Hindawi
Journal of Electrical and Computer Engineering
Volume 2018, Article ID 2179049, 11 pages
https://doi.org/10.1155/2018/2179049

Research Article

Laplace Graph Embedding Class Specific Dictionary Learning for Face Recognition

Li Wang, Yan-Jiang Wang, and Bao-Di Liu

College of Information and Control Engineering, China University of Petroleum (East China), Qingdao, China

Correspondence should be addressed to Yan-Jiang Wang; yjwang@upc.edu.cn

Received 23 September 2017; Revised 2 December 2017; Accepted 6 December 2017; Published 7 February 2018

Academic Editor: Tongliang Liu

The sparse representation based classification (SRC) method and collaborative representation based classification (CRC) method have attracted more and more attention in recent years due to their promising results and robustness. However, both SRC and CRC algorithms directly use the training samples as the dictionary, which leads to a large fitting error. In this paper, we propose the Laplace graph embedding class specific dictionary learning (LGECSDL) algorithm, which trains a weight matrix and embeds a Laplace graph to reconstruct the dictionary. Firstly, it can increase the dimension of the dictionary matrix, which can be used to classify the small sample database. Secondly, it gives different dictionary atoms with different weights to improve classification accuracy. Additionally, in each class dictionary training process, the LGECSDL algorithm introduces the Laplace graph embedding method to the objective function in order to keep the local structure of each class, and the proposed method is capable of improving the performance of face recognition according to the class specific dictionary learning and Laplace graph embedding regularizer. Moreover, we also extend the proposed method to an arbitrary kernel space. Extensive experimental results on several face recognition benchmark databases demonstrate the superior performance of our proposed algorithm.

1. Introduction

The sparse representation algorithm based on dictionary learning (dictionary learning for sparse representation) is attracting more and more attention in computer vision due to its impressive performance in many applications, such as image processing, image ranking [1], human activity recognition [2], and image classification [3, 4]. Different from the traditional subspace methods, such as PCA, the sparse representation algorithm allows the bases of a dictionary to be much larger than the dimension of the sample characteristics, so the sample can be fitted more effectively.

We know that deep learning-based methods are currently the mainstream methods in image classification. Taigman et al. [5] proposed a DeepFace neural network for face recognition, which has achieved human-level performance. Ding and Tao [6] proposed a comprehensive framework based on Convolutional Neural Networks to overcome challenges in video-based face recognition. Florian Schroff et al. [7] proposed a FaceNet which can learn the mapping from face images to a compact Euclidean space. Sun et al. [8] proposed a DeepID2+ convolutional network which increases the dimension of hidden representations and adds supervision to early convolutional layers. Liu et al. [9] proposed a multipatch deep CNN and deep metric learning method to extract discriminative features for face recognition. However, the depth learning method performed well when the sample size was large, and the effect was not satisfactory under the condition of a small database. Therefore, we propose a dictionary learning method based on Laplacian embedding and sparse representation, which can still achieve good results in the case of very small samples.

The sparse representation based classifier has been widely used in the field of face recognition. Normally, classifying the samples involves two stages: first, obtain the sample feature, and then the sample feature can be sent to the classifier for classification. In the process of feature extraction, many subspace methods are proposed. The principal component analysis method was proposed to reduce a complex data set to a lower dimension to reveal the hidden, simplified data structures [10]. The linear discriminant analysis (LDA) algorithm was proposed to find the projection hyperplane

that minimizes the interclass variance and maximizes the distance between the projected means of the classes [11]. Tao et al. [12] proposed a general tensor discriminant analysis method as a preprocessing step for the LDA algorithm to reduce the undersampling problem. The locality preserving projection algorithm was proposed to preserve the neighborhood structure of the data [13]. In the procedure of classification, Liu et al. [14] proposed a new belief-based k-nearest neighbor classifier to make the classification result more robust to misclassification errors. Noh et al. [15] proposed a nearest neighbor algorithm to enhance the performance of the nearest neighbor classification by learning a local metric. Although the k-nearest neighbor classifier and the nearest neighbor classifier have achieved good results on some data sets, they did not select the most discriminatory feature of the sample to classify. So, subspace-based classifier design methods were proposed to improve the classification effect.

The sparse representation based classification algorithm uses training samples to construct an overcomplete dictionary, and the test samples can be well represented as a sparse linear combination of elements from the dictionary [16]. But the subsequent research shows that sparseness cannot extract the most discriminatory features of the samples. Collaborative representation based classification (CRC) was proposed, which uses the L2-norm constraint to reveal the internal structure of the testing sample [17]. Although the SRC and CRC methods have achieved superior performance in visual recognition, both SRC and CRC algorithms directly use the training samples as the dictionary matrix. The direct use of training samples to build dictionaries can lead to two drawbacks: first, very few samples to build an overcomplete dictionary, which may result in low classification accuracy, and second, very redundant dictionary samples, which prevent the original signals from being effectively expressed, resulting in poor classifier performance.

So, dictionary learning methods are proposed to improve the classification effect. Discriminative dictionary learning approaches can be divided into three types: shared dictionary learning, class specific dictionary learning, and hybrid dictionary learning. The shared dictionary learning method usually uses all training samples to obtain a classification dictionary. Lu et al. [18] proposed a locality weighted sparse representation based classification (WSRC) method which utilizes both data locality and linearity to train a classification dictionary. Yang et al. [19] proposed a novel dictionary learning method based on the Fisher discrimination criterion to improve the pattern classification performance. Yang et al. [20] proposed a latent dictionary learning method to learn a discriminative dictionary and build its relationship to class labels adaptively. Jiang et al. [21] proposed an algorithm to learn a single overcomplete dictionary and an optimal linear classifier for face recognition. Zhou et al. [22] presented a dictionary learning algorithm to exploit the visual correlation within a group of visually similar object categories for dictionary learning where a commonly shared dictionary and multiple category-specific dictionaries are accordingly modeled. The class specific dictionary learning method trained a dictionary for each class of samples. Sun et al. [23] learned a class specific subdictionary for each class and a common subdictionary shared by all classes to improve the classification performance. Wang and Kong [24] proposed a method to explicitly learn a class specific dictionary for each category, which captures the most discriminative features of this category, and simultaneously learn a common pattern pool, whose atoms are shared by all the categories and only contribute to representation of the data rather than discrimination.

