Discriminative Sparse Neighbor Coding

Xiao Bai, Cheng Yan, Peng Ren, Lu Bai, Jun Zhou

***Abstract*—Sparse coding has received increasing attention in recent years due to its state-of-the-art performance in image classification. Traditional sparse coding strategies tend to approximately reconstruct each local feature in terms of a linear combination of a set of basis vectors, without considering feature neighboring relationships. In this scenario, similar instances in the feature space may result in totally different sparse codes. To address this problem, we propose to explore the distribution of similar features over basis vectors to get more descriptive feature representations. The new feature representations thus obtained are expected to be discriminative and class-specific. To this end, we present a discriminative sparse neighbor coding scheme. We commence by developing two modules to enhance the discrimination of sparse representation, i.e., eliminating the non-discriminative features for each specific class, and eliminating the non-informative visual words. We then explore the distribution of similar features over the dominant basis vectors for each class. Specifically, we incorporate the feature distribution into the objective function as the coefficients for sparse coding. This results in a low dimensional subspace within-class for more effective feature representation. Extensive experiments on various image classification tasks show that the proposed approach consistently outperforms several state-of-the-art methods.**

I. Introduction

Image representation is a fundamental problem in computer vision, which has attracted enormous attention in recent years. One of the most popular image coding methods is the bag-of-words (BoW) model which converts the image into a histogram-based representation. The BoW model shows its outstanding performance, especially its robustness to spatial variations [1], [2]. The process of BoW model is normally composed of two main steps: (i) dictionary generation and quantization of the local features which are extracted from the images [3]; (ii) feature pooling in image level, such as max pooling and sum pooling. Recently, sparse coding techniques have been used and achieved state-of-the-art performance in many applications such as object detection [4], tracking [5], image classification [6], [7] and face recognition [8].

In the BoW model, each image is presented as a histogram and each bin of the histogram is the occurrence number of its corresponding visual word. When sparse coding is applied, each feature is represented as a linear combination of a number of basis vectors. Sparse coding extracts global features from signals, which are flexible in the signal representation and data compression. To obtain sparse code, some methods compute the dictionary and histogram-based representation separately [9], and some others manage to learn the optimal dictionary and coding parameters for local features simultaneously [2]. In order to reduce the computational complexity of sparse coding, Wang et al. [9] used the k-nearest bases to encode each feature. Gao et al. [7] added the Laplacian term in the optimization of sparse coding to guarantee that the sparse code changes smoothly on the data manifold. However, all these methods have ignored the distribution of local features over the basis vectors. Such distribution is important in effectively reflecting the relationship between similar features. It may avoid the case that similar local features in the Euclidean space turn out to be different in the sparse representation [7].

The motivation of our work is that we want to explore the useful information of local feature distribution and integrate it into the objective function. Specifically, the aims of our work are two-fold: (i) exploring class-specific similar features to increase the discriminative capability of image representations for different classes, and (ii) learning more informative dictionaries. Most existing methods related to our first aim tend to search the similar features from the whole training set. This mixes up features from foreground and background, and also reduces the discrimination of the sparse code [9], [7]. On the other hand, the normal strategies related to our second aim are to learn a discriminative dictionary for each class and then assign each test image to its predicted class by minimizing the information loss between image representation and classes [10], [11]. Chiang et al. [12] learned a component-level dictionary in each image group which exploited group characteristics to derive the sparse code. Lazebnik et al. [10] learned discriminative visual vocabularies by joining the features and posterior distributions for each class. However, such strategies are not optimal in the label prediction [13].

To overcome the shortcomings described above, we propose a discriminative sparse neighbor coding method. Firstly, to boost the discrimination of the sparse codes, we develop two modules in the sparse coding process: (i) eliminating the non-discriminative features for each specific class; (ii) eliminating the non-informative visual words. Module (i) is also a feature selection process which keeps the class-relevant features and highlights the high-level class knowledge of images. Then, in the coding stage, for each feature, its discriminative neighbors will be selected. The frequencies of the local features and their neighbors over the dictionary will be calculated and integrated into the objective function. Such scheme is useful for feature coding because local features are likely to have common neighboring visual words if they are close in the Euclidean space.

The contributions of this paper are three-fold. Firstly, we employ an iterative method to eliminate non-discriminative features in each class. This is to address the problem that

X. Bai and Cheng Yan are with School of Computer Science and Engineering, Beihang University, Beijing 100191, China. (e-mail: baixiao@buaa.edu.cn)

P. Ren is with College of Information and Control Engineering, China University of Petroleum (Huadong), Qingdao 266580, China

L. Bai is with School of Information, Central University of Finance and Economics, Beijing 10081, China.

J. Zhou is with School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia.

class-irrelevant features in each class may reduce the accuracy of the neighbor information. Secondly, we adopt a statistical model to eliminate the non-informative visual words which not only are ineffective in representing the content of image but also degrade the coding discriminative capability. Finally, to characterize the relationship between local features and classes, we propose a coding method called sparse neighbor coding. We calculate the dominant basis vectors for each class and use the neighbor features to get the frequency distribution over the basis vectors in each class, which leads to more discriminative sparse code.

In the experiments, we demonstrate the benefit of the proposed method for image classification on several publicly available datasets. The performance of individual components of our framework is also verified in the experiments.

The remainder of this paper is organized as follows. Section II reviews related works on sparse coding and presents the overview of the proposed method. Section III presents the details on feature selection and visual words elimination. The proposed sparse neighbor coding method is described in Section IV. Section V reports the experimental results that validate the effectiveness of the proposed method. Section VI summarizes the key contributions of this paper and discusses the further work.

II. Related Work and Overview of the Proposed Approach

A. Related Work

Bag-of-words (BoW) model has proved to be very useful in image coding. In the hard-assignment coding scheme, each coding coefficient vector has only one non-zero element that indicates which cluster each feature belongs to. Since such restriction may cause severe information loss, soft-assignment coding method [14] has been proposed to relax the constraint and computes coding coefficients on all visual words based on their distances to the local feature. Moreover, to cope with the loss of spatial information caused by the BoW model, Lazebnik et al. [1] introduced a spatial pyramid matching (SPM) model to derive the image representation from the spatial perspective.

Recently, sparse coding strategies have shown effectiveness in feature representation. Given an input data matrix D and the signal x to be encoded, sparse coding aims to find a linear combination of a few basis vectors from the D to reconstruct signal x. Yang et al. [15] combined the sparse coding with SPM model and achieved state-of-the-art performance for image classification task.

The transformation from a feature vector to its sparse representation causes information loss. To cope with the information loss in the sparse coding, several techniques make use of the relationships among features to get better sparse representations. Wang et al. [9] suggested that locality plays a more significant role than sparsity in sparse coding and proposed an approximation solution to obtain the sparse code with only k nearest basis vectors. Lu et al. [16] proposed a method which preserves the incoherence of dictionary entries based on the non-local self-similarity and manifold learning. Zhang et al. [17] developed a graph regularized sparse coding method by taking into account of the local manifold structure of the data. The manifold structure has also been combined with random walk model to find the nearest neighbors of encoded feature to boost the representation of encoded code [18]. Comparing with methods that encode feature separately, these methods can preserve the similarity relations for different features.

