

Received December 20, 2019, accepted December 25, 2019, date of publication December 31, 2019, date of current version January 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963279

Convolutional Subspace Clustering Network With Block Diagonal Prior

JUNJIAN ZHANG^{ID}, (Student Member, IEEE), CHUN-GUANG LI^{ID}, (Member, IEEE),
TIANMING DU^{ID}, (Student Member, IEEE), HONGGANG ZHANG^{ID}, (Senior Member, IEEE),
AND JUN GUO^{ID}, (Member, IEEE)

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Chun-Guang Li (lichunguang@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876022, and in part by the Open Project Fund from the Ministry of Education (MOE) Key Laboratory of Machine Perception, Peking University.

ABSTRACT Standard methods of subspace clustering are based on self-expressiveness in the original data space, which states that a data point in a subspace can be expressed as a linear combination of other points. However, the real data in raw form are usually not well aligned with the linear subspace model. Therefore, it is crucial to obtain a proper feature space for performing high quality subspace clustering. Inspired by the success of Convolutional Neural Networks (CNN) for extraction powerful features from visual data and the block diagonal prior for learning a good affinity matrix from self-expression coefficients, in this paper, we propose a jointly trainable feature extraction and affinity learning framework with the block diagonal prior, termed as Convolutional Subspace Clustering Network with Block Diagonal prior (ConvSCN-BD), in which we solve the joint optimization problem in ConvSCN-BD via an alternating minimization algorithm, which updates the parameters in the convolutional modules and the self-expression coefficients with stochastic gradients descent and updates other variables with close-form solutions alternately. In addition, we derive the connection between the block diagonal prior and the subspace structured norm, and reveal that using the block diagonal prior on the affinity matrix is essentially incorporating the feedback information from spectral clustering. Experiments on three benchmark datasets demonstrated the effectiveness of our proposal.

INDEX TERMS Subspace clustering, block diagonal prior, convolutional subspace clustering, spectral clustering.

I. INTRODUCTION

In many problems across computer vision and pattern recognition, we need to deal with high-dimensional datasets, such as images, videos, text, and more. Such high-dimensional data can often be well approximated by a union of low-dimensional subspaces, corresponding to multiple classes or categories [1]. For example, under Lambertian reflectance, the face images of one subject with a fixed pose and varying lighting conditions lie in a linear subspace of dimension up to 9 [2]. Similarly, the feature point trajectories associated with a rigidly moving object in a video lie in a union of subspaces with dimension up to 3 [3], and the images of each handwritten digit with different variations also lie in a low-dimensional subspace [4]. Grouping the data points

into their respective subspace leads to an important problem, called *subspace clustering*. In the past decade, subspace clustering has found many applications in image representation and compression [5], motion segmentation [3], and temporal video segmentation [6], *etc.*

A. SUBSPACE CLUSTERING

Problem 1: (Subspace Clustering) Given the data matrix $X \in \mathbf{R}^{D \times N}$ whose columns are drawn from a union of k subspaces $\{S_i\}_{i=1}^k$, where subspace S_i contains N_i data points and $\sum_{i=1}^k N_i = N$. The goal of *subspace clustering* is to segment the columns of X into their corresponding subspaces.

In the past decade, subspace clustering has received a lot of attentions and a number of methods have been developed. Among them, the most popular methods are based on spectral clustering, e.g., [7]–[16], which divide the problem into

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy^{ID}.

two steps. In the first step, an affinity matrix $A = [a_{ij}]$ is built from the data, where a_{ij} measures the similarity of the i -th and j -th data points. It is expected that a_{ij} is nonzero only for data points lying in the same subspace (which is referred as subspace preserving property [1], [6]). In the second step, the segmentation of the data can be obtained by applying spectral clustering to the affinity matrix A .

The typical methods for building the affinity matrix are based on the self-expression model, which states that a data point in a subspace can be expressed as a linear combination of other data points [6], [8], i.e., $\mathbf{x}_j = \sum_{i \neq j} c_{ij} \mathbf{x}_i + \mathbf{e}_j$, where $\mathbf{e}_j$ is used to tolerate errors. To find the solution of subspace-preserving property, a proper regularizer is usually imposed on the coefficients, leading to the problem as follows:

$$\min_{\mathbf{c}_j} \|\mathbf{c}_j\|_\ell + \frac{\lambda}{2} \|\mathbf{x}_j - X\mathbf{c}_j\|_2^2 \quad \text{s.t. } c_{jj} = 0, \quad (1)$$

for $j = 1, \dots, N$, where $\lambda > 0$ is a tradeoff parameter, $\|\cdot\|_\ell$ is a properly chosen norm on the coefficient matrix, such as the ℓ_1 norm used in sparse representation [8], the nuclear norm used in low rank representation [9]. If we arrange the N coefficients vectors $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}$ into the coefficients matrix C , where $C = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N] \in \mathbb{R}^{N \times N}$, then we turn the problem into a compact formulation

$$\min_C \|C\|_\ell + \frac{\lambda}{2} \|X - XC\|_F^2 \quad \text{s.t. } \text{diag}(C) = 0, \quad (2)$$

where the constraint $\text{diag}(C) = 0$ is sometimes used to avoid the trivial solution $C = I$.

Once the coefficients matrix C has been learned from the data, the affinity matrix is usually induced from C , via $A = \frac{1}{2}(|C| + |C^T|)$, and spectral clustering is used to find the segmentation of the data.

B. BLOCK DIAGONAL PRIOR

Block diagonal prior [17]–[20] means that the affinity matrix learned from data has a block diagonal structure, which consists of k connected components, corresponding to data points in k subspaces. Given the data matrix $X \in \mathbb{R}^{D \times N}$ drawn from a union of k subspace $[S_1, S_2, \dots, S_k]$, the coefficient matrix C is block diagonal if C has the following form

$$C = \begin{bmatrix} C_1 & 0 & \dots & 0 \\ 0 & C_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C_k \end{bmatrix}, \quad (3)$$

where $C_i \in \mathbb{R}^{N_i \times N_i}$ and the nonzero entries in C_i correspond to a pair of data points only from the subspace S_i .

