Research Article

A Weighted Block Dictionary Learning Algorithm for Classification

Zhongrong Shi

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China

Correspondence should be addressed to Zhongrong Shi; shizrong@163.com

Received 3 April 2016; Accepted 30 June 2016

Academic Editor: Yaguo Lei

Copyright © 2016 Zhongrong Shi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Discriminative dictionary learning, playing a critical role in sparse representation based classification, has led to state-of-the-art classification results. Among the existing discriminative dictionary learning methods, two different approaches, shared dictionary and class-specific dictionary, which associate each dictionary atom to all classes or a single class, have been studied. The shared dictionary is a compact method but with lack of discriminative information; the class-specific dictionary contains discriminative information but consists of redundant atoms among different class dictionaries. To combine the advantages of both methods, we propose a new weighted block dictionary learning method. This method introduces proto dictionary and class dictionary. The proto dictionary is a base dictionary without label information. The class dictionary is a class-specific dictionary, which is a weighted proto dictionary. The weight value indicates the contribution of each proto dictionary block when constructing a class dictionary. These weight values can be computed conveniently as they are designed to adapt sparse coefficients. Different class dictionaries have different weight vectors but share the same proto dictionary, which results in higher discriminative power and lower redundancy. Experimental results demonstrate that the proposed algorithm has better classification results compared with several dictionary learning algorithms.

1. Introduction

Recently, sparse representation based classifications have been extensively discussed with encouraging results. In these methods, choosing a proper dictionary is the first and most important step. In literatures, there are two ways to design a dictionary: prespecified versus adaptive. At the early stage, for the sake of simplicity, predetermined dictionaries (e.g., over-complete DCT or wavelets dictionaries) are often resorted to. Later on, dictionaries learned from training data [1] obtain more attentions, because the learned dictionaries usually lead to better representation and achieve much success in applications such as classification.

In last decades, many famous dictionary learning approaches have been proposed. These approaches can be divided into two categories: unsupervised dictionary learning (UDL) approaches [2] and supervised dictionary learning (SDL) approaches [3–7]. UDL learns dictionary using unlabeled training samples. SDL learns dictionary using labeled training samples. K-SVD algorithm [2] is a popular

UDL algorithm that learns a compact dictionary by singular value decomposition from a set of unlabeled samples. It has been widely applied to image processing tasks, such as image compression [8, 9], image restoration [10, 11], image deblurring [12, 13], super-resolution [14, 15], and visual tracking [16, 17]. K-SVD mainly focuses on the representational power of sparse representation but ignores the discriminative power of sparse representation which is critical for pattern classification. Representational power of sparse representation is the capability to sparsely reconstruct sample using sparse coefficient and dictionary. Discriminative power is the capability that these sparse coefficients can be well distinguished when they belong to different categories; this results in the application of these sparse coefficients to classify these samples.

Depending on whether training samples have been labeled, current dictionary learning approaches can be divided into two types: UDL approach and SDL approach. However, depending on whether atoms have been labeled, current dictionary learning approaches also can be divided into two main types: shared dictionary learning approaches

[1, 19–23] and class-specific dictionary learning approaches [3–7]. In shared dictionary learning approaches, all atoms do not have label information and are shared by samples from all classes. Shared dictionary learning approaches can be UDL approaches, but they also can be SDL approaches. For example, K-SVD algorithm [2] is a shared and unsupervised dictionary learning approach, but D-KSVD algorithm [20] is an extension of K-SVD algorithm, which is a shared and supervised dictionary learning approach. D-KSVD learns a discriminative dictionary by incorporating linear classification error into objective function. In class-specific dictionary learning approaches, all atoms are labeled, each of which can only be shared by the same class samples. Class-specific dictionary learning approaches must be SDL approach. For class-specific dictionary, class-specific reconstruction errors can be utilized to classify samples. Moreover, some discriminant criteria can be incorporated into dictionary learning processing. For example, Zhang et al. presented a low rank constraint [24], Yang et al. added a Fisher discrimination criterion [5], Ramirez et al. proposed a structure incoherent constraint [4], and so forth. However, when there are a large number of classes, the size of learned dictionary will be very large, and the redundancy of learned dictionary could become serious. Recently, some hybrid methods which combine shared dictionary and class-specific dictionary have been proposed [25–27]. In these methods, shared and class-specific parts need to be predefined, and the balance of the two parts is not a trivial task which is usually determined empirically.

Although above-mentioned dictionary learning methods have achieved good classification results, labels of these dictionary atoms are predefined and fixed, which may not be able to accurately interpret true structure of data. Yang et al. [28] proposed a latent dictionary learning (LDL) method. LDL learns a latent matrix to build the relationship between dictionary atoms and class labels; this mechanism has achieved very high classification accuracy.

In this paper, we propose a new dictionary learning method, named weighted block dictionary learning (WBDL) method. This method is a compromise between shared dictionary and class-specific dictionary. As shown in Figure 1(a), WBDL learns a proto dictionary which can be shared by all class dictionaries. Proto dictionary B contains m blocks. Assuming training samples have C classes, model should learn C subdictionaries, $D_1, D_2, \dots, D_C$, where D_i is a class-specific dictionary corresponding to class i . Each class dictionary is obtained by multiplying proto dictionary with the corresponding weight vector. For the weight vector, each value in it indicates the contribution of a block when constructing class dictionary. Class dictionary is a class-specific dictionary which represents samples coming from the same class. The sparse coefficients Z_i are obtained by sparsely represented samples into the class dictionary. As shown in Figure 1(b), a new test sample x is represented by each class dictionary, and we get C sparse coefficients, $\hat{z}_1, \hat{z}_2, \dots, \hat{z}_C$. Those sparse coefficients can be utilized to classify this test sample. For WBDL model, instead of predefined each block belonging to only a single class, each block of proto dictionary can belong to all classes. The shared dictionary [1, 19–23] could be

regarded as a special case of our WBDL model when weight matrix is an all-one matrix. The class-specific dictionary [3–7] also could be regarded as a special case of our WBDL model when weight vector has only one unique nonzero element 1. Compared to shared dictionary and class-specific dictionary, our proposed model is more flexible, and it increases discriminability and reduces redundancy simultaneously.

Our specific contributions are listed as below.

Firstly, for higher discriminative power and lower redundancy, we design a proto dictionary and some class dictionaries, where each class dictionary is a weighted proto dictionary. Our goal is to learn a compact proto dictionary and some discriminative class dictionaries. The class dictionary can represent samples sparsely and discriminatively.

Secondly, sparse coefficients obtained by sparse representation can be utilized to implement the classification. Two classification algorithms are proposed: local classification algorithm and global classification algorithm. When training samples for each class are enough, test samples are locally coded into all class dictionaries. On the contrary, test samples are globally coded into total dictionary. Global classification algorithm is a simplification of local classification algorithm.