A number of researchers focus on group sparse coding which encodes similar features into similar sparse codes by learning a common dictionary over multiple different groups of data [19], [20], [15]. In group sparse coding, ℓ_1/ℓ_2 replaces ℓ_1 norm in the sparse coding formulation. Julien et al. [19] acquired the sparse codes with respect to a subset of dictionary by jointly decomposing groups of similar signals. As a consequence, the similarity between features can be maintained. Mosci et al. [21] proposed an efficient optimization procedure for computing the solution of group lasso with overlapping groups of variables.

To obtain the discriminative sparse representation, some researchers focus on finding an optimal dictionary that leads to the lowest reconstruction loss with a set of sparse coefficients. In this context, dictionaries are learned for each classes. In [22], [23], each patch of the test image is approximated with respect to a set of dictionaries in different classes. Then the image class is predicted by calculating the residual errors in different classes. Julien et al. [24] proposed an online learning method to deal with large datasets with millions of training samples. This method can effectively handle the problem of high computation complexity when the training set is large. Liu et al. [25] showed the importance of non-negativity property and discriminating capability in the sparse representation.

Before the coding stage, several methods are used to guarantee the discriminative property of the dictionary and image representation. Some approaches focus on selecting the useful local features for training. For instance, Turcot et al. [26] proposed a match-based method to augment the feature representation based on a graph model, which only keeps the useful features. In [27], a pairwise image matching method was presented to select discriminative foreground features. On the other hand, some researchers [28], [29], [30] paid more attention to removing the noise visual words. Sivic et al. [28] considered the frequencies of visual words occurring in images, which is borrowed from the text retrieval technique. Tirilly et al. [29] proposed a method to eliminate useless visual words based on the geometric properties of the local features and probabilistic latent semantic analysis (pLSA).

The literature reviewed above focuses on the different aspects in the process of feature coding, such as feature selection and dictionary learning. The aim of these methods is to reduce the information loss of sparse coding and boost the effectiveness of image presentation. Different from above sparse coding methods, we weight the dominant basis vectors by using the frequency distribution of similar local features. Our method explores the class-specific subspace for encoding local features, preserving the similarity of the local features after sparse coding.

B. Overview of the Proposed Approach

In this paper, we propose a discriminative sparse neighbor coding method. We use the frequency distribution of the similar features over the basis vectors in the coding stage, and retain the similarity between local features. In order to keep the discriminative features in each class and eliminate the non-informative visual words, we develop two modules to boost the discrimination of the sparse code.

In detail, the proposed method comprises the following steps:

1) *Discriminative feature selection*: An image matching based feature selection method is employed to select the discriminative class-specific features from each image.

2) *Non-informative visual words elimination*: A statistical method is utilized to automatically discover the non-informative visual words and eliminate them to strengthen the discriminative power of the visual words.

3) *Neighbors searching*:

Find the similar features (i.e. *neighbors*) in each class for the given local feature through offline strategies.

4) *Sparse neighbor coding*: The distribution of the neighbors of given local feature over the basis vectors is calculated. Such distribution is formulated as weighted coefficients which are integrated with the dominant basis vectors in each class into the objective function to obtain the sparse neighbor code.

Following the sparse coding stage, max pooling and SPM are used to compute the image-level representation. Then one-vs-rest classifier is employed for image classification. The framework of the proposed method is illustrated in Figure 1.

III. Discriminative Feature and Visual Word Selection

Neighborhood information is helpful in encoding local features. The class-irrelevant (i.e. the features in the cluttered background) features in each class reduce the performance of encoding. Therefore, we aim to detect and eliminate these class-irrelevant features in each class to boost the representation of sparse code. Furthermore, some of the generated visual words may not be useful to represent visual contents. Hence, these visual words need to be eliminated, which also can reduce the size of dictionary and computation cost in the following sparse coding phase. To achieve these goals, we introduce a method based on image matching to highlight the class-specific features. Furthermore, a statistical model is also adopted to eliminate the non-informative visual words.

A. Discriminative Feature Selection

The similarity between features is important for sparse representation.

Some strategies tend to integrate the information of neighbors in the objective function to encode each local feature [7], [17]. But the features, either from irrelevant objects or from background, may reduce the performance of these strategies. For example, for coding local features supposed to locate on the surface of an object, the performance of sparse coding will decline if their neighbors from the cluttered background area are treated as object features in training. As illustrated in Figure. 2, the searched neighbors may come from the background area. Although they are similar to the encoded feature in the feature space, they are not visually relevant. This confusion thereby reduces the performance of feature coding stage. Therefore, if these features within the specific class can be detected and eliminated, the encoded sparse codes will be more discriminative.

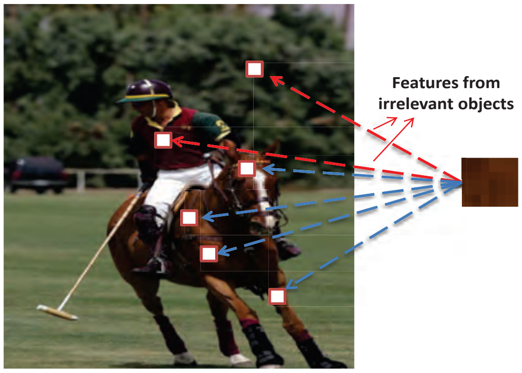

Fig. 2: Features from irrelevant objects.

We propose an image matching based iterative strategy to select the discriminative features [31]. This method is based on Earth Mover's Distance (EMD) [32], which finds the optimal correspondences between features and can be used for computing the similarity between images. We adopt this model because it can be properly integrated into our feature selection model. The discriminative features can be shared by images from the same class but not those from different classes. The EMD measure strategy not only computes the distance between two images, but also characterizes the feature matching contribution, which can be used to update the weight attached to each feature.

Suppose $F = \{(f_1, w_1), \dots, (f_{|F|}, w_{|F|})\}$ is the set of local features extracted from image I, where $|F|$ is the number of local features, f_i is the local feature and w_i is its corresponding weight. Initially, each w_i is set as 1 and it is then updated based on its contribution to the image matching process. Given two images I_p and I_q, the EMD distance is defined as

$$\begin{aligned} EMD(I_p, I_q) &= (\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij}) / (\sum_{i,j} f_{ij}) \\ s.t \quad f_{ij} &\geq 0, \sum_j f_{ij} \leq w_i, \sum_i f_{ij} \leq w_j \\ \sum_{i,j} f_{ij} &= \min(\sum_i w_i, \sum_j w_j) \end{aligned} \tag{1}$$

where $\{f_{ij}\}$ is the flow matrix and each f_{ij} denotes the flow between features f_i and f_j. $\{d_{ij}\}$ is the threshold distance matrix and each element d_{ij} is defined as $d_{ij} = \min(d(i,j), t)$, where $d(i,j)$ is calculated by using Euclidean metric between features f_i and f_j. The parameter t controls the speed of the EMD computation and we set $t = 10$ in our work.

