Toward Efficient Image Representation: Sparse Concept Discriminant Matrix Factorization

Meng Pang, Yiu-Ming Cheung, *Fellow, IEEE*, Risheng Liu, *Member, IEEE*, Jian Lou, and Chuang Lin, *Member, IEEE*

***Abstract*—The key ingredients of matrix factorization lie in basic learning and coefficient representation. To enhance the discriminant ability of the learned basis, discriminant graph embedding is usually introduced in the matrix factorization model. However, the existing matrix factorization methods based on graph embedding generally conduct discriminant analysis via a single type of adjacency graph, either similarity-based graphs (e.g., Laplacian eigenmaps graph) or reconstruction-based graphs (e.g., L_1-graph), while ignoring the cooperation of the different types of adjacency graphs that can better depict the discriminant structure of original data. To address the above issue, we propose a novel Fisher-like criterion, based on graph embedding, to extract sufficient discriminant information via two different types of adjacency graphs. One graph preserves the reconstruction relationships of neighboring samples in the same category, and the other suppresses the similarity relationships of neighboring samples from different categories. Moreover, we also leverage the sparse coding to promote the sparsity of the coefficients. By virtue of the proposed Fisher-like criterion and sparse coding, a new matrix factorization framework called Sparse concept Discriminant Matrix Factorization (SDMF) is proposed for efficient image representation. Furthermore, we extend the Fisher-like criterion to an unsupervised context, thus yielding an unsupervised version of SDMF. Experimental results on seven benchmark datasets demonstrate the effectiveness and efficiency of the proposed SDMFs on both image classification and clustering tasks.**

***Index Terms*—Matrix factorization, image representation, graph embedding, Fisher-like criterion, sparse coding.**

I. Introduction

How to make an efficient image representation is a fundamental problem in image processing as input images are typically of high dimensionality. One expects to seek a lower-dimensional hidden subspace to represent original high-dimensional images. Accordingly, matrix factorization-based techniques, which lower down the input dimensionality, have received considerable attentions in the fields of image representation [1]–[3], image classification and clustering [4], [5], image retrieval [6]–[9], and visual tracking [10], to name a few. Specifically, given a matrix with N images $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \Re^{m\times N}$, each column of the matrix is an m-dimensional sample vector corresponding to one image. Matrix factorization aims to find two lower-dimensional matrices: basis matrix $\mathbf{U} \in \Re^{m\times r}$ and coefficient matrix $\mathbf{V} \in \Re^{r\times N}$, satisfying $\mathbf{X} \approx \mathbf{UV}$.

As depicted in [4] and [11]–[14], it is desired for a matrix factorization method to find the basis matrix that is able to uncover the intrinsic structure as well as to capture highly discriminant information of image data. To satisfy these requirements, the graph embedding framework [15] is usually introduced in existing matrix factorization methods by incorporating a graph-regularized constraint that characterizes meaningful structures of image data into the oracle matrix factorization model.

In unsupervised graph embedding framework, the pivotal point is how to knit the graphs to depict the relationships of image data. Currently, there are two major types to build an adjacency graph, one is based on pairwise similarities and the other is based on reconstruction weights [16]. In the former, the typical methods include Laplacian eigenmaps (LE) [17] and its linearized version, i.e., locality preserving projection (LPP) [18]. The target of LE and LPP is to keep the similarity relationships between neighboring image samples. By contrast, the latter methods [19]–[25] assume that each image sample can be represented as a linear combination of other samples in the same subspace. The simplest reconstruction-based graph is LLE-graph [19], which is usually adopted by unsupervised graph embedding methods such as neighborhood preserving embedding (NPE) [20] to preserve the reconstruction relationships among neighboring image samples. However, due to the existence of noise, e.g., illuminations, shadows and corruptions, LLE-graph may not be amenable to reflect the intrinsic geometrical structure of image samples because the reconstruction weights will be seriously affected by the noise. Recently, with the huge success of sparse representation (SR) [26], collaborative representation (CR) [27] and low rank representation (LRR) [28], three more robust reconstruction-based graphs, i.e., L_1-graph [21], L_2-graph [23] and LRR-graph [24], [25], have been developed

Manuscript received August 3, 2017; revised March 10, 2018; accepted October 30, 2018. Date of publication November 6, 2018; date of current version October 29, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61672444, Grant 61272366, and Grant 61672125, in part by the SZSTI Grant JCYJ20160531194006833, in part by the Faculty Research Grant of Hong Kong Baptist University under Project FRG2/17-18/082, and in part by the Fundamental Research Funds for the Central Universities. This paper was recommended by Associate Editor X.-P. Zhang. *(Corresponding author: Yiu-Ming Cheung.)*

M. Pang, Y.-M. Cheung, and J. Lou are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: mengpang@comp.hkbu.edu.hk; ymc@comp.hkbu.edu.hk; jianlou@comp.hkbu.edu.hk).

R. Liu is with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: rsliu@dlut.edu.cn).

C. Lin is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: chuang.lin@siat.ac.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2879833

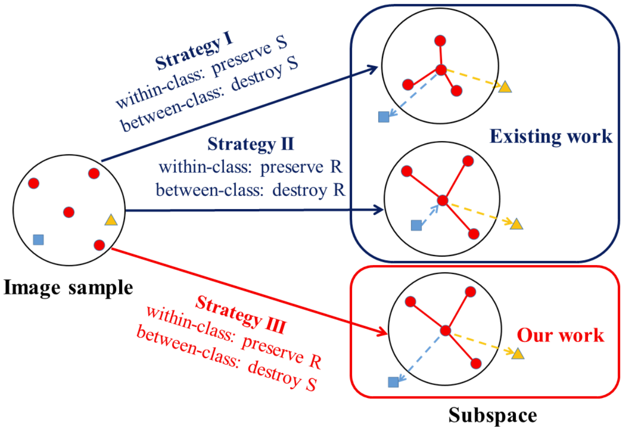


Fig. 1. Comparison of existing two-graph based graph embedding strategies and our Fisher-like criterion. S and R represent similarity-based graph and reconstruction-based graph, respectively. The points with the same color indicate samples in the same category.

to tolerate data noise. Specifically, L_1-graph adds an L_1-norm constraint based on LLE-graph to achieve the sparsest representations of reconstruction weights. L_2-graph characterizes the collaborative representation-based reconstruction relationships with faster speed than L_1-graph by replacing the L_1-norm constraint with the L_2-norm one. LRR-graph recovers the subspace structure from the corrupted image data by imposing a low rank constraint on the reconstruction weights.

When label information of image data is available, the above-mentioned similarity-based graphs or reconstruction-based graphs have also been adopted by some semi-supervised and supervised graph embedding methods [2], [13], [29]–[31], such as supervised LPP (sLPP) [29], robust and discriminative LRR (RDLRR) [31], robust structured subspace learning (RSSL) [2] and discriminant sparse neighborhood preserving embedding (DSNPE) [12], to maintain the similarity or reconstruction relationships of the within-class image samples. Moreover, some attempts [13], [32]–[34] have further been made to use two separate graphs to characterize the within-class structure as well as the between-class structure of image data. The typical two-graph based supervised graph embedding methods include locality sensitive discriminant analysis (LSDA) [13], discriminant locality preserving projection (DLPP) [32], neighbourhood sensitive preserving embedding (NSPE) [33] and discriminative sparsity preserving projection (DSPP) [34]. Although the above two-graph based supervised graph embedding methods can better characterize the discriminant structure by additionally considering the between-class samples, they are still restricted to the mode that uses only one type of adjacency graphs, i.e., similarity-based graphs (e.g., LE-graph) or reconstruction-based graphs (e.g., LLE-graph), to build both the within-class and between-class graphs. For instance, LSDA and DLPP utilize a similarity-based graph (i.e., LE-graph) to build both the within-class and between-class graphs, while NSPE and DSPP build both graphs by replacing LE-graph with LLE-graph and L_1-graph, respectively.

To the best of our knowledge, existing two-graph based supervised graph embedding methods generally leverage the *first* and the *second* graph embedding strategies in Fig. 1, to learn a discriminative subspace. However, we emphasize that the two graph embedding strategies cannot capture the intrinsic discriminant structure of the original image data. This can be intuitively explained as follows:

- The first graph embedding strategy targets learning a subspace where the neighboring within-class image samples are pulled close while the neighboring between-class samples are kept far apart. However, in this graph embedding strategy, the crucial reconstruction structure of the within-class image samples is ignored. As a result, the recovered subspace structure may be skewed from the true subspace structure.
- The second graph embedding strategy targets preserving the reconstruction structure of the within-class image samples while destroying the reconstruction structure of the between-class samples. In this strategy, although the reconstruction structure and the similarity relationships of the within-class image samples can both be reserved, the between-class samples may still have chance to stay nearby because destroying the between-class reconstruction relationships is a weaker penalty compared to directly suppressing the similarities of the between-class samples.

To address these issues, we develop a Fisher-like criterion by conducting discriminant analysis across both reconstruction-based graph and similarity-based graph. The former preserves the reconstruction relationships of neighboring image samples in the same category, and the latter suppresses the similarities of neighboring samples from different categories (see the third graph embedding strategy in Fig. 1).

To handle unlabeled data, we further extend the Fisher-like criterion to an unsupervised context. One simple strategy is to utilize the k-nearest neighboring image samples and the remaining ones to build the within-class reconstruction-based graph and between-class similarity-based graph, respectively. However, this k-NN strategy is sensitive to data noise (e.g., illumination and shadow). Alternatively, it is found that two image samples are likely from different categories if they neither 1) belong to the k-nearest neighbors of each other, nor 2) lie in the same subspace. We further design a more robust heuristic strategy called **h**ybrid **n**earest **n**eighbor (H-NN). Technically, the H-NN strategy takes advantage of k-NN and SR to jointly select the suspected within-class and between-class candidate samples to build both graphs.

After basis learning based on graph embedding, coefficient representation is another key ingredient of matrix factorization. In this paper, we propose a new matrix factorization framework called Sparse concept Discriminant Matrix Factorization (SDMF) by combining the Fisher-like criterion with sparse coding, for efficient image representation. SDMF features the finding of the basis that is able to 1) capture highly discriminant information of the original image data, 2) reflect the intrinsic geometrical structure, and 3) yield a sparse representation under the learnt discriminant basis. Moreover, different from most existing matrix factorization methods that alternate between updating the basis $\mathbf{U}$ and coefficient $\mathbf{V}$, SDMF launches a more aggressive optimization strategy to separately optimize the basis $\mathbf{U}$ and coefficient $\mathbf{V}$

by solving a sparse eigen-problem and two regressions, which is beneficial to computational tractability.

The contributions of our work are summarized as follows:

- We develop a novel Fisher-like criterion to extract sufficient discriminant information via two different types of adjacency graphs, i.e., reconstruction-based graph and similarity-based graph, which can provide a new insight into the graph embedding strategies for building local relationships of image data.
- We extend the Fisher-like criterion to an unsupervised context by proposing two heuristic strategies, i.e., k-NN strategy and H-NN strategy.
- By virtue of the Fisher-like criterion and sparse coding, we propose a new matrix factorization framework, i.e., SDMF, for efficient image representation. Moreover, for the sake of computational cost, we launch an aggressive optimization strategy to solve the SDMF model.

Compared to our preliminary studies in [35], this paper has made four major extensions. 1) We propose the H-NN strategy for the Fisher-like criterion in unsupervised learning. 2) We supply with theoretical supports to manifest that the solving solution of the optimal basis could address the *small sample size (SSS) problem* [36] without pre-dimensionality reduction. 3) The parameter sensitivity of SDMF is studied. 4) More extensive experiments are carried out to evaluate the classification and clustering performance of SDMF, and compared with other state-of-the-art methods, including popular deep learning based methods.

The rest of this paper is organized as follows: In Section II, we will introduce the proposed SDMF model in detail. The experimental results are presented in Section III. Finally, we give the concluding remarks in Section IV.

II. Sparse Concept Discriminant Matrix Factorization (SDMF)

Our SDMF framework integrates the Fisher-like criterion and sparse coding into a unified framework, which is generic enough to incorporate various graphs for capturing highly discriminant information. In particular, we derive our model with two prevalent graphs of LLE-graph and LE-graph for the supervised SDMF. The detailed formulation and learning algorithm are presented below in this section.