Recently, a few works incorporate the block diagonal prior into subspace clustering [17], [18], and demonstrate that the subspace clustering performance could be improved by adding the block diagonal prior on the coefficients matrix or the affinity matrix. For example, [17] seeks the coefficient matrix to be exactly k -block diagonal by enforcing a hard Laplacian constraint; [18] seeks for a block diagonal representation directly to construct the affinity matrix; in

addition, [19] incorporates the block diagonal prior into Low-Rank Representation [9] in an implicit feature space, leading to promising subspace clustering performance.

Traditional subspace clustering methods usually work in the original space, however, due to lack of feature extraction, the potential application of subspace clustering has been seriously limited. To address this problem, attempts to joint feature extraction and subspace clustering have been carried out. For example, manual features, such as, SIFT [21], HOG [22], and a two-layer auto-encoder network is used before performing subspace clustering [23]; in [24], a stacked convolutional auto-encoder network is adopted to jointly perform feature extraction for subspace clustering; in [25], a stacked convolutional auto-encoder network is used in a generative adversarial network framework for subspace clustering. While these works show that a proper feature extraction is usually crucial for subspace clustering, more efforts are needed to explore the systematic scheme of feature extraction based subspace clustering.

C. OUR CONTRIBUTIONS

In this paper, we propose a novel subspace clustering framework, called Convolutional Subspace Clustering Network with Block Diagonal prior (ConvSCN-BD), in which the block diagonal prior is integrated into a jointly trainable convolution feature based self-expression model. By doing so, the block diagonal prior helps to learn a better affinity matrix, and then the affinity matrix provides informative feedback to refine the self-expression coefficients. Specifically, our contributions can be highlighted as follows.

- We propose a jointly trainable framework for subspace clustering, in which the convolution feature extraction and the affinity learning with the block diagonal prior are jointly optimized.
- We propose an alternating minimization algorithm to solve the joint optimization problem in ConvSCN-BD, in which the parameters in convolutional feature extraction module and the coefficients in self-expression model are updated via stochastic gradient descent and the other variables are updated with closed-form solutions alternately.
- We conduct experiments on three benchmark datasets to demonstrate the effectiveness of the proposed joint optimization framework.

Paper Outline: The remainder of this paper is organized as follows. Section II reviews the relevant work. Section III presents our proposed ConvSCN-BD and the training strategy. Section IV shows experiments with discussions and Section V concludes the paper.

II. RELATED WORK

In this section, we review the relevant previous work about the block diagonal prior in subspace clustering. For clarity, we group them into two categories: a) subspace clustering with block diagonal prior in original space; b) subspace clustering with block diagonal prior in feature space.

A. SUBSPACE CLUSTERING WITH BLOCK DIAGONAL PRIOR IN ORIGINAL SPACE

Definition 1 (*k-Block Diagonal Matrix*): For any matrix $A \in \mathbf{R}^{N \times N}$, it is *k-block diagonal* if A has *k* connected components.

For a block diagonal affinity matrix A , the number of connected components relates to the spectral property of the corresponding Laplacian matrix $L_A = D - A$, where D is a diagonal degree matrix with diagonal element $D_{ii} = \sum_{j=1}^N a_{ij}$.

Theorem 1 [26]: Let A be an affinity matrix, the multiplicity number of the eigenvalue 0 of the corresponding Laplacian matrix L_A equals the number of connected components (blocks) in A .

Let $\lambda_i(L_A)$, $i = 1, \dots, N$, be the eigenvalues of L_A in decreasing order. Then with the Theorem 1, A is *k-block diagonal* if and only if $\lambda_i(L_A) > 0$ for $i = 1, \dots, N - k$ and $\lambda_i(L_A) = 0$ for $i = N - k + 1, \dots, N$.

In [17], the block diagonal prior is introduced by imposing a Laplacian Constraint (LC) $\text{rank}(L_A) = N - k$, which is added directly in the self-expression model as follows:

$$\begin{aligned} \min_C & \|C\|_\ell + \lambda \|E\|_F^2 \\ \text{s.t. } & X = XC + E, \text{diag}(C) = 0, C \in \mathcal{K}, \end{aligned} \quad (4)$$

where $\mathcal{K}$ is the space of *k*-block-diagonal matrix which is defined as

$$\mathcal{K} := \{C : \text{rank}(L_A) = N - k, A = \frac{1}{2}(|C| + |C^T|)\}. \quad (5)$$

While the explicit constraint on the rank of the graph Laplacian is a good choice to explore the block diagonal prior, it is hard to optimize due to the NP-hard nature.

To address this issue, Lu *et al.*s [18] relax the rank constraint into an ℓ_1 norm regularizer on the minor eigenvalues of the graph Laplacian matrix, which is defined as follows.

Definition 2 (*k-Block Diagonal Regularizer*): Let $A \in \mathbf{R}^{N \times N}$ be an affinity matrix computed by the coefficient matrix, e.g., $A = \frac{1}{2}(|C| + |C^T|)$, the *k*-block diagonal regularizer is defined as the sum of the *k* smallest eigenvalues of L_A , i.e.

$$\|A\|_{\square} := \sum_{i=N-k+1}^N \lambda_i(L_A), \quad (6)$$

where $\lambda_i(L_A)$ denotes the *i*-th eigenvalue of L_A in decreasing order.

Obviously, if the affinity matrix A is *k*-block diagonal, $\|A\|_{\square} = 0$. In turn, if $\|A\|_{\square} = 0$, the affinity matrix A exists *k* blocks. As $\|A\|_{\square}$ measures the block diagonal degree of affinity A , it can be used as a regularizer to learn a block diagonal representation. In [18], the block diagonal representation is formulated as the following:

$$\begin{aligned} \min_C & \|C\|_{\square} + \frac{\lambda}{2} \|X - XC\|_F^2 \\ \text{s.t. } & \text{diag}(C) = 0, \quad C \geq 0, \quad C = C^T, \end{aligned} \quad (7)$$

where $\|C\|_{\square}$ is the block diagonal regularization on the coefficient matrix, and $\lambda > 0$ is a tradeoff parameter.

