ScSPM is that each data point $\mathbf{x}_i$ is encoded independently, and thus the sparse representation $\mathbf{c}_i$ cannot reflect the class structure. Moreover, the computational complexity of sparse coding is very high so that any medium-sized data set will bring up scalability issue with ScSPM. Motivated by SPM and ScSPM, a lot of works have been proposed, e.g., nonlinear extensions [17], supervised extensions [18], and multiple descriptors fusing [29].

3. Fast Low Rank Representation Learning for Spatial Pyramid Matching

In this section, we introduce LrrSPM in three steps. First, we give the basic formulation of LrrSPM which requires solving a rank-minimization problem. Moreover, we theoretically show that the representation is also low rank if the data space spanned by the codebook is low rank. This provides a theoretical foundation for our method. Second, we further improve the generalization ability and robustness of the basic model by adopting the regularization technique and recent theoretical development in robustness learning. Finally, a real-world example is given to show the effectiveness of the obtained representation.

LRR can capture the relations among different subjects, which has been widely studied in image clustering [28], semi-supervised learning [30], and dimension reduction [31]. In this paper, we introduce LRR into SPM to hierarchically encode each local descriptor. Note that, it is nontrivial to incorporate LRR into the framework of SPM due to the following reasons. 1) the traditional LRR are generally used for clustering, which cannot be directly used for classification. In the context of classification, we need to reformulate the objective function that must lead to a different optimization problem. 2) To improve the robustness, the traditional LRR enforces ℓ_1-norm over the possible errors, which results a very high computational complexity. In this paper, we do not take this error-removal strategy but perform truncated operator into the projection space to eliminate the effect of errors.

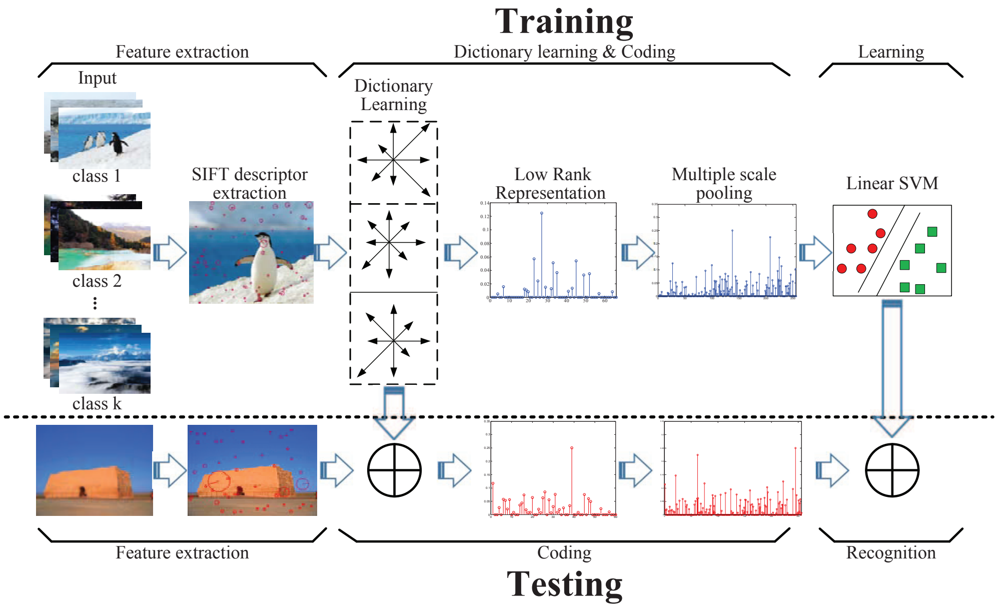


Figure 2: Flow chart of the proposed algorithm.

Figure 2 provides a flow chart of the proposed algorithm. Different from the existing LRR methods, we use the training data rather than all samples as codebook. We aim at solving

$$\min_{\mathbf{C}} \ rank(\mathbf{C}) \quad \text{s.t. } \mathbf{X} = \mathbf{DC}, \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ denotes the collection of SIFT descriptors, $\mathbf{C}^{n\times k}$ denotes the representation of $\mathbf{X}$ over the codebook $\mathbf{D}^{m\times k}$, and $\mathbf{D}^{m\times k}$ generally consists of k cluster centers.

Note that, LrrSPM (i.e., eq.4) enforces rank minimization operator over the representation matrix, which is significantly different from the standard SPM (i.e., eq.1) and ScSPM (i.e., eq.3). LrrSPM exploits the grouping effect of data points in 2-dimensional space, whereas SPM and ScSPM obtain the representation in 1-dimensional space (i.e., vector).

The optimal solution to eq.4 is given by

$$\mathbf{C}^* = \mathbf{D}^{-1}\mathbf{X}, \tag{5}$$

where $\mathbf{D}^{-1}$ denotes the inverse of $\mathbf{D}$. Note that, one always calculates the pseudo-inverse of $\mathbf{D}$ in practice, denoted by $\mathbf{C}^* = \mathbf{D}^\dagger \mathbf{X}$.

