Multi-Scale Fusion Subspace Clustering Using Similarity Constraint

Zhiyuan Dang[1], Cheng Deng[1]*, Xu Yang[1], Heng Huang[2,3]
[1]School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2]Department of Electrical and Computer Engineering, University of Pittsburgh, PA 15260, USA
[3]JD Finance America Corporation, Mountain View, CA 94043, USA
{zydang,xyang_01}@stu.xidian.edu.cn, chdeng@mail.xidian.edu.cn, heng.huang@pitt.edu

Abstract

Classical subspace clustering methods often assume that the raw form data lie in a union of the low-dimension linear subspace. This assumption is too strict in practice, which largely limits the generalization of subspace clustering. To tackle this issue, deep subspace clustering (DSC) networks based on deep autoencoder (DAE) have been proposed, which non-linearly map the raw form data into a latent space well-adapted to subspace clustering. However, existing DSC models ignore the important multi-scale information embedded in DAE, thus abandon the much more useful deep features, leading their suboptimal clustering results. In this paper, we propose the Multi-Scale Fusion Subspace Clustering Using Similarity Constraint (SC-MSFSC) network, which learns a more discriminative self-expression coefficient matrix by a novel multi-scale fusion module. More importantly, it introduces a similarity constraint module to guide the fused self-expression coefficient matrix in training. Specifically, the multi-scale fusion module is framed to generate the self-expression coefficient matrix of each convolutional layer in DAE and then fuses them with the convolutional kernel. In addition, the similarity constraint module is to supervise the fused self-expression coefficient matrix by the designed similarity matrix. Extensive experimental results on four benchmark datasets demonstrate the superiority of our new model against state-of-the-art methods.

1. Introduction

In recent years, subspace clustering [33] has aroused widespread research interests in unsupervised learning and is successfully exploited to various applications, such as image segmentation [22, 36], motion segmentation [14, 4], image clustering [40, 5, 37, 38, 11], genes expression microarray clustering [24] and so on. Subspace clustering aims to segment data drawn from a union of low-dimension subspaces in an unsupervised way, and actually, many data equip this property. For example, the face images of a subject taken under fixed pose, varying lighting conditions and Lambertian reflectance, occupy a low-dimension subspace whose dimension close to nine [2, 13], and the handwritten digit images of a single digit also form a low-dimension subspace [10]. Therefore, we can apply subspace clustering to segment the data into multiple groups according to whether they belong to the same subspace or not.

Majority of the subspace clustering methods [35, 3, 4, 19, 21] rely on the assumption that the raw form data locate in a union of low-dimension linear subspace. In fact, this assumption is too strict for some practical environments. For example, in face image clustering, the reflectance is normally non-Lambertian and the pose of the subject is not always fixed [13]. Under such situations, the images corresponding to the same face no longer lie in linear subspaces. Subsequently, kernel-based methods [30, 29, 39] are developed to implicitly map raw data into high-dimension spaces, expecting to address the problem of non-linear subspace embedding. However, it is difficult to choose proper kernel function and its corresponding hyper-parameter, and more importantly, there is no clear theoretical guarantee such kernel existing [43].

By virtue of the powerful deep learning [17], DAEs have been widely used to non-linearly transform data into latent space for unsupervised learning. DSC based on DAE is the latest work [13] in the subspace clustering field, which successfully makes the generated latent space well-adapted to subspace clustering and obtains promising results. Beyond that, according to the self-expression property of data that indicates a data point can be expressed as a linear combination of other data points in the same subspace, DSC [13] first introduces this property into the deep network and substitutes the self-expression coefficient matrix with the novel self-expression layer, whose weights are viewed as the coefficient matrix. In subsequent works of DSC [43, 45, 42], DAE is still used to extract features from input data. As

*Corresponding author.

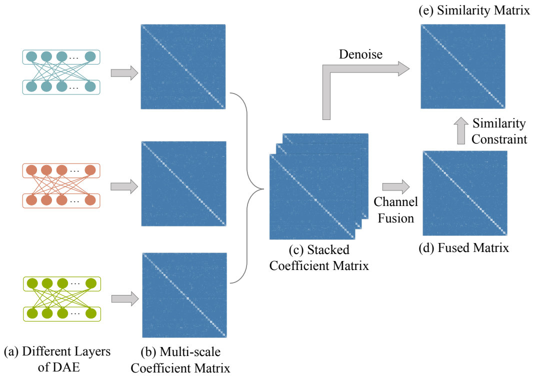

(e) Similarity Matrix
Denoise
Similarity Constraint
Channel Fusion
(c) Stacked Coefficient Matrix
(d) Fused Matrix
(a) Different Layers of DAE
(b) Multi-scale Coefficient Matrix

