Deep Subspace Clustering

Xi Peng, *Member, IEEE*, Jiashi Feng, Joey Tianyi Zhou, Yingjie Lei, and Shuicheng Yan, *Fellow, IEEE*

***Abstract*— In this article, we propose a deep extension of sparse subspace clustering, termed deep subspace clustering with L1-norm (DSC-L1). Regularized by the unit sphere distribution assumption for the learned deep features, DSC-L1 can infer a new data affinity matrix by simultaneously satisfying the sparsity principle of SSC and the nonlinearity given by neural networks. One of the appealing advantages brought by DSC-L1 is that when original real-world data do not meet the class-specific linear subspace distribution assumption, DSC-L1 can employ neural networks to make the assumption valid with its nonlinear transformations. Moreover, we prove that our neural network could sufficiently approximate the minimizer under mild conditions. To the best of our knowledge, this could be one of the first deep-learning-based subspace clustering methods. Extensive experiments are conducted on four real-world data sets to show that the proposed method is significantly superior to 17 existing methods for subspace clustering on handcrafted features and raw data.**

***Index Terms*— Least square regression (LSR) clustering, low-rank representation (LRR), sparse subspace clustering (SSC), subspace clustering.**

I. Introduction

SUBSPACE clustering aims at simultaneously implicitly finding out an underlying subspace to fit each group of data points and performing clustering based on the learned subspaces, which has attracted a lot of interest from the computer vision and image processing community [1]. Most existing subspace clustering methods can be roughly divided into following categories: algebraic methods [2], iterative methods [3], statistical methods [4], and spectral clustering-based methods [5], [6].

Recently, a large number of spectral clustering-based methods have been proposed [7]–[19], which first form an affinity matrix using the linear reconstruction coefficients of the whole data set and then obtain the clustering results by applying spectral clustering on the affinity matrix. Those methods differ from each other mainly in their adopted priors of the coefficients. For example, ℓ_1-norm-based sparse subspace clustering (SSC) [8] and its ℓ_0-norm-based variant [18], low-rank representation (LRR) [13], and thresholding ridge regression (TRR) [20]–[22] build the affinity matrix using the linear representation coefficients under the constraint of ℓ_1-, nuclear-, and ℓ_2-norm, respectively. Formally, SSC, LRR, TRR, and many of their variants learn the representation coefficients to build the affinity matrix by

$$\min_{\mathbf{C}} \mathcal{L}(\mathbf{X} - \mathbf{X}\mathbf{C}) + \mathcal{R}(\mathbf{C}) \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{n\times n}$ denotes the linear representation of the input $\mathbf{X} \in \mathbb{R}^{d\times n}$, d denotes the dimension of data, n is the number of data points, $\mathcal{R}(\mathbf{C})$ denotes certain imposed structure prior over $\mathbf{C}$, and the choice of representation error function $\mathcal{L}(\cdot)$ is usually dependent on the distribution assumption of $\mathbf{X}$, e.g., a typical loss function is $\mathcal{L}(\mathbf{X} - \mathbf{X}\mathbf{C}) = \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F$.

Although these methods have achieved impressive performance for subspace clustering, they generally suffer from the following limitations. First of all, those methods assume that each sample can be linearly reconstructed by the whole sample collection. However, in real-world situations, the data may not be linearly represented by each other in the input space. Therefore, the performance of those methods usually drops in practice. To address this problem, several recent works [23]–[26] have developed kernel-based approaches, which have shown their effectiveness in subspace clustering. However, kernel-based approaches are similar to template-based approaches, whose performance heavily depends on the choice of kernel functions. Moreover, the approaches cannot give explicit nonlinear transformations, causing difficulties in handling large-scale data sets.

Inspired by the remarkable success of deep learning in various applications [27], [28], in this article, we propose a new subspace clustering framework based on neural networks [namely deep subspace clustering (DSC)] and apply the framework to extend the well-known SSC to develop a new method termed deep subspace clustering with L1-norm (DSC-L1). The basic idea of DSC-L1 (see Fig. 1) is simple but effective. It uses a neural network to project data into another space in which SSC is valid to the nonlinear subspace case.

Manuscript received March 18, 2019; revised August 8, 2019 and December 19, 2019; accepted January 19, 2020. This work was supported in part by the NFSC under Grant 61806135, Grant U19A2081, and Grant 61625204, in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949, in part by the Singapore government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain) under Grant A18A1b0045, in part by the Key Research and Development Program of Sichuan Province under Grant 2019YFG0409, and in part by the Beijing Academy of Artificial Intelligence (BAAI). *(Corresponding author: Joey Tianyi Zhou.)*

Xi Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: pengx.gm@gmail.com).

Jiashi Feng is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: elefjia@nus.edu.sg).

Joey Tianyi Zhou is with the Institute of High Performance Computing, A*STAR, Singapore 138632 (e-mail: joey.tianyi.zhou@gmail.com).

Yingjie Lei is with the College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China (e-mail: yinjie@scu.edu.cn).

Shuicheng Yan is with YITU Tech, Beijing 100086, China (e-mail: Shuicheng.yan@yitu-inc.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

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/TNNLS.2020.2968848

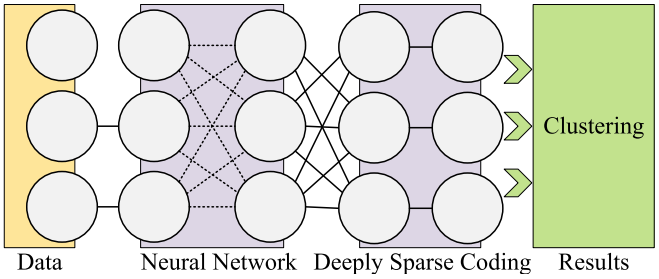

Fig. 1. Flowchart of the proposed DSC-L1 method. For a given data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, we project it into the feature space given $\mathbf{H}^{(M)} = [\mathbf{h}_1^{(M)}, \mathbf{h}_2^{(M)}, \ldots, \mathbf{h}_n^{(M)}]$ through a series of nonlinear transformations; and learn the self-sparse representation of inputs at the top layer of the neural network. Here, M is the index of the top layer of the neural network. We apply spectral clustering on the affinity matrix built by the obtained representation such as SSC. The proposed method simultaneously enjoys the sparsity induced by the ℓ_1-norm regularization and the expressive nonlinearity of the neural network.

Unlike most existing subspace clustering methods, our method simultaneously learns a set of transformations parameterized by a neural network and the reconstruction coefficients to represent each mapped sample as a combination of others. Compared with kernel-based approaches, DSC-L1 is a deep instead of shallow model, which can explicitly map samples from the input space into a latent space, with parameters in the transformations learned in a data-driven way. To the best of our knowledge, DSC-L1 could be the first deep extension of SSC, which satisfies the sparsity principle of SSC and, meanwhile, makes SSC valid to nonlinear subspace case.

The contribution of this article is twofold. From the view of subspace clustering, we show how to make it benefit from the success of deep neural networks so that the nonlinear subspace clustering could be achieved. From the view of neural networks, we show that it is feasible to integrate the advantages of existing subspace clustering methods and deep learning to develop new unsupervised learning algorithms.

Notations: Throughout this article, lower case bold letters represent column vectors and upper case bold ones denote matrices. $\mathbf{A}^\top$ denotes the transpose of the matrix $\mathbf{A}$ and $\mathbf{I}$ denotes an identity matrix.

II. Related Works

A. Subspace Clustering

The past decade saw an upsurge of subspace clustering methods with various applications in computer vision, e.g., motion segmentation [4], [8], [13], [14], face clustering [11], [15], image processing [9], [18], multiview clustering [29], and video analysis [25]. In particular, among these works, spectral clustering-based methods have achieved state-of-the-art results. The key to these methods is to learn an affinity matrix $\mathbf{A}$ in which $\mathbf{A}_{ij}$ denotes the similarity between the ith and the jth sample. Ideally, $\mathbf{A}_{ij} \neq 0$ only if the corresponding data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are drawn from the same subspace. To this end, some recent works, e.g., SSC [8]), L0-SSC [18], LRR [13], least square regression (LSR) [14], and smooth representation (SMR) [11], assume that any given sample can be linearly reconstructed by other samples in the input space. Based on the self-representation, an affinity matrix (or called similarity graph) can be constructed and fed to spectral clustering algorithms to obtain the final clustering results. In practice, however, high-dimensional data (such as face images) usually reside on the nonlinear manifold. Unfortunately, linear reconstruction assumption may not be satisfied in the original space, and in this case, the methods may fail to capture the intrinsic nonlinearity of manifold. To address this limitation, the kernel approach is used to first project samples into a high-dimensional feature space in which the representation of the whole data set is computed [23]–[26]. After that, the clustering result is achieved by performing traditional subspace clustering methods in the kernel space. However, the kernel-based methods behave like template-based approaches that usually require the prior knowledge on the data distribution to choose a desirable kernel function. Clearly, such a prior is hard to obtain in practice. Moreover, they cannot learn an explicit nonlinear mapping function from data sets, thus suffering from the scalability issue and the out-of-sample problem [30], [31].

Unlike these classical subspace clustering approaches, our method learns a set of explicit nonlinear mapping functions from data set to map the input into another space and calculates the affinity matrix using the representation of the samples in the new space.

B. Deep Learning

Aimed at learning high-level features from inputs, deep learning has shown promising results in numerous computer vision tasks in the scenario of supervised learning [32]–[34]. In contrast, less attention [35]–[38] has been paid to the applications with unsupervised learning scheme. Recently, some works [12], [39], [40]–[49] have devoted to combining deep learning and unsupervised learning and some of them shown impressive results in a clustering analysis. Most of these methods share the same basic idea, i.e., using deep learning to learn a good representation and then achieving clustering with the existing clustering methods, such as the vanilla k-means [50]. The major differences among them reside on the neural network structure and the objective function.