A. Formulation

Most existing matrix factorization methods, e.g., sparse coding (SC) [37]–[39] and nonnegative matrix factorization (NMF) [4], [11], [14] methods, usually alternate between updating the basis $\mathbf{U}$ and coefficient $\mathbf{V}$, which is very time-consuming. By contrast, the SDMF model is proposed to separately optimize the basis $\mathbf{U}$ and coefficient $\mathbf{V}$ by directly solving the three sequential optimization problems as follows:

- **Basis learning:**

$$\mathbf{Y} = \arg\min_{\mathbf{Y}} \frac{\mathbf{Y}^T\mathbf{M}^w\mathbf{Y}}{\mathbf{Y}^T\mathbf{L}^b\mathbf{Y}}, \quad (1)$$

$$\mathbf{U} = \arg\min_{\mathbf{U}} \|\mathbf{X}^T\mathbf{U} - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{U}\|^2. \quad (2)$$

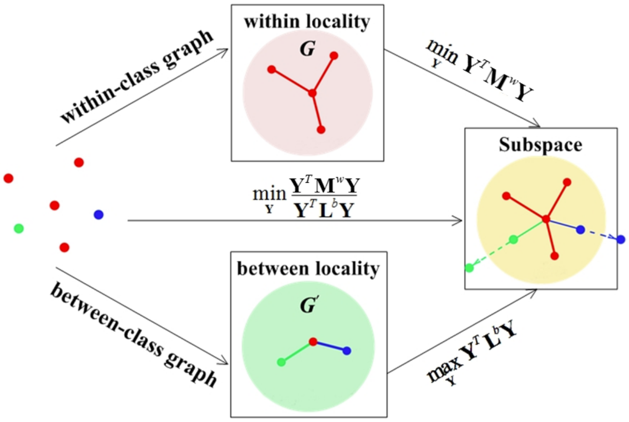

within-class graph
within locality
G
$\min_{\mathbf{Y}} \mathbf{Y}^T\mathbf{M}^w\mathbf{Y}$
$\min_{\mathbf{Y}} \frac{\mathbf{Y}^T\mathbf{M}^w\mathbf{Y}}{\mathbf{Y}^T\mathbf{L}^b\mathbf{Y}}$
Subspace
between locality
G'
between-class graph
$\max_{\mathbf{Y}} \mathbf{Y}^T\mathbf{L}^b\mathbf{Y}$

Fig. 2. Fisher-like criterion in S-SDMF. The points with the same color belong to the same category.

- **Coefficient representation:**

$$\mathbf{V} = \arg\min_{\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \beta\|\mathbf{V}\|_1. \quad (3)$$

Basis learning:

- In Eq. (1), the minimization of $\frac{\mathbf{Y}^T\mathbf{M}^w\mathbf{Y}}{\mathbf{Y}^T\mathbf{L}^b\mathbf{Y}}$ interprets the Fisher-like criterion, which aims to find good mapping $\mathbf{Y} \in \Re^{N\times r}$ capturing highly discriminant information as well as the intrinsic manifold structure in the training set. Specifically, in the supervised version of SDMF (S-SDMF), the numerator $\mathbf{Y}^T\mathbf{M}^w\mathbf{Y}$ is generated from LLE-graph [19] to maintain the reconstruction relationship for the image samples in the same category, while the denominator $\mathbf{Y}^T\mathbf{L}^b\mathbf{Y}$ is extracted from the LE-graph [17] to keep away neighboring samples from different categories. $\mathbf{M}^w$ and $\mathbf{L}^b$ denote the within-class and between-class scatter matrix, respectively (details refer to Subsection II-B). On the other hand, in the unsupervised version of SDMFs (U-SDMFs), the two graphs that depict the local discriminant structure can be built resorting to two heuristic strategies, i.e., k-NN and H-NN strategies. The Fisher-like criterion in S-SDMF is illustrated in Fig. 2.
- Eq. (2) is a relaxation of the linear equation system $\mathbf{X}^T\mathbf{U} = \mathbf{Y}$, which enables the mapping $\mathbf{Y}$ to extend to all image samples (including testing ones), and find the corresponding basis $\mathbf{U} \in \Re^{m\times r}$ to inherit highly discriminant ability of $\mathbf{Y}$.

Coefficient representation: In Eq. (3), the minimization of $\|\mathbf{V}\|_1$ aims at achieving sparse representation of the coefficients under the learnt discriminant basis, thus enabling each image sample to be represented by a linear combination of only few key basis vectors.

To summarize, SDMF is a two-step model for matrix factorization, i.e, basis learning followed by coefficient representation. In SDMF, the mapping $\mathbf{Y}$, basis $\mathbf{U}$ and coefficient $\mathbf{V}$ are optimized forward and sequentially, the flow is somewhat different from the alternating direction iteration strategy that optimizes one variable by fixing the values of other variables as the outputs at the previous iteration. The SDMF model only needs to solve a sparse eigen-problem and two regression problems, which makes SDMF time-efficient.

Please note that, the transition from the mapping $\mathbf{Y}$ to the basis $\mathbf{U}$ in Eq. (2) is an innovative way to address the *SSS problem* without pre-dimensionality reduction, and meanwhile can alleviate the over-fitting in the training phase, which has already been approved by the popular spectral regression framework [40]. Besides, one related method named sparse concept coding (SCC) has been reported in [41]. Although SDMF and SCC both incorporate spectral regression and share similar optimization scheme, SDMF is different from SCC from two aspects. First, SCC simply uses LE-graph to characterize the data structure, while SDMF leverages the two-graph based Fisher-like criterion to uncover the intrinsic manifold structure as well as to capture hidden discriminant information. Second, SCC is an unsupervised method, while SDMF is implemented in both the supervised and unsupervised forms to tackle image classification and clustering tasks.

B. Basis Learning

The structure of basis learning process can be stated formally as follows.

1) Graph Construction Strategy: Let G and G' denote two adjacency graphs both over training dataset. Let $\mathbf{S}_{ij}^{w}$ and $\mathbf{S}_{ij}^{b}$ be the reconstruction weight and affinity weight of the edge joining vertices i and j in G and G', respectively. We define $N_k(\mathbf{x}_i)$ as the set of k nearest neighbors of $\mathbf{x}_i$. c_i and c_j denote the class label of $\mathbf{x}_i$ and $\mathbf{x}_j$, the total number and the feature dimension of image samples are defined as N and m, respectively.

For S-SDMF, we build the within-class graph G and the between-class graph G' by using LLE-graph and LE-graph, respectively. Accordingly, we let $\mathbf{S}_{ij}^{w} = \arg\min_{\mathbf{S}_{ij}^{w}} \sum_i \|\mathbf{x}_i - \sum_j \mathbf{S}_{ij}^{w}\mathbf{x}_j\|^2$, if $c_i = c_j$, and $\mathbf{S}_{ij}^{w} = 0$ otherwise. $\mathbf{S}_{ij}^{b}$ can be defined in a "simple-minded" or "heat-kernel" way [17]. For simplicity, we use the former way to define $\mathbf{S}_{ij}^{b}$; namely, if $c_i \neq c_j$, $\mathbf{S}_{ij}^{b} = 1$, otherwise $\mathbf{S}_{ij}^{b} = 0$.

For U-SDMFs, two heuristic strategies, i.e., k-NN strategy and H-NN strategy, are proposed to build the intrinsic graph G and the penalty graph G'. Then the $\mathbf{S}^{w}$ in graph G and $\mathbf{S}^{b}$ in graph G' can be computed according to the following criteria.

In k-NN strategy, we simply use pair-wise Euclidean distance to measure the similarity of image samples, and select k nearest neighboring samples to build G and the remaining ones to build G'. $\mathbf{S}_{ij}^{w}$ is defined as $\mathbf{S}_{ij}^{w} = \arg\min_{\mathbf{S}_{ij}^{w}} \sum_i \|\mathbf{x}_i - \sum_j \mathbf{S}_{ij}^{w}\mathbf{x}_j\|^2$ if $\mathbf{x}_i \in N_k(\mathbf{x}_j)$ or $\mathbf{x}_j \in N_k(\mathbf{x}_i)$, and $\mathbf{S}_{ij}^{w} = 0$ otherwise. Subsequently, we define $\mathbf{S}_{ij}^{b} = 0$, if $\mathbf{x}_i \in N_k(\mathbf{x}_j)$ or $\mathbf{x}_j \in N_k(\mathbf{x}_i)$, and $\mathbf{S}_{ij}^{b} = 1$ otherwise.

The H-NN strategy draws inspiration from a novel observation that two image samples are likely from different categories if they neither 1) belong to the k-nearest neighbors of each other, nor 2) lie in the same subspace. Considering that SR has been approved to possess natural discriminating power and could characterize the data subspace structure implicitly [22], [26], [34], we then attempt to combine the ideas of k-NN and SR to cooperatively select the suspected within-class and between-class candidate samples.

- First, we compute the sparse representation weight matrix $\mathbf{W}$ of $\mathbf{X}$ by the following optimization problem:

$$\mathbf{W} = \arg\min_{\mathbf{W}} \|\mathbf{W}\|_1 + \frac{\gamma}{2}\|\mathbf{Z}\|_F^2 \quad s.t.\ \mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{Z},\ diag(\mathbf{W}) = 0, \tag{4}$$

  where the L_1-norm promotes the sparsity of the columns of $\mathbf{W}$, γ is a balance parameter, and $\mathbf{Z}$ indicates the noise matrix. The solution $\mathbf{W}$ of Eq. (4) could be solved efficiently using convex programming tools, and the reconstruction weight matrix $\mathbf{S}^{w}$ is computed by $\mathbf{S}^{w} = |\mathbf{W}| + |\mathbf{W}|^T$. For each image $\mathbf{x}_i$, we select the samples with non-negative values in $\mathbf{S}_i^{w}$ to form the suspected within-class candidate set, i.e., $N_w(\mathbf{x}_i)$, and the remaining ones with zero values are then grouped to the set $N_r(\mathbf{x}_i)$ that are likely from different subspaces.
- Furthermore, the k-NN strategy could also make contribution to the selection of neighboring samples. For each image $\mathbf{x}_i$, assuming the number of samples in $N_w(\mathbf{x}_i)$ is k_w, we then apply k-NN to get k_w nearest neighboring samples. In this case, the remaining $N - k_w$ image samples are then grouped to the set $\widehat{N}_r(\mathbf{x}_i)$. According to the above observation, the suspected between-class candidate set $N_b(\mathbf{x}_i)$ for image $\mathbf{x}_i$ can be decided by the intersection of $N_r(\mathbf{x}_i)$ and $\widehat{N}_r(\mathbf{x}_i)$, i.e., $N_r(\mathbf{x}_i) \cap \widehat{N}_r(\mathbf{x}_i)$. Hence, the affinity weight $\mathbf{S}_{ij}^{b}$ is defined by letting $\mathbf{S}_{ij}^{b} = 1$ if $\mathbf{x}_j \in N_b(\mathbf{x}_i)$ or $\mathbf{x}_i \in N_b(\mathbf{x}_j)$, and $\mathbf{S}_{ij}^{b} = 0$ otherwise.

The k-NN and H-NN are both effective strategies to build local relationships of image data for unsupervised learning. The k-NN strategy is succinct and time-efficient. However, it is sensitive to the imbalance of the feature distribution and data noise. By contrast, the H-NN strategy constructs a hybrid-graph by using both the tangent space in the neighborhood [42] and sparse reconstructed subspace to characterize the local geometrical structure of image data, which can be more discriminative and robust compared to the k-NN strategy. The clustering experiments in Section III will validate the rationality and effectiveness of the H-NN strategy. It is worth noting that, for k-NN strategy, there is one parameter k needs to be manually assigned. While in H-NN strategy, we only need to determine the value of the balance parameter γ in Eq. (4), and the suspected within-class and between-class candidate set $N_w(\mathbf{x}_i)$ and $N_b(\mathbf{x}_i)$ can be obtained automatically following the H-NN procedure.

