Learning Category-Specific Dictionary and Shared Dictionary for Fine-Grained Image Categorization

Shenghua Gao, Ivor Wai-Hung Tsang, Yi Ma

***Abstract*—This paper targets fine-grained image categorization by learning a category-specific dictionary for each category and a shared dictionary for all the categories. Such category-specific dictionaries encode subtle visual differences among different categories, while the shared dictionary encodes common visual patterns among all the categories. To this end, we impose incoherence constraints among the different dictionaries in the objective of feature coding. Moreover, to make the learnt dictionary stable, we also impose the constraint that each dictionary should be self-incoherent. Our proposed dictionary learning formulation not only applies to fine-grained classification, but also improves conventional basic-level object categorization and other tasks such as event recognition. Experimental results on five datasets show that our method can outperform the state-of-the-art fine-grained image categorization frameworks as well as sparse coding based dictionary learning frameworks. All these results demonstrate the effectiveness of our method.**

***Index Terms*—class-specific dictionary, shared dictionary, fine-grained classification.**

I. Introduction

Image classification is a classical problem in computer vision. Many efforts [1][2][3] have been made to tackle this problem in the last couple of decades, including sparse coding based image representation [4][5] or deep learning based unsupervised image representation [6][7] followed by support vector machine (SVM) [8] based classifier, *etc*. These methods have demonstrated very promising performance on many challenging datasets, like Caltech 256 dataset [9], SUN dataset [10], Event dataset [11], *etc*. However, traditional image classification problem typically focuses on scene classification [2], such as differentiating street view versus seashore, or basic-level object categorization [9][12], such as differentiating a pigeon versus a seahorse (see Fig. 1 (a)). Compared to basic-level object categorization, fine-grained categorization (also known as subordinate-level categorization) is a more challenging task because differences between different fine-grained categories are much more subtle and minute. As fine-grained categorization is also very important in many scenarios, like bird recognition [13], flower recognition [14][15], some researchers have made some good attempts on this topic and achieved some promising results [16][13][17].

The main difference between basic-level image classification and fine-grained categorization is the level of difference between different categories. As shown in Fig. 1, for basic-level image classification, the difference between different categories, like pigeon versus seahorse, are evident, and usually most image content is different. By using the conventional sparse coding based image representation [4], which aims at preserving the image content as much as possible in feature coding, the resultant image representations of different categories would likely be very different and simple classifiers like SVM would be able to tell them apart. But for fine-grained categorization, as shown in Fig. 1 (b) and (c) for example, playing a bassoon versus holding a bassoon, or sunflower versus dandelion, the differences between different categories are very minute and subtle. In other words, the common parts dominate images of different categories. Therefore, if we apply conventional representation strategies like sparse coding [4], the learnt dictionary is likely to be dominated by these common parts, so that most dictionary atoms are used to encode common features, and only a very small fraction of atoms are used to encode the differences. As a result, representations of two images from different categories would be dominated by these common features, and the differences between them could be buried by such commons features, which results in poor classification accuracy for fine-grained image categorization. Therefore, for fine-grained image categorization, we would like to amplify the differences while to suppress the common features in the representation of different categories.

(a): pigeon vs. seahorse (From: Caltech-101) (b): playing bassoon vs. with bassoon (From: PPMI) (c): sunflower vs. dandelion (From: Oxford Flower-17)

Fig. 1. Some images for basic-level image categorization and fine-grained image categorization. The sources of the images are indicated in the bracket. The differences between basic-level image categories are significant, but the differences between fine-grained image categories are subtle.

In this paper, to improve the discriminality of image representation and facilitate fine-grained image categorization, we propose to learn a category-specific dictionary for each category and a shared dictionary for all the categories. The category-specific dictionary encodes category-specific parts/features between different categories, while the shared dictionary encodes the common parts among different categories. In this way, we can separate the shared and different parts of each image,

Shenghua Gao is with Advanced Digital Sciences Center, Singapore.
Ivor Wai-Hung Tsang is with Nanyang Technological University, Singapore.
Yi Ma is with Microsoft Research Asia, Beijing, China.
E-mail: {shenghua.gao}@adsc.com.sg

which results in a more discriminative image representation and would boost the image categorization performance.

The main contributions of this paper can be summarized as follows: (i) We propose to learn a category-specific dictionary and a shared dictionary to separate the different and common components of each image for fine-grained image categorization. As discussed in [17] and the introduction of our paper, image representation based on conventional feature encoding which learns a dictionary for all the categories usually fails for fine-grained image categorization. In this paper, we analyze the reason for such failure, and propose a remedy for such failure, i.e., we learn the category-specific dictionary and shared dictionary for feature encoding rather than learn a dictionary for all the categories. By incorporating such dictionary learning strategy into traditional BoW based image representation (feature extraction + feature coding + feature pooling), we show that our method achieves better performance for fine-grained image categorization. (ii) We propose to impose cross-dictionary incoherent constraint and self-dictionary incoherent terms in the objective function for learning such dictionaries. The setting of fine-grained classification and the incoherence have been proposed independently and separately. This is the first work to use incoherence to improve the performance of fine-grained classification method, and we show the incoherence does improve the performance a lot for fine-grained image categorization. (iii) We perform a systematic study on the effect of different feature encoding strategies towards the classification accuracy. Moreover, it is worth noting that our framework not only improves fine-grained image categorization, but also is applicable to other conventional image classification tasks, including basic-level object categorization and event classification.

The rest of this paper is organized as follows: In Section II, we will briefly introduce the related work, including the general image classification methods and fine-grained image categorization methods. In Section III, we will introduce our dictionary learning methods for fine-grained image categorization in details, including the formulation, optimization, and how to use the learnt dictionary to encode and represent test images. We conduct extensive experiments to evaluate the proposed method in Section IV and conclude our work and propose the future work in Section V.

II. Related Work

A. Work Related to General Image Classification. Image classification usually involves two processes: image representation and pattern classification [1][2][4]. Bag-of-Words (BoW) model [1] is a generally used image representation strategy because of its compact representation, and it consists of three modules: (i): Feature extraction, which extracts lots of local features from each image. (ii): Codebook generation and feature coding, which generates a codebook which contains the statistical information and uses the codebook to encode/approximate the local features; In BoW model, k-means is usually adopted to generate the codebook and hard assignment is used for feature coding. However, in such hard assignment based feature coding, each local feature is approximated by its nearest entry in codebook, which will cause severe information loss. (iii): Feature pooling, which aggregates the encoded coefficients of all the local features. In BoW model, average pooling is performed, which calculates the histogram of all the codewords over the whole image. Such pooling strategy loses the spatial information which is important for scene and object recognition.

Based on BoW, many other more advanced image representation methods are proposed. In [2], a Spatial Pyramid Matching (SPM) model is proposed which divides each image into increasing finer subregions and performs average pooling in different subregions. SPM model captures the image spatial information, therefore it improves the object and scene recognition. In [4], a sparse coding based feature coding based image representation which is named as Sparse Coding based Spatial Pyramid Matching (ScSPM) is proposed, which preserves the information as much as possible in feature coding process. Therefore such sparse coding based feature coding improves the accuracy of image representation. Beside sparse coding, a Locality-constrained Linear Coding (LLC) is also proposed for feature coding [18]. LLC considers the locality information among the features. Therefore it improves image classification accuracy. In addition, max pooling [4][19] is used for feature pooling, which experimental demonstrates better performance for the sparse coding based image representation. Moreover, other sparse coding based image representations [5][20][21] have also been proposed and demonstrate good performance for basic-level image classification.

