Discriminative K-SVD for dictionary learning in face recognition

Conference Paper *in* Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition · June 2010

DOI: 10.1109/CVPR.2010.5539989 · Source: DBLP

Qiang Zhang
Samsung

Discriminative K-SVD for Dictionary Learning in Face Recognition

Qiang Zhang and Baoxin Li
Computer Science and Engineering
Arizona State University, Tempe, AZ
{qzhang3,baoxin.li}@asu.edu

Abstract

In a sparse-representation-based face recognition scheme, the desired dictionary should have good representational power (i.e., being able to span the subspace of all faces) while supporting optimal discrimination of the classes (i.e., different human subjects). We propose a method to learn an over-complete dictionary that attempts to simultaneously achieve the above two goals. The proposed method, discriminative K-SVD (D-KSVD), is based on extending the K-SVD algorithm by incorporating the classification error into the objective function, thus allowing the performance of a linear classifier and the representational power of the dictionary being considered at the same time by the same optimization procedure. The D-KSVD algorithm finds the dictionary and solves for the classifier using a procedure derived from the K-SVD algorithm, which has proven efficiency and performance. This is in contrast to most existing work that relies on iteratively solving sub-problems with the hope of achieving the global optimal through iterative approximation. We evaluate the proposed method using two commonly-used face databases, the Extended YaleB database and the AR database, with detailed comparison to 3 alternative approaches, including the leading state-of-the-art in the literature. The experiments show that the proposed method outperforms these competing methods in most of the cases. Further, using Fisher criterion and dictionary incoherence, we also show that the learned dictionary and the corresponding classifier are indeed better-posed to support sparse-representation-based recognition.

1. Introduction

Face recognition is a challenging computer vision task that has seen active research for many years [23]. Well-known, conventional approaches include Eigenface [21] and Fisherface [3], among others. These methods usually involve two stages: feature extraction and classification. Recently, a lot of attention has been given to applying sparse-representation-based techniques to computer vision and image processing problems, such as image denoising [10], image inpainting [16], image compression [4][5]. In particular, the SRC algorithm proposed in [22] uses sparse representation for face recognition: training face images are used to form a dictionary, and classifying a new face image is achieved through finding its sparse coefficients with respect to this dictionary. Unlike conventional methods such as Eigenface and Fisherface, SRC does not need an explicit feature extraction stage. The superior performance reported in [22] suggests that this is a promising direction for face recognition.

The basic way of forming the dictionary by using all the training images may result in a huge size for the dictionary, which is detrimental to the subsequent sparse solver. For example, we may have 32 images for each person (e.g., as in the Extended YaleB database [12]). Then the number of atoms in the dictionary will be 32 times the number of people. Thus for a large face database with thousands of people, the sheer size of the dictionary becomes a practical concern. One may manually select a subset of the training images to be used in the dictionary, as done in [22]. But this is not only tedious but also sub-optimal since there is no guarantee that the manually-selected images form the best dictionary. Methods for learning a small-sized dictionary for sparse-coding from the training data have been proposed recently. For example, the K-SVD algorithm [1] learns an over-complete dictionary from a set of signals. The algorithm has been shown to work well in image compression and denoising. K-SVD focuses on only the representational power of the dictionary (or the efficiency of sparse coding under the dictionary) without considering its capability for discrimination. Another recent work [18] attempts to address this issue by further iteratively updating the K-SVD-trained dictionary based on the outcome of a linear classifier, hence obtaining a dictionary that may be also good for classification in addition to having the representational power. Other efforts along similar direction include [14] and [15], which use more sophisticated objective functions in dictionary optimization in the training stage in order to gain some discriminative power for the dictionary.

In this paper, we propose to extend the K-SVD algorithm to learn an over-complete dictionary from a set of labeled training face images. By directly incorporating the labels in the dictionary-learning stage (as opposed to relying on iteratively updating the dictionary using feedback from the classification stage as in [18]), we can efficiently obtain a dictionary that retains the representational power while making the dictionary discriminative (i.e., supporting sparse-coding-based classification). We also propose a corresponding classification algorithm based on the learned dictionary. Incorporating the classification stage directly into the dictionary-learning procedure has the potential of avoiding the local minima that may be encountered more often in the approach of [18], which computes the sub-optimal solution by alternating between solving subset of parameters while fixing others. Furthermore, the complexity of the proposed method is bounded by that of the K-SVD, while the approach of [18] involves multiple additional optimization procedures.

