Learning A Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD

Zhuolin Jiang[†], Zhe Lin[§], Larry S. Davis[†]
[†]University of Maryland, College Park, MD, 20742
[§]Adobe Systems Incorporated, San Jose, CA, 95110
{zhuolin,lsd}@umiacs.umd.edu, zlin@adobe.com

Abstract

A label consistent K-SVD (LC-KSVD) algorithm to learn a discriminative dictionary for sparse coding is presented. In addition to using class labels of training data, we also associate label information with each dictionary item (columns of the dictionary matrix) to enforce discriminability in sparse codes during the dictionary learning process. More specifically, we introduce a new label consistent constraint called 'discriminative sparse-code error' and combine it with the reconstruction error and the classification error to form a unified objective function. The optimal solution is efficiently obtained using the K-SVD algorithm. Our algorithm learns a single over-complete dictionary and an optimal linear classifier jointly. It yields dictionaries so that feature points with the same class labels have similar sparse codes. Experimental results demonstrate that our algorithm outperforms many recently proposed sparse coding techniques for face and object category recognition under the same learning conditions.

1. Introduction

Sparse coding has been successfully applied to a variety of problems in computer vision and image analysis, including image denoising [5], image restoration [22, 25], and image classification [29, 28, 4]. Sparse coding approximates an input signal, y, by a sparse linear combination of items from an over-complete dictionary D. The performance of sparse coding relies on the quality of D. [28] employs the entire set of training samples as the dictionary for discriminative sparse coding, and achieves impressive performances on face recognition. Many algorithms [16, 18, 27] have been proposed to efficiently learn an over-complete dictionary that enforces some discriminative criteria.

To scale to large training sets, small-size dictionary learning has been developed by [28, 19, 1, 24, 33, 6]. In [28], training samples are manually selected to construct the dictionary. [19] learns a separate dictionary for each class, with classification then being performed based on reconstruction error. However, dictionary construction during training and sparse coding during testing are typically time-consuming when there are a large number of classes. In [1], a dictionary learning algorithm, K-SVD, is introduced to efficiently learn an over-complete dictionary from a set of training signals. This method has been applied to infill missing pixels and to image compression. K-SVD focuses on the representational power (best sparse representation for the training signals) of the learned dictionary, but does not consider the discrimination capability of the dictionary. The method of optimal directions (MOD) [6] shares the same effective sparse coding as K-SVD. [24] obtains a discriminative dictionary by iteratively updating dictionary items based on the results of a linear predictive classier. Discriminative K-SVD algorithm (D-KSVD) proposed in [33] unifies the dictionary and classifier learning processes.

We present a supervised algorithm to learn a compact and discriminative dictionary for sparse coding. We explicitly incorporate a 'discriminative' sparse coding error criterion and an 'optimal' classification performance criterion into the objective function and optimize it using the K-SVD algorithm. The learned dictionary is then both reconstructive and discriminative, in contrast to traditional constructive ones [6, 1, 28, 22]. The learned dictionary provides 'discriminative' sparse representations of signals; hence we achieve good accuracy on object classification even with a simple multiclass linear classifier, in contrast to other existing sparse coding approaches [30, 20, 19, 29] which learn one classifier for each pair of categories. Our approach is efficient and bounded by the complexity of K-SVD. It learns the discriminative dictionary and the linear classifier simultaneously. This is in contrast to dictionary learning approaches such as [30, 20] which iteratively solve subproblems in order to approximate a joint solution, or those approaches [13, 3, 19, 21] which learn the dictionary and classifier separately.

Sec. 2 presents the objective function for learning a reconstructive dictionary and a discriminative dictionary. Sec. 3 describes a label consistent K-SVD algorithm for simultaneously learning a dictionary with a discriminative and reconstructive criteria, and an optimal multiclass linear classifier. Sec. 4 presents experimental results and analysis. Sec. 5 concludes the paper and discusses future work.

1.1. Related Work

Supervised dictionary learning techniques for sparse coding have attracted much attention in recent years. Some approaches learn multiple dictionaries or category-specific dictionaries [34, 31, 19]. Ref. [34, 31] incorporate a boosting procedure into dictionary learning and learn multiple dictionaries with complementary power. In [19], one dictionary is learned for each class; Classification is based on the corresponding reconstruction error - it does not leverage the sparse codes.

Algorithms which attempt to incorporate discriminative terms into the objective function during training have been described in [30, 20, 33, 19, 21, 24, 3, 13]. The discrimination criteria include softmax discriminative cost function [19, 21, 20, 3], Fisher discrimination criterion [13], linear predictive classification error [33, 24] and hinge loss function [30, 17].

Most previous approaches treat dictionary learning and classifier training as two separate processes, *e.g.* [13, 3, 19, 12, 21, 34, 26]. In these approaches, a dictionary is typically learned first and then a classifier is trained based on it. Sparse representations are used as image features trained later with a classifier, *e.g.* SVM. More sophisticated approaches [30, 24, 20, 33] unify these two processes into a mixed reconstructive and discriminative formulation. Our approach also falls into this category. Ref. [30, 20] learn simultaneously an over-complete dictionary and multiple classification models for each class, which might not scale well to a large number of classes. The approach in [24] iteratively updates the dictionary based on the outcome of a linear classifier; it may suffer from a local minimum problem because it alternates between dictionary construction and classifier design. Ref. [33] incorporated classification error into the objective function, but it does not guarantee the discriminability of the resulting sparse codes when using a small-size dictionary.

