

Received June 25, 2020, accepted September 3, 2020, date of publication September 9, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023031

A Category-Specific Dictionary Learning Method Tailored for Reconstruction-Based Feature Coding

YE XU¹, LIHUA DUAN¹, XIAODONG YU², TIAN WANG², AND YINGZHONG SHI¹

¹School of IoT Technology, Wuxi Institute of Technology, Wuxi 214000, China

²School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213032, China

Corresponding author: Ye Xu (xuye@wxit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 31800392 and Grant 41901323, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under 18KJB520004, and in part by the Basic Research Plan for Application of Science and Technology Project in Changzhou City under Grant CJ20180038.


ABSTRACT Bag-of-Visual-Words (BoVW) is still a useful image classification model when there is not enough data to use Deep Learning. In BoVW model, the practice of reducing the reconstruction errors of local features can improve the classification accuracy owing to the decrease of information loss. Many reconstruction-based coding methods are proposed to learn a visual dictionary and encode local features via minimizing the reconstruction errors of local features with constraints. Besides this, the accuracy can also be improved by learning the category-specific dictionaries and then encoding features based on these dictionaries. By considering the two practices together, we propose a simple category-specific dictionary learning method tailored for reconstruction-based feature coding. Our method can be used as a universal one to improve the classification accuracies of many reconstruction-based coding methods, which is the highlight of our method. Concretely, a universal dictionary is learned by employing a reconstruction-based coding method and then refined for each category to obtain the category-specific dictionary of this category. When encoding a feature by a category-specific dictionary, the visual words for encoding it are decided in advance by the indices, which correspond to the non-zero elements of its coding vector obtained with the universal dictionary. The effectiveness of our method is validated by observing whether there is an accuracy improvement after applying our method. Our results on Scene-15, Caltech-101, and UIUC-Sports datasets show that the accuracies of four representative coding methods are improved by about 0.3% to 2.7%, which experimentally demonstrates the universality and effectiveness of our method.

INDEX TERMS Image classification, bag-of-visual-words, dictionary learning, reconstruction, category-specific.

I. INTRODUCTION

In decades, many methods for image classification have been presented in the field of computer vision. The challenges such as the change in viewpoint, illumination, partial occlusion, clutter, inter and intra-category visual diversity, make image classification a difficult task. Until now, there are two representative image classification models, i.e., Bag-of-Visual-Words (BoVW) and convolutional neural network (CNN), which have achieved many encouraging results in the past ten years.

BoVW model divides the process of converting an image to a vector into five stages [1]. In the beginning, image

The associate editor coordinating the review of this manuscript and approving it for publication was Junchi Yan .

patches are extracted from training images in a dense or random manner. Then, image patches are described as feature descriptors (local features) via statistical analysis over pixels of image patches. Scale-invariant feature transform (SIFT) [2] is widely used to describe image patches as 128-dimensional vectors. Next, a visual dictionary is learned using local features from training images by a learning algorithm such as K -means [3] or sparse coding [4]. After this, local features are encoded as coding vectors by the learned dictionary. In the end, all coding vectors are pooled together to form an image representation vector by maximum pooling or average pooling [5]. CNN [6] is a deep neural network of exploiting image space structure. It consists of convolutional layers, pooling layers, non-linear activations, and fully connected layers. Convolutional layers capture the

existence of various patterns, which are detected by different convolutional kernels. Pooling layers preserve the maximum saliencies in local regions by downsampling the convolutional layers. Fully connected layers are generally appended at the end, which simply represents a multi-layer perceptron. Non-linear activation functions are necessary to learn a complex function. A predominant difference between BoVW and CNN is that BoVW works with hand-crafted features such as SIFT and Histogram of Oriented Gradient (HOG), while CNN can extract automatically image features after training on a significant amount of data.

In recent years, CNN has achieved many superior results on some challenging image datasets such as ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [7]. However, when there is not enough training data, CNN shows a poor performance due to over-fitting [8]. To solve this, an effective method is to modify a CNN already trained on another large dataset, known as transfer learning [9]. The first layers of a pre-learned CNN are used as a mid-level feature extractor, and the last layers are modified to fit a certain target task. When training the modified CNN, only the weights of the modified layers are updated, while the weights of other layers are fixed. Some methods [10]–[12] based on transfer learning have reported obvious improvement over BoVW and achieved significant results in medical image classification.

Despite the significant effectiveness of transfer learning, its success depends on a pre-learned CNN. A pre-learned CNN requires a huge amount of data and time for training. It is worth noting that, the choice of the source dataset used for pre-learning a CNN and the number of the images in the target dataset influence the classification result [13]. If there is a large visual difference between the source dataset and the target dataset, negative transfer is likely to happen, resulting in poor performance. In addition, the number of the parameters of CNN is large for some popular CNNs, leading to considerable memory space consumption, such as 520MB for VGG-16 [14]. At the same time, BoVW model is a plug-and-play method that can be used without any prior initialization or very time-consuming training [15]. Hence, BoVW model might work well when dealing with some classification tasks that only provide a small amount of training data. Moreover, BoVW model has evolved in an understandable way in the past 15 years or so. By analyzing the target classification task, it is feasible to make use of human knowledge obtained to improve BoVW model from the aspects of feature extraction, feature description, dictionary learning, feature coding, and feature pooling. In view of this, for some simple tasks, BoVW model is probably capable of attaining satisfactory results. Besides, BoVW model can also be used jointly with CNN to acquire higher classification accuracies especially when training data are lacking, as done in a very recent research [16]. In consequence, we advocate the conventional yet effective BoVW model in this article.

In BoVW model, the practice of reducing the reconstruction errors of local features can improve classification accuracy owing to the decrease of information loss.

To this end, reconstruction-based coding methods are proposed to reconstruct local features via resolving a least-square optimization problem with constraints. Besides, at the training stage, the reconstruction-based coding method also learns a visual dictionary using local features extracted from training images. Nowadays, a number of reconstruction-based coding methods have been proposed, such as sparse coding [4], locality-constraint linear coding (LLC) [17], laplacian sparse coding (LSC) [18] and so on. The main difference among various reconstruction-based methods lies in the constraint. Except for this practice, another effective way of improving classification accuracy is to learn the category-specific visual dictionaries and then encoding features by the learned dictionaries. Various category-specific dictionary learning methods [19]–[22] have been presented in the last decade.

By considering the above two practices together, we propose a simple category-specific dictionary learning method tailored for reconstruction-based feature coding in this article. We aim to reduce the reconstruction errors of local features from positive samples and increase the errors of features from negative samples via encoding based on category-specific dictionaries. Specifically, a universal dictionary is learned by employing a reconstruction-based coding method and then refined for each category to obtain the category-specific dictionary of this category. For each category, its category-specific dictionary is learned only using the local features of this category. When encoding a local feature by a category-specific dictionary, the visual words for encoding it are decided in advance by the indices, which correspond to the non-zero elements of its coding vector obtained with the universal dictionary. Our method can be used as a universal one to improve the classification accuracies of many reconstruction-based coding methods theoretically, which is the highlight of our method. In this article, we apply our method on four representative coding methods, i.e., sparse coding, approximated LLC (aLLC) [17], LLC, and LSC. The effectiveness of our method is validated by observing whether there is an accuracy improvement after applying our method. We also investigate the computation time spent by our method to evaluate the practicability of our method. Besides, our method is carefully compared to a common method. By comparison, we observe that they have different performance characteristics according to the mixability of spatial distributions even if they follow very similar ideas of learning category-specific dictionaries, which is not yet illustrated in the existing works to our knowledge. The experiments are conducted on three small datasets, i.e., Scene-15 [3], Caltech-101 [23] and UIUC-Sports [24]. Our results show that the classification accuracies of the four coding methods are improved by about 0.3% to 2.7%, and the computation time spent by our method is acceptable. This phenomenon implies that our method is capable of improving the classification accuracies of many reconstruction-based coding methods with added yet acceptable computation time.

The remainder of this article is organized as follows: the proceeding section is about the related works.

Section III illustrates our work in detail. Experimental evaluation and analysis are reported in Section IV, and the conclusion is drawn in Section V.

II. RELATED WORKS

Many works have focused on dictionary learning in the past ten years. An early method calculates the clustering centers using local features from training images by K -means and takes each clustering center as a visual word [3]. To reduce the quantization errors of local features, the reconstruction-based coding method is proposed to learn a visual dictionary using local features from training images via resolving a least-square optimization problem with constraints. Different constraints result in different dictionaries. Sparse coding adds a l_1 -norm constraint on the coding vectors of local features to keep their sparsity. Wang *et al.* [17] added a locality constraint to project the local features into their local coordinate systems. Gao *et al.* [18] required spatially close and similar local features to have similar coding vectors. An extended work [25] to [18] considered the similarity among local regions instead of local features. Bengio *et al.* [26] encouraged that local features from the images of the same category are encoded with fixed visual words, by imposing a mixed-norm regularization.

Some works are devoted to learning category-specific dictionaries. In [19], the authors used Gaussian Mixture Model (GMM) to learn a universal dictionary and adapted it for each category to generate the category-specific dictionary of this category. Kong *et al.* [20] proposed a category-specific dictionary learning method named DL-COPAR, which aims at separating commonality and particularity. In their method, local features are encoded with the union of a universal dictionary and all category-specific dictionaries. In [21], the authors learned a universal dictionary and multiple category-specific dictionaries jointly by adding a discriminative constraint according to Fisher discrimination criterion. Gao *et al.* [22] also proposed to learn a universal dictionary and multiple category-specific dictionaries for fine-grained classification, by imposing cross-dictionary incoherent constraint and self-dictionary incoherent terms. Yang and Xiong [27] employed K -means and K -SVD to learn a category-specific dictionary for each category using the local features of this category. Based on the learned category-specific dictionaries, the authors proposed a kind of feature named category-sensitive saliency feature and used it to obtain image representation vectors. This method achieved comparable or better results in comparison to many advanced BoVW methods.

In order to preserve the relationship among neighboring local features, [28] and [29] proposed visual phrase and visual local graph, respectively. Accordingly, visual phrase dictionary and visual graph dictionary are learned in [28] and [29], respectively. In [5] and [30], the authors combined spatially close local features into joint features, and then learned a dictionary using joint features by sparse coding.

In recent years, Analysis Dictionary Learning (ADL) has shown excellent performance in image classification tasks, such as [31] and [32]. It is worth noting that ADL is often applied to image representation vectors to acquire more discriminative vectors. Therefore, the dictionary obtained by ADL is not used for encoding local features, which is different from the one in BoVW model.