To demonstrate the effectiveness and the advantage of the proposed method for face recognition, extensive experiments have been carried out using two commonly-used face databases: the extended YaleB database [13] and the AR database [17]. In addition to comparing the recognition rates of our method with those from existing state-of-the-art approaches, we also analyze and compare the performance of the classifiers based on Fisher criterion. The learned dictionaries are also compared in terms of dictionary incoherence [7][9]. The experimental results show that the proposed method has some clear advantages. In particular, the experiments show that with the same dictionary size or with dictionaries of randomly-chosen training images, our method can obtain better recognition rate than the SRC algorithm.

The rest of paper is organized as follows. We first briefly describe in Section 2 the basic formulation of the problem of face recognition based on spare-representation using an over-complete dictionary. Then we present the proposed algorithm in Section 3. The experiments and analysis of the results are reported in Section 4. We conclude with discussion in Section 5.

2. Basic Formulation of the Problem

It has been observed that images of human face under varying illumination conditions and expressions lie on a special low-dimensional space [3][2]. In a sparse-representation-based face recognition scheme like the SRC algorithm, this observation is exploited for recognition through sparse-coding of a testing face image using an over-complete dictionary of the training faces. Our method follows this scheme, which we briefly outline in the below.

Given sufficient samples of the i-th person, $A_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}] \in R^{m*n_i}$, any test sample $y \in R^m$ from the same class will approximately lie in the subspace spanned by the training samples associated with this class:

$$y = a_{i,1} * v_{i,1} + a_{i,2} * v_{i,2} + \cdots + a_{i,n_i} * v_{i,n_i} \qquad (1)$$

where $a_{i,j}$ is a scalar.

By grouping samples from all the classes, we form a dictionary D:

$$D = [A_1, A_2, \ldots, A_k] = [v_{1,1}, v_{1,2}, \ldots, v_{k,n_k}] \qquad (2)$$

where k is the number of classes. Then the linear representation of y can be written in terms of all samples as:

$$y = a_{1,1}*v_{1,1}+a_{1,2}*v_{1,2}+\cdots+a_{k,n_k}*v_{k,n_k} = D*x_0 \quad (3)$$

where $x_0 = [0, , \ldots, 0, a_{i,1}, a_{i,2}, \ldots, a_{i,n_i}, 0, \ldots, 0] \in R^n$ is a vector of coefficients whose entries are all zero except for those associated with the i-th class.

If we extract the coefficient $\alpha_0(j)$ associated with the j-th person and reconstruct the image as

$$y(j) = D * \alpha_0(j) \qquad (4)$$

we can expect that the reconstruction error $err(j) = \|y - y(j)\|_2$ will be large for any general $j \neq i$ except for $err(i)$. We can use this idea to recognize the test sample. While such a scheme has been shown to be able to generate the state-of-art results in [22], there are a few practical drawbacks associated with the method. For example,

1. In order to improve the representational power of the dictionary, we need to use a large number of training samples for each person. But a large dictionary is detrimental for the subsequent sparse solver.

2. In order to ensure that the dictionary atoms can span the underlying subspace reasonably well, we need to carefully choose the training images. For example, in [22], for the AR database, the authors manually chose 7 normal images (without artificial disguise) from Section 1 for each person.

3. Proposed Method

3.1. Adding Discrimination Ability to K-SVD

The above drawbacks associated with the SRC algorithm may be overcome if we can learn a smaller-sized dictionary from the given training images while maintaining the representational power of the dictionary. For example, the K-SVD algorithm [1] may be employed for this purpose, which finds a solution for the following problem:

$$< D, \alpha >= \underset{D,\alpha}{\operatorname{argmin}} \|Y - D * \alpha\|_2 \; subject\ to \; \|\alpha\|_0 \leq T \qquad (5)$$

where Y is the matrix of all input signals (the training face images in our case), and T is a parameter to impose the sparsity prior. In Eqn. 5, each column of D is normalized to have unit norm. This K-SVD formulation has been found to work well for real images in applications such as image denoising and face image compression. However, since the objective function in Eqn. 5 considers only the reconstruction error and the sparsity of the coefficient, the learned dictionary is not optimized for a classification task. In other words, the learned dictionary may not have the best discriminative power despite its representational power.