Compared to these approaches, our approach more effectively learns a single compact discriminative dictionary and a universal multiclass linear classifier (for all categories) simultaneously. Our learned dictionary has good representational power (spanning all subspaces of object classes), and enforces better discrimination capabilities for all classes.

2. Dictionary Learning

2.1. Dictionary Learning for Reconstruction and Sparse Coding

Let Y be a set of n-dimensional N input signals, *i.e.* $Y = [y_1 ... y_N] \in R^{n \times N}$. Learning a reconstructive dictionary with K items for sparse representation of Y can be accomplished by solving the following problem:

$$< D, X >= \arg\min_{D,X} \|Y - DX\|_2^2 \quad s.t. \forall i, \|x_i\|_0 \leq T \quad (1)$$

where $D = [d_1 ... d_K] \in R^{n \times K}$ ($K > n$, making the dictionary over-complete) is the learned dictionary, $X = [x_1, ..., x_N] \in R^{K \times N}$ are the sparse codes of input signals Y, and T is a sparsity constraint factor (each signal has fewer than T items in its decomposition). The term $\|Y - DX\|_2^2$ denotes the reconstruction error.

The construction of D is achieved by minimizing the reconstruction error and satisfying the sparsity constraints. The K-SVD algorithm [1] is an iterative approach to minimize the energy in Equation 1 and learns a reconstructive dictionary for sparse representations of signals. It is highly efficient and works well in applications such as image restoration and compression.

Given D, sparse coding computes the sparse representation X of Y by solving:

$$X = \arg\min_{X} \|Y - DX\|_2^2 \quad s.t. \forall i, \|x_i\|_0 \leq T \quad (2)$$

2.2. Dictionary Learning for Classification

The sparse code x can be directly used as a feature for classification. A good classifier $f(x)$ can be obtained by determining its model parameters $W \in R^{m \times K}$ satisfying:

$$W = \arg\min_{W} \sum_i \mathcal{L}\{h_i, f(x_i, W)\} + \lambda_1 \|W\|_F^2 \quad (3)$$

where $\mathcal{L}$ is the classification loss function, h_i is the label of y_i and λ_1 is a regularization parameter (which prevents overfitting). Typical loss functions are the logistic loss function [20], square hinge loss [30] and a simple quadratic loss function [24, 33].

Separating the dictionary learning from the classifier learning might make D suboptimal for classification. It is possible to jointly learn the dictionary and classification model, as in [30, 24, 20, 33], which attempt to optimize the learned dictionary for classification tasks. In this case, an objective function for learning D and W jointly can be defined as:

$$< D, W, X >= \arg\min_{D,W,X} \|Y - DX\|_2^2$$
$$+ \sum_i \mathcal{L}\{h_i, f(x_i, W)\} + \lambda_1 \|W\|_F^2 \; s.t. \forall i, \|x_i\|_0 \leq T \quad (4)$$

Practically, these approaches appear to require learning relatively large dictionaries to achieve good classification performance, leading to high computation cost. This problem is aggravated when good classification results can only be obtained using a classification architecture based on multiple pairwise classifiers, as in [30, 20].

We will show that good classification results are obtained using only a small, single unified dictionary (and a single multiclass linear classifier) by a simple extension to the objective function for joint dictionary and classifier construction. This extension enforces a label consistency constraint

on the dictionary - intuitively that the class distributions that a dictionary element 'contributes' to during classification are highly peaked in one class. We refer to this method as label consistent K-SVD (LC-KSVD) since it employs the original K-SVD algorithm to obtain its solution.

3. Label Consistent K-SVD

We aim to leverage the supervised information (i.e. labels) of input signals to learn a reconstructive and discriminative dictionary. Each dictionary item will be chosen so that it represents a subset of the training signals ideally from a single class, so each dictionary item d_k can be associated with a particular label. Hence there is an explicit correspondence between dictionary items and the labels in our approach.

We subsequently focus on the effects of adding a label consistent regularization term, and a joint classification error and label consistent regularization term into the objective function in Equation 1 for learning a dictionary with more balanced reconstructive and discriminative power. We refer to them as LC-KSVD1 and LC-KSVD2, respectively, in the following.