The hybrid dictionary learning method is the combination of the above two methods. Rodriguez and Sapiro [25] proposed a new dictionary learning method which uses a class-dependent supervised constraint and orthogonal constraint; this method learns the intraclass structure while increasing the interclass discrimination and expands the difference between classes. Gao et al. [26] learned a category-specific dictionary for each category and a shared dictionary for all the categories, and this method improves conventional basic-level object categorization. Liu et al. [27] proposed a locality sensitive dictionary learning algorithm with global consistency and smoothness constraint to overcome the restriction of linearity at a relatively low cost. Although the hybrid dictionary learning method achieved good results, these methods usually operate in the original Euclidean space, which cannot capture nonlinear structures hidden in data. So, many kernel-based classifiers are designed to solve this problem. Nguyen et al. [28] presented a dictionary learning method for sparse representation based on the kernel method. Liu et al. [29] proposed a multiple-view self-explanatory sparse representation dictionary learning algorithm (MSSR) to capture and combine various salient regions and structures from different kernel spaces, and this method achieved superior performance in the field of face recognition.

As better effects had been achieved by MSSR algorithm, this algorithm neither took into consideration the details of training samples in the original sample space nor protected this powerful information conducive to classification in the dictionary space. Therefore, in this algorithm, the Laplace constraint is added to the objective function to make the closely similar samples in low dimensional space also very close in the high dimensional dictionary space.

Motivated by this, we proposed a Laplace graph embedding class specific dictionary learning algorithm and extended this method to arbitrary kernel space. The main contribution is listed in four aspects. (1) We propose a Laplace embedding sparse representation algorithm. It combines the advantages of SRC's discriminant ability and maintains the intrinsic local geometric feature of the sample features by Laplace embedding. (2) We propose a Laplace embedding constraint dictionary learning algorithm to construct superior subspace and reduce the residual error. (3) We extend this algorithm to arbitrary kernel space to find the nonlinear structure of face images. (4) Experimental results on several benchmark databases demonstrate the superior performance of our proposed algorithm.

The rest of the paper is organized as follows. Section 2 overviews the three classical face recognition algorithms. Section 3 proposes our Laplace graph embedding class specific dictionary learning algorithm with kernels. The solution to the minimization of the objective function is elaborated in

Section 4. Then, experimental results and analysis are shown in Section 5. Finally, discussions and conclusions are drawn in Section 6.

2. Overview of SRC and CRC

In this section, we will briefly overview two classical face recognition algorithms, SRC and CRC.

Suppose that there are C classes in the training samples and each class has N_c elements. $N = \sum_{c=1}^{C} N_c$, where N represents the total number of training samples; X represents all the training samples, $X = [X^1, X^2, \ldots, X^c, \ldots, X^N]$, where $X^c \in R^{D \times N_c}$; D represents the dimension of the sample features; X^c represents the cth class of the training samples. Supposing that y is a test sample and $y \in R^{D \times 1}$, the sparse representation of sample y can be expressed as

$$\widehat{S} = \arg\min_{s} \left\{ \left\| y - X^c s \right\|_2^2 + 2\alpha \left\| s \right\|_1 \right\}, \tag{1}$$

where s is the sparse coding of sample y in the cth dictionary and α is the regularization parameter in formula (1), which is used to control the sparsity and accuracy of the expression.

The collaborative representation based classification algorithm applies L2-norm constraint on the object function; the objective of the CRC algorithm can be rewritten as follows:

$$\widehat{S} = \arg\min_{s} \left\{ \left\| y - X^c s \right\|_2^2 + 2\alpha \left\| s \right\|_2^2 \right\}, \tag{2}$$

where β is the regularization parameter to control the expression accuracy of the object function.

Both SRC and CRC methods directly use the training samples as the dictionary. And each base in the dictionary has the same contribution to the sample expression. The testing sample y can be encoded as

$$y \approx X^c s. \tag{3}$$

Here, X^c is the dictionary matrix composed of the cth class training samples, and s is the sparse coding of y.

Directly using the training samples as the dictionary leads to high residual error. Liu et al. [29] proposed a single-view self-explanatory sparse representation dictionary learning algorithm (SSSR). Supposing that c represents the class number of the training samples and X^c means the collection of sample characteristics of class c, the objective function of the SSSR method can be formulated as

$$\begin{aligned} \min_{W^c, S^c} \quad & f\left(W^c, S^c\right) = \left\| X^c - X^c W^c S^c \right\|_F^2 + 2\alpha \sum_{i=1}^{N_c} \left\| \left(S_i^c\right) \right\|_1 \\ \text{s.t.} \quad & \left\| X^c \left(W^c\right)_k \right\| \le 1, \quad \forall k = 1, 2, \ldots, K, \end{aligned} \tag{4}$$

where S^c is the sparse codes of the cth class and S_i^c represents the ith column of S^c. The SSSR algorithm reconstructed the dictionary matrix, W^c is the dictionary weight matrix, $W^c \in R^{N_c \times K}$, and N_c is the number of the cth classes. W^c expands the original dictionary space into a more complete dictionary space; when the identity matrix $W^c \in I^{N_c \times N_c}$ appears, the class specific dictionary learning algorithm evolves into the SRC method. The existence of W^c matrix makes dictionary learning more flexible in the process of expression, and the reconstruction error may be reduced as well.

Meanwhile, Liu et al. [29] extended the SSSR algorithm into kernel spaces, which can map the original sample features into a high dimensional nonlinear space for better mining of nonlinear relationships between samples. The objective function of the multiple-view kernel-based class specific dictionary learning algorithm (KCSDL) is shown as follows:

$$\begin{aligned} \min_{W^c, S^c} \quad & f\left(W^c, S^c\right) \\ & = \left\| \phi\left(X^c\right) - \phi\left(X^c\right) W^c S^c \right\|_F^2 + 2\alpha \sum_{i=1}^{N_c} \left\| \left(S_i^c\right) \right\|_1 \\ \text{s.t.} \quad & \left\| \phi\left(X^c\right) \left(W^c\right)_k \right\| \le 1, \quad \forall k = 1, 2, \ldots, K, \end{aligned} \tag{5}$$

where $\phi : R^D \to R^t$ means the kernel function; it maps the original feature space into a high dimensional kernel space.

3. Our Proposed Approach

Although the above methods have achieved good results in the field of face recognition, there are still some deficiencies. The SSSR algorithm uses a reconstructed dictionary matrix to make sparse representation on samples; however, it does not take into account the fact that only the sparsity constraint on the target is not necessary to gain results for better classification.

Motivated by this, we have proposed the sparse representation algorithm based on Laplace graph embedding, while taking into account the sparse representation on the samples; this algorithm mines the details implicit in the training samples; therefore, the same sample is more concentrated in the sparse expression space, so as to reduce the fitting error and improve the classification effect.

The objective function of our proposed sparse representation algorithm based on Laplace graph embedding now becomes

$$\begin{aligned} & f\left(S^c\right) \\ & = \left\| \phi\left(X^c\right) - \phi\left(X^c\right) W^c S^c \right\|_F^2 + 2\alpha \sum_{n=1}^{N_c} \left\| S_n^c \right\|_1 \\ & \quad + \beta \sum_{i \ne j} \left\| S_i^c - S_j^c \right\| p_{ij} \\ \text{s.t.} \quad & \left|\left| \phi\left(X^c\right) \left(W^c\right)_k \right|\right| \le 1, \quad \forall k = 1, 2, \ldots, K, \end{aligned} \tag{6}$$

where X^c means class c training samples, W^c means dictionary weight matrix, S^c means the dictionary representation of class c samples, and S_n^c represents the nth column of S^c.