Figure 1. Illustration of our idea. Each layer in DAE could have itself self-expression coefficient matrix ((a) and (b)). We stack these matrices (c) and then fuse them by convolutional kernel (d). Afterwards, the fused coefficient matrix (d) could be supervised by the similarity matrix (e) obtained from the similarity constraint module. These above matrices are the real results from the experiment about the ORL dataset (40 classes). We notice that there is an explicit block-diagonal structure on matrices in (b), (d) and (e), which is the particular character of the self-expression coefficient matrix. Note that the final spectral clustering error is 2.00%.

we know, the shallower layers in DAE encoder learn more pixel-level or texture-level information, and the deeper ones extract more semantic-level or abstract-level information. Hence, the multi-scale features of input data have been inherently embedded in different layers of DAE.

However, these existing works only consider the features extracted from deeper layers, regardless of the helpful features in shallower ones and the fusion of multi-scale information embedded in DAE. Wasting plentiful off-the-shelf yet useful deep convolutional features, they always achieve unfavorable clustering performance. Since DSC merely considers the coefficient matrix of the deepest convolutional layer, we think that each layer exists in DAE encoder should have itself self-expression coefficient matrix (see Figure 1). In this way, the latent multi-scale information embedded in DAE has been passed to the corresponding self-expression coefficient matrix. Therefore, how to integrate these matrices and fully utilize the multi-scale information embedded in themselves is a challenging problem, which is vital to further improve the clustering performance of classical DSC networks.

In this paper, we propose Multi-Scale Fusion Subspace Clustering Using Similarity Constraint (SC-MSFSC) network that contains four modules: feature extraction module, self-expression module, multi-scale fusion module, and similarity constraint module. The feature extraction module, *i.e.*, DAE, is used to extract features from input data. The self-expression module is used to obtain a self-expression coefficient matrix for subspace clustering. And the two proposed novel modules: multi-scale fusion and similarity constraint module, are designed to fuse the multi-scale information extracted from different layers in DAE and stabilize the training process respectively.

Our major contributions can be summarized as follows:

- We propose a novel multi-scale fusion module that fuses the multi-scale information extracted from different layers in DAE, which is achieved by stacking the coefficient matrix extracted from different self-expression layers and then applying a convolutional kernel on the stacked coefficient matrix to fuse its channel. After that, we constraint the fused coefficient matrix with self-expression loss and reconstruction loss.

- We introduce a novel similarity constraint module that stabilizes the training process and supervises the fused coefficient matrix. In this module, we design a similarity matrix that is garnered by denoising the stacked coefficient matrix per channel and then averaging the values of entries. The similarity matrix is used to supervise the fused coefficient matrix in training process.

- Extensive experimental results on four benchmark datasets demonstrate the superiority of SC-MSFSC against other state-of-the-art methods.

2. Related Work

The current subspace clustering algorithms could be divided into two subproblems. The first subproblem is to estimate an affinity matrix from data, and the second one is to apply spectral clustering on affinity matrix [28]. These two subproblems could be optimized sequentially in one-pass [4, 20, 21] or optimized alternatively in multi-pass [18, 6]. Between these two subproblems, constructing a discriminative affinity matrix is more significant. The way to build affinity matrix could be roughly split into three categories: factorization based methods [9, 25], model based methods [3, 31], and self-expression based methods [4, 12, 19, 34].

Considering the robustness against noise and outliers, and the lower computation complexity compared to other competitors, self-expression methods have been the more popular in subspace clustering [35]. Most of the existing works rely on the linear subspace assumption, however, as mentioned above, this assumption is not tenable in practical problems. A few works have been proposed to solve the problem of non-linear subspace embedding by introducing a pre-defined kernel matrix (such as polynomial kernel and Gaussian RBF kernel) [30, 29, 39]. However, there is no definite indication of how to choose proper kernel function and whether the feature spaces generated by kernel tricks are suitable for linear subspace clustering or not.