Thirdly, weight vector corresponding to each class dictionary is easy to learn as it adapts to sparse coefficients of the samples coming from the same category. These weights can be computed from these sparse coefficients directly. Compared to traditional dictionary learning algorithms, WBDL algorithm would not significantly increase computational complexity. Experiments on some databases show that WBDL algorithm is competitive to some algorithms such as [2, 3, 5, 20, 23, 29].

This paper is organized as follows. In Section 2, we illustrate the related work, including shared dictionary and class-specific dictionary. In Section 3, weighted block dictionary learning model is proposed and analyzed. Two WBDL classification approaches also are proposed in this section. In Section 4, optimization of WBDL model is described and its two classification algorithms are given. In Section 5, experiments are performed on face recognition and object classification datasets to compare our algorithm with several state-of-the-art methods. We end this paper with a conclusion in Section 6.

2. Related Work

In this section, we review two types of dictionaries, shared dictionary and class-specific dictionary.

2.1. Shared Dictionary. K-SVD algorithm is a popular UDL algorithm, which learns a shared dictionary. K-SVD optimizes the following objective function:

$$\begin{aligned} \arg \min_{D, Z} \quad & \|X - DZ\|_F^2 \\ \text{s.t.} \quad & \forall i, \\ & \|z_i\|_0 \leq T, \end{aligned} \quad (1)$$

where $X = [x_1, x_2, \dots, x_N] \in \mathcal{R}^{n \times N}$ are N input signals; each is n dimension. $D = [d_1, d_2, \dots, d_K] \in \mathcal{R}^{n \times K}$ ($K \gg n$,

FIGURE 1: (a) Learning of proto dictionary and class dictionary. (b) Classification of a new test sample x .

making D overcomplete) is a dictionary with K atoms. $Z = [z_1, z_2, \dots, z_N] \in \mathbb{R}^{K \times N}$ are N sparse codes of input signals X . T is a constant which controls the number of nonzero elements in z_i less than T .

The minimization of (1) is solved by a two-step iterative algorithm. Firstly, dictionary D is fixed and sparse coefficients Z can be found. This is a sparse coding problem, which can be solved by OMP [30], and so forth. Secondly, sparse coefficient matrix Z is fixed and dictionary D is updated one atom at a time while fixing all other atoms in D .

For each atom d_k and the corresponding k th row of coefficient matrix Z denoted by z_T^k , define the group of samples that use this atom as $\omega_k = \{i \mid 1 \leq i \leq N, z_T^k(i) \neq 0\}$. The error matrix is computed: $E_k = X - \sum_{i \neq k} d_i \times z_T^i$; restrict E_k by choosing only the columns in ω_k and obtain E_k^R . Then the following problem is solved:

$$\arg \min_{d_k, z_T^k} \|E_k^R - d_k \times z_T^k\|_F^2. \quad (2)$$

A singular value decomposition (SVD) is performed, $E_k^R = U \Delta V^T$. $d_k = U(:, 1)$, $z_T^k = \Delta(1, 1) \times V(1, :)$, where $U(:, 1)$ denotes the first column of U and $V(1, :)$ for the first row of V .

D-KSVD algorithm is a supervised extension of K-SVD algorithm, which is SDL algorithm, but it also learns a shared dictionary. D-KSVD optimizes the following problem:

$$\begin{aligned} \arg \min_{D, G, Z} \quad & \|X - DZ\|_F^2 + \gamma \|H - GZ\|_F^2 \\ \text{s.t.} \quad & \forall i, \\ & \|z_i\|_0 \leq T. \end{aligned} \quad (3)$$

$H = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{C \times N}$ is the label matrix of input signals X . GZ is a linear classifier; $G \in \mathbb{R}^{C \times K}$ is its weight matrix. The term $\|X - DZ\|_F^2$ denotes reconstruction error. The new term $\|H - GZ\|_F^2$ is a classification error of linear classifier. γ controls balance between reconstruction and discriminant.

Compared to K-SVD algorithm, D-KSVD algorithm adds the second term classification error. D-KSVD dictionary learning method utilizes the class labels of training samples and its dictionary is more discriminative. However, the class labels of atoms have not been taken into account in D-KSVD algorithm.

2.2. Class-Specific Dictionary. Class-specific dictionary should be learned using a SDL algorithm. Suppose that there are C class samples; a class-specific dictionary is denoted as $D = [D_1, D_2, \dots, D_C]$; each D_i is a subdictionary corresponding to class i . All atoms of dictionary have been labeled, and subdictionary should be learned or constructed class by class.

Sparse representation based classification (SRC) [3] method is a popular method to construct class-specific dictionary. Suppose that there are C classes of samples; $X = [X_1, X_2, \dots, X_C]$ is the set of training samples and X_i is the subset belonging to the i th class. SRC can be summarized into two stages. Supposing x is a query sample, x is sparsely represented into constructed dictionary $X = [X_1, X_2, \dots, X_C]$ through $\hat{z} = \arg \min_z (\|x - Xz\|_2^2 + \gamma \|z\|_1)$. Then $\text{identity}(x) = \arg \min_i \|x - X_i \hat{z}_i\|_2$, where $\hat{z} = [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_C]^T$ can be used to classify x . Obviously, SRC utilizes representation residual to classify a test sample.

SRC is a constructed dictionary, and the subset X_i can be denoted as the i th class dictionary. Generally, class-specific dictionaries can be learned class by class using the following objective function:

$$\begin{aligned} \min_{D_i, Z_i} \quad & \|X_i - D_i Z_i\|_F^2 + \lambda \|Z_i\|_1 \\ \text{s.t.} \quad & \forall i, \\ & \|d_i\|_2 = 1. \end{aligned} \quad (4)$$

Equation (4) can be regarded as a basic model of class-specific dictionary learning method. This model does not consider the relationship between subdictionaries. Yang et al. proposed a Fisher discrimination dictionary learning method (FDDL) [5], which learns subdictionary for each class. FDDL model can be described as follows:

$$\begin{aligned} \min_{D, Z} \quad & \sum_{i=1}^C r(X_i, D, Z_i) + \lambda_1 \|Z\|_1 \\ & + \lambda_2 \text{tr}(S_W(Z) - S_B(Z)) + \eta \|Z\|_F^2 \\ \text{s.t.} \quad & \forall i, \\ & \|d_i\|_2 = 1, \end{aligned} \quad (5)$$

where the first term is data fidelity term, the second term is sparsity penalty, the third term is discrimination term, and the fourth term could make the function smooth and convex. FDDL model makes class-specific dictionary more distinctive.

In this paper, we integrate shared dictionary and class-specific dictionary into a new dictionary learning model and propose WBDL model.

The block structure of WBDL model is ensured by a mixed $L_{2,1}$ norm regularization. Most aforementioned methods simply adopt L_0 norm or L_1 norm for sparsity regularization. L_1 norm sparsity regularization has also been referred to as Lasso [31]. Inspired by the success of structured sparsity (Group Lasso) in the area of compressed sensing, some methods have been proposed for structured sparsity regularization. For example, Bengio et al. proposed group sparse coding (GSC) [21], which joins each category training sample into the same group and regularizes using mixed $L_{2,1}$ norm. This method encourages the same group samples encoded using the same dictionary atoms. Elhamifar and Vidal proposed block sparse coding (BSC) [32], which also uses mixed $L_{2,1}$ norm for regularization, but this regularization is used on a sparse coefficient vector rather than on a sparse coefficient matrix. The block sparsity regularization encourages block structure of the learned dictionary. Chi et al. proposed block and group regularized sparse coding (BGSC) [33], which combines group sparse coding and block sparse coding together. As shown by above methods, mixed $L_{2,1}$ norm is a suitable tool to learn a block structure of proto dictionary.