4. Optimization of the Objective Function

In this section, we focus on solving the optimization problem for the proposed Laplace graph embedding class specific

dictionary learning algorithm. The dictionary weight matrix W^c and sparse representation matrix S^c can be optimized by iterative approaches.

When each element in the W^c matrix is updated, the remaining elements in W^c matrix and S^c matrix are fixed; at this time, the objective function is changed into an L2-norm constrained least-squares minimization subproblem. Similarly, when each element in S^c matrix is updated, W^c matrix and the remaining elements in S^c matrix are fixed. The objective function can be seen as an L1-norm constrained least-squares minimization subproblem.

4.1. L1-Norm Regularized Minimization Subproblem. When updating the elements in S^c matrix, the nonupdated elements in S^c and W^c matrix will be fixed. Here, the objective function can be formulated as

$$f(S^c) = \left\|\phi(X^c) - \phi(X^c) W^c S^c\right\|_F^2 + 2\alpha \sum_{n=1}^{N_c} \left\|S_n^c\right\|_1 + \beta \sum_{i \neq j} \left\|S_i^c - S_j^c\right\| p_{ij}, \tag{7}$$

where p_{ij} is the weight value which describes the neighboring degree of x_i^c and x_j^c and $p_{ij} = e^{-\|x_i^c - x_j^c\|/t}$. x_i^c and x_j^c are training samples that belong to the cth class, and t is a constant which controls the range of p_{ij}. Formula (7) can be simplified as

$$\begin{aligned}
&f(S^c) \\
&= \operatorname{trace}\left\{\phi(X^c)^T \phi(X^c) - 2\phi(X^c)^T \phi(X^c) W^c S^c\right\} \\
&\quad + \operatorname{trace}\left\{(S^c)^T \left(W^{cT} \phi(X^c)^T \phi(X^c) W^c\right) S^c\right\} \\
&\quad + 2\alpha \sum_{n=1}^{N_c} \left\|S_n^c\right\|_1 + \beta \operatorname{trace}\left\{S^c L S^{cT}\right\} \\
&= \operatorname{trace}\left\{\kappa(X^c, X^c)\right\} - 2\sum_{n=1}^{N_c} \left[\kappa(X^c, X^c) W^c\right]_n S_n^c \\
&\quad + \sum_{n=1}^{N_c} \left(S^{cT}\right)_n \left[W^{cT} \kappa(X^c, X^c) W^c\right] S_n^c \\
&\quad + 2\alpha \sum_{n=1}^{N_c} \left\|S_n^c\right\|_1 + \beta \sum_{k=1}^{K} \left[S_k^c L S_k^{cT}\right] \\
&= \operatorname{trace}\left\{\kappa(X^c, X^c)\right\} \\
&\quad - 2\sum_{n=1}^{N_c} \sum_{k=1}^{K} \left[\kappa(X^c, X^c) W^c\right]_{nk} S_{kn}^c \\
&\quad + \sum_{n=1}^{N_c} \sum_{k=1}^{K} \left[\sum_{l=1}^{K} \left(S^{cT}\right)_{nl} \left(W^{cT} \kappa(X^c, X^c) W^c\right)_{lk}\right] S_{kn}^c \\
&\quad + 2\alpha \sum_{k=1}^{K} \sum_{n=1}^{N_c} \left|S_{kn}^c\right| \\
&\quad + \beta \sum_{k=1}^{K} \left[\sum_{n=1}^{N_c} \left(\sum_{l=1}^{N_c} S_{kl}^c L_{ln}\right) \left(S^T\right)_{nk}\right],
\end{aligned} \tag{8}$$

where $L = D - P$, $D_{ii} = \sum_j P_{ij}$, and matrix P is the weight matrix expressing the sample neighboring distance. $\kappa(X^c, X^c)$ means the kernel function of the sample, and $\kappa(X^c, X^c)$ is calculated prior to dictionary updating. $\kappa(X^c, X^c) = \phi(X^c)^T \phi(X^c)$.

In this algorithm, each element in S_c is updated sequentially; when S_{kn} is updated, the other elements in the S matrix are regarded as constants. After ignoring the constant term of formula (8), formula (8) can be simplified as

$$\begin{aligned}
f(S_{kn}^c) &= \left(W^{cT} \kappa(X^c, X^c) W^c\right)_{kk} (S_{kn}^c)^2 \\
&\quad + 2\sum_{l=1, l \neq k}^{K} \left(W^{cT} \kappa(X^c, X^c) W^c\right)_{kl} S_{ln}^c S_{kn}^c \\
&\quad - 2\left[\kappa(X^c, X^c) W^c\right]_{nk} S_{kn}^c + 2\alpha \left|S_{kn}^c\right| + \beta L_{nn} (S_{kn}^c)^2 \\
&\quad + 2\beta \sum_{l=1, l \neq n}^{N} L_{ln} S_{kl}^c S_{kn}^c = \left[\left(W^{cT} \kappa(X^c, X^c) W^c\right)_{kk}\right. \\
&\quad \left. + \beta L_{nn}\right] (S_{kn}^c)^2 - 2\left[\left(\kappa(X^c, X^c) W^c\right)_{nk}\right. \\
&\quad \left. - \sum_{l=1, l \neq k}^{K} \left(W^{cT} \kappa(X^c, X^c) W^c\right)_{kl} S_{ln}^c - \beta \sum_{l=1, l \neq n}^{N} L_{ln} S_{kl}^c\right] \\
&\quad \cdot S_{kn}^c + 2\alpha \left|S_{kn}^c\right|.
\end{aligned} \tag{9}$$

According to the solving method in [29], it is easy to obtain the solution of the minimum value of $f(S_{kn}^c)$ under the current iteration condition:

$$\begin{aligned}
S_{kn}^c &= \frac{1}{1 + \beta L_{nn}} \min\left\{\left(\kappa(X^c, X^c) W^c\right)_{nk} - \left[E \overline{S^c}^{kn}\right]_{kn}\right. \\
&\quad \left. - \beta \left[\overline{S^c}^{kn} L\right]_{kn}, -\alpha\right\} + \frac{1}{1 + \beta L_{nn}} \\
&\quad \cdot \max\left\{\left(\kappa(X^c, X^c) W^c\right)_{nk} - \left[E \overline{S^c}^{kn}\right]_{kn}\right. \\
&\quad \left. - \beta \left[\overline{S^c}^{kn} L\right]_{kn}, \alpha\right\},
\end{aligned} \tag{10}$$

where $E = W^{cT} \kappa(X^c, X^c) W^c$ and $\overline{S^c}^{kn} = \{S_{pq}^c, p \neq k \parallel q \neq n;\ 0, p = k, q = n\}$.