Based on the powerful non-linear mapping ability of the DAE, DSC network [13] determines a subspace-friendly la-

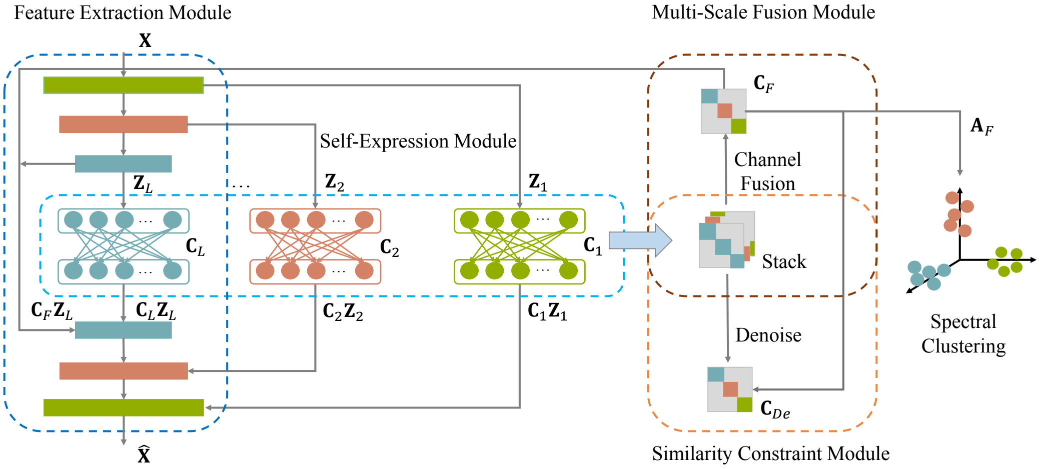


Figure 2. Structure of the proposed network: Multi-Scale Fusion Subspace Clustering Using Similarity Constraint (SC-MSFSC). As the figure is shown, our network consists of four modules: a) feature extraction module which is used to provide the learned multi-scale convolutional features, b) self-expression module which is used to learn the self-expression coefficient matrices of different layers in DAE and also make the fused coefficient matrix maintains the self-expression property, c) multi-scale fusion module which is used to obtain the fused coefficient matrix and also constrain this matrix through self-expression and reconstruction loss d) similarity constraint module which is used to generate similarity matrix from the average of denoised stacked multi-scale self-expression coefficient matrices and then supervise the fused coefficient matrix. Once the network is trained, we execute spectral clustering on the fused coefficient matrix.

tent space, which perfectly overcomes the limitation of the linear assumption. Furthermore, the main contribution of the DSC network is to design the novel self-expression layer and the related loss function that models the self-expression property of data into DAE. This self-expression layer skillfully represents the self-expression coefficient matrix as the weights of a fully-connected layer without any activation and bias [13]. Benefit from this novel layer, DSC greatly improves clustering performance on various datasets. There are some follow-up researches [45, 43, 42] to further ameliorate the performance of DSC. Deep adversarial subspace clustering [45] adopts a subspace-specific GAN based adversarial learning network to supervise the representation of samples. Zhang et al. [42] proposed a dual self-supervised convolutional network that utilizes the results of spectral clustering to supervise the learning process of classification module and self-expression module. Zhou et al. [44] introduced a distribution consistency loss to guide the learning of distribution-preserving latent representation. Reformulating the subspace clustering as a classification problem, [43] freed the spectral clustering step from the classical DSC network, which is a true sense of end-to-end framework.

In a nutshell, most of these previous works only consider supervising the final clustering results by either the generated deep features of DAE [45, 42, 44] or the refined self-expression coefficient matrix [43, 42]. Nonetheless, they both neglect the multi-scale information embed in different layers of DAE, which wastes lots of deep convolutional features beneficial to clustering. Unlike existing works, our proposed network not only supervises the fused self-expression coefficient matrix by the designed similarity matrix (which could be viewed as a kind of self-supervised mechanism) but also integrates the multi-scale information from the different layers of DAE.

Moreover, to the best of our knowledge, it is the first attempt to integrate the multi-scale information to a joint deep learning neural network framework in the subspace clustering field, which also lays the foundation for constructing a more discriminative affinity matrix in other unsupervised learning problems.

3. Multi-Scale Fusion Subspace Clustering Using Similarity Constraint (SC-MSFSC)

In this section, we describe our deep subspace clustering learning network, named SC-MSFSC. We first introduce our network formulation (see Figure 2) and then present an effective algorithm to optimize the proposed network. Specifically, our proposed network consists of four important modules, *i.e.,* feature extraction module, self-expression module, multi-scale fusion module, and similarity constraint module.