Generally, dictionary learning model has two unknown variables dictionary and sparse coefficient; WBDL introduces a new variable weight vector. We find that when a dictionary block and the j th class samples are more similar than the other dictionary block, this dictionary block is more suitable to represent the j th class samples. In consequence, the sparse coefficients corresponding to this block are relatively larger than others. Inspired by this observation, weight vector corresponding to the j th class dictionary can be obtained through sparse coefficients of the j th class samples. In order to avoid increasing computation complexity, weight vector is constructed through sparse coefficients directly in WBDL model.

3. Weighted Block Dictionary Learning

Shared dictionary learning algorithm ignores class labels of these dictionary atoms. Recently various class-specific dictionary learning approaches [3–7] have been proposed. These class-specific dictionary learning approaches are based on the assumption that the class label of each atom is invariable during the dictionary learning process. However, since dictionary atoms have been updated, the class label of these atoms should be reassigned in accordance with the updating of these dictionary atoms. The goal of our proposed weighted block dictionary learning model is to learn a labeled adaptive dictionary, which is composed of a proto dictionary and a weight matrix. Each column of the weight matrix is a weight vector that indicates the contribution of each proto dictionary block to construct a class dictionary. As a result, class dictionary is obtained by the product of a weight vector and the proto dictionary. In this section, firstly, we propose weighted block dictionary learning model. Secondly, we discuss the construction of weight matrix. Thirdly, we compare the difference of our WBDL model with BDL model. Finally, two classification approaches are proposed using WBDL model.

3.1. Weighted Block Dictionary Learning Model. Assume that $x \in \mathcal{R}^n$ is an n -dimensional signal with class label $y \in \{1, 2, \dots, C\}$. The training set with N samples is denoted as $X = [X_1, X_2, \dots, X_C] = [x_1, x_2, \dots, x_N] \in \mathcal{R}^{n \times N}$, where X_i is the subset associated with class i . We design a proto dictionary $B = [B_1, B_2, \dots, B_m] = [b_1, b_2, \dots, b_K] \in \mathcal{R}^{n \times K}$, where B_i is the i th block of proto dictionary, m is the number of blocks, and K is the total number of dictionary atoms. To better describe the relationship between a proto dictionary and C class dictionaries, a weight matrix $U = [u_1, u_2, \dots, u_C] \in \mathcal{R}^{m \times C}$ is introduced into our WBDL model, where $u_j = [u_{1,j}, u_{2,j}, \dots, u_{m,j}]^T \in \mathcal{R}^{m \times 1}$ is a vector to indicate the contribution of each proto dictionary block when constructing the j th class dictionary. For instance, $u_{m,j}$ is the weight value of the m th block proto dictionary to construct the j th class dictionary. Correspondingly, the i th class dictionary $D_i = [d_1^i, d_2^i, \dots, d_K^i] \in \mathcal{R}^{n \times K}$, $i \in \{1, 2, \dots, C\}$, is denoted as $D_i = B \text{diag}(\vec{u}_i)$. $\text{diag}(\vec{u}_i)$ is a diagonal matrix with vector $\vec{u}_i$ as its diagonal vector. In order to represent weight value of each atom, the size of diagonal matrix $\text{diag}(\vec{u}_i)$ is $K \times K$, so the weight vector u_j must be resized from m to $K_1 + K_2 + \dots + K_m = K$, where K_i is the number of atoms in i th block. For example, when the number of atoms in each block is 2, a weight vector should be resized from $[1, 0, 0, 0]^T$ to $[1, 1, 0, 0, 0, 0, 0]^T$. Finally, a sparse representation to encode data on the corresponding class dictionary is obtained. Take the j th class data X_j as an example; the j th class data can be represented as $X_j = D_j Z_j$.

The objective function of our proposed weighted block dictionary learning (WBDL) model can be described as follows:

$$\arg \min_{D, Z} \sum_{j=1}^C \left(\|X_j - D_j Z_j\|_F^2 + \tau \sum_{y_i=j}^m \sum_{r=1}^m \|z_{[r]}^i\|_2 \right), \quad (6)$$

$$D_j = B \text{diag}(\vec{u}_j) = [d_1^j, d_2^j, \dots, d_K^j],$$

$$\|d_i^j\|_2 \leq 1,$$

where B is a proto dictionary, $D_j = B \text{diag}(\vec{u}_j)$ is the j th class dictionary, and $Z = [Z_1, Z_2, \dots, Z_C] = [z_1, z_2, \dots, z_N] \in \mathcal{R}^{K \times N}$ are the sparse codes of X . $z_{[r]}^i$ is the r th block sparse coefficient of the i th sample. The first term denotes the reconstruction error of all j th category samples. The second term is the block sparse regularization of all j th category samples. τ is a scalar controlling the trade-off between reconstruction and sparsity. In order to avoid a trivial solution of sparse coefficient z_i , each dictionary atom be constrained, $\|d_i^j\|_2 \leq 1$.

As shown in (6), WBDL model is a nonconvex optimization problem, in which three unknown variables B , U , and Z need to be optimized. We propose a two-step iterative algorithm to solve this problem. The first step is the following: weight matrix U is fixed and coefficient matrix Z and dictionary B are learned, which is a general dictionary learning problem. The second step is the following: coefficient matrix Z and dictionary B are fixed and weight matrix U is

constructed. Construction of the weight matrix U is crucial for this new dictionary learning model.

3.2. Construction of Weight Matrix U . Without loss of generality, a weight value is required to be nonnegative, and the sum of which is equal to 1; that is, $u_{i,j} \geq 0$, $\sum_{i=1}^m u_{i,j} = 1$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, C$. When proto dictionary and sparse coefficients are fixed, how to calculate weight value for every block is crucial for us. If we do not take into account weight matrix, we can rewrite the block representation of the j th class samples as follows:

$$X_j = [B_1, B_2, \dots, B_m] [Z_{1,j}, Z_{2,j}, \dots, Z_{m,j}]^T, \quad (7)$$

where X_j is the j th class samples, B_i is the i th block of proto dictionary, and $Z_{i,j}$ is the i th block sparse coefficient corresponding to proto dictionary block B_i . Observing the sparse coefficients obtained by block sparse representation, a phenomenon is found. When B_i is more similar to the j th class samples, $u_{i,j}$ should be larger than $u_{k,j}$ ($k \neq i$, $k = 1, 2, \dots, m$); we find that the value of $Z_{i,j}$ is larger than the other coefficient block $Z_{k,j}$ ($k \neq i$, $k = 1, 2, \dots, m$). Inspired by this observation, weight value $u_{i,j}$ can be calculated using value of sparse coefficient $Z_{i,j}$. Consistent with Frobenius norm of reconstruction error, Frobenius norm of $Z_{i,j}$ is utilized to compute weight. Thus, our objective function can be rewritten as follows:

$$\arg \min_{D, Z} \sum_{j=1}^C \left(\|X_j - D_j Z_j\|_F^2 + \tau \sum_{y_i=j}^m \sum_{r=1}^m \|z_{[r]}^i\|_2 \right),$$

$$D_j = B \text{diag}(\vec{u}_j) = [d_1^j, d_2^j, \dots, d_K^j],$$

$$\|d_i^j\|_2 \leq 1, \quad (8)$$

$$u_j = \left[\frac{\|Z_{1,j}\|_F^2}{\|Z_j\|_F^2}, \frac{\|Z_{2,j}\|_F^2}{\|Z_j\|_F^2}, \dots, \frac{\|Z_{m,j}\|_F^2}{\|Z_j\|_F^2} \right]^T.$$

Obviously, weight value can be computed from other variables directly; three variables that needed to be optimized have been reduced to two variables. The weight value in (8) is nonnegative and satisfies the former constraint, $\sum_{i=1}^m u_{i,j} = 1$, $j = 1, 2, \dots, C$.

3.3. A Discussion about BDL Model and WBDL Model. Compared to block dictionary learning (BDL) model [32], our weighted block dictionary learning (WBDL) model introduces a weight vector into dictionary learning. The objective function of original block dictionary learning model can be described as

$$\arg \min_{B, Z} \sum_{i=1}^N \left(\|x_i - B z_i\|_2^2 + \tau \sum_{r=1}^m \|z_{[r]}^i\|_2 \right),$$

$$B = [B_1, B_2, \dots, B_m],$$

$$\|b_i\|_2 \leq 1, \quad (9)$$

where B_i is the i th block dictionary. The objective function of (9) can be rewritten:

$$\begin{aligned} \arg \min_{B,Z} \quad & \sum_{j=1}^C \left(\|X_j - B \operatorname{diag}(\vec{u}_j) Z_j\|_F^2 + \tau \sum_{y_i=j}^m \sum_{r=1}^m \|u_{r,j} z_{[r]}^i\|_2 \right), \\ B = & [B_1, B_2, \dots, B_m], \\ \|b_i\|_2 \leq & 1, \\ u_j = & \left[\frac{\|Z_{1,j}\|_F^2}{\|Z_j\|_F^2}, \frac{\|Z_{2,j}\|_F^2}{\|Z_j\|_F^2}, \dots, \frac{\|Z_{m,j}\|_F^2}{\|Z_j\|_F^2} \right]^T. \end{aligned} \quad (10)$$

Compared to the objective function of BDL [32] in (10), WBDL model in (8) deletes weight $u_{r,j}$ in block sparse regularization. When weight value $u_{r,j}$ of the r th block is larger than weights of other blocks, the r th block represents the j th class samples better than other blocks, so sparse coefficients corresponding to this block should be even larger than others. In BDL model [32], a big $u_{r,j}$ value will suppress $z_{[r]}^i$ and force the solution $z_{[r]}^i$ to be small. In our proposed WBDL model, we delete weight $u_{r,j}$, and this deletion will bring a relative increase of block sparse coefficients $z_{[r]}^i$. These increased sparse coefficients improve discriminative power compared with the original BDL model.

For example, a proto dictionary has 4 dictionary blocks, and each block has 2 atoms. A weight vector corresponding to j th class dictionary is $\vec{u}_j = [0.8, 0.8, 0.2, 0.2, 0, 0, 0, 0]^T$. If a sparse coefficient of j th class sample is $z = [1, 1, 1, 1, 0, 0, 0, 0]^T$, this sparse coefficient is taken as initialization of our model; then how will our new model modify this initial coefficient? Firstly, according to equation $\operatorname{diag}(\vec{u}_j)\tilde{z} = z$, new sparse coefficient of our WBDL model can be obtained, $\tilde{z} = [5/4, 5/4, 5/4, 5/4, 0, 0, 0, 0]^T$. Now we observe the new sparse coefficient $\tilde{z} = [5/4, 5/4, 5/4, 5/4, 0, 0, 0, 0]^T$; obviously, this sparse coefficient cannot clearly denote representational power of each dictionary block to represent the j th class samples. The 2nd block sparse coefficient is

very large in the block sparse regularization term. This will be modified at the next iteration, and the 2nd block sparse coefficient will decrease. For example, the final sparse coefficient can be denoted as $\tilde{z} = [5/4, 5/4, t, t, 0, 0, 0, 0]^T$ ($t < 5$). In the end, weighted sparse coefficient $\operatorname{diag}(\vec{u}_j)\tilde{z} = [1, 1, 0.2t, 0.2t, 0, 0, 0, 0]^T$ ($t < 5$) is used to classify this sample. Obviously, this weighted sparse coefficient is more coherent with weight vector $\vec{u}_j = [0.8, 0.8, 0.2, 0.2, 0, 0, 0, 0]^T$ compared with initial sparse coefficient $z = [1, 1, 1, 1, 0, 0, 0, 0]^T$. Weighted sparse coefficient obtained by our model will be more coherent with weight vector, and this will improve classification accuracy in the following classification task. An experiment has been taken to test the two models on a subset of AR database. We selected 60 face images for testing. The sparse coefficients of testing images in BDL model are shown in Figure 2(a) and the weighted sparse coefficients of the same testing images in WBDL model are shown in Figure 2(b). As shown in Figures 2(a) and 2(b), compared to BDL model, weighted sparse coefficients in WBDL model are more compact and discriminant.

3.4. WBDL Classification Model. The weighted sparse coefficient obtained by WBDL model will be used to classify sample. A linear classifier G can be learned while we learn a dictionary as D-KSVD algorithm does. Combining classification error of a linear classifier, WBDL classification model can be described as follows:

$$\begin{aligned} \arg \min_{D,G,Z} \quad & \sum_{j=1}^C \left(\|X_j - D_j Z_j\|_F^2 + \gamma \|H_j - G_j Z_j\|_F^2 + \tau \sum_{y_i=j}^m \sum_{r=1}^m \|z_{[r]}^i\|_2 \right), \\ D_j = & B \operatorname{diag}(\vec{u}_j) = [d_1^j, d_2^j, \dots, d_K^j], \\ \|d_i^j\|_2 \leq & 1, \\ G_j = & G \operatorname{diag}(\vec{u}_j) \in \mathfrak{R}^{C \times K}, \\ u_j = & \left[\frac{\|Z_{1,j}\|_F^2}{\|Z_j\|_F^2}, \frac{\|Z_{2,j}\|_F^2}{\|Z_j\|_F^2}, \dots, \frac{\|Z_{m,j}\|_F^2}{\|Z_j\|_F^2} \right]^T, \end{aligned} \quad (11)$$