3.1. LC-KSVD1

The performance of the linear classifier depends on the discriminability of the input sparse codes x. For obtaining discriminative sparse codes x with the learned D, an objective function for dictionary construction is defined as:

$$< D, A, X >= \arg \min_{D,A,X} \|Y - DX\|_2^2 + \alpha \|Q - AX\|_2^2 \; s.t. \forall i, \|x_i\|_0 \leq T \quad (5)$$

where α controls the relative contribution between reconstruction and label consistent regularization, and $Q = [q_1 ... q_N] \in R^{K \times N}$ are the 'discriminative' sparse codes of input signals Y for classification. We say that $q_i = [q_i^1 ... q_i^K]^t = [0...1, 1, ...0]^t \in R^K$ is a 'discriminative' sparse code corresponding to an input signal y_i, if the non-zero values of q_i occur at those indices where the input signal y_i and the dictionary item d_k share the same label. For example, assuming $D = [d_1 ... d_6]$ and $Y = [y_1 ... y_6]$, where y_1, y_2, d_1 and d_2 are from class 1, y_3, y_4, d_3 and d_4 are from class 2, and y_5, y_6, d_5 and d_6 are from class 3, Q can be defined as:

$$Q \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

A is a linear transformation matrix. Here we identify a linear transformation, $g(x; A) = Ax$, which transforms the original sparse codes x to be most discriminative in sparse feature space R^K.

The term $\|Q - AX\|_2^2$ represents the discriminative sparse-code error, which enforces that the sparse codes X approximate the discriminative sparse codes Q. It forces the signals from the same class to have very similar sparse representations (*i.e.* encouraging label consistency in resulting sparse codes), which results in good classification performance using a simple linear classifier.

3.2. LC-KSVD2

As in [30, 24, 20, 33], we aim to include the classification error as a term in the objective function for dictionary learning, in order to make the dictionary optimal for classification. Here we use a linear predictive classifier $f(x; W) = Wx$. An objective function for learning a dictionary D having both reconstructive and discriminative power can be defined as follows:

$$< D, W, A, X >= \arg \min_{D,W,A,X} \|Y - DX\|_2^2 + \alpha \|Q - AX\|^2 + \beta \|H - WX\|_2^2 \; s.t. \forall i, \|x_i\|_0 \leq T \quad (6)$$

where the term $\|H - WX\|_2^2$ represents the classification error. W denotes the classifier parameters. $H = [h_1 ... h_N] \in R^{m \times N}$ are the class labels of input signals Y. $h_i = [0, 0...1...0, 0]^t \in R^m$ is a label vector corresponding to an input signal y_i, where the non-zero position indicates the class of y_i. α and β are the scalars controlling the relative contribution of the corresponding terms.

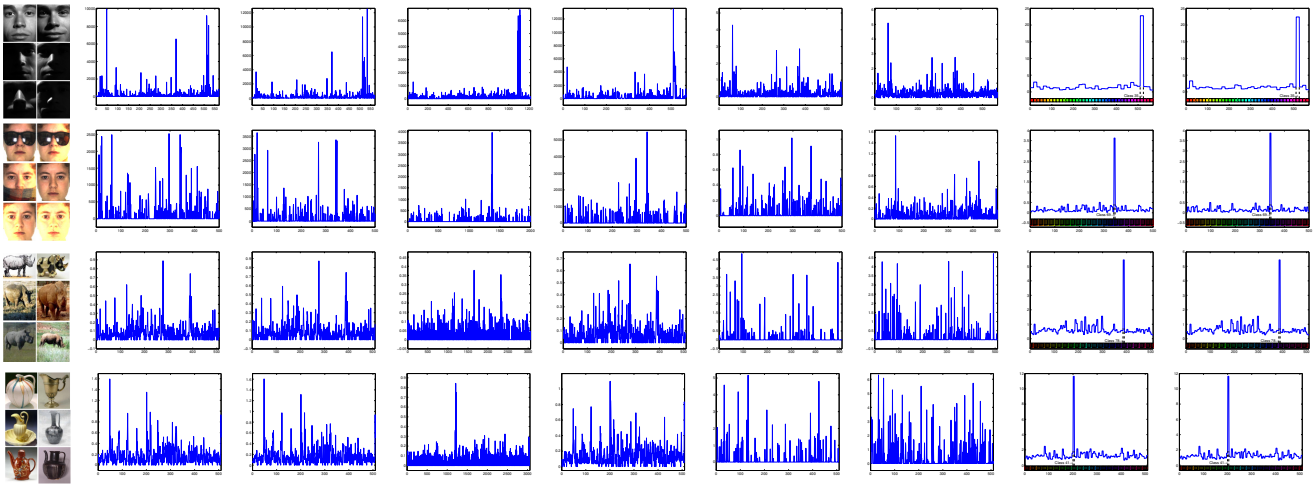
Assuming discriminative sparse codes $X' = AX$ and $A \in R^{K \times K}$ is invertible, then $D' = DA^{-1}$,$W' = WT^{-1}$. The objective function in Equation 6 can be rewritten as:

$$< D', W', X' >= \arg \min_{D',W',X'} \|Y - D'X'\|_2^2 + \alpha \|Q - X'\|_2^2 + \beta \|H - W'X'\|_2^2 \; s.t. \forall i, \|x_i\|_0 \leq T \quad (7)$$

The first term represents the reconstruction error, the second term the discriminative sparse-code error, and the third term the classification error. The second term $\|Q - X'\|_2^2$ can make the sparse codes discriminative between classes while the third term $\|H - W'X'\|$ supports learning an optimal classifier.