Then the weight of each local feature is updated on the basis of feature matching during the EMD calculation. The

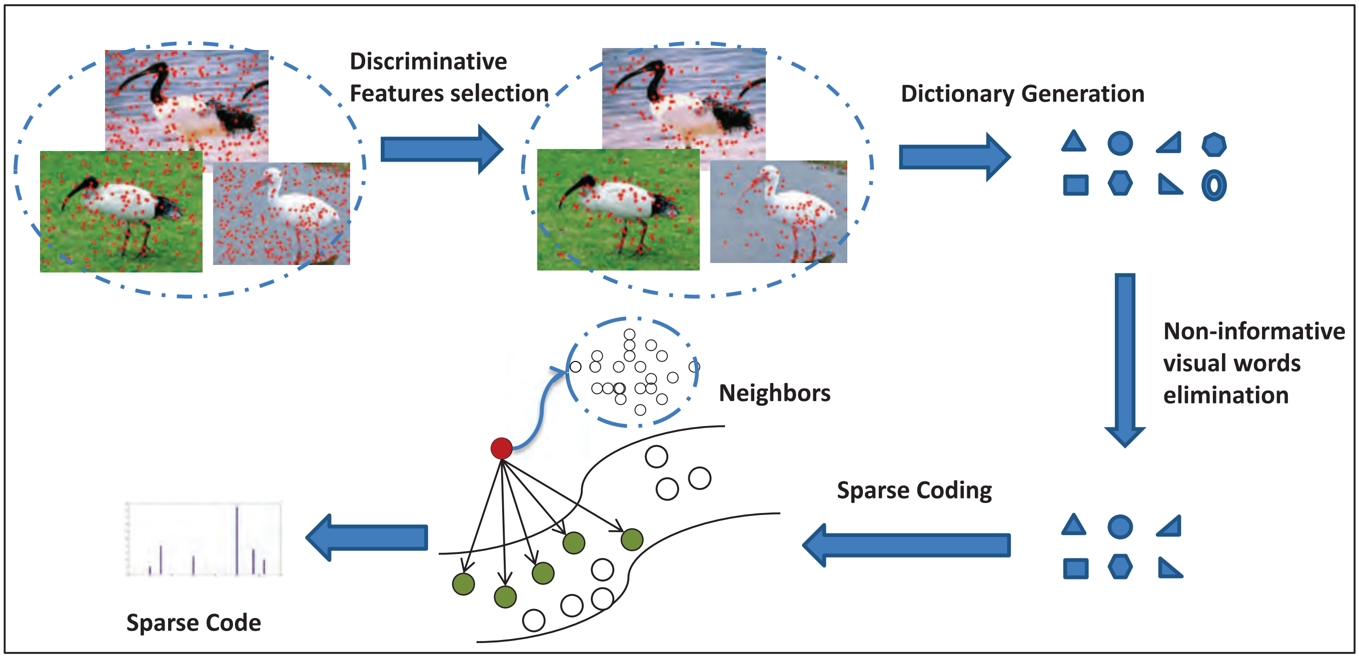

Fig. 1: An overview of the proposed method.

contribution of f_i of image I_q is calculated as

$$c_q(i) = \sum_j f_{ij} \times \delta_j / d_{ij} \tag{2}$$

The term $\delta_j = \frac{|I_q| \times w_j}{\sum_{k=1}^{|I_q|} w_k}$ is a normalizing factor, where $|I_q|$ is the number of local features in image I_q. The weight of feature f_i is updated using all related contributions in a class. Specifically, the weight of feature f_i is reassigned with

$$w_i = \frac{1}{M-1} \sum_{q=1}^{M-1} c_q(i) \tag{3}$$

where M is the number of images in the class. In this way, the class-specific local features with strong matches across all images in the same class are selected.

The pairwise matching and feature weight update steps are performed iteratively to highlight the discriminative features in each class. Initially, the weight of each feature is set to an equal value, i.e., 1, and EMD method is used to compute the matrices $\{f_{ij}\}$ and $\{d_{ij}\}$. Then each weight is updated according to Eq.(2) and Eq.(3). The stopping criterion for this iterative updating procedure is the separability of the training setthe details of which can be found in [31].

The non-discriminative features can be eliminated by using the above method. We thus obtain one more effective similar features searching step and the sparse representation becomes more robust.

B. Non-informative Visual Words Elimination

Our motivation for non-informative visual words elimination is from noisy word elimination in text documents, in which noisy words sometimes occur frequently and influence the text categorization. The noisy words, e.g. in, of, on, if, the, are also called *stop words* in text processing [33], [34]. In compute vision, there also are non-informative visual words that are not useful in image classification and retrieval.

In sparse coding, traditionally, the basis vector visual words are usually obtained by clustering algorithms, thus the semantic information of the visual words can not be predefined. In this paper, we utilize the Chi-square model [34] to find the non-informative visual words based on the relationship between visual words and image classes. A visual word is

TABLE I: The contingency table of visual word v_i

	c_1	c_2	$\dots$	c_m	Total
v_i	n_{11}	n_{12}	$\dots$	n_{1m}	n_{1+}
$\bar{v_i}$	n_{21}	n_{22}	$\dots$	n_{2m}	n_{2+}
Total	n_{+1}	n_{+2}	$\dots$	n_{+m}	N

considered as non-informative if it satisfies the following two conditions:

- It has high frequency in many images. Because one visual word cannot present any specific image or object if it exists in many images.
- It has small statistical correlations with all the classes. The non-informative visual word cannot characterize the relation between visual word and class, which will reduce the discriminative ability of the final encoded feature representation.

Suppose the dictionary $D' = \{v_1, v_2, \dots, v_{K'}\}$ $(K' \geq 1)$ is generated based on the selected features obtained in the last step and C is the total number of classes. The relation between visual word v_i and class is shown in Table I.

In the contingency table, the meanings of the items are described as follows:

- n_{1j} denotes the number of images containing visual word v_i in class c_j;
- n_{2j} denotes the number of images which do not contain visual word v_i in class c_j;
- n_{+j} denotes the total number of images in class c_j;
- n_{i+} denotes the total number of images containing visual word v_i in training set
- N denotes the number of total training images.

The independence between visual word v_i and all classes is computed using following weighted Chi-square statistics

$$\chi^{(i)}_{weighted}{}^{2} = \chi^{(i)^2} / IF_{v_i} \tag{4}$$

where

$$\chi^{(i)^2} = \sum_{i=1}^{2} \sum_{j=1}^{K'} \frac{(Nn_{ij} - n_{i+}n_{+j})^2}{Nn_{i+}n_{+j}} \tag{5}$$

In Eq. (5), $\chi^{(i)^2}$ denotes the association between visual word and class. The smaller it is, the weaker it is correlated with the classes. The term IF_{v_i} in Eq. 4 denotes the occurrence frequency of visual word v_i in the images, which is a trade-off factor. This factor balances the relationship between the visual word in each class and frequency of visual word in the images. Consequently, all visual words are listed in a descending order according to the value of weighted Chi-squared statistics. Those visual words with high values will be chosen if they are above a given threshold determined by cross-validation [35]. In the experiments we obtain the threshold by leave-one-out cross-validation on the training set for each trial and choose the one which leads to the best classification accuracy.