4.2. ℓ_2 Norm Constrained Minimization Subproblem. When updating the dictionary matrix W^c, S^c and the nonupdated elements in W^c matrix will be fixed. The objective function can be transformed into the following form:

$$\begin{aligned}
&f(W^c) = \left\|\phi(X^c) - \phi(X^c) W^c S^c\right\|_F^2 \\
&\text{s.t.} \quad \left\|\phi(X^c) W_k^c\right\|_2^2 \leqslant 1, \quad \forall k = 1, 2, \ldots, K.
\end{aligned} \tag{11}$$

The Lagrange multiplier method is used to optimize the above problems; then, the objective function can be reduced to

$$\begin{aligned} L\left(W^c, \lambda_k\right) = & -2\sum_{k=1}^{K}\left[S^c\kappa\left(X^c, X^c\right)\right]_k W_k^c \\ & + \sum_{k=1}^{K} W_k^c\left[\kappa\left(X^c, X^c\right) W^c S^c S^{cT}\right]_k \\ & + \lambda_k\left(1 - \left[W^{cT}\kappa\left(X^c, X^c\right) W^c\right]_{kk}\right). \end{aligned} \tag{12}$$

Here, λ_k is a variable. Meanwhile, the algorithm uses Karush-Kuhn-Tucker (KKT) conditions to optimize the objective function, and Karush-Kuhn-Tucker (KKT) conditions meet the following three criteria:

$$\begin{aligned} &(1): \quad \frac{\partial L\left(W^c, \lambda_k\right)}{\partial W_k^c} = 0; \\ &(2): \quad 1 - \left[W^{cT}\kappa\left(X^c, X^c\right) W^c\right]_{kk} = 0; \\ &(3): \quad \lambda_k > 0. \end{aligned} \tag{13}$$

Hence, the solution to W_k^c becomes

$$W_k^c = \frac{S_k^c - \left[\overline{W^c}^k F\right]_k}{\sqrt{\left(S_k^{cT} - \left[\overline{W^c}^k F\right]_k\right)^T \kappa\left(X^c, X^c\right)\left(S_k^{cT} - \left[\overline{W^c}^k F\right]_k\right)}}, \tag{14}$$

where $F = S^c S^{cT}$ and $\overline{W^c}^k = \{W_p^c, p \neq k;\ 0, p = k\}$.

5. Experimental Results

In this section, we present experimental results on five benchmark databases to illustrate the effectiveness of our method. We compare the Laplace graph embedding class specific dictionary learning algorithm (LGECSDL) with some state-of-the-art methods. In the following section, we introduce the experimental environment setting, database descriptions, and experimental results. In the end, we accordingly analyze the experimental results.

5.1. Experimental Settings. In this section, we evaluate our method on five benchmark databases. The proposed LGECSDL algorithm is compared with another seven classical face recognition algorithms: nearest neighbor (NN) classification, collaborative representation based classification (CRC) [30], sparse representation based classification (SRC) [31], kernel-based probabilistic collaborative representation based classifier (ProKCRC) [32], VGG19 [33], kernel-based class specific dictionary learning (KCSDL) algorithm [29], and SVM [34].

There are two parameters in the objective function of the LGECSDL algorithm that need to be specified. α is an important parameter in the LGECSDL algorithm which is used to adjust the trade-off between the reconstruction error and the sparsity. We increase α from 2^{-12} to 2^{-1} in each experiment and find the best α in our experiments.

β is another important factor in the LGECSDL algorithm. β is used to control the trade-off between the reconstruction error and the collaborative information. We increase β from 2^{-12} to 2^{-1} and find the best β in all of our experiments.

We also evaluate the effect of different kernels for the LGECSDL algorithm. Three different kernel functions are adopted: as linear kernel ($\kappa(x, y) = x^T y$), Hellinger kernel ($\kappa(x, y) = \sum_{d=1}^{D}\sqrt{x_d y_d}$), and polynomial kernel ($\kappa(x, y) = (p + x^T y)^q$). Here, in our experiments, p and q are set to be 4 and 2, respectively.

Figure 1: Examples of the Extended YaleB database.

5.2. Database Descriptions. There are five image databases involved in our experiments. The first one is the Extended YaleB database, which contains 38 categories and 2414 frontal-face images. All the images are captured under varying illumination conditions. In our experiments, the image has been cropped and normalized to 32 × 32 pixels. Figure 1 shows several example images in the Extended YaleB database.

The second one is the AR database. The AR database contains over 3000 images of 126 individuals; images are shot under different conditions of expression, illumination, and occlusion, and each person has 26 images. Figure 3 shows some examples in the AR database.

The third database is the CMU-PIE database. The CMU-PIE database consists of 41368 pieces of pictures, which are captured under different lighting conditions, poses, and expressions. The database contains 68 individuals in total, and each person has 43 different kinds of images with 13 different poses. We selected two types of images to carry out our experiment: five near-frontal poses and all different illumination conditions. We chose 11,554 images in total for our evaluation. Each person has about 170 images. Figure 5 shows some example images in the CMU-PIE database.

We also selected the Caltech101 database to verify the LGECSDL algorithm. The Caltech101 database contains 9144 images belonging to 101 categories; each class has 31 to 800 images. We selected 5 images as training images in each class and the rest as test images. Figure 7 shows some examples in the Caltech101 database.

The fifth database is Oxford-102 flower database that contains 8,189 flower images belonging to 102 categories. Each image contains 40 to 250 images and the minimum edge

Table 1: Recognition rate on the Extended YaleB database (%).

Methods	Linear	Hellinger	Polynomial
NN	33.17 ± 1.45	NA	NA
VGG19	53.79 ± 1.21	NA	NA
SVM	65.52 ± 2.77	79.97 ± 2.57	64.89 ± 2.43
CRC	78.07 ± 2.16	87.23 ± 1.44	76.33 ± 2.17
SRC	77.59 ± 1.59	88.58 ± 1.56	74.82 ± 2.35
ProKCRC	76.42 ± 2.13	87.84 ± 1.89	74.94 ± 2.03
KCSDL	78.55 ± 2.25	88.98 ± 1.62	79.68 ± 2.41
LGECSDL	**80.18 ± 1.33**	**91.93 ± 1.31**	**81.05 ± 1.42**

length of the image is greater than 500 pixels. The Oxford-102 flower database contains pictures of flowers taken in different illumination, angle, and occlusion environments, and each kind of flower image has a high degree of similarity. In our experiments, all the images are manually cropped and resized to 224 × 224 pixels. Figure 9 shows several images in the Oxford-102 flower database.