Different from these works, we propose a new model to bridge subspace clustering and neural networks to achieve nonlinear subspace clustering and focus on subspace clustering rather than clustering. To be specific, our framework, i.e., DSC, simultaneously learns high-level features from inputs and self-representation in a joint way, whereas these existing methods do not enjoy the effectiveness of the self-expressive subspace clustering. We believe that such a general framework is complementary with the existing shallow subspace clustering methods since it can incorporate the success of these methods into deep learning. To the best of our knowledge, this could be the first of several DSC methods. Our model is also significantly different from [41] and its low-rank extension [44] in the following aspects.

1) The motivation is different. Peng *et al.* [41] required the data that could be linearly reconstructed in the input space, and thus, the obtained representation coefficients

can be effectively embedded into a latent space useful for clustering. In contrast, our model aims to solve the problem of nonlinear subspace clustering, i.e., the data cannot be linearly represented in the input space.

2) The objective function is different. Peng *et al.* [41] explicitly minimized the reconstruction error between inputs and outputs, which is an autoencoder. In contrast, our method is a feedforward neural network, which only implicitly uses the autoencoder to initialize our model.
3) Our method works in an end-to-end manner to optimize the affinity matrix and the parametric neural network, whereas [41] treats these two steps separately.
4) The proposed DSC-L1 can be regarded as a deep nonlinear extension of the well-known SSC, which makes SSC handling nonlinear subspace clustering possible. In contrast, [41] and [49] cannot be interpreted in this way and do not have such a foundation.

Moreover, this article is also different from another recent independent work, i.e., deep subspace clustering network (DSCN) [12] in given aspects. First, the object function is different. Second, the structure of neural networks is different. To be exact, [12] adopts an autoencoder structure such as [21], whereas our DSC is a forward neural network that does not require to reconstruct the input itself. In consequence, our method does not seek a good tradeoff between the reconstruction and self-expression errors and then enjoy a smaller parameter size.

III. DSC

In this section, we first briefly review SSC and then present the details of our DSC method.

A. SSC

For a given data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, SSC seeks to linearly reconstruct the ith sample $\mathbf{x}_i$ using a few of other samples. In other words, the representation coefficients are expected to be sparse by employing the following formulation:

$$\min_{\mathbf{c}_i} \frac{1}{2}\|\mathbf{x}_i - \mathbf{X}\mathbf{c}_i\|_F^2 + \gamma\|\mathbf{c}_i\|_1 \quad \text{s.t. } c_{ii} = 0 \tag{2}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm (i.e., the sum of absolute values of all elements in a vector) that acts as a relaxation of ℓ_0-norm and c_{ii} denotes the ith element in $\mathbf{c}_i$. Specifically, penalizing $\|\mathbf{c}_i\|_1$ encourages $\mathbf{c}_i$ to be sparse, and enforcing the constraint $c_{ii} = 0$ avoids trivial solutions. To deal with the optimization problem (2), the alternating direction method of multipliers (ADMM) [51], [52] is often used.

Once the sparse representation of the whole data set is obtained by solving (2), an affinity matrix in SSC is calculated as $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^\top$, and then, spectral clustering is applied to $\mathbf{A}$ to give the clustering results.

B. DSC

In most existing subspace clustering methods including SSC, each sample is encoded as a linear combination of the whole data set. However, when dealing with high-dimensional data that usually lie on nonlinear manifolds, such methods may fail to capture the nonlinear structure, thus leading to inferior results. To address this issue, we propose a deep-learning-based method that maps the given samples using in a neural network and simultaneously learns the reconstruction coefficients (i.e., the affinity) to represent each mapped sample as a combination of others.

As shown in Fig. 1, the neural network in our proposed framework consists of $M+1$ stacked layers with M nonlinear transformations, which takes a given sample $\mathbf{x}$ as the input to the first layer. For ease of presentation, we make several definitions as follows. For the first layer of our neural network, we define its input as $\mathbf{h}^{(0)} = \mathbf{x} \in \mathbb{R}^d$. Moreover, for the subsequent layers, let

$$\mathbf{h}^{(m)} = g(\mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}) \in \mathbb{R}^{d^{(m)}} \tag{3}$$

be the output of the mth layer (in which $m = 1, 2, \ldots, M$ indexes the layer), where $g(\cdot)$ is a nonlinear activation function, $d^{(m)}$ is the dimension of the output of the mth layer, and $\mathbf{W}^{(m)} \in \mathbb{R}^{d^{(m)}\times d^{(m-1)}}$ and $\mathbf{b}^{(m)} \in \mathbb{R}^{d^{(m)}}$ denote the weights and bias associated with the mth layer, respectively. Let $\mathbf{x}$ be the input of the first layer, and the output at the top layer of our neural network is

$$\mathbf{h}^{(M)} = g(\mathbf{W}^{(M)}\mathbf{h}^{(M-1)} + \mathbf{b}^{(M)}). \tag{4}$$

In fact, if denoting (4) as $f(\mathbf{x})$, we can observe that $f(\cdot)$: $\mathbb{R}^d \to \mathbf{R}^{d^{(m)}}$ is a nonlinear function determined by the weights and biases of our neural network (i.e., $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^M$) as well as the choice of activation function $g(\cdot)$. Furthermore, for n samples, we define $\mathbf{H}^{(M)}$ as the collection of the corresponding outputs given by our neural network, that is

$$\mathbf{H}^{(M)} = \left[\mathbf{h}_1^{(M)}, \mathbf{h}_2^{(M)}, \ldots, \mathbf{h}_n^{(M)}\right]. \tag{5}$$

With the earelier definitions, the proposed objective function is in the following form:

$$\min_{\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\}_{m=1}^M,\mathbf{C}} \mathcal{J} = \mathcal{J}_1 + \lambda\mathcal{J}_2 \tag{6}$$

where λ is a positive tradeoff parameter and $\{\mathcal{J}_i\}_{i=1}^2$ are defined in the following. Intuitively, the first term $\mathcal{J}_1$ is designed to minimize the discrepancy between $\mathbf{H}^{(M)}$ and its self-expressed representation. Moreover, it meanwhile regularizes $\mathbf{C}$ for some desired properties. To be specific, $\mathcal{J}_1$ can be expressed in the form of

$$\mathcal{J}_1 = \mathcal{L}(\mathbf{H}^{(M)} - \mathbf{H}^{(M)}\mathbf{C}) + \mathcal{R}(\mathbf{C}) + \mathcal{F}(\mathbf{C}) \tag{7}$$

where $\mathcal{F}(\mathbf{C})$ takes the value of $+\infty$ if $\mathbf{C}$ is not in some feasible domains and 0 otherwise. Note that, the form of $\mathcal{L}(\cdot), \mathcal{R}(\cdot)$, and $\mathcal{F}(\cdot)$ may be adopted from many existing subspace clustering works. In this article, we take $\mathcal{L}(\cdot) = \|\cdot\|_F^2$, $\mathcal{R}(\mathbf{C}) = \|\mathbf{C}\|_p$, and $\mathcal{F}(\mathbf{C}) = +\infty$ if the condition, such as $\text{diag}(\mathbf{C}) = \mathbf{0}$, is violated. Otherwise, $\mathcal{F}(\mathbf{C}) = 0$. Furthermore, $\|\cdot\|_p$ denotes the ℓ_p-norm and three choices are most popular for subspace clustering, namely, ℓ_1-norm [8], nuclear-norm [13], and ℓ_2-norm [20].

The second part $\mathcal{J}_2$ is designed to remove an arbitrary scaling factor in the latent space. In this article, we set

$$\mathcal{J}_2 = \frac{1}{4}\sum_{i=1}^{n} \left\| \left(\mathbf{h}_i^{(M)}\right)^\top \mathbf{h}_i^{(M)} - 1 \right\|_2^2. \tag{8}$$

Note that, without the above term, our neural network may collapse in the trivial solutions such as $\mathbf{H}^{(M)} = \mathbf{0}$. However, if some approaches are adopted to solve this issue, $\mathcal{J}_2$ could be removed. For example, the autoencoder structure is explicitly or implicitly incorporated into the objective function (6), such as [12] and [41].

With $\{\mathcal{J}_i\}_{i=1}^2$ detailed earlier, the optimization problem of the proposed DSC can be expressed as follows:

$$\min_{\Theta,\mathbf{C}} \frac{1}{2}\|\mathbf{H}^{(M)} - \mathbf{H}^{(M)}\mathbf{C}\|_F^2 + \gamma \|\mathbf{C}\|_p + \mathcal{F}(C) + \frac{\lambda}{4}\sum_{i=1}^{n} \left\| \left(\mathbf{h}_i^{(M)}\right)^\top \mathbf{h}_i^{(M)} - 1 \right\|_2^2 \tag{9}$$

where Θ denotes the parametric neural network, i.e., $\Theta = \{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^M$.

In this article, one major goal is to develop a deep extension of SSC (denoted by DSC-L1), thus leading to the following objective function:

$$\begin{aligned} \min_{\Theta,\mathbf{C}} \ & \frac{1}{2}\|\mathbf{H}^{(M)} - \mathbf{H}^{(M)}\mathbf{C}\|_F^2 + \gamma \|\mathbf{C}\|_1 \\ & + \frac{\lambda}{4}\sum_{i=1}^{n} \left\| \left(\mathbf{h}_i^{(M)}\right)^\top \mathbf{h}_i^{(M)} - 1 \right\|_2^2 \\ \text{s.t. } & \operatorname{diag}(\mathbf{C}) = 0 \end{aligned} \tag{10}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm, which could guarantee the sparsity of $\mathbf{C}$. Note that other norms could also be complementary with our method. We adopt the L1-norm here for two reasons. On one hand, L1-norm is provable to achieve block-diagonal affinity [8], which is essential to subspace clustering. On the other hand, we also investigate the L2-norm for evaluations and the experiment shows that it is relatively inferior to the L1-norm in our neural network.