IV. Sparse Neighbor Coding

In this section, we describe the sparse neighbor coding method which converts low-level feature into sparse code. Each class has a potential low-dimensional linear subspace that can be used to approximately construct sparse codes. Our contribution comes from the consideration of feature frequency distribution information which has been ignored in existing sparse coding methods [13], [9]. We propose to incorporate the neighbor information in the optimization to obtain the discriminative sparse code. Moreover, instead of computing a set of basis vectors for each class and predicting the label based on the residual error, we weigh each basis vector by calculating its importance to each class.

A. Dominant Basis Vector Learning

In image representation, data samples belonging to the same class tend to lie in the same low-dimensional subspace. This means that a new sample can be reconstructed with lower computation load by using only a few basis vectors (atoms) in its corresponding class.

In the light of this observation, we commence by finding the dominant basis vectors, which have high relevance to each corresponding class. These basis vectors can be used to construct a more discriminate sparse code for each local feature. To this end, we start from finding the basis vectors with less reconstruction errors for each class.

Suppose $D \in R^{d \times K}$ is the dictionary which non-informative visual words have been eliminated. Each column in D represents a basis vector. To encode each feature x_i, we use the sparse coding with ℓ_1 norm. Sparse coding ameliorates the quantization loss of hard vector quantization (VQ). In VQ method, only the closest basic vector is active. However, sparse coding relaxes this constraint by using a sparsity regularization term, which can be formulated as follows

$$\arg\min_{z_i} \|x_i - Dz_i\|_2^2 + \lambda \|z_i\|_1 \tag{6}$$

where λ is the constraint that makes the trade-off between reconstruction error and sparsity of coefficients. This convex problem can be solved efficiently by Sparse Modeling Library (SPAMS) [36].

Because of the sparsity of coefficient z_i, only a few basis vectors are active to represent feature x_i. Let $Z = [z_1, z_2, \dots, z_n]$ be the sparse code for the images in class c, we define the significance of each basis vector v_j by computing the sum of response among these samples:

$$s_j^{(c)} = \frac{\sum_{i=1}^{n} |z_{ij}|}{\sum_{k=1}^{K} \sum_{i=1}^{n} |z_{ik}|} \tag{7}$$

Each $s_j^{(c)}$ indicates its significance to the class c. n is the number of image in class c and K is the class number. z_{ij} is the jth dimensional coefficient for ith sparse code for class c. The activated visual words in sparse representation are mainly in the same sub-space with low-level feature vectors in the same class. Hence, we force the nonzero coefficients to lie in subset of dictionary D, and ignore the other basis vectors with less significance. To this end, we set the weight of each basis vector for class c as

$$s_j^{(c)} = \begin{cases} s_j^{(c)}, & s_j^{(c)} \leq T^{(c)} \\ 0, & s_j^{(c)} > T^{(c)} \end{cases} \tag{8}$$

where $T^{(c)} = \beta \times \sum_j s_j^{(c)}/K$ is a threshold. β is empirically set to 0.3, which ensures that the most significant coefficients are kept. These basis vectors with non-zero weights form the class-specific dictionary for each class, which are denoted as $D^{(c)}$. Then $s^{(c)}$ is normalized into the range $[0, 1]$. The more dominant a basis vector is, the larger its correspondence significance value $s^{(c)}$ is. We introduce how to utilize the dominant visual words to effectively encode each local feature in section IV-C.

B. Neighbor Searching

One problem in sparse coding based methods is that local features similar in the feature space may be quantized into different visual words. In order to preserve their similarity, we capture the correlations between similar features and exploit the distribution of these similar features over the visual words to help encode each feature.

In this section, we introduce a graph-based method to find the similar features while simultaneously keeping the accuracy and efficiency. Then we describe how to use the similar features to obtain the sparse code in the next section.

To find similar features, we utilize the minimum dominating set (MDS) [37], which is a graph model. Consider an undirected graph $G(V, E)$ where V denotes the set of vertices and $E \subseteq V \times V$ denotes the set of edges. In the graph, the vertices represent local features and the edges describe how similar two adjacent features are. The dissimilarity between two local features x and y is measured in terms of the Euclidean distance $d_E(x, y) = \|x - y\|^2$. During the graph construction, edges whose weights are greater than a chosen threshold are discarded.

For a graph $G(V, E)$, one vertex $\alpha \in V$ is thought of being covered by a set of vertices if either of the two conditions are satisfied: (i) α is in the set, or (ii) α is adjacent (i.e. a neighbour) to a vertex in the set. For $G(V, E)$, one vertex subset $S \subseteq V$ is a dominating set if S covers all the vertices in V. For a vertex $\alpha \in V$ in G, α and its adjacent vertices form a subgraph. Each subgraph contains a vertex in S and has high similarity between adjacent vertices since we have discarded some dissimilar edges in the process of graph construction. This graph will be used to find the similar features (*neighbors*). To make the searching stage more efficient, the size of S should be as small as possible. Therefore, we use the minimum dominating set, which has minimum size of S.

Given a feature x_i, it is compared with the vertices in set S. The top vertex which shows high similarity with x_i is selected as the neighbor of x_i. Then the features corresponding to the selected vertices are selected.

Minimum dominating set model is effective since the vertices within a specific subgraph have great similarity. To compute the minimum dominating set, we exploit a simple greedy algorithm to obtain an approximate solution [38] For each class, constructing the graph model requires $O(n^2m)$ operations, where n is the number of local features and m is the dimension of each feature. In addition, the time complexity of the approximate algorithm for obtaining minimum dominating set is $O(e)$, where e is the number of edges in G and $e < n^2m$. This searching operation requires $O(mlogp)$, where p is the size of S. To balance the time complexity and the performance of our method, we select 1000 features to construct the minimum dominating set, which are obtained through clustering.

In the rest of this paper, we refer to the set containing neighbors as neighbor set.

C. Formulation

In Section IV-A, we obtain the low dimensional subspace for each class c, which is represented as a subset of the dictionary $D^{(c)}$ and it contains $K^{(c)}$ visual words. Furthermore, each visual word has a weight $w_j^{(c)}$ to denote its significance. Computing the sparse code of the local feature in class c based on the dictionary $D^{(c)}$ will lead to a class-specific sparse code. However, the similarity of the local features may be lost since the sparse coding approach may select diverse basis vectors for similar features, which reduces the performance of the sparse code. To preserve the similarity during sparse coding phase, we use the neighbor set (see Section IV-B) in each class to help encode the feature.

Given a feature x_i, suppose its corresponding neighbor set for class c is $NS_i^{(c)}$. We compute the frequency distribution of neighbor set $NS_i^{(c)}$ over the dictionary $D^{(c)}$ based on Euclidean distance. Each neighbor is mapped to its closest visual words in $D^{(c)}$. Then the frequency distribution on the $D^{(c)}$ is calculated as

$$\epsilon_{ip}^{(c)} = \sum_j f(v_p, x_j) \tag{9}$$

with

$$f(v_p, x_j) = \begin{cases} 1, & \text{if } x_j \text{ is closest to } v_p \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where v_p is the visual word in the $D^{(c)}$ and x_j is the feature in the neighbor set $NS_i^{(c)}$. Based on this formulation, the relation between local features can be described. If the neighbors of feature x_i locate mostly in a few specific visual word, the given feature x_i will have high response to these visual words (see Figure. 3).