5.3. Experiments on the Extended YaleB Database. We randomly selected 5 images as the training samples in each category and 10 images as the testing samples. In our experiments, we set the weight of the sparsity term α as 2^{-9}, 2^{-7}, and 2^{-7} for the linear kernel, Hellinger kernel, and polynomial kernel, respectively. The optimal β is 2^{-10}, 2^{-8}, and 2^{-10} for the linear kernel, Hellinger kernel, and polynomial kernel, respectively. We independently performed all the methods ten times and then reported the average recognition rates. Table 1 shows the recognition rates of all the algorithms using different kernel methods.

From Table 1, we can clearly see that LGECSDL achieves the best recognition rates of 80.18%, 91.93%, and 81.05% in the linear kernel, Hellinger kernel, and polynomial kernel space, respectively, while KCSDL, the second best method, arrives at 78.55%, 88.98%, and 79.68%. Since illumination variations of images are relatively large in the Extended YaleB database, these experiment results validate the effectiveness and robustness of LGECSDL for image recognition with illumination variations. VGG19 neural network in this experiment can only achieve the highest recognition rate of 53.79%. Using a small database to train neural networks does not take advantage of neural networks. We also verify the effect of α and β on the LGECSDL algorithm, and the experimental results are shown in Figure 2.

From Figure 2, we can easily know that the LGECSDL algorithm has achieved better recognition results in Hellinger kernel space. With the parameter α varied from 2^{-13} to 2^{-3}, the recognition rate increased gradually and then decreased. The influence of parameter β on the LGECSDL algorithm is similar to that of the parameter α. The highest recognition rate was achieved when $\alpha = 2^{-7}$ and $\beta = 2^{-8}$ in Hellinger kernel space. In the linear kernel space, the recognition rate achieves the maximum value at $\alpha = 2^{-9}$ and $\beta = 2^{-10}$, and in the polynomial kernel space, the maximum recognition rate was obtained at $\alpha = 2^{-7}$ and $\beta = 2^{-10}$.

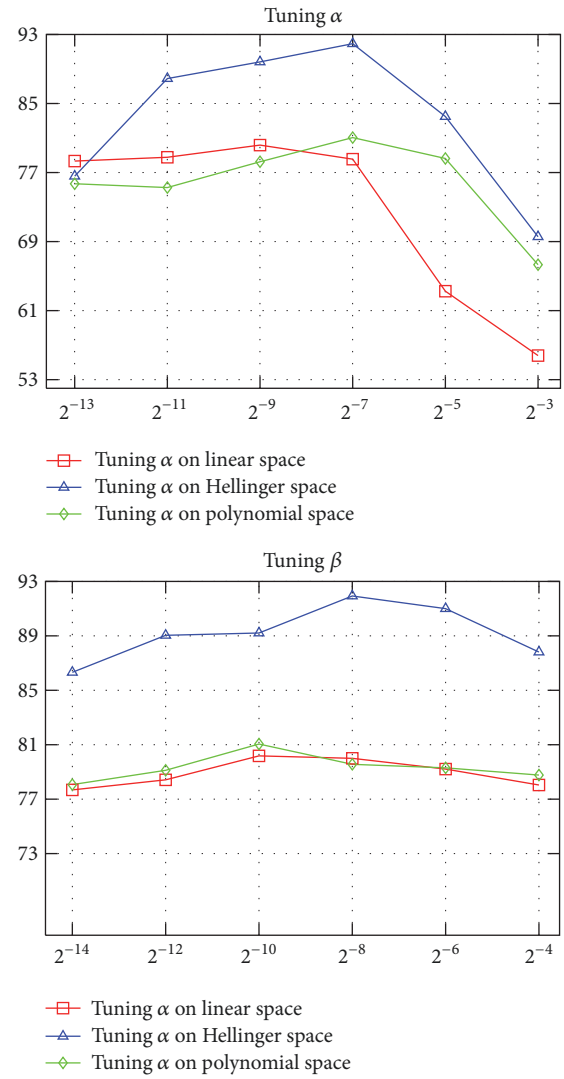

Figure 2: Parameter tuning on the Extended YaleB database.

5.4. AR Database. In this experiment, we randomly selected 5 images of each individual as training samples and the rest for testing. Each image has been cropped to 32 × 32 and pulled into a column vector; the image vectors have been performed by $l2$ normalization. All the methods are independently run ten times, and the average recognition rates are reported. The recognition rate of AR database is shown in Table 2.

From Table 2, we can clearly see that LGECSDL algorithm outperforms the other methods in all kernel spaces. The LGECSDL algorithm achieves the best recognition rate of 94.6% in the polynomial kernel space; in the linear kernel space and Hellinger kernel space, the recognition rates are 94.5% and 94.13%, respectively.

Moreover, we can also know from Table 2 that the KCSDL algorithm achieves the best recognition rate of 91.12% in the linear kernel space, which is the highest one among the other methods. From these experimental results, we further confirm the effectiveness and robustness of the LGECSDL algorithm for image recognition with illumination variations

Table 2: Recognition rate on the AR database (%).

Methods	Linear	Hellinger	Polynomial
NN	32.99 ± 1.97	NA	NA
VGG19	65.80 ± 1.35	NA	NA
SVM	80.67 ± 1.32	80.59 ± 1.26	80.84 ± 1.57
CRC	91.98 ± 0.86	92.18 ± 0.78	92.43 ± 0.75
SRC	89.17 ± 1.14	85.32 ± 1.12	85.25 ± 1.41
ProKCRC	87.74 ± 1.43	91.04 ± 0.76	89.90 ± 1.21
KCSDL	91.12 ± 1.57	89.77 ± 1.27	89.02 ± 1.31
LGECSDL	**94.50 ± 1.01**	**94.13 ± 0.91**	**94.60 ± 0.86**

Table 3: Recognition rate on the CMU-PIE database (%).

Methods	Linear	Hellinger	Polynomial
NN	30.09 ± 1.67	NA	NA
VGG19	61.80 ± 1.28	NA	NA
SVM	66.79 ± 2.63	65.71 ± 2.54	65.46 ± 2.76
CRC	72.89 ± 2.21	75.19 ± 2.09	73.19 ± 2.20
SRC	72.16 ± 2.09	70.26 ± 2.23	69.19 ± 2.17
ProKCRC	68.91 ± 1.97	75.41 ± 1.86	72.64 ± 2.05
KCSDL	74.46 ± 2.01	74.78 ± 2.05	73.49 ± 2.22
LGECSDL	**79.12 ± 1.52**	**81.03 ± 1.34**	**80.05 ± 1.27**

and expression changes. We also verify the effect of α and β on the AR database; Figure 4 shows the experiment results.