C. Optimization

Our model could be optimized in two ways. First, such as [12], $\mathbf{C}$ is treated as a layer stacked on the top of the neural network, and then, Θ and $\mathbf{C}$ are optimized by the existing deep learning libraries, such as TensorFlow. Second, it is decomposed into two subproblems, which is helpful to the following convergence proof. In this section, we elaborate the second way.

For ease of presentation, we first rewrite (10) with the ℓ_1-norm as follows:

$$\min_{\Theta,\mathbf{c}_i} \sum_{i=1}^{n} \Big(\frac{1}{2}\left\|\mathbf{h}_i^{(M)} - \mathbf{H}_i^{(M)}\mathbf{c}_i\right\|_F^2 + \gamma \|\mathbf{c}_i\|_1 + \frac{\lambda}{4}\left\| \left(\mathbf{h}_i^{(M)}\right)^\top \mathbf{h}_i^{(M)} - 1 \right\|_2^2 \Big) \tag{11}$$

where $\mathbf{H}_i^{(M)}$ is a variant of $\mathbf{H}^{(M)}$, which is obtained by simply replacing $\mathbf{h}_i^{(M)}$ in $\mathbf{H}^{(M)}$ with $\mathbf{0}$.

Given n data points, DSC-L1 simultaneously learns M nonlinear mapping functions $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^M$ and n sparse codes $\{\mathbf{c}_i\}_{i=1}^n$ by solving (11). As (11) is a multiple-variable optimization problem, we employ an alternating minimization algorithm by alternatively updating one of the variables while fixing the others.

Step 1: Fix $\mathbf{c}_i$ and $\mathbf{H}_i^{(m)}$, update Θ, and (11) can be rewritten as

$$\min_{\Theta} \frac{1}{2}\left\|\mathbf{h}_i^{(M)} - \mathbf{H}_i^{(M)}\mathbf{c}_i\right\|_2^2 + \alpha_i + \frac{\lambda}{4}\left\| \left(\mathbf{h}_i^{(M)}\right)^\top \mathbf{h}_i^{(M)} - 1 \right\|_2^2 \tag{12}$$

where $\alpha_i = \sum_{j\neq i} \|\mathbf{h}_j^{(M)} - \mathbf{H}_j^{(M)}\mathbf{c}_j\|_2^2 + \lambda\|(\mathbf{h}_i^{(M)})^\top \mathbf{h}_i^{(M)} - 1\|_2^2$ is a constant.

To solve (12), we adopt the stochastic subgradient descent (SGD) algorithm to obtain the parameters $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^M$. Moreover, we also enforce the ℓ_2-norm on the parameters to avoid overfitting [53], [54].

Step 2: Fix $\{\mathbf{h}_i^{(M)}\}_{i=1}^n$ and update $\mathbf{c}_i$ by

$$\min_{\mathbf{c}_i} \frac{1}{2}\left\|\mathbf{h}_i^{(M)} - \mathbf{H}_i^{(M)}\mathbf{c}_i\right\|_F^2 + \gamma \|\mathbf{c}_i\|_1 + \beta_i \tag{13}$$

where

$$\beta_i = \sum_{j\neq i} \left(\frac{1}{2}\left\|\mathbf{h}_j^{(M)} - \mathbf{H}_j^{(M)}\mathbf{c}_j\right\|_2^2 + \gamma \|\mathbf{c}_j\|_1 \right)$$

is a constant. Note that (13) is a standard ℓ_1-minimization problem faced by SSC, which can be solved by using many existing ℓ_1-solvers [55]. Steps 1 and 2 are repeated until convergence.

After obtaining $\mathbf{C}$ with either the first or the second optimization way, we construct an affinity matrix via $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^\top$ and obtain the clustering results based on $\mathbf{A}$. The aforementioned optimization procedure is summarized in Algorithm 1.

D. Discussion

Our approach DSC-L1 can provide a satisfactory subspace clustering performance befitting from the following factors. First, different from SSC, DSC-L1 performs sparse coding in a latent representation space learned by the neural network from data instead of the original one. By transforming into the latent space, the samples become more favorable for sparse reconstruction. Note that such an extension is nontrivial since the proposed objective function also includes learning the neural network parameters and the induced data representation. Clearly, it is different from SSC, which only learns the representation coefficients. To the best of our knowledge, this could be one of the first research works attempt to make subspace clustering benefiting from the development of deep learning. Furthermore, this article is also complementary with deep learning since it demonstrates the potential of subspace clustering to facilitating the development and deployment of unsupervised deep-learning methods and unsupervised deep learning is a key challenge in deep-learning community [35]. Second, DSC-L1 can also be deemed as a kernel-based method, which automatically learns the kernel functions and transformations in a data-driven way. Considering the demonstrated effectiveness of kernel-based subspace clustering approaches, such

Algorithm 1 DSC

Input: A given data set $\mathbf{X}$ and the tradeoff parameters λ.
// Initialization:
Initialize $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$, and $\mathbf{H}^{(0)} = \mathbf{X}$.
for $m = 1, 2 \cdots, M$ **do**
  Do forward propagation to get $\{\mathbf{H}^{(m)}\}_{m=1}^{M}$ and $\mathbf{C}$ via solving (3) and (13), respectively.
end
// Optimization
while *not converge* **do**
  for $i = 1, 2, \cdots, n$ **do**
    Randomly select a data point $\mathbf{x}_i$ and let $\mathbf{h}_i^0 = \mathbf{x}_i$,
    for $m = 1, 2 \cdots, M$ **do**
      Compute $\mathbf{h}_i^{(m)}$ via (3).
    end
    Compute $\mathbf{c}_i$ using $\mathbf{h}_i^{(M)}$ via (13).
    for $m = M, M-1 \cdots, 1$ **do**
      Calculate the gradient using the SGD algorithm.
    end
    for $m = 1, 2, \cdots, M$ **do**
      Update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ with the gradient.
    end
  end
end
Output: $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$ and $\mathbf{C}$.

as [24] and [25], DSC-L1 is well expected to show even better performance for subspace clustering due to the outstanding representative capacity of deep neural networks.

In the proposed DSC, we do not explicitly minimize the reconstruction error by adopting an autoencoder structure due to the following reasons. First, one major goal of subspace clustering is to obtain a good representation so that similar inputs could be grouped into the same cluster and dissimilar inputs are separated into different clusters. As proved in [8], an ℓ_1-norm-based self-expression could enjoy such properties because it is able to learn a block-diagonal affinity matrix, i.e., embracing the intracluster compactness and intercluster scatter. In contrast, autoencoder aims to achieve compressive representations by exploring the latent structure of each single data point. If the reconstruction loss is incorporated, one has to find an optimal tradeoff between the compression and block diagonality that are induced by the reconstruction and the self-expression, respectively. Furthermore, we experimentally found that explicitly optimizing the reconstruction error did not give better performance. The comparison with DSCN [12] would support this choice since the only one difference between our method and DSCN is that DSCN adopts the structure of autoencoder.

It should be pointed out that the proposed method could adopt a similar structure with deep metric learning networks (DMLNs) [56]–[58], i.e., a set of fully connected layers to perform nonlinear transformation and then perform specific task on the output of neural network. The major differences among them are as follows. First, the objective functions are different. Our method aims to segment different samples into different subspaces, whereas these metric learning networks aim to learn the similarity function that measures how similar or related two data points are. Second, our DSC-L1 is unsupervised, whereas DMLNs are supervised approaches, which require the label information to train neural networks. Furthermore, our method could also be compatible with other neural networks, such as convolutional neural networks (CNNs). Such a possibility has been verified in the following experiments on the raw data.

IV. Convergence Analysis

Our objective function could be decomposed into two subproblems [i.e., (12) and (13)]. Clearly, (13) is a standard ℓ_1-norm-based optimization whose convergence property has been well studied in numerous works [59], [60]. Thus, we only focus on the convergence property of subproblem (12). Specifically, we will show that the corresponding loss and the weights of the neural network keep decreasing at each step under mild conditions (Conditions 1 and 2).

Although deep learning has achieved remarkable success in a variety of applications, only a few works [61]–[63] have analyzed its convergence behavior and all of them focused on two-layer networks due to two reasons. On the one hand, a two-layer network could approximate any continuous function [64]. On the other hand, a multiple-layer network always involves a nonconvex problem whose convergence behavior still remains an opening question. Following the setting in these works and motivated by them, we consider a two-layer network with the weight Θ, where the basis $\mathbf{b}$ and weight $\mathbf{W}$ are enveloped into Θ by rewriting $\Theta = [\mathbf{W}\ \mathbf{b}]$ without loss of generality.

For simplicity of presentation, let $\mathcal{L}$ be the loss of (12), $\mathcal{L}^*$ denote the smallest loss, and $\mathcal{L}_t^*$ be the smallest loss found at the t-step so far. Similarly, Θ^* denotes the desirable parametric model. We consider the standard SGD to optimize our network, that is

$$\Theta_{t+1} = \Theta_t - \eta_t \nabla \mathcal{L}(\Theta_t) \tag{14}$$

where $\nabla \mathcal{L}(\Theta_t)$ denotes the gradient of $\mathcal{L}$ with respect to Θ_t. In the following, we will alternatively use $\nabla \mathcal{L}(\Theta_t)$ and $\nabla \mathcal{L}_t$ without causing confusion.