2) Updating Mapping Y: Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_r] \in \Re^{N \times r}$ be the mapping from the graph to the real line [40]. The optimal $\mathbf{Y}$ is given by optimizing the following objective functions generated from LLE-graph and LE-graph, respectively:

$$\min \Phi^{w}(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_j \mathbf{S}_{ij}^{w}\mathbf{y}_j\|_F^2, \tag{5}$$

$$\max \Phi^{b}(\mathbf{Y}) = \frac{1}{2}\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|_F^2 \mathbf{S}_{ij}^{b}. \tag{6}$$

By employing some algebraic steps, we have:

$$\begin{aligned}
\Phi^{w}(\mathbf{Y}) &= \sum_{i} \|\mathbf{y}_i - \sum_{j} \mathbf{S}^{w}_{ij}\mathbf{y}_j\|_F^2 \\
&= \|(\mathbf{I}-\mathbf{S}^{w})\mathbf{Y}\|_F^2 \\
&= tr\{\mathbf{Y}^T[\mathbf{I}-(\mathbf{S}^{w})^T](\mathbf{I}-\mathbf{S}^{w})\mathbf{Y}\} \\
&= tr\{\mathbf{Y}^T[\mathbf{I}-(\mathbf{S}^{w})^T-\mathbf{S}^{w}+(\mathbf{S}^{w})^T\mathbf{S}^{w}]\mathbf{Y}\} \\
&= tr(\mathbf{Y}^T\mathbf{M}^{w}\mathbf{Y}), \\
\Phi^{b}(\mathbf{Y}) &= \frac{1}{2}\sum_{i,j}\|\mathbf{y}_i-\mathbf{y}_j\|_F^2\mathbf{S}^{b}_{ij} \\
&= \frac{1}{2}\sum_{i,j}(\mathbf{y}_i-\mathbf{y}_j)^T(\mathbf{y}_i-\mathbf{y}_j)\mathbf{S}^{b}_{ij} \\
&= \frac{1}{2}tr(\mathbf{2Y}^T\mathbf{D}^{b}\mathbf{Y}-\mathbf{2Y}^T\mathbf{S}^{b}\mathbf{Y}) \\
&= tr[\mathbf{Y}^T(\mathbf{D}^{b}-\mathbf{S}^{b})\mathbf{Y}] \\
&= tr(\mathbf{Y}^T\mathbf{L}^{b}\mathbf{Y}),
\end{aligned}$$

where $\mathbf{M}^{w} = (\mathbf{I}-\mathbf{S}^{w})^T(\mathbf{I}-\mathbf{S}^{w})$, $\mathbf{D}^{b}_{ii} = \Sigma_j\mathbf{S}^{b}_{ij}$, $\mathbf{L}^{b} = \mathbf{D}^{b}-\mathbf{S}^{b}$. Then, the minimization problem in Eq. (5) and the maximization problem in Eq. (6) can be summarized as the following novel Fisher-like criterion:

$$\min_{\mathbf{Y}} \frac{\Phi^{w}(\mathbf{Y})}{\Phi^{b}(\mathbf{Y})} = \frac{tr(\mathbf{Y}^T\mathbf{M}^{w}\mathbf{Y})}{tr(\mathbf{Y}^T\mathbf{L}^{b}\mathbf{Y})}. \tag{7}$$

The above minimization problem leads to solving the following generalized eigenvalue problem:

$$\mathbf{M}^{w}\mathbf{Y} = \lambda\mathbf{L}^{b}\mathbf{Y}, \tag{8}$$

where λ is the eigenvalue to the problem. The solution $\mathbf{Y}$ is the eigenvectors of the above generalized eigen-problem with respect to the smallest eigenvalues. Each row of $\mathbf{Y}$ can be viewed as the flat embedding for the data points which unfold the training data manifold [41].

3) Updating Basis U: The graph embedding approach only provides the mapping $\mathbf{Y}$ for the graph vertices in the training set. Extending to all samples, including new testing samples, our SDMF tends to learn the basis $\mathbf{U}$ which satisfies $\mathbf{X}^T\mathbf{U} = \mathbf{Y}$. Then, the Eq. (8) could be converted to the general eigen-decomposition problem in graph embedding framework [15]:

$$\mathbf{XM}^{w}\mathbf{X}^T\mathbf{U} = \lambda\mathbf{XL}^{b}\mathbf{X}^T\mathbf{U}. \tag{9}$$

Unfortunately, in practical applications, the number of samples is always far less than the dimension of features ($N \ll m$), then we can say the linear equations system $\mathbf{X}^T\mathbf{U} = \mathbf{Y}$ is underdetermined. As a result, the matrix $\mathbf{XL}^{b}\mathbf{X}^T$ could be singular ($rank(\mathbf{XL}^{b}\mathbf{X}^T) < m$), and the Eq. (9) doesn't work under the circumstance.

Unlike the conventional subspace learning methods [12], [13], [18], [20], [22], which introduce an additional PCA [43] projection preprocessing procedure to reduce the dimensionality of image features, we leverage the spectral regression framework [40] to address the above *SSS problem* [36]. Accordingly, we relax the linear equation system $\mathbf{X}^T\mathbf{U} = \mathbf{Y}$ as follows:

$$\mathbf{U} = \arg\min_{\mathbf{U}} \ \|\mathbf{X}^T\mathbf{U}-\mathbf{Y}\|_F^2 + \alpha\|\mathbf{U}\|^2, \tag{10}$$

where $\alpha \geq 0$ is a parameter to control the amounts of shrinkage of the above *ridge regression* problem. By taking the derivative of Eq. (10) with respect to $\mathbf{U}$ and setting it to zero, we can obtain the optimal basis $\mathbf{U}^*$ by

$$\mathbf{U}^* = (\mathbf{XX}^T + \alpha\mathbf{I})^{-1}\mathbf{XY}. \tag{11}$$

Obviously, when $\alpha > 0$, the optimal solution obtained by Eq. (11) will not satisfy the linear equation system $\mathbf{X}^T\mathbf{U} = \mathbf{Y}$, and $\mathbf{U}^*$ cannot be the eigenvector matrix of the eigen-decomposition problem in Eq. (9). Hence, it's important to find when Eq. (11) gives the exact solution of Eq. (9). Fortunately, there exists the following **Theorem 1**:

Theorem 1: Suppose $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_r] \in \Re^{N\times r}$, where $\boldsymbol{y}$ is the eigenvector of eigen-decomposition problem in Eq. (8), r is the eigenvector number. If $\boldsymbol{y}$ is in the space spanned by row vectors of $\boldsymbol{X}$, the corresponding projective basis $\boldsymbol{U}^*$ calculated in Eq. (11) will be the exact solution in Eq. (9) when regularization parameter α deceases to zero (Proof see Appendix).

C. Coefficient Representation

After we get the basis $\mathbf{U}$, the representation $\mathbf{V}$ can be calculated column by column independently through the following *lasso regression* [44] problem:

$$\min_{\mathbf{V}} \ \|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|_F^2 + \beta|\mathbf{v}_i|, \tag{12}$$

where $\mathbf{x}_i$ and $\mathbf{v}_i$ are the i-th columns of $\mathbf{X}$ and $\mathbf{V}$, respectively. $|\mathbf{v}_i|$ denotes the L_1-norm of $\mathbf{v}_i$, which is added to ensure the sparseness of $\mathbf{v}_i$. Subsequently, we employ the *Least Angel Regressions* (LARs) [45] to solve the following equivalent formulation to the above regression problem in Eq. (12):

$$\min_{\mathbf{V}} \|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|_F^2 \quad s.t. \ |\mathbf{v}_i| \leq \tau. \tag{13}$$

It is worth noting that, LARs choose to control the sparseness of $\mathbf{v}_i$ by specifying the cardinality (the number of non-zero entries) of $\mathbf{v}_i$ instead of setting the parameter τ. Moreover, the coefficient representation of a testing sample can also be computed in a similar way as Eq. (13).

D. Algorithm Complexity Analysis

The proposed SDMF algorithm involves two phases: basis learning and coefficient representation.

Basis learning can be divided into three parts, including 1) construction of the reconstruction-based graph G and similarity-based graph G', 2) calculation of the reconstruction weight matrix $\mathbf{S}^{w}$ and affinity weight matrix $\mathbf{S}^{b}$, and 3) spectral regression. While in coefficient representation phase, the time complexity of LARs is considered.

1) S-SDMF: In basis learning phase, the two adjacency graphs are built directly by using label information, then the computational time of the *first part* could be ignored. For the *second part*, we denote the average number of samples in each category as k_c, then the complexities of computing the reconstruction weight matrix $\mathbf{S}^{w}$ and the affinity weight matrix $\mathbf{S}^{b}$ are $O(mNk_c^3)$ [46] and $O(N^2)$, respectively. The spectral regression computation in the *third part* involves two steps: response generation (calculate the eigenvectors of sparse

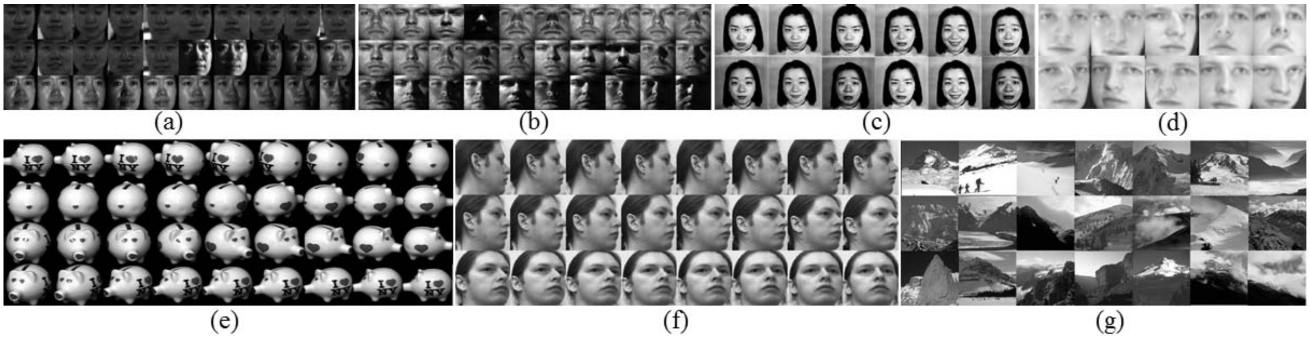


Fig. 3. Some samples for one subject of seven benchmark datasets: (a) CMU PIE; (b) E-YaleB; (c) JAFFE; (d) ORL; (e) COIL20; (f) UMIST; (g) Scene15.

eigen-decomposition problem in Eq. (8)) and regularized least squares. Let r denote the number of eigenvectors, then the cost of the first step requires $O(rNk_c)$ [41]. The regularized least squares problem in Eq. (10) can be efficiently solved by the iterative algorithm LSQR [40]. Denote t as the iteration number of LSQR algorithm, then the cost for r projective functions is $rt(2mN + 3N + 5m)$. Thus the computational complexity of the spectral regression can be $O(rtmN+rNk_c)$.

In coefficient representation phase, LARs can compute the entire solution path (the solutions with all the possible cardinality on $\mathbf{u}_i$) of the problem Eq. (13) in $O(N^3 + mN^2)$. Overall, because $r \ll N, k_c \ll N$, the time complexity of S-SDMF can be summarized as

$$O(N^3 + mN^2 + rtmN + mNk_c^3).$$

Obviously, if the iteration number t is of the same order of magnitude as N, the complexity of S-SDMF is comparable to the ordinary non-sparse solution solved by generalized eigen-problem ($O(N^3 + mN^2)$), i.e. sNPE [30], NSPE [33], which is time-efficient in reality.

2) U-SDMFs: Regarding U-SDMF with k-NN strategy, the time complexity of finding k nearest neighboring samples and the remaining ones to build both adjacency graphs (G and G') in basis learning is $O(mN^2)$ [47]. Similarly for S-SDMF, the overall time complexity of U-SDMF with k-NN strategy can be summarized as

$$O(N^3 + mN^2 + rtmN + mNk^3).$$

By contrast, U-SDMF with H-NN strategy requires an additional step by solving the optimization problem in Eq. (4). As a result, the total complexity of U-SDMF with H-NN strategy is correlated to the selection of the optimization tools.