B. Work Related to Dictionary Learning. The idea of learning category-specific or/and shared dictionary has been proposed in previous works [22][23][24][25], but these works are usually used for general image classification, and they are not suitable for fine-grained image categorization. In [22], a class dictionary is adapted from a universal dictionary, and the work is based on GMM model. Each image is represented by the universal dictionary and each class dictionary separately in such model. As a result, the subtle difference among classes is lot. Hence this model is not suitable to fine-grained image categorization. Though [25][21] propose to category-specific dictionary for each category, the commonalities among different categories which are adverse for fine-grained image categorization cannot be get rid of the image representation. Zhou *et al.* [26] propose to learn the category-specific dictionary and shared dictionary by minimizing the within-classes scatter of the sparse codes belonging to the same category and maximizing the between-class scatter of the sparse codes over the shared dictionary. However, it is still not suitable for fine-grained categorization where the commons between different categories are more significant than the differences. Moreover, Kong *et al.* propose a category-specific dictionary learning method named DL-COPAR [24] which aims at separating commonality and particularity. There method demonstrates good performance for basic-level categorization. Our formulation are different from these works in terms of both formulation and application.

C. Work Related to Fine-Grained Image Categorization. Fine-grained image categorization has been previously explored, like flower recognition on the Oxford Flower 17

dataset [15] and the Oxford Flower 102 dataset [14], bird recognition [27], *etc*. In [28], a Grouplet feature is proposed. This Grouplet feature demonstrates good performance for recognizing whether the people is playing the instrument or just holding the instrument because it captures the structured information by considering the discrimination and spatial configuration. In [16], the discriminative information of the features is used to distinguish the differences between different classes, meanwhile the randomization is also adopted to make the algorithm scalable to the number of the features. Moreover, some researchers also propose to use volumetric primitives and pose-normalized appearance [13], or template matching [17] method for bird recognition.

III. Category-Specific Dictionary and Shared Dictionary for Image Representation

In this section, we will first briefly revisit the sparse coding based feature coding [4], then we will propose our category-specific dictionary learning method, including its formulation and its optimization. We will also describe our image representation method using the learnt sparse codes. We use $[Q_1; Q_2]$ to denote the vertical concatenation of two matrices with the same columns, and we use $[Q_1, Q_2]$ to denote the horizontal concatenation of two matrices with the same rows.

A. Review of Sparse Coding Based Feature Coding

Feature coding encodes the local feature by using the dictionary. Denote the feature space as $X = [x_1, x_2, \ldots, x_p]$ ($x_i \in \mathbb{R}^d$ is a local feature), denote the codebook as $U = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^{d\times k}$ (Each column in codebook is called as a codeword or atom, and the codebook is also named as dictionary.). Linear feature coding usually approximates the local feature x_i by using the codebook U, i.e., $x_i \approx Uv_i$, here v_i is the reconstruction coefficients. In BoW model, k-means based feature coding is used, and each feature is approximated by one codebook entry. Therefore only one entry in v_i equals to 1, and all the rest entries in v_i equal to zero. The k-means based feature coding can be formulated as follows:

$$\min_{U,V} \quad \sum_{i=1}^{p} \|x_i - Uv_i\|_2^2 \ \text{ s.t. } \ \|v_i\|_0 = \|v_i\|_1 = 1. \qquad (1)$$

Here $V = [v_1, v_2, \ldots, v_p]$ (where $v_i \in \mathbb{R}_+^{k\times 1}$), and U is the cluster centers of k-means in BoW model. As aforementioned, the constraint that each local feature is only approximated by one codeword is too strict and will cause the information loss, especially for those points with similar distances to several codewords. To reduce the information loss, the hard constraint $\|v_i\|_0 = 1$ on v_i is relaxed, and the to avoid each feature to be assigned with too many clusters, the sparse constraint is imposed on the weight vector v_i. Then, we arrive at the sparse coding based feature coding [4].

$$\min_{U,V} \quad \sum_{i=1}^{p} \|x_i - Uv_i\|_2^2 + \lambda\|v_i\|_1 \ \text{ s.t. } \|u_j\|_2 = 1. \qquad (2)$$

Here the constraint on each codeword u_j is used to avoid the trivial solution, and v_i is the sparse reconstruction coefficients (sparse codes).

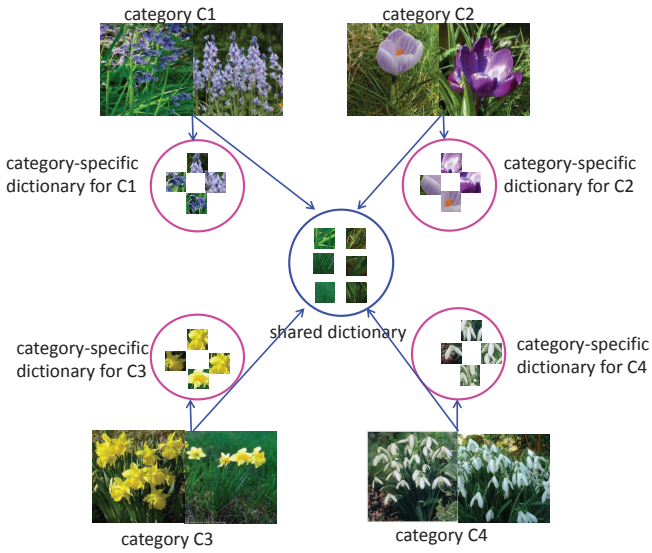


Fig. 2. An illustration of learning category-specific dictionary and shared dictionary for fine-grained image categorization. Note that the features from each category are encoded by using the atoms in the corresponding category-specific dictionary and the shared dictionary. Since our work extracts the features from image patches, for sake of brevity and better visualization, in the diagram we just show the patches representing the features instead of the atoms.

B. Learning Class-Specific and Shared Dictionary

In sparse coding based feature coding [4], a dictionary is learnt to minimize the reconstruction error of all the features from the all the categories, therefore it does not distinguish which codewords are category-specific. Nevertheless, it works well for basic-level classification because the difference between different categories are typically rather significant. Therefore, the sparse codes of images from different categories are different, and result in discriminative image representation for classification. But for fine-grained image categorization, most image contents from different categories are similar, and the difference between them is subtle. If we use the same coding scheme, images of different categories would share lots of similar codes, and the proportion of different/discriminative codes would be very small. That difference can be dominated by those similar sparse codes and even disappear at the feature pooling stage. Hence it is desirable to find a dictionary that could encode category-specific features of different categories with their category-specific codewords. Such dictionary would obviously boost the differences of the representations for images from different categories, and improves the consequent image categorization.

To this end, we propose to learn a category-specific dictionary for each category and a shared dictionary for all categories in the feature coding process. When a feature comes in, if it is category-specific feature, it would be encoded by its corresponding category-specific dictionary; If it is a common feature shared by many categories, then it should be encoded by the shared dictionary. By controlling the size of category-specific dictionary and shared dictionary, we can amplify the differences among different categories. We illustrate such dictionary learning methods for fine-grained image categorization

in Fig. 2.