From Figure 4, we can clearly see that the recognition rate reached the maximum value when α is 2^{-7} and β is 2^{-8} in Hellinger kernel space and polynomial kernel space; in the linear kernel space, the recognition rate achieves the highest value when α is equal to 2^{-9} and β is 2^{-8}. With α changed from 2^{-13} to 2^{-3}, the recognition rate increased first and then decreased. The β parameter shows a similar trend, and when β is greater than 2^{-8}, the recognition rate decreases rapidly.

5.5. CMU-PIE Database. In this experiment, we chose the CMU-PIE database to evaluate the performance of the LGECSDL algorithm. Five images of each individual are randomly selected for training and the remainder for testing. We also cropped each image to 32×32 and then pulled them into a column vector. Finally, we normalized all the vectors by $l2$ normalization. We independently ran all the methods ten times and then reported the average recognition rates. Table 3 gives the recognition rates of all the methods under different kernel spaces.

From Table 3, we can see that the LGECSDL algorithm always achieves the highest recognition rates under all different kernel spaces. In the polynomial kernel space, the LGECSDL algorithm outperforms KCSDL, which achieves the second highest recognition rate, by more than 4% improvement of recognition rate. In Hellinger kernel space, the LGECSDL achieves the best recognition rate of 81.03% and 6% points higher than the KCSDL algorithm. In the linear kernel space, the face recognition rate of LGECSDL and KCSDL is 79.12% and 74.46%, respectively. From these experimental results, we confirm the effectiveness and robustness

Figure 3: Examples of the AR database.

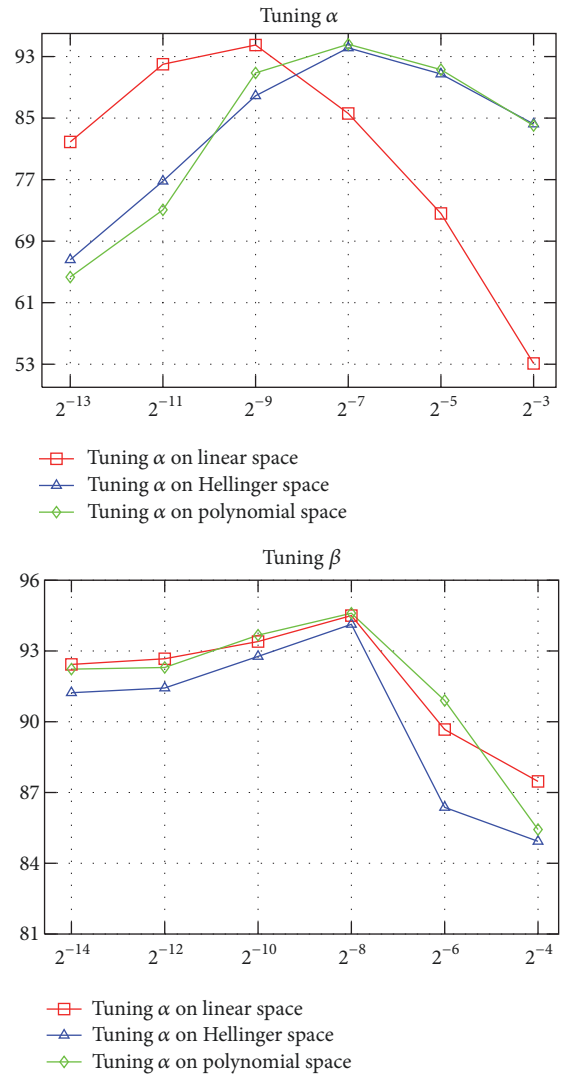

Figure 4: Parameter tuning on the AR database.

of the LGECSDL algorithm. We also evaluate the effect of α and β on the CMU-PIE database; Figure 6 shows the experiment results.

From Figure 6, we can see that when α is 2^{-7} and β is 2^{-8}, the face recognition rate reaches the highest value of 81.03% in Hellinger kernel space, and when α is 2^{-7} and β is 2^{-10}, the highest face recognition rate is obtained in the polynomial kernel space. We can also know from Figure 6 that when α is

Figure 5: Examples of the CMU-PIE database.

Figure 7: Examples of the Caltech101 database.

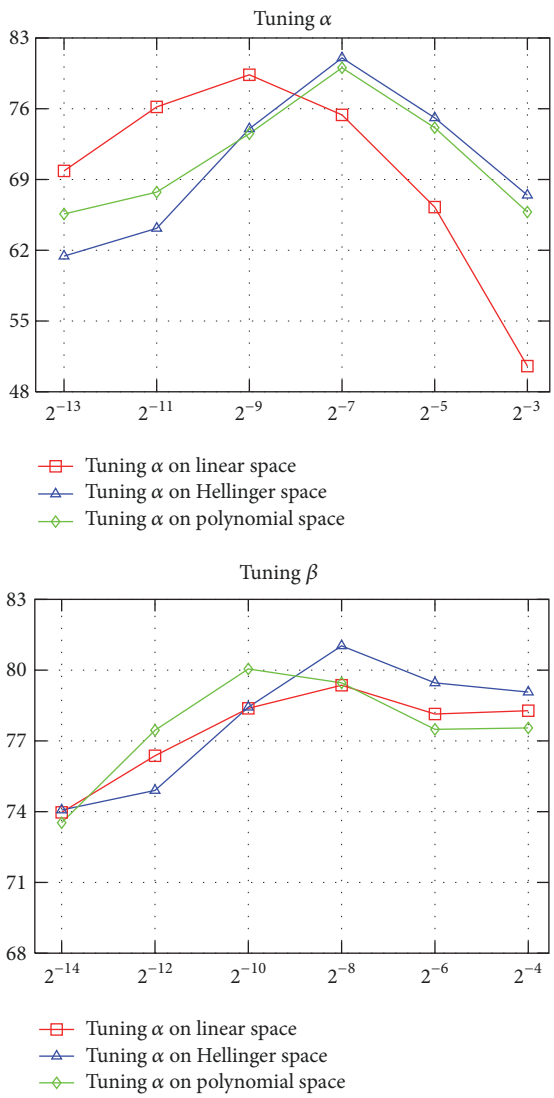

Figure 6: Parameter tuning on the CMU-PIE database.