FIGURE 2: Performance comparisons of sparse coefficients from BDL model and from WBDL model. We selected 200 images from 10 random persons in AR database [18] for training and the remaining 60 images for testing. Proto dictionary contains 10 blocks; each block contains 5 atoms. The sparse coefficients of 60 testing samples in BDL model are shown in (a) and the weighted sparse coefficients of the same 60 testing samples in WBDL model are shown in (b).

where $H = [H_1, H_2, \dots, H_C] = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{C \times N}$ is label matrix of X , in which H_j is the label matrix of all the j th class samples. Each column of H is a vector to indicate class label of the corresponding sample. For example, $h_i = [0, 0, \dots, 1, \dots, 0]^T \in \mathbb{R}^{C \times 1}$, where the position of nonzero element indicates the class label. $G \in \mathbb{R}^{C \times K}$ is the coefficient matrix of C linear classifier; the i th row of G is coefficient vector of the i th linear classifier. The first term $\|X_j - D_j Z_j\|_F^2$ denotes the reconstruction error of the j th class samples, the second term $\|X_j - G_j Z_j\|_F^2$ is classification error of the linear classifier, and the third term $\sum_{y_i=j} \sum_{r=1}^m \|z_{[r]}^i\|_2$ is block sparse regularization. γ and τ are scalars controlling trade-off between reconstruction, discriminant, and sparsity.

Concatenating the first term and the second term, let $\bar{X}_j = (X_j^T, \sqrt{\gamma} H_j^T)^T$ and $\bar{D}_j = (D_j^T, \sqrt{\gamma} G_j^T)^T$; (11) can be rewritten as follows:

$$\begin{aligned} \arg \min_{\bar{D}, Z} & \sum_{j=1}^C \left(\|\bar{X}_j - \bar{D}_j Z_j\|_F^2 + \tau \sum_{y_i=j} \sum_{r=1}^m \|z_{[r]}^i\|_2 \right), \\ \bar{X}_j &= (X_j^T, \sqrt{\gamma} H_j^T)^T, \\ \bar{D}_j &= (D_j^T, \sqrt{\gamma} G_j^T)^T, \\ D_j &= B \text{diag}(\vec{u}_j) = [d_1^j, d_2^j, \dots, d_K^j], \\ \|d_i^j\|_2 &\leq 1, \\ G_j &= G \text{diag}(\vec{u}_j) \in \mathbb{R}^{C \times K}, \\ u_j &= \left[\frac{\|Z_{1,j}\|_F^2}{\|Z_j\|_F^2}, \frac{\|Z_{2,j}\|_F^2}{\|Z_j\|_F^2}, \dots, \frac{\|Z_{m,j}\|_F^2}{\|Z_j\|_F^2} \right]^T. \end{aligned} \quad (12)$$

Upon the training of the labeled data, we learn a weight matrix U and an extended dictionary $(B^T, G^T)^T$, which is the concatenation of a proto dictionary B and a linear classifier G . However, since B and G are normalized jointly in the previous learning process, B does not support sparse code of a new test sample. As proposed in [20], proto dictionary B and corresponding classifier G are normalized as follows:

$$\begin{aligned} B &= \left\{ \frac{b_1}{\|b_1\|_2}, \frac{b_2}{\|b_2\|_2}, \dots, \frac{b_K}{\|b_K\|_2} \right\}, \\ G &= \left\{ \frac{g_1}{\|b_1\|_2}, \frac{g_2}{\|b_2\|_2}, \dots, \frac{g_K}{\|b_K\|_2} \right\}. \end{aligned} \quad (13)$$

For normalized proto dictionary B and weighted matrix U , we can obtain a sparse code for a new test sample x by two coding strategies: local coding and global coding.

When training samples for each class are enough, a test sample x is locally coded into all class dictionaries, respectively. Taking the j th class local code as an example, we have the following:

Local sparse code:

$$\hat{z}_j = \arg \min_z \|x - D_j z\|_2^2 + \tau \sum_{r=1}^m \|z_{[r]}\|_2. \quad (14)$$

The final classification of this test sample x is based on the following classifier:

Local classifier:

$$j = \arg \min_j \|x - D_j \hat{z}_j\|_2^2, \quad j = 1, 2, \dots, C. \quad (15)$$

The label of test sample x is determined by the class label of sparse code which has the smallest reconstruction error.

When training samples for each class are not enough, a test sample x can be globally coded. We define a total weight vector, denoted by $w = \sum_{j=1}^C u_j$, which reflects the total relationship between each block of proto dictionary and all involved classes. A big value of w_i shows that proto dictionary block B_i is important to represent all classes. Global sparse code can be computed as follows:

Global sparse code:

$$\hat{z} = \arg \min_z \|x - B \text{diag}(\tilde{w}) z\|_2^2 + \tau \sum_{r=1}^m \|z_{[r]}\|_2. \quad (16)$$

Utilizing the former learned linear classifier G , the final classification of test sample x can be obtained by the following classifier:

Global classifier:

$$j = \arg \max_j (l = G \text{diag}(\tilde{w}) \hat{z})_j, \quad j = 1, 2, \dots, C, \quad (17)$$

where $l \in \mathfrak{R}^{C \times 1}$ is a vector. The label of test sample x is determined by the index of the largest element in l .

4. Optimization of WBDL Model

In the objective function of our proposed WBDL model, there are two unknown variables B and Z , a variable U which can be computed from Z directly. We adopt alternated optimization to solve such multivariable problem and design a two-step iterative algorithm. Firstly, weight matrix U is fixed but coefficient matrix Z and proto dictionary B are learned, which is actually a general dictionary learning problem. Secondly, coefficient matrix Z and proto dictionary B are fixed and weight matrix U is updated, which is a process to learn weight matrix. In this section, we describe the optimization of each step separately and give the whole algorithm at last.

4.1. Dictionary Learning. When weight matrix U is fixed, coefficient matrix Z and proto dictionary B are learned. Firstly, we fix proto dictionary B and learn coefficient matrix Z ; this is block sparse coding. Secondly, we fix coefficient matrix Z and update proto dictionary B , which is dictionary updating.