III. OUR WORK

In this section, we first illustrate our work under the framework of BoVW model. Afterward, the detail on our method is presented clearly.

A. PROCESS OF IMAGE CLASSIFICATION

In the last decade, BoVW model has formed a unified framework consisting of five basic steps [1]. It includes image patch extraction, image patch description, dictionary learning, feature coding, and feature pooling. Our work only involves in dictionary learning and feature coding. Fig. 1 shows the process of image classification including our work. As shown in Fig. 1, the input image is classified through the six stages (a) to (f).

The first stage (a) is to extract the image patches from the input image. This process is implemented via sampling local areas of the image usually in a dense manner, e.g., the dense patches of 16×16 pixels with the step of 8 pixels.

Then, the image patches are described as the feature descriptors (local features) at the second stage (b). Many methods such as Scale-invariant Feature Transform (SIFT), Local Binary Pattern (LBP), and HoG, can be used for describing image patches. For example, SIFT is widely employed to describe image patches as 128-dimensional vectors.

Next, we obtain the indices of the visual words for encoding local features at the third stage (c). For each feature, its indices indicating which visual words are involved in the coding process are recorded by a 0-1 vector, where the 1 elements correspond to the non-zero coding coefficients of its coding vector obtained with a universal dictionary. The universal dictionary is learned using local features from training images via solving a least-square optimization problem with constraints (illustrated in Section III(B.2)).

Afterward, at the fourth stage (d), each local feature is encoded as C coding vectors by C category-specific dictionaries, respectively (illustrated in Section III(B.4)). At this stage, the visual words for encoding each feature are decided in advance by the 1 elements in its 0-1 vector. The category-specific dictionary of each category is learned by refining the universal dictionary with the local features from the training images of this category (illustrated in Section III(B.3)).

At the next stage (e), the coding vectors are aggregated into one vector by performing a pooling operation, such as maximum pooling and average pooling. In many existing methods, Spatial Pyramid Matching (SPM) [3] is widely used to incorporate the spatial information of the local features

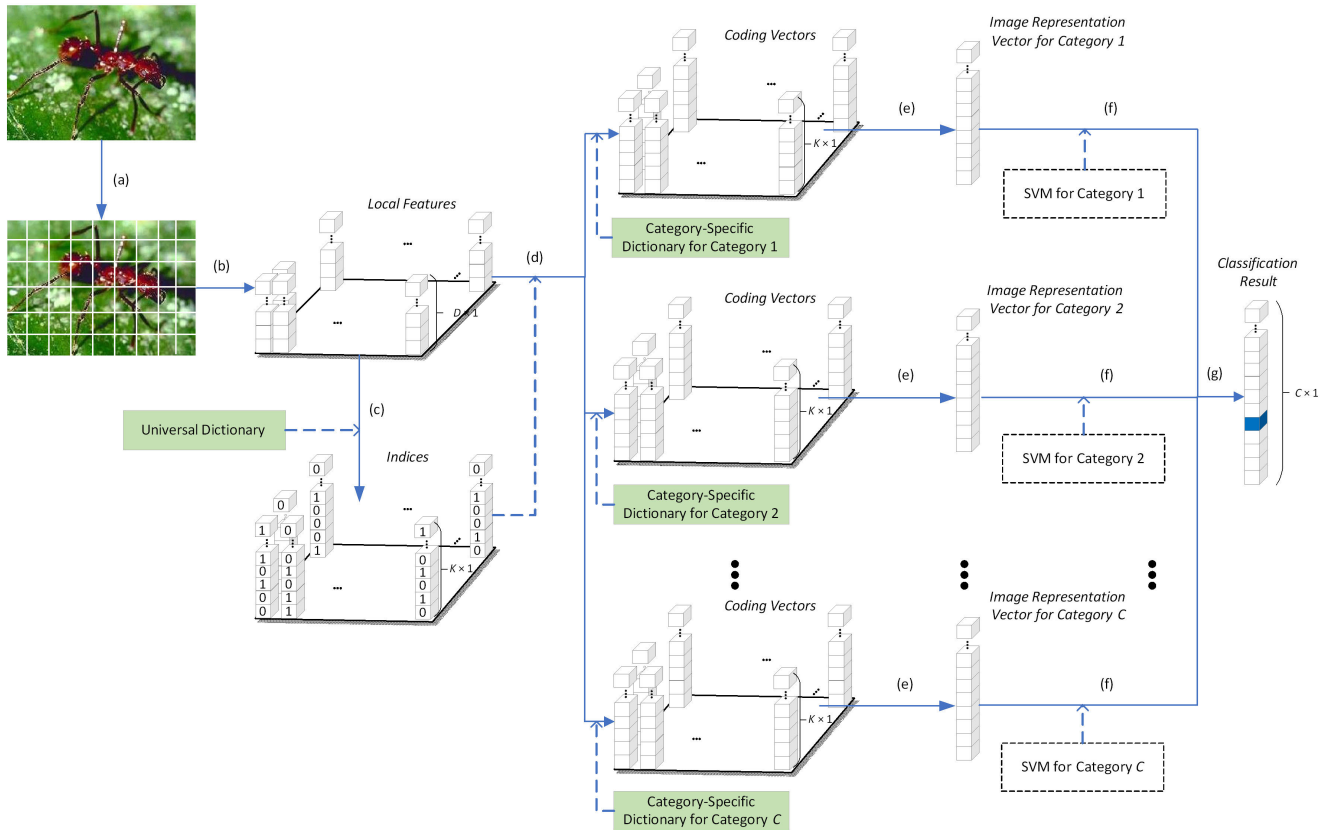


FIGURE 1. Process of image classification. (a) extracting the image patches from the input image; (b) describing the image patches as the D -dimensional feature descriptors; (c) for each descriptor, obtaining its 0-1 vector where the 1 elements correspond to the non-zero coding coefficients of its coding vector (obtained with a universal dictionary consisting of K visual words); (d) for each descriptor, encoding it as C coding vectors by C category-specific dictionaries; (e) pooling together the coding vectors of each category to form the image representation vector for this category; (f) classifying the i th ($i = 1, \dots, C$) image representation vector for the i th Support Vector Machine (SVM); (g) selecting the category indicated by the maximum as the category of the input image.

from an image into an image representation vector. It partitions the whole image region into the multiple blocks at the different resolutions levels of $1 \times 1, 2 \times 2$ and 4×4 . The coding vectors in each block are pooled together to form a pooling vector, and then the pooling vectors of all the blocks are concatenated into one vector, namely, image representation vector.

At the last stage (f), the image representation vector of each category is fed into the SVM trained for this category to obtain the score denoting how the input image belongs to this category. The category indicated by the maximum score is taken as the category of the input image. For any category, its SVM is a one-versus-rest linear SVM, which is trained on the image representation vectors obtained with the category-specific dictionary of this category. The training images of this category are taken as the positive samples and the training images of other categories are the negative samples.

B. CATEGORY-SPECIFIC DICTIONARY LEARNING

In theory, classification accuracy can be improved by this practice, i.e., reducing the reconstruction errors of the local

features extracted from positive samples and increasing the errors of the local features from negative samples in general. The reason is that the images restored from the coding vectors of positive samples become “clear” (information loss decreases), while the images restored from the coding vectors of negative samples become “blur” (information loss increases). Hence, the difference between positive and negative samples increases by this practice, which is beneficial to classification tasks.

To achieve this, we learn its category-specific dictionary for each category using the local features extracted from the training images of this category. The category-specific dictionary is obtained by resolving a least-square optimization problem. In this case, for any category, the visual words from its category-specific dictionary are specifically learned to minimize the reconstruction errors of the local features of this category. As a result, for the category-specific dictionary of any category, it tends to generate low reconstruction errors for the local features of this category and high reconstruction errors for the features of other categories.

Concretely, we first use the local features extracted from all training images to learn a universal dictionary via