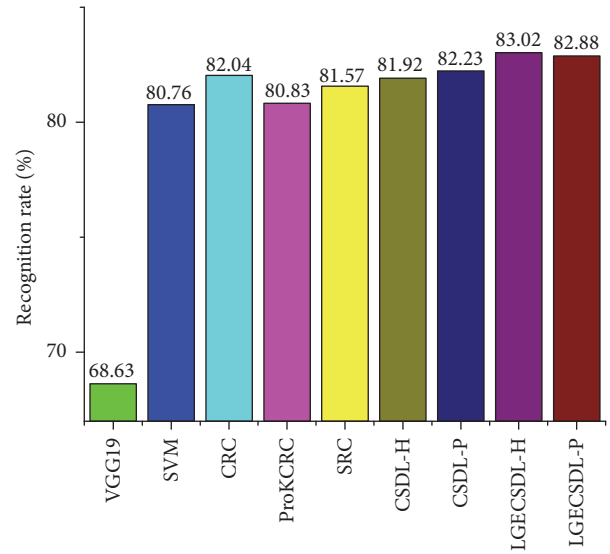

Figure 8: The best recognition rates of all the methods on Caltech101.

greater or less than the maximum value, the recognition rate decreases rapidly, and parameter β also has the same effect on the face recognition rate.

5.6. Caltech101 Database. In this experiment, we further evaluate the performance of the LGECSDL algorithm for image recognition on the Caltech101 database. Figure 7 shows some examples in the Caltech101 database. We randomly split the Caltech101 database into two parts. Part one, which contains about 5 images of each subject, is used as a training set, and the other part is used as a testing set. We use the VGG_ILSVRC_19_layers model to obtain the features of each image. Here, we employ the second fully connected layer outputs as the input features whose dimension size is 4096. We independently ran all the methods ten times and then gave the average recognition rates of all the methods in Figure 8.

LGECSDL-H represents the LGECSDL algorithm in Hellinger kernel space, and LGECSDL-P represents the LGECSDL algorithm in the polynomial kernel space, similarly to the CSDL algorithm. From Figure 8, we can easily see that the LGECSDL-H algorithm achieves the highest recognition rate, and LGECSDL-P is the second one. More concretely, LGECSDL-H and LGECSDL-P achieve the recognition rates of 83.02% and 82.88%, respectively, while CSDL-P, the third best method, arrives at 82.23%. Experimental results show that training the VGG19 network with a small database did not give the desired results. VGG19 network achieved the recognition rate of 68.63%.

We also verified the computational time of each method in Caltech101 database. The experimental environment consists of the following: Core i7 CPU (2.4 GHz), 8 GB memory,

Table 4: Computational time of each method in Caltech101.

Methods	*VGG19*	*SVM*	*CRC*
Time	9.4 ms/pic	83.97 ms/pic	105.723 ms/pic
Methods	*ProKCRC*	*SRC*	*CSDL-H*
Time	128.77 ms/pic	110.99 ms/pic	106.54 ms/pic
Methods	*CSDL-P*	*LGECSDL-H*	*LGECSDL-P*
Time	166.68 ms/pic	109.50 ms/pic	112.25 ms/pic

Figure 9: Examples of the Oxford-102 flower database.

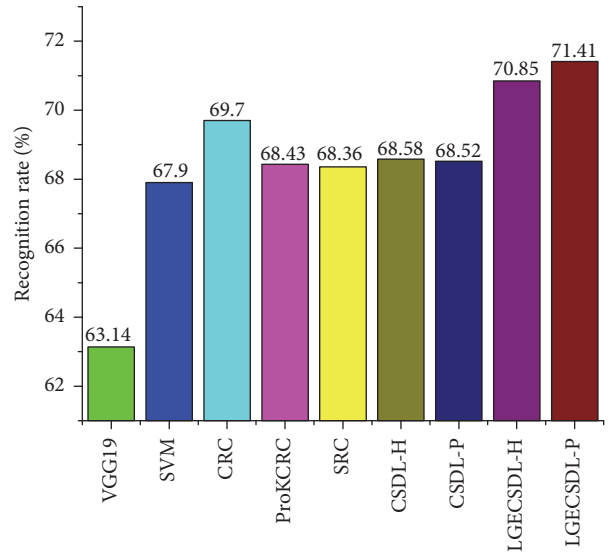

Figure 10: The best recognition rates of all the methods on Oxford-102.

Windows 7 operating system, and NVIDIA Quadro K2100M computer graphics processor. From Table 4, we can see that the VGG19 method achieves the best result. This is mainly due to the neural network GPU accelerated architecture that saves most of the computational time. The second is SVM, followed by CSDL-H algorithm. The LGECSDL-H algorithm needs 109.50 milliseconds to classify each picture, whereas the LGECSDL-P algorithm requires 112.25 milliseconds.

5.7. Oxford-102 Flower Database. In this experiment, we chose the Oxford-102 database to evaluate the performance of the LGECSDL algorithm in the case of image recognition with precise image classification. Five images of each individual are randomly selected for training, and the rest of the images are for testing. The image features are obtained from the outputs of a pretrained VGG_ILSVRC_19_layers network which contains five convolutional and three fully connected layers. Here, we use the second fully connected layer outputs as the input features whose dimension size is 4096. We independently ran all the methods ten times and then reported the average recognition rates. The best recognition rates of all the methods are presented in Figure 10.

From Figure 10, we can clearly know that the LGECSDL achieves the best recognition rates of 71.41% and 70.85% in polynomial kernel space and Hellinger kernel space, respectively, while CRC arrives at 69.7%, which is the highest one among those of the other methods. LGECSDL-H and LGECSDL-P outperform VGG19, SVM, SRC, CRC, ProKCRC, CSDL-H, and CSDL-P by at least 1.2% improvement of recognition rate. The experimental results show that, in the small database experiment, other methods have a higher recognition rate than the VGG19 neural network. The classical method has more advantages in the small sample base experiment.

We also verify the performance of the LGECSDL algorithm with different values of α or β in different kernel spaces on the Oxford-102 database. The performance with different values of α or β is reported in Figures 11 and 12.

From Figure 11, we can see that the LGECSDL algorithm achieves the maximum value when $\alpha = 2^{-4}$ and $\beta = 2^{-4}$; when α is fixed, the recognition rate increases firstly and then decreases with the increase of β. Similarly, the recognition rate also increases firstly and then decreases with the increase of α.

From Figure 12, we can see that the LGECSDL algorithm in the polynomial kernel space achieves the maximum recognition rate when $\alpha = 2^{-5}$ and $\beta = 2^{-5}$. In the polynomial kernel space, the influence of α and β on the algorithm is similar to that in Hellinger kernel space.