Condition 1 (Lipschitz Continuity): A function $f(x)$ is a Lipschitz continuous function on the set Ω, if there exists a constant $\epsilon > 0$ $\forall x_1, x_2 \in \Omega$ such that

$$\|f(x_1) - f(x_2)\| \leq \epsilon \|x_1 - x_2\| \tag{15}$$

where ϵ is termed the Lipschitz constant.

Clearly, (12) is the Lipschitz continuity if $\nabla \mathcal{L}_t$ is upper bounded by ϵ, that is

$$\|\nabla \mathcal{L}_t\|_F \leq \epsilon. \tag{16}$$

Under condition (16), we could have the following result.

Theorem 1: Let $\alpha = \|\Theta_1 - \Theta^*\|_F$ and $\|\nabla \mathcal{L}_t\|_F \leq \epsilon$, and one could find an optimal model $\mathcal{L}_T^*$, which is sufficiently close to the desired $\mathcal{L}^*$. Mathematically

$$\mathcal{L}_T^* - \mathcal{L}^* \leq \frac{\alpha + \epsilon^2 \sum_{t=1}^{T} \eta_t^2}{2 \sum_{t=1}^{T} \eta_t}. \tag{17}$$

From Theorem 1, it is easy to obtain Corollaries 1 and 2.

Corollary 1: For the constant step size (i.e., $\eta_t = \eta$) and $T \to \infty$

$$\mathcal{L}_T^* - \mathcal{L}^* \to \frac{\eta\epsilon^2}{2}. \tag{18}$$

Corollary 2: For the constant step length (i.e., $\eta_t = \eta/\|\nabla\mathcal{L}_t\|_F$) and $T \to \infty$

$$\mathcal{L}_T^* - \mathcal{L}^* \to \frac{\eta\epsilon}{2}. \tag{19}$$

Corollaries 1 and 2 show that the loss will converge to $\mathcal{L}^*$ with a radius of $(\eta\epsilon^2/2)$ and $(\eta\epsilon/2)$ within T steps.

Besides, with the Lipschitz continuity, one could have another convergence property with the following condition [62].

Condition 2: A function $f(\mathbf{x})$ is called δ-one point strongly convex in the domain $\mathcal{D}$ with respect to $\mathbf{x}^*$, if $\forall\mathbf{x} \in \mathcal{D}$ such that

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \alpha \tag{20}$$

$$\langle -\nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x}\rangle > \delta\|\mathbf{x}^* - \mathbf{x}\|_2^2. \tag{21}$$

Theorem 2: Suppose that $\mathcal{L}(\Theta_t)$ is δ-one point strongly convex at each step $\forall\Theta_t \in \mathcal{D}$ and satisfies the Lipschitz condition, and if

$$\|\Theta_t - \Theta^*\|_F^2 \leq \frac{\eta_t\epsilon^2}{2\delta} \tag{22}$$

then Θ will monotonically close to the minimizer Θ^* by a factor of $1 - \eta\delta$, that is

$$\|\Theta_{t+1} - \Theta^*\|_F^2 \leq (1 - \eta_t\delta)\|\Theta_t - \Theta^*\|_F^2. \tag{23}$$

From Theorem 2, Corollaries 3 and 4 are derived.

Corollary 3: For the constant step size ($\eta_t = \eta$), after T steps and $\forall\alpha > 0$, Θ_T and $\mathcal{L}_T$ will sufficiently close to Θ^* and $\mathcal{L}(\Theta^*)$ by

$$\|\Theta_T - \Theta^*\|_F^2 \leq \alpha e^{-\frac{\eta\delta T}{2}} \tag{24}$$

and

$$\|\mathcal{L}_T - \mathcal{L}^*\|_2^2 \leq \epsilon\alpha e^{-\frac{\eta\delta T}{2}}. \tag{25}$$

Corollary 4: For the constant step length ($\eta_t = \eta/\|\nabla\mathcal{L}_t\|_F$), after T steps and $\forall\alpha > 0$, Θ_T and $\mathcal{L}_T$ will sufficiently close to Θ^* and $\mathcal{L}(\Theta^*)$ by

$$\|\Theta_t - \Theta^*\|_F^2 \leq \alpha e^{-\frac{\eta\delta T}{2\epsilon}} \tag{26}$$

and

$$\|\mathcal{L}_t - \mathcal{L}^*\|_2^2 \leq \epsilon\alpha e^{-\frac{\eta\delta T}{2\epsilon}}. \tag{27}$$

V. Experiments

In this section, we compare our method with 17 popular subspace clustering methods on five different real-world data sets in terms of four clustering performance metrics. The experiments consist of two parts, namely, clustering on handcrafted features and clustering on raw data.

A. Data Sets and Experimental Settings

1) Data Sets: Five different data sets are used in our experiments, i.e., COIL20 object images [65], COIL100 object images [65], the MNIST handwritten digital database [66], AR facial images [67], and the BF0502 video face data set [68].

1) The COIL20 database contains 1440 samples distributed over 20 objects, where each image is with the size of 32×32.
2) The COIL100 database contains 7200 samples distributed over 100 objects, where each image is with the size of 32×32.
3) The MNIST data set includes 60 000 handwritten digit images of which the first 2000 training images and the first 2000 testing images are used in our experiments, where the size of each image is 28×28.
4) The AR database is one of the most popular facial image data sets for subspace clustering. In our experiments, we use a widely used subset of the AR database [69], which consists of 1400 undisguised faces evenly distributed over 50 males and 50 females, where the size of each image is 165×120.
5) The BF0502 data set contains the facial images detected from the TV series *Buffy the Vampire Slayer*. Following [25], a subset of BF0502 is used, which includes 17 337 faces in 229 tracks from 6 main casts. Each facial image is represented as a 1937-D vector extracted from 13 facial landmark points (e.g., the left and right corners of each eye). In our experiments, we use the first 200 samples from each category, thus resulting in 1200 images in total.

For a comprehensive investigation, we design two experiments. The first experiment aims to examine the ability of nonlinear subspace clustering, which is carried out on the handcrafted features extracted from COIL20, MNIST, and AR (detailed later). The second experiment aims to show the effectiveness of the proposed method in learning from raw data, which is conducted on the full COIL20 and COIL100 raw data by following the setting in [12].

2) Implementation Details: Here, we introduce the implementation details of the used activation functions and the initialization of $\{\mathbf{W}^{(m)}, \mathbf{b}^m\}$. To be specific, the activation functions can be chosen from various forms. In our experiments, we use the tan$h$ function for the fully connected network (see Section V-B) and rectified linear unit (ReLu) for the convolutional network (see Section V-C).

Regarding the initializations of $\{\mathbf{W}^{(m)}, \mathbf{b}^m\}$, we initialize $\mathbf{W}^{(m)}$ as a rectangular matrix with ones at the main diagonal and zeros as other elements. Moreover, $\mathbf{b}^{(m)}$ is initialized as $\mathbf{0}$.

In our implementation, we adopt two popular convergence criteria, namely, max training epoch (fixed to 100) and convergence threshold (fixed to 10^3), where the second criterion is based on the difference in the loss between two continuous training epochs. Either of these two conditions is satisfied, and the network is regarded as converged.

In the experiments, we train a DSC-L1 consisting of three layers, with 300, 200, and 150 neurons respectively. Moreover, we set $\lambda = 10^{-3}/\text{n}$, $\varphi = 10^{-3}$, and the convergence threshold

as 10^{-3} for DSC-L1 and adopt early stopping technique (with respect to the parameter τ) to avoid overfitting by following [54], where n is the data size.

3) Baseline Methods: We compare DSC-L1 with 17 clustering methods, i.e., SSC [8], Kernel SSC (KSSC) [24], LRR [13], low-rank subspace clustering (LRSC) [10], Kernel LRR [25], LSR [14], SMR [11], low-rank constrained autoencoder (LRAE) [44], DSCN [12], DEC [12], and IDEC [45]. LSR has two variants that are denoted by LSR1 and LSR2. KSSC and KLRR have also two variants that are based on the RBF function (KSSC1/KLRR1) and the polynomial function (KSSC2/KLRR2), respectively. LRAE has three variants and we evaluate the best one in our experiments according to [44]. DSCN has two variants, i.e., DSCN-L1 and DSCN-L2 that consider ℓ_1- and ℓ_2-norm-based coefficients, respectively. Moreover, we have also used the deep autoencoder (DAE) with SSC as a baseline to show the efficacy of our method. More specifically, we adopt the pretraining and fine-tuning strategy [70] to train a DAE that consists of five layers with 300, 200, 150, 200, and 300 neurons. In the experiments, we investigate the performance of DAE with two popular nonlinear activation functions, i.e., the sigmoid function (DAEg) and the saturating linear transfer function (DAEs). After the DAE converging, we perform SSC on the output of the third layer to obtain the clustering results. For fair comparisons, we use the same ℓ_1-solver (i.e., the Homotopy method [55]) to solve the ℓ_1-minimization problem in DSC-L1, SSC, and DAE. Note that we adopt the Keras implementation[1] of DEC since it experimentally performs better than the MxNet implementation.

4) Experimental Settings: In the first experiment, we adopt cross validation for selecting the optimal parameters for all the tested methods [54].[2] More specifically, we equally split each data set into two partitions and tune parameters using one partition. With the tuned parameters, we repeat each algorithm ten times on the other partition and report the achieved mean and standard deviation of the used clustering performance metrics. Note that we directly use the tuned parameters γ (sparsity) and δ (tolerance) of SSC for DSC-L1. If these two parameters are tuned specifically, the performance of DSC-L1 could be further improved. In the second experiment, we directly adopt the setting used in [12].

5) Evaluation Criteria: Like [49], we adopt four popular metrics to evaluate the clustering performance of our algorithm, i.e., accuracy (ACC) or called purity, normalized mutual information (NMI), adjusted rand index (ARI), and Fscore. Higher value of these metrics indicates a better performance.