Efforts have been reported on improving a dictionary-learning procedure for classification tasks. In [14] and [15], an extra term was introduced to consider the classifier performance in dictionary learning. For a binary classifier, this term can be represented by

$$< \theta >= \underset{\theta}{\operatorname{argmin}} \sum_i C(h_i * f(\alpha_i, \theta)) + \lambda_1 * \|\theta\|_2 \quad (6)$$

where θ is the parameter of the classifier, h_i is the label and $C(x) = log(1 + e^{-x})$ is a logistic loss function . The resultant problem is very complex and thus there is no direct method to find the solution. Instead, projected gradient descent was used in finding approximate solutions in the paper. Another example is [18], which uses a simpler formulation for considering the classifier performance:

$$< W, b >= \underset{W,b}{\operatorname{argmin}} \|H - W * \alpha - b\|_2 + \beta' \|W\|_2 \quad (7)$$

where W, b are parameters for a linear classifier $H = W * \alpha + b$. Each column of H is a vector: $h_i = [0, 0, \ldots, 1 \ldots, 0, 0]$, where the position of non-zero element indicates the class. So $\|H - W * \alpha + b\|_2$ is the classification error and $\|W\|_2$ is the regularization penalty term. We can set b to zero for simplicity.

Considering Eqn. 5 and Eqn. 7 at the same time, we can define the following problem for learning a dictionary with both discriminative power and representative power:

$$\begin{aligned} &< D, W, \alpha >= \underset{D,W,\alpha}{\operatorname{argmin}} \\ &\|Y - D * \alpha\|_2 + \gamma * \|H - W * \alpha\|_2 + \beta * \|W\|_2 \\ &subject\ to \|\alpha\|_0 \leq T \end{aligned} \quad (8)$$

where Y is the set of input signals, D the dictionary, α the coefficient, H the label of the training images, W the parameter of the classifier, and γ and β are scalars controlling the relative contribution of the corresponding terms.

The above formulation may be viewed as a special case of [18] without considering the unlabeled data therein. However, our emphasis is on viewing the formulation of Eqn. 8 as an extended K-SVD problem and thus the solution (to be presented in subsequent subsections) will be solved by a K-SVD-like algorithm. This is in contrast with the the sophisticated (and computationally involving) optimization procedures used in [14], [15], and [18]. To better illustrated this point, we describe the following iterative procedure for solving the problem of Eqn. 8 (which we will refer to as the Baseline Algorithm later):

1. Initialize D and α with K-SVD by Eqn. 5;
2. Calculate W in Eqn. 7 when D and α are fixed;
3. Calculate α when D and W are fixed;
4. Calculate D when α and W are fixed;
5. Iterate Steps 2 to 4 until some criterion are met.

Essentially, the above procedure is effectively the algorithm of [18], except that here we only consider labeled data. Hence this Baseline Algorithm will be used in our comparison of the proposed method with that of [18].

3.2. A New Algorithm: Discriminative K-SVD

The Baseline Algorithm described above can only find an approximate solution to the problem of Eqn. 8, since in each step of the method, it only finds the solution for a sub-problem of Eqn. 8. While practically speaking, the final solution may converge to the real solution, the method has big potential of getting stuck at local minima of the sub-problems. Additionally, as is obvious from the Baseline Algorithm, in each iteration there are three optimization problems involved and thus the convergence, if it happens, will be slow to reach. To get around these issues, and to leverage the proven performance of the K-SVD algorithm, we propose the following Discriminative K-SVD (D-KSVD) algorithm, which uses K-SVD to find the globally optimal solution for all the parameters simultaneously. The task is formulated as solving the following problem

$$\begin{aligned} &< D, W, \alpha >= \underset{D,W,\alpha}{\operatorname{argmin}} \\ &\|\begin{pmatrix} Y \\ \sqrt{\gamma} * H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\gamma} * W \end{pmatrix} * \alpha\|_2 + \beta * \|W\|_2 \\ &subject\ to\ \|\alpha\|_0 \leq T \end{aligned} \quad (9)$$

We adopt the protocol in the original K-SVD algorithm: the matrix $\begin{pmatrix} D \\ \sqrt{\gamma} * W \end{pmatrix}$ is always normalized column-wise. Therefore, we can further drop the regularization penalty term $\|W\|_2$, and thus the final formulation of the problem can be written as:

$$\begin{aligned} &< D, W, \alpha >= \underset{D,W,\alpha}{\operatorname{argmin}} \\ &\|\begin{pmatrix} Y \\ \sqrt{\gamma} * H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\gamma} * W \end{pmatrix} * \alpha\|_2 \\ &subject\ to\ \|\alpha\|_0 \leq T \end{aligned} \quad (10)$$