We denote the number of local features from the i^{th} category as m_i, and the number of codewords corresponding to the i^{th} category-specific dictionary as n_i. Denote features associated with the i^{th} category as X_i ($X_i \in \mathbb{R}^{d\times m_i}$), its corresponding category-specific dictionary as U_i ($U_i \in \mathbb{R}^{d\times n_i}$), and the shared dictionary as U_0 (($U_0 \in \mathbb{R}^{d\times n_0}$)). Assuming there are N categories in total, the complete dictionary is $U = [U_0, U_1, \ldots, U_N] \in \mathbb{R}^{d\times n}$, with $n = \sum_{i=0}^{N} n_i$. Denote the sparse reconstruction coefficients of X_i corresponding to U_j as V_i^j. Mathematically we can express X_i as follows: $X_i \approx U_0V_i^0+U_1V_i^1+\ldots+U_NV_i^N$. At the training stage, since X_i only contains features from the i^{th} category, ideally only the coefficients associated with U_i and U_0, i.e., V_i^i and V_i^0, can be non-zero. By abuse of notation, we let $V_i = [V_i^0; V_i^i]$ ($i = 1, \ldots, N$). Denote U_{-i} ($i = 0, \ldots, N$) as the submatrix by removing U_i from U, for example, $U_{-0} = [U_1, \ldots, U_N]$, and $U_{-k} = [U_0, \ldots, U_{k-1}, U_{k+1}, \ldots, U_N]$. Mathematically we can formulate the overall dictionary learning as follows ($\|\cdot\|_F$ as the Frobenius norm.):

$$\begin{aligned}\min_{U_0,U_i,V_i} &\sum_{i=1}^{N}\big(\|X_i - [U_0, U_i]V_i\|_F^2 + \lambda\|V_i\|_1\big)\\ &+\sum_{i=0}^{N} m_i\big(\frac{\eta_{i1}}{n_i^2}\|U_i^TU_i - I_{n_i}\|_F^2 + \frac{\eta_{i2}}{n_i(n-n_i)}\|U_i^TU_{-i}\|_F^2\big)\\ &\text{s.t. } \|U_k(:,j)\| = 1, \forall k, j,\end{aligned} \tag{3}$$

Here $m_0 = \sum_{i=1}^{N} m_i$ because we use all the features to update U_0 (Please refer to equation (6) and (7)). I_{n_i} is a identity matrix whose size is $n_i \times n_i$, and η_{i1} and η_{i2} ($i = 0, \ldots, N$) are the weight for self-incoherent term and cross-incoherent term for the i^{th} dictionary respectively. In the above formulation, constraining each codeword to be norm 1 is to avoid trivial solution. The term $\|U_i^TU_{-i}\|_F^2$ is to enforce sub-dictionaries to be incoherent [23]. Without this constraint, each category-specific dictionary only needs to best encode features of its own category whereas the shared dictionary can be empty. The self-incoherent term $\|U_i^TU_i - I_{n_i}\|_F^2$ is to stabilize the learnt dictionary for each category [25]. Otherwise it may lead to many codewords being zeros in the category-specific dictionaries.[1] Finally, we normalize the reconstruction error and coefficient terms by the number of features of each category; and normalize the incoherence with the number of atoms in each category. For simplification, we name our method as *Category-Specific Dictionary Learning* (CSDL).

Remarks: Compared with [25][21], besides the category-specific dictionary, we also learn a shared dictionary for all the categories which encodes the common patterns shared by different categories. In this way, we can amplify the differences among different categories and boosts the fine-grained image categorization. As shown in the works [24][26] and our experimental results on PPMI+(7 Categories) and PPMI (24 categories), such shared dictionary does paly an important role for fine-grained image categorization. Moreover, our work is different from [24] in following three aspects. Firstly, we use the self-incoherent term for each dictionary, which avoids the codebook to be empty. [2] Secondly, we weight self-incoherence term and cross incoherence with the number of features, which alleviates the effect of feature size in dictionary learning, but [24] doesn't do this. Specifically, in our formulation, we use $\frac{m_i\eta_{i1}}{n_i^2}$ to weight the self-incoherent term ($\|U_i^TU_i - I_{n_i}\|_F^2$) and use $\frac{m_i\eta_{i2}}{n_i(n-n_i)}$ to weight the cross-incoherent term ($\|U_i^TU_{-i}\|_F^2$). In equation (3), the first term (reconstruction error of all the features used for dictionary learning) scales with the number of features, but the self-incoherent and cross-incoherent terms are only in the scale of the dictionary size. Therefore, these three terms are optimized with different scales of normalization terms. To control the influences of self-incoherent and cross-incoherent terms in learning dictionaries, we need to weight the self-incoherent and cross-incoherent terms with the number of features used for dictionaries. In other words, such weighting strategy can make the learned dictionary more stable for image classification. Moreover, when m_i is large enough, the learned dictionary is usually insensitive to m_i. We have clarified this in the revised version of this manuscript. Thirdly, the coefficients corresponding to the shared dictionary, which are discarded in [24], are also used for classification and as shown in Fig. 4 that such coefficients usually improve the performance of image categorization.

C. Optimization Procedure

The objective function of the above Category-Specific Dictionary Learning problem (3) is not convex. Following popular optimization strategies [29][23], we alternatively update the sparse codes of features from each category, each category-specific dictionary, and the shared dictionary while keeping all the rest variables fixed. In the remaining of this subsection, we describe our optimization strategy and summarize the overall optimization process in Algorithm 1.

1. Inferring the Sparse Codes. With all other factors fixed in the objective of CSDL, the part depending on V_i is as follows:

$$\min_{V_i} \ \|X_i - [U_0, U_i]V_i\|_F^2 + \lambda\|V_i\|_1. \tag{4}$$

This problem is in the form of standard robust regression (the Lasso) or sparse coding. Many efficient methods [30][31][32] have been developed for optimizing such an objective function. In this work, we adopt the Feature Sign Search algorithm [29], due to its good performance and fast speed.

2. Learning the Category-Specific Dictionary. Define $Z_r = X_r - U_0V_r^0$, with $Z_r \in \mathbb{R}^{d\times m_r}$. When the sparse codes and the shared dictionary are fixed, we arrive at the optimization problem with respect to the r^{th} category-specific dictionary

[1] Following the work of efficient sparse coding [29], the constraint on each codeword is $\|U_k(:,j)\| \leq 1$, and in optimization, the objective is usually optimized without considering such constraint and then the solution is normalized. Therefore the codebook may contain zero entries.

[2] In [24], the self-incoherent constraint is not imposed on each category-specific dictionary, and the optimization of [24] firstly update the codebook without considering the norm 1 constraint, therefore the codebook for each category can be empty, but our self-incoherent term can avoid this.