Based on the above results, we have

$$\begin{aligned} \operatorname{rank}(\mathbf{C}^*) &\leq \min\{\operatorname{rank}(\mathbf{D}^\dagger), \operatorname{rank}(\mathbf{X})\} \\ &= \min\{\operatorname{rank}(\mathbf{D}), \operatorname{rank}(\mathbf{X})\}. \end{aligned} \tag{6}$$

This shows how the rank of $\mathbf{D}$ affects that of $\mathbf{C}$. Moreover, it also verifies our motivation once again, i.e., $\mathbf{C}$ must be low rank when the dictionary $\mathbf{D}$ is low rank.

To avoid overfitting, we further incorporate the regularization technique and obtain the following solution:

$$\mathbf{C}^* = (\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^T\mathbf{X}, \tag{7}$$

where $\lambda \geq 0$ is the regularization parameter and $\mathbf{I}$ denotes the identity matrix. Note that, nuclear-norm based representation is actually equivalent to the frobenius-norm based representation under some conditions, please refer to [32] for more theoretical details.

In practice, $\mathbf{D}$ probably contains the errors such as noise, and thus the obtained representation may be sensitive to various corruptions. To achieve robust results, we recently proved that *the trivial coefficients (i.e., small coefficients) always correspond to the representation over errors* in ℓ_2-norm based projection space, i.e.,

Lemma 1 ([33])**.** *For any nonzero data point* $\mathbf{x}$ *in the subspace* $\mathcal{S}_{\mathbf{D}_x}$ *except the intersection between* $\mathcal{S}_{\mathbf{D}_x}$ *and* $\mathcal{S}_{\mathbf{D}_{-x}}$*, i.e.,* $\mathbf{x} \in \{\mathcal{S} | \mathcal{S} = \mathcal{S}_{\mathbf{D}_x} \backslash \mathcal{S}_{\mathbf{D}_{-x}}\}$*, the optimal solution of*

$$\min \ \|\mathbf{c}\|_2 \ \ \text{s.t.}\ \mathbf{x} = \mathbf{D}\mathbf{c}, \tag{8}$$

over $\mathbf{D}$ *is given by* $\mathbf{c}^*$ *which is partitioned according to the sets* $\mathbf{D}_x$ *and* $\mathbf{D}_{-x}$*, i.e.,* $\mathbf{c}^* = \begin{bmatrix} \mathbf{c}_x^* \\ \mathbf{c}_{-x}^* \end{bmatrix}$*. Thus, we must have* $[\mathbf{c}_x^*]_{r0,1} > [\mathbf{c}_{-x}^*]_{1,1}$*.* $\mathbf{D}_x$ *consists of the intra-subject data points of* $\mathbf{x}$ *and* $\mathbf{D}_{-x}$ *consists of the inter-subject data points of* $\mathbf{x}$*.* $[\mathbf{c}_x^*]_{r_x,1}$ *denotes the* r_x*-th largest absolute value of the entries of* $\mathbf{c}_x^*$*, and* r_x *is the dimensionality of* $\mathcal{S}_{\mathbf{D}}$*. Note that, noise and outlier could be regarded as a kind of inter-subject data point of* $\mathbf{x}$*.*

Lemma 2 ([33])**.** *Consider a nonzero data point* $\mathbf{x}$ *in the intersection between* $\mathcal{S}_{\mathbf{D}_x}$ *and* $\mathcal{S}_{\mathbf{D}_{-x}}$*, i.e.,* $\mathbf{x} \in \{\mathcal{S} | \mathcal{S} = \mathcal{S}_{\mathbf{D}_x} \cap \mathcal{S}_{\mathbf{D}_{-x}}\}$*. Let* $\mathbf{c}^*$*,* $\mathbf{z}_0$*, and* $\mathbf{z}_e$ *be the* *optimal solution of*

$$\min \ \|\mathbf{c}\|_2 \ \ \text{s.t.}\ \mathbf{x} = \mathbf{Dc} \tag{9}$$

over $\mathbf{D}$*,* $\mathbf{D}_x$*, and* $\mathbf{D}_{-x}$*, and* $\mathbf{c}^* = \begin{bmatrix} \mathbf{c}_x^* \\ \mathbf{c}_{-x}^* \end{bmatrix}$ *is partitioned according to the sets* $\mathbf{D} = [\mathbf{D}_x \ \mathbf{D}_{-x}]$*. If* $\|\mathbf{z}_0\|_p < \|\mathbf{z}_e\|_p$*, then* $\mathbf{c}_x^* \neq \mathbf{0}$ *and* $\mathbf{c}_{-x}^* = \mathbf{0}$*.*

Based on Lemmas 1 and 2, we can obtain a robust representation by truncating the coefficients over errors. Mathematically,

$$\mathbf{Z} = \mathcal{H}_\epsilon(\mathbf{C}^*), \tag{10}$$

where the hard thresholding operator $\mathcal{H}_\epsilon(\mathbf{C})$ keeps large entries and eliminates trivial ones for each column of $\mathbf{C}^*$. $\mathbf{C}^*$ is the optimal solution of eq.7 which is also the minimizer of eq.8.