3.1. Feature Extraction Module

The foundational component of our SC-MSFSC is the feature extraction module, which non-linearly transforms the raw data into a latent space appropriate to subspace clustering. To extract more multi-scale features from DAEs, the convolutional version of DAEs are adopted as the backbone network. Given an input $\mathbf{X}$, the latent variable $\mathbf{Z}$ can be obtained through the encoder, and then feed $\mathbf{Z}$ into the decoder to gain the reconstructed input data $\hat{\mathbf{X}}$ (in fine-tune stage, we feed $\mathbf{CZ}$ into the decoder to gain the reconstructed input data, seen Section 3.5). To ensure that the learned latent variable $\mathbf{Z}$ could completely represent the input $\mathbf{X}$, the loss function of this auto-encoder network is set as:

$$\frac{1}{2}\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2. \tag{1}$$

Note that $\mathbf{X}, \hat{\mathbf{X}} \in \mathbf{R}^{N \times H \times W \times C}$, where N, H, W and C are the number, height, width and channel of the input data respectively.

3.2. Self-Expression Module

The self-expression module is used to obtain a self-expression coefficient matrix of the input latent variable $\mathbf{Z}$, which models the self-expression property of data into a fully-connected layer without any activation and bias [13, 42, 43]. When N data points are stacked into a data matrix $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N]$, $\mathbf{z}_i \in \mathbf{R}^d, i = 1, \cdots, N$, the self-expression property can be defined as the matrix multiply formulation, *i.e.,* $\mathbf{Z} = \mathbf{CZ}$, where $\mathbf{C} \in \mathbf{R}^{N \times N}$ is the self-expression coefficient matrix. As shown in [12], under the independent subspace assumption, if we regularize $\mathbf{C}$ with certain norms, $\mathbf{C}$ will have a block-diagonal structure, where a non-zero entry means that data point $\mathbf{z}_i$ and $\mathbf{z}_j$ lie in the same subspace. To obtain desirable block-diagonal coefficient matrix, the loss of the self-expression module consists of regularization term and self-expression term:

$$\|\mathbf{C}\|_p + \frac{1}{2}\|\mathbf{Z}-\mathbf{CZ}\|_F^2 \quad s.t. \quad (\mathrm{diag}(\mathbf{C}) = \mathbf{0}), \tag{2}$$

where $\|\cdot\|_p$ represents an arbitrary regularization norm, *e.g.*, ℓ_1 norm [4, 5], nuclear norm [20, 19], and Frobenius norm [12, 21]. Additionally, the optional diagonal constraint on $\mathbf{C}$ ($\mathrm{diag}(\mathbf{C}) = \mathbf{0}$) is used to prevent the trial solutions of $\mathbf{C} = \mathbf{I}$ for sparsity inducing norms, such as ℓ_1 norm [5]. $\mathbf{Z}$ indicates the latent variable matrix after the encoder. In view of the satisfied performance of Frobenius norm in [12, 21], therefore, we only adopt the Frobenius norm in our experimental settings.

3.3. Multi-Scale Fusion Module

The multi-scale fusion module is used to integrate the self-expression coefficient matrix of each convolutional layer in the encoder. Since current deep subspace clustering algorithms neglect the multi-scale information indwelled in the DAE, we design this module to exploit the multi-scale information for further improving the performance of DSC network. For clarity, suppose the input $\mathbf{X}$ after the first l-th layer in the encoder as $\mathbf{Z}_l \in \mathbf{R}^{N \times D_l}$ (D_l depends on the convolutional kernel size of the current l-th layer), and the latent variable matrix $\mathbf{Z}_l$ after the last l-th layer in the decoder as $\hat{\mathbf{X}}_i$. Assume that both the encoder and the decoder have L layers, the reconstruction loss function (1) could be extended as the multi-scale version of that:

$$\frac{1}{2}\sum_{l}^{L}\|\mathbf{X}_l-\hat{\mathbf{X}}_l\|_F^2. \tag{3}$$

where $l = 1, \cdots, L$. Through multi-scale reconstruction loss (3), the scale information embedded in the DAE have been passed to latent variable matrix $\mathbf{Z}_l$. In order to completely use the information existing in each latent variable matrix $\mathbf{Z}_l$, we generalize the self-expression loss function (2) to the multi-scale scenario. Therefore, we formulate it as follows:

$$\sum_{l}^{L}\|\mathbf{C}_l\|_p + \frac{1}{2}\sum_{l}^{L}\|\mathbf{Z}_l-\mathbf{C}_l\mathbf{Z}_l\|_F^2 \quad s.t. \quad (\mathrm{diag}(\mathbf{C}_l) = \mathbf{0}). \tag{4}$$

For a smaller l, the learned coefficient matrix $\mathbf{C}_l$ holds more pixel-level information, and oppositely, for a larger l, $\mathbf{C}_l$ holds more semantics-level information. Considering different coefficient matrix $\mathbf{C}_l$ possessing various information about the input data, it is better to fuse these matrix $\mathbf{C}_l$ into a more discriminative coefficient matrix $\mathbf{C}_F$.