III. Experimental Results

In this section, six experiments are performed to show the effectiveness and efficiency of the proposed SDMF method from different perspectives. The six experimental parts are listed as follows:

- We evaluate the discriminant ability of the proposed Fisher-like criterion in Subsection III-B.
- We evaluate the performance of S-SDMF on face recognition task in Subsection III-C.
- We evaluate the performance of the U-SDMFs on image clustering task in Subsection III-D.
- We study the parameter sensitivity of S-SDMF and U-SDMFs in Subsection III-E.
- We test the computational time of S-SDMF and U-SDMFs in Subsection III-F.
- We investigate the feasibility of combining S-SDMF with deep features on challenging scene classification task in Subsection III-G.

All experiments are carried out on a PC (CPU: Intel Core i7-4790K 4.00GHz, RAM: 16GB).

A. Data Description and Presentation

We utilize 7 benchmark datasets, including CMU PIE, Extended YaleB, JAFFE, ORL, COIL20, UMIST and Scene15 datasets. Fig. 3 shows some sample images of the involved datasets, and some brief descriptions are presented below:

- CMU PIE [48] consists of 41,368 images of 68 subjects, and each subject involves 43 different illumination conditions, 13 different poses, and 4 different expressions.
- Extended YaleB (E-YaleB) [49] consists of 2414 frontal face images of 38 subjects under 9 poses and 64 illumination conditions.
- JAFFE [50] contains 213 images of 7 facial expressions posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects.
- ORL [51] contains 40 individuals. Each of them includes 10 different images, which show variations in facial expressions, facial details and poses.
- COIL20 [52] contains gray scale images of 20 objects viewed from varying angles and each object has 72 images.
- UMIST [53] contains 20 individuals of 575 face images. Each of them includes 19 to 48 different images, which show variations in poses.
- Scene15 [54] includes 4,485 images from 15 outdoor and indoor scene categories, each category contains 200 to 400 gray images.

B. Discriminant Ability Evaluation

This subsection evaluates the discriminant ability of the proposed Fisher-like criterion by performing dimensionality reduction and then visualizing the subspace representation.

The comparing methods include PCA [43] and 4 related supervised graph embedding techniques, i.e., supervised LPP (sLPP) [29], supervised NPE (sNPE) [30], LSDA [13]

TABLE I

GRAPH EMBEDDING STRATEGIES OF THE INVOLVED METHODS

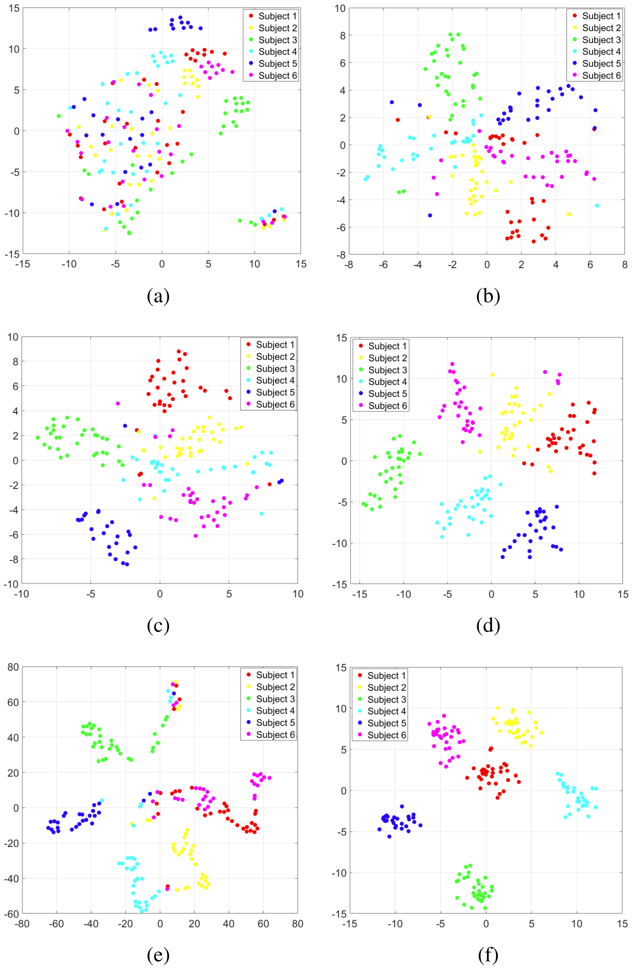
Method	Within-class	Between-class
sLPP	preserve LE-graph	–
sNPE	preserve LLE-graph	–
LSDA	preserve LE-graph	destroy LE-graph
NSPE	preserve LLE-graph	destroy LLE-graph
DLPE	preserve LLE-graph	destroy LE-graph

and NSPE [33]. Note that, to make a fair comparison with these methods, we leave out the sparse representation phase and simply utilize the projection basis learnt by the Fisher-like criterion to obtain the lower-dimensional subspace representation (i.e., $\mathbf{V} = \mathbf{U}^T\mathbf{X}$). For simplicity, this graph embedding method that leverages the Fisher-like criterion is called **d**iscriminant **l**ocality **p**reserving **e**mbedding (DLPE) thereinafter.

In this experiment, we randomly select 6 subjects (Subject 1-6, each subject contains 32 images) on E-YaleB dataset for evaluation. We start by using PCA, sLPP, sNPE, LSDA, NSPE and DLPE to reduce the dimensionality of the evaluated data to 50. Then, we utilize the powerful visualization tool, i.e., t-SNE [55], to convert the 50-dimensional representation to a two-dimensional map. Table I summarizes the graph embedding strategies of the involved methods, and the visualization of the subspace representations are presented in Fig. 4. From the 6 sub-figures in Fig.4, we have the following observations. First, in the subspace representations, the separability of class clusters of DLPE is obvious better than that using PCA, sLPP, sNPE, LSDA and NSPE. Second, the representations after PCA projection tend to be rather scattered and the between-class marginal is not clear. Third, the scatterings of sLPP and sNPE representations are better than those in PCA to some degree, but there are still a large amount of overlaps between different classes. Forth, the class clusters of LSDA and NSPE are well separated with a small amount of overlapping points. In a nutshell, the above sub-figures empirically verify the superior discriminant ability of DLPE over the comparing graph embedding methods.

The discussions of the above experimental phenomena are presented below:

- sLPP and sNPE are both one-graph based graph embedding methods. The two methods respectively use LE-graph and LLE-graph to maintain the similarity or reconstruction relationships of the within-class image samples, while ignoring punishing the between-class samples. As a result, there exist a large amount of overlaps between different classes after sLPP and sNPE projections.
- LSDA, NSPE and DLPE are all two-graph based graph embedding methods. LSDA uses LE-graph to build the within-class graph and between-class graph, while NSPE leverages LLE-graph to build both graphs. By additionally considering the between-class image samples, LSDA and NSPE can achieve better subspace distributions than sLPP and sNPE. However, in LSDA, preserving the within-class LE-graph cannot maintain the reconstruction structure in each class cluster; while in NSPE, destroying the between-class LLE-graph is a weak penalty which may not successfully keep away the neighboring between-class image samples. Hence, there are still a small amount of overlapping points after LSDA and NSPE projections. By contrast, DLPE combines the advantages of LLE-graph and LE-graph, which successfully preserves the within-class reconstruction structure and meanwhile maximizing the between-class marginal. Consequently, DLPE outperforms LSDA and NSPE with respect to the discriminant ability.



(a) (b) (c) (d) (e) (f)

Fig. 4. Visualization of the subspace representations after (a) PCA, (b) sLPP, (c) sNPE, (d) LSDA, (e) NSPE, and (f) DLPE projections on E-YaleB dataset.

C. Face Recognition Experiments

In this subsection, the recognition performance of S-SDMF is evaluated on CMU PIE [48] and E-YaleB [49], respectively. For CMU PIE dataset, we select a subset of 1700 images of 10 subjects that contain five near frontal poses (C05, C07, C09, C27, C29) for evaluation. l images ($l = 45, 65, 85$) are randomly selected from the image gallery of each individual to form the training set Gm (G45, G65, G85), and the remaining $170 - l$ images are taken to form the testing set Pn (P125, P105, P85). For E-YaleB dataset, we randomly take 1280 images of 20 subjects for evaluation. A random subset with Gm (G16, G24, G32) is taken to form the training set,

TABLE II

THE TOP AVERAGE RECOGNITION RATES OF DIFFERENT METHODS ON CMU PIE DATASET

Train/Test	PCA	SPP	LRPE	LDA	sLPP	sNPE	LSDA	NSPE	DSNPE	DSPP	SRC	CRC	FDDL	S-SDMF
G45/P125	79.3	82.3	83.2	90.5	87.4	86.7	89.3	88.6	92.0	90.3	91.5	92.3	92.6	**94.3**
G65/P105	83.8	87.7	87.3	90.2	88.0	89.4	90.7	92.8	95.4	94.8	95.4	95.7	97.1	**97.8**
G85/P85	85.8	88.7	89.8	90.7	89.5	90.6	91.1	91.9	96.0	94.2	94.5	95.9	97.8	**98.5**
Avg.	82.9	86.2	86.8	90.4	88.3	88.9	90.4	91.1	94.4	93.1	93.8	94.6	95.8	**96.9**

TABLE III

THE TOP AVERAGE RECOGNITION RATES OF DIFFERENT METHODS ON E-YALEB DATASET

Train/Test	PCA	SPP	LRPE	LDA	sLPP	sNPE	LSDA	NSPE	DSNPE	DSPP	SRC	CRC	FDDL	S-SDMF
G16/P48	61.4	63.8	63.9	63.5	63.4	71.7	63.1	73.7	70.0	74.3	74.7	75.4	71.8	**79.6**
G24/P40	63.3	70.3	75.9	70.1	70.3	78.3	74.9	78.4	77.4	78.6	71.1	78.1	83.6	**85.8**
G32/P32	67.9	74.3	79.9	74.4	76.1	79.7	77.2	78.3	82.8	82.4	72.0	79.9	**87.9**	**87.9**
Avg.	64.2	69.5	73.2	69.3	69.9	76.5	71.7	76.8	76.7	78.4	72.6	77.8	81.1	**84.4**

and the remaining part Pn (P48, P40, P32) as the testing set. The images in two face datasets are all cropped to 32 × 32, and the recognition experiments are repeated 5 times for each training-testing partition.

1) Comparing Methods: In face recognition experiments, 13 representative comparing methods, including 10 popular subspace learning methods, i.e., PCA [43], graph-based LDA [18], sparse preserving projection (SPP) [22], supervised LPP (sLPP) [29], supervised NPE (sNPE) [30], LSDA [13], NSPE [33], DSNPE [12], low-rank preserving embedding (LRPE) [25] and DSPP [34], 2 recent representation based classifiers, i.e., sparse representation classifier (SRC) [26] and collaborative representation classifier (CRC) [27], and the state-of-the-art dictionary learning method, i.e., Fisher discrimination dictionary learning (FDDL) [56], are used for comparison.