B. SUBSPACE CLUSTERING WITH BLOCK DIAGONAL PRIOR IN FEATURE SPACE

Typical subspace clustering methods in the original space suffer some limitations with the basic linearity assumption, which requires that each sample can be linearly reconstructed by the other samples. However, it may not be linearly represented in many practice application. To address the issue, kernel trick is employed to yield a latent feature space for subspace clustering, e.g., polynomial kernel and Gaussian kernel are used to map the original data into high dimensional feature space implicitly, e.g., [27]–[30]. Recently, the block diagonal prior is used for subspace clustering in the latent space, which is implicitly built via kernel trick. For example, [19] develops a low-rank representation based approach with the block diagonal prior in the implicit feature space, which defines the optimization problem as following:

$$\begin{aligned} \min_C & \|A\|_{\square} + \gamma \|C\|_* + \frac{\lambda}{2} \|\Phi(X) - \Phi(X)C\|_{2,1} \\ \text{s.t. } & A = \frac{1}{2}(|C| + |C^T|), \end{aligned} \quad (8)$$

where $\lambda > 0$, and $\gamma > 0$ are trade-off parameters, Φ is a chosen kernel function, such as polynomial or Gaussian kernel, $\|A\|_{\square}$ is the block diagonal regularizer term, which encourages the affinity matrix $A = \frac{1}{2}(|C| + |C^T|)$ to be *k*-block diagonal, and $\|\cdot\|_*$ denotes the nuclear norm (i.e., the sum of the singular values of the matrix).

While the aforementioned subspace clustering methods in the latent feature space bring promising performance improvements, it is not guaranteed that the data in the pre-defined kernels induced latent feature space will align with the union of subspaces model. In addition, choosing a proper kernel function is not trivial.

Recently, a few attempts have explored subspace clustering with deep neural networks to handle the issue that linear subspace clustering methods was suffering in the original space. For example, in [23], hand-crafted features (e.g., SIFT or HOG features) and sparse self-expression model was combined by a fully connected deep auto-encoder network; in [31], a non-linear subspace clustering (NSC) method is proposed for image clustering via a neural network; similarly in [32], a trained Self-Organizing Map (SOM) neural network is proposed for subspace clustering; in [24], Deep Subspace Clustering Networks (DSC-Nets) are introduced to tackle the non-linearity in subspace clustering, where the raw data are non-linearly mapped with convolutional auto-encoders into a feature space in which the self-expression model is conducted; in [25], a deep adversarial framework is adopted for subspace clustering. Although these methods have reported impressive performance improvements, the diagonal block prior has not been considered.

In this work, we incorporate the block diagonal prior into DSC-Nets [24] and develops a trainable framework for joint

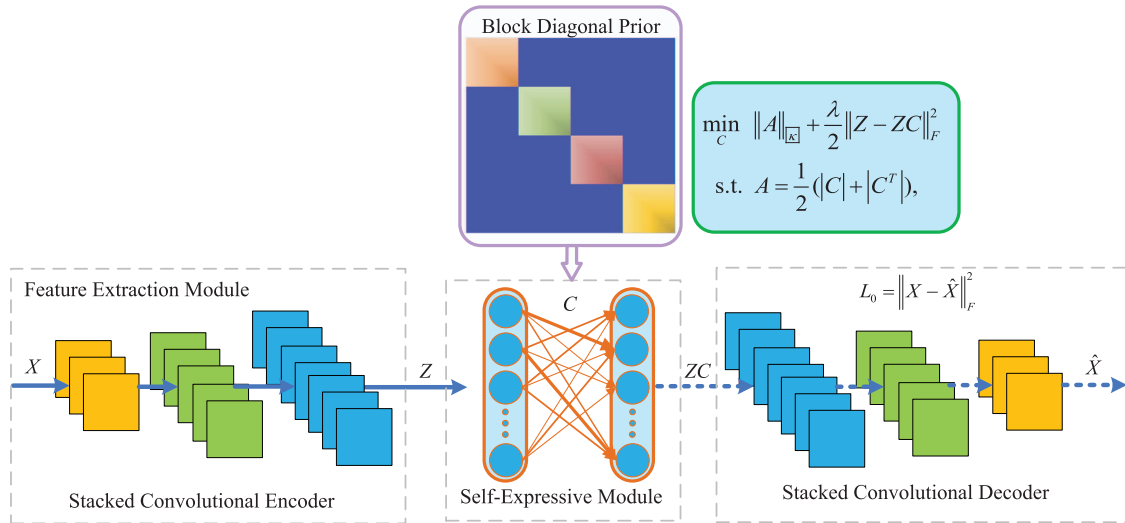


FIGURE 1. Architecture of the proposed Convolutional Subspace Clustering Network with Block Diagonal prior (ConvSCN-BD). It consists of mainly three modules: i) stacked convolutional encoder module, which is used to extract convolutional features; ii) stacked convolutional decoder module, which is used with the encoder module to initialize the convolutional module; iii) self-expression module with block diagonal prior, which is used to learn a better affinity matrix from the self-expression coefficients.

convolution feature extraction and affinity learning with the block diagonal prior. In the training, we make use of the block diagonal prior to help learning a better affinity from the self-expression coefficients and then provide a feedback information to help the self-expression model.

III. OUR PROPOSAL: CONVOLUTIONAL SUBSPACE CLUSTERING NETWORK WITH BLOCK DIAGONAL PRIOR (CONVSCN-BD)

In this section, we present our proposal ConvSCN-BD for subspace clustering, which jointly learns the convolutional feature and the affinity matrix.

A. NETWORK FORMULATION

As illustrated in Figure 1, our network consists of a convolution feature extraction module and a self-expression module with block diagonal prior. Once the affinity is constructed, the segmentation of the data can be obtained by applying spectral clustering.