4.1.1. Block Sparse Coding. For example, we compute the block sparse coefficient of a j th class sample x . Firstly, we obtain the j th class dictionary, $D_j = B \text{diag}(\vec{u}_j)$, and then we formulate the minimization of (8) only for the r th block of sparse coefficient z ; this optimization is similar to that of BGSC method [33]. By fixing D_j , (8) can be written as

$$z_{[r]} = \arg \min_{z_{[r]}} \left\| x - \sum_{i \neq r} (D_j)_{[i]} z_{[i]} - (D_j)_{[r]} z_{[r]} \right\|_2^2 + \tau \|z_{[r]}\|_2 + c, \quad (18)$$

where $(D_j)_{[r]}$ is the r th block of class dictionary D_j and c is the term that does not depend on $z_{[r]}$. Computing the gradient of (18) with respect to $z_{[r]}$, we can obtain the following condition:

$$- (D_j)_{[r]}^T x + (D_j)_{[r]}^T \sum_{i \neq r} (D_j)_{[i]} z_{[i]} + (D_j)_{[r]}^T (D_j)_{[r]} z_{[r]} + \frac{\tau}{2} \frac{z_{[r]}}{\|z_{[r]}\|_2} = 0. \quad (19)$$

Assuming $\|z_{[r]}\|_2 > 0$, denoting the first two terms by $-N$, substituting the semidefinite matrix $(D_j)_{[r]}^T (D_j)_{[r]}$ with its Eigen-decomposition $V \Sigma V^T$, and multiplying with V^T on both sides, (19) can be formulated into

$$\Sigma V^T z_{[r]} + \frac{\tau}{2} \frac{V^T z_{[r]}}{\|z_{[r]}\|_2} = V^T N. \quad (20)$$

Denoting new variable $e = V^T z_{[r]}$, and $\hat{N} = V^T N$, we have

$$\Sigma e + \frac{\tau}{2} \frac{e}{\|e\|_2} = \hat{N}. \quad (21)$$

Setting $\kappa = \|e\|_2$, and $\hat{e} = e/\|e\|_2$, we have

$$\hat{e} = \left(\kappa \Sigma + \frac{\tau}{2} I \right)^{-1} \hat{N} \quad (22)$$

$$\text{s.t. } \|\hat{e}\|_2 = 1.$$

We can compute κ using Newton's method. Once κ is known, we can compute $\hat{e}$ and e . Finally, $z_{[r]}$ can be obtained, $z_{[r]} = V e$.

When the solution of κ is not positive, the above assumption $\|z_{[r]}\|_2 > 0$ does not hold. In this case, the optimality solution is $z_{[r]} = 0$. The proof can be found in [32].

4.1.2. Dictionary Updating. Let $\bar{Z}_j = \text{diag}(\vec{u}_j) Z_j$ denote weighted sparse coefficients; we fix $\bar{Z} = [\bar{Z}_1, \bar{Z}_2, \dots, \bar{Z}_C]$ and update proto dictionary B ; the objective function subject to B is as follows:

$$B = \arg \min_B \|X - B \bar{Z}\|_F^2, \quad (23)$$

$$\|b_i\|_2 \leq 1.$$

We can minimize the objective function by Lagrange dual method [34].

4.2. Learning of Weight Matrix U . We find that sparse coefficients inherit weight information of class dictionary. Motivated by this observation and the details described in Section 3.2, weight matrix $U = [u_1, u_2, \dots, u_C] \in \mathfrak{R}^{m \times C}$ can be constructed as follows:

$$u_j = \left[\frac{\|Z_{1,j}\|_F^2}{\|Z_j\|_F^2}, \frac{\|Z_{2,j}\|_F^2}{\|Z_j\|_F^2}, \dots, \frac{\|Z_{m,j}\|_F^2}{\|Z_j\|_F^2} \right]^T, \quad (24)$$

$$j = 1, 2, \dots, C,$$

where $Z_{i,j}$ is the i th block of matrix Z_j ; Z_j is the sparse code matrix of the j th class samples. The adopted norm is Frobenius norm. $u_{i,j}$ satisfies the following conditions:

$$\sum_{i=1}^m u_{i,j} = 1, \quad u_{i,j} \geq 0. \quad (25)$$

In the process of block sparse coding, sparse coefficient is computed one by one. When learning weight vector, each weight vector is computed from all the sparse coefficients of the same class. A weight vector is reflected by all the sparse coefficients of the same class. As a result, the computation of weight vector has not been integrated with block sparse coding. Weight vector is computed after all sparse coefficients have been obtained.

In our proposed WBDL model, for each proto dictionary block, rather than assigning it to only one class, we assign C weight values to indicate its relationship to all class dictionaries. The weight matrix preserves more class-label information. The construction of a weight matrix in the model adapts to block sparse coefficients and has not improved computation complexity significantly.

WBDL algorithm and its two classification algorithms, local classification algorithm and global classification algorithm, are described as follows.

Algorithm 1 (weighted block dictionary learning algorithm).

Input. A training sample set $X = [X_1, X_2, \dots, X_C] \in \mathbb{R}^{n \times N}$ and its class label $y_i \in \{1, 2, \dots, C\}$, $i = 1, 2, \dots, N$.

Output. Proto dictionary B and weight matrix U .

Step 1. Initialize U to all-one matrix.

Step 2. Dictionary learning

Repeat

Block sparse coding: compute sparse coefficient Z by minimizing (18) while fixing the corresponding class dictionary.

Dictionary updating: update proto dictionary B by minimizing (23) while fixing the weighted sparse coefficients.

Until convergence.

Step 3. Construct weight matrix U by the definition in (24).

Step 4. Return to Step 2 until the values of the objective function in (8) are close enough or the maximum number of iterations is reached.

Output B and U .

Algorithm 2 (WBDL local classification algorithm).

Input. A training sample set $X = [X_1, X_2, \dots, X_C] \in \mathbb{R}^{n \times N}$ and its class label $y_i \in \{1, 2, \dots, C\}$, $i = 1, 2, \dots, N$, a test sample x .

Output. Proto dictionary B , weight matrix U , and the classification result of test sample x .

Step 1. Learn proto dictionary and weight matrix using WBDL algorithm (Algorithm 1).

Step 2. For a test sample x , compute C local code $\hat{z}_j$, $j = 1, 2, \dots, C$, using (14).

Step 3. Compute the label of x using (15).

Algorithm 3 (WBDL global classification algorithm).

Input. A training sample set $X = [X_1, X_2, \dots, X_C] \in \mathbb{R}^{n \times N}$ and its class label $y_i \in \{1, 2, \dots, C\}$, $i = 1, 2, \dots, N$, a test sample x .

Output. Proto dictionary B , weight matrix U , classifier G , and the classification result of test sample x .

Step 1. Generate label matrix H .

Step 2. Generate new data $\bar{X} = (X^t, \sqrt{\gamma}H^t)^t$.

Step 3. Learn extended dictionary $(B^T, G^T)^T$ and U using WBDL algorithm (Algorithm 1).

Step 4. Separate extended dictionary into B and G ; normalize B and G by (13).

Step 5. Compute total weight vector as $w = \sum_{j=1}^C u_j$.

Step 6. For a test sample x , compute global code $\hat{z}$ by (16).

Step 7. Compute the label of x using (17).

5. Experimental Results

In this section, WBDL algorithm was evaluated on three classification tasks of simulation experiment, face recognition, and object recognition. For simulation experiment, we compared Fisher value when block structure and weight matrix were separately introduced. For face recognition, we experimented on two face databases: AR [18] and Extended Yale B [35]. For object recognition, Caltech101 database [36] was adopted for evaluation. For all experiments, the randomly selected samples from the same class were taken as the initialization of proto dictionary, and an all-one matrix was taken as the initialization of weight matrix. For global classification algorithm, the scalar γ controlling discriminant term was set to 1.