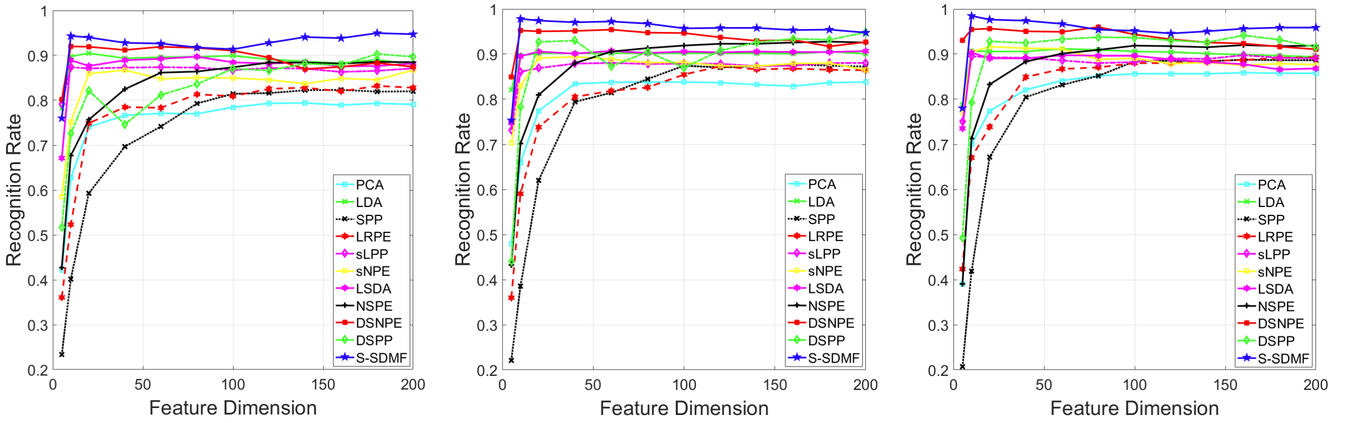
2) Parameter Settings: PCA, SPP and LRPE are unsupervised subspace learning methods, and the major parameter for adjusting is the subspace dimension. The model parameters for sLPP, sNPE, LSDA and NSPE are empirically set in accordance with [35]. For DSNPE and DSPP, the values of the trade-off parameter γ in DSNPE and ρ in DSPP are set as $\gamma = 1$ and $\rho = 0.0002$ according to the suggestions in [12] and [34], respectively. SRC and CRC both generate eigenfaces to perform dimensionality reduction, the values of the regularization parameter λ for SRC and CRC are searched from $\{0.01, 0.05, 0.1, 0.5, 1\}$. For FDDL, the parameters are chosen via cross-validation as depicted in [56]. For the proposed S-SDMF, we construct two adjacency graphs by directly using label information, so we only need to set the value of ridge regularization parameter α. For two recognition experiments, we empirically set $\alpha = 0.01$ on both datasets. It is worth noting that, to avoid the *SSS problem*, the comparing subspace learning methods, including LDA, sLPP, sNPE, LSDA, NSPE, SPP, LRPE, DSNPE and DSPP, all require an additional preprocessing step to reduce the input dimensionality (m) to the number of samples (N) by PCA.

3) Recognition Results: Table II and Table III list the top average recognition rates of S-SDMF and the comparing methods on CMU PIE and E-YaleB datasets, respectively. From the two tables, we have the following observations. First, S-SDMF consistently outperforms the subspace learning methods and representation based classifiers in all the cases we have tried. Particularly, on E-YaleB dataset, S-SDMF boosts the average recognition rates over DSPP and DSNPE by 6% and 7.7%, respectively. In addition, on CMU PIE dataset, S-SDMF also increases over 5% improvement compared with NSPE and LSDA, and nearly 8% improvement compared with sNPE and sLPP, w.r.t. the average recognition rates. Second, S-SDMF obtains comparable or even better recognition results compared to the state-of-the-art FDDL over both datasets with much less training time (refer to Subsection III-F), which confirms the effectiveness and efficiency of the proposed S-SDMF. Third, the unsupervised SPP and LRPE perform better than PCA, and even achieve similar results with some supervised subspace learning methods such as LDA and sLPP on E-YaleB dataset, again demonstrating the implicit discriminating power of the L_1-graph and LRR-graph.

Fig.5 (a)-(c) and Fig. 6 (a)-(c) show the top average recognition rates of S-SDMF and the comparing subspace learning methods versus the variation of feature dimensions on CMU PIE and E-YaleB datasets, respectively. It is clear to see that SDMF outperforms the comparing subspace learning methods almost across all the dimensions. Besides, we also observe that S-SDMF still achieves promising recognition performance even when the number of dimensions is small (e.g., less than 100). It maybe because the highly discriminant ability of the Fisher-like criterion and the sparse constraint on coefficient, which enables the new representations of S-SDMF to capture the hidden discriminative semantic concept even in the low-dimensional subspace. Moreover, the spectral regression in the S-SDMF model can also contribute to the recognition performance by reserving the original information (avoiding pre-dimensionality reduction) as well as alleviating the over-fitting in the training phase.

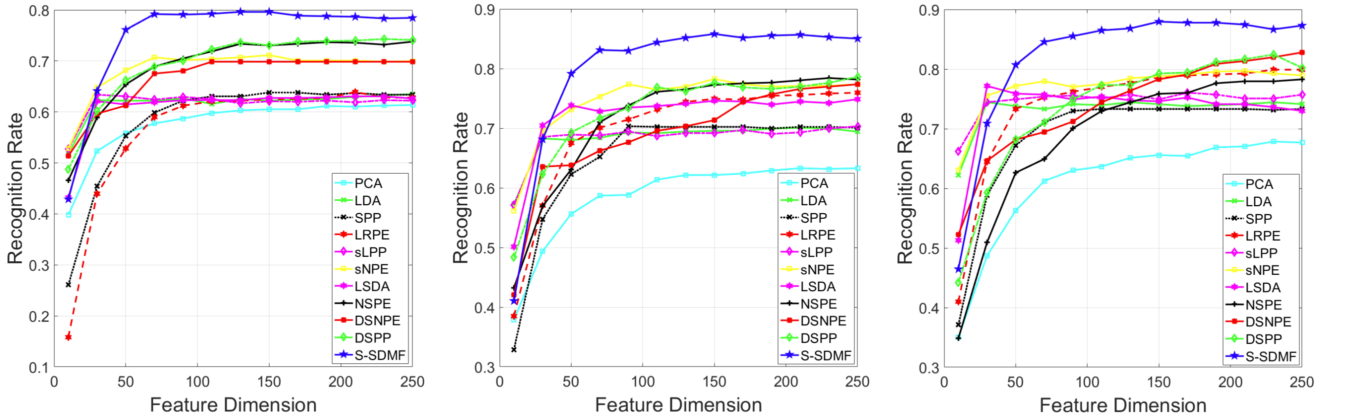
D. Image Clustering Experiment

In this subsection, we evaluate the clustering performance of U-SDMFs with k-NN and H-NN strategies, on COIL20, UMIST, JAFFE and ORL four benchmark datasets. For simplicity, the U-SDMF with k-NN strategy and U-SDMF with H-NN strategy are called U-SDMF1 and U-SDMF2, respectively. In order to randomize the experiments, we evaluate the clustering performance with different number of clusters for each evaluated dataset. For each given cluster number (except the total cluster number), 10 tests are conducted on different randomly chosen classes.



(a) (b) (c)

Fig. 5. The top average recognition rates on CMU PIE dataset versus feature dimensions from 5 to 200 of (a) G45/P125, (b) G65/P105, and (c) G85/P85, respectively.



(a) (b) (c)

Fig. 6. The top average recognition rates on E-YaleB dataset versus feature dimensions from 10 to 250 of (a) G16/P48, (b) G24/P40, and (c) G32/P32, respectively.

1) Comparing Methods: In the clustering experiment, 9 popular clustering methods including K-means (baseline), graph regularized nonnegative matrix factorization (GNMF) [11], graph regularized nonnegative matrix factorization with sparse coding (GRNMFSC) [57], Laplacian sparse coding (LapSC) [37], sparse concept coding (SCC) [41], LRR [28], Latent LRR (LatLRR) [58], the state-of-the-art sparse subspace clustering (SSC) [59] and robust graph regularized nonnegative matrix factorization (RGNMF) [60], are selected as the comparing methods.

2) Parameter Settings: As to the parameter settings, we report the matrix factorization based methods (except for K-means, LRR, LatLRR and SSC) with the number of basis vectors equal to the number of clusters. For LRR, LatLRR and SSC, the rank of adjacency graph (the number of subspaces) is also set as the number of clusters. GNMF, GRNMFSC and RGNMF are solved by multiplicative updates, the maximum iterations of the above three NMF variants are all set as 1000, and the sparsity regularization parameter λ in GRNMFSC is set as 0.01. The parameters of SCC and LapSC are set according to the suggestion in [41]. There are two parameters in U-SDMF1: the number of nearest neighbors k and the ridge regularization parameter α. We empirically set $k = 4$ and $\alpha = 0.02$ over four evaluated datasets. Regarding the U-SDMF2, in this work we use Alternating Direction Method of Multipliers (ADMM) [61] framework to solve the sparse optimization problem in Eq. (4). As a result, there are also two parameters in U-SDMF2: the balance parameter γ (actually, we scale γ with $\gamma = \frac{\widehat{\gamma}}{\min_i \max_{j\neq i} |\mathbf{x}_i^T \mathbf{x}_j|}$) and the ridge regularization parameter α. We empirically set $\widehat{\gamma} = 4$ for UMIST dataset, and $\widehat{\gamma} = 6$ for the other three datasets. In addition, the parameter α is also fixed as 0.02 in U-SDMF2.

a) Evaluation Metrics: The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the dataset. Three metrics, the accuracy (ACC), the normalized mutual information metric (NMI) and purity are used to measure the clustering performance.

i) ACC: As depicted in [62], the clustering ACC is defined as follows:

$$ACC = \frac{\Sigma_{i=1}^{N} \delta(c_i, map(l_i))}{N}, \tag{14}$$

where N is the total number of samples, c_i stands for the provided label, $map(l_i)$ is a mapping function that maps the obtained cluster label l_i to the equivalent label from the data corpus. $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise.

ii) NMI: Let C denote the set of clusters obtained from the ground truth and $\widehat{C}$ obtained from our algorithm. Their mutual information metric $MI(C, \widehat{C})$ is defined according

TABLE IV

CLUSTERING PERFORMANCE ON COIL20 DATASET

	ACC(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	83.3±12.6	89.9±14.3	90.6±10.5	95.9±7.3	82.9±12.8	91.1±10.4	85.0±11.6	83.4±12.5	94.4±13.7	**97.5±6.8**	96.1±11.8
8	72.2±12.3	86.4±6.7	87.8±8.4	91.4±6.6	76.2±13.9	81.2±8.4	80.1±11.2	73.8±5.4	90.5±9.6	89.5±10.6	**94.8±5.6**
12	70.3±6.7	79.5±6.4	81.4±9.8	78.5±4.8	70.9±7.7	79.4±5.4	73.7±4.2	72.4±7.3	84.6±6.0	83.9±5.7	**86.8±2.2**
16	62.8±3.3	78.5±5.9	82.0±3.6	84.8±4.9	67.4±2.8	77.9±4.7	70.9±4.6	67.8±3.6	82.5±6.5	84.9±5.1	**88.1±2.9**
20	64.2	77.7	79.6	83.2	67.0	76.2	70.3	64.2	84.0	84.2	**90.0**
Avg.	70.5	82.4	84.3	86.8	72.9	81.2	76.0	72.3	87.2	88.0	**91.2**

	NMI(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	74.6±14.9	84.1±19.9	88.3±10.6	93.8±7.7	74.0±17.0	88.4±12.2	80.0±12.1	75.3±10.1	95.2±9.2	96.0±10.3	**96.8±8.6**
8	72.9±9.9	87.0±5.4	90.7±6.9	90.5±5.5	75.4±13.8	85.8±6.9	82.0±9.8	74.8±1.9	**95.5±4.2**	91.1±6.1	95.2±3.0
12	74.9±5.1	84.4±3.8	87.2±5.3	85.0±4.5	76.0±9.7	83.7±4.7	78.8±4.3	76.2±9.6	93.1±5.6	90.5±3.9	**93.2±2.5**
16	72.6±2.4	85.2±2.9	86.7±2.1	90.6±2.9	75.5±2.0	84.9±2.9	76.9±4.0	74.9±3.9	93.3±4.1	91.6±2.3	**94.0±2.7**
20	74.1	85.7	87.9	89.6	76.8	83.1	78.7	73.9	93.5	93.5	**95.2**
Avg.	73.8	85.3	88.1	89.9	75.5	85.1	79.3	75.0	94.1	92.5	**94.9**