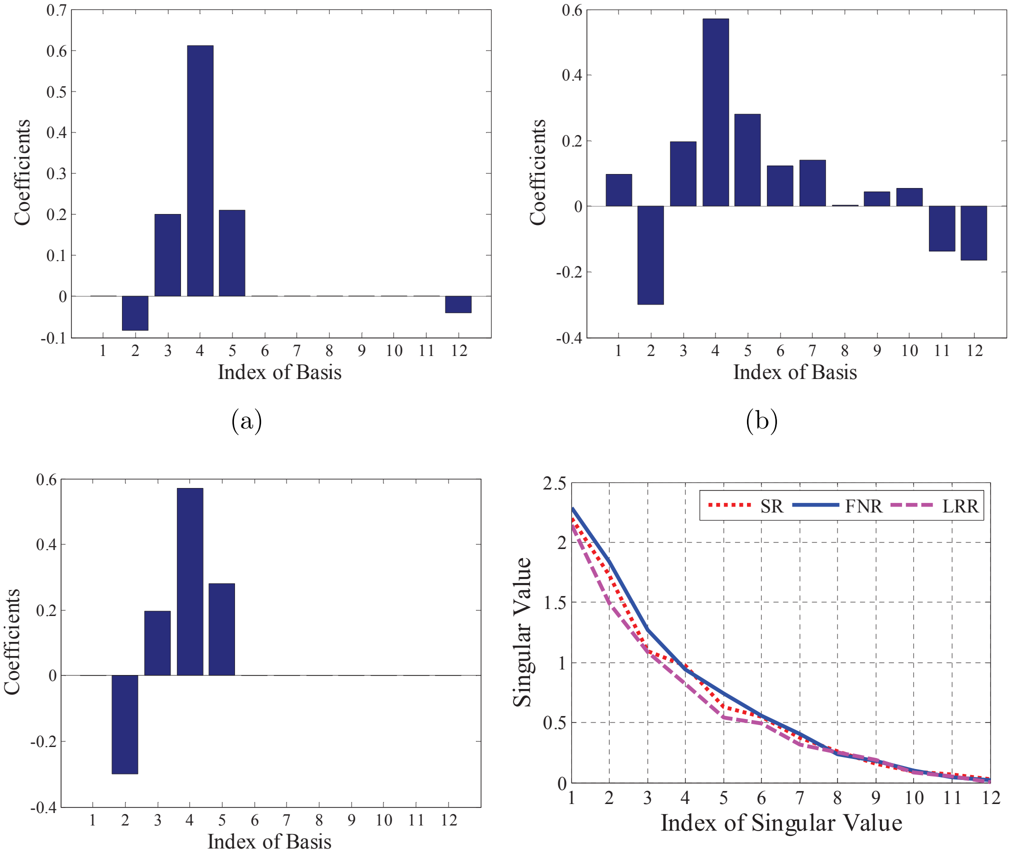
Figure 3 shows a comparison among sparse code (eq.3), ℓ_2-norm regularized representation (eq.7), and LrrSPM (eq.10). In this example, we carry out experiments using two subsets of Extended Yale database B [34], where the dictionary subset and the testing subset consists of 12 sample, respectively. We randomly select one sample from the first subject as testing sample and calculated its representation. Figure 3(a)–3(c) illustrates the obtained representations and Figure 3(d) shows the singular values of the representation matrix for all testing data. From the results, the proposed method has the following advantages: 1) LrrSPM is more discriminative since its coefficients over the second subjects are zeroes. 2) it provides a compact representation with better representative capacity.

Algorithm 1 summarizes our algorithm. Similar to [13, 24], the codebook $\mathbf{D}$ can be generated by the k-means clustering method or dictionary learning methods such as [35]. For training or testing purpose, LrrSPM can get the low rank representation in an online way, which further explores the potential of LRR in online and incremental learning. Moreover, our method is very efficient since its coding process only involves a simple projection operation.

4. Experiments

4.1. Baseline Algorithms and Databases

We compared our method with four SPM methods using nine image databases. The MATLAB code of LrrSPM can be downloaded from the authors' website `www.machineilab.org/users/pengxi` and the codes of the



(a) (b) (c) (d)

Figure 3: An example using two subsets of Extended Yale database B, where the testing subset and the dictionary subset consist of 12 samples, respectively. The first and the last six atoms of the dictionary belong to two different subjects. (a)–(c) sparse representation (eq.3), ℓ_2-regularized representation (eq.7), and the proposed LrrSPM (eq.10) of a given querying sample that belongs to the first subject. (d) The singular value of the coefficient matrices. Figure 3(a)–3(c) show that only the coefficients of LRR over the second subject are zeroes. This makes our model more discriminative. Moreover, Figure 3(d) shows that the energy of our method is more concentrated, i.e., our model is more competitive in terms of the principle of minimum description length.

baseline methods are publicly accessible. Besides our own experimental results, we also quote some results in the literature.