U_r as follows:

$$\begin{aligned}\min_{U_r} \quad & \frac{1}{m_r}\|Z_r - U_r V_r^r\|_F^2 + \frac{\eta_{r1}}{n_r^2}\|U_r^T U_r - I_{n_r}\|_F^2 \\ & + \frac{\eta_{r2}}{n_r(n-n_r)}\|U_r^T U_{-r}\|_F^2 \\ \text{s.t.} \quad & \|U_r(:,j)\| = 1, \forall j.\end{aligned} \tag{5}$$

Following the work of [23], we use a gradient descent algorithm to optimize this objective, and the step size is chosen according to the Armijo rule.[3] Then we normalize the solution so that the ℓ_2 norm of each codeword is 1.

3. Learning the Shared Dictionary. When the sparse codes and all the category-specific dictionaries are fixed, the optimization problem respect to the shared dictionary U_0 is as follows:

$$\begin{aligned}\min_{U_0} \quad & \sum_{i=1}^{N} \frac{1}{m_0}\|X_i - [U_0, U_i]V_i\|_F^2 + \frac{\eta_{01}}{n_0^2}\|U_0^T U_0 - I_{n_0}\|_F^2 \\ & + \frac{\eta_{02}}{n_0(n-n_0)}\|U_0^T U_{-0}\|_F^2 \\ \text{s.t.} \quad & \|U_0(:,j)\| = 1, \forall j.\end{aligned} \tag{6}$$

Define $Y = [X_1 - U_1 V_1^1, \ldots, X_N - U_N V_N^N] \in \mathbb{R}^{d \times m_0}$ and $V_0 = [V_1^0, \ldots, V_N^0]$, then we can rewrite the above objective with respect to U_0 as follows:

$$\begin{aligned}\min_{U_0} \quad & \frac{1}{m_0}\|Y - U_0 V_0\|_F^2 + \frac{\eta_{01}}{n_0^2}\|U_0^T U_0 - I_{n_0}\|_F^2 \\ & + \frac{\eta_{02}}{n_0(n-n_0)}\|U_0^T U_{-0}\|_F^2 \\ \text{s.t.} \quad & \|U_0(:,j)\| = 1, \forall j.\end{aligned} \tag{7}$$

Following the work [23], we also use gradient descent method to update the shared dictionary.

4. Implementation Details. We learn the dictionaries in the following steps. i) Initialize dictionaries with K-SVD. K-SVD [33] is an algorithm of learning overcomplete dictionary in a singular value decomposition approach, and it has demonstrated good performance on many computer vision applications [34][35]. It alternatively updates the dictionary and sparse reconstruction coefficients. Following the work [24], we also initialize each category-specific dictionary with the K-SVD [33] by using the features from its corresponding category, and we initialize the shared dictionary with the K-SVD over the features from all the categories. ii) Update the dictionaries with Lagrange Dual Algorithm to guarantee that the dictionaries should satisfy the small reconstruction criteria first. As stated in the work [23], small reconstruction error is important for dictionary learning. Therefore, following the work [23], before learning the dictionaries with CSDL formulation in equation 5 and equation 7, we also update each dictionary (without the self-incoherent and cross-incoherent terms) with Lagrange Dual Algorithm [29]. That is, all the dictionaries should satisfy the small reconstruction error constraints first. iii) Use the dictionaries satisfying the small reconstruction error as starting points to learn the dictionaries with CSDL formulation.

[3]More details about the gradient descent algorithm we used can be found from the codes of [23] which are available from the following website: http://iie.fing.edu.uy/~nacho/?static3/software.

Algorithm 1 Learning Category-Specific Dictionary and Shared Dictionary

Input: The features of the i^{th} category: $X_i,\ i = 1, \ldots, N$; The number of features in X_i: m_i; $\lambda, \eta_{i1}, , \eta_{i1}$; The size of each Specific-Dictionary n_i; The size of the Shared Dictionary: n_0.
Initialize each U_i with X_i by using K-SVD. Initialize U_0 with $X = [X_1, \ldots, X_N]$ by using K-SVD.
Update all the dictionaries with Lagrange Dual Algorithm.
repeat
  for $i = 1$ **to** N **do**
    Infer the sparse codes V_i of X_i with feature sign search algorithm (Equation (4));
  end for
  for $i = 1$ **to** N **do**
    Update each U_i using gradient descent (Equation (5));
  end for
  Update the shared dictionary U_0 using gradient descent (Equation (7)).
until stopping criteria is reached.
Output: The Category-Specific Dictionary: $U_i\ \forall i$; The Shared Dictionary: U_0.

D. Encode Local Features with Learnt Dictionary

In this section, we propose two image representation strategies based on learnt dictionaries, namely, **G**lobal **E**ncoding and **L**ocal **E**ncoding, to encode the local features. Because SVM is used for the classification, therefore the entries at the same locations of the final image representation vectors should correspond to the response to the same atom in the codebook used for feature encoding for both training and test images. Therefor we need to represent the training and test samples in the same way, i.e., we need to encode the features under the same dictionary, and use the same feature pooling strategy to postprocess the sparse coefficients for image representation.

1. Encode local features with the global dictionary. We stack all the dictionary together $U = [U_0, U_1, \ldots, U_N]$. We use this shared dictionary to encode features F from each image by solving the following problem:

$$\min_{S} \quad \|F - U S_{GE}\|_F^2 + \lambda \|S_{GE}\|_1, \tag{8}$$

then perform feature pooling on S. We denote this feature encoding technique as **Global Encoding (GE)**.

2. Encode local features with all category-specific dictionaries. We stack the shared dictionary with the each category-specific dictionary together, and get N combined dictionaries: $\hat{U}_1 = [U_0, U_1], \ldots, \hat{U}_N = [U_0, U_N]$. Then we use each combined dictionary to encode the features from each image by solving the following problem:

$$\min_{S_i} \quad \|F - \hat{U}_i S_i\|_F^2 + \lambda \|S_i\|_1, \quad \forall i \tag{9}$$

then we stack the sparse codes w.r.t. all N dictionaries together, i.e., $S_{LE} = [S_1; \ldots; S_N]$. Then we use S to perform feature

pooling in the later stage. We denote this feature encoding technique as **Local Encoding (LE)**.

E. Feature Pooling and Image Representation

Because of the good performance of max pooling and its simplification [4][36], we use the max pooling on the sparse codes. Denote $S \in \mathbb{R}^{l\times s}$ as the reconstruction coefficients corresponding to the features within certain image region where S can be obtained by either S_{GE} or S_{LE} , max pooling only preserves the largest response of all these features to each atom in the dictionary. After max pooling, this image region is represented by a feature vector $y \in \mathbb{R}^l$, and

$$y_i = \max_{1\le j\le s} S_{ij} \tag{10}$$

Moreover, to preserve the spatial information, we also use the Spatial Pyramid Matching [2] for image representation. Specifically, we use three layers of SPM, i.e., each image is evenly divided into 1×1, 2×2 and 4×4 subregions. Then we perform the max pooling of local features in each subregion. Then we concatenate the results of max pooling in each subregion with the same weight and normalize its ℓ_2 norm to be 1. We illustrate the flowchart of our CSDL-based image representation for fine-grained image categorization in Fig. 3.

IV. Experiments

In this section, we will experimentally evaluate our proposed CSDL algorithm on five publicly available datasets, and compare it with existing works. We firstly evaluate our method on PPMI dataset and use it to set the parameters in the CSDL model empirically. Then we evaluate our method on other fine-grained categorization datasets and general image classification datasets.