Now, the problem of Eqn. 10 can be efficiently solved by updating the dictionary atom by atom with the following method: For each atom d_k and the corresponding coefficient α_k, we solve the following problem

$$< d_k, \alpha_k >= \underset{d_k,\alpha_k}{\operatorname{argmin}} \|E_k - d_k * \alpha_k\|_F \tag{11}$$

where $E_k = Y - \sum_{i \neq k} d_i * \alpha_i$ and Y is the training data. $\|\|_F$ denotes the Frobenius norm. This is essentially the same problem that K-SVD has solved and thus the the solution to Eqn. 11 is given by

$$\begin{aligned} U * \Sigma * V &= SVD(E_k) \\ \tilde{d_k} &= U(:,1) \\ \tilde{\alpha_k} &= \Sigma(1,1) * V(1,:) \end{aligned} \tag{12}$$

where $U(:,1)$ denotes the first column of U and $V(1,:)$ for the first row of V.

3.3. The Algorithm for Classification

Upon the completion of training with the labeled data in the previous D-KSVD algorithm, we obtain an learned dictionary D and a classifier W. However, the dictionary D does not readily support a sparse-coding based representation of a new test image, since D and W are normalized jointly in the previous learning algorithm, i.e,

$$\|\begin{pmatrix} d_i \\ \sqrt{\gamma} * w_i \end{pmatrix}\|_2 = 1 \tag{13}$$

Note that we cannot simply re-normalize D column-wise by itself, since in the training stage W is obtained with the original, un-normalized D. Hence, we need to figure out a way of obtaining a valid (normalized) dictionary and the corresponding classifier, based on the learning results, D and W. To this end, we prove the following lemma which establishes the relationship between the desired (D', W') and the learned (D, W).

Lemma: The normalized dictionary D' and the corresponding classifier W' can be computed as

$$\begin{aligned} D' &= \{d'_1, d'_2, \ldots, d'_k\} \\ &= \{\frac{d_1}{\|d_1\|_2}, \frac{d_2}{\|d_2\|_2}, \ldots, \frac{d_k}{\|d_k\|_2}\} \\ W' &= \{w'_1, w'_2, \ldots, w'_k\} \\ &= \{\frac{w_1}{\|d_1\|_2}, \frac{w_2}{\|d_2\|_2}, \ldots, \frac{w_k}{\|d_k\|_2}\} \end{aligned} \tag{14}$$

where d_i and w_i denote the i-th column of D and W, respectively.

Proof: If y is a vectorized image, then

$$\begin{aligned} y &= D * \alpha = \sum_i \alpha_i * d_i \\ &= \sum_i \alpha_i * \|d_i\|_2 * \frac{d_i}{\|d_i\|_2} \\ &= \sum_i \alpha'_i * d'_i = D' * \alpha' \\ label &= W * \alpha = \sum_i \alpha_i * d_i \\ &= \sum_i \alpha_i * \|d_i\|_2 * \frac{w_i}{\|d_i\|_2} \\ &= \sum_i \alpha'_i * w'_i = W' * \alpha' \end{aligned} \tag{15}$$

where $d'_i = \frac{d_i}{\|d_i\|_2}$ and $w'_i = \frac{w_i}{\|d_i\|_2}$ are the i-th column of D and W respectively.

With the normalized D', we can find the sparse coefficients for a given test image y by solving the following problem

$$< \alpha' >= \underset{\alpha'}{\operatorname{argmin}} \|y - D' * \alpha'\|_2 + \sigma * \|\alpha'\|_0 \tag{16}$$

This is the typical sparse-coding problem and in practice we often resort to the following convex optimization problem

$$< \alpha' >= \underset{\alpha'}{\operatorname{argmin}} \|y - D' * \alpha'\|_2 + \sigma * \|\alpha'\|_1 \tag{17}$$

which can be solved by many L1 optimization methods, such as GPSR [11], L1 magic [6] and so on. The stability of the solution depends on the incoherence of D' and sparsity of α' [20]. When α' is sparse and D' is sufficiently incoherent, Orthonormal Matching Pursuit [8] can also find the sparse coefficient [11]. According to our experiments with large face databases, OMP works well and run faster than other L1-optimization methods mentioned above. Thus the results reported in this paper are based on the OMP method.