1) FEATURE EXTRACTION MODULE

Feature extraction module is a basic component of our proposed ConvSCN-BD, which is used to extract features from raw data that are suitable to subspace clustering. To handle the non-linearity in subspace clustering, we employ convolutional auto-encoders which is comprised of multiple convolutional layers.

Let $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ denote the set of input samples, and $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$ denote the set of corresponding latent representations learned by the feature extraction module. For clarity, we denote the parameters in the convolutional feature extraction module as Θ and thus the learned convolutional feature $z_j = \varphi(x_j, \Theta)$, where $\varphi(\cdot)$ refers to the mapping

from the input data to the convolution feature. To train the convolution feature extraction module, we form a convolution auto-encoder architecture as in [24] and pre-train the network by minimizing the following reconstruction loss:

$$\mathcal{L}_0 = \frac{1}{2} \sum_{j=1}^N \|x_j - \hat{x}_j\|_2^2 = \frac{1}{2} \|X - \hat{X}\|_F^2, \quad (9)$$

where reconstruction $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ is the output of convolutional decoders, and N is the number of images in the training set.

Once the convolution feature extraction module is trained, we add the self-expression model with the block diagonal prior on the top of the feature which is concatenated by the feature maps in the output layer of the convolutional encoder, as illustrated in Figure 1.

2) SELF-EXPRESSION MODULE IN FEATURE SPACE WITH BLOCK DIAGONAL PRIOR

As mentioned above, self-expressiveness property is the foundation of subspace clustering, that is, a data point can be represented as the linear combination of all the other points. In this paper, we propose to seed the block diagonal representation in the convolutional feature space by solving the following optimization problem:

$$\begin{aligned} \min_C \quad & \|A\|_K + \frac{\lambda}{2} \|Z - ZC\|_F^2 \\ \text{s.t.} \quad & A = \frac{1}{2} (|C| + |C^T|), \end{aligned} \quad (10)$$

where $Z = \varphi(x_j, \Theta)$ is the matrix of concatenated convolution features, $\lambda > 0$ is a tradeoff parameter, the second term $\|Z - ZC\|_F^2$ is to tolerate errors, and $\|A\|_K$ is the block diagonal regularizer in Definition 2.

Regarding to the optimization with the block diagonal prior term $\|A\|_{\square}$, we have the following result.

Theorem 2 [18]: Let $\lambda_i(\cdot)$ denote the i -th eigenvalues of a matrix in decreasing order, $A \in \mathbf{R}^{N \times N}$ is an affinity matrix and $L_A = D - A$ is the Laplacian matrix, then we have:

$$\begin{aligned} \|A\|_{\square} &= \sum_{i=N-k+1}^N \lambda_i(L_A) \\ &= \min_W \langle L_A, W \rangle \text{ s.t. } 0 \leq W \leq I, \text{ trace}(W) = k. \end{aligned} \quad (11)$$

To understand the block diagonal prior $\|A\|_{\square}$, we denote $W = QQ^T$, where $Q \in \mathbf{R}^{N \times k}$ and the columns of Q are of unit ℓ_2 norm and orthogonal to each other, i.e., $Q^T Q = I$. It is easy to check that $W = QQ^T$ is a feasible solution. Then, we reformulate (11) as follows:

$$\begin{aligned} \|A\|_{\square} &= \min_W \langle L_A, W \rangle \text{ s.t. } 0 \leq W \leq I, \text{ trace}(W) = k \\ &= \min_{Q \in \mathbf{R}^{N \times k}} \text{trace}\{\text{diag}(A1) - A, QQ^T\} \text{ s.t. } Q^T Q = I \\ &= \min_{Q \in \mathbf{R}^{N \times k}} \text{trace}\{Q^T (\text{diag}(A1) - A) Q\} \text{ s.t. } Q^T Q = I \\ &= \min_{Q \in \mathbf{R}^{N \times k}} \sum_{i,j} a_{ij} \frac{1}{2} \|q^{(i)} - q^{(j)}\|_2^2 \text{ s.t. } Q^T Q = I, \end{aligned} \quad (12)$$

where $q^{(i)}$ and $q^{(j)}$ are the row vectors of matrix Q .

On the other hand, given the affinity matrix $A = \frac{1}{2}(|C| + |C^T|)$, we apply spectral clustering to solve the following problem:

$$\min_Q \sum_{i,j} a_{ij} \frac{1}{2} \|q^{(i)} - q^{(j)}\|_2^2 \text{ s.t. } Q \in \mathcal{Q}, \quad (13)$$

where $\mathcal{Q}$ is the segmentation matrix, and $\mathcal{Q} = \{Q \in \{0, 1\}^{N \times N} : \mathbf{1}^T Q = \mathbf{1}^T, \text{ and } \text{rank}(Q) = k\}$ is a set of all valid segmentation matrices. In practice, as the search over all $Q \in \mathcal{Q}$ is combinatorial, to make the problem tractable, we usually relax the constraint $Q \in \mathcal{Q}$ to $Q^T Q = I$ and thus solve the problem as follows:

$$\min_{Q \in \mathbf{R}^{N \times k}} \text{trace}(Q^T L_A Q) \text{ s.t. } Q^T Q = I. \quad (14)$$

which can be solved effectively by eigenvalue decomposition. It is now clear that the optimal solution of Q in (12) is the segmentation matrix of spectral clustering in (14). Moreover, by instituting $A = \frac{1}{2}(|C| + |C^T|)$ into the objective function in (14), we have that $\text{trace}(Q^T L_A Q) = \|C\|_Q$, which links the block diagonal prior $\|A\|_{\square}$ to the subspace structured norm $\|C\|_Q$ in [14].

B. ALTERNATING MINIMIZATION ALGORITHM FOR SOLVING THE JOINT OPTIMIZATION PROBLEM IN CONVSCN-BD

In this subsection, we present an alternating minimization algorithm to solve the joint optimization problem in ConvSCN-BD.