The baseline methods include BOF [10] with linear SVM (LinearBOF) and kernel SVM (KernelBOF), SPM [13] with linear SVM (LinearSPM) and

Algorithm 1 Fast LRR for Spatial Pyramid Matching (LrrSPM).

```
Input: The codebook $\mathbf{D} \in \mathbb{R}^{m \times k}$, the input image $\mathbf{y}$, and the regularization parameter $\lambda$.
Calculate $\mathbf{P} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T$ and store it.
For each image $\mathbf{y}$, detect and extract the SIFT descriptors $\mathbf{X}$ from $\mathbf{y}$.
Calculate the representation of $\mathbf{y}$ via $\mathbf{C} = \mathbf{P}\mathbf{X}$ and normalize each column of $\mathbf{C}$ to have a unit $\ell_2$-norm.
If the dictionary contains errors, obtain the LRR of $\mathbf{y}$ by thresholding the trivial entires of $\mathbf{c}_{ji} = [c_{1i}, c_{2i}, \cdots, c_{ki}]^T$ at $\epsilon$ (generally, $\epsilon = 98\%$) via
    $c_{ji} = \begin{cases} c_{ji} & k\frac{c_{ji}}{\sum_j c_{ji}} < \epsilon \\ 0 & otherwise \end{cases} \quad (11)$
Divide $\mathbf{C}$ into $2^l \times 2^l$ blocks, where $l$ denotes the scale or the level of the pyramid. For each block at each level, perform max pooling for each block at each level via $\mathbf{z}_i = \max\{|\mathbf{c}_i^1|, |\mathbf{c}_i^2|, \cdots, |\mathbf{c}_i^b|\}$, where $\mathbf{c}_i^j$ denotes the $j$-th LRR vector belonging to the $i$-th block, and $b = 2^l \times 2^l$.
Output: Form a single representation vector for $\mathbf{y}$ by concatenating the set of $\mathbf{z}_i$.
```

kernel SVM (KernelSPM), Sparse Coding based SPM with linear SVM(ScSPM) [24], and Locality-constrained Linear Coding with linear SVM (LLC) [15].

The used databases include five scene image data sets, three object image data sets (i.e., 17flowers [36], COIL20 [37] and COIL100 [19]), and one facial image database (i.e., Extended Yale B [34]). The scene image data sets are from Oliva and Torralba [38], Fei-Fei and Perona [10], Lazebnik et al. [13], Fei-Fei et al. [20], and Griffin et al. [39] which are referred to as OT, FP, LS, Caltech101, and Caltech256, respectively. Table 2 gives a brief review on these data sets.

4.2. Experimental setup

To be consistent with the existing works [13, 15, 24], we use dense sampling technique to divide each image into $2^l \times 2^l$ blocks (patches) with a step size of 6 pixels, where $l = 0, 1, 2$ denotes the scale. And we extract the SIFT descriptors from each block as features. To obtain the codebook, we use the k-means clustering algorithm to find 256 cluster centers for each data set and use the same codebook for different algorithms. In each test, we split the samples per subject into two parts, one is for training and the other is for

Table 2: A summarization of the evaluated databases. s denotes the number of classes and p denotes the number of samples for each subject.

Databases	Type	Data Size	Image Size	p	s
OT	scene	2688	256 × 256	260–410	8
FP	scene	3860	250 × 300	210–410	13
LS	scene	4486	250 × 300	210–410	15
Caltech101	scene	9144	300 × 200	31–800	102
Caltech256	scene	30,607	300 × 200	80–827	256
17flowers	flowers	1,360	–	80	17
COIL20	object	1440	128 × 128	72	20
COIL100	object	7200	128 × 128	72	100
Extended Yale B	face	2414	168 × 192	59–64	38

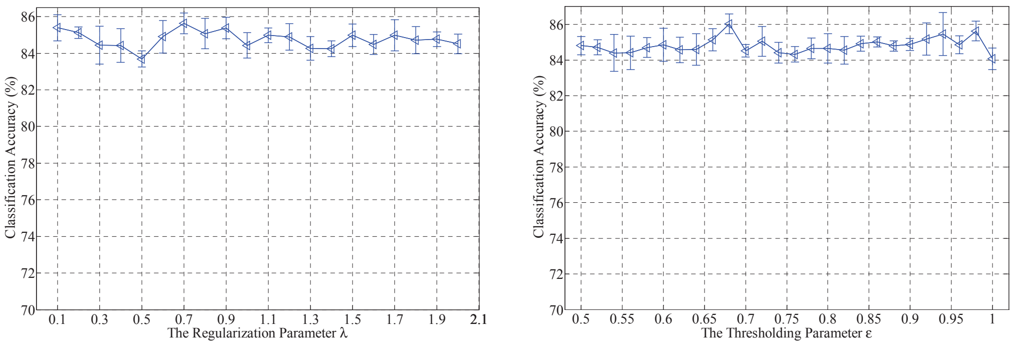
testing. Following the common benchmarking procedures, we repeat each experiment five times with different training and testing data partitions and record the average of per-subject recognition rates and the time costs for each test. We report the final results by the mean and standard deviation of the recognition rates and the time costs. For the LrrSPM approach, we fix $\epsilon = 0.98$ and assign different λ for different databases. For the competing approaches, we directly adopt the parameters configuration in the original works [13, 15, 24]. Moreover, we also quote the performance of these methods reported in the original works.