The final classification of a test image is based on its sparse coefficient α', which carries most discriminative information. We can simply apply the linear classifier W' to α' and obtain the label of the image:

$$l = W' * \alpha' \tag{18}$$

where l is a vector.

Note that the coefficient α' can be viewed as the weight of each atom in reconstructing the test image. Thus we can view each column w'_k of W' as a factor for measuring the similarity of atom d'_k to each class. Therefore, $l = W' * \alpha'$ is the weighted similarity of the test image y to each class. In this sense, the label of test image y is decided by the index i where l_i is the largest among all elements of the l

computed in Eqn. 18. Obviously, in the ideal case, l will be of the form $l = \{0, 0, \ldots, 1, \ldots, 0, 0\}$ (i.e, with only one non-zero entry, which equals to 1).

4. Experiments and Analysis

In this section, we first use a simulation experiment (still based real face images) to compare the proposed D-KSVD method with the method of [18]. (As there is no code publicly available for the method in [18], our comparison is based on our implementation of the Baseline Algorithm discussed in Sect. 3.1, which is essentially the same as that of [18].) Then we evaluate our method on two commonly-used face databases: the Extended YaleB database and the AR database. For comparison purpose, we also implemented the SRC algorithm. To gain more insights into how the proposed D-KSVD method may gain over a plain K-SVD technique, we also implemented an algorithm that directly uses the dictionary learned by the original K-SVD algorithm for face recognition. The training stage of this algorithm runs as follows:

1. Train D with K-SVD according to Eqn. 5;
2. Train W with equation:

$$W = (\alpha^T \alpha + \beta' * I)^{-1} * \alpha * H^T \tag{19}$$

In this algorithm, D and W are trained independently. The test stage is done similarly to what described in Sect. 3.3. For simplicity, we will refer to this method simply as K-SVD thereafter.

All the experiments were run on Matlab 2008a. The PC we used has an Intel P4 2.8GHz CPU and 1 GB RAM.

4.1. Simulation Experiments

We used 52 images from 2 random persons in the AR database (26 images each person) for this simulation. These images contain all the possible conditions in the AR database: varying expressions, varying illumination, and different occlusion conditions. We used the same parameters in running the two competing methods: the proposed and the Baseline Algorithm (or the method of [18]).

First, we compare the methods based on the Fisher criterion, which is commonly used to evaluate the performance of classifiers. Fisher criterion measures the ratio of between-class variance and in-class variance. A bigger value usually means a better classification result. For a two-class problem, the Fisher criterion can be computed as follows:

$$S = \frac{(\mu_1 - \mu_2)^2}{\frac{1}{C_1} * \sum_{i=1}^{C_1} (x_1(i) - \mu_1)^2 + \frac{1}{C_2} * \sum_{i=1}^{C_2} (x_2(i) - \mu_2)^2} \tag{20}$$

Table 1. The result for fisher criterion in simulation experiments.

Method	D-KSVD	Baseline
Fisher Criterion	1.2431	1.0924

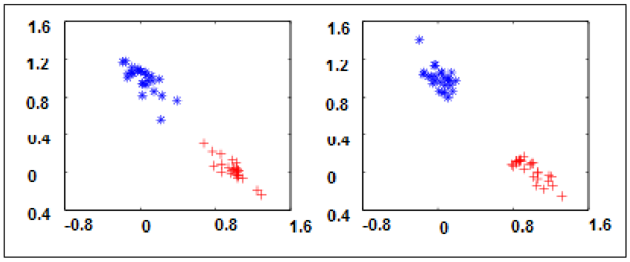

Figure 1. Visualizing the computed l. The plot on the left is from the Baseline Algorithm (or [18]). The plot on the right is from the proposed D-KSVD method.

Table 2. The maximal pair-wised correlation for dictionary learned by D-KSVD and Baseline in simulation experiment.

Method	D-KSVD	Baseline
Max R value	0.7633	0.7830

where, μ is the mean of the data, the subscripts are the class labels, and x is the data, which is the l computed in Eqn. 5 in this analysis (since we wanted to see how well the l's computed by the two methods are). We have visualized l for all 52 images in Fig. 1. The computed Fisher criteria of the two methods are listed in Table 1. It shows that our method get a bigger value for fisher criterion, which means that our method can do better classification than the method in [18] does.