5.1. Simulation Experiment. Compared to general dictionary learning algorithm, WBDL model introduces block structure and weight matrix. In this section, we measure the discrimination of this model by Fisher criterion, which is the ratio of between-class variance and in-class variance. Fisher criterion can be defined as follows:

$$S = \frac{\|\mu_1 - \mu_2\|_2^2}{(1/C_1) \times \sum_{i=1}^{C_1} \|z_i^1 - \mu_1\|_2^2 + (1/C_2) \times \sum_{i=1}^{C_2} \|z_i^2 - \mu_2\|_2^2}, \quad (26)$$

TABLE 1: The values for Fisher criterion in simulation experiment.

Method	Fisher criterion
D-KSVD	0.35
WDL	0.46
BDL	1.20
WBDL (global)	1.27
WBDL (local)	1.36

where u_j ($j = 1, 2$) is the mean of the j th class sparse codes, C_j ($j = 1, 2$) is the total number of the j th class samples, and z_i^j ($i = 1, \dots, C_j$; $j = 1, 2$) is the sparse code of the i th sample belonging to the j th class. A bigger Fisher value means a better classification result. We used 52 images from 2 random persons in AR database for this simulation. For each person, we randomly selected 20 images for training and the remaining 6 images for testing. We used the same parameters for all the following four methods, D-KSVD [20], WDL, BDL, and WBDL. WDL is the algorithm which only introduces weight vector, and BDL is the algorithm only introducing block structure. WBDL is the algorithm adding weight vector and block structure, simultaneously. The obtained Fisher values are listed in Table 1. The results show that all weight vector and block structure have improved discriminant performance of dictionary learning; WBDL algorithm is more discriminative for dictionary learning. Particularly, WBDL local classification algorithm is more competitive than WBDL global classification algorithm. Local classification algorithm can improve Fisher value 0.09 better than global classification algorithm. Just because local classification algorithm fully takes advantage of weight vector but global classification algorithm does not, global classification algorithm can be taken as simplicity of local classification algorithm.

5.2. Face Recognition on AR. Face recognition is a popular application of computer vision and pattern recognition in recent years. In this section, WBDL algorithm is evaluated through face recognition task on AR face database [18]. As shown in Figure 3, we show 10 face images of different two subjects. These images in AR database include much more facial variations, including expression, illumination, and facial disguises (sunglasses and scarves). AR database consists of over 4,000 color images of 126 persons, and each person has 26 face images. A subset consisting of 2600 images from 50 male subjects and 50 female subjects was used in this experiment. For each person, we randomly selected 20 images for training and the remaining 6 images for testing. The average of the results on six such random splits of training and testing images is taken as the final results.

In all experiments, AR face image $\in \mathbb{R}^{192 \times 168}$ was projected into a vector $\in \mathbb{R}^{540}$ with Randomface [3]. The learned proto dictionary had 500 atoms, 5 items per block. The regularization parameter τ was set to 0.03. WBDL algorithm is compared with several recently proposed algorithms including SRC [3], KSVD [2], D-KSVD [20], and LC-KSVD [23]. Recognition results are summarized in Table 2. As shown in Table 2, WBDL global classification algorithm

TABLE 2: The recognition accuracy (%) of SRC, K-SVD, D-KSVD, LC-DKSVD, and WBDL on AR database.

Method	Accuracy (%)
SRC	77.34
K-SVD	89.41
D-KSVD	90.60
LC-KSVD	91.78
WBDL (global)	94.00
WBDL (local)	95.33

and WBDL local classification algorithm all outperform the competing methods. In these experiments, dictionary learning processes are all the same; local classification algorithm can improve accuracy 1.33% better than global classification algorithm. Generally speaking, when there are adequate training samples, WBDL local classification algorithm is more competitive than WBDL global classification algorithm with respect to making full use of weight vector.

In addition, when block structure and weight vector were introduced, we recorded the decrease of objective function values in (8) with varied number of iterations. Figure 4 displays the values of objective function with different number of iterations. As shown in Figure 4, after 10 runs, the objective function values decrease very lowly, so WBDL algorithm converges much faster.

5.3. Face Recognition on Extended Yale B. In this section, we evaluate WBDL algorithm with existing dictionary learning methods on Extended Yale B face database [35]. Extended Yale B database consists of 2,432 cropped frontal face images of 38 individuals. For each person, there are 64 face images that are captured under various lighting conditions. As shown in Figure 5, we show 12 face images of different two subjects. The key challenge of this database is due to varying illumination and expression. Since the dimension 192×168 of original face images is large, we reduce dimension of images to $n = 132$ using Randomface [3]. To compare proposed algorithm with other methods, we randomly chose the half for training and the rest for testing for each subject in all experiments. For simplicity of analysis, we learned 38 dictionary blocks. Assuming that all proto dictionary blocks have the same number of atoms, we learned $k \in \{9, 18, 25, 32\}$ atoms for each block. Regularization parameter τ was set to 0.06, 0.07, 0.07, and 0.08 for each block size. Test samples were globally coded and locally coded, separately. Experiments were repeated 6 times for random split of training data and testing data; the average classification rates among all the trials were taken as the final results.

The proposed WBDL algorithm is compared with several recently proposed algorithms including SRC [3], KSVD [2], D-KSVD [20], LC-KSVD [23], Pl2/1 [32], and SVGDL [29]. Recognition results are presented in Figure 6. As the results showed, WBDL algorithm always outperforms other methods, especially when dictionary size is small. When dictionary size is bigger, for example, block size being 32, classification accuracies of these learned dictionaries (K-SVD, D-KSVD,

FIGURE 3: Example images from AR database.

FIGURE 4: Convergence behavior: the values of objective function generated by WBDL model on AR database with varying number of iterations.

LC-KSVD, and SVGDL) do not exceed accuracies of those constructed dictionaries (SRC, Pl2/l1), but the classification accuracies of our two WBDL classification algorithms are far in excess of 3% and 3.6% compared with SRC.

5.4. Object Classification on Caltech101. Caltech101 database [36] contains 101 object classes and a “background” class with high shape variability. The number of images per category varies from 31 to 800. Most images are medium resolution of about 300×300 pixels. As shown in Figure 7, we show 15 images of Caltech101 database; those images come from 15 different categories.

We firstly extracted SIFT [37] descriptors from 16×16 patches which were densely sampled using a grid with step

size of 6 pixels. Secondly, we extracted the spatial pyramid features based on SIFT features with three grids of sizes 1×1 , 2×2 , and 4×4 in each spatial subregion of the spatial pyramid. Thirdly, the features were pooled together to form pooled features 128×21 . Max pooling and l_1 normalization were used for pooling and normalization, respectively, which were evaluated in [38] being superior to other pooling and normalization methods. Fourthly, we trained the codebook for spatial pyramid features using standard k -means clustering with $k = 1024$; then the spatial pyramid features were reduced to 3000 dimensions from 1024×21 dimensions by PCA. Finally, we trained class dictionary and learned classifier on the final spatial pyramid features using WBDL algorithm. Following the common experimental settings, we trained on 5, 10, 15, 20, 25, and 30 samples per category and tested on the rest, and the test samples were globally coded and locally coded, separately. We repeated experiments 6 times with different random splits of training and testing images; the average results of each run were reported as final recognition rates. The sparsity controlling τ used in all the experiments is 0.06. The results compared with the popular ScSPM [38], SRC [3], K-SVD [2], D-KSVD [20], LC-KSVD [23], FDDL [5], and SVGDL [29] algorithms are listed in Table 3. As shown in Table 3, WBDL algorithm maintains the highest classification accuracies when we trained on 5, 10, 20, 25, and 30 samples per category. SVGDL algorithm obtains the highest classification accuracy when 15 samples per category were selected to train the dictionary. In general, WBDL algorithm maintains the higher classification accuracies.