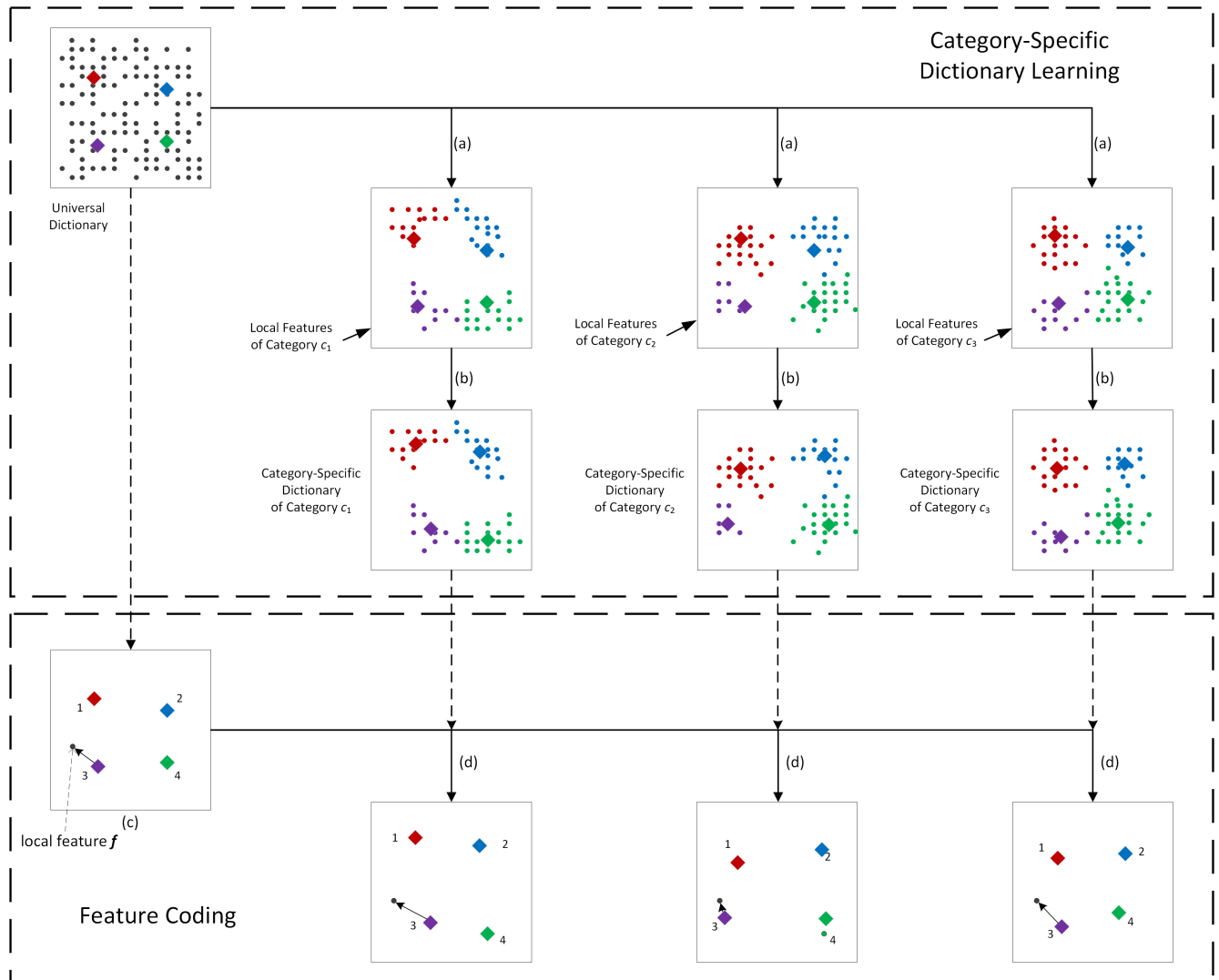


FIGURE 2. Toy example of category-specific dictionary learning and feature coding. In these square boxes denoting the same 2-dimensional feature space, the diamonds represent the visual words and the small circles are the local features. Each local feature is encoded only by the nearest word to it, and its color indicates which word encodes it. (a) for each local feature of each category, obtaining the index (denoted by a kind of color) of the nearest word to it; (b) refining the universal dictionary for each category using the local features of this category according to their indices. (c) for the local feature f , obtaining the index of the word from the universal dictionary for encoding it; (d) encoding the feature f with the word from each category-specific dictionary indicated by its index.

resolving a least-square problem P . Then, the universal dictionary is refined for each category to obtain the category-specific dictionary of this category, by resolving another least-square problem P' using the local features of this category. The problem P' is a non-convex function including two kinds of variables, i.e., dictionary and coding coefficients, thus they need to be solved alternatively. When solving the coding coefficients of a local feature, only the visual words indicated by its indices are used for encoding it. The indices of a local feature correspond to the non-zero coding coefficients of its coding vector, which is obtained by encoding the feature based on the universal dictionary. After attaining the category-specific dictionaries of all categories, a local feature is encoded in the following steps. Given a local feature f , firstly, it is encoded by the universal dictionary

as a coding vector. Afterward, the indices corresponding to the non-zero coefficients of its coding vector are recorded, which indicate which visual words of the universal dictionary encode f . At last, for each category, only the visual words from its category-specific dictionary indicated by the indices of f , are used to encode f to obtain the coding vector for this category. Fig. 2 illustrates the principle of our method by a toy example.

From the viewpoint of feature space, for a local feature, the visual words indicated by the non-zero elements in its coding vector (obtained with a universal dictionary), define a hyperplane it projects onto. Different coding methods give different projection schemes for the same feature. When solving the coding coefficients of a feature at the two stages of refining dictionary and encoding features by any

TABLE 1. Constraint $\phi(x)$ used in the four different reconstruction-based coding methods. $x(j)$ denotes the j th element in x .

Coding Methods	$\phi(x)$	subject to
Sparse Coding [4]	$\sum_{j=1}^K x(j) $	-
Non-negative Sparse Coding [33]	$\sum_{j=1}^K x(j) $	$x(j) \geq 0$
LLC [17]	$\sum_{j=1}^K x(j) \exp(\ f - b_j\ _2 / \sigma)$	$\sum_{j=1}^K x(j) = 1$
LSC [18]	$\sum_{j=1}^K x(j) + \frac{\beta}{2\lambda} \sum_m \ \mathbf{x} - \mathbf{x}_m\ ^2 W_m$	-

category-specific dictionary, the feature is always projected onto the “same” hyperplane, which is constructed by the visual words from a non-universal dictionary indicated by the non-zero elements in its coding vector (obtained with a universal dictionary).

In the following, the unified representation of reconstruction-based coding methods is given in Section III(B.1). On the basis of the unified representation, the details on how to learn a universal dictionary are presented in Section III(B.2). Our category-specific dictionary learning method is illustrated in Section III(B.3), and the coding method is explained in Section III(B.4).

1) UNIFIED REPRESENTATION OF RECONSTRUCTION-BASED CODING METHOD

The core idea of reconstruction-based coding is to reconstruct a feature with visual words via resolving a least-square optimization problem with constraints [1]. The unified representation of reconstruction-based coding can be generally written as

$$\arg \min_x G(x) = \arg \min_x \|f - Bx\|_2^2 + \lambda \phi(x), \quad (1)$$

where $f = [f_1, f_2, \dots, f_D]^T \in \mathbb{R}^{D \times 1}$ denotes a D -dimensional local feature, $B = [b_1, b_2, \dots, b_K] \in \mathbb{R}^{D \times K}$ denotes the visual dictionary consisting of K visual words, $x = [x_1, x_2, \dots, x_K] \in \mathbb{R}^{K \times 1}$ is the coding vector of the local feature f and λ balances the least-square term $\|f - Bx\|_2^2$ and the constraint term $\phi(x)$. The least-square term $\|f - Bx\|_2^2$ pursues accurate reconstruction, and the constraint term $\phi(x)$ makes the coding vector have a certain characteristic, for example, similar/different features obtain similar/different coding vectors [18]. The main difference among various reconstruction-based coding methods lies in the constraint term $\phi(x)$. Table. 1 lists the constraints of sparse coding [4], non-negative sparse coding [33], LLC [17] and LSC [18] as examples, which have different purposes as illustrated in Section II.

At the stage of dictionary learning, the reconstruction-based coding method is also used to learn a visual dictionary using the local features extracted from training images. In this case, the optimization problem includes not only the variable x but also the variable B . It can be written as

$$\arg \min_{X, B} G(X, B) = \arg \min_{X, B} \|F - BX\|_F^2 + \lambda \sum_{i=1}^N \phi(x_i, B), \quad (2)$$

where $F = [f_1, f_2, \dots, f_N] \in \mathbb{R}^{D \times N}$ are the local features extracted from training images, $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{K \times N}$. Since the target function $G(X, B)$ is a non-convex function, the variables X and B are solved alternatively. Specially, the constraint $\|b_k\|_2 \leq 1$ is required in some methods such as [17] and [18]. After obtaining the dictionary B , the coding vector x of local feature can be obtained by resolving formula (1).

2) UNIVERSAL DICTIONARY LEARNING

In our method, we need to learn a universal dictionary at first. Given a training image set with C categories, we extract a fixed-size set S of local features from the training images of each category, and gather all the sets $\{S_1, S_2, \dots, S_C\}$ into a set S^u for learning a universal dictionary. The set S^u is formed as the matrix F^u . The universal dictionary B^u is learned by resolving formula (2) with the input F^u .

3) CATEGORY-SPECIFIC DICTIONARY LEARNING

In order to obtain the category-specific dictionaries of all categories, the universal dictionary B^u is refined for each category using the local features of this category individually. There are two necessary steps, i.e., obtaining the indices of the visual words from B^u for encoding local feature, and learning the category-specific dictionaries $B_1^c, B_2^c, \dots, B_C^c$ via resolving a least-square problem.

Clearly, for the s th category, the set S_s of local features is extracted from the training images of the s th category. For each feature $f_{s,i}$ in S_s , its coding vector $x_{s,i}^u$ is calculated by resolving the formula (1) with the universal dictionary B^u . The 0-1 vector $v_{s,i}$ recording the indices of the visual words for encoding $f_{s,i}$, is obtained by:

$$v_{s,i}(j) = \begin{cases} 1, & \text{if } x_{s,i}^u(j) \neq 0 \\ 0, & \text{if } x_{s,i}^u(j) = 0, \end{cases} \quad (3)$$

where $v_{s,i}(j)$ is the j th element in $v_{s,i}$, and $x_{s,i}^u(j)$ is the j th coding coefficient in $x_{s,i}^u$.

After obtaining the 0-1 vectors of all the local features in S_s , the category-specific dictionary B_s^c of the s th category is learned by minimizing the below target function, which is written as:

$$\begin{aligned} \arg \min_{X_s^c, B_s^c} G(X_s^c, B_s^c) &= \arg \min_{X_s^c, B_s^c} \|F_s - B_s^c X_s^c\|_F^2 \\ \text{s.t. } \|b_{s,k}^c\|_2 &\leq 1, \quad k = 1, 2, \dots, K; \\ (1 - v_{s,i})^T x_{s,i}^c &= 0, \quad i = 1, 2, \dots, N_s; \end{aligned} \quad (4)$$

where F_s is the matrix form of S_s , $b_{s,k}^c$ is the k th visual word in B_s^c and N_s is the number of the local features in S_s . The constraint term $(1 - v_{s,i})^T x_{s,i}^c = 0$ ensures that each feature is encoded only by the words indicated by the 1 elements in its 0-1 vector. The variables X_s^c and B_s^c are solved alternatively. The universal dictionary B^u is used to initialize B_s^c . When B_s^c is fixed, the coding vector of each feature in S_s can be solved, respectively. For the i th feature $f_{s,i}$, its coding vector $x_{s,i}^c$ can be obtained by:

$$\arg \min_{\tilde{x}_{s,i}} G(\tilde{x}_{s,i}) = \arg \min_{\tilde{x}_{s,i}} \|f_{s,i} - \tilde{B}_{s,i}^c \tilde{x}_{s,i}\|_2^2, \quad (5)$$

where $\tilde{B}_{s,i}^c$ is a small dictionary consisting of the visual words from B_s^c indicated by the 1 elements in the 0-1 vector of $f_{s,i}$, and the element values in $\tilde{x}_{s,i}^c$ are the values in $x_{s,i}^c$ indicated by the 1 elements in its 0-1 vector. When X_s^c is fixed, B_s^c can be solved by a projection gradient descent algorithm. In the process of solving X_s^c , since the size (e.g., 5) of $\tilde{B}_{s,i}^c$ is far less than the size (e.g., 1024) of B_s^c , solving X_s^c is fast.

In this step, considering that almost all reconstruction-based coding methods include the least-square term $\|f - Bx\|_2^2$, we directly minimize this term (resolving the formula (4)) to learn a dictionary that provides the smallest lower bound of reconstruction error for F_s under the constraint $(1 - v_{s,i})^T x_{s,i}^c = 0$. Although encoding local feature is performed with constraints (as shown in Table. 1) at the stage of feature coding, the dictionary obtained by resolving formula (4) leads to lower reconstruction error compared with the universal dictionary (demonstrated in Section IV(C)).