Second, we measure the incoherence of the dictionary which is critical for sparse representation. Y. Sharon *et al.* [19] proposed Equivalence Breakdown Point (EBP) for measuring the incoherence of the dictionary. However, computing EBP is computationally prohibitive for large dictionaries (e.g., of the size 600*500 as in our experiments). Thus we used the correlation coefficients from pairs of the atoms in the dictionary instead. This is calculated as

$$R(x, y) = \frac{cov(x, y)}{\sqrt{cov(x, x) * cov(y, y)}} \tag{21}$$

where x and y are two atoms in the dictionary, and cov computes the covariance. A smaller coefficient R between two atoms means that they are more incoherent. Ideally, we want to have a small R for all possible pairs from the learned dictionary. We computed the largest R from all pairs in the two dictionaries learned from the Baseline Algorithm and the proposed D-KSVD algorithm, as reported in Table 2, which shows that the proposed method learns a better dictionary.

4.2. Results with the Extended YaleB Database

The Extended YaleB database contains about 2414 frontal face images of 38 individuals. Following [22], we used the cropped and normalized face images of 192*168 pixels, which were taken under varying illumination conditions[13]. We randomly split the database into two halves. One half, which contains 32 images for each person, was used for training the dictionary. The other half was used for testing. Further, we projected the face image $\in R^{192*168}$ into a vector $\in R^{504}$ with a randomly generated matrix, which is called Randomface [22]. The learned dictionary contains 304 atoms, which corresponds to, on average, roughly 8 atoms for each person (but we must point out that, unlike in the SRC algorithm, in our method there is no explicit correspondence between the atoms and the labels of the people, since all the information is encapsulated into the discriminative dictionary and the corresponding classifier). The sparsity prior assumed in the learning was set to $T = 16$.

With this database, we tested 4 methods: SRC, K-SVD (as defined earlier in the beginning of Section 4.1), the Baseline Algorithm (or equivalently the method of [18]), and the proposed D-KSVD method. The best result reported for SRC is 98.26% when there are 32 images per person in the dictionary. We also tested the performance of SRC when the dictionary is smaller (8 atoms per people). This set of results are denoted by $SRC\dagger$ in the subsequent tables. In [18], the authors only used a few images (at most 4) per person for training and the recognition result was very poor (about 66.4%). For a fair comparison, we tested the Baseline Algorithm (essentially the method of [18]) with more training images. In short, the key learning parameters used in the four methods were kept to be the same in our experiments.

All the results are summarized in Table 3. In the experiments, the scalar β and γ were set to 1. From the experiments, we found that most of the failure cases (about 46 out of 54) are from images under extreme illumination conditions. Some examples of these cases are given in Fig. 2. Thus, we performed another set of experiments with these "bad" images excluded (13 for each person). This was intended to show the true performance of the competing methods without the interference of images of extremely bad quality. The results of this new round experiments are listed in the last row of the table. From the table, it is clear that the proposed D-KSVD method always obtains better results than the K-SVD method and the Baseline (or the method of [18]); In addition, for dictionaries of the same size, our method performs better than the SRC method.

Table 3. The performance of the algorithms (recognition rate in %) for the Extended YaleB database. The 2nd row is the result when we used all 64 images for each people. The 3rd row is the result when we excluded 13 poor-quality images for each person. The images for training the dictionary were randomly selected.

D-KSVD	SRC	SRC†	K-SVD	Baseline
95.56	99.31	80.76	93.17	93.17
99.58	99.72	93.85	99.30	98.89

Table 4. The time for classifying one test image using the SRC method and the D-KSVD method on the extended YaleB database. We record the time for all the test images and then divide it by the number of images. The value is the average over 4 rounds. The unit is millisecond. The 2nd row is the result when we use all 64 images for each people. The 3rd row is the result when we use 51 images for each people.

Method	D-KSVD	SRC	SRC†
Case 1	84	120	83
Case 2	78	121	82

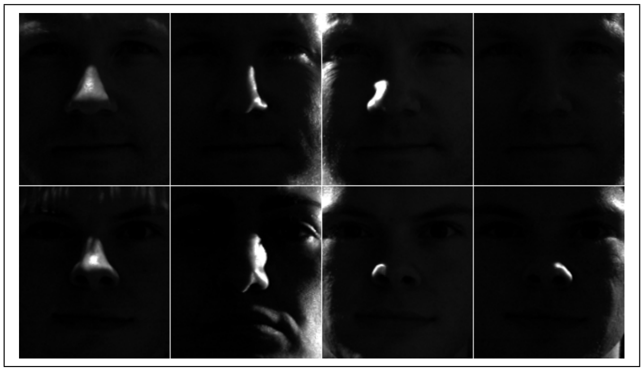

Figure 2. Sample images under extreme illumination conditions. The left two are from one person and the right two from another person.