B. Comparison on Handcrafted Features

For the purpose of nonlinear subspace clustering, we use the following four types of features extracted from the COIL20, MNIST, and AR data sets in experiments, i.e., dense scale-invariant feature transform (DSIFT) [71], the histogram of oriented gradients (HOG) [72], local binary pattern (LBP) [73], and local phase quantization (LPQ) [74]. The details of extracting these features are introduced as follows.

1) *DSIFT:* We divide each image into multiple nonoverlapping patches and then densely sample SIFT descriptors from each patch. The patch sizes of AR, COIL20, and MNIST are set as 15×15, 8×8, and 4×4, respectively. By concatenating these SIFT descriptors extracted from each image, we obtain a feature vector with the dimension of 11 264 (AR), 2048 (COIL20), and 6272 (MNIST).
2) *HOG:* We first divide each image into multiple blocks with two scales, i.e., 8×8 and 4×4 for AR and 4×4 and 2×2 for MNIST and COIL20. Then, we extract a 9-D HOG feature from each block. By concatenating these features for each image, the dimensions of the feature vector are 13 770 (AR), 2205 (MNIST), and 2880 (COIL20).
3) *LBP:* Like DSIFT, we divide each image into multiple nonoverlapping patches and then extract LBP features using eight sampling points on a circle of radius 1. Thus, we obtain a 59-D LBP feature vector from each patch. By concatenating the descriptors of each image, we obtain a feature vector with the dimensions of 7788 (COIL20) and 2891 (MNIST).
4) *LPQ:* The patch size is set as 8×8 for COIL20 and MNIST. For all the tested data sets, we set the size of the LPQ window as 3, 5, and 7. By concatenating the features of all patches of each image, the dimension of each feature is 12 288 for COIL20 and 6912 for MNIST.

For computational efficiency, we perform the principle component analysis (PCA) to reduce the dimension of handcrafted features of all data sets to 300 by following [8] and [56].

In this evaluation, DSC-L1 adopts a fully connected network that consists of five layers with 300, 200, 150, 200, and 300 neurons, where the last two layers are discarded after initialization. To achieve the nonlinear mapping, the tan*h* function is used as the activation function. Note that the first experiment will compare our method with all tested approaches excepted DSCN [12] since it is based on the CNN and cannot handle the handcrafted features.

1) On COIL20: We first investigate the performance of DSC-L1 using the COIL20 data set. Tables I and II report the results from which one can see that the following.

1) DSC-L1 consistently outperforms other tested methods in terms of all of the used performance metrics. Regarding the used four features, DSC-L1 achieves at least 1.86%, 2.09%, 0.96%, and 3.52% relative improvement over the ACC of the best baseline.
2) SSC usually outperforms DAEs and DAEg, whereas our DSC-L1 method consistently outperforms SSC in all the settings. This shows that it is hard to achieve a desirable performance by simply introducing deep learning into subspace clustering since unsupervised deep learning is an open challenging issue [35].
3) DEC slightly outperforms IDEC in some cases, e.g., on the DSIFT and HOG features. We assume that such a performance advantage should attribute to the

[1] https://github.com/XifengGuo/DEC-keras

[2] The following parameters are tuned with the cross-validation technique: DSC-L1 (μ and τ), SSC (γ and δ), KSSC (γ and δ), DAE (γ and δ), LRR (λ), KLRR (λ), LRSC (λ), LSR (λ), SMR (α and k), LRAE (λ_1 and λ_2), and IDEC (γ).

TABLE I

CLUSTERING RESULTS ON THE COIL20 DATA SET. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Features	DSIFT					HOG				
Methods	ACC	NMI	ARI	Fscore	Para.	ACC	NMI	ARI	Fscore	Para.
DSC-L1	**80.82±2.88**	**90.52±0.94**	**77.63±2.09**	**78.88±1.96**	2^{-12}, 20	**87.10±2.82**	**91.67±1.07**	**82.56±1.26**	**83.51±2.12**	2^{-12}, 30
SSC	78.96±3.12	89.06±1.03	76.46±2.31	77.59±2.17	0.5, 0.2	85.01±0.85	89.99±0.38	81.13±1.08	82.08±1.02	0.5, 0.1
KSSC1	71.00±2.13	78.72±0.98	63.33±1.85	65.18±1.75	10^{-3}, 18	75.29±0.97	82.75±0.49	66.46±1.43	68.20±1.33	10^{-2}, 18
KSSC2	72.01±2.68	83.84±0.89	64.22±3.47	66.22±3.16	10^{-3}, 18	69.53±1.30	81.27±0.69	61.16±1.83	63.32±1.69	10^{-2}, 18
DAEg	55.83±2.80	70.42±1.43	47.06±2.74	50.00±2.52	0.5, 0.2	69.60±1.00	78.52±0.47	59.38±0.79	61.63±0.74	0.5, 0.1
DAEs	55.81±2.60	70.71±1.68	48.49±3.31	51.46±3.05	0.5, 0.2	64.75±1.31	77.48±0.60	56.81±1.12	59.13±1.06	0.5, 0.1
LRR	71.03±1.47	80.52±1.05	63.83±2.09	65.70±1.97	5e-2	76.89±1.46	86.52±0.78	70.79±1.73	72.39±1.62	5e-3
KLRR1	70.46±1.55	79.61±1.01	61.25±1.94	63.35±1.81	500	76.74±0.27	82.00±0.14	69.43±0.48	70.96±0.45	10
KLRR2	70.85±1.37	80.09±1.15	62.75±1.54	64.73±1.46	100	72.33±2.65	80.98±1.21	63.11±2.88	65.07±2.68	5
LRSC	71.82±0.28	77.65±0.23	62.72±0.52	64.62±0.49	0.08	57.11±1.24	69.91±0.73	46.27±1.57	49.20±1.48	0.01
LSR1	63.93±2.15	73.18±1.12	53.29±2.26	55.75±2.14	0.6	54.81±1.80	64.44±0.94	42.28±1.55	45.35±1.44	0.5
LSR2	68.11±1.14	75.33±0.62	56.29±1.56	58.61±1.41	0.9	53.81±1.51	63.00±1.22	42.07±1.5	45.19±1.42	0.3
SMR	76.97±0.96	85.30±0.58	71.56±1.02	73.02±0.96	2^{-16},10^{-3}	80.15±0.87	85.93±0.6	73.51±1.06	74.87±1.01	2^{-16},10^{-3}
LRAE	77.25±1.87	88.45±0.96	73.93±1.33	72.75±1.42	5, 3	83.92±1.27	88.99±1.21	78.49±1.97	76.03±2.09	2, 1e-3
DEC	66.00±0.62	75.18±0.62	58.94±0.67	61.11±0.66	-	62.61±0.61	72.22±0.62	53.00±0.63	55.50±0.63	-
IDEC	58.94±0.75	74.13±0.72	55.14±0.72	57.56±0.72	0.1	61.33±0.69	73.16±0.62	53.63±0.79	56.11±0.78	0.1

TABLE II

CLUSTERING RESULTS ON THE COIL20 DATA SET. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Features	LBP					LPQ				
Methods	ACC	NMI	ARI	Fscore	Para.	ACC	NMI	ARI	Fscore	Para.
DSC-L1	**72.89±1.41**	**84.32±0.79**	**67.31±1.96**	**69.01±1.85**	2^{-13}, 40	**78.12±2.09**	**85.38±0.77**	**71.35±1.34**	**72.87±1.25**	2^{-12}, 60
SSC	70.17±0.65	82.66±0.19	64.19±0.60	66.07±0.58	10^{-3}, 10^{-2}	74.60±0.81	84.21±0.49	67.69±0.83	69.35±0.79	10^{-3}, 0.1
KSSC1	69.33±1.97	80.65±0.86	61.15±1.91	63.18±1.79	1, 16	68.49±2.38	79.28±1.27	59.06±2.37	61.23±2.21	0.1, 12
KSSC2	70.42±1.13	83.67±0.69	65.28±1.23	68.03±1.16	1, 16	69.24±2.33	79.52±0.93	61.07±1.72	63.17±1.62	0.1, 12
DAEg	40.96±2.18	53.54±0.89	26.27±1.33	30.57±1.22	10^{-3}, 10^{-2}	62.19±0.90	72.04±0.54	51.51±0.75	54.15±0.72	10^{-3}, 0.1
DAEs	40.68±1.13	52.12±0.92	23.67±1.30	28.26±1.10	10^{-3}, 10^{-2}	59.64±2.46	67.44±1.06	44.90±1.81	47.98±1.65	10^{-3}, 0.1
LRR	71.60±4.02	84.45±1.78	65.47±5.68	66.29±5.21	0.5	69.00±1.09	80.31±0.88	60.12±1.51	62.29±1.41	0.1
KLRR1	65.83±0.31	77.34±0.30	56.41±0.50	58.60±0.47	30	69.43±1.46	77.34±0.53	57.01±1.02	59.24±0.96	500
KLRR2	70.10±1.27	79.58±0.13	62.91±0.51	64.82±0.47	1000	65.33±2.48	76.41±1.13	54.22±2.11	56.69±1.94	100
LRSC	62.96±0.61	73.38±0.79	53.31±1.06	55.67±1.01	0.04	66.38±0.50	78.73±0.58	58.81±0.97	60.99±0.91	0.08
LSR1	70.24±2.90	82.40±1.41	64.54±2.85	67.33±2.69	1	66.97±1.68	74.42±0.62	55.48±1.52	57.74±1.43	0.2
LSR2	70.54±3.26	81.63±1.16	63.71±2.58	66.59±2.41	0.6	65.25±1.55	73.81±1.29	54.34±1.65	56.66±1.56	0.3
SMR	71.93±1.35	81.17±0.39	63.54±1.41	66.39±1.32	2^{-16}, 10^{-3}	70.56±0.57	80.68±0.41	61.68±0.49	63.68±0.45	2^{-16}, 10^{-3}
LRAE	71.50±1.17	83.50±0.18	67.14±0.71	66.66±1.39	0.1, 0.5	73.44±2.02	82.80±0.70	65.84±0.86	64.27±1.33	3, 0.1
DEC	59.86±0.65	73.46±0.64	52.87±0.70	55.47±0.88	-	57.00±0.95	71.86±0.80	49.15±0.81	51.97±0.81	-
IDEC	62.33±0.67	73.95±0.64	52.73±0.79	55.25±0.88	0.1	58.47±0.86	72.48±0.80	49.74±0.92	52.47±0.92	0.2

Keras Implementation by the IDEC authors. However, one will see that IDEC is still superior to DEC on such as the LBP feature and the BF0502 data set.