4) FEATURE CODING

At the stage of feature coding, the universal dictionary B^u and all the category-specific dictionaries $B_1^c, B_2^c, \dots, B_C^c$ are used jointly to encode local features. Given a local feature f_i , firstly, its coding vector x_i^u is calculated by resolving the formula (1) with the universal dictionary B^u , and then its 0-1 vector v_i is obtained by the formula (3). Next, we use the category-specific dictionary of each category to encode the local feature. For the s th category, its coding vector $x_{i,s}^c$ for this category is attained by resolving the following formula with the category-specific dictionary B_s^c of this category.

$$\arg \min_{\tilde{x}_{i,s}} P(\tilde{x}_{i,s}) = \arg \min_{\tilde{x}_{i,s}} \|f_i - \tilde{B}_{i,s}^c \tilde{x}_{i,s}\|_2^2 + \lambda \phi(\tilde{x}_{i,s}), \quad (6)$$

where $\tilde{B}_{i,s}^c$ is a small dictionary consisting of the visual words from B_s^c indicated by the 1 elements in v_i , and the element values in $\tilde{x}_{i,s}^c$ are the values in $x_{i,s}^c$ indicated by the 1 elements in v_i . Providing there are C categories, C coding vectors will be generated for each local feature. In comparison to the computational time spent on encoding by the universal dictionary B^u , the time spent on encoding by category-specific dictionary B^c is much less owing to the small dictionary \tilde{B}^c constructed from B^c (demonstrated in Section IV(F)).

IV. EXPERIMENTS

A. DATASETS

In our experiments, three small datasets are used to evaluate the classification performance of our proposed method.

Scene-15: Scene-15 dataset consists of 15 scene categories. There are 4492 images in total. The number of images per category varies from 260 to 440. We consider 100 training images per category. The remaining images are used as testing images.

Caltech-101: Caltech-101 dataset is a challenging object recognition dataset, which contains 9,144 images in 101 object categories and one background category. The number of images per category ranges from 31 to 800. We choose randomly 30 training images from each category to form the training set, and up to 30 testing images from each category to form the testing set.

UIUC-Sports: It consists of 8 sport event categories. There are 1579 images in total, and each category has from 137 to 250 images. 70 and 60 images from each category are used for training and testing, respectively.

B. IMPLEMENTATION DETAILS

In this article, we apply our proposed method on four representative coding methods, i.e., sparse coding, aLLC, LLC and LSC. For aLLC, the visual words for encoding a local feature are the K -nearest words to it in feature space, which are decided by calculating the Euclidean distance between the feature and each word. Therefore, the indices of the words for encoding a feature are the indices of its K -nearest visual words from universal dictionary. For LLC, when encoding a feature f by any category-specific dictionary, the distance $\|f - b_i\|_2$ (as shown in Table. 1) needs to be recalculated. We only need to compute the distances from the feature to the words from category-specific dictionary indicated by the indices of f . For LSC, before encoding feature by the s th category-specific dictionary ($s = 1, 2, \dots, C$), the coding vector of each template feature (introduced in [18]) needs to be updated by the s th category-specific dictionary under the constraint $(1 - v_m)^T x_{m,s} = 0, m = 1, 2, \dots, M$, where $x_{m,s}$ is the coding vector of the m th template feature calculated for the s th category and M is the number of template features. All the updated coding vectors $\{x_{1,s}, \dots, x_{M,s}\}$ are employed to replace the vectors $\{x_1, \dots, x_M\}$ used in this term $\sum_m \|x - x_m\|^2 W_m$ (as shown in Table. 1) when encoding feature by the s th category-specific dictionary.

For images from all the datasets, we extract the dense patches of 16×16 pixels. The step between two neighboring patches is set to 8 pixels for Scenes-15 and UIUC-Sports, 6 pixels for Caltech-101. Each patch is described as a SIFT descriptor (128-dimensional vector). The dictionary size is set to 1024 for Scene-15 and UIUC-Sports, and 2048 for Caltech-101. For aLLC, K -means is employed to learn a universal dictionary. As suggested in [17], the number of the visual words to encode a local feature is set to 5. SPM is applied to incorporate spatial information of local features

into image representation vectors. For all the datasets, a one-versus-rest linear SVM for each category is trained. All the experiments are conducted on a 64-bit Windows 10 with Intel Core i5-4590 at 3.30 GHz * 4 on 16GB RAM.

In order to more accurately evaluate whether the classification accuracy is improved after applying our method, the following setups are taken. For each dataset, we randomly split it 6 times to obtain 6 training sets and 6 testing sets, and conduct experiment 6 times on these training sets and testing sets for each experimental setup. The average of the classification accuracies of 6 experiments is reported in this article. The accuracies obtained with universal dictionaries (like done in [4], [17], [18]), are treated as baselines for comparison.

C. RECONSTRUCTION ERROR

In this section, we investigate on Scenes-15 whether the reconstruction errors of local features are reduced after applying our method. To achieve this, ten thousand local features are randomly extracted from the images of each category, respectively, and two reconstruction errors are calculated for each feature. For a feature f_i from the s th category, its two errors $\|f_i - B^u x_i\|_2$ and $\|f_i - B_s^c x_{i,s}\|_2$, are computed, where x_i is obtained by resolving the formula (1) with B^u , and $x_{i,s}$ is obtained by resolving the formula (6) with the small dictionary $\tilde{B}_{i,s}^c$, which is built from B_s^c in terms of the indices of f_i .

For each kind of reconstruction error, the average of the errors of all the local features is computed and reported in Table. 2. As shown, the average errors of the four coding methods all decrease after applying our method. Among these coding methods, aLLC achieves the largest error drop since the universal dictionary used by aLLC is generated by K -means instead of minimizing a target function including the term $\|F - BX\|_F^2$.

TABLE 2. Comparison of reconstruction errors.

Coding Method	Error(universal dict.)	Error(category-specific dict.)
sparse coding	0.394	0.384
aLLC	0.403	0.389
LLC	0.396	0.387
LSC	0.400	0.392

We compare the two reconstruction errors of local features when sparse coding is used as a coding method. Fig. 3

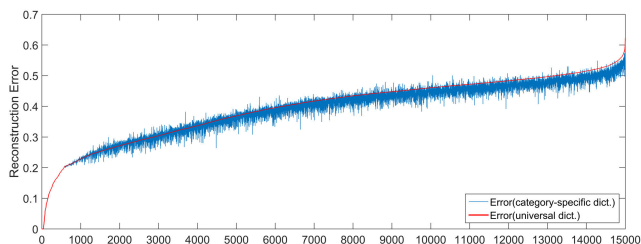


FIGURE 3. Comparison between the two reconstruction errors of 15000 local features.

shows the errors of 15000 local features (1000 features per category). Overall, the errors (universal dict.) of most local features decrease after applying our method. However, the errors (universal dict.) of about 3000 features increase a little, and about 600 features have no error drop. In addition, we also note that, for the local features with high reconstruction error (universal dict.) (e.g., 0.4 to 0.7), most of the features have relatively obvious error drop, and only a small number of features increase a little in reconstruction error.

We further investigate the reconstruction errors obtained when the local features of each category are encoded by the category-specific dictionary of every other category, as shown in Fig. 4. For the local features of the i th category, we compute their average error e_{ij} when they are encoded by the j th category-specific dictionary, and show the value $\max(e_{i1}, e_{i2}, \dots, e_{iC}) - e_{ij}$ at the i th row and the j th column. As shown, any element at the main diagonal is the brightest one at the row and the column it locates at. This phenomenon means that, for any category, its category-specific dictionary learned by our method tends to generate relatively low errors for the features of this category and relatively high errors for the features of other categories.

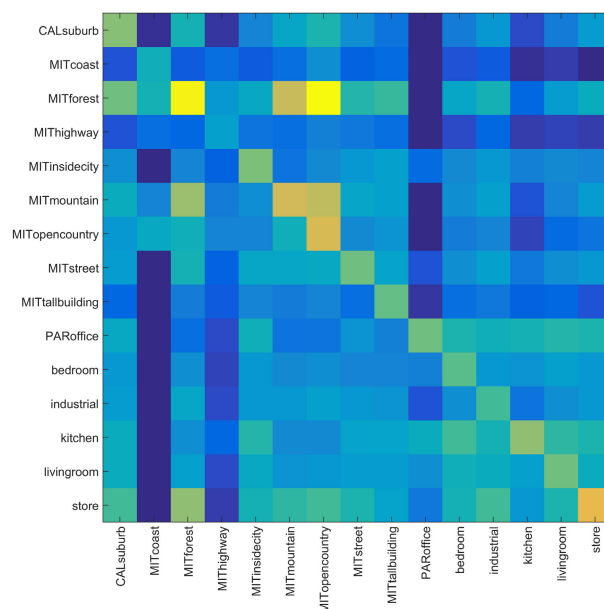


FIGURE 4. Heatmap of reconstruction errors. The brighter the small square, the lower the reconstruction error.

D. EFFECTIVENESS VALIDATION

In this section, we evaluate whether classification accuracy is improved after applying our method. Table. 3 reports the experimental results. As shown, for all the coding methods, the category-specific dictionaries learned by our method result in the better accuracies on the three datasets compared with the universal dictionaries. Despite that the different coding methods lead to the different classification accuracies, the accuracies are all improved a little after applying our

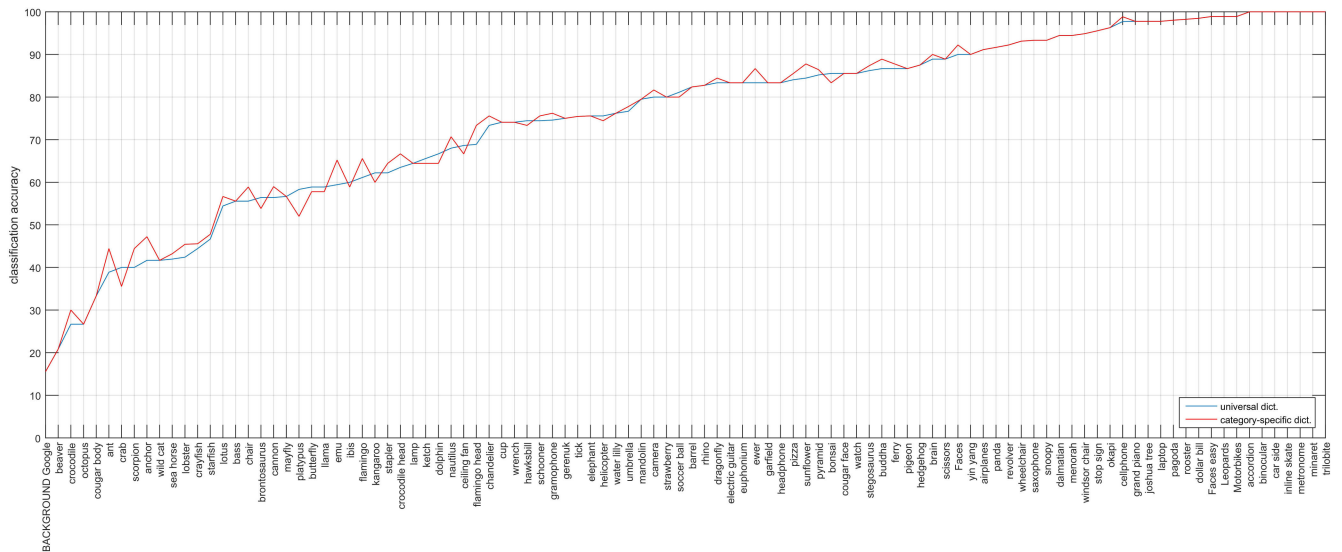


FIGURE 5. Comparison between the two classification accuracies of each category in Caltech-101.