Then coding with class sub-space and distribution information on basis vectors transforms the normal sparse coding formulation into

$$\begin{aligned} &\arg\min_{z_i} \|x_i - D^{(c)} z_i\|^2 + \gamma \|z_i\|_1 + \beta \|q_i^{(c)} z_i\|^2 \\ &\textit{s.t.} \qquad 1^\top z_i = 1 \end{aligned} \tag{11}$$

The ℓ_1 norm regularization results in the sparsity of the representation. The coefficient $q_i^{(c)} = 1/(\epsilon_i^{(c)} \times s^{(c)})$ integrates the dominant basis vectors with the distribution information, where both $\epsilon_i^{(c)}$ and $s^{(c)}$ are normalized vectors. Eq.(11) controls the coding coefficient vector z_i to achieve the minimization of quantization loss and meets the following properties: (i) the value of the coefficient z_{ij} is larger if there are a large portion of neighbors locating on the j-th basis vector, thus preserving the similar response on the basis vectors for

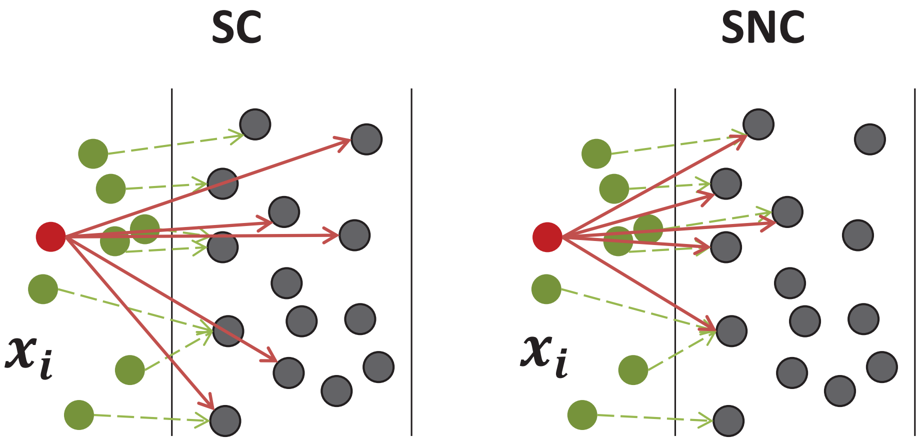

Fig. 3: Left: Traditional Sparse Coding method; Right: Our method. The sparse coding selects different basis vectors to encode the similar features. Our method encodes each feature together with its neighbor distribution on the basis vectors, which enables feature similarities to be preserved in sparse representation.

similar features; (ii) similar features are encoded based on similar basis vectors, therefore the neighboring local feature distribution enables similar responses over basis vectors for similar features. In this way, if two features are close in the feature space, they are likely to relate to the similar visual words and thus resulting in the similar sparse codes.

Recent studies [9], [7] suggest that construction locality produces better performance on the feature coding. Thus we can also use the k most similar basis vectors to encode each feature. The locality guarantees the sparsity, and the ℓ_1 term in Eq.(11) can thus being ignored. Only k basis vectors are used to construct the feature, which also improves the computation efficiency. To compute the optimal solution to Eq.(11), we initialize the variables in terms of $z_i = D^{-1}x_i$, and then iteratively update z_i based on coordinate descent.

The process of the proposed sparse neighbor coding method is summarized in Alg. 1:

Algorithm 1 Sparse Neighbor Coding

```
Input: The images in dataset and local featurs $\{(\{x_i\}^{(c)}_{i=1,\dots,|I|}, I^{(c)})\}, c = 1, \dots, C$, which are extracted from these images.
1: Feature selection: Set the weights of local features to the same and highlight the discriminative features based on an iterative feature selection algorithm as described in Section III-A.
2: Non-informative visual words elimination: Calculate the weighted Chi-square statistic $\chi^2_{weighted}$ for each visual word and select those with great values as described in Section III-B.
3: Sparse neighbor coding:
   1). Find the dominant basis vectors $D^{(c)}$ for each class as described in Section IV-A.
   2). For each local feature $x_i$, find its neighbors set $NS_i^{(c)}$ in each class as described in Section IV-B.
   3). Compute the distribution of $NS_i^{(c)}$ over dictionary $D^{(c)}$ in each class, and then compute the sparse code by solving the Eq.(11).
```

D. Inference

Given a new test image, we need to calculate its sparse representation for each class c ($c = 1, \dots, C$). Suppose one image region has m local features, maximum pooling is employed to aggregate these features in the same region. Each local feature will be presented as a vector with dictionary size K and the $u_j^{(c)}$ entry is the maximum response to the j-th basis vector

$$u_j^{(c)} = max\{|x_{1j}|, |x_{2j}|, \dots, |x_{mj}|\} \tag{12}$$

To preserve the spatial information, Spatial Pyramid Matching[1] is also employed in our method. Both spatial layout and more basic pattern responses are retained by dividing the whole image into multiple fine regions. Then we apply one-vs-rest SVM classifier to compute the probability $P(C|u)$ that the test image belonging to each class. The classification label is assigned whereby finding the highest probability value

$$c* = \arg\max_{c \in C} P(C = c|u^{(c)}) \tag{13}$$

V. Experiments

In this section, we report experimental results on four widely used datasets: Scene 15 [39], UIUC 8-Sport [40], Caltech-101 [41], PASCAL VOC 2007 [42]. There are several alternative state-of-the-arts methods for comparison in the literature. ScSPM [2] is a sparse coding method that incorporates spatial pyramid matching. KSPM [1] performs spatial pyramid matching and SVM classification using histogram intersection kernel. HIK+OCSVM [43] uses histogram intersection kernel and one class SVM to quantize local feature. LScSPM [7] is a Laplacian sparse coding approach based on spatial pyramid matching. LR-Sc^+SPM [44] performs non-negative sparse coding along with max pooling and spatial pyramid matching. NBNN [45] is a nearest-neighbor approach in local image feature space. LLC is the locality-constrained linear coding method. LR-LGSC[46] is a method that investigates group generation for group sparse coding with Laplacian constraints. Zhang et al. [6] proposed an image representation based on structured low-rank. We compare our method with the above state-of-the-arts methods.

A. Parameters Setting

Local feature descriptor is essential to image representation. In our work, we adopt the widely used 128 dimensional SIFT feature [47]. Dense SIFT features are extracted with step size set to 8 and size of patches set to 16 × 16. The whole images are processed in gray scale. The extracted features are then normalized with ℓ_2-norm. For Scene-15, UIUC 8-Sport and Caltech-101 datasets, we construct the SPM model in three levels, i.e., 1 × 1, 2 × 2 and 4 × 4, as described in [1]. For the PASCAL VOC 2007 dataset, we obtain the spatial regions by dividing the image in 1 × 1, 3 × 1 and 2 × 2 grids, which follows [48]. In the SPM construction, each layer is assigned the same weight. To train the codebook, we utilize the standard k-means clustering method. The codebook size is fixed to 1024. In the classification step, we use one-vs-rest linear SVM [49] provided by Yang et al. [2] due to its advantages in speed and good performance in max pooling based image classification. Following the common benchmarks procedures, we repeat the experiments with randomly selected training and testing samples, and record the average accuracy and the standard deviation.