The dictionary learned in this way will be adaptive to the underlying structure of the training data (leading to a good representation for each member in the set with strict sparsity constraints), and will generate discriminative sparse codes X regardless of the size of the dictionary. These sparse codes can be utilized directly by a classifier, such as in [28]. The discriminative property of sparse code x is very important for the performance of a linear classifier.

In the following section, we describe the optimization procedure for LC-KSVD2. LC-KSVD1 utilizes the same procedure except that the H and W components in Equation 8 and Equation 14 are excluded. During training, D, A and X are computed first by Equation 5, and then the matrix W is trained using Equation 14 for classification.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 1. Examples of sparse codes using different sparse coding approaches on the three evaluated datasets. X axis indicates the dimensions of sparse codes, Y axis indicates a sum of absolute sparse codes for different testing samples from the same class. The curves in 1_{st}, 2_{nd}, 3_{rd} and 4_{th} row correspond to class 35 in Extended YaleB (32 testing frames), class 69 in AR Face (6 testing frames), class 78 (29 testing frames) and class 41 (55 testing frames) in Caltech101 respectively. (a) are sample images from these classes. The sparse coding approaches include: (b) K-SVD [1]; (c) D-KSVD [33]; (d) SRC [28]; (e) SRC* [28]; (f) LLC [27] (30 local bases); (g) LLC [27] (70 local bases); (h) LC-KSVD1; (i) LC-KSVD2. Each color from the color bars in (h) and (i) represents one class for a subset of dictionary items. The black dashed lines demonstrate that the curves are highly peaked in one class. The two examples from Caltech101 demonstrate that a large intra-class difference is enforced between classes via LC-KSVD. The figure is best viewed in color and 600% zoom in.

3.3. Optimization

We use the efficient K-SVD algorithm to find the optimal solution for all parameters simultaneously. Equation 6 can be rewritten as:

$$< D, W, A, X >= \arg \min_{D,W,A,X} \| \begin{pmatrix} Y \\ \sqrt{\alpha}Q \\ \sqrt{\beta}H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\alpha}A \\ \sqrt{\beta}W \end{pmatrix} X \|_2^2 \quad s.t. \forall i, \|x_i\|_0 \leq T \quad (8)$$

Let $Y_{new} = (Y^t, \sqrt{\alpha}Q^t, \sqrt{\beta}H^t)^t$, $D_{new} = (D^t, \sqrt{\alpha}A^t, \sqrt{\beta}W^t)^t$. The matrix D_{new} is L_2 normalized column-wise. The optimization of Equation 8 is equivalent to solving the following problems:

$$< D_{new}, X >= \arg \min_{D_{new},X} \{\|Y_{new} - D_{new}X\|_2^2\} \quad s.t. \forall i, \|x_i\|_0 \leq T \quad (9)$$

This is exactly the problem that K-SVD [1] solves. Following K-SVD, d_k and its corresponding coefficients, the k-th row in X, denoted as x_R^k, are updated at a time. Let $E_k = (Y - \sum_{j \neq k} d_j x_R^j)$, and $\widetilde{x}_R^k$, $\widetilde{E}_k$ denote the result of discarding the zero entries in x_R^k and E_k, respectively. d_k and $\widetilde{x}_R^k$ can be computed by solving the following problem:

$$< d_k, \widetilde{x}_R^k >= \arg \min_{d_k, \widetilde{x}_R^k} \{\|\widetilde{E}_k - d_k \widetilde{x}_R^k\|_F^2\} \quad (10)$$

A SVD operation is performed for $\widetilde{E}_k$, *i.e.* $U\Sigma V^t = \mathtt{SVD}(\widetilde{E}_k)$. Then d_k and $\widetilde{x}_R^k$ are computed as:

$$\begin{aligned} d_k &= U(:,1) \\ \widetilde{x}_R^k &= \Sigma(1,1)V(:,1) \end{aligned} \quad (11)$$

Finally $\widetilde{x}_R^k$ is used to replace the non-zero values in x_R^k.

LC-KSVD learns D, A and W simultaneously, which avoids the problem of local minima and is scalable to a large number of classes. In addition, it allows us to easily combine another discriminative term, *i.e.* discriminative sparse-code error, into the objective function. It produces a discriminative sparse representation regardless of the size of the dictionary. Figure 1 shows examples of sparse codes of one testing class from three evaluated datasets using different approaches. From this figure, we can see that LC-KSVD ensures that signals from the same class have similar sparse codes, which is very important for linear classification.

3.3.1 Initialization of LC-KSVD

We need to initialize the parameters D_0, A_0 and W_0 for LC-KSVD. For D_0, we employ several iterations of K-SVD within each class and then combine all the outputs (*i.e* dictionary items learning from each class) of each K-SVD. The label of each dictionary item d_k is then initialized based on the class it corresponds to and will remain fixed during the

entire dictionary learning process [1], although d_k is updated during the learning process. Dictionary elements are uniformly allocated to each class with the number of the elements proportional to the dictionary size.