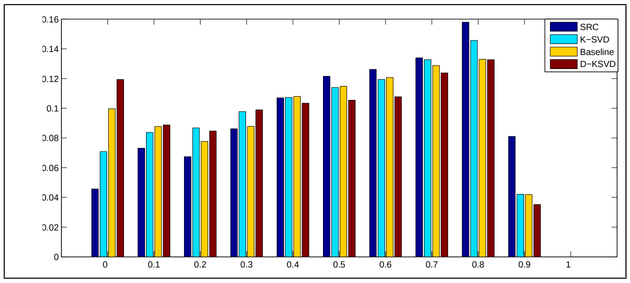

Figure 3. The histogram of pairwise correlation coefficients of the atoms in the learned dictionary. The dictionaries were trained by 4 different methods with the extended YaleB database.

We also evaluated the incoherence of the learned dictionaries in the experiments by calculating the correlation coefficient for each pair of atoms in the dictionary. Since the experiments involve multiple classes, the correlation of the atoms may exhibit more complex patterns. To avoid the situation that a big R from a single pair of atoms overshadows the correlation of all other pairs, in this case, we

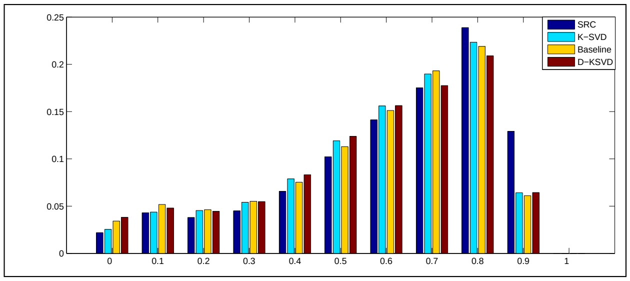

Figure 4. The histogram of pairwise correlation coefficients of the atoms in learned dictionary. The dictionaries were trained by 4 different methods with the extended YaleB database with 13 poor-quality images excluded.

plot the histogram of the correlation coefficients. The results are given in Fig. 3 and Fig. 4 respectively. From these plots, it was found that the proposed D-KSVD method was able to generate a dictionary that contains more less-correlated atom pairs. That is, in the plots, the bars from the proposed method are on average slightly taller towards the left side of the axis of the correlation coefficients. (Probably one cannot expect to see dramatic difference in these plots, given that the performance of the algorithms are already very close and the improvement is at most a couple of percents. However, these few last percents are the hardest to obtain.)

In addition to the classification performance of the D-KSVD method and the SRC method, we also compared their speed performance for classifying one test image. We recorded the total time for classifying all the test images, then divided it by the number of the test images, hence obtaining the average processing time for each testing image. We ran this for 4 rounds and calculated the average result, as shown in Table 4. From the results in Table 4, we can see that, with a smaller dictionary (304 atoms in the dictionary for D-KSVD and SRC†, and 1216 atoms in the dictionary for SRC), we can save about 1/3 of the time in testing. With a database involving more people, we can expect a smaller dictionary can save even more time (see the results below for the AR database too).

4.3. Results with the AR database

The AR database contains over 4000 color images for 126 people. For each person, there are 26 images taken in two different sections. These images contains 3 different illumination conditions, 3 different expressions and 2 different facial disguises (with sunglasses and scarf respectively). Thus this is a more challenging dataset. In our experiments, we used 2600 images from 50 male and 50 female. For each person, we randomly selected 20 images for training and the other 6 for testing, which generally contain all the possible variations in the database. The results reported in the subsequent table are from the average of three such random spits of the training and testing images. The learned dictionary contains 500 atoms, i.e., roughly 5 atoms per person (but again, as discussed earlier, in our method there is no explicit correspondence between the atoms and the people). The sparsity prior was set to $T = 10$.

Table 5. The result reported for SRC. SRC‡ means first breaking the images into blocks, then classifying each block, finally using voting policy to decide image's label.

Test images	Without disguise	Sunglasses	Scarves
SRC	94.7%	87.0%	59.5%
SRC‡	NA	97.5%	93.5%

Table 6. The performance of the algorithms (recognition rate in %) for the AR database. SRC_n means there are n atoms per person in the dictionary learned by SRC. All other 3 methods use 500 atoms (roughly 5 per person). The images for training the dictionary were randomly selected.