4.3. Influence of the parameters

LrrSPM has two user-specified parameters, the regularization parameter λ is used to avoid overfitting and the thresholding parameter ϵ is used to eliminate the effect of the errors. In this section, we investigate the influence of these two parameters on OT data set. We fix $\epsilon = 0.98$ ($\lambda = 0.7$) and reported the mean classification accuracy of LrrSPM with the varying λ (ϵ). Figure 3 shows the results, from which one can see that LrrSPM is robust to the choice of the parameters. When λ increases from 0.2 to 2.0 with an interval of 0.1, the accuracy ranges from 83.68% to 85.63%; When ϵ increases from 50% to 100% with an interval of 2%, the accuracy ranges from 84.07% to 86.03%.

4.4. Performance with Different Sized Codebooks

In this Section, we evaluate the performance of LrrSPM when the size of codebook increases from 256 to 4096. we carry out experiments on the Caltech101 data set by randomly selecting 30 samples per subject for training



(a) The influence of λ, where $\epsilon = 0.98$.

(b) The influence of ϵ, where $\lambda = 0.70$.

Figure 4: The mean and standard deviation of the recognition rates of LrrSPM on the OT database.

Table 3: The performance of LrrSPM on Caltech101 w.r.t. increasing size of codebook, where the size of codebook k increases from 256 to 4096.

Metric	$k = 256$	$k = 512$	$k = 1024$	$k = 2048$	$k = 4096$
Accuracy	65.89 ± 1.03	67.75 ± 0.92	68.43 ± 1.04	69.81 ± 1.01	70.37 ± 0.98
Time cost	640.83	1005.74	1691.35	3697.03	8952.43

and using the rest for testing. The λ is set as 0.7 for LrrSPM. Table 3 shows that with increasing k, LrrSPM achieves better recognition results but takes more time for coding and classification. Specifically, when k increases from 256 to 4096, the accuracy increases by 4.48%, but the computing time increases by 1397%.

4.5. Scene Classification

In this section, the experimental studies consist of two parts. The first part reports the performance of LrrSPM on three scene image databases. The codebook consists of 256 bases identifying by the k-means method. For each data set, we randomly choose 100 samples from each subject for training and used the rest for testing.

Table 4 shows that LrrSPM is slightly better than the other evaluated algorithms in most tests. Although LrrSPM is not the fastest method, it finds a good balance between the efficiency and the classification rate. On the OT database, the speed of LrrSPM is about 5.49 and 46.07 times faster

Table 4: The classification accuracy and the time cost of different methods on the OT, FP, and LS databases.

Algorithms	the OT database		the FP database		the LS database	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
LrrSPM (Ours)	**85.63±0.56**	116.67	**80.90±0.75**	159.84	**76.34±0.58**	189.79
LinearBOF	78.95 ± 0.12	102.76	72.13 ± 0.43	195.85	66.38 ± 0.88	169.25
KernelBOF	76.28 ± 0.16	103.65	68.18 ± 0.24	203.89	62.67 ± 0.78	178.34
linearSPM	76.09 ± 0.55	209.14	71.63 ± 0.51	239.44	66.91 ± 0.78	188.24
KernelSPM	73.52 ± 0.64	196.20	64.97 ± 1.50	266.96	60.83 ± 0.39	196.35
ScSPM	84.44 ± 0.24	5375.41	79.04 ± 0.91	5841.89	74.40 ± 0.45	9539.62
LLC	85.55 ± 0.34	640.27	80.34 ± 0.76	943.81	**76.99±1.21**	1059.96
Rasiwasia's method	-	-	76.20	-	72.50 ± 0.30	-

than ScSPM and LLC, respectively. On the LS database, the speedups are 5.59 and 50.26 times.

The second part of experiment reports the performance of the evaluated methods using Caltech101, Caltech256, and Oxford 17flowers database by randomly selecting 30 samples per subject for training and using the rest for testing. In the tests, the dictionary contains 256 bases identified by the k-means clustering method. We fix $\lambda = 0.7$, $\lambda = 0.24$, and $\lambda = 0.145$ for LrrSPM on these three data sets.