For clarity, we formulate the whole joint optimization problem in ConvSCN-BD (as illustrated in Figure 1) as follows:

$$\begin{aligned} \min_{C, \Theta} & \|A\|_{\square} + \frac{\lambda}{2} \|Z - ZC\|_F^2 + \frac{1}{2} \|X - \hat{X}\|_F^2 \\ \text{s.t. } & A = \frac{1}{2}(|C| + |C^T|), \quad Z = \varphi(X, \Theta), \end{aligned} \quad (15)$$

where Θ refers to the parameters in the convolution feature extraction module.

To remedy the difficulty of solving the problem, we introduce an auxiliary variable J with a penalty term $\frac{\beta}{2} \|C - J\|_F^2$, where $\beta > 0$, and thus relax the problem as the following:

$$\begin{aligned} \min_{C, J, \Theta} & \|A\|_{\square} + \frac{\beta}{2} \|C - J\|_F^2 + \frac{\lambda}{2} \|\varphi(X, \Theta) - \varphi(X, \Theta)C\|_F^2 \\ & + \frac{1}{2} \|X - \hat{X}\|_F^2 \text{ s.t. } A = \frac{1}{2}(|J| + |J^T|), \end{aligned} \quad (16)$$

where Z is replaced by $\varphi(X, \Theta)$ for clarity.

Owing to the intermediate term $\frac{\beta}{2} \|C - J\|_F^2$, we can solve the joint optimization problem in (16) by alternately solving the following two subproblems:

- Given J , we solve C and Θ from the subproblem

$$\begin{aligned} \min_{C, \Theta} & \frac{\beta}{2} \|C - J\|_F^2 + \frac{\lambda}{2} \|\varphi(X, \Theta) - \varphi(X, \Theta)C\|_F^2 \\ & + \frac{1}{2} \|X - \hat{X}\|_F^2. \end{aligned} \quad (17)$$

This problem can be solved via stochastic gradient descent (SGD) [33].

- Given Θ and C , we solve J from the subproblem

$$\begin{aligned} \min_J & \|A\|_{\square} + \frac{\beta}{2} \|C - J\|_F^2 \\ \text{s.t. } & A = \frac{1}{2}(|J| + |J^T|). \end{aligned} \quad (18)$$

The solution to this problem will be given in the following subsection.

1) UPDATING FOR Θ AND C

When J is given (or initialized), we can update Θ and C by solving the subproblem (17).

Note that the objective function in (17) is differentiable with respect to both Θ and C , and that with the help of pre-training the convolutional module, we are assumed to have a good initialization for Θ . Then, we update Θ and C with stochastic gradient descent [33].

2) UPDATING FOR J

When C is given (or initialized), we can update J by solving the subproblem (18). By using Theorem 2, we reformulate subproblem (18) as follows:

$$\begin{aligned} \min_{J, W} & \langle L_A, W \rangle + \frac{\beta}{2} \|C - J\|_F^2 \\ \text{s.t. } & A = \frac{1}{2}(|J| + |J^T|), \quad 0 \leq W \leq I, \text{ trace}(W) = k. \end{aligned} \quad (19)$$

- Given J (and C), the problem for solving W reduces to

$$\begin{aligned} \min_W \langle L_A, W \rangle \\ \text{s.t. } A = \frac{1}{2}(|J| + |J^\top|), \quad 0 \preceq W \preceq I, \quad \text{trace}(W) = k. \end{aligned} \quad (20)$$

As discussed in (12), $W = QQ^\top$ is a feasible solution, where Q is defined as in (12), and thus the optimal solution of W can be indirectly computed via Q , which is the optimal solution of the following problem:

$$\min_{Q \in \mathbb{R}^{N \times k}} \text{trace}(Q^\top L_A Q) \quad \text{s.t. } Q^\top Q = I, \quad (21)$$

The closed form solution of Q is the ending k eigenvectors associated with the k smallest eigenvalues of the graph Lagrangian L_A .

- Given W , the problem for solving J is:

$$\begin{aligned} \min_J \langle L_A, W \rangle + \frac{\beta}{2} \|C - J\|_F^2 \\ \text{s.t. } A = \frac{1}{2}(|J| + |J^\top|). \end{aligned} \quad (22)$$

Noted that except the intermediate term, other terms in objective of (22) only contains $|J|$. Furthermore, the elements of J have the same sign as in C . Then the solution of (22) can be computed as $J = \hat{J} \odot \text{sign}(C)$, where $\odot$ is the Hadamard product and $\hat{J}$ is the solution of the following problem:

$$\begin{aligned} \min_{\hat{J}} \langle L_A, W \rangle + \frac{\beta}{2} \|C - \hat{J}\|_F^2 \\ \text{s.t. } A = \frac{1}{2}(\hat{J} + \hat{J}^\top), \quad \hat{J} \geq 0, \end{aligned} \quad (23)$$

which is equivalent to:

$$\begin{aligned} \min_{\hat{J}} \frac{1}{2} \|\hat{J} - |C|\|_F^2 + \frac{1}{2\beta} \|D_W + D_W^\top\|_F^2 \\ \text{s.t. } \hat{J} \geq 0, \end{aligned} \quad (24)$$

where $D_W = \text{diag}(W)\mathbf{1}^\top - W$. As derived in [19], the closed-form solution of (24) is given as: $\hat{J} = (|C| - \frac{1}{2\beta}(D_W + D_W^\top))_+$.

C. TRAINING PROCEDURE FOR CONVSCN-BD

In this subsection, we provide a two-stage training procedure for the proposed ConvSCN-BD. In the first stage, we pre-train the stacked convolutional layers to initialize our ConvSCN-BD, and in the second stage, we fine-tune the whole network to jointly optimize the convolutional feature extraction and the affinity learning with the block diagonal prior.

- In the pre-training stage, we use the loss $\mathcal{L}_0$ as defined in (9) and pre-train the stacked convolutional autoencoder network at first. Then, we add the self-expression module and pre-train the stacked convolutional autoencoder

Algorithm 1 Procedure for Training ConvSCN-BD

Require: Input data, tradeoff parameters, maximum iteration $T_{\max}$, T_0 , and $t=1$.