2) On MNIST: We also investigate the performance of DSC-L1 by using the MNIST data set.

Tables III and IV show the result, from which one could obverse that the ACC of DSC-L1 with the DSIFT feature is 72.65%, which improves SSC by 10.20% and the best baseline algorithm by 3.50%. With respect to the other three features, the improvement of DSC-L1 compared with all the baseline approaches is also significant, which is 1.82%, 1.02%, and 1.71% in terms of ARI. It should be pointed out that all the tested methods perform very stable on this data set, whose standard deviations on these four performance metrics are close to 0.

C. Comparisons on Raw Data

In this section, we carry out experiments on raw data from the full COIL20 and COIL100 data sets and compare DSC with a very recently proposed deep-learning-based method, i.e., DSCN [12]. As it has been shown that DSCN significantly outperforms the popular methods, including LRR, LRSC, SSC, DAE+SSC, and KSSC on these two data sets [12], we only compare the performance of DSCN, DSC, and the corresponding DAE.

For fair comparisons, we directly adopt the experimental setting used in [12]. In detail, the deep learning library is TensorFlow, the optimizer is Adam [75], the CNN with ReLU is used as the backbone, and the same network structure is used for DSC and DSCN. The network consists of two convolutional layers and one fully connected layer. The kernel size is set to 3 for COIL20 and 5 for COIL100, and the hidden fully connected layer size is fix to 15 for COIL20 and 50 for COIL100, respectively. In the horizontal and vertical directions, the stride is fixed to 2. Moreover, the well-tuned parameters for DSCN are also used for our DSC. In other words, we do not specifically tune hyperparameters of our model on the evaluated two data sets. As DSCN has two variants that are based on the ℓ_1- and ℓ_2-norm, we also develop two variants of DSC by considering these two norms as discussed in Section III-B, denoted by DSC-L1 and DSC-L2.

TABLE III

CLUSTERING RESULTS ON THE MNIST DATA SET. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Features	DSIFT					HOG				
Methods	ACC	NMI	ARI	Fscore	Para.	ACC	NMI	ARI	Fscore	Para.
DSC-L1	**72.65±0.00**	**70.42±0.00**	**61.80±0.00**	**65.79±0.00**	2^{-13}, 20	**78.10±0.00**	**77.51±0.00**	**68.72±0.00**	**72.03±0.00**	2^{-17}, 30
SSC	62.45±0.00	65.75±0.00	53.75±0.00	58.81±0.00	1, 10^{-2}	77.35±0.00	75.70±0.00	66.90±0.00	70.23±0.00	10, 10^{-2}
KSSC1	50.90±0.00	49.75±0.00	35.28±0.00	41.80±0.00	10^{-3}, 10	66.90±0.00	70.20±0.00	56.79±0.00	61.35±0.00	10^{-2}, 12
KSSC2	60.80±0.00	63.96±0.00	50.81±0.00	56.26±0.00	10^{-3}, 10	68.00±0.00	70.74±0.00	57.69±0.00	62.12±0.00	10^{-2}, 12
DAEg	52.55±0.00	58.36±0.00	40.98±0.00	47.48±0.00	1, 10^{-2}	23.36±0.66	11.56±0.78	5.83±0.28	15.89±0.32	10, 10^{-2}
DAEs	42.28±1.19	48.70±0.56	31.70±0.31	39.85±0.22	1, 10^{-2}	23.07±1.14	10.48±0.52	4.91±0.31	15.02±0.29	10, 10^{-2}
LRR	63.20±0.00	68.34±0.00	54.11±0.00	59.48±0.00	0.05	73.30±0.00	74.49±0.00	63.20±0.00	67.12±0.00	0.01
KLRR1	57.05±0.00	57.97±0.00	44.63±0.00	50.70±0.00	3000	72.15±0.00	70.95±0.00	61.38±0.00	65.42±0.00	30
KLRR2	22.63±0.89	12.55±1.37	9.02±0.46	22.98±0.14	1000	73.55±0.00	73.30±0.00	63.69±0.00	67.50±0.00	3
LRSC	59.30±0.00	58.84±0.00	46.90±0.00	52.37±0.00	0.1	61.20±0.00	59.65±0.00	47.05±0.00	52.59±0.00	0.01
LSR1	63.50±0.00	60.39±0.00	49.02±0.00	54.29±0.00	0.1	58.42±0.09	56.41±0.07	44.85±0.12	50.79±0.11	0.2
LSR2	63.55±0.00	60.53±0.00	49.14±0.00	54.39±0.00	0.4	60.40±0.02	57.78±0.00	46.45±0.00	51.98±0.00	0.1
SMR	69.15±0.00	68.90±0.00	59.17±0.00	63.40±0.00	2^{-16}, 10^{-3}	77.22±0.05	77.25±0.00	66.85±0.00	71.11±0.00	2^{-16}, 10^{-3}
LRAE	58.87±0.22	65.49±0.60	52.32±0.34	62.11±0.30	1, 0.5	69.80±0.07	75.66±0.60	62.26±0.27	69.08±0.25	8, 0.01
DEC	65.14±0.13	61.63±0.12	50.17±0.16	55.49±0.08	-	75.58±0.10	70.82±0.13	62.46±0.10	66.52±0.04	-
IDEC	62.93±0.01	58.67±0.01	48.53±0.01	53.93±0.01	0.2	62.87±0.01	60.72±0.00	47.34±0.02	52.97±0.01	0.1

TABLE IV

CLUSTERING RESULTS ON THE MNIST DATA SET. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Features	LBP					LPQ				
Methods	ACC	NMI	ARI	Fscore	Para.	ACC	NMI	ARI	Fscore	Para.
DSC-L1	**61.70±0.00**	**54.23±0.00**	**44.54±0.00**	**50.25±0.00**	2^{-15}, 40	**65.04±0.02**	**54.85±0.01**	**46.38±0.00**	**52.04±0.00**	2^{-16}, 70
SSC	59.75±0.00	53.83±0.00	43.52±0.00	49.02±0.00	0.1, 0.01	62.35±0.00	53.86±0.00	44.67±0.00	50.42±0.00	1, 0.01
KSSC1	58.50±0.00	53.96±0.00	41.58±0.00	47.67±0.00	0.1, 10	44.00±0.00	34.97±0.00	23.49±0.00	31.26±0.00	10^{-3}, 16
KSSC2	57.70±0.00	54.27±0.00	41.95±0.00	47.96±0.00	0.1, 10	54.99±0.03	51.07±0.01	36.62±0.01	43.21±0.01	10^{-3}, 16
DAEg	36.20±0.00	27.81±0.00	16.68±0.00	25.42±0.00	0.1, 0.01	30.61±0.51	22.05±0.32	12.19±0.05	22.07±0.05	10, 0.01
DAEs	32.20±0.00	24.85±0.00	14.75±0.00	23.39±0.00	0.1, 0.01	34.10±0.00	22.71±0.02	12.41±0.01	22.43±0.01	10, 0.01
LRR	55.70±0.00	45.70±0.00	37.77±0.00	44.58±0.00	0.5	52.15±0.00	50.63±0.00	37.86±0.00	44.63±0.00	0.5
KLRR1	54.12±0.06	50.98±0.01	37.84±0.07	44.25±0.06	1000	55.60±0.00	51.66±0.00	38.42±0.00	44.77±0.00	3000
KLRR2	53.75±0.00	50.70±0.00	37.05±0.00	43.55±0.00	300	56.75±0.00	51.69±0.00	38.87±0.00	45.16±0.00	1000
LRSC	42.45±0.00	35.60±0.00	23.42±0.00	31.43±0.00	0.03	53.20±0.00	42.49±0.00	32.03±0.00	39.04±0.00	0.05
LSR1	53.12±0.05	45.81±0.05	35.27±0.01	41.87±0.01	0.1	52.60±0.00	46.93±0.00	34.93±0.00	41.71±0.00	1
LSR2	52.93±0.04	45.65±0.03	34.98±0.05	41.61±0.05	0.1	53.25±0.05	47.57±0.11	35.54±0.06	42.28±0.06	1
SMR	49.90±0.00	44.29±0.00	32.16±0.00	39.18±0.00	2^{-14}, 10^{-3}	48.90±0.00	43.43±0.00	30.18±0.00	37.62±0.00	2^{-14}, 10^{-3}
LRAE	53.15±0.00	48.90±0.00	39.71±0.00	48.45±0.00	5, 0.1	53.26±0.47	49.83±0.51	38.00±0.25	43.80±0.22	2, 10
DEC	52.02±0.06	44.72±0.02	34.59±0.02	41.90±0.07	-	52.95±0.05	45.92±0.08	34.91±0.08	41.91±0.06	-
IDEC	52.40±0.24	45.53±0.06	35.04±0.18	41.71±0.15	0.1	50.13±0.03	41.80±0.01	30.67±0.01	37.75±0.01	0.1