Table 5 reports the results from which we can find that, on the Caltech101 data set, the recognition rates of LrrSPM is 28.78% higher than that of LinearBOF, 20.73% higher than that of Kernel BOF, 22.36% higher than that of LinearSPM, 12.38% higher than that of KernelSPM, 0.5% higher than that of ScSPM and 1.97% lower than that of LLC. However, LrrSPM only takes about 3% (30%) CPU time of ScSPM (LLC). On the 17flowers database, LrrSPM outperforms the other evaluated methods by a considerable performance margin. Its recognition rate is 4.98% higher than ScSPM with 21 times speedup.

Besides our experimental implementations, Table 6 summarizes some state-of-the-art results reported by [1, 13, 24, 40, 41, 42] on Caltech101. One can find that we do not reproduce the results reported in the literature for some evaluated methods. This could be attributed to the subtle engineering details. Specifically, Lazebnik et al. [13] only used a subset of Caltech101 (50 images for each subject) rather than all samples. [15, 24] used a larger codebook ($k = 2048$) and the codebook could probably be different even

Table 5: The classification accuracy and the time cost of different methods on the Caltech101, Caltech256, and 17flowers database.

Algorithms	Caltech101		Caltech256		17flowers	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
LrrSPM (Ours)	65.89± 1.03	640.83	27.43± 0.98	2518.13	61.42± 1.46	121.00
LinearBoF	37.11± 1.11	307.50	15.09± 1.45	2474.96	40.19± 1.90	45.79
KernelBoF	45.16± 0.81	315.62	11.25± 0.46	3035.73	34.70± 2.79	47.81
linearSPM	43.53± 1.17	410.79	23.12± 0.27	2511.19	44.12± 2.45	86.66
KernelSPM	53.51± 1.11	467.68	12.03± 0.48	5819.58	36.35± 2.20	94.79
ScSPM	65.39± 1.21	18964.84	28.60± 0.15	58313.03	56.44± 0.54	2614.39
LLC	67.86± 1.17	2203.58	29.35± 0.42	6893.68	59.89± 1.55	747.49

Table 6: The classification accuracy on Caltech101 and Caltech256 reported by some recent literatures. DBN, CNN, and OtC are the abbreviations of deep belief network, convolutional neural network, and object to class method, respectively.

KernelSPM	ScSPM	LLC	GMatching	DBN	CNN	OtC
64.60 ± 0.80	73.20 ± 0.54	73.44	80.30 ± 1.2	65.40	66.30	64.26
–	34.02 ± 0.35	41.19	38.1	–	–	–

though the size of codebook is fixed due to the randomness. Duchenne et al. [1] reported the state-of-the-art accuracy of 80.30% and 38.1% on Caltech101 and Caltech256 by using graph matching based method to improve the performance of classifier. With multiple kernel learning, Yang et al. [29] proposed a model which achieves 84.3% of accuracy on Caltech101. This significant improvement may attribute to the nonlinearity of kernel functions. Moreover, Todorovic and Ahuja [43] showed that the performance of model can be further improved by employing ensemble learning method over multiple descriptors. Their method achieves 49.5% of accuracy on Caltech256 by fusing six different descriptors.

4.6. Object and Face Recognition

This section investigates the performance of LrrSPM on two object image data sets (i.e., COIL20 and COIL100) and one facial image database (i.e., Extended Yale Database B). To analyze the time costs of the examined methods, we also report the time costs of the methods for encoding and

Table 7: Object image classification results of different methods on the COIL20 database with different training samples.

Algorithms	Training Images for Each Subject				
	10	20	30	40	50
LrrSPM (Ours)	**97.90±0.42**	**99.52±0.87**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
LinearBOF	87.55 ± 0.17	94.08 ± 0.94	96.65 ± 0.42	97.38 ± 0.26	98.46 ± 0.84
KernelBOF	86.47 ± 0.27	95.60 ± 1.62	97.43 ± 1.10	98.41 ± 1.24	98.46 ± 1.09
linearSPM	85.00 ± 1.00	92.23 ± 1.85	94.17 ± 1.39	97.09 ± 1.17	97.09 ± 0.57
KernelSPM	86.61 ± 0.76	93.14 ± 2.30	95.41 ± 0.63	98.28 ± 1.56	98.64 ± 1.41
ScSPM	97.09 ± 1.13	98.85 ± 0.98	99.65 ± 0.76	**100.00±0.00**	**100.00±0.00**
LLC	97.17 ± 1.14	99.32 ± 1.00	99.64 ± 0.89	99.84 ± 0.89	**100.00±0.00**

Table 8: Object image classification results of different methods on the COIL100 database with different training samples.

Algorithms	Training Images for Each Subject				
	10	20	30	40	50
LrrSPM (Ours)	**91.19±0.65**	**97.39±0.78**	**99.29±0.21**	**99.87±0.36**	**99.85±0.07**
LinearBOF	84.32 ± 1.15	91.65 ± 0.32	94.76 ± 0.35	95.99 ± 0.42	96.81 ± 0.36
KernelBOF	82.32 ± 1.12	92.77 ± 0.53	94.01 ± 0.75	96.36 ± 0.66	97.20 ± 0.48
linearSPM	84.84 ± 0.64	92.17 ± 0.63	95.30 ± 0.53	96.46 ± 0.28	97.64 ± 0.33
KernelSPM	86.01 ± 0.12	92.62 ± 0.61	96.49 ± 0.98	97.56 ± 0.88	98.29 ± 0.32
ScSPM	90.56 ± 0.34	94.73 ± 0.57	97.62 ± 0.15	98.44 ± 0.10	99.81 ± 0.07
LLC	**91.26±0.42**	96.35 ± 0.65	97.97 ± 0.34	98.49 ± 0.24	99.81 ± 0.19

classifying.