- 1) Pre-train the convolutional module.
- 2) Pre-train the convolutional module with the self-expression module and initialize J with C .
- 3) **while** $t \leq T_{\max}$ **do**
 - a) Fixed $J^{(t)}$ and $C^{(t)}$, update $W^{(t)}$ by the solution of (21).
 - b) Fixed $W^{(t)}$, update $J^{(t)}$ by the solution of (24).
 - c) Update C and Θ by minimizing loss (27) via SGD with back propagation for T_0 times, and set $t \leftarrow t+1$.
- 4) **end while**
- 5) Run spectral clustering.

Ensure: trained ConvSCN-BD and Q .

network with the self-expression module in which the loss function is $\mathcal{L}_0 + \lambda \mathcal{L}_1$ where $\mathcal{L}_1$ is defined as

$$\begin{aligned} \mathcal{L}_1 &= \frac{1}{2} \|Z - ZC\|_F^2 \\ &= \frac{1}{2} \|\varphi(X, \Theta) - \varphi(X, \Theta)C\|_F^2. \end{aligned} \quad (25)$$

- In the fine-tuning stage, we use the loss function

$$\mathcal{L}' = \mathcal{L}_0 + \lambda \mathcal{L}_1 + \beta \mathcal{L}_2 \quad (26)$$

where $\mathcal{L}_2 = \frac{1}{2} \|C - J\|_F^2$ to jointly update the parameters Θ in the convolutional module and the coefficients C in self-expression module where J is initialized as the pre-trained coefficients matrix C . Then, given C , we solve the affinity matrix A immediately by updating W and J in subproblem (19) with the close-form solutions.

To summarize, the total loss function of our proposed ConvSCN-BD is as follows:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_1 + \beta \mathcal{L}_2 + \mathcal{L}_3, \quad (27)$$

where $\mathcal{L}_3 = \|A\|_{\text{KL}}$. For clarity, we list the procedure for training the whole ConvSCN-BD in Algorithm 1.

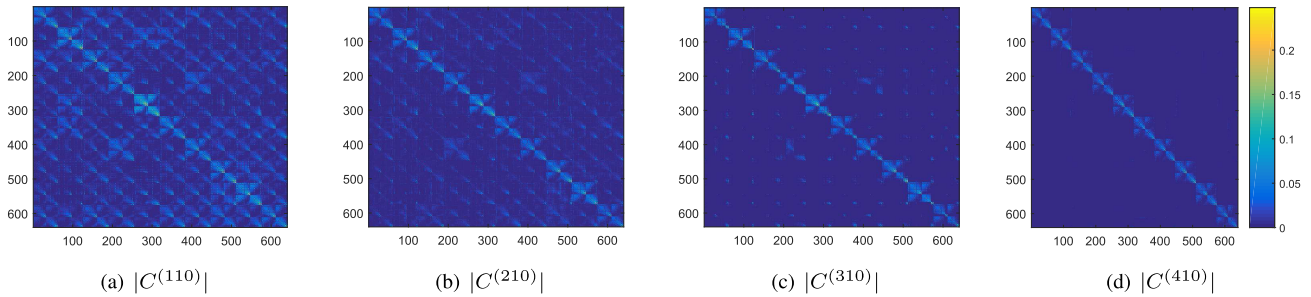
IV. EXPERIMENTAL EVALUATIONS

To evaluate the performance of our proposed ConvSCN-BD, we conduct experiments on two face benchmark datasets: the Extended Yale B and ORL, and one handwritten digit dataset: MNIST.

As baseline, we select the following algorithms, including subspace clustering in the original space: Low Rank Representation (LRR) [9], Low Rank Subspace Clustering (LRSC) [34], Sparse Subspace Clustering (SSC) [8], Efficient Dense Subspace Clustering (EDSC) [35], subspace clustering in the latent space: Kernel Sparse Subspace Clustering

TABLE 1. Clustering results (%) of different methods on extended Yale B. The best results are in bold and the second best results are underlined.

Methods	LRR	LRSC	SSC	AE+ SSC	KSSC	EDSC	AE+ EDSC	BDR	DSCNet- ℓ_1	DSCNet- ℓ_2	ConvSCN-BD
10 subjects											
Mean	22.22	30.95	10.22	17.06	14.49	5.64	5.46	2.95	2.23	1.59	1.48
Median	23.49	29.38	11.09	17.75	15.78	5.47	6.09	2.81	2.03	1.25	<u>1.25</u>
15 subjects											
Mean	23.22	31.47	13.13	18.65	16.22	7.63	6.70	6.12	2.17	1.69	1.32
Median	23.49	31.64	13.40	17.76	17.34	6.41	5.52	6.04	2.03	1.72	1.25
20 subjects											
Mean	30.23	28.76	19.75	18.23	16.55	9.30	7.67	7.54	2.17	1.73	1.40
Median	29.30	28.91	21.17	16.80	17.34	10.31	6.56	7.24	2.11	1.80	1.41
25 subjects											
Mean	27.92	27.81	26.22	18.72	18.56	10.67	10.27	8.125	2.53	1.75	1.61
Median	28.13	26.81	26.66	17.88	18.03	10.84	10.22	8.11	2.19	1.81	1.72
30 subjects											
Mean	37.98	30.64	28.76	19.99	20.49	11.24	11.56	9.22	2.63	2.07	1.98
Median	36.82	30.31	28.59	20.00	20.94	11.09	10.36	9.02	2.81	2.19	1.83
35 subjects											
Mean	41.85	31.35	28.55	22.13	26.07	13.10	13.28	18.17	3.09	2.65	1.86
Median	41.81	31.74	29.04	21.74	25.92	13.10	13.21	17.87	3.10	2.64	1.92
38 subjects											
Mean	34.87	29.89	27.51	25.33	27.75	11.64	12.66	19.56	3.33	2.67	2.42
Median	34.87	29.89	27.51	25.33	27.75	11.64	12.66	19.56	3.33	2.67	2.42

**FIGURE 2. The coefficients matrix $|C^{(t)}|$ during the training period on Extended Yale B ($n = 10$). At the beginning of training, $|C^{(110)}|$ (i.e., $t = 110$, has a rough block diagonal as shown in subfigure (a). From subfigure (a) to (d), we observe that the block diagonal structure in $|C|$ is getting clearer during iteration. In subfigure (d), we show $|C^{(410)}|$ which has a clear block diagonal structure.**

(KSSC) [28], SSC with the pre-trained convolutional auto-encoder features (AE+SSC), EDSC with the pre-trained convolutional auto-encoder features (AE+EDSC), Subspace Clustering by Block Diagonal Representation (BDR) [18], and Deep Subspace Clustering Networks (DSCNet) [24]. The results of these algorithms are reported in Table 1 and 2.