TABLE 3. Effectiveness validation on Scene-15, Caltech-101, and UIUC-Sports.

Dataset	Coding Method	Dictionary	Accuracy, %
Scene-15	sparse coding	category-specific dict.	83.41(0.61)
		universal dict.	83.00(0.63)
	aLLC	category-specific dict.	83.07(0.87)
		universal dict.	80.36(0.36)
	LLC	category-specific dict.	83.40(0.38)
		universal dict.	82.99(0.39)
LSC	category-specific dict.	89.65(0.48)	
	universal dict.	89.22(0.44)	
Caltech-101	sparse coding	category-specific dict.	75.82(0.46)
		universal dict.	75.32(0.35)
	aLLC	category-specific dict.	74.29(0.49)
		universal dict.	72.35(0.38)
	LLC	category-specific dict.	74.66(0.30)
		universal dict.	74.10(0.35)
LSC	category-specific dict.	78.71(0.33)	
	universal dict.	78.40(0.31)	
UIUC-Sports	sparse coding	category-specific dict.	84.43(1.27)
		universal dict.	83.96(1.11)
	aLLC	category-specific dict.	84.21(1.01)
		universal dict.	83.39(0.81)
	LLC	category-specific dict.	84.32(1.18)
		universal dict.	83.90(1.08)
LSC	category-specific dict.	86.17(0.89)	
	universal dict.	85.82(0.91)	

method. This phenomenon demonstrates the universality and effectiveness of our method. Among these coding methods, aLLC always acquires the largest gain since it has the largest reconstruction error drop (as shown in Table. 2).

Based on sparse coding, we further investigate the classification accuracy of each category in Caltech-101. Fig. 5 reports the two accuracies of each category, which are obtained with the universal dictionary and the category-specific dictionaries, respectively. As shown, the accuracies of some categories are improved obviously such as “ant” (5.6%) and “anchor” (5.6%), but not all the categories have an accuracy improvement. Some categories

show an obvious drop such as “crab” (−4.4%) and “platypus” (−6.4%). Overall, the large accuracy improvements are almost achieved on the categories corresponding to the low accuracies (e.g. less than 60%). By analyzing the confusion matrix, we note that the category “crab” becomes more confused with the category “pizza” after applying our method. Fig. 6 lists some similar images of the two categories. It is easily found that the images of the category “crab” are similar to the ones of the category “pizza” to some extent. Each of these images has an ellipse shape. This means that they have some similar local features, which are extracted on the edges of the ellipses they all have. In this case, the category-specific dictionary B_{crab}^c refined for the category “crab” also reduces the reconstruction errors of the local features from the edges of ellipses. In other words, the edges of the ellipses in the “pizza” images (restored from the coding vectors obtained with B_{crab}^c), become “clear”, while the interior zones of the ellipses become “blur”. Consequently, the restored “pizza” images become more similar to some “crab” images, reducing the difference between the category “crab” and the category “pizza”.

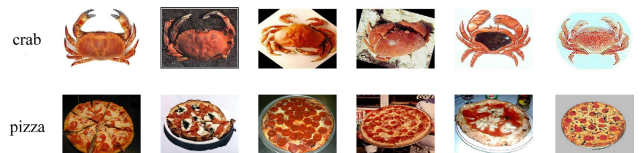


FIGURE 6. Similar images of the category “crab” and the category “pizza”.

E. COMPARISON WITH A COMMON METHOD

There is a common method (Com. Method) which follows a very similar idea to our method. In this method, the local features from the training images of each category are used

TABLE 4. Comparison among the Category-Specific Dictionary learned by Com. Method, the Category-Specific Dictionary learned by Our Method, and the Universal Dictionary.

Dataset	Coding Method	Dictionary	Accuracy,%	Reconstruction Error
Scene-15	sparse coding	category-specific dict.(our method)	83.41(0.61)	0.384
		category-specific dict.(com. method)	82.98(0.92)	0.387
		universal dict.	83.00(0.63)	0.394
	aLLC	category-specific dict.(our method)	83.07(0.87)	0.389
		category-specific dict.(com. method)	82.01(0.53)	0.394
		universal dict.	80.36(0.36)	0.403
Caltech-101	sparse coding	category-specific dict.(our method)	75.82(0.46)	0.343
		category-specific dict.(com. method)	76.19(0.49)	0.337
		universal dict.	75.32(0.35)	0.352
	aLLC	category-specific dict.(our method)	74.29(0.49)	0.358
		category-specific dict.(com. method)	74.59(0.56)	0.341
		universal dict.	72.35(0.38)	0.365
UIUC-Sports	sparse coding	category-specific dict.(our method)	84.43(1.27)	0.395
		category-specific dict.(com. method)	84.43(1.65)	0.398
		universal dict.	83.96(1.11)	0.406
	aLLC	category-specific dict.(our method)	84.21(1.01)	0.396
		category-specific dict.(com. method)	83.61(1.57)	0.401
		universal dict.	83.39(0.81)	0.410

to learn the category-specific dictionary of this category by directly solving the formula (2). The accuracies obtained by the common method on the three datasets are reported in Table. 4. Besides, the average reconstruction error of local features is also calculated as done in Section IV(C). LLC and LSC are not adopted in this section due to the huge computation time. As shown, the highest accuracy always corresponds to the lowest error. This means that the practice of reducing the reconstruction errors of local features is beneficial to image classification tasks. Besides, the category-specific dictionary (com. method) results in the best accuracy on Caltech-101 but performs worse than the category-specific dictionary (our method) on Scene-15. We also note that, for Scene-15, although the error (0.387) achieved by the category-specific dictionary (com. method) is smaller than the one (0.394) by the universal dictionary, its corresponding accuracy (82.98%) is still lower than the one (83.00%) achieved by the universal dictionary. This implies that only reducing reconstruction error not necessarily improves classification accuracy. The errors reported in Fig. 7 further supports this conclusion. The highest accuracy of each category in Scene-15 does not always correspond to the lowest reconstruction error.

Here, we report in Fig. 7 the three classification accuracies of each category in Scene-15 when sparse coding is used as a coding method. As shown, our method leads to better results than the common method on the indoor categories (e.g., PARoffice, bedroom, kitchen, living room, and store), whose images include a number of similar local features. This can be explained by that, in our method, when the local features of a category are encoded by the category-specific dictionary of this category (or any other category), the visual words indicated by their indices are specific to decrease (or increase) their reconstruction error overall. As shown in Fig. 2, for the representative local feature f of the category c_2 , the lowest error is obtained when it is encoded by the category-specific dictionary of the category c_2 .

In contrast, the common method does not have this trait. Compared with the accuracies obtained with the universal dictionary, the accuracies obtained by the common method decrease a little on the categories of “PARoffice”, “industrial”, “kitchen” and “store”. The reason is that the images of these categories include a large number of similar features, thus the dictionaries learned for these categories tend to reduce the reconstruction errors of these similar features, as a result, increasing the similarity among these categories instead. On the other hand, the common method performs better than our method on some categories such as MITforest and MITmountain. For anyone of the two categories, it has an obviously different visual presentation with other categories. In other words, the spatial distributions of the local features of the two categories both have a low mixability with the distributions of the features of other categories. Based on this observation, we infer that, for the features from positive samples and the features from negative samples, if their spatial distributions in feature space have a low mixability, the dictionary learned by the common method for positive samples, is better in reducing the reconstruction errors of the features from positive samples. As shown in Fig. 8, the category-specific dictionary learned by the common method gives rise to the lowest reconstruction errors for the features (denoted by the small triangles) from positive samples. In contrast, in our method, the index of the visual words for encoding local feature has no change in the refining process, as a result, restraining the refined dictionary to achieve lower error.