Tables 7– 9 report the recognition rate of the tested approaches on COIL20, COIL100, and Extended Yale B, respectively. In most cases, our method achieves the best results and is followed by ScSPM and LLC. When 50 samples per subject of COIL20 and COIL100 are used for training the classifier, LrrSPM groups all the testing images into the correct categories. On the Extended Yale B, LrrSPM also classifies almost all the samples into the correct categories (the recognition rate is about 99.81%).

Table 10 shows the efficiency of the evaluated methods. One can find that LrrSPM, BOF, and SPM are more efficient than ScSPM and LLC both in the process of encoding and classification. Specifically, the CPU time of LrrSPM is only about 2.35%–3.90% of that of ScSPM and about 5.99%–10.44% of that of LLC.

Table 9: Face image classification results of different methods on the Extended YaleB Database B with different training samples.

Algorithms	Training Images for Each Subject				
	10	20	30	40	50
LrrSPM (Ours)	**87.08±0.41**	**96.03±0.89**	**98.28±0.55**	**99.23±0.81**	**99.81±0.83**
LinearBOF	50.26 ± 1.25	64.13 ± 0.73	70.66 ± 1.28	73.78 ± 0.60	77.21 ± 2.14
KernelBOF	59.59 ± 2.33	63.51 ± 1.27	71.35 ± 1.40	76.25 ± 1.67	83.56 ± 1.98
linearSPM	50.21 ± 3.60	70.05 ± 1.20	80.82 ± 1.95	84.39 ± 0.60	88.68 ± 1.73
KernelSPM	54.49 ± 2.52	71.80 ± 1.32	85.54 ± 3.53	84.84 ± 2.10	88.90 ± 3.10
ScSPM	86.79 ± 0.20	94.22 ± 0.45	98.05 ± 0.44	99.00 ± 0.87	99.57 ± 1.16
LLC	84.79 ± 0.59	95.45 ± 0.64	98.05 ± 0.41	98.98 ± 0.25	99.23 ± 0.93

Table 10: The time costs (seconds) for encoding and classification (including training and testing) of different methods on three image databases. The speed of LrrSPM is 25.67–42.58 times faster than ScSPM and 9.58–16.68 times faster than LLC.

Algorithms	COIL20		COIL100		Extended Yale B	
	Coding	Classification	Coding	Classification	Coding	Classification
LrrSPM	16.54	1.4	43.91	4.49	49.67	2.69
LinearBOF	11.54	0.12	11.53	0.11	59.92	0.26
KernelBOF	11.54	11.54	11.53	0.78	59.92	2.05
linearSPM	12.15	0.17	78.92	2.28	93.38	0.52
KernelSPM	12.15	1.5	78.92	36.79	93.38	8.25
ScSPM	424.48	0.79	1837.03	4.07	2114.88	3.08
LLC	275.94	3.86	432.2	6.34	475.94	3.86

5. Conclusion

In this paper, we proposed a spatial pyramid matching method which is based on the lowest rank representation (LRR) of the SIFT descriptors. The proposed method, named as LrrSPM, formulates the quantization of the SIFT descriptors as a rank minimization problem and utilizes the multiple-scale representation to characterize the statistical information of the image. LrrSPM is very efficient in computation while still maintaining a competitive accuracy on a range of data sets. In general, LrrSPM is 25–50 times faster than ScSPM and 5–16 times faster than LLC. Experimental results based on several well-known data sets show the good performance of LrrSPM.

Each approach has its own advantages and disadvantages. LrrSPM is based on the low rank assumption of data space. If this assumption is unsatisfied, the performance of our method may be degraded. Moreover, although

LrrSPM performs comparable to ScSPM and LLC with significant speedup, its performance can be further improved by referring to the recently-proposed methods. By referring to [17], one can develop the nonlinear LrrSPM by incorporating kernel function into our objective function. By referring to [18], one can utilize the label information to design supervised LrrSPM. Moreover, the performance of LrrSPM can also be improved by fusing multiple descriptors as [43] does.

Acknowledgement

The authors would like to thank the anonymous editors ad reviewers for their valuable comments and suggestions to improve the quality of this paper. This work was supported by National Nature Science Foundation of China under grant No.61432012.

Reference

[1] O. Duchenne, A. Joulin, J. Ponce, A graph-matching kernel for object categorization, in: Proc. of IEEE International Conference on Computer Vision, 2011, pp. 1792–1799.