In addition, there are several parameters to be set in our method. The sparsity of sparse codes λ is fixed at 0.3. The regularization parameter C in linear SVM is set to 10.

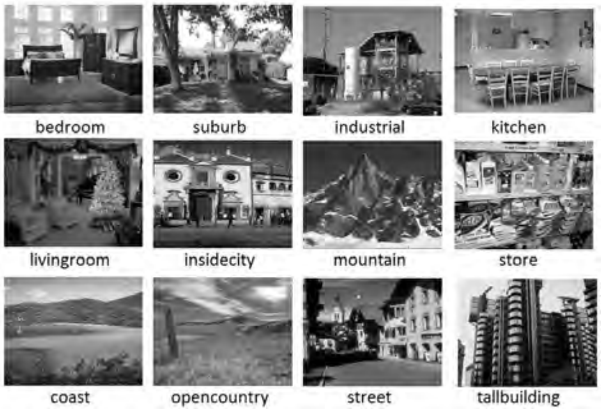

Fig. 4: Example images for the Scene 15 dataset.

B. Scene 15 Dataset

We evaluate our method for scene classification on the Scene 15[1] dataset which contains 4485 images from 15 categories, with category size varying from 200 to 400. The image contents are diversified, containing not only indoor scenes, such as bedrooms and kitchens, but also outdoor scenes, such as buildings and villages. The average image size is 300 × 250 (pixels). In the experiment, we resized the maximum side (length/width) of each image to 300 pixels with aspect ratio remaining unchanged. Figure 4 shows some sample images in this dataset. To compare with alternative methods in the literature, 100 images are randomly selected from each class as the training data and the rest are used as the testing data. The experimental results are listed in Table II with the comparison against several alternative approaches. The confusion matrix for the results for the Scene 15 dataset is shown in Figure. 5.

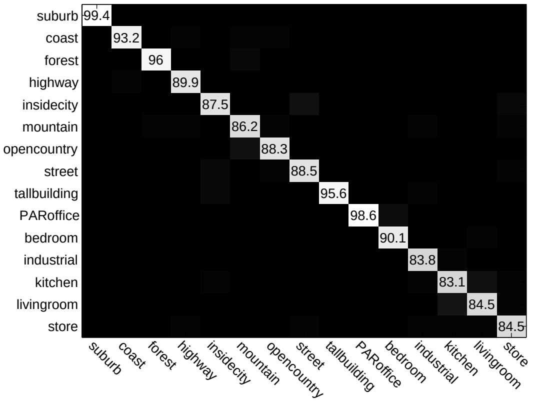

Fig. 5: Confusion matrix on Scene 15 Classification (%). Each entry in the diagonal is the average classification rate for an individual class. The entry in the ith row and jth column is the percentage of images from class i which were misidentified as class j.

Method	Classification Accuracy
KSPM [1]	81.40 ± 0.50
HIK+OCSVM [43]	84.00 ± 0.46
ScSPM [2]	80.28 ± 0.93
LLC [9]	81.53 ± 0.65
LScSPM [7]	89.75 ± 0.50
LR-Sc^+SPM [44]	90.03 ± 0.70
Ours	**89.83 ± 0.74**

TABLE II: Performance Comparison on Scene 15 Dataset (%).

Table II shows that the average accuracy of our method is 89.83%, which outperforms five alternative methods and is close to LR-Sc^+SPM method. However, it should be noticed that LLC and LScSPM use neighborhood data to help the construction of the sparse codes. The results validate the observation that by exploiting the relationship between sparse code and class specific information, the obtained sparse code is more powerful for image representation.

From Figure 5, we observe that the proposed method works well on several scene categories, including suburb, coast, forest, highway, tallbuilding and office. However, the accuracies are relatively low for industrial, kitchen, livingroom, and store classes. The reason for the low accuracy is that the patches in these classes are visually similar with other classes. So it's hard to extract class specific information for further analysis.

[1] http://www-cvr.ai.uiuc.edu/ponce_grp/data/scene_categories/scene_categories.zip

C. UIUC Sport Dataset

UIUC 8-Sport[2] data set was introduced in [40] for image-based event classification. These 8 categories are badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snow boarding. There are 1579 images in total, and the size of each category ranges from 137 to 250. In this data set, the maximum size is set to 400 because its images have higher resolutions. Figure 6 shows some sample images of this dataset. In the experiment, we randomly select 70 images from each class as the training data and the rest as the testing data.

Fig. 6: Example images in the UIUC-Sports dataset.

Method	Classification Accuracy
ScSPM [2]	82.74 ± 1.46
HIK+OCSVM [43]	83.54 ± 1.13
LScSPM [7]	85.31 ± 0.51
LLC [9]	81.77 ± 1.51
LR-Sc^{+}SPM [44]	86.69 ± 1.66
LR-LGSC [46]	87.14 ± 1.52
Ours	**87.13** ± **1.29**

TABLE III: Performance Comparison on UIUC 8-Sport Dataset (%).

Table III gives the performance comparison of the proposed method and several other methods on the UIUC Sport dataset. The proposed sparse neighbor coding method has achieved 87.13%, with 0.44% superiority to LR-Sc^{+}SPM. The confusion matrix for the results on this dataset is shown in Figure. 7.

D. Caltech 101 Data Set

The Caltech-101[3] dataset contains 102 classes with high intra-class appearance shape variability. The number of images per category varies from 31 to 800 images and most of these images are in medium resolution. In the experiment, the images are resized to be less than 300 × 300 with aspect ratio kept. All 102 classes are used in this experiment. Following the standard experimental setting, we used 15 and 30 images per class for training while leaving the remaining for test.

Table IV provides the performance comparison of the proposed method with several alternative methods [43], [50], [2], [9], [44], [6] on the Caltech-101 dataset. Our method has outperformed the listed algorithms, achieving 70.04 ± 0.42 when the training size is 15 per class and 76.96 ± 0.87 when the training size is 30 per class. These results have validated the effectiveness of our method.

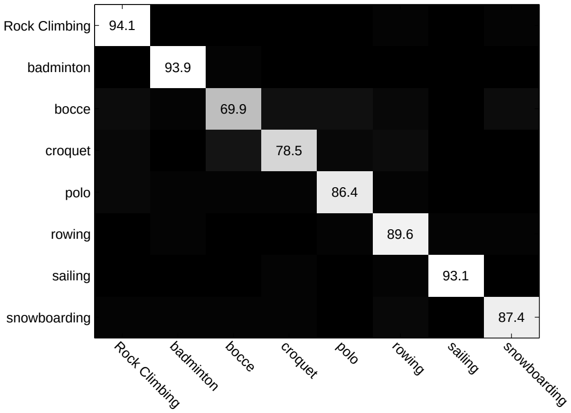

Fig. 7: Confusion matrix on UIUC Sport Classification (%).

Fig. 8: Example images in the Caltech-101 dataset.