In order to initialize A_0, we employ the multivariate ridge regression model [10], with the quadratic loss and L_2 norm regularization, as follows:

$$A = \arg\min_A \|Q - AX\|^2 + \lambda_2 \|A\|_2^2 \tag{12}$$

which yields the following solution:

$$A = (XX^t + \lambda_2 I)^{-1} XQ^t \tag{13}$$

Similarly, for W_0, we again use the ridge regression model and obtain the following solution:

$$W = (XX^t + \lambda_1 I)^{-1} XH^t \tag{14}$$

Given the initialized D_0, we perform the original K-SVD algorithm to compute the sparse codes X of training signals Y. Then the X can be used to compute the initial A_0 in Equation 13 and W_0 in Equation 14.

3.4. Classification Approach

We obtain $D = \{d_1 ... d_K\}$, $A = \{a_1 ... a_K\}$ and $W = \{w_1 ... w_K\}$ from D_{new} by employing the K-SVD algorithm [1]. We cannot simply use D, A and W for testing since D, A and W are L_2-normalized in D_{new} jointly in the LC-KSVD algorithm, *i.e.*$\forall k, \|(d_k^t, \sqrt{\alpha} a_k^t, \sqrt{\beta} w_k^t)^t\|_2 = 1$. The desired dictionary $\hat{D}$, transform parameters $\hat{A}$ and classifier parameters $\hat{W}$ are computed as follows:

$$\begin{aligned} \hat{D} &= \{\frac{d_1}{\|d_1\|_2}, \frac{d_2}{\|d_2\|_2} ... \frac{d_K}{\|d_K\|_2}\} \\ \hat{A} &= \{\frac{a_1}{\|d_1\|_2}, \frac{a_2}{\|d_2\|_2} ... \frac{a_K}{\|d_K\|_2}\} \\ \hat{W} &= \{\frac{w_1}{\|d_1\|_2}, \frac{w_2}{\|d_2\|_2} ... \frac{w_K}{\|d_K\|_2}\} \end{aligned} \tag{15}$$

For a test image y_i, we first compute its sparse representation x_i by solving the optimization problem:

$$x_i = \arg\min_{x_i} \{\|y_i - \hat{D} x_i\|_2^2\} \quad s.t. \|x_i\|_0 \leq T \tag{16}$$

Then we simply use the linear predictive classifier $\hat{W}$ to estimate the label j of the image y_i:

$$j = \arg\max_j (l = \hat{W} x_i) \tag{17}$$

where $l \in R^m$ is the class label vector.

[1] We do associate a unique and fixed class label to each dictionary item, but an input signal of a class certainly can (and does) use dictionary items from other classes, as the sparse codes in Figure 1(h) and 1(i) illustrate.

4. Experiments

We evaluate our approach on two public face databases: Extended YaleB database [9], AR face database [23], and on a multiclass object category dataset: Caltech101 [7]. We compare our results with K-SVD [1], D-KSVD [33], sparse representation-based classification (SRC) [28] and LLC [27] [2].

The feature descriptors used in the Extended YaleB database and AR Face database are random faces [28, 33], which are generated by projecting a face image onto a random vector. We follow the standard setting in [33]. The size of a random-face feature in Extended YaleB is 504 while the size in AR face is 540. For the Caltech101 dataset, we first extract sift descriptors from 16×16 patches which are densely sampled using a grid with a step size of 6 pixels; then we extract the spatial pyramid feature [15] based on the extracted sift features with three grids of size 1×1, 2×2 and 4×4. To train the codebook for spatial pyramid, we use the standard *k*-means clustering with $k = 1024$. Finally, the spatial pyramid feature is reduced to 3000 dimensions by PCA.

In each spatial sub-region of the spatial pyramid, the vector quantization codes are pooled together to form a pooled feature. These pooled feature from each sub-region are concatenated and normalized as the final spatial pyramid feature of an image. The sparse codes for the Caltech101 dataset are computed from spatial pyramid features. There are two kinds of pooling methods: (1) sum pooling [15]: $x_{out} = x_1 +, ..., + x_n$; (2) max pooling [29]: $x_{out} = max(x_1, ..., x_n)$, where x_i is the vector quantization code. Then these pooled features are normalized by: (1) L_1 normalization: $x_{out} = x_{out} / \sum_i x_i$; (2) L_2 normalization: $x_{out} = x_{out} / \|x_{out}\|_2$. Different combinations are evaluated in our experiment. Following the common evaluation procedure, we repeat the experiments 10 times with different random spits of the training and testing images to obtain reliable results. The final recognition rates are reported as the average of each run. The sparsity factor used in all of our experiments is 30.

4.1. Evaluation on the Extended YaleB Database

The Extended YaleB database contains $2,414$ frontal-face images of 38 persons [9]. There are about 64 images for each person. The original images were cropped to 192×168 pixels. This database is challenging due to varying illumination conditions and expressions. We randomly select half of the images (about 32 images per person) as training and the other half for testing. Each face image is projected onto a 504-dimensional vector with a randomly generated matrix from a zero-mean normal dis-

[2] D-KSVD and SRC results are based on our own implementations, which allowed us to standardize the learning parameters across methods.

Table 1. Recognition results using random-face features on the Extended YaleB database. The 2nd column is the result when we used all 64 images for each person. The 3rd column is the result when we removed 10 poor-quality images for each person.