	Purity(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	82.9±12.5	88.9±9.5	91.5±8.9	97.0±4.0	84.2±10.9	93.8±6.9	85.8±12.2	81.6±11.5	96.6±13.7	97.7±6.2	**98.6±3.9**
8	74.8±8.3	87.9±7.3	90.3±6.9	92.6±4.5	78.2±13.3	89.2±5.9	82.5±8.0	73.8±11.5	94.7±5.1	92.6±5.0	**96.0±2.3**
12	73.9±6.3	84.5±3.7	88.9±4.6	86.5±4.6	74.3±10.0	82.7±5.5	74.9±5.7	73.0±5.3	94.3±4.3	91.3±4.7	**94.5±2.9**
16	70.1±3.8	83.2±3.3	87.9±2.0	90.8±3.1	69.2±4.2	83.5±5.3	72.6±3.0	74.5±4.9	94.0±3.5	91.0±1.8	**94.7±2.9**
20	67.6	83.4	85.9	89.1	71.2	79.7	76.1	70.8	94.1	94.0	**95.8**
Avg.	73.9	85.6	88.9	91.2	75.4	85.7	78.4	74.7	94.7	93.3	**95.9**

TABLE V

CLUSTERING PERFORMANCE ON UMIST DATASET

	ACC(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
5	55.6±9.6	60.4±10.6	67.6±12.6	63.9±10.2	53.7±6.1	64.9±11.2	59.6±11.2	50.3±6.1	72.0±12.5	72.8±12.8	**77.7±13.4**
10	47.4±5.2	52.8±5.6	63.9±8.9	59.3±8.3	46.8±6.4	59.8±10.7	50.1±6.2	48.4±3.6	67.0±8.6	69.5±6.5	**74.4±7.7**
15	43.5±2.6	49.9±3.4	59.6±4.9	57.9±3.2	44.1±1.9	52.5±4.1	46.8±3.6	45.7±4.0	66.7±4.2	65.5±3.0	**69.0±3.1**
20	39.0	48.2	56.7	52.8	42.9	54.6	46.7	43.6	62.7	64.3	**71.0**
Avg.	46.4	52.8	61.9	58.5	46.9	57.9	50.8	47.0	67.1	68.0	**73.0**

	NMI(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
5	51.4±12.9	58.7±11.1	70.3±11.4	62.3±11.4	49.7±5.8	64.2±11.4	57.8±9.9	43.4±11.6	72.1±13.1	73.8±12.6	**75.2±12.8**
10	56.4±5.1	65.1±4.5	77.3±5.8	71.1±7.3	57.1±5.9	71.6±7.8	60.3±4.9	57.9±4.6	77.6±7.5	78.4±5.2	**81.7±4.1**
15	60.3±2.5	68.8±2.7	76.5±3.3	72.8±3.5	61.3±1.8	68.1±3.9	61.8±1.9	59.4±3.2	76.8±2.7	**79.1±3.9**	78.1±4.8
20	59.5	71.3	75.5	73.5	62.3	69.2	64.2	62.2	78.2	78.9	**80.9**
Avg.	56.9	66.0	74.9	69.9	57.6	68.3	61.0	55.7	76.2	77.5	**79.0**

	Purity(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
5	60.3±8.2	65.9±9.4	80.8±9.0	69.5±11.5	60.2±12.0	72.3±6.6	65.2±9.0	60.8±5.5	77.8±9.5	81.6±11.6	**85.7±7.5**
10	51.8±5.9	59.4±5.5	78.4±7.9	68.9±8.9	51.3±3.8	66.4±7.8	55.4±4.7	56.2±3.7	74.6±8.8	81.0±6.3	**86.7±4.9**
15	49.2±2.6	57.7±3.4	71.6±5.1	69.1±3.6	50.4±3.5	61.8±4.5	50.7±2.6	52.3±3.4	73.5±3.3	73.4±2.0	**78.0±4.3**
20	46.1	57.2	68.9	63.3	48.4	61.6	51.7	52.6	69.3	71.0	**79.1**
Avg.	51.8	60.0	74.9	67.7	52.6	65.5	55.8	55.5	73.8	76.8	**82.4**

to [11]:

$$MI(C,\widehat{C}) = \sum_{c_i \in C, \widehat{c}_j \in \widehat{C}} p(c_i,\widehat{c}_j).log\frac{p(c_i,\widehat{c}_j)}{p(c_i)p(\widehat{c}_j)}, \quad (15)$$

where $p(c_i)$ and $p(\widehat{c}_j)$ denote the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and $\widehat{c}_j$, respectively. $p(c_i,\widehat{c}_j)$ is the joint probability that the arbitrarily selected sample belongs to the clusters c_i and $\widehat{c}_j$ at the same time. In our experiment, we use the normalized MI (NMI) metric to evaluate the clustering performance:

$$NMI(C,\widehat{C}) = \frac{MI(C,\widehat{C})}{max(H(C),H(\widehat{C}))}, \quad (16)$$

where $H(C)$ and $H(\widehat{C})$ are the entropies of C and $\widehat{C}$, respectively. NMI metric reflects the similarity of the distribution of C and $\widehat{C}$, if the two sets of clusters are identical, NMI equals to 1, otherwise NMI falls in between 0 and 1.

iii) Purity: Purity [60] measures the extent to which each cluster contains data points from primarily one class. Accordingly, the purity of a clustering is computed as follows:

$$Purity = \frac{1}{N}\sum_{i=1}^{K}\max(n_i^j), \quad (17)$$

where n_i^j is the number of data points in the jth cluster that belong to the ith class, K is the number of the clusters, and N is the total number of the data points.

b) Image Clustering Results: Tables IV-VII list the ACC, NMI and purity of different methods over four benchmark datasets, the mean and standard error of the performance are reported in the four tables.

From Tables IV-VII, we have the following observations. First, the two versions of U-SDMFs always result in inspiring clustering performance in all the cases we have tried. By simply using k-means on the low-dimensional sparse representation, U-SDMF1 and U-SDMF2 achieve better clustering results compared with the state-of-the-art SSC and RGNMF

TABLE VI

CLUSTERING PERFORMANCE ON JAFFE DATASET

	ACC(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	96.5±4.6	97.2±2.9	97.9±2.9	96.8±5.5	93.8±9.9	98.1±2.8	96.7±3.1	95.6±2.8	96.0±8.8	**98.4±3.7**	96.3±8.1
6	94.8±4.3	95.2±7.2	93.2±8.6	91.9±10.2	93.3±6.3	96.1±7.3	94.4±2.9	**97.9±1.3**	93.9±8.8	94.5±1.2	95.8±6.1
8	89.5±5.7	92.3±6.7	91.5±7.4	95.4±5.4	93.0±4.6	95.2±7.4	92.2±3.6	**98.0±1.0**	88.7±5.5	95.2±5.0	97.2±4.1
10	84.2	90.9	91.1	98.1	86.5	91.9	91.3	97.9	88.3	95.3	**98.6**
Avg.	91.2	93.9	93.4	95.6	91.7	95.3	93.7	**97.3**	91.7	95.9	97.0

	NMI(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	93.9±6.2	94.3±4.5	96.0±4.9	94.4±8.2	90.8±14.4	96.1±4.8	92.5±6.7	90.6±4.9	94.8±9.9	**97.4±5.9**	95.9±6.9
6	93.1±4.9	94.8±3.8	93.7±5.6	92.6±8.2	91.6±4.8	**96.4±5.6**	90.4±4.4	96.0±2.5	91.9±7.3	94.4±2.6	95.3±5.3
8	89.0±4.4	94.1±2.3	92.2±4.7	95.6±3.8	91.5±5.0	95.8±4.9	89.5±3.7	96.6±1.6	91.3±3.6	93.9±5.4	**97.1±2.4**
10	86.4	92.9	91.2	97.3	88.9	93.8	90.2	96.5	92.1	94.4	**98.2**
Avg.	90.6	94.0	93.3	95.0	90.7	95.5	90.7	94.9	92.5	95.0	**96.6**

	Purity(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
4	94.6±4.2	96.8±4.2	97.9±2.9	96.8±5.5	91.6±8.9	98.1±2.8	96.2±3.6	95.3±2.5	98.1±3.0	96.1±2.5	**98.7±2.6**
6	94.9±4.1	95.4±3.8	96.4±3.2	94.4±5.5	91.8±8.3	96.6±4.4	94.4±2.2	97.0±1.2	96.9±3.6	94.0±6.3	**97.0±2.8**
8	91.3±3.2	95.0±2.7	93.9±3.8	97.3±2.3	93.0±4.6	96.1±3.1	90.9±4.4	97.3±0.9	96.6±2.1	94.8±4.7	**98.5±1.0**
10	85.6	94.7	91.1	98.1	89.4	95.3	93.3	96.7	96.2	95.2	**98.6**
Avg.	91.6	95.4	94.8	96.7	91.5	96.5	93.7	96.6	96.9	95.0	**98.2**

TABLE VII

CLUSTERING PERFORMANCE ON ORL DATASET

	ACC(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
10	71.3±6.6	73.4±5.2	74.6±7.9	74.6±6.8	70.7±6.5	76.2±8.5	75.0±6.8	79.2±9.4	77.7±6.8	77.4±5.7	**80.1±8.5**
20	60.9±7.0	64.7±4.4	66.5±5.8	71.8±5.3	60.4±5.7	64.5±5.5	65.7±3.2	71.8±4.9	72.0±4.7	71.1±4.8	**72.6±7.2**
30	56.7±3.5	59.8±3.9	59.7±1.9	66.6±4.4	61.1±3.2	61.2±4.7	63.2±2.5	65.0±4.1	67.5±2.8	65.3±2.8	**68.3±5.4**
40	53.5	57.4	58.1	62.0	56.0	57.6	61.3	62.3	68.7	66.0	**69.5**
Avg.	60.6	63.8	64.7	68.8	62.1	64.9	66.3	70.6	71.4	70.0	**72.6**

	NMI(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
10	75.2±5.7	78.8±5.4	79.7±6.1	77.4±6.5	74.7±5.1	80.5±6.4	78.0±6.7	79.4±8.6	**84.0±4.4**	81.4±5.2	83.4±6.8
20	73.6±5.8	76.9±3.9	78.4±4.0	79.8±3.3	72.5±3.9	75.0±4.1	76.1±2.7	80.9±2.7	80.2±4.7	81.0±3.1	**81.1±4.4**
30	71.9±2.6	75.2±2.3	75.2±2.1	78.1±3.2	76.4±2.3	75.7±2.9	76.8±0.7	79.6±2.7	80.8±2.5	79.5±1.5	**82.5±3.6**
40	71.5	74.2	76.3	77.9	74.3	75.3	76.7	77.6	83.5	80.5	**84.6**
Avg.	73.0	76.3	77.4	78.3	74.5	76.6	76.9	79.4	82.1	80.6	**82.9**

	Purity(%)										
K	K-means	GNMF	GRNMFSC	RGNMF	LapSC	SCC	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
10	77.2±4.8	79.1±6.5	81.1±6.4	78.9±7.4	74.7±4.2	81.8±5.9	78.3±8.9	81.8±12.3	85.7±4.2	83.1±5.9	**86.5±5.9**
20	68.7±5.6	70.0±5.6	76.2±5.1	77.4±4.8	70.7±7.7	68.2±6.8	71.2±3.3	77.8±3.4	**80.1±2.9**	78.6±5.1	78.9±3.6
30	65.1±2.9	69.7±4.8	70.2±3.3	71.9±4.2	65.6±4.2	64.4±4.4	67.2±2.9	69.8±3.2	77.7±3.4	74.7±2.4	**78.4±4.9**
40	63.6	66.5	67.4	68.5	64.5	65.0	63.8	69.0	76.5	74.3	**79.0**
Avg.	68.7	71.3	73.7	74.2	68.9	69.8	70.1	74.6	80.0	77.7	**80.7**

algorithms in almost all cases. Second, although SCC shares similar optimization scheme with U-SDMFs, the performance of our methods is much better than that of SCC. For example, on UMIST dataset, U-SDMF1 has a gain over SCC by 10.1%, 9.2% and 11.3% w.r.t. the average ACC, NMI and purity, respectively. This significant improvement is because, SCC simply uses LE-graph to preserve the similarities of the neighboring image samples, while U-SDMFs leverage the Fisher-like criterion to 1) maintain the reconstruction structure of the suspected within-class candidate samples, and 2) keep the suspected between-class candidate samples distant. Consequently, U-SDMFs can better capture the discriminant information of image data than SCC. Third, U-SDMF2 performs better than U-SDMF1 over four datasets by delivering 1.1%-5.0%, 1.5%-2.4% and 2.6%-5.6% of improvements w.r.t. the average ACC, NMI and purity, respectively, which confirms the superiority of the H-NN strategy. Forth, LatLRR and LRR cannot always obtain satisfactory clustering results over four datasets. For example, LatLRR and LRR achieve goodish clustering results next to U-SDMF2 on JAFFE dataset, but perform quite poor on COIL20 and UMIST datasets. The reason can be that, although LatLRR and LRR are multi-subspace learning methods with error correction, the two methods both ignore analyzing the hidden discriminant structure of image data. As a result, they may not be amenable to deal with different types of image variations. Fifth, RGNMF performs better than GRNMFSC, GNMF and LapSC, while K-means performs the worst among the comparing methods.