A. Experimental Setup

Histograms of Oriented Gradients (HOG) feature [37] is commonly used for local patch description [13][10]. In this paper we also adopt the HOG feature. Specifically, we divide each image into 8×8 non-overlapped dense grids, and use HOG to characterize each grid. Then the four descriptors from 2×2 neighboring grids are concatenated and form a $124D$ feature vector. Each feature is normalized with the ℓ_2 normalization. Considering computational speed and memory issue, we randomly sample about 12K-15K features from each training category to initialize and to learn the category-specific dictionaries. As for the weight of sparsity term (λ), it has been experimentally shown that good performance can be achieved when it is set to be 0.3 [4]. For simplification, we also set λ to be 0.3 in our work. As for the weight of incoherent terms, we simply set the ratio between the entries in self-incoherence term and cross-incoherence term to be 1:2, i.e., $\eta_{i1} = \frac{\eta_{i2} n_i}{2(n-n_i)}$. In the case that the size of category-specific dictionary is the same with that of the shared dictionary, we can easily get $\eta_{i1} = \frac{\eta_{i2}}{2N}$, $\forall i$, here N is the class number. For the classification module, Liblinear SVM [38] is used to train the one-vs.-all classifiers because of its fast speed. The results listed in all following sections are based on the standard train/test splits if they are provided in the datasets. Otherwise the results are based on 10 independent experiments with randomly generated train/test split. Moreover, all category-specific dictionaries and shared dictionary have the same size in all following experiments.

B. Evaluation on PPMI

The People-Playing-Music-Instruments (PPMI) dataset [28] is a commonly used dataset for fine-grained categorization, and it has two versions. One version contains 7 instrument categories: *bassoon*, *erhu*, *flute*, *French horn*, *guitar*, *saxophone*, and *violin*, and each category also contains two subcategories: people are playing the instruments (this subcategory is denoted as *playing the instruments*), and people are with (but not playing) the instruments (this subcategory is denoted as *with the instruments*). The other version of PPMI contains 12 instrument categories. Besides the 7 categories in the first version, it also includes *cello*, *clarinet*, *harp*, *recorder*, and *trumpet*, and each category also contains *playing the instruments* subcategory and *with the instruments* subcategory. Following the work of [28], we also denote the subset that people play the instruments as PPMI+ and denote the subset of dataset that people are with the instruments as PPMI-. Following the previous work [28][16], we use the normalized images whose size is 256×256 pixels for feature extraction. We also use the standard train/test split, i.e., 100 images are used as training data and 100 images which are not overlapped with training data are used as test data for each subcategory.

1. Feature Encoding and Feature Pooling. We firstly evaluate the effect of Global Encoding (**GE**) and Local Encoding (**LE**) by performing the classification task on PPMI+ (7 categories). We denote the pooling by using all the sparse codes over the global/combined dictionary as Total Sparse Codes Pooling (**TSCP**), and denote the pooling by using the sparse codes corresponding to the category-specific dictionary (without the parts corresponding to the shared dictionary) as Local Sparse Codes Pooling (**LSCP**). In this experiment, the size of each category-specific dictionary and shared dictionary is fixed to be 128. The performance of different feature encoding and pooling techniques corresponding to $\eta_{i2} = 0.1, \forall i$ is shown in Fig. 4. Results show that Global Encoding usually achieves better performance, and this agrees with previous findings that the collaborative representation is helpful for performance improvement [39]. Moreover, Fig. 4 also shows that the TSCP pooling usually boosts the classification accuracy. The possible reasons for the better performance of TSCP pooling are: (i) we have already amplified the subtle difference by controlling the size of category-specific dictionaries and shared dictionary; and (ii) sometimes the common patterns don't appear in all the categories. Based on the observations in Fig. 4, Global Encoding with the total sparse codes pooling are adopted in the following experiments.

2. Parameter Selection We also show the effect of η_{i2} on the classification accuracy in Fig. 5, and it reveals that the classification accuracy is better when $\eta_{i2} = 0.1, \forall i$. For simplification, we set $\eta_{i2} = 0.1, \forall i$ in the following

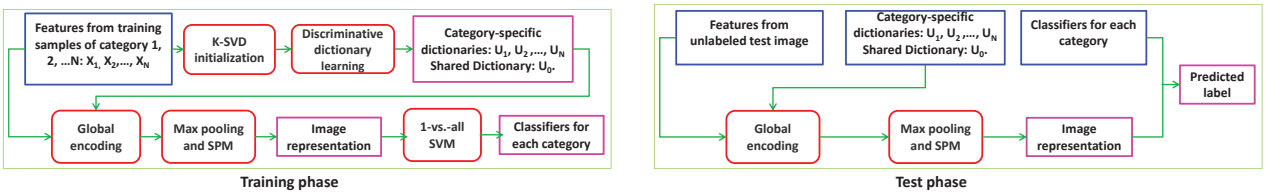


Fig. 3. The flowchart of our CSDL-based image representation for fine-grained image categorization. The blue boxes are the inputs, the red boxes are the key processing modules, and the pink boxes are the outputs.

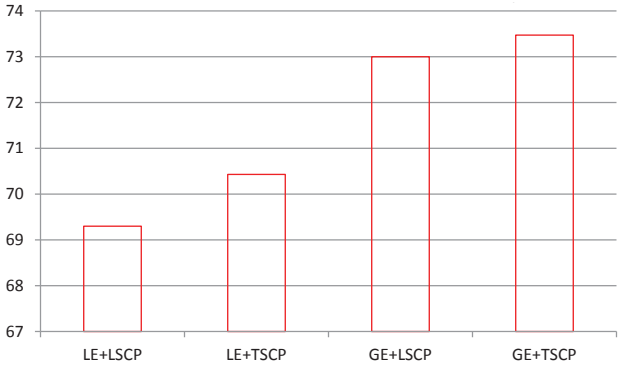


Fig. 4. The performance under different encoding and pooling methods on the PPMI+ dataset (7 category) ($\eta_{i2} = 0.1, \forall i$).

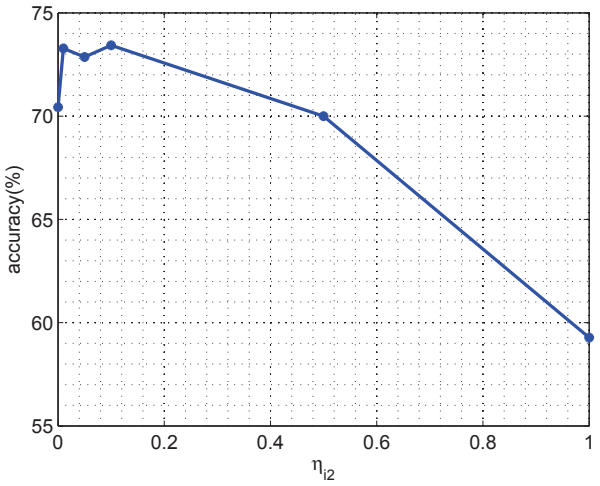


Fig. 5. The performance under different η_{i2}'s on the PPMI+ dataset (7 category). We set $\eta_{i2}, \forall i$ to be 0, 0.01, 0.05, 0.1, 0.5, and 1, and TSCP is used.

experiments. Please note that such parameters are fixed on the all the datasets, and the performance corresponding such parameters are already good enough. We believe other fine-tuned parameters may further improve the performance.