For setting of the convolutional layers, the stride of kernel is 2 in both horizontal and vertical directions, and non-linear activation is the Rectified Linear Unit (ReLU). In addition, we set the learning rate to 1.0×10^{-3} in our all experiments.

A. EXPERIMENTS ON EXTENDED YALE B

The Extended Yale B dataset [36] consists of 38 subjects, 2432 images in total. With approximately 64 frontal face images per subject taken under varying lighting conditions, the samples span a linear subspace of dimension up to nine [2]. In this experiment, we use a three convolutional auto-encoder and a three deconvolutional auto-decoder, with a self-expressive layer for the block diagonal affinity learning. The convolutional kernel sizes are 5-3-3-3-5 and channels are 10-20-30-30-20-10. Besides, each face image is down-sampled from 192×168 to 48×42 pixels, and experiments are conducted using all choices of

$n = \{10, 15, 20, 25, 30, 35, 38\}$. For the detailed setting on Extended Yale B, we set $\lambda = 1.0 \times 10^{-3}$, $\beta = 36$ for the intermediate term $\|C - J\|_F^2$, $T_0 = 10$ and $T_{max} = 10 + 40n$, where n is the number of cluster.

The experiment results are presented in Table 1, with $n \in \{10, 15, 20, 25, 30, 35, 38\}$, our ConvSCN-BD reaches the lowest clustering error than other baseline methods. In particular, as shown in Figure 2, with the training going on, the coefficient matrix presents clearer the block diagonal structure.

To show the convergence behavior, we list the clustering error and each part of cost function during training period on Extended Yale B with $n = 10$. As shown in Figure 3, the clustering error and the cost $\mathcal{L}$, $\mathcal{L}_0$, $\mathcal{L}_1$, and $\mathcal{L}_2$ are decreased rapidly. In subfigure (d), there is a dramatic gap at the epoch when the clustering error is going to stable, which suggest a feasible way to stop the training of our ConvSCN-BD.

To evaluate the sensitivity of the performance with the used tradeoff parameters, we conduct experiments on Extended Yale B to evaluate the performance of ConvSCN-BD under varying the parameters, and show the clustering accuracy with respect to the tradeoff parameter λ and β , respectively,

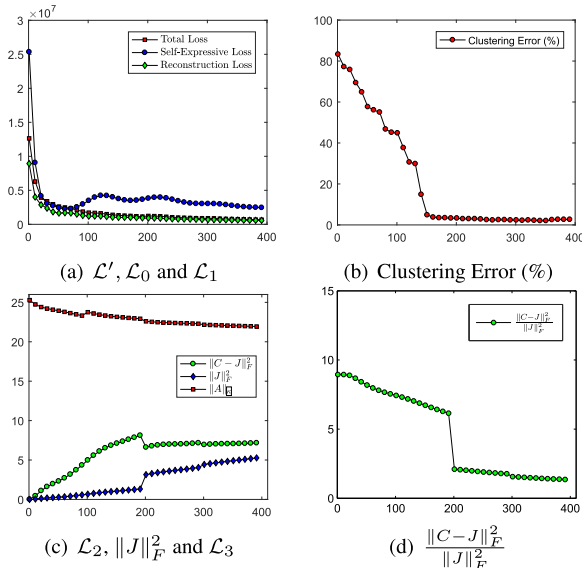


FIGURE 3. The cost functions and clustering error of ConvSCN-BD during training period on Extended Yale B ($n = 10$).

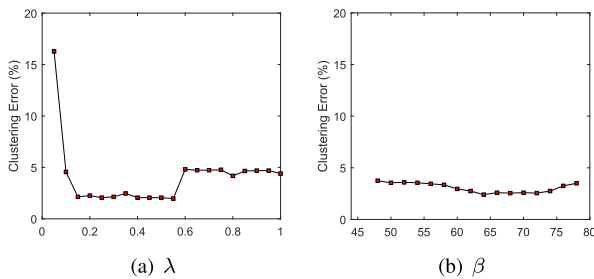


FIGURE 4. Parameters λ and β tuning of ConvSCN-BD on Extended Yale B ($n = 38$).

in Figure 4. By setting λ in $0.2 - 0.5$ and β in $65 - 67$, our proposed ConvSCN-BD yields satisfactory performance.

B. EXPERIMENTS ON ORL

The ORL data set [37] is more challenging for subspace clustering than the Extended Yale B dataset, which consists of 40 subjects, 400 face images totally, where each subjects having 10 face images under varying lighting conditions, with different facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses).

In the experiment, each original face image is down-sampled from 112×92 to 32×32 . While the kernel size in convolutional layers is reduced to 3×3 and the channels are 3-3-5-5-3-3. For the tradeoff parameters, we set $\lambda = 1$ and $\beta = 64$ for our ConvSCN-BD. In the fine-tuning stage, we set $T_0 = 10$ and $T_{max} = 940$.

Experiment results are listed in Table 2. Due to the non-linearity and small sample size per subject, the traditional linear subspace methods are suffering some limitation when meet such issue. Benefiting both the non-linear mapping and the powerful feature extraction of neural networks, methods based on convolutional layers obtain a better performance, and our ConvSCN-BD yield the best results again.

TABLE 2. Clustering error (%) on ORL and MNIST.