Method	Acc.(%)	Acc.(%)
K-SVD(15 per person) [1]	93.1	98.0
D-KSVD(15 per person) [33]	94.1	98.0
SRC(all train. samp.) [28]	**97.2**	**99.0**
SRC*(15 per person) [28]	80.5	86.7
LLC(30 local bases) [27]	82.2	92.1
LLC(70 local bases) [27]	90.7	96.7
LC-KSVD1(15 per person)	94.5	98.3
LC-KSVD2(15 per person)	95.0	98.8
LC-KSVD2(all train. samp.)	96.7	**99.0**

Table 2. Computation time for classifying a test image on the Extended YaleB database.

Method	Avg. Time (ms)
SRC(all training samples) [28]	20.78
SRC*(15 per person) [28]	11.22
LC-KSVD1(15 per person)	0.52
LC-KSVD2(15 per person)	0.49

tribution. Each row of the matrix is L_2 normalized. The learned dictionary consists of 570 items, which corresponds to an average of 15 items per person. Unlike the K-SVD [1] and D-KSVD [33], there is an explicit correspondence between the dictionary items and the labels of people in our approach, which is similar to the SRC algorithm [28] but our approach uses fewer training samples.

We evaluate our approach and compare it with K-SVD [1], D-KSVD [33], SRC [28] and the recently proposed LLC algorithm [27]. We measure the performance of the SRC algorithm using dictionaries with two different sizes (all training samples and 15 samples per person). For fair comparison, the number of local bases, which determines the sparsity of the LLC codes, is chosen as the same value as the sparsity factor (*i.e.* $T = 30$) used in our implementation. For evaluating the importance of the number of local bases, we also evaluate the LLC algorithm with 70 local bases.

The results are summarized in Table 1. The weight factors α and β are set to 16 and 4 respectively in our experiment. Most of the failure cases are from images taken under extremely bad illumination conditions. Hence we perform another experiment with these bad images excluded (about 10 for each person). The results of this experiment are listed in the third column of Table 1. Our approaches always achieve better results than the KSVD, D-KSVD, LLC. Additionally, our approach outperforms the competing SRC algorithm when it uses the same size dictionary (*i.e.* SRC*).

In addition, we compare our approaches with SRC in terms of the computation time for classifying one test image. The time is computed as an average over all the test images. As shown in Table 2, our approach is approximately 22 times faster than SRC*. If a database is provided with more categories, a small dictionary for sparse coding can save even more time (see the results for AR face database).

Table 3. Recognition results using random face features on the AR face database.

Method	Acc. (%)
K-SVD(5 per person) [1]	86.5
D-KSVD(5 per person) [33]	88.8
SRC(all train. samp.) [28]	97.5
SRC*(5 per person) [28]	66.5
LLC(30 local bases) [27]	69.5
LLC(70 local bases) [27]	88.7
LC-KSVD1(5 per person)	92.5
LC-KSVD2(5 per person)	93.7
LC-KSVD2(all train. samp.)	**97.8**

Table 4. Computation time for classifying a test image on the AR face database.

Method	Avg. Time (ms)
SRC(all training samples) [28]	83.79
SRC*(5 per person) [28]	17.76
LC-KSVD1(5 per person)	0.541
LC-KSVD2(5 per person)	0.479

4.2. Evaluation on the AR Face Database

The AR face database consists of over $4,000$ color images of 126 persons. Each person has 26 face images taken during two sessions. Compared to the Extended YaleB database, these images include more facial variations including different illumination conditions, different expressions and different facial 'disguises' (sunglasses and scarves). Following the standard evaluation procedure, we use a subset of the database consisting of 2600 images from 50 male subjects and 50 female subjects. For each person, we randomly select 20 images for training and the other 6 for testing. Each face image of size 165×120 pixels, is projected onto a 540-dimensional vector with a randomly generated matrix. The learned dictionary has 500 dictionary items, $i.e.$ 5 items per person. As discussed earlier, there is an explicit correspondence between the dictionary items and the labels of people.

We evaluate our approaches using random face features and compared with state-of-art approaches including K-SVD [1], D-KSVD [33], SRC [28] and LLC [27]. For SRC, we learn two dictionaries with two different dictionary sizes. All the approaches use the same learning parameters. The recognition results are summarized in Table 3. Our approaches outperform K-SVD, D-KSVD, LLC and SRC*. Note that SRC degrades dramatically when it uses 5 samples per person.

In addition, we compare our approaches with SRC in terms of the computation time for classifying one test image. As shown in Table 4, our approach is approximately 35 times faster than SRC*. As expected, a small dictionary can save more time for a database consisting of many training images, and the performance does not degrade much compared to using the entire set of training images as the dictionary.

4.3. Evaluation on the Caltech101 Dataset

The Caltech101 dataset [7] contains 9144 images from 102 classes (*i.e.* 101 object classes and a 'background'

class) including animals, vehicles, flowers, etc. The samples from each category have significant shape variability. The number of images in each category varies from 31 to 800. Following the common experimental settings, we train on 5, 10, 15, 20, 25 and 30 samples per category and test on the rest.