3. Classification Performance on PPMI. Following the setting in [28][16], we perform the image categorization on PPMI+ (7 categories), and on all the 24 categories of PPMI (24 categories). We list the classification accuracy on PPMI+ (7 categories) in Table I, and the performance of different methods on PPMI (12 categories PPMI+ and 12 categories PPMI-) in terms of classification accuracy and **M**ean **A**verage **P**recision (**MAP**) in Table II. Similar to the work of [16], here the MAP is calculated based on PASCAL VOC 2010 Evaluation Protocol, and it is calculated as follows, "(1): Compute a version of the measured precision/recall curve with precision monotonically decreasing, by setting the precision for recall r to the maximum precision obtained for any recall $r' > r$.(2): Compute the AP as the area under this curve by numerical integration. No approximation is involved since the curve is piecewise constant." [40].

Results in Table I show the average classification accuracy of our method on PPMI+ (7 Categories) is 7.73% higher than that of Grouplet method [28] which is specially designed for the classification of the PPMI dataset. Furthermore, results in Table II shows that the MAP of our method significantly outperforms that of Grouplet by 9.84% on PPMI (24 categories) classification task. Please note that the results of LLC and ScSPM are based on the implementation of the same features and the codes provided by the authors of the corresponding paper. Following the work [16], 4-layer SPM is used on this dataset. Meanwhile, when features extracted from grids of 2 scales (8×8 and 16×16) are used, our method outperforms work [16] which achieves the best performance in PASCAL Action Classification Challenge 2011, and achieves the best performance. Please note that [16] uses MULTI-SCALE (5 scales) feature while our work only uses two scales feature, and previous work [18] has shown that multi-scale features do help to improve the classification performance. Table I and Table II also demonstrate that our method outperforms DL-COPAR [24] which learns the commonality and particularity for basic-level image object recognition. Moreover, our method achieves better performance than ScSPM [4] and LLC [18]. It is worth noting that ScSPM and LLC achieve the state-of-the-art performance in basic-level object categorization. We also include the confusion matrix of our method on PPMI+ (7 Categories) in Fig. 6, which reflects the quality of each classifier in classifying the images in this dataset.

4. The Importance of Self-Incoherent Term. In Table I and Table II, we also compare our method with the formulation without self-incoherent term. Our method outperforms the formulation without self-incoherent term by 3% (70.43% vs. 73.43%) in terms of classification accuracy on PPMI+ (7 Categories), and it also outperforms the performance based on the formulation without self-incoherent term by 2% in terms of classification accuracy (46.75% vs. 48.75%) on PPMI (24 Categories). These results demonstrate the usefulness of such self-incoherent term.

5. Comparison with Method Based Category-Specific

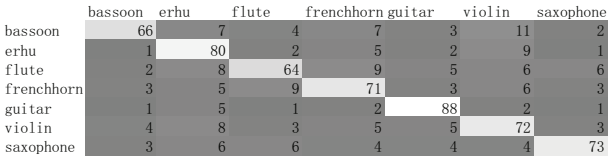


Fig. 6. The confusion matrix (%) on the PPMI+ dataset (7 category). In confusion matrix, the entry located in i^{th} row, j^{th} column in confusion matrix represents the percentage of class i being misclassified to class j.

TABLE I
AVERAGE CLASSIFICATION ACCURACY ON THE PPMI+ DATASET (7 CATEGORIES) (%).

Methods	Accuracy
DPM [41]	54.9
SPM [2]	59.9
ScSPM [4]	71.57
LLC [18]	70.43
Grouplet+SVM [28]	65.7
Grouplet+Model [28]	60.1
DL-COPAR [24]	53.86
CSDL($\eta_1 = 0$)	70.43
CSDL	**73.43**

Dictionary. To further demonstrate the effectiveness of our method, we also design another baseline: We learn the category-specific dictionary with the sparse coding formulation by using the features from each category, then we combine these category-specific dictionaries together in the feature coding process. For fair comparison, max pooling and Spatial Pyramid Representation are also used. Here we denote such baseline as weakly supervised ScSPM (wsScSPM) because the label information is only used in the dictionary learning process. It is worth noting that wsScSPM actually learns the dictionary in the same way with the work [21]. We list the performance of our method, ScSPM and wsScSPM in the Table III. For ScSPM, it learns a global dictionary for all the categories, because the high similarity between different categories for fine-grained categorization, the minute differences between different categories, which is extremely important for fine-grained image categorization, may be easily ignored in ScSPM. For wsScSPM, it learns a dictionary for each category, but the similarity between different categories is very high, therefore the learnt dictionaries may also be similar. Therefore, the common patterns/features from certain category may be encoded by using atoms from dictionaries of other categories, which makes the categorization difficult. However, our method can avoid the disadvantages in both ScSPM and wsScSPM, and the performance of our method on the PPMI datasets in Table III proves the effectiveness of our method. Moreover, to further demonstrate the differences between our CSDL and wsScSPM, we calculate the average percentage of non-zero entries located on the corresponding category-specific dictionary and shared dictionary in the feature coding process in our CSDL. We also calculate the average percentage of non-zero entries located on the corresponding category-specific dictionary in scScSPM. We show such statistics in Fig. 7. This figure shows that compared with wsScSPM, the features are more prone to be reconstructed by using the corresponding category-specific dictionary and shared dictionary in our method. Therefore the image representations of images from different categories may be more different, and be more easily to be classified.

TABLE II
PERFORMANCE COMPARISONS BETWEEN DIFFERENT METHODS ON PPMI CLASSIFICATION (24 CATEGORIES). R+D: RANDOMIZATION+DISCRIMINATION. THE NUMBER IN THE BRACKETS INDICATES THE NUMBER OF GRID SCALES USED FOR FEATURE EXTRACTION.(%)

Methods	Accuracy	MAP
ScSPM [4]	41.54	40.60
LLC [18]	39.70	38.46
Grouplet [28]	NA	36.7
R+D [16] (5)	NA	47.0
DL-COPAR [24]	39.29	38.21
CSDL ($\eta_1 = 0$) (2)	46.75	45.10
CSDL (1)	**45.11**	**46.54**
CSDL (2)	**48.75**	**47.44**

TABLE III
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS (%)

Method	ScSPM	wsScSPM	CSDL
PPMI+ (7 Categories)	71.57	70.43	**73.43**
PPMI (24 Categories)	41.54	43.96	**45.11**

C. Evaluation on The Oxford Flower-17 Dataset

The Oxford Flower-17 dataset [4] contains 17 categories, and each category contains 80 images in which 40 images are used as training samples, therefore the total image number is 1360. We use the standard train/test split provided by the dataset. Following the work of [4], the images are resized to make the maximum side to be 300 pixels meanwhile the aspect ratio is kept. The size for each category-specific dictionary and shared dictionary is also fixed to be 128. The comparisons of our method with other state-of-the-art methods are listed in Table IV. We can see that our method consistently outperforms all the rest methods, including the method of JDL [26], which also learns a dictionary for each category and a shared dictionary for all the category. But as aforementioned, it uses the inter-class variance and intra-class variance as the criteria for dictionary learning, which may suffer from the high visual similarity among the different categories in fine-grained categorization. We also show the confusion matrix on the Oxford Flower-17 dataset in Fig. 8.