We also report in Fig. 9 the three classification accuracies of each category in Caltech-101. In comparison to the universal dictionary, the category-specific dictionary learned by our method increases the accuracies of 33 categories but decreases the accuracies of 14 categories, while the dictionary learned by the common method increases the accuracies of 45 categories but decreases the accuracies of 26 categories. Overall, although our method performs more stable than
















<p>CALsuburb</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>99.57</td><td>0.412</td></tr> <tr><td>com.</td><td>99.57</td><td>0.402</td></tr> <tr><td>our</td><td>99.86</td><td>0.397</td></tr> </table>	dict.	accu.	err.	uni.	99.57	0.412	com.	99.57	0.402	our	99.86	0.397	<p>MITcoast</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>86.08</td><td>0.339</td></tr> <tr><td>com.</td><td>85.92</td><td>0.332</td></tr> <tr><td>our</td><td>86.23</td><td>0.335</td></tr> </table>	dict.	accu.	err.	uni.	86.08	0.339	com.	85.92	0.332	our	86.23	0.335	<p>MITforest</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>96.23</td><td>0.461</td></tr> <tr><td>com.</td><td>96.67</td><td>0.461</td></tr> <tr><td>our</td><td>96.49</td><td>0.452</td></tr> </table>	dict.	accu.	err.	uni.	96.23	0.461	com.	96.67	0.461	our	96.49	0.452	<p>MIThighway</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>90.13</td><td>0.315</td></tr> <tr><td>com.</td><td>90.25</td><td>0.308</td></tr> <tr><td>our</td><td>90.75</td><td>0.310</td></tr> </table>	dict.	accu.	err.	uni.	90.13	0.315	com.	90.25	0.308	our	90.75	0.310	<p>MITinsidicity</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>85.87</td><td>0.405</td></tr> <tr><td>com.</td><td>86.63</td><td>0.396</td></tr> <tr><td>our</td><td>85.96</td><td>0.394</td></tr> </table>	dict.	accu.	err.	uni.	85.87	0.405	com.	86.63	0.396	our	85.96	0.394
dict.	accu.	err.																																																														
uni.	99.57	0.412																																																														
com.	99.57	0.402																																																														
our	99.86	0.397																																																														
dict.	accu.	err.																																																														
uni.	86.08	0.339																																																														
com.	85.92	0.332																																																														
our	86.23	0.335																																																														
dict.	accu.	err.																																																														
uni.	96.23	0.461																																																														
com.	96.67	0.461																																																														
our	96.49	0.452																																																														
dict.	accu.	err.																																																														
uni.	90.13	0.315																																																														
com.	90.25	0.308																																																														
our	90.75	0.310																																																														
dict.	accu.	err.																																																														
uni.	85.87	0.405																																																														
com.	86.63	0.396																																																														
our	85.96	0.394																																																														
<p>MITmountain</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>90.88</td><td>0.397</td></tr> <tr><td>com.</td><td>91.02</td><td>0.392</td></tr> <tr><td>our</td><td>89.49</td><td>0.387</td></tr> </table>	dict.	accu.	err.	uni.	90.88	0.397	com.	91.02	0.392	our	89.49	0.387	<p>MITopencountry</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>72.77</td><td>0.388</td></tr> <tr><td>com.</td><td>72.39</td><td>0.388</td></tr> <tr><td>our</td><td>73.03</td><td>0.381</td></tr> </table>	dict.	accu.	err.	uni.	72.77	0.388	com.	72.39	0.388	our	73.03	0.381	<p>MITstreet</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>91.56</td><td>0.398</td></tr> <tr><td>com.</td><td>91.87</td><td>0.394</td></tr> <tr><td>our</td><td>91.87</td><td>0.389</td></tr> </table>	dict.	accu.	err.	uni.	91.56	0.398	com.	91.87	0.394	our	91.87	0.389	<p>MITtallbuilding</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>92.66</td><td>0.377</td></tr> <tr><td>com.</td><td>92.34</td><td>0.369</td></tr> <tr><td>our</td><td>92.19</td><td>0.367</td></tr> </table>	dict.	accu.	err.	uni.	92.66	0.377	com.	92.34	0.369	our	92.19	0.367	<p>PARoffice</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>92.00</td><td>0.385</td></tr> <tr><td>com.</td><td>91.30</td><td>0.365</td></tr> <tr><td>our</td><td>92.87</td><td>0.371</td></tr> </table>	dict.	accu.	err.	uni.	92.00	0.385	com.	91.30	0.365	our	92.87	0.371
dict.	accu.	err.																																																														
uni.	90.88	0.397																																																														
com.	91.02	0.392																																																														
our	89.49	0.387																																																														
dict.	accu.	err.																																																														
uni.	72.77	0.388																																																														
com.	72.39	0.388																																																														
our	73.03	0.381																																																														
dict.	accu.	err.																																																														
uni.	91.56	0.398																																																														
com.	91.87	0.394																																																														
our	91.87	0.389																																																														
dict.	accu.	err.																																																														
uni.	92.66	0.377																																																														
com.	92.34	0.369																																																														
our	92.19	0.367																																																														
dict.	accu.	err.																																																														
uni.	92.00	0.385																																																														
com.	91.30	0.365																																																														
our	92.87	0.371																																																														
<p>bedroom</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>67.76</td><td>0.395</td></tr> <tr><td>com.</td><td>68.10</td><td>0.385</td></tr> <tr><td>our</td><td>68.28</td><td>0.382</td></tr> </table>	dict.	accu.	err.	uni.	67.76	0.395	com.	68.10	0.385	our	68.28	0.382	<p>industrial</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>69.00</td><td>0.400</td></tr> <tr><td>com.</td><td>67.20</td><td>0.394</td></tr> <tr><td>our</td><td>69.19</td><td>0.390</td></tr> </table>	dict.	accu.	err.	uni.	69.00	0.400	com.	67.20	0.394	our	69.19	0.390	<p>kitchen</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>73.09</td><td>0.392</td></tr> <tr><td>com.</td><td>72.55</td><td>0.376</td></tr> <tr><td>our</td><td>73.45</td><td>0.378</td></tr> </table>	dict.	accu.	err.	uni.	73.09	0.392	com.	72.55	0.376	our	73.45	0.378	<p>livingroom</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>58.62</td><td>0.412</td></tr> <tr><td>com.</td><td>61.16</td><td>0.405</td></tr> <tr><td>our</td><td>62.01</td><td>0.401</td></tr> </table>	dict.	accu.	err.	uni.	58.62	0.412	com.	61.16	0.405	our	62.01	0.401	<p>store</p>  <table border="1"> <tr><td>dict.</td><td>accu.</td><td>err.</td></tr> <tr><td>uni.</td><td>78.79</td><td>0.445</td></tr> <tr><td>com.</td><td>77.67</td><td>0.438</td></tr> <tr><td>our</td><td>79.53</td><td>0.430</td></tr> </table>	dict.	accu.	err.	uni.	78.79	0.445	com.	77.67	0.438	our	79.53	0.430
dict.	accu.	err.																																																														
uni.	67.76	0.395																																																														
com.	68.10	0.385																																																														
our	68.28	0.382																																																														
dict.	accu.	err.																																																														
uni.	69.00	0.400																																																														
com.	67.20	0.394																																																														
our	69.19	0.390																																																														
dict.	accu.	err.																																																														
uni.	73.09	0.392																																																														
com.	72.55	0.376																																																														
our	73.45	0.378																																																														
dict.	accu.	err.																																																														
uni.	58.62	0.412																																																														
com.	61.16	0.405																																																														
our	62.01	0.401																																																														
dict.	accu.	err.																																																														
uni.	78.79	0.445																																																														
com.	77.67	0.438																																																														
our	79.53	0.430																																																														

FIGURE 7. Comparison of the three classification accuracies of each category in Scene-15. (uni.: universal dictionary, com.: Com. Method, our: our method, accu.: classification accuracy, err.: reconstruction error).

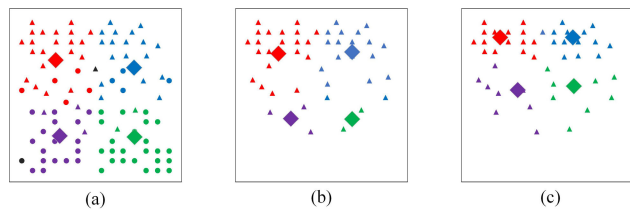


FIGURE 8. Toy example of a universal dictionary, a category-specific dictionary (our method), and a category-specific dictionary (Com. Method). The features denoted by the small triangles represent the ones extracted from positive samples, and the small circles indicate the ones extracted from negative samples. The diamonds represent the visual words. Each local feature is encoded only by the nearest word to it, and its color indicates which word encodes it. (a) the universal dictionary learned using all the features; (b) the category-specific dictionary learned by our method using the features from positive samples. (c) the category-specific dictionary learned by Com. Method using the features from positive samples.

the common method, the average classification accuracy of the common method is higher than our method. This phenomenon can be explained as follows. For most of the

categories in the object recognition dataset Caltech-101, each of the images of these categories includes only one object, as shown in Fig. 10. This means that the spatial distributions of the local features of these categories in feature space have relatively low mixability compared with Scene-15. In this case, the common method can achieve low reconstruction errors for the features highly relevant to these object categories. Consequently, the common method performs better on Caltech-101 than on Scene-15. Nevertheless, some categories have a considerable accuracy drop after applying the common method, as shown in Fig. 10. We find by analyzing that, these categories have a large intra-category visual diversity (e.g., mayfly), or a large change in viewpoint (e.g., water lily), or only a small number of key features highly relevant to object category (e.g., strawberry). This means that, for anyone of these categories, the dictionary learned using the features from training images, is not capable enough of reducing the errors of the features from testing images, as a result, the images (restored from coding vectors) of this category

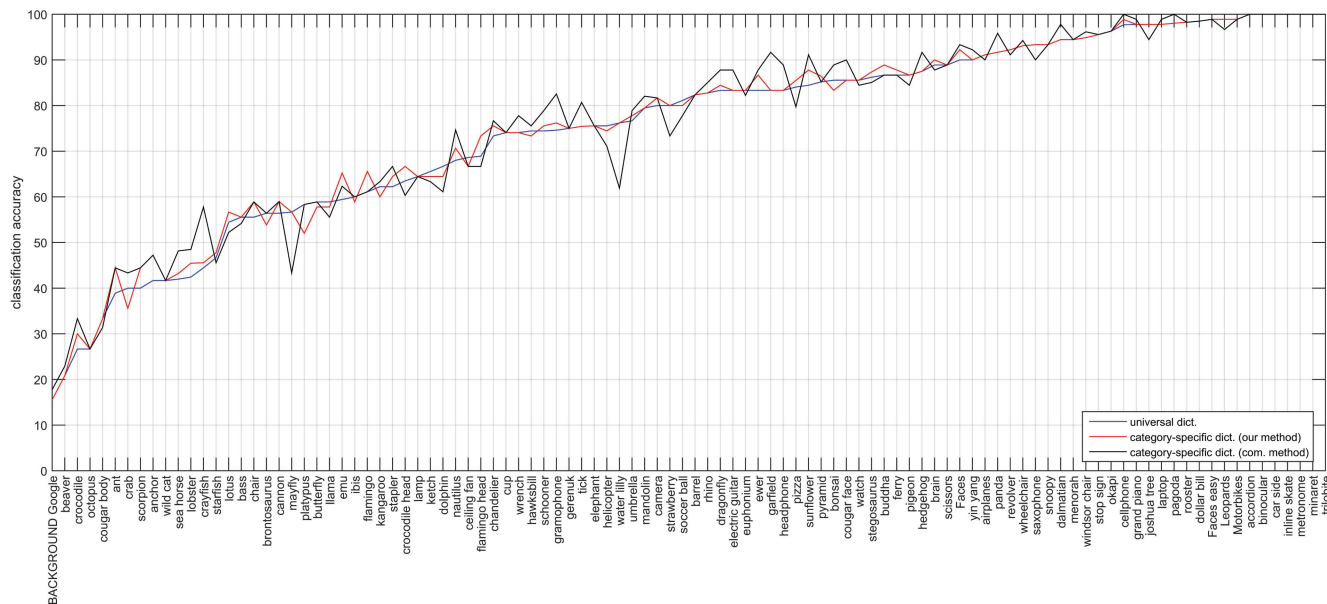


FIGURE 9. Comparison of the three classification accuracies of each category in Caltech-101.