D-KSVD	SRC_{20}	SRC_5	K-SVD	Baseline
95.0	90.50	68.14	88.17	93.0

Table 7. The time for classifying one test image using the SRC method and the D-KSVD method on AR database. We record the time for all the test images and then divide it by the number of images. The value is the average over 4 rounds. The unit is millisecond.

Method	D-KSVD	SRC_{20}	SRC_5
Result	62	131	76

For direct comparison, we quoted the performance of the SRC algorithm on this dataset from [22], as listed in Table 5. It is worth noting that, in their experiments, they manually selected 7 images without facial disguise for each people from first section to build the dictionary. In our experiments, we tested the SRC algorithm with randomly-selected images for building the dictionary and experimented with two different dictionary size. We also tested K-SVD and the method of [18] (the Baseline) on the AR database. In the experiments, all four methods used the same parameters. We also did the random projection as described earlier and in this case, this was from face images $\in R^{165*120}$ into vectors $\in R^{540}$.

The final results from all the methods are listed in Table 6. The experiments show that the performance of the SRC algorithm degraded dramatically when the training of the dictionary was based on randomly-selected images: when there are 5 images per person in the dictionary, the result is merely 68.14%. From the table, the proposed method outperforms all the competing methods.

As in the experiments with the Extended YaleB database, we also compared the speed performance of the D-KSVD method and the SRC method for classifying one test image on the AR database. The same method was used here. The result is shown in Table 7. For SRC_5 and D-KSVD, the dictionary has 500 atoms, and the size is 2000 atoms for SRC_{20}. As expected, for a database involving more people, a smaller dictionary can save more time, which is about 1/2 from the table.

5. Conclusion

We proposed a dictionary-learning approach, Discriminative K-SVD (D-KSVD), for face recognition. By adding a discriminative term into the objective function of the original K-SVD algorithm, we can ensure that the learned overcomplete dictionary is both representative and discriminative. The solution of the new formulation follows a procedure derived from the original K-SVD algorithm and thus can be efficiently solved. Unlike existing approaches that iteratively solve sub-problems in order to approximate a global solution, our method directly finds all the parameters (the dictionary and the classifier) simultaneously. With experiments on two large, commonly-used face databases, we demonstrated the advantages of the proposed method. The experimental results shows that: under the same learning condition, our method always outperforms K-SVD and the method of [18]; with the same dictionary size or with randomly chosen training images, our method outperforms the SRC algorithm. Our future work includes exploring both theoretically and empirically the structure of the learned dictionaries from our method and the competing methods, so as to reveal deeper insights on how to incorporate label information into dictionary learning. More extensive analysis on the speed performance of the algorithms is also another direction of interest and of practical importance.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006. 1, 2

[2] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 218–233, 2003. 2

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linearprojection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997. 1, 2

[4] O. Bryt and M. Elad. Compression of facial images using the K-SVD algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008. 1

[5] O. Bryt and M. Elad. Improving the k-svd facial image compression using a linear deblocking method. In *IEEE 25th Convention of Electrical and Electronics Engineers in Israel, 2008. IEEEI 2008*, pages 533–537, 2008. 1

[6] E. Candes and J. Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*. 4

[7] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006. 2

[8] S. Chen, C. Cowan, P. Grant, et al. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on neural networks*, 2(2):302–309, 1991. 4

[9] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2

[10] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 1

[11] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal on selected topics in Signal Processing*, 1(4):586–597, 2007. 4

[12] A. Georghiades, P. Belhumeur, and D. Kriegman. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. 1

[13] K. Lee, J. Ho, and D. Kriegman. Acquiring Linear Subspaces for Face Recognition under Variable Lighting . *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005. 2, 6

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008. 1, 3

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *Adv. NIPS*, 21, 2009. 1, 3

[16] J. Mairal, M. Elad, G. Sapiro, et al. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53, 2008. 1

[17] A. Martinez and R. Benavente. The AR face database. *Univ. Purdue, CVC Tech. Rep*, 24, 1998. 2

[18] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008. 1, 2, 3, 5, 6, 7, 8

[19] Y. Sharon, J. Wright, and Y. Ma. Computation and relaxation of conditions for equivalence between l1 and l0 minimization. *submitted to IEEE Transactions on Information Theory*, 2007. 5

[20] J. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006. 4

[21] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. 1

[22] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 210–227, 2009. 1, 2, 6, 7

[23] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003. 1