D. The Effect of Dictionary Size on Classification Accuracy.

We also experimentally investigate the relationship between the performance and dictionary size. In Fig. 9 (a), we fix the size of category-specific dictionary to be the same with that of shared dictionary on the PPMI+ (7 categories) and Oxford Flower-17 datasets. It shows that classification accuracy increases with the increasing size of each sub-dictionary, and the possible reason is that more details would be captured in the dictionary by increasing the size of the dictionary. In Fig. 9 (b), we fix the size of category-specific dictionary and

[4]www.robots.ox.ac.uk/∼vgg/data/flowers/17/index.html

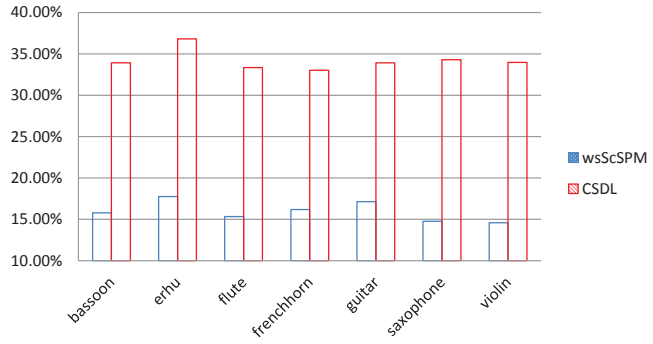


Fig. 7. The percentage of non-zero sparse codes located on the corresponding category-specific/shared dictionaries in our method, and the percentage of non-zero sparse codes located on the corresponding category-specific dictionary in wsScSPM.

TABLE IV
AVERAGE CLASSIFICATION ACCURACY ON THE OXFORD FLOWER-17 DATASET (%).

Methods	Accuracy
ScSPM [4]	52.35
MCLP [42]	66.74
MTJSRC [43]	68.43±1.03
JDL [26]	68.69
DL-COPAR [24]	59.02±1.45
CSDL	**72.65 ± 1.79**

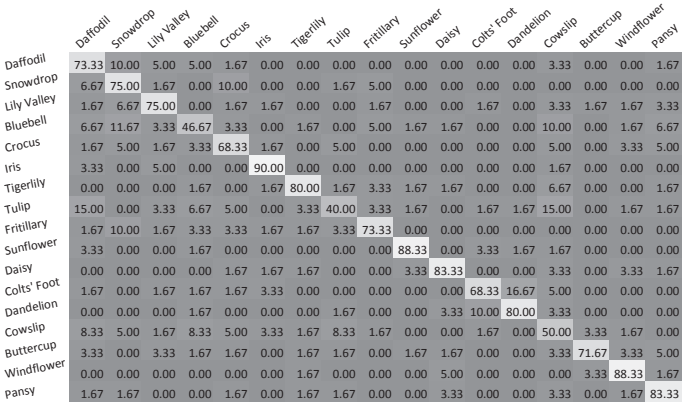
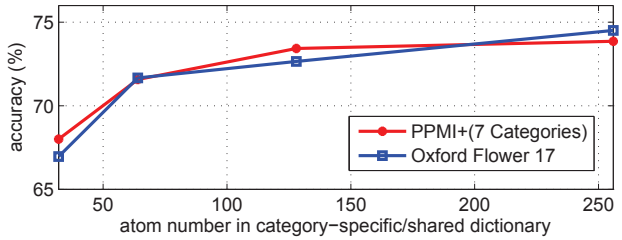
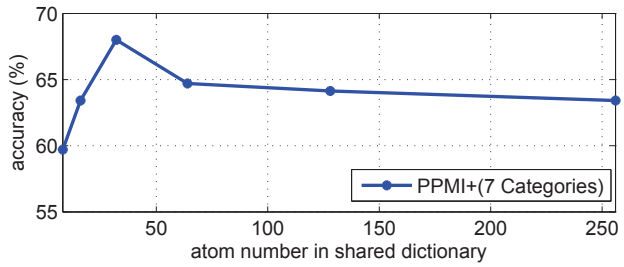


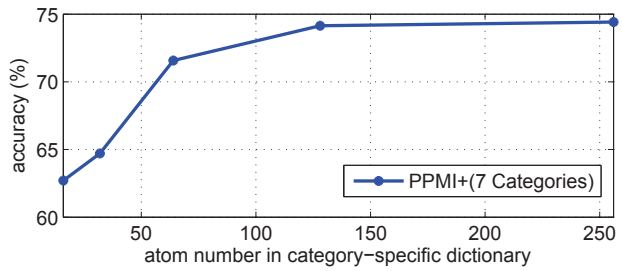
Fig. 8. The confusion matrix (%) on the Oxford Flower-17 dataset.



(a) Atom number in category-specific dictionary is the same as atom number in shared dictionary from 32, 64, 128, 256.



(b) Atom number in category-specific dictionary = 32, atom number in shared dictionary is 0, 16, 32, 64, 128 and 256.



(c) Atom number in shared dictionary = 64, atom number in category-specific dictionary is 32,64, 128 and 256.

Fig. 9. The effect of dictionary size on the image classification accuracy.

increase the size of shared dictionary. At first, classification accuracy increases with the increasing size of the shared dictionary. The possible reason for this is that more common patterns would be captured with larger shared dictionary. But when the shared dictionary reaches to a certain size, further increasing the atoms in the shared dictionary will decrease the classification accuracy. Though larger dictionary captures the details, further increasing shared dictionary may amplify the common patterns in image representation, as a result, the accuracy of fine-grained image categorization drops. In Fig. 9 (c), we fix the size of shared dictionary and increase the size of category-specific dictionary, and we can see that the classification accuracy increases with the increasing size of each category-specific dictionary, and the possible reason for this is that more category-specific details would be captured by larger category-specific dictionary.[5][6]

E. Evaluation on The Corel 10 and Event Datasets

Besides the fine-grained classification, our method can also be applied to other classification tasks. Here we evaluate our method for basic-level object recognition and event classification. The **Corel-10** dataset [44] contains 10 categories: *skiing, beach, buildings, tigers, owls, elephants, flowers, horses, mountains* and *food*, and each category contains 100 images. Following the work of Lu *et al.* [44], each image is resized to make the maximum side to be 300 pixels with the same aspect ratio. We also randomly select 50 images as training samples and use the rest images as test samples. The **Event** dataset [11] contains 8 categories: *badminton*,

[5]It is worth noting that though (a) and (c) shows larger category-specific dictionary boosts the classification accuracy, a very large category-specific dictionary probably brings down the the performance. It is because such dictionary may be too category-specific for the training data and may be ineffective for test data. Moreover the computational cost is very expensive for too large dictionary, hence we don't conduct such investigation here.

[6]We guess the number of the dictionary atoms in each category-specific dictionary should related to the content complexity of that category, and we will study the relationships among the image content complexity, the number of features used learning each category-specific dictionary, and the classification accuracy in our future work.

TABLE V
AVERAGE CLASSIFICATION ACCURACY ON THE COREL-10 AND EVENT DATASETS (SMM: SPATIAL MARKOV MODEL,%).