Method	Classification Accuracy with 15 training samples	Classification Accuracy with 30 training samples
KSPM [43]	56.40	64.40 ± 0.80
NBNN [50]	65.00 ± 1.14	70.40
ScSPM [2]	67.00 ± 0.45	73.20 ± 0.54
LLC [9]	65.43	73.44
LR-Sc^{+}SPM [44]	69.58 ± 0.97	75.68 ± 0.89
Zhang et al. [6]	66.1	73.6
LR-LGSC [46]	68.15 ± 0.42	76.52 ± 0.47
Ours	**70.04** ± **0.42**	**76.96** ± **0.87**

TABLE IV: Performance Comparison on the Caltech-101 dataset(%).

E. PASCAL VOC 2007 Data Set

This data set consists of 10,000 images from 20 classes, with objects in a variety of scales, locations and viewpoints. In the experiments, 5011 images are used for training and 4952 images for testing by random splitting. The performance measure is the mean average precision (mAP), which is a standard metric used by the PASCAL challenge. It computes the area under the Precision/Recall curve. The higher scores reflect better the performance.

[2] http://vision.stanford.edu/lijiali/event_dataset/

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

In Table V, we list the mAP scores for all 20 categories from different methods. It can be seen that our method has achieved the performance superior to alternative methods on 5 classes: bicyle (68.6%), car (80.3%), cow (50.1%), person (86.2%) and tv (57.7%). The Fisher kernel has obtained the best mAP among the methods with dictionary size 256. This is because it encodes additional information on the distribution of the descriptors. Our method has only 0.5 percent inferiorly than the Fisher kernel method and shows significant improvement than other methods. This result demonstrates the effectiveness of the proposed method.

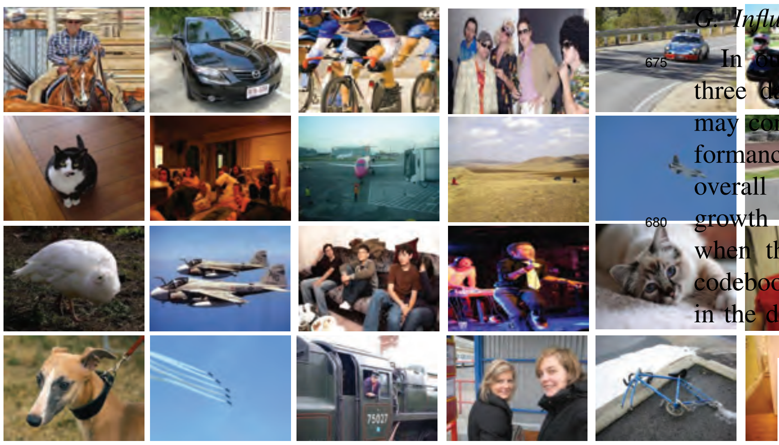

Fig. 9: Example images in the PASCAL VOC 2007 data set.

Method	aero	bicycle	bird	boat	bottle	bus	dining
LLC	74.8	65.2	50.7	70.9	28.7	68.8	51.8
Best'07	77.5	63.6	**56.1**	**71.9**	**33.1**	60.6	54.9
FK	**79.0**	67.4	51.9	70.9	30.8	**72.2**	**58.4**
SV	74.3	63.8	47.0	69.4	29.1	66.5	51.9
Ours	75.4	**68.6**	54.2	71.6	30.2	69.4	56.4

Method	dog	horse	motbike	person	plant	sheep	sofa	train	tv	mAP
LLC	44.1	76.6	66.9	83.5	30.8	44.6	53.4	78.2	53.5	59.3
Best'07	**45.8**	77.5	64.0	85.9	**36.3**	44.7	50.9	79.2	53.2	59.4
FK	44.8	**78.8**	**70.8**	85.0	31.7	**51.0**	**56.4**	**80.2**	57.5	**61.7**
SV	44.1	77.9	67.1	83.1	27.6	48.5	51.1	75.5	52.3	58.2
Ours	44.5	78.3	69.6	**86.2**	33.4	47.3	54.6	78.8	**57.7**	61.2

TABLE V: Comparison of image classification performance in terms of test accuracy on the PASCAL VOC 2007 dataset. **LLC** – locally-constrained linear coding [9]; **FK** – Fisher kernel [51]; **SV** – super vector coding [52].

F. Time Analysis of Feature Coding

Dataset	LLC [9]	LScSPM [7]	ScSPM[2]	LR-Sc$^+$SPM [44]	Ours
UIUC 8	4	17	15	28	16
Scene 15	7	39	37	46	35
Caltech 101	31	158	156	195	172
VOC 07	37	185	187	262	156

TABLE VI: Time compleixty on four datasets in feature coding phase. (min)

From the Table. VI, we can see the numeric time complexity of feature coding on four datasets during testing phase. The number of testing images in the four datasets are 480, 1500, 3030 and 4953, separately. LLC method has the least time in coding the testing images. As the normal setting, we set the number of neighbors k to 5. The time cost of LLC method mostly depends on the kNN searching. In ScSPM, we choose 200 neighbors for each feature to get the sparse code. It costs more time than LLC, but obtains better classification in some datasets. The overall coding time of ScSPM and LScSPM are quite the same. Besides, the time cost of our method is greater than that of LLC method and nearly the same with those of LScSPM and ScSPM.

G. Influence of Codebook Size

In our experiment, we test the classification accuracy on three datasets according to different codebook sizes, which may considerably influence classification results [3]. The performance is illustrated in Fig. 10, from which we can see the overall tendency is that the performance increases with the growth of codebook size. Moreover, the curves grow faster when the codebook size is smaller. This is because small codebooks cannot present the various patches of the images in the dataset.

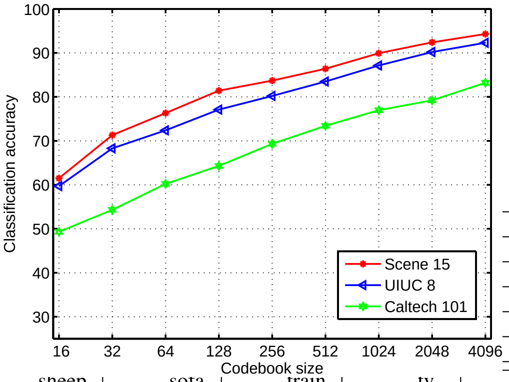

Fig. 10: Classification performance on different codebook size(%).

H. Influence of Individual Components

In this subsection, the importance of each component is tested and the results are shown in Table VII. Here we can see that the proposed sparse neighbor coding performs better than LLC method by 3.03%, 2.99%, 1.42% and 0.5% improvements separately. Besides, by using the discriminative feature selection and visual word selection strategies, the performance are boosted comparing with that of the basic sparse neighbor coding method. Therefore, it is evident that these two modules are effective and lead to better sparse code. And the best results are obtained by combining these three modules.