[2] X. Du, J. J.-Y. Wang, Support image set machine: Jointly learning representation and classifier for image set classification, Knowledge-Based Systems 78 (2015) 51–58.

[3] N. Acosta-Mendoza, A. Gago-Alonso, J. E. Medina-Pagola, Frequent approximate subgraphs as features for graph-based image classification, Knowledge-Based Systems 27 (2012) 381–392.

[4] J. Lu, G. Wang, P. Moulin, Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning, in: Proc. of IEEE International Conference on Computer Vision, 2013, pp. 329–336.

[5] L. Yang, S. Yang, S. Li, R. Zhang, F. Liu, L. Jiao, Coupled compressed sensing inspired sparse spatial-spectral {LSSVM} for hyperspectral image classification, Knowledge-Based Systems 79 (2015) 80 – 89.

[6] T. Joachims, A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization., Tech. rep., DTIC Document (1996).

[7] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[8] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: Proc. of IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.

[9] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Proc. of workshop on statistical learning in computer vision, ECCV, Vol. 1, 2004, pp. 1–2.

[10] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2005, pp. 524–531.

[11] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, in: Proc. of IEEE International Conference on Computer Vision, Vol. 2, 2005, pp. 1458–1465.

[12] A. Bolovinou, I. Pratikakis, S. Perantonis, Bag of spatio-visual words for context inference in scene classification, Pattern Recognition 46 (3) (2013) 1039–1053.

[13] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

[14] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Proc. of Advances in neural information processing systems, 2009, pp. 2223–2231.

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[16] K. Yu, Y. Lin, J. Lafferty, Learning image representations from the pixel level via hierarchical sparse coding, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1713–1720.

[17] S. Gao, I. W.-H. Tsang, L.-T. Chia, Sparse representation with kernels, IEEE Transactions on Image Processing 22 (2) (2013) 423–434.

[18] L. Zhou, Z. Zhou, D. Hu, Scene classification using a multi-resolution bag-of-features model, Pattern Recognition 46 (1) (2013) 424–433.

[19] S. K. Nayar, S. A. Nene, H. Murase, Columbia object image library (coil 100), Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96.

[20] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 594–611.

[21] J. Hur, H. Lim, C. Park, S. C. Ahn, Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1392–1400.

[22] L. Zhang, Y. Gao, Y. Xia, Q. Dai, X. Li, A fine-grained image categorization system by cellet-encoded spatial pyramid modeling, IEEE Transactions on Industrial Electronics 62 (1) (2015) 564–571.

[23] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proc. of European Conference on Computer Vision, Springer, 2014, pp. 346–361.

[24] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[25] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[26] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 171–184.

[27] P. Favaro, R. Vidal, A. Ravichandran, A closed form solution to robust subspace estimation and clustering, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1801–1807.

[28] S. Xiao, M. Tan, D. Xu, Weighted block-sparse low rank representation for face clustering in videos, in: Proc. of European Conference on Computer Vision, 2014, pp. 123–138.

[29] J. Yang, Y. Li, Y. Tian, L. Duan, W. Gao, Group-sensitive multiple kernel learning for object categorization, in: Proc. of IEEE International Conference on Computer Vision, 2009, pp. 436–443.

[30] S. Yang, Z. Feng, Y. Ren, H. Liu, L. Jiao, Semi-supervised classification via kernel low-rank representation graph, Knowledge-Based Systems 69 (2014) 150–158.

[31] G. Liu, S. Yan, Latent low-rank representation for subspace segmentation and feature extraction, in: Proc. of IEEE International Conference on Computer Vision, 2011, pp. 1615–1622.

[32] X. Peng, C. Lu, Z. Yi, H. Tang, Connections between nuclear norm and frobenius norm based representation, arXiv:1502.07423v1.

[33] X. Peng, Z. Yi, H. Tang, Robust subspace clustering via thresholding ridge regression, in: AAAI Conference on Artificial Intelligence (AAAI), AAAI, 2015, pp. 3827–3833.

[34] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[35] S. Gao, I.-H. Tsang, Y. Ma, Learning category-specific dictionary and shared dictionary for fine-grained image categorization, IEEE Transactions on Image Processing 23 (2) (2014) 623–634.

[36] M.-E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1447–1454.

[37] S. A. Nene, S. K. Nayar, H. Murase, et al., Columbia object image library (coil-20), Tech. rep., Technical Report CUCS-005-96 (1996).

[38] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International journal of computer vision 42 (3) (2001) 145–175.

[39] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset.

[40] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proc. of Annual International Conference on Machine Learning, ACM, 2009, pp. 609–616.

[41] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, Y. L. Cun, Learning convolutional feature hierarchies for visual recognition, in: Proc. of Advances in neural information processing systems, 2010, pp. 1090–1098.

[42] L. Zhang, X. Zhen, L. Shao, Learning object-to-class kernels for scene classification, IEEE Transactions on Image Processing 23 (8) (2014) 3241–3253.

[43] S. Todorovic, N. Ahuja, Learning subcategory relevances for category recognition, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.