TABLE V

COMPARISONS ON RAW DATA. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05. THE RESULT ON THE LAST TWO ROWS ARE DIRECTLY REFERRED FROM [12]

Methods	COIL20				COIL100			
	ACC	NMI	ARI	Fscore	ACC	NMI	ARI	Fscore
DSC-L1	**98.01±1.02**	**97.91±1.06**	**95.13±0.94**	**95.98±1.10**	**69.63±1.13**	**87.51±1.07**	58.35±0.97	59.15±0.87
DSC-L2	95.82±0.81	97.02±0.88	92.68±0.71	92.66±0.83	69.03±0.92	86.75±0.97	**60.01±0.84**	**59.91±0.79**
DSCN-L1	95.12±1.21	96.81±1.16	93.24±1.31	93.52±1.38	67.61±1.09	85.02±1.10	57.89±1.08	57.95±0.92
DSCN-L2	96.17±1.10	96.83±0.98	93.22±1.27	93.61±1.28	67.68±1.11	85.91±1.30	57.76±0.94	57.69±1.07
DAE	79.25±2.31	79.45±1.78	76.67±2.52	75.88±1.97	43.03±2.17	52.12±1.83	35.28±2.10	38.06±2.35
DSCN-L1 [12]	93.05	-	-	-	66.38	-	-	-
DSCN-L2 [12]	94.86	-	-	-	69.04	-	-	-

Besides our implementation results, Table V also reports the performance of DSCN demonstrated in [12]. One could see that our methods achieve the best result on these two data sets. In terms of ACC, for example, DSC is $+1.84\%$ at least higher than DSCN on COIL20 and the corresponding gap on COIL100 is about $+1.95\%$.

D. Deep Model Versus Shallow Models

In this section, we investigate the influence of the depth of DSC-L1 using the AR data set with DSIFT and HOG features. More specifically, we report the performance of DSC-L1 with three hidden layers ($M = 3$), two hidden layers ($M = 2$), and one hidden layer ($M = 1$). In the case $M = 3$, the number of new hidden neurons is set as 200. In the case $M = 1$, the number of hidden neurons is set as 150. Note that KSSC1 and KSSC2 can be regarded as two shallow models of SSC with one nonlinear hidden layer.

Table VI shows the clustering results of the tested methods, as well as the tuned parameters. We observe that our DSC-L1 ($M = 2$) consistently outperform the shallow models in

TABLE VI

DEEP VERSUS SHALLOW MODELS ON THE AR DATA SET. RESULTS IN BOLDFACE ARE SIGNIFICANTLY BETTER THAN THE OTHERS, ACCORDING TO THE T-TEST WITH A SIGNIFICANCE LEVEL AT 0.05

Features	DSIFT					HOG				
Methods	ACC	NMI	ARI	Fscore	Para.	ACC	NMI	ARI	Fscore	Para.
DSC-L1 (M=3)	84.68±1.10	94.17±0.33	80.99±1.30	80.26±1.18	2^{-10}, 20	85.03±1.55	93.59±0.57	79.32±1.31	78.18±0.70	2^{-9},10
DSC-L1 (M=2)	**85.38±1.08**	**95.17±0.17**	**82.15±0.63**	**82.35±0.62**	2^{-11}, 50	**85.05±1.53**	**94.36±0.43**	**78.98±1.58**	**79.21±1.56**	2^{-12}, 30
DSC-L1 (M=1)	83.81±1.72	94.57±0.45	81.23±1.94	81.42±1.92	2^{-11}, 30	81.90±0.96	91.93±0.35	71.87±1.97	72.17±1.95	2^{-12}, 20
SSC	74.83±1.27	89.91±0.38	66.43±1.44	66.81±1.42	10^{-2}, 10^{-3}	81.65±1.18	92.48±0.41	74.23±1.76	74.52±1.74	0.5, 10^{-3}
KSSC1	70.27±1.66	87.29±0.53	58.61±1.78	59.08±1.76	1, 18	83.12±0.90	93.07±0.34	75.68±1.37	75.94±1.36	10^{-2}, 20
KSSC2	78.28±1.78	91.55±0.39	71.13±1.44	71.44±1.43	1, 18	83.22±1.34	92.71±0.32	74.56±1.06	74.84±1.05	10^{-2}, 20
DAEg	74.37±1.20	89.53±0.43	65.42±1.56	65.81±1.54	10^{-2}, 10^{-3}	74.67±1.25	89.07±0.49	63.77±1.52	64.17±1.50	0.5, 10^{-3}
DAEs	72.65±0.91	88.54±0.52	62.23±1.81	62.67±1.78	10^{-2}, 10^{-3}	73.32±1.31	88.17±0.43	61.12±1.42	61.56±1.40	0.5, 10^{-3}
LRR	82.67±1.00	93.48±0.33	77.33±1.37	77.60±1.35	0.1	83.00±1.36	93.27±0.46	77.34±3.21	77.61±3.16	0.01
KLRR1	79.92±1.52	91.56±0.51	71.08±2.13	71.42±2.10	300	83.92±1.26	93.00±0.45	77.49±1.49	77.73±1.47	100
KLRR2	23.08±0.36	52.01±0.62	5.31±0.24	6.73±0.24	100	76.07±1.69	88.78±0.74	63.93±2.67	64.34±2.63	5
LRSC	83.55±1.20	92.84±0.37	78.33±1.39	78.57±1.38	0.06	83.42±1.43	92.67±0.48	73.86±1.73	74.15±1.71	0.02
LSR1	82.43±1.31	92.69±0.49	74.94±1.87	75.22±1.85	0.3	83.32±1.70	92.45±0.49	73.11±2.24	73.40±2.21	0.8
LSR2	82.45±1.58	92.64±0.42	74.49±1.80	74.77±1.78	0.7	83.65±1.07	92.45±0.45	73.24±1.77	73.54±1.75	1
SMR	71.07±1.91	87.01±0.52	60.82±2.22	61.26±2.19	2^{-16}, 10^{-2}	81.38±0.73	91.75±0.27	72.51±0.85	72.81±0.84	2^{-15}, 10^{-2}
LRAE	78.20±0.95	87.10±0.49	73.82±1.20	75.49±1.72	10^{-2}, 1e-3	83.53±1.58	93.40±0.47	76.89±1.40	76.25±1.43	0.50, 10^{-3}
DEC	56.90±0.53	79.47±0.55	43.97±0.50	44.59±0.50	-	65.83±0.56	84.20±0.52	55.16±0.61	55.63±0.61	-
IDEC	53.43±0.56	77.93±0.61	41.47±0.63	42.05±0.53	0.1	59.04±0.56	80.58±0.61	46.40±0.52	46.96±0.46	0.1

terms of all of these evaluation metrics. The results also verify our claim and motivation, i.e., our deep model DSC-L1 significantly benefit from deep learning.

E. Influence of Different Activation Functions

In this section, we investigate the influence of different nonlinear activation functions in DSC-L1. The investigated functions are sigmoid, nonsaturating sigmoid (nssigmoid), the ReLu [76], and the leaky relu [77]. We carry out th experiment on the BF0502 data set that contains the facial images detected from the TV series *Buffy the Vampire Slayer.*

From Table VII, one can observe that DSC-L1 with different activation functions outperforms SSC by a considerable performance margin. With the sigmoid function, DSC-L1 is about 3.17%, 4.18%, 9.32%, and 2.96% higher than SSC in terms of ACC, NMI, ARI, and Fscore, respectively. It is worth noting that although tan*h* is not the best activation function, it is more stable than the other activation functions in our experiments. Thus, we use the tan*h* function as the activation function for comparisons, as shown in Section V-B.

F. Convergence Analysis and Time Cost

In this section, we examine the convergence speed and time cost of our DSC-L1 on the BF0502 data set. From Fig. 2, we can see that the loss of DSC-L1 generally keeps unchanged after 90–100 epochs, i.e., achieving the convergence. For each epoch, DSC-L1 takes about 2.2 s to obtain the results on a MacBook with a 2.6-GHz Intel Core i5 CPU and 8-GB memory. Like other deep-learning-based methods, the computational cost of DSC-L1 can be remarkably reduced by GPU.

VI. CONCLUSION

In this article, we proposed a new deep-learning-based framework (i.e., DSC) for simultaneous data representation learning and subspace clustering. Experimental results show the efficacy of our method on the facial, object, and handwritten digit image data set in terms of four performance evaluation metrics.

TABLE VII

INFLUENCE OF DIFFERENT ACTIVATION FUNCTIONS OF DSC-L1 ON THE BF0502 DATABASE

Methods	ACC	NMI	ARI	Fscore	Para.
DSC-L1 (tanh)	79.50	71.02	65.11	71.09	2^{-13},90
DSC-L1 (sigmoid)	**82.67**	**79.01**	**71.69**	66.55	2^{-17},60
DSC-L1 (nssigmoid)	75.08	67.72	59.17	**72.11**	2^{-17},10
DSC-L1 (relu)	80.08	75.60	65.67	**72.11**	2^{-17},10
DSC-L1 (leaky relu)	79.50	72.21	62.24	68.10	2^{-17}, 20
SSC	79.50	74.83	62.37	69.15	1.0.2
KSSC1	74.50	71.99	61.95	68.85	0.1,12
KSSC2	77.83	69.89	70.65	70.55	0.1,12
DAEg	55.50	38.16	30.69	43.15	-
DAEs	21.67	6.07	0.85	28.65	-
LRR	78.17	74.89	70.57	70.58	0.01
KLRR1	75.33	66.60	56.83	64.07	3
KLRR2	75.00	69.32	68.35	74.16	3
LRSC	69.17	60.60	53.28	61.71	0.01
LSR1	67.50	57.53	51.36	60.19	1.00
LSR2	77.00	59.91	56.27	63.60	0.50
SMR	76.00	74.69	58.09	71.87	2^{-16},10^{-2}
LRAE	75.33	74.91	72.66	69.85	0.01, 10
DEC	77.57	68.25	61.41	68.11	-
IDEC	80.73	70.21	61.99	68.52	0.1

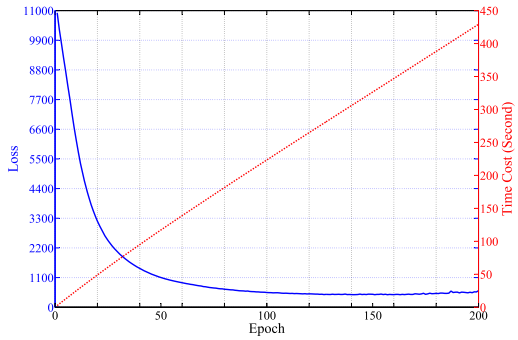

Fig. 2. Convergence curve and time cost of DSC-L1. The left y-axis indicates the loss at each epoch and the right one is the total time cost taken by our method.