Corel-10		Event	
Methods	Accuracy	Methods	Accuracy
SMM[46]	77.9	HIK[45]	83.54±1.13
ScSPM[4]	86.6±1.01	ScSPM[4]	82.74±1.46
LLC[18]	87.93±1.04	LLC[18]	85.36±1.02
CSDL	**90.32±1.17**	**CSDL**	**86.54±0.56**

bocce,croquet, polo, rock climbing, rowing, sailing and *snow boarding*. The number in each category ranges from 137 to 250, and the total image number is 1792. We normalize the images to make the maximum side to be 400 pixels on this dataset. Following the commonly used setting [45], we also randomly select 70 images from each category as training data and select the 60 images as test data. We list the classification accuracy of different methods on these two datasets in Tabel V. We can see our method outperforms all the rest image classification methods, including LLC [18] which is the state-of-the-art feature coding technique for image classification, and achieves the best performance. This proves the effectiveness of our category-specific dictionary learning framework. Though our method achieves the best performance for basic-level object recognition and event classification, we notice that the improvement of our CSDL over ScSPM and LLC is less significant on these two datasets than that of fine-grained classification tasks. The reason is that for fine-grained image categorization, different categories are similar. Our method amplifies the difference, therefore can greatly outperform ScSPM and LLC. But for these two datasets, different categories are already significantly different, therefore ScSPM and LLC can easily achieve good performance, though our method further boosts the differences, the improvement is not that significant.

F. Convergence of The Algorithm

We alternatively update each component dictionary and the sparse codes, and each step can reduce the objective value, so the whole optimization process converges. Here we plot the change of the objective value with respect to the iterations of update on the PPMI+ (7 categories), Flower-17, Corel-10 and Event datasets in Fig. 10. It shows that the objective values decreases very fast. Here the iteration is the outer iteration in Algorithm 1. After only about 6-7 iterations, the objective value converges. Therefore we fix the stopping criteria in Algorithm 1 to be 15 for outer iteration on all the datasets.

V. CONCLUSION AND FUTURE WORK

To tackle the fine-grained image categorization, we propose to learn a category-specific dictionary for each category, and a shared dictionary for all the categories. Such category-specific dictionary can encode the subtle differences between different categories, and the shared dictionary can encode the common patterns. Our proposed framework also applies to other image classification tasks, like basic-level object recognition and event classification. Extensive experimental results demonstrate the effectiveness of our dictionary learning framework and solution.

Our method uses the label information to learn the dictionary, and [47] and [48] propose to learn the dictionary of sparse coding and SVM classifier together. So it is possible to jointly learn the category-specific dictionary and SVM together in our future work. Moreover, in future we can use more advanced feature pooling techniques [49][50] to further improve the performance of fine-grained image categorization.

REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[3] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the International Conference on Machine Learning*, 2007.

[4] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[5] S. Gao, I. W. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Transactons on Image Processing*, vol. 22, no. 2, pp. 423–434, 2013.

[6] K. Sohn, D. Y. Jung, H. Lee, and A. Hero III, "Efficient learning of sparse, distributed, convolutional feature representations for object recognition," in *Proceedings of 13th International Conference on Computer Vision*, 2011.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, pp. 273–297, 1995.

[9] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep., 2007.

[10] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3485–3492.

[11] L.-J. Li and L. Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[12] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 129–136.

[13] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *International Conference on Computer Vision*, 2011.

[14] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

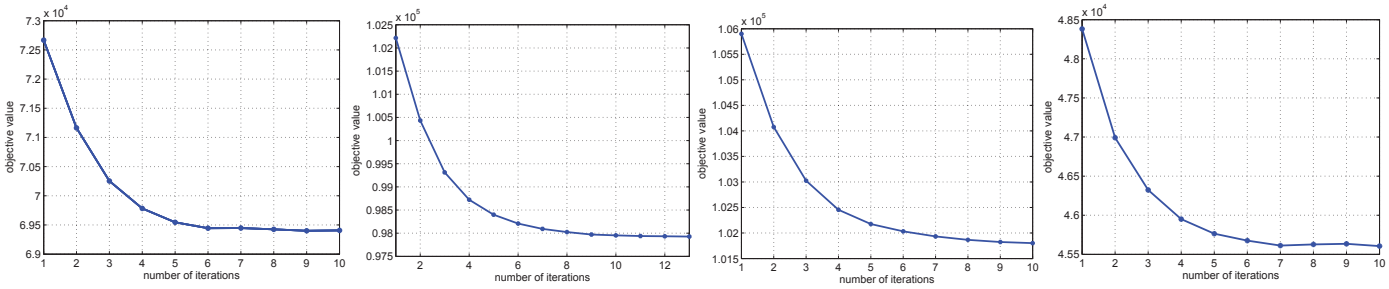
[15] ——, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[16] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[17] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[18] J. Wang, J. Yang, K. Yu, F. Lv, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[19] Y.-L. Boureau, J. Ponce, and Y. Lecun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the International Conference on Machine Learning*, 2010.



(a) The PPMI+ Dataset (a) The Flower-17 Dataset (a) The Corel-10 Dataset (a) The Event Dataset

Fig. 10. The change of the objective value with respect to the number of iterations on PPMI+ (7 categories), Oxford Flower-17, Corel-10, and Event datasets. The iteration is the outer iteration in Algorithm 1.

[20] X. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, 2012.

[21] A. Castrodad, Z. Xing, J. B. Greer, E. Bosch, L. Carin, and G. Sapiro, "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4263–4281, 2011.

[22] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.

[23] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[24] S. Kong and D. Wang, "A dictionary learning approach for classification: separating the particularity and the commonality," in *Proceedings of the European Conference on Computer Vision*, 2012.

[25] I. Ramirez, F. Lecumberry, and G. Sapiro, "Universal priors for sparse modeling," in *Computational Advances in MultiSensor Adaptive Processing CAMSAP 2009 3rd IEEE International Workshop on*, 2009, pp. 197–200.

[26] N. Zhou and J. Fan, "Learning inter-related visual dictionary for object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[27] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[28] B. Yao and L. Fei-Fei, "Grouplet: A structured image representation for recognizing human and object interactions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[29] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proceedings of the Conference on Neural Information Processing Systems*, 2006.

[30] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications,*, vol. 109, no. 3, pp. 475–494, 2001.

[31] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[32] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2479 – 2493, 2009.

[33] M. Aharon, M. Elad, and A. Bruckstein., "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4311–4322, November 2006.

[34] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the 7th international conference on Curves and Surfaces*, 2012.

[35] M. Aharon and M. Elad, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, pp. 3736–3745, December 2006.

[36] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[38] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[39] L. Zhang and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proceedings of the International Conference on Computer Vision*, 2011.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[41] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *IEEE 12th International Conference on Computer Vision*, 2009.

[43] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[44] Z. Lu and H. H. Ip, "Image categorization with spatial mismatch kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[45] J. Wu and J. M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[46] Z. Lu and H. H. Ip, "Image categorization by learning with context and consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[47] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proceedings of the Conference on Neural Information Processing Systems*, 2008.

[48] J. Y. J. Yang, K. Y. K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[49] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[50] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: Multi-way local pooling for image recognition," in *International Conference on Computer Vision*, 2011.