Methods	ORL	MNIST
LRR	38.25	44.82
LRSC	32.50	44.27
SSC	32.50	55.95
AE+SSC	26.75	42.26
KSSC	34.25	41.52
EDSC	27.25	36.10
AE+EDSC	26.25	35.26
BDR	27.25	43.54
DSC- ℓ_1	14.25	33.28
DSC- ℓ_2	14.00	32.06
ConvSCN-BD	12.50	25.50

C. EXPERIMENTS ON MNIST

The MNIST dataset contains grey scale images of handwritten digits 0 ~ 9, totally 70000 images of size 28×28 . Since methods based on spectral clustering, such as SSC, LRR, KSSC, DSC-Net, can not apply on the whole dataset, we only use 10000 samples to evaluate the performance, which each subject has 1000 randomly sampled images. For the network, we use a six convolutional layers for feature extraction and a self-expressive layer for affinity learning. The convolutional kernel sizes are 5-3-3-3-5 and channels are 10-20-30-30-20-10. For the hyper parameters, we set $\lambda = 1$, $\beta = 16$ for the intermediate term $\|C - J\|_F^2$, $T_0 = 10$ and $T_{max} = 300$.

We report the clustering result in Table 2. Methods based on the convolutional neural networks, such as DSC-Net and our ConvSCN-BD, outperforms the methods using the original samples. As can be read, our ConvSCN-BD boosts the performance through incorporating the block diagonal prior and feature extraction for subspace clustering, and achieves the best results.

V. CONCLUSION

We have presented a trainable framework, called Convolutional Subspace Clustering Network with Block Diagonal prior (ConvSCN-BD), in which the convolutional feature extraction and affinity learning are jointly optimized with the help of the block diagonal prior. Specifically, we solved the joint optimization problem by an alternating minimization algorithm that alternately update the parameters in the convolutional module, self-expression model, and the affinity matrix. By exploiting the block diagonal prior, we can learn better affinity matrix from the self-expression coefficients and also feedback useful information from spectral clustering to the self-expression model. We conducted experiments on three benchmark image datasets and verified the effectiveness of our proposal.

As the future work, we will explore more sophisticated strategy to exploit the feedback information from spectral clustering, more efficient way to solve the optimization problem, and more applications for the proposed approach.

REFERENCES

- [1] 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.
- [2] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Nov. 2003, pp. 11–18.
- [3] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1832–1845, Oct. 2010.
- [4] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Stat. Sci.*, vol. 13, no. 1, pp. 54–65, Feb. 1998.
- [5] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3655–3671, Dec. 2006.
- [6] R. Vidal, Y. Ma, and S. Sastry, *Generalized Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2016.
- [7] G. Chen and G. Lerman, "Spectral curvature clustering (SCC)," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 317–330, Mar. 2009.
- [8] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2790–2797.
- [9] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 663–670.
- [10] 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.
- [11] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 347–360.
- [12] Y. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 64–72.
- [13] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace lasso," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1345–1352.
- [14] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 277–286.
- [15] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3918–3927.
- [16] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3928–3937.
- [17] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014.
- [18] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.
- [19] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 477–489, Jan. 2018.
- [20] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 304–313, Jan. 2018.
- [21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005.
- [23] X. Peng, S. Xiao, J. Feng, W. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1925–1931.
- [24] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Neural Inf. Process. Syst.*, 2017.
- [25] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1596–1604.
- [26] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997.
- [27] V. M. Patel, H. V. Nguyen, and R. Vidal, "Latent space sparse and low-rank subspace clustering," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 691–701, Jun. 2015.
- [28] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2849–2853.
- [29] H. Nguyen, W. Yang, F. Shen, and C. Sun, "Kernel low-rank representation for face recognition," *Neurocomputing*, vol. 155, pp. 32–42, May 2015.
- [30] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, Nov. 2016.
- [31] W. Zhu, J. Lu, and J. Zhou, "Nonlinear subspace clustering for image clustering," *Pattern Recognit. Lett.*, vol. 107, pp. 131–136, May 2018.
- [32] J. Tian and M. Gu, "Subspace clustering based on self-organizing map," in *Proc. Int. Conf. Ind. Eng. Eng. Manage.*, vol. 13, 2018, pp. 978–981.
- [33] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [34] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2014.
- [35] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 461–468.
- [36] A. Georgiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
- [37] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop Appl. Comput. Vis.*, Dec. 2002, pp. 138–142.



JUNJIAN ZHANG (Student Member, IEEE) is currently pursuing the Ph.D. degree with the Pattern Recognition and Intelligent System Laboratory, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include machine learning, pattern recognition, and computer vision.



CHUN-GUANG LI (Member, IEEE) received the B.E. degree in telecommunication engineering from Jilin University, Jilin, China, in 2002, and the Ph.D. degree in signal processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2007.

From 2011 to 2012, he visited the Visual Computing Group, Microsoft Research Asia, Beijing. From 2012 to 2013, he also visited the Vision, Dynamics, and Learning Laboratory, Johns Hopkins University, Baltimore, MD, USA. He is currently an Associate Professor with the School of Information and Communication Engineering, BUPT. His current research interests include statistical signal processing and machine learning. Dr. Li is a member of the ACM and CCF.



TIANMING DU (Student Member, IEEE) received the B.S. degree in network engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree in telecommunication engineering. From 2016 to 2019, he was a Project Leader with the Pattern Recognition and Intelligent System Laboratory, BUPT. His research interests include the development of subspace clustering in unsupervised learning and biological/medical imaging processing.



HONGGANG ZHANG (Senior Member, IEEE) received the B.S. degree from Shandong University, Jinan, China, in 1996, and the M.S. and Ph.D. degrees from the School of Information Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999 and 2003, respectively.

He was a Visiting Scholar with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, from 2007 to 2008. He is currently an Associate Professor and the Director of the Web Search Center, BUPT. His current research interests include image retrieval, computer vision, and pattern recognition. He has published over 60 technical articles in his fields.



JUN GUO (Member, IEEE) received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree from Tohoku Gakuin University, Sendai, Japan, in 1993.

He is currently a Professor and the Vice President of BUPT. His current research interests include pattern recognition theory and application, information retrieval, content-based information security, and network management. He has published over 100 technical articles in his fields.

• • •