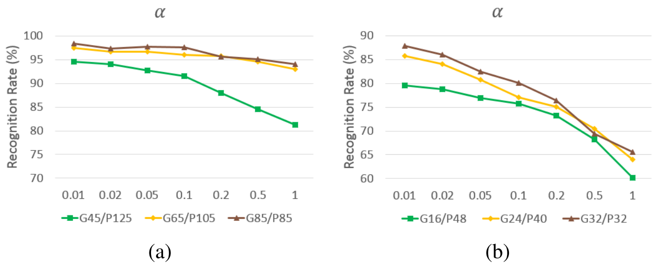


Fig. 7. Performance of S-SDMF with different α values on (a) CMU PIE and (b) E-YaleB datasets, respectively.

E. Study of Parameter Selection

This subsection further probes the effects of the parameters in S-SDMF and U-SDMFs. Fig. 7 shows the recognition results of S-SDMF as α varies from 0.01 to 1. We observe that

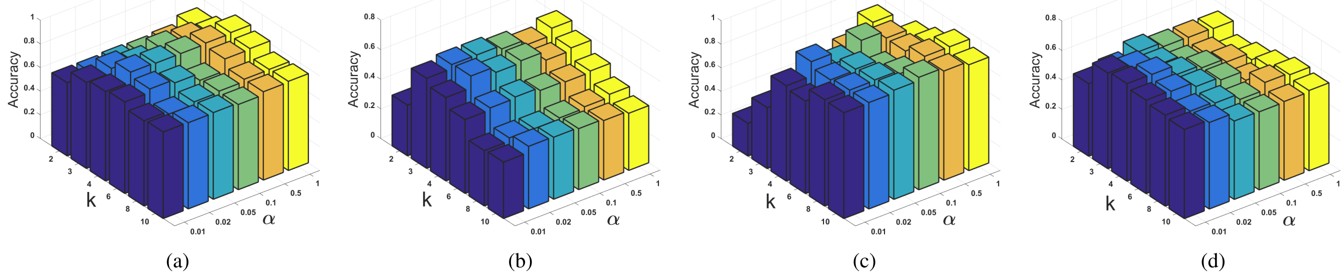


Fig. 8. Clustering accuracies of U-SDMF1 with different combinations of k and α values. (a)-(d) show the performance on COIL20, UMIST, JAFFE and ORL datasets, respectively.

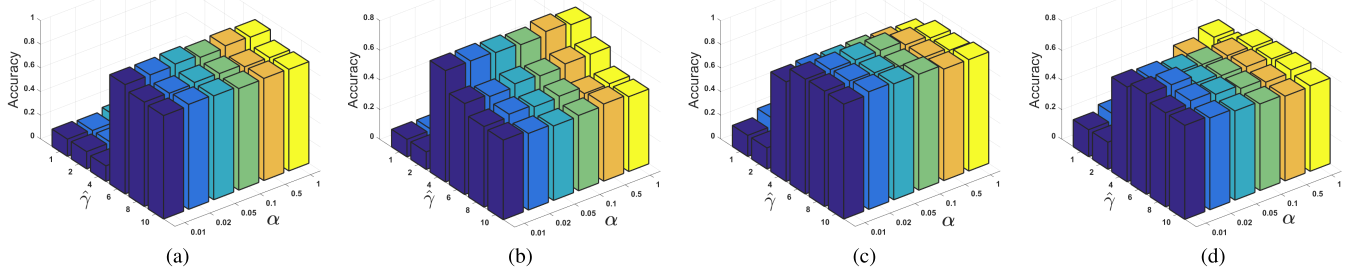


Fig. 9. Clustering accuracies of U-SDMF2 with different combinations of $\widehat{\gamma}$ and α values. (a)-(d) show the performance on COIL20, UMIST, JAFFE and ORL datasets, respectively.

TABLE VIII

THE TRAINING TIME AND THE RECOGNITION ACCURACIES OF DIFFERENT METHODS ON CMU PIE (G45/P125)

Methods	PCA	LDA	LSDA	NSPE	SPP	DSNPE	DSPP	LRPE	FDDL	S-SDMF
Training time	0.0725	0.1771	0.2080	0.4236	10.0451	10.1774	11.3125	18.9495	50.5608	0.4842
Recognition accuracy (%)	79.3	90.5	89.3	88.6	82.3	92.0	90.3	83.2	92.6	94.3

TABLE IX

THE CLUSTERING TIME AND THE PERFORMANCE OF DIFFERENT METHODS ON UMIST ($K = 20$)

Methods	K-means	SCC	GNMF	GRNMFSC	RGNMF	LRR	LatLRR	SSC	U-SDMF1	U-SDMF2
Clustering time	0.5018	3.2027	9.8771	10.3207	28.0730	11.1539	17.7744	10.7845	3.6165	9.5866
ACC (%)	39.0	54.6	48.2	56.7	52.8	46.7	43.6	62.7	64.3	71.0
NMI (%)	59.5	69.2	71.3	75.5	73.5	64.2	62.2	78.2	78.9	80.9
Purity (%)	46.1	61.6	57.2	68.9	63.3	51.7	52.6	69.3	71.0	79.1

the recognition rates of S-SDMF present a gradual decreasing tendency with the increase of the value of α, and obtain promising results ranging from 0.01-0.1 over both CMU PIE and E-YaleB datasets. Fig. 8 and Fig. 9 show the effects of two parameters: k ($\widehat{\gamma}$) and α on the clustering accuracies of U-SDMF1 and U-SDMF2 over COIL20, UMIST, JAFFE and ORL datasets. It is clear that the clustering accuracy of U-SDMF1 is sensitive to the neighborhood size k. By contrast, the clustering accuracy of U-SDMF2 is quite insensitive to the ridge regularization parameter α, and can achieve stable clustering results when the value of $\widehat{\gamma}$ exceeds some certain threshold (i.e., $\widehat{\gamma} \geq 4$). Through experiments, we also observe that the clustering performance of U-SDMFs using the other two metrics (NMI and purity) shares the similar trend with accuracy when tuning the values of the two parameters.[1]

F. Computational Time Evaluation

In this subsection, we first list the training time of S-SDMF on CMU PIE (G45/P125), then compare it with PCA, LDA, LSDA, NSPE, SPP, DSNPE, DSPP, LRPE and FDDL in Table VIII. For ease of observation, we also report their recognition performance in Table VIII. It is clear that S-SDMF achieves the best performance with a reasonable low time cost. Specifically, the training time of S-SDMF is just comparable to NSPE and even achieves order-of-magnitude speedup over the other subspace learning methods, i.e., SPP, DSPP, LRPE, DSNPE, and the dictionary learning FDDL method. Moreover, we also compute the average recognition time of a testing sample for S-SDMF as 0.0078 seconds, which is fast and far less than the acceptable 0.5 seconds.

Furthermore, we list the clustering time of U-SDMFs on UMIST ($K = 20$), then compare it with K-means, SCC, GNMF, GRNMFSC, RGNMF, SSC, LRR and LatLRR in Table IX. For reference, the clustering performance of these methods have also been reported in Table IX. As shown in Table IX, U-SDMF1 is time-efficient and reduces much time compared with the NMF variants and LRR, LatLRR and SSC. In contrast, the clustering time of U-SDMF2 is close to that of SSC, as it requires an additional step to solve the problem in Eq. (4) via ADMM framework. However, please note

[1] The results are available in https://pan.baidu.com/s/1jJI8p5W.

that our U-SDMF2 also harvests 8.3%, 2.7% and 9.8% of improvements over the state-of-the-art SSC w.r.t. the ACC, NMI and purity, respectively.

G. Scene Classification With Deep Features

In this subsection, we further evaluate the classification performance of S-SDMF with deep features on the challenging Scene15 dataset. Following the setting in [63], we use 50 images per category for training, and the rest images for testing. We employ the MatConvNet [64] toolbox, with a 21-layer VGG16 [65] pre-trained on ImageNet [66] being used. Then, we choose the vector-based feature (dimension = 1000) generated from the 20th layer as the input feature for S-SDMF. The classification accuracies of the VGG16+S-SDMF and the comparing ImageNet-VGG16, ImageNet-AlexNet [63] and ImageNet-GoogLeNet [63] are reported as **90.3%**, 86.2%, 84.1% and 85.0%, respectively. The inspiring results demonstrate the discriminating power of the deep features and offer the new direction by combining SDMF with deep features to handle the practical scene classification tasks.

IV. Concluding Remarks

In this paper, we have proposed a novel Fisher-like criterion to capture the intrinsic discriminant structure of image data by conducting discriminant analysis across both reconstruction-based graph and similarity-based graph. Furthermore, by incorporating the Fisher-like criterion with sparse coding, we further propose a new SDMF framework for efficient image representation. SDMF enables the learnt basis to capture highly discriminant information as well as the intrinsic manifold structure of original data, and meanwhile to ensure the sparseness of new representations under the learnt basis. As a result, these promising properties guarantee each image sample can be represented by a linear combination of only few key basis vectors, thus making SDMF particularly suitable for image classification and clustering. The experimental results have demonstrated the effectiveness and efficiency of SDMFs on both image classification and clustering tasks.

Please note that, SDMF framework is generic enough to incorporate various graphs for capturing highly discriminant information. In this paper, we derive SDMF model with two prevalent graphs of LLE-graph and LE-graph for simplicity. To further promote the classification and clustering performance, the LLE-graph and LE-graph can be replaced by some more complex reconstruction-based graphs and similarity-based graphs with an increase in computational cost. Hence, in practical classification and clustering applications, it is desired to design flexible combinations of graphs to balance the time, accuracy and robustness requirements, which can be a potential direction in our future study.

Appendix
The Proof of Theorem 1

Proof: Assume $rank(\mathbf{X}) = k$, then $\mathbf{X}$ can be factorized by singular vector decomposition (SVD) as follows:

$$\mathbf{X} = \widehat{\mathbf{U}}\Sigma\widehat{\mathbf{V}}^T,$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_k)$, $\widehat{\mathbf{U}} \in \Re^{m\times k}$, $\widehat{\mathbf{V}} \in \Re^{N\times k}$, and $\widehat{\mathbf{U}}^T\widehat{\mathbf{U}} = \widehat{\mathbf{V}}^T\widehat{\mathbf{V}} = \mathbf{I}$, the column vectors of $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ are linear independent. $\mathbf{y}$ is known to be in the space spanned by row vectors of $\mathbf{X}$, thus $\mathbf{y}$ can also be uniquely represented as the linear combination of the column vectors of $\widehat{\mathbf{V}}$. Suppose the combination coefficients are represented as $\mathbf{a} = [\mathbf{a}_1, \cdots, \mathbf{a}_k]^T \in \Re^k$, we have:

$$\widehat{\mathbf{V}}\mathbf{a} = \mathbf{y} \Rightarrow \widehat{\mathbf{V}}^T\widehat{\mathbf{V}}\mathbf{a} = \widehat{\mathbf{V}}^T\mathbf{y} \Rightarrow \mathbf{I}\mathbf{a} = \widehat{\mathbf{V}}^T\mathbf{y}.$$

Then, we can easily obtain $\widehat{\mathbf{V}}\widehat{\mathbf{V}}^T\mathbf{y} = \mathbf{y}$. Subsequently, we introduce the concept of pseudo inverse, which is denoted as $(.)^+$ hereinafter. We denote the pseudo inverse of $\mathbf{X}$ as $\mathbf{X}^+$, which can be computed via two different ways: $\mathbf{X}^+ = (\widehat{\mathbf{U}}\Sigma\widehat{\mathbf{V}}^T)^+ = (\widehat{\mathbf{V}}^T)^+\Sigma^+\widehat{\mathbf{U}}^+ = \widehat{\mathbf{V}}\Sigma^{-1}\widehat{\mathbf{U}}^T$ or $\mathbf{X}^+ = lim_{\alpha\to 0}(\mathbf{X}^T\mathbf{X}+\alpha\mathbf{I})^{-1}\mathbf{X}^T$.