VI. Conclusion and Future Work

The neighbor information in the feature space is of great importance for image representation. To explore the neighbor information, we have presented a sparse neighbor coding

Method	Scene 15	UIUC 8	Caltech 101	VOC 2007
SNC	84.56 ± 0.83	84.76 ± 0.56	74.86 ± 0.71	59.8
SNC+FS	86.83 ± 0.78	86.28 ± 0.79	75.19 ± 0.63	60.4
SNC+VWS	85.75 ± 0.53	85.37 ± 1.06	75.02 ± 0.71	60.8
SNC+FS+VWS	**89.83 ± 0.74**	**87.13 ± 1.29**	**76.96 ± 0.87**	**61.2**

TABLE VII: Classification performance by combining different component. VOC 2007 dataset is evaluated by mAP and others are evaluated by classification accuracy. **SNC** – sparse neighbor coding; **FS** – feature selection; **VWS** – visual word selection.

method. We have developed two modules, which are used to keep the discriminative feature in each class and eliminate the non-informative visual words, to boost the discrimination of the resulted sparse code. Based on the observation that feature vectors from a certain class should be better represented by basis vectors in the sub-space of that class, we have selected the dominant basis vectors for each class. We have also demonstrated that by combining the frequency distribution of the similar features over the basis vectors, the relationship between local features can be retained during sparse coding. The experiments on four databases have validated the effectiveness of our method.

In the future work, we will explore more relational information between the features to be encoded. Furthermore, we will investigate the manifold structural information, which has proved to be an effective approach to characterizing the structure of descriptors.

References

References

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

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

[3] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: a comprehensive study., IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 493.

[4] X. Ren, D. Ramanan, Histograms of sparse codes for object detection, in: Computer Vision and Pattern Recognition, IEEE, 2013, pp. 3246–3253.

[5] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Low-rank sparse learning for robust visual tracking, in: European conference on Computer Vision, 2012, pp. 470–484.

[6] Y. Zhang, Z. Jiang, L. S. Davis, Learning structured low-rank representations for image classification, in: Computer Vision and Pattern Recognition, IEEE, 2013, pp. 676–683.

[7] S. Gao, I. W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely–laplacian sparse coding for image classification, in: Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3555–3561.

[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[9] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification (2010) 3360–3367.

[10] S. Lazebnik, M. Raginsky, Supervised learning of quantizer codebooks by information loss minimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (7) (2009) 1294–1309.

[11] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, in: International Conference on Computer Vision, Vol. 2, IEEE, 2005, pp. 1800–1807.

[12] C.-K. Chiang, C.-H. Duan, S.-H. Lai, S.-F. Chang, Learning component-level sparse representation using histogram information for image classification, in: International Conference on Computer Vision, IEEE, 2011, pp. 1519–1526.

[13] J. Yang, J. Wang, T. Huang, Learning the sparse representation for classification, in: International Conference Multimedia and Expo, IEEE, 2011, pp. 1–6.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[15] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.

[16] X. Lu, H. Yuan, P. Yan, Y. Yuan, X. Li, Geometry constrained sparse coding for single image super-resolution, in: Computer Vision and Pattern Recognition, 2012, pp. 1648–1655.

[17] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Transactions on Image Processing 20 (5) (2011) 1327–1336.

[18] A. Shaban, H. R. Rabiee, M. Farajtabar, M. Ghazvininejad, From local similarity to global coding: An application to image classification, in: Computer Vision and Pattern Recognition, IEEE, 2013, pp. 2794–2801.

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: International Conference on Computer Vision, IEEE, 2009, pp. 2272–2279.

[20] S. Bengio, F. Pereira, Y. Singer, D. Strelow, Group sparse coding, in: Advances in Neural Information Processing Systems, 2009, pp. 82–89.

[21] S. Mosci, S. Villa, A. Verri, L. Rosasco, A primal-dual algorithm for group sparse regularization with overlapping groups., in: Neural Information Processing Systems, 2010, pp. 2604–2612.

[22] G. Peyré, Sparse modeling of textures, Journal of Mathematical Imaging and Vision 34 (1) (2009) 17–31.

[23] K. Skretting, J. H. Husøy, Texture classification using sparse frame-based representations, EURASIP journal on applied signal processing 2006 (2006) 102–102.

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: International Conference on Machine Learning, ACM, 2009, pp. 689–696.

[25] Y. Liu, F. Wu, Z. Zhang, Y. Zhuang, S. Yan, Sparse representation using nonnegative curds and whey, in: Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3578–3585.

[26] P. Turcot, D. G. Lowe, Better matching with fewer features: The selection of useful features in large database recognition problems, in: International Conference on Computer Vision Workshops, IEEE, 2009, pp. 2109–2116.

[27] G. Kim, C. Faloutsos, M. Hebert, Unsupervised modeling of object categories using link analysis techniques, in: Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[28] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: International Conference on Computer Vision, 2003, pp. 1470–1477 vol.2.

[29] P. Tirilly, V. Claveau, P. Gros, Language modeling for bag-of-visual words image categorization, in: International Conference on Content-based Image and Video Retrieval, ACM, 2008, pp. 249–258.

[30] J. Yuan, Y. Wu, M. Yang, Discovery of collocation patterns: from visual words to visual phrases, in: Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[31] S. Liu, X. Bai, Discriminative features for image classification and retrieval, Pattern Recognition Letters 33 (6) (2012) 744–751.

[32] O. Pele, M. Werman, Fast and robust earth mover's distances, in: International Conference on Computer Vision, 2009, pp. 460–467.

[33] H. A. K. H. Nakagawa, Maeda, Chinese term extraction from web pages based on compound word productivity, in: IJCNLP, 2005, pp. 269–279.

[34] L. Hao, L. Hao, Automatic identification of stop words in chinese text classification, in: International Conference on Computer Science and Software Engineering, Vol. 1, 2008, pp. 718–722.

[35] K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Machine learning 39 (2-3) (2000) 103–134.

[36] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, The Journal of Machine Learning Research 11 (2010) 19–60.

[37] T. Haynes, S. Hedetniemi, P. Slater, Fundamentals of Domination in Graphs, Chapman & Hall/CRC Pure and Applied Mathematics, Taylor & Francis, 1998.
URL http://books.google.com/books?id=Bp9fot_HyL8C

[38] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, Algorithmica (1998) 374–387.

[39] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Computer Vision and Pattern Recognition, IEEE, 2005, pp. 524–531.

[40] L.-J. Li, L. Fei-Fei, What, where and who? classifying events by scene and object recognition, in: International Conference on Computer Vision, IEEE, 2007, pp. 1–8.

[41] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories (2004) 59–70.

[42] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[43] J. Wu, J. M. Rehg, Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel, in: International Conference on Computer Vision, IEEE, 2009, pp. 630–637.

[44] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, S. Ma, Image classification by non-negative sparse coding, low-rank and sparse decomposition, in: Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1673–1680.

[45] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: International Conference on Computer Vision, IEEE, 2011, pp. 2486–2493.

[46] L. Zhang, C. Ma, Low-rank decomposition and laplacian group sparse coding for image classification, Neurocomputing 135 (2014) 339–347.

[47] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[48] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods (2011) 1–12.

[49] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[50] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[51] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European conference on Computer Vision, Springer, 2010, pp. 143–156.

[52] X. Zhou, K. Yu, T. Zhang, T. S. Huang, Image classification using super-vector coding of local image descriptors, in: European conference on Computer Vision, Springer, 2010, pp. 141–154.