In the future, we plan to investigate the performance of our proposed framework when adopting other loss/regularization functions and extend our proposed framework for other

applications, such as weakly supervised learning. Furthermore, although we have proved the convergence property of DSC, the proof is based on the popular two-layer network setting. For more complex cases such as multilayer network, the convergence property still remains opening and challenging.

ACKNOWLEDGMENT

The authors would like to thank the anonymous associate editor and reviewers for their valuable comments and constructive suggestions to improve the quality of this article.

REFERENCES

[1] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[2] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[3] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," in *Proc. 23th Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, Jun. 2006, pp. 593–600.

[4] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. 21th IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[5] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2001, pp. 849–856.

[6] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[7] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. 24th Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 55–63.

[8] 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.

[9] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3818–3825.

[10] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.

[11] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3834–3841.

[12] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. 29th Adv. Neural Inf. Process. Syst.*, Montréal, QC, Canada, Dec. 2017, pp. 24–33.

[13] 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.

[14] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 347–360.

[15] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2014.

[16] R. He, Y. Zhang, Z. Sun, and Q. Yin, "Robust subspace clustering with complex noise," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4001–4013, Nov. 2015.

[17] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. 29th IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 3918–3927.

[18] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "l_0-sparse subspace clustering," in *Proc. 14th Euro. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 731–747.

[19] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[20] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 3827–3833.

[21] 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.

[22] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear-norm and frobenius-norm-based representations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 218–224, Jan. 2018.

[23] V. Patel, H. V. Nguyen, and R. Vidal, "Latent space sparse subspace clustering," in *Proc. 14th IEEE Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 225–232.

[24] V. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process.*, Paris, Oct. 2014, pp. 2849–2853.

[25] 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.

[26] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *Proc. 29th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5157–5164.

[27] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4032–4044, Aug. 2019.

[28] G.-S. Xie, X.-Y. Zhang, S. Yan, and C.-L. Liu, "Hybrid CNN and dictionary-based models for scene recognition and domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1263–1274, Jun. 2017.

[29] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc 36th Int Conf Mach. Learn.*, Long Beach, CA, USA, vol. 97, Jun. 2019, pp. 5092–5101.

[30] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 430–437.

[31] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[32] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.

[33] W. Zhang, X. Lu, and X. Li, "A coarse-to-fine semi-supervised change detection for multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3587–3599, Jun. 2018.

[34] G.-S. Xie, X.-Y. Zhang, S. Yan, and C.-L. Liu, "SDE: A novel selective, discriminative and equalizing feature representation for visual recognition," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 145–168, Sep. 2017.

[35] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[36] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Commun. ACM*, vol. 54, no. 10, pp. 95–103, Oct. 2011.

[37] Z. Y. Wang, Q. Ling, and T. S. Huang, "Learning deep l0 encoders," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 2194–2200.

[38] G.-S. Xie *et al.*, "Attentive region embedding network for zero-shot learning," in *Proc. 33th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9384–9393.

[39] M. Shao, S. Li, Z. Ding, and Y. Fu, "Deep linear coding for fast graph clustering," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3798–3804.

[40] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33th Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 478–487.

[41] X. Peng, S. Xiao, J. Feng, W. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, Jul. 2016, pp. 1925–1931.

[42] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang, "Learning a task-specific deep architecture for clustering," in *Proc. SIAM Int. Conf. Data Mining*, Miami, FL, USA, May 2015, pp. 369–377.

[43] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. 29th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5147–5156.

[44] Y. Chen, L. Zhang, and Z. Yi, "Subspace clustering using a low-rank constrained autoencoder," *Inf. Sci.*, vol. 424, pp. 27–38, Jan. 2018.

[45] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[46] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[47] T. Zhang, G. Su, C. Qing, X. Xu, B. Cai, and X. Xing, "Hierarchical lifelong learning by sharing representations and integrating hypothesis," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, doi: 10.1109/TSMC.2018.2884996.

[48] Y. Zhang, F.-L. Chung, and S. Wang, "Fast reduced set-based exemplar finding and cluster assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 917–931, May 2019.

[49] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[50] J. Peña, J. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the K-means algorithm," *Pattern Recognit. Lett.*, vol. 20, no. 10, pp. 1027–1040, Oct. 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865599000690

[51] 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, Jan. 2011.

[52] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. 25th Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 612–620.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, Dec. 2012, pp. 1097–1105.

[54] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks Trade, Reloaded*, 2nd ed. (Lecture Notes in Computer Science), vol. 7700. Berlin, Germany: Springer, 2012.

[55] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2010-13, Feb. 2010.

[56] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1875–1882.

[57] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. 28th IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 815–823.

[58] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. 22nd Int Conf. Pattern Recognit.*, Stockholm, Sweden, Aug. 2014, pp. 34–39.

[59] S. Tao, D. Boley, and S. Zhang, "Local linear convergence of ISTA and FISTA on the LASSO problem," *SIAM J. Optim.*, vol. 26, no. 1, pp. 313–336, Jan. 2016.

[60] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *Ann. Appl. Stat.*, vol. 2, no. 1, pp. 224–244, Mar. 2008.

[61] K. Kawaguchi, "Deep learning without poor local minima," in *Proc. 30th Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 586–594.

[62] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Proc. 31th Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 1–11.

[63] A. M. Saxe, J. Mcclelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *Proc. 2nd Int. Conf. Learn. Rep.*, Banff, AB, Canada, Apr. 2014, pp. 1–22.

[64] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[65] S. A. Nene *et al.*, "Columbia object image library (coil-20)," Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.

[66] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[67] A. Martinez, "The AR face database," CVC, Barcelona, Spain, Tech. Rep., 1998, vol. 24.

[68] J. Sivic, M. Everingham, and A. Zisserman, "Who are you?—Learning person specific classifiers from video," in *Proc. 22th IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1145–1152.

[69] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.

[70] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[71] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[72] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. 18th IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, vol. 1, Jun. 2005, pp. 886–893.

[73] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[74] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Image and Signal Processing*. Springer, 2008, pp. 236–243.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3nd Int. Conf. Learn. Rep.*, San Diego, CA, USA, May 2015, pp. 1–22.

[76] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 807–814.

[77] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn. Workshop Deep Learn. Audio, Speech Lang. Process.*, Atlanta, GA, USA, Jun. 2013, pp. 1–6.

Xi Peng (Member, IEEE) is currently a Full Professor with the College of Computer Science, Sichuan University, Chengdu, China. His current research interest includes machine intelligence. He has authored more than 50 articles in this area.

Dr. Peng has served as an Associate Editor/Guest Editor for six journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and the IEEE TRANSACTIONS ON NEURAL NETWORK AND LEARNING SYSTEMS, and an Area Chair/Senior Program Committee Member for the conferences, such as International Joint Conference on Artificial Intelligence (IJCAI) and IEEE International Conference on Multimedia & Expo (ICME).

Jiashi Feng received the B.E. degree from the University of Science and Technology, Hefei, China, in 2007, and the Ph.D. degree from the National University of Singapore, Singapore, in 2014.

He was a Post-Doctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA, from 2014 to 2015. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering with the National University of Singapore, Singapore. His current research interests focus on machine learning and computer vision techniques for large-scale data analysis.

Joey Tianyi Zhou received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015.

He is currently a Scientist with the Institute of High Performance Computing, A*STAR, Singapore. His current research interests include differentiable programming, transfer learning, and sparse coding.

Dr. Zhou was a recipient of the Best Poster Honorable Mention at ACML 2012, the Best Paper Nomination at ECCV 2016, and the NIPS 2017 Best Reviewer Award.

Yingjie Lei received the M.S. degree in pattern recognition and image processing from Sichuan University (SCU), Chengdu, China, in 2009, and the Ph.D. degree in computer vision from The University of Western Australia (UWA), Perth, WA, Australia, in 2013.

He is currently an Associate Professor with the College of Electronics and Information Engineering, SCU. His research interests mainly include deep learning, saliency detection, 3-D biometrics, object recognition, and semantic segmentation. He has authored over 60 articles in these areas.

Shuicheng Yan (Fellow, IEEE) is currently the Chief Technology Officer of YITU Tech, China.

Dr. Yan is a fellow of International Association for Pattern Recognition (IAPR) and a Distinguished Scientist of the Association for Computing Machinery (ACM). His team received seven times winner or honorable mention prizes in five years over the PASCAL Visual Object Classes and the ImageNet Large Scale Visual Recognition Challenge competitions that are the core competitions in the field of computer vision, along with over ten times best (student) paper awards. He was a Thomson Reuters Highly Cited Researcher from 2014 to 2016.