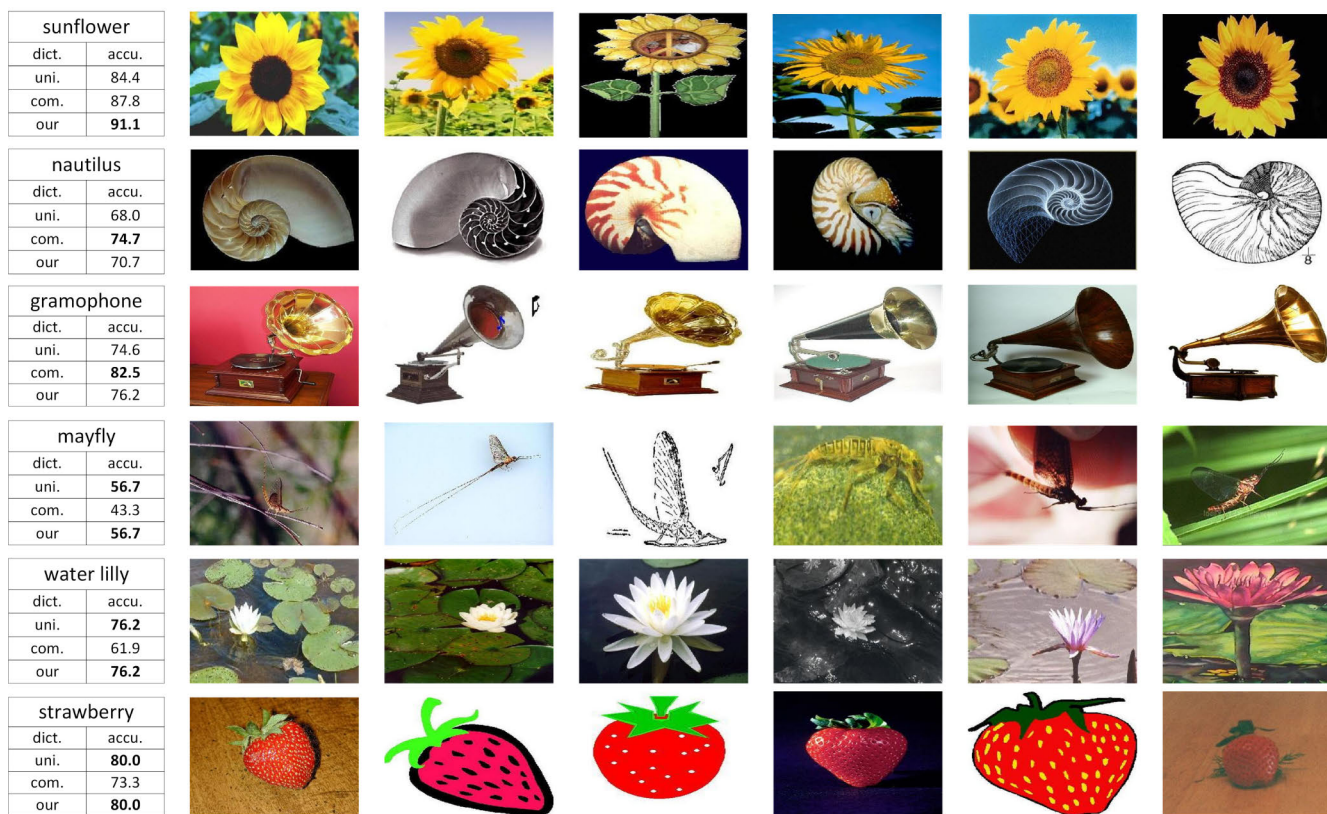


FIGURE 10. Example images from Caltech-101. (uni.: universal dictionary, com.: Com. Method, our: our method, accu.: classification accuracy)

and other categories all become more “blur”, i.e., increasing the similarity of this category with others.

By the above analysis, it can be concluded that, despite that our method and the common method follow the very

similar ideas of learning category-specific dictionary, they have different performance characteristics according to the mixability of the two spatial distributions of the features from positive samples and the features from negative samples.

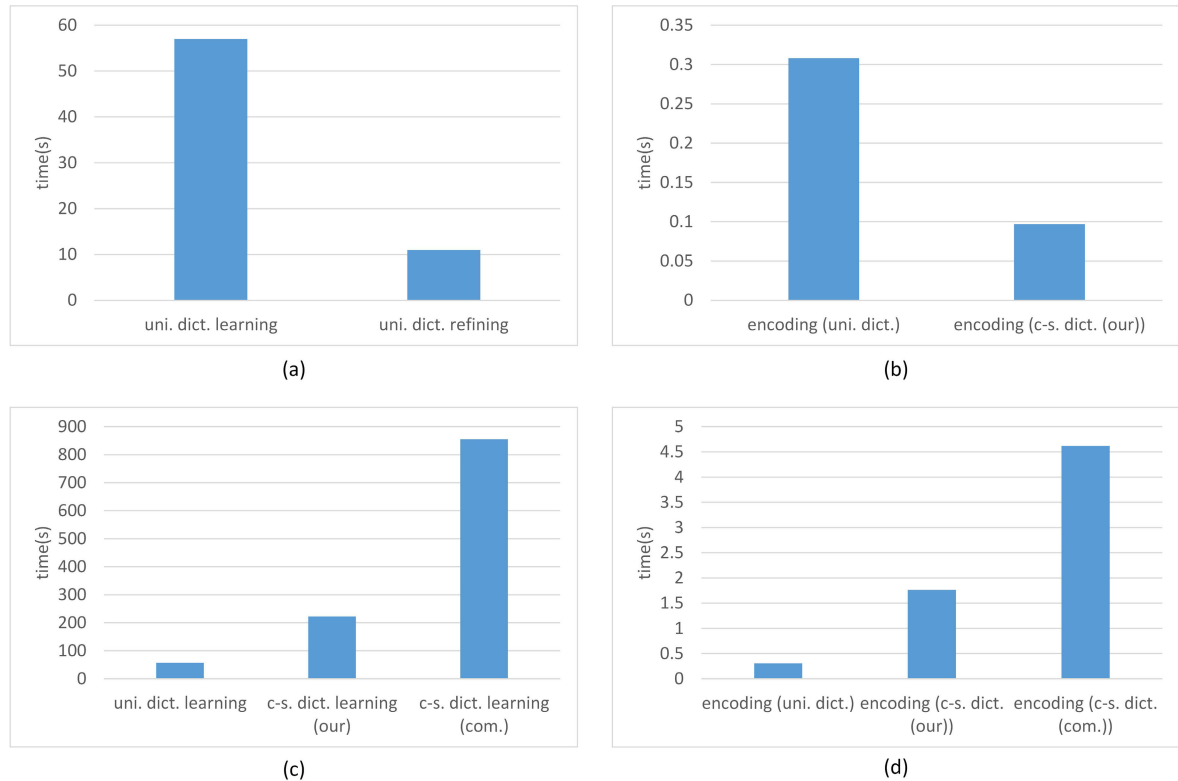


FIGURE 11. Comparison on the computation time spent on dictionary learning and feature coding. Fig. 11(a) compares the time used for learning a universal dictionary and the time used for refining the universal dictionary. Fig. 11(b) reports the average time spent on encoding the features from an image by a universal dictionary (resolving the formula (1)), and by a category-specific dictionary (resolving the formula (6)). Fig. 11(c) lists the time spent on learning a universal dictionary, learning the category-specific dictionaries of all the categories by our method, and learning the category-specific dictionaries of all the categories by Com. Method. Fig. 11(d) reports the average time consumed on encoding the features from an image by the universal dictionary, by all the category-specific dictionaries learned by our method, and by all the category-specific dictionaries learned by Com. Method.

By and large, our method can perform better than the common method when the mixability is high, and the common method can be good when the mixability is low.

F. COMPUTATION TIME

In this section, the computation time spent on dictionary learning and feature coding is investigated separately on Scene-15 when sparse coding is used as a coding method. The results are shown in Fig. 11.

As shown in Fig. 11(a) and Fig. 11(b), the time spent on learning a category-specific dictionary is much less than learning a universal dictionary, and the time spent on encoding feature by a category-specific dictionary is also obviously less than by a universal dictionary. The reason is that, when refining a universal dictionary (learning a category-specific dictionary) and encoding local features, small dictionaries are constructed according to the indices of features to solve coding coefficients. However, our method needs to learn its category-specific dictionary for each category and then encode features by all the category-specific dictionaries. For our method, the time spent on dictionary learning is the sum of the time on universal dictionary learning and the time on category-specific dictionary learning, the time spent on

encoding the features of an image is the sum of the time on obtaining the indices of the features and the time on encoding the features by all the category-specific dictionaries. Hence, the time consumed by our method on dictionary learning and feature coding increases a lot, as shown in Fig. 11(c) and Fig. 11(d). Nonetheless, the time required by our method is much less than Com. Method. The reason is that, for Com. Method, the time spent on learning a category-specific dictionary for each category is almost the time on learning a universal dictionary, and the time on encoding by a category-specific dictionary is almost the time on encoding by a universal dictionary. For our method and the common method, the time spent on dictionary learning and feature coding increase linearly as the number of categories increases.

G. ACCURACY COMPARISON WITH OTHER METHODS

In this section, we compare the classification accuracies achieved by our method and other methods, including the deep learning methods (e.g., [34], [36], [38], [39]) and the BoVW methods (e.g., [27], [35]). The results of these methods are obtained without the help of transfer learning, i.e., using a pre-learned CNN to extract image features.

TABLE 5. Classification accuracy on Scene-15.

Method	Accuracy, %
Our(sparse coding)	83.41(0.61)
Our(aLLC)	83.07(0.87)
Our(LLC)	83.40(0.38)
Our(LSC)	89.65(0.48)
CDBN [34]	78.52(0.63)
Y.L. Boureau et al. [35]	83.1(0.7)
SS-RBM [36]	84.1(0.8)
H. Goh et al. [37]	85.2(0.5)
Y.L. Boureau et al. [5]	85.6(0.2)
CSSC+ [27]	86.01(0.5)

TABLE 6. Classification accuracy on Caltech-101.