We evaluate our approach using spatial pyramid features and compare the results with K-SVD [1], D-KSVD [33], SRC [28] and other state-of-art approaches [32, 15, 11, 2, 14, 24, 8, 29, 27]. The comparative results are shown in Table 5. Our approaches consistently outperform all the competing approaches. The basic reason for the good recognition performance, even with only a few training examples, is that the new label consistent constraint encourages the input signals from the same class to have similar sparse codes and those from different classes to have dissimilar sparse codes.

We randomly select 30 images per category as training data, and evaluate our approach using different dictionary sizes $K = 510, 1020, 1530, 2040, 2550$ and 3060; We compare the classification accuracy of K-SVD [1], D-KSVD [33] and SRC [28]. Figure 2(a) shows that our approaches maintain a high classification accuracy and outperform the other three competing approaches significantly when we use a smaller size dictionary.

We also compare the training time of a dictionary with K-SVD and D-KSVD. As shown in Figure 2(b), we can train approximately 13 times faster when we use a dictionary with a size of 510 rather than 3060. More importantly, as shown in Figure 2(a), the classification accuracy of LC-KSVD degrades only slightly when using the dictionary of size 510. We compare our approach with SRC in terms of computation time of classifying one test images, using different dictionary sizes. As shown in Table 6, the computation time of our approach is significantly faster (about 310 times) than SRC.

In addition, we evaluate different combinations of pooling methods and normalization methods for computing spatial pyramid features. As can be seen from Figure 2(c), 'max pooling' followed by 'L_2 normalization' generates the best classification accuracy. Note that 'sum pooling' followed by 'L_1 normalization' generates the histograms, which were used in the original spatial pyramid features [15].

There were a total of 9 classes achieving 100% classification accuracy when using 30 training images per category. Figure 3 shows some samples from seven of these classes.

5. Conclusions

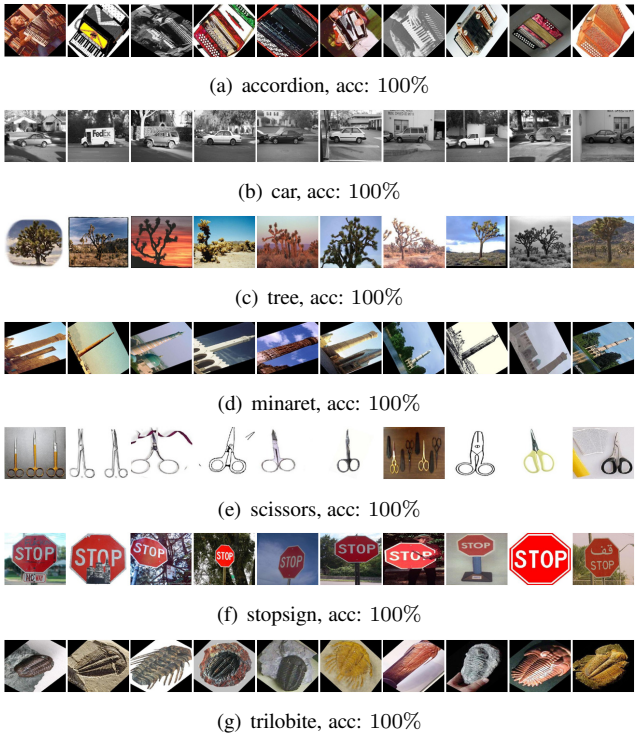
We proposed a new dictionary learning approach, label consistent K-SVD (LC-KSVD) algorithm, for sparse coding. Our main contribution lies in explicitly integrating the 'discriminative' sparse codes and a single predictive linear classifier into the objective function for dictionary learning. Additionally, the solution to the new objective function is efficiently achieved by simply employing the original K-SVD algorithm. Unlike most existing dictionary learning approaches that rely on iteratively solving sub-problems in order to approximate a global solution, our approach is able to learn the dictionary, discriminative coding parameters

Table 5. Recognition results using spatial pyramid features on the Caltech101 dataset.

number of train. samp.	5	10	15	20	25	30
Malik [32]	46.6	55.8	59.1	62.0	-	66.20
Lazebnik [15]	-	-	56.4	-	-	64.6
Griffin [11]	44.2	54.5	59.0	63.3	65.8	67.60
Irani [2]	-	-	65.0	-	-	70.40
Grauman [14]	-	-	61.0	-	-	69.10
Venkatesh [24]	-	-	42.0	-	-	-
Gemert [8]	-	-	-	-	-	64.16
Yang [29]	-	-	67.0	-	-	73.20
Wang [27]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [28]	48.8	60.1	64.9	67.7	69.2	70.7
K-SVD [1]	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD [33]	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**

Table 6. Computation time (ms) for classifying a test image on the Caltech101 dataset.