6. Conclusion

We present a novel Laplace graph embedding class specific dictionary learning algorithm with kernels. The proposed LGECSDL algorithm improves the classical classification algorithm threefold. First, it concisely combines the discriminant ability (sparse representation) to enhance the interpretability of face recognition. Second, it greatly reduces the residual error according to Laplace constraint dictionary learning. Third, it easily finds the nonlinear structure hidden in face images by extending the LGECSDL algorithm to arbitrary kernel space. Experimental results on several publicly available databases have demonstrated that LGECSDL can provide superior performance to the traditional face recognition approaches.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

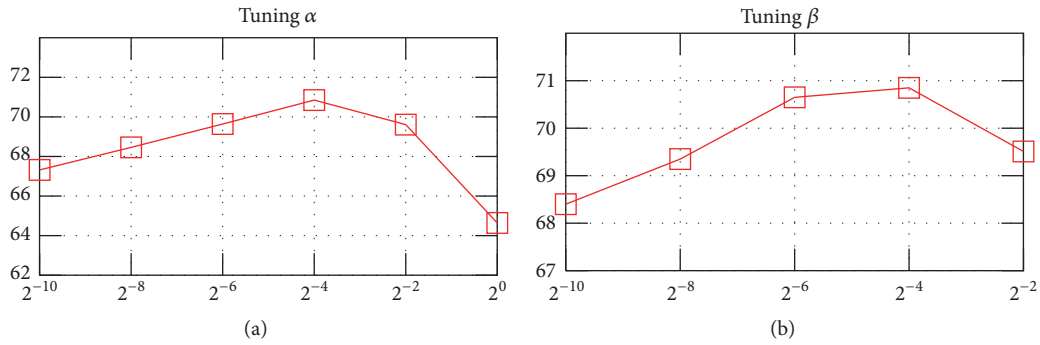

Figure 11: Parameter tuning on Oxford-102 database with Hellinger kernel. (a) Tuning α with $\beta = 2^{-4}$. (b) Tuning β with $\alpha = 2^{-4}$.

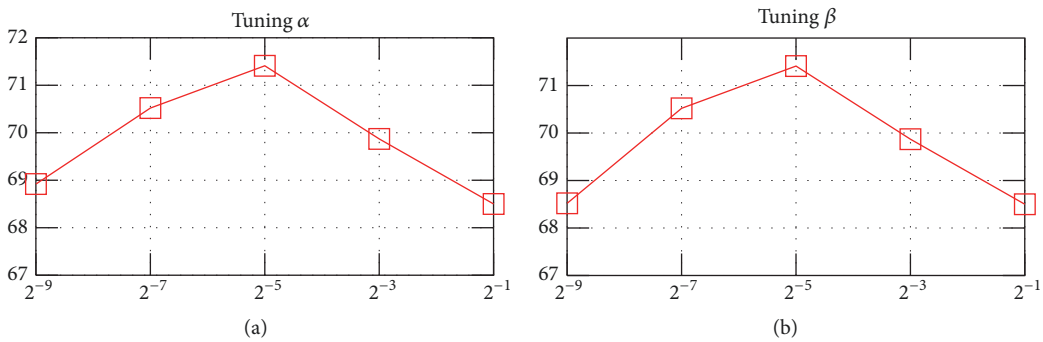

Figure 12: Parameter tuning on Oxford-102 database with polynomial kernel. (a) Tuning α with $\beta = 2^{-5}$. (b) Tuning β with $\alpha = 2^{-5}$.

Acknowledgments

This paper is supported partly by the National Natural Science Foundation of China (Grants nos. 61402535 and 61271407), the Natural Science Foundation for Youths of Shandong Province, China (Grant no. ZR2014FQ001), the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) (Grant no. 16CX02060A), and the International S and T Cooperation Program of China (Grant no. 2015DFG12050).

References

[1] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4014–4024, 2016.

[2] W. Liu, Z.-J. Zha, Y. Wang, K. Lu, and D. Tao, "p-laplacian regularized sparse coding for human activity recognition," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5120–5129, 2016.

[3] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 447–461, 2016.

[4] D. C. Tao, X. Li, X. D. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 260–274, 2009.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, Columbus, Ohio, USA, June 2014.

[6] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 99, pp. 1-1, 2017.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 815–823, IEEE, Boston, Mass, USA, June 2015.

[8] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 2892–2900, Boston, Ma, USA, June 2015.

[9] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: face recognition via deep embedding," 2015, https://arxiv.org/abs/1506.07310.

[10] T. Næs, P. B. Brockhoff, and O. Tomic, "Principal component analysis," in *Statistics for Sensory and Consumer Science*, pp. 209–225, John Wiley & Sons, Hoboken, NJ, USA, 2010.

[11] P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, "Linear discriminant analysis," in *Robust Data Mining*, SpringerBriefs in Optimization, pp. 27–33, Springer, New York, NY, USA, 2013.

[12] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700–1715, 2007.

[13] X. Niyogi, "Locality preserving projections," in *Neural Information Processing Systems*, vol. 16, p. 153, MIT Press, Cambridge, Mass, USA, 2004.

[14] Z.-G. Liu, Q. Pan, and J. Dezert, "A new belief-based K-nearest neighbor classification method," *Pattern Recognition*, vol. 46, no. 3, pp. 834–844, 2013.

[15] Y. Noh, B. Zhang, and D. D. Lee, "Generative local metric learning for nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 106–118, 2018.

[16] 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.

[17] L. Zhang, M. Yang, F. Xiangchu, Y. Ma, and D. Zhang, "Collaborative representation based classification for face recognition," 2012, https://arxiv.org/abs/1204.2358.

[18] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, "Face recognition via weighted sparse representation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 111–116, 2013.

[19] M. Yang, L. Zhang, X. C. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 543–550, Barcelona, Spain, November 2011.

[20] 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.

[21] Z. L. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1697–1704, IEEE, Providence, RI, USA, June 2011.

[22] 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.

[23] Y. Sun, Q. Liu, J. Tang, and D. Tao, "Learning discriminative dictionary for group sparse representation," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3816–3828, 2014.

[24] D. Wang and S. Kong, "A classification-oriented dictionary learning model: explicitly learning the particularity and commonality across categories," *Pattern Recognition*, vol. 47, no. 2, pp. 885–898, 2014.

[25] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: learning discriminative and reconstructive non-parametric dictionaries," ADA513220, Defense Technical Information Center, Fort Belvoir, Va, USA, 2008.

[26] S. Gao, I. W. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 623–634, 2014.

[27] B.-D. Liu, B. Shen, and X. Li, "Locality sensitive dictionary learning for image classification," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '15)*, pp. 3807–3811, Quebec City, Canada, September 2015.

[28] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 2021–2024, IEEE, Kyoto, Japan, March 2012.

[29] B.-D. Liu, Y.-X. Wang, B. Shen, Y.-J. Zhang, and M. Hebert, "Self-explanatory sparse representation for image classification," in *European Conference on Computer Vision*, vol. 8690 of *Lecture Notes in Computer Science*, pp. 600–616, Springer, Berlin, Germany, 2014.

[30] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 471–478, Barcelona, Spain, November 2011.

[31] 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.

[32] S. Cai, L. Zhang, W. Zuo, and X. Feng, "A probabilistic collaborative representation based approach for pattern classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 2950–2959, Las Vegas, Nev, USA, July 2016.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[34] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.