We also compare classification accuracy with SRC [3], K-SVD [2], D-KSVD [20], LC-KSVD [23], and SVGDL [29] using different dictionary sizes $K = 510, 1020, 1530, 2040, 2550$, and 3060 when we randomly select 30 images per category as training data. As shown in Figure 8, WBDL algorithm maintains the highest classification accuracy in all dictionary

FIGURE 5: Example images from Extended Yale B database.

FIGURE 6: Performance comparisons of recognition accuracies on Extended Yale B database with varying block size.

size compared with other six methods. Experiment results in Table 3 demonstrate the increased classification accuracy while adding the numbers of training samples. The experiment results in Figure 8 describe the increased classification accuracy while adding the size of dictionary. Our results in Figure 8 maintain the highest accuracy in all size, which are better than the results of SVGDL algorithm. Since WBDL algorithm is more sensitive to size of dictionary block SVGDL algorithm is more sensitive to numbers of training samples. Compared to other six dictionary learning algorithms, when dictionary size is small, for example, 510, WBDL algorithm and SVGDL algorithm all have improvement in classification accuracy.

TABLE 3: The recognition accuracies (%) of ScSPM, SRC, K-SVD, D-KSVD, LC-KSVD1, LC-KSVD2, FDDL, SVGDL, and WBDL on Caltech101 database.

Training number	5	10	15	20	25	30
ScSPM	—	—	67.0	—	—	73.2
SRC	48.8	60.1	64.9	67.7	69.2	70.7
K-SVD	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD	49.6	59.5	65.1	68.6	71.1	73.0
LC-KSVD1	53.5	61.9	66.8	70.3	72.1	73.4
LC-KSVD2	54.0	63.1	67.7	70.5	72.3	73.6
FDDL	53.6	63.6	66.8	69.8	71.7	73.1
SVGDL	55.3	64.3	69.6	72.3	75.1	76.7
WBDL (global)	55.8	65.1	68.8	72.8	75.2	77.0
WBDL (local)	56.0	65.3	69.2	72.8	75.3	77.0

6. Conclusions

In this paper, a weighted block dictionary learning algorithm is proposed, which is a compromise of shared dictionary and class-specific dictionary. This WBDL dictionary is the product of proto dictionary and corresponding weight vector. Proto dictionary is a shared dictionary. Weighted proto dictionary is a class-specific dictionary. WBDL dictionary learning method reduces the redundancy and enhances the discriminative ability, and it is beneficial to explore the intrinsic structures of dictionary. The experimental results on three databases demonstrate that WBDL algorithm maintains the higher classification accuracies. Compared to the other dictionary learning algorithms, the proposed WBDL algorithm is more discriminative with small dictionary size. Just because WBDL takes advantage of

FIGURE 7: Example images from Caltech101 database.

FIGURE 8: Performance comparison of classification accuracies on Caltech101 database with varying dictionary size.

weight vector, the dictionary learned by WBDL algorithm is more discriminative and compact.

Competing Interests

The author of this paper declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Fund of China (Grant nos. 60632050, 9082004, and 61202318), the National 863 Project (Grant no. 2006AA04Z238), and the Basic Key Technology Project of the Ministry of Industry and Information Technology of China (Grant no. E0310/1112/JC01).

References

- [1] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.
- [2] 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, no. 11, pp. 4311–4322, 2006.
- [3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

- [4] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3501–3508, June 2010.
- [5] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.
- [6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [7] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," *International Journal of Computer Vision*, vol. 100, no. 1, pp. 1–15, 2012.
- [8] G. Shao, Y. Wu, A. Yong, X. Liu, and T. Guo, "Fingerprint compression based on sparse representation," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 489–501, 2014.
- [9] O. G. Sezer, O. G. Guleryuz, and Y. Altunbasak, "Approximation and compression with sparse orthonormal transforms," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2328–2343, 2015.
- [10] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014.
- [11] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [12] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.
- [13] M. Carlván and L. Blanc-Féraud, "Sparse Poisson noisy image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1834–1846, 2012.
- [14] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.
- [15] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [16] B. Liu, J. Huang, C. Kulikowski, and L. Yang, "Robust visual tracking using local sparse appearance model and k-selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2968–2981, 2013.
- [17] W. Hu, W. Li, X. Zhang, and S. Maybank, "Single and multiple object tracking using a multi-feature joint sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 816–833, 2015.
- [18] A. M. Martinez and R. Benavente, "The AR face database," CVC Technical Report 24, The Ohio State University, 1998.
- [19] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3517–3524, IEEE, San Francisco, Calif, USA, June 2010.
- [20] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2691–2698, San Francisco, Calif, USA, June 2010.
- [21] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 82–89, December 2009.
- [22] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems*, pp. 1033–1040, 2009.
- [23] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.
- [24] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 676–683, Portland, Ore, USA, June 2013.
- [25] S. Kong and D. Wang, "A dictionary learning approach for classification: separating the particularity and the commonality," in *Computer Vision—ECCV 2012*, pp. 186–199, Springer, 2012.
- [26] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3490–3497, Providence, RI, USA, June 2012.
- [27] L. Shen, S. Wang, G. Sun, S. Jiang, and Q. Huang, "Multi-level discriminative dictionary learning towards hierarchical visual categorization," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 383–390, Portland, Ore, USA, June 2013.
- [28] M. Yang, D. Dai, L. Shen, and L. Van Gool, "Latent dictionary learning for sparse representation based classification," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 4138–4145, June 2014.
- [29] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang, "Support vector guided dictionary learning," in *Computer Vision—ECCV 2014*, pp. 624–639, Springer, 2014.
- [30] Y. C. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, 1993.
- [31] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B. Methodological*, vol. 58, no. 1, pp. 267–288, 1996.
- [32] E. Elhamifar and R. Vidal, "Robust classification using structured sparse representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1873–1879, June 2011.
- [33] Y.-T. Chi, M. Ali, A. Rajwade, and J. Ho, "Block and group regularized sparse modeling for dictionary learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 377–382, Portland, Ore, USA, June 2013.
- [34] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 801–808, December 2006.
- [35] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643–660, 2001.

- [36] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories,” *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
- [37] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1150–1157, Kerkyra, Greece, September 1999.
- [38] 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 (CVPR '09)*, pp. 1794–1801, 2009.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>