Dictionary size	510	1020	1530	2040	2550	3060
SRC [28]	173.44	343.12	520.88	662.40	835.34	987.55
LC-KSVD1	0.59	1.09	1.62	2.21	2.83	3.50
LC-KSVD2	0.54	0.98	1.44	1.94	2.50	3.17



(a) accordion, acc: 100%

(b) car, acc: 100%

(c) tree, acc: 100%

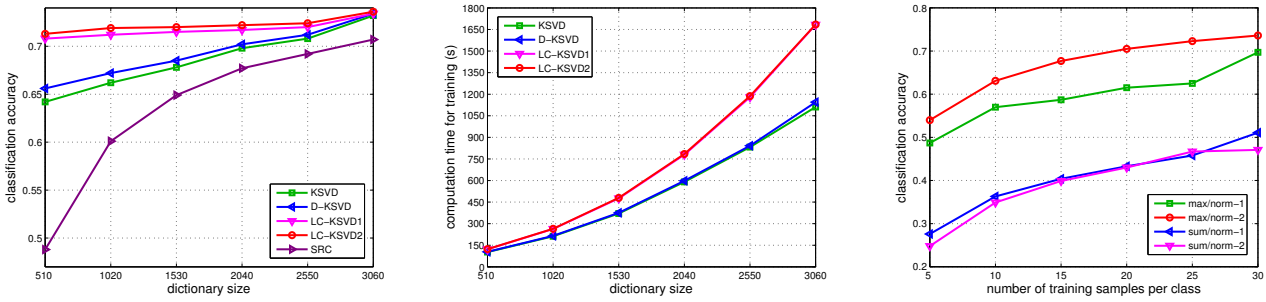
(d) minaret, acc: 100%

(e) scissors, acc: 100%

(f) stopsign, acc: 100%

(g) trilobite, acc: 100%

Figure 3. Example images from classes with high classification accuracy from the caltech101 dataset.



(a) (b) (c)

Figure 2. Performance comparisons on the Caltech101. (a) Performance on the Caltech101 with varying dictionary size; (b) Training time on the Caltech101 with varying dictionary size; (c) Performance on the Caltech101 with different spatial-pyramid-matching settings.

and classifier parameters simultaneously.

The experimental results show that our approach yields very good classification results on three well-known public datasets with only one simple linear classifier, which is unlike some competing approaches learning multiple classifiers for categories to gain discrimination between classes. Our approach outperforms recently proposed methods including D-KSVD [33], SRC [28] and LLC [27], especially when the number of training samples is small. Possible future work includes extending our approach to learn a dictionary with a discriminative and representation power based on local image patches, and apply it to object recognition.

Acknowledgement

This work was funded, in part, by Army Research MURI project (contract number: W911NF-09-10383).

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionries for sparse representation. *IEEE Trans. on Signal Processing*, 54(1):4311–4322, 2006.

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighor based image classification, 2008. *CVPR*.

[3] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition, 2010. *CVPR*.

[4] D. Bradley and J. Bagnell. Differential sparse coding, 2008. *NIPS*.

[5] M. Elad and M. Aharon. Image denosing via sparse and redundant representations over learned dictionaries. *IEEE Trans. Img. Proc.*, 54(12):3736–3745, 2006.

[6] K. Engan, S. Aase, and J. Husϕy. Frame based signal compression using method of optimal directions (mod), 1999. *IEEE Intern. Symp. Circ. Syst., 1999*.

[7] L. FeiFei, R. Fergus, and P. Perona. Learning generative visual models from few training samples: An incremental bayesian appoach tested on 101 object categories, 2004. *CVPR Workshop on Generative Model Based Vision*.

[8] J. Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization, 2008. *ECCV*.

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI*, 23(6):643–660, 2001.

[10] G. Golub, P. Hansen, and D. O'leary. Tikhonov regularization and total least squares. *SIM J. Matrix Anal. Appl.*, 21(1):185–194, 1999.

[11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset, 2007. *CIT Technical Report 7694*.

[12] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification, 2007. *Conf. on Uncertainty in AI*.

[13] K. Huang and S. Aviyente. Sparse representation for signal classification, 2007. *NIPS*.

[14] P. Jain, B. Kullis, and K. Grauman. Fast image search for learned metrics, 2008. *CVPR*.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, 2007. *CVPR*.

[16] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms, 2006. *NIPS*.

[17] X. Lian, Z. Li, B. Lu, and L. Zhang. Max-margin dictionary learning for multiclass image categorization, 2010. *ECCV*.

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

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis, 2008. *CVPR*.

[20] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning, 2009. *NIPS*.

[21] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edage detection and image interpretation, 2008. *ECCV*.

[22] J. Marial, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Img. Proc.*, 2008.

[23] A. Martinez and R. Benavente. The ar face database, 1998. *CVC Technical Report 24*.

[24] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition, 2008. *CVPR*.

[25] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model, 2006. *NIPS*.

[26] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaryies, 2007. *IMA Preprint 2213*.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. huang, and Y. Gong. Locality-constrained linear coding for image classification, 2010. *CVPR*.

[28] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2):210–227, 2009.

[29] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification, 2009. *CVPR*.

[30] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding, 2010. *CVPR*.

[31] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook genearation with classifier training for object category recognition, 2008. *CVPR*.

[32] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition, 2006. *CVPR*.

[33] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition, 2010. *CVPR*.

[34] W. Zhang, A. Surve, X. Fern, and T. Dietterich. Learning non-redundant codebooks for classifying complex objects, 2009. *ICML*.