The two ways to express $\mathbf{X}^+$ are feasible even if $\mathbf{X}^T\mathbf{X}$ is singular and $(\mathbf{X}^T\mathbf{X})^{-1}$ does not exist. Hence, when α decreases to 0, the regularized least squares solution in Eq. (11) could be rewritten as: $\mathbf{U}^* = (\mathbf{X}\mathbf{X}^T + \alpha\mathbf{I})^{-1}\mathbf{X}\mathbf{Y} = (\mathbf{X}^T)^+\mathbf{Y} = \widehat{\mathbf{U}}\Sigma^{-1}\widehat{\mathbf{V}}^T\mathbf{Y}$. Since $\widehat{\mathbf{V}}\widehat{\mathbf{V}}^T\mathbf{y} = \mathbf{y}$, we thus have

$$\begin{aligned}\mathbf{X}^T\mathbf{U}^* &= \widehat{\mathbf{V}}\Sigma\widehat{\mathbf{U}}^T\mathbf{U}^*\\ &= \widehat{\mathbf{V}}\Sigma\widehat{\mathbf{U}}^T\widehat{\mathbf{U}}\Sigma^{-1}\widehat{\mathbf{V}}^T\mathbf{Y}\\ &= \widehat{\mathbf{V}}\widehat{\mathbf{V}}^T\mathbf{Y}\\ &= \mathbf{Y}.\end{aligned}$$

□

Hence, when α decreases to zero, $\mathbf{U}^*$ obtained by Eq. (11) is the exact solution in Eq. (9).

Acknowledgment

The authors would like to express their sincere gratitude to all the reviewers and Dr. Jiawei Li for the constructive and valuable comments.

References

[1] H. Liu, G. Yang, Z. Wu, and D. Cai, "Constrained concept factorization for image representation," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1214–1224, Jul. 2014.

[2] Z. Li, J. Liu, J. Tang, and H. Lu, "Robust structured subspace learning for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2085–2098, Oct. 2015.

[3] Z. Li, J. Tang, and X. He, "Robust structured nonnegative matrix factorization for image representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1947–1960, May 2018.

[4] Y. Lu, Z. Lai, Y. Xu, X. Li, D. Zhang, and C. Yuan, "Nonnegative discriminant matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1392–1405, Jul. 2017.

[5] M. Pang, B. Wang, Y.-M. Cheung, and C. Lin, "Discriminant manifold learning via sparse coding for robust feature extraction," *IEEE Access*, vol. 5, pp. 13978–13991, 2017.

[6] Z. Li and J. Tang, "Weakly supervised deep matrix factorization for social image understanding," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 276–288, Jan. 2017.

[7] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, Apr. 2015.

[8] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2019–2032, May 2014.

[9] J. Tang *et al.*, "Tri-clustered tensor completion for social-aware image tag refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1662–1674, Aug. 2017.

[10] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, “Robust object tracking via key patch sparse representation,” *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 354–364, Feb. 2017.

[11] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[12] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji, “Discriminant sparse neighborhood preserving embedding for face recognition,” *Pattern Recog.*, vol. 45, no. 8, pp. 2884–2893, 2012.

[13] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, “Locality sensitive discriminant analysis,” in *Proc. 20th Int. Joint Conf. Artifical Intell. (IJCAI)*, 2007, pp. 708–713.

[14] N. Guan, D. Tao, Z. Luo, and B. Yuan, “Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent,” *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.

[15] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jun. 2007.

[16] X. Peng, Z. Yu, Z. Yi, and H. Tang, “Constructing the l2-graph for robust subspace learning and subspace clustering,” *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[17] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proc. Adv. Conf. Neural Inf. Process. Syst.*, vol. 14, 2002, pp. 585–591.

[18] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, “Face recognition using laplacianfaces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[19] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[20] X. He, D. Cai, S. Yan, and H.-J. Zhang, “Neighborhood preserving embedding,” in *Proc. Int. Conf. Comput. Vis. (ICCV)*, vol. 2, 2005, pp. 1208–1213.

[21] S. Yan and H. Wang, “Semi-supervised learning by sparse representation,” in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 792–801.

[22] L. Qiao, S. Chen, and X. Tan, “Sparsity preserving projections with applications to face recognition,” *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, 2010.

[23] W. Yang, Z. Wang, and C. Sun, “A collaborative representation based projections method for feature extraction,” *Pattern Recognit.*, vol. 48, no. 1, pp. 20–27, 2015.

[24] W. K. Wong, Z. Lai, J. Wen, X. Fang, and Y. Lu, “Low-rank embedding for robust image feature extraction,” *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2905–2917, Jun. 2017.

[25] Y. Zhang, M. Xiang, and B. Yang, “Low-rank preserving embedding,” *Pattern Recognit.*, vol. 70, pp. 112–125, 2017.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[27] L. Zhang, M. Yang, and X. Feng, “Sparse representation or collaborative representation: Which helps face recognition?” in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 471–478.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[29] Z. Zheng, F. Yang, W. Tan, J. Jia, and J. Yang, “Gabor feature-based face recognition using supervised locality preserving projection,” *Signal Process.*, vol. 87, no. 10, pp. 2473–2483, 2007.

[30] X. Zeng and S. Luo, “A supervised subspace learning algorithm: Supervised neighborhood preserving embedding,” in *Proc. Int. Conf. Adv. Data Mining Appl. (ADMA)*, 2007, pp. 81–88.

[31] G. Gao, J. Yang, X.-Y. Jing, F. Shen, W. Yang, and D. Yue, “Learning robust and discriminative low-rank representations for face recognition with occlusion,” *Pattern Recognit.*, vol. 66, pp. 129–143, 2017.

[32] F. Zhong, J. Zhang, and D. Li, “Discriminant locality preserving projections based on L1-norm maximization,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2065–2074, Nov. 2014.

[33] B.-H. Wang, C. Lin, X.-F. Zhao, and Z.-M. Lu, “Neighbourhood sensitive preserving embedding for pattern classification,” *IET Image Process.*, vol. 8, no. 8, pp. 489–497, Aug. 2014.

[34] Q. Gao, Y. Huang, H. Zhang, X. Hong, K. Li, and Y. Wang, “Discriminative sparsity preserving projections for image recognition,” *Pattern Recognit.*, vol. 48, no. 8, pp. 2543–2553, 2015.

[35] M. Pang, C. Lin, R. Liu, X. Fan, J. Jiang, and Z. Luo, “Sparse concept discriminant matrix factorization for image representation,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2015, pp. 1255–1259.

[36] A. Iosifidis, A. Tefas, and I. Pitas, “Minimum class variance extreme learning machine for human action recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1968–1979, Nov. 2013.

[37] S. Gao, I. W.-H. Tsang, and L.-T. Chia, “Laplacian sparse coding, hypergraph laplacian sparse coding, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.

[38] M. Zheng *et al.*, “Graph regularized sparse coding for image representation,” *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[39] S. Yi, Z. He, Y.-M. Cheung, and W.-S. Chen, “Unified sparse subspace learning via self-contained regression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2537–2550, Oct. 2018.

[40] D. Cai, X. He, and J. Han, “Spectral regression for efficient regularized subspace learning,” in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[41] D. Cai, H. Bao, and X. He, “Sparse concept coding for visual analysis,” in *Proc. CVPR*, 2011, pp. 2905–2910.

[42] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment,” *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2004.

[43] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Proc. IEEE Comput. Society Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1991, pp. 586–591.

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics), 2nd ed. New York, NY, USA, 2009.

[45] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[46] K. Hui and C. Wang, “Clustering-based locally linear embedding,” in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, 2008, pp. 1–4.

[47] D. Cai, X. He, and J. Han, “Spectral regression: A unified subspace learning framework for content-based image retrieval,” in *Proc. 15th ACM Int. Conf. Multimedia (ACM MM)*, 2007, pp. 403–412.

[48] T. Sim, S. Baker, and M. Bsat, “The CMU pose, illumination, and expression (PIE) database,” in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2002, pp. 53–58.

[49] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[50] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, “The Japanese female facial expression (JAFFE) database,” in *Proc. 3rd Int. Conf. Autom. Face Gesture Recognit. (FG)*, 1998, pp. 14–16.

[51] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142.

[52] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (COIL-20),” Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.

[53] D. B. Graham and N. M. Allinson, “Characterizing virtual eigensignatures for general purpose face recognition,” in *Face Recognition: From Theory to Applications*, vol. 163. 1998, pp. 446–456.

[54] L. Fei-Fei and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 524–531.

[55] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[56] M. Yang, L. Zhang, X. Feng, and D. Zhang, “Sparse representation based Fisher discrimination dictionary learning for image classification,” *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, 2014.

[57] C. Lin and M. Pang, “Graph regularized nonnegative matrix factorization with sparse coding,” *Math. Problems Eng.*, vol. 2015, Feb. 2015, pp. 1–11.

[58] G. Liu and S. Yan, “Latent low-rank representation,” in *Proc. Low-Rank Sparse Model. Vis. Anal.*, 2014, pp. 23–38.

[59] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[60] C. Peng, Z. Kang, Y. Hu, J. Cheng, and Q. Cheng, “Robust graph regularized nonnegative matrix factorization for clustering,” *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 3, 2017, Art. no. 33.

[61] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends. Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[62] H. Jia, Y.-M. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, May 2016.

[63] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. (2016). "Places: An image database for deep scene understanding." [Online]. Available: https://arxiv.org/abs/1610.02055

[64] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia (ACM MM)*, 2015, pp. 689–692.

[65] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

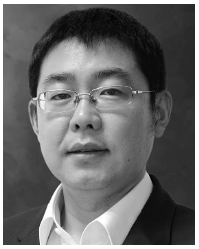
[66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

Meng Pang received the B.Sc. degree in embedded engineering and the M.Sc. degree in software engineering from the Dalian University of Technology, Dalian, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China. His research interests include image processing, pattern recognition, and data mining.

Yiu-Ming Cheung (SM'06–F'18) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. He is currently a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, pattern recognition, visual computing, and optimization. He is an IET Fellow, BCS Fellow, RSA Fellow, and IETI Distinguished Fellow. He serves as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON CYBERNETICS, PATTERN RECOGNITION.

Risheng Liu received the B.Sc. and Ph.D. degrees in mathematics from the Dalian University of Technology, Dalian, China, in 2007 and 2012, respectively. From 2010 to 2012, he was a Visiting Scholar at the Robotic Institute, Carnegie Mellon University. Since 2016, he has been a Hong Kong Scholar Research Fellow with The Hong Kong Polytechnic University, Hong Kong. He is currently an Associate Professor with the International School of Information and Software Technology, Dalian University of Technology, Dalian, China. His research interests include machine learning, optimization, computer vision, and multimedia.

Jian Lou received the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University. His research interests include statistical learning, numerical optimization and method, and privacy-preserving for machine learning.



Chuang Lin (M'14) received the M.Sc. and Ph.D. degrees in signal processing from the Harbin Institute of Technology, Harbin, China, in 2004 and 2008, respectively. He is currently an Associate Professor with the CAS Key Laboratory of Human–Machine Intelligence-Synergy Systems, Research Center for Neural Engineering, Institute of Biomedical and Health Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include biomedical signal processing, pattern recognition, and machine learning.