Method	Accuracy, %
Our(sparse coding)	75.82(0.46)
Our(aLLC)	74.29(0.49)
Our(LLC)	74.66(0.30)
Our(LSC)	78.71(0.33)
CDBN [34]	65.4(0.5)
CNN [38]	66.3(1.5)
Deconvolutional Network [39]	66.9(1.1)
Hierarchical SC [40]	74.0(1.5)
SS-RBM [36]	75.1(1.2)
NBNN kernel [41]	75.2(1.2)
SCDAE [42]	78.6(1.2)
CSSC+ [27]	79.8(1.0)

TABLE 7. Classification accuracy on UIUC-Sports.

Method	Accuracy, %
Our(sparse coding)	84.43(1.27)
Our(aLLC)	84.21(1.01)
Our(LLC)	84.32(1.18)
Our(LSC)	86.17(0.89)
R-FCN [16]	77.12
SPPNet [16]	78.86
LPR-RBF [16]	86.2
CSSC+ [27]	88.66(0.5)
ResNet [16]	89.45
X. Li et al. [43]	90.00(0.7)

As shown in Table. 3 and Tables. 5-7, the classification performance of our method relies on the performance of the coding method adopted in our method. Our method achieves the high accuracies on the three datasets when adopting LSC as a coding method. Compared with other methods, although the accuracies obtained by our method are not attractive enough, our method is easy to be combined with a number of BoVW methods to obtain higher classification accuracy (illustrated in Section III(H)). We also note that the deep learning methods do not achieve obvious improvement over the BoVW methods due to the lack of training data.

H. DISCUSSION

The characteristics of our method are listed as follows:

- Our method can be used as a universal method to improve the classification accuracies of many reconstruction-based coding methods with added yet acceptable computation time. Except for the four coding methods (i.e., sparse coding, aLLC, LLC, and LSC)

adopted in our work, many coding methods such as extending LSC [25], nonnegative sparse coding [33], hierarchical sparse coding [40], local coordinate coding [44] and so on, can also have an accuracy improvement by applying our method, theoretically.

- Despite the universality and effectiveness of our method, there is computation time associated with dictionary learning and feature coding, as shown in Fig. 11. However, compared with the common method, the time required by our method is much less. This trait ensures the practicability of our method.
- Although the classification accuracies obtained by our method are not attractive enough in comparison to some existing methods, our method is easy to be combined with a number of BoVW methods to achieve higher classification accuracy. The reason is that it is just a reinforcement method tailored for reconstruction-based coding methods, and it only involves in the stages of dictionary learning and feature coding. Therefore, In addition to employing more advanced coding methods, the advanced methods focusing on feature extraction, feature description, and feature pooling can also be used jointly to further improve the classification accuracy. Besides, the works on Analysis Dictionary Learning (ADL) can also be applied to the image representation vectors obtained by our method, resulting in the more discriminative image representation vectors.
- Some works can benefit from our method owing to the relatively less computation time consumed on category-specific dictionary learning. For example, the method proposed in [27] achieves comparable results to many advanced BoVW methods, but it suffers from the huge computation time spent on the category-specific dictionary learning. In this method, the category-specific dictionary of each category is learned by K -means or K -SVD using the local features of this category. Therefore, our method can be used to replace the category-specific dictionary learning method adopted in [27], as a result, improving the practicability of [27]. Besides, some methods such as [45] and [46] learn block-specific dictionaries for the blocks divided by SPM. For each block, its block-specific dictionary is learned by applying a reconstruction-based method on the local features extracted in this block. Similarly, the time consumed on block-specific dictionary learning can be reduced by a modified version of our method. In this version, the universal dictionary is refined for each block instead of each category.

V. CONCLUSION

In this article, we proposed a category-specific dictionary learning method tailored for reconstruction-based feature coding. The category-specific dictionary of each category was learned by refining a universal dictionary using the local features of this category. When encoding a local feature by a category-specific dictionary, the visual words for encoding it

are decided in advance by the indices, which correspond to the non-zero elements of its coding vector obtained with the universal dictionary. Our results on three small datasets show that the classification accuracies of four representative coding methods were improved by about 0.3% to 2.7%, which experimentally demonstrates the universality and effectiveness of our method. Furthermore, by comparing our method with a common method, we found that they have different performance characteristics according to the mixability of the two spatial distributions of the features from positive samples and the features from negative samples. By and large, our method can perform better than the common method when the mixability is high, and the common method can be good when the mixability is low. The future works we are pursuing are: 1) taking into account the discriminability of local features when refining a universal dictionary, to make features with high discriminability have low reconstruction errors; 2) applying the thought of our method to transfer learning.

REFERENCES

- [1] Y. Huang, Z. Wu, L. Wang, and T. Tang, "Feature coding in image classification: A comprehensive study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–505, Mar. 2014.
- [2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.
- [4] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.
- [5] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.
- [6] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [8] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.
- [11] M. Hussain, J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," in *Proc. Adv. Comput. Intell. Syst.*, 2018, pp. 191–202.
- [12] V. Cheplygina, "Cats or CAT scans: Transfer learning from natural or medical image source data sets?" *Current Opinion Biomed. Eng.*, vol. 9, pp. 21–27, Mar. 2019.
- [13] T. Schlegl, J. Ofner, and G. Langs, "Unsupervised pre-training across image domains improves lung tissue classification," in *Proc. Int. Conf. Med. Comput. Vis.*, 2014, pp. 82–93.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1–14.
- [15] S. A. Vassou, N. Anagnostopoulos, A. Amanatiadis, K. Christodoulou, and S. A. Chatzichristofis, "Unsupervised pre-training across image domains improves lung tissue classification," in *Proc. Int. Conf. Med. Comput. Vis.*, 2014, pp. 82–93.
- [16] A. G. Sorkhi, H. Hassanpour, and M. Fateh, "A comprehensive system for image scene classification," *Multimedia Tools Appl.*, vol. 79, no. 25, pp. 18033–18058, Feb. 2020.
- [17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.
- [18] S. Gao, I. Tsang, L. Chia, and P. Zhao, "Local features are not lonely—Laplacian sparse coding for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 3555–3561.
- [19] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1243–1256, Jul. 2008.
- [20] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *Proc. Int. Conf. Eur. Conf. Comput. Vis.*, 2012, pp. 186–199.
- [21] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3490–3497.
- [22] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.
- [23] F. F. Li, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1134–1141.
- [24] L. J. Li and F. F. Li, "What, where and who? Classifying events by scene and object recognition," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [25] M. Mejdoub, M. Dammak, and C. B. Amar, "Extending Laplacian sparse coding by the incorporation of the image spatial context," *Neurocomputing*, vol. 166, pp. 44–52, Oct. 2015.
- [26] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 82–89.
- [27] Z. Yang and H. Xiong, "Image classification based on saliency coding with category-specific codebooks," *Neurocomputing*, vol. 184, pp. 188–195, Apr. 2016.
- [28] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, "Generating descriptive visual words and visual phrases for large-scale image applications," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2664–2677, Sep. 2011.
- [29] M. Dammak, M. Mejdoub, and C. Ben Amar, "Histogram of dense subgraphs for image representation," *IET Image Process.*, vol. 9, no. 3, pp. 184–191, Mar. 2015.
- [30] N. Morioka and S. Satoh, "Learning directional local pairwise bases with sparse coding," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 1–11.
- [31] W. Tang, A. Panahi, H. Krim, and L. Dai, "Analysis dictionary learning: An efficient and discriminative solution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019.
- [32] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.
- [33] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma, "Image classification by non-negative sparse coding, low-rank and sparse decomposition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1673–1680.
- [34] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.
- [35] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: Multi-way local pooling for image recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2651–2658.
- [36] G. Anlin, T. Nicolas, C. Matthieu, and L. Joo-Hwee, "Unsupervised and supervised visual codes with restricted Boltzmann machines," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 298–311.
- [37] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2212–2225, Dec. 2014.
- [38] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1605–1612.
- [39] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
- [40] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1713–1720.

[41] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, "The NBNN kernel," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1824–1831.

[42] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2017.

[43] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4204–4212, May 2019.

[44] K. Yu, T. Wang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2223–2231.

[45] H. Luo, M. Guo, and F. Kong, "An image classification method based on multiple level spatial visual dictionary ensemble," *Acta Electronica Sinica*, vol. 43, no. 4, pp. 684–693, 2015.

[46] C. Zhang, S. Wang, Q. Huang, J. Liu, C. Liang, and Q. Tian, "Image classification using spatial pyramid robust sparse coding," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 1046–1052, Jul. 2013.



XIAODONG YU received the B.S. degree in applied physics from the Nanjing University of Information Science and Technology, Nanjing, China, in 2006, and Ph.D. degree in control science and engineering from Jiangnan University, in 2017. He has worked with the College of IoT Engineering, Nanjing University of Information Science and Technology. His research interests include computer vision, graph mining, and pattern recognition.



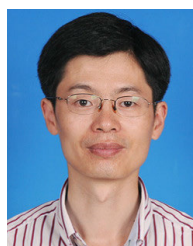
YE XU received the B.S. degree in communication engineering from Anqing Normal University, Anqing, China, in 2010, and the Ph.D. degree in signal and information processing from the Communication University of China, Beijing, China, in 2017. He is currently a Lecturer with the School of IoT Technology, Wuxi Institute of Technology. His research interests include computer vision, pattern recognition, and deep learning.



TIAN WANG received the M.S. degree from Jiangsu Normal University, China, in 2012, and the Ph.D. degree from Beijing Forestry University, in 2017. She is currently a Lecturer with the College of Computer and Information Engineering, Changzhou Institute of Technology, China. She has worked on remote sensing and geographic information systems and their applications in ecology. Her research interests include landscape pattern optimization, assessment of eco-service function, and the Internet of Things Technology.



LIHUA DUAN received the Ph.D. degree in computer science from the University of Windsor, Canada, in 2009. She is currently an Associate Professor with the Wuxi Institute of Technology, China. She has published over ten articles in major international journals and conferences such as the *International Journal of Web Services Research*, *Journal of Systems and Software*, *Software Testing, Verification and Reliability*, and so on. Her research interests include big data technology and applications.



YINGZHONG SHI received the Ph.D. degree from Jiangnan University, Wuxi, China, in 2016. He is currently a Professor with the School of IoT Technology, Wuxi Institute of Technology. His current research interests include pattern recognition, intelligent computation, and applications.

...