In fact, how to combine these multi-scale coefficient matrix $\mathbf{C}_l$ is challenging. The naive method is to obtain $\mathbf{C}_S \in \mathbf{R}^{N \times N \times L}$ by stacking $\mathbf{C}_l$ along channel dimension, and then average (or sum) each entry of $\mathbf{C}_S$ over channel dimension. However, in such a method, the more discriminative coefficient matrix and the less one will be treated equally. Thus, this solution is not optimal. Another choice is to learn a common self-expression coefficient matrix $\mathbf{C}_c$ from multi-scale information in DAE, *i.e.,* $\|\mathbf{C}_c\|_p + \frac{1}{2}\sum_{l}^{L}\|\mathbf{Z}_l - \mathbf{C}_c\mathbf{Z}_l\|_F^2$. However, applying more constraints over $\mathbf{C}_c$ may destabilize the training process and thus leads to worser clustering results (see Table 5).

As a matter of fact, it is better to use a convolutional kernel k to integrate the channels of the $\mathbf{C}_S$, $\mathbf{C}_F \in \mathbf{R}^{N \times N} = k \bigotimes \mathbf{C}_S$, where $\bigotimes$ means the convolutional operation. If we adopt a proper kernel size, $\mathbf{C}_F$ will could capture more local information on each $\mathbf{C}_l$ due to block-diagonal structure. This statement is proofed in the different experimental settings about ORL dataset (see Table 5). Obviously, $\mathbf{C}_F$ should also be applied to the self-expression loss (2):

$$\|\mathbf{C}_F\|_p + \frac{1}{2}\|\mathbf{Z}-\mathbf{C}_F\mathbf{Z}\|_F^2 \quad s.t. \quad (\mathrm{diag}(\mathbf{C}_F) = \mathbf{0}), \tag{5}$$

where $\mathbf{Z}$ is the more discriminative one in (4), *i.e.*, $\mathbf{Z}_L$. Besides that, we also rewrite the reconstruction loss of the deepest layer L, $\frac{1}{2}\|\mathbf{X}_L - \hat{\mathbf{X}}_L\|_F^2$, as that of the fused coefficient matrix:

$$\frac{1}{2}\|\mathbf{X}_L-\hat{\mathbf{X}}_F\|_F^2, \tag{6}$$

where $\hat{\mathbf{X}}_F$ is the output of the decoder when $\mathbf{C}_F\mathbf{Z}_L$ are fed.

3.4. Similarity Constraint Module

The similarity constraint module is used to supervise the fused coefficient matrix obtained from the multi-scale fusion module. Since the self-expression coefficient matrix contains noise, which will influence the optimizing process, we design a similarity constraint loss to stabilize the training and keep the coefficient matrix discriminative.

Recently, a thresholding method has been adopted to denoise the obtained coefficient matrix in literatures [5, 12, 13], *i.e.*, given a threshold value $\alpha \in (0, 1)$, for each column of self-expression coefficient matrix, we only keep the values of entries whose summation is α of the summation of the whole column, and set the values of other entries 0. We denote the stacked coefficient matrix $\mathbf{C}_S$ after denoising as $\mathbf{C}_{S\_De}$. Obviously, $\mathbf{C}_{S\_De}$ is more sparse than $\mathbf{C}_S$.

To avoid extra computations, we set the average of $\mathbf{C}_{S\_De}$ as the similarity matrix $\mathbf{C}_{De}$. Since there is some information lost in the thresholding method, we set $\mathbf{C}_{De}$ as the target, and enforce $\mathbf{C}_F$ to approximate it. Based on these intuitions, we propose the similarity constraint loss:

$$\|\mathbf{C}_{De} - \mathbf{C}_F\|_F^2. \tag{7}$$

As we mentioned above, we hope $\mathbf{C}_F$ will lie in the intermediate state between the output of the multi-scale fusion module and the similarity constraint module. With this loss, the obtained $\mathbf{C}_F$ not only have sparse entries but keep some information lost in the denoising procedure. The ablation study in Table 2 shows that similarity constraint loss successfully guides $\mathbf{C}_F$ to a proper intermediate state.

3.5. Training Settings

Similar to [13, 42], we train the proposed SC-MCFSC network with a two-stage strategy: 1) pre-train the stacked autoencoder without self-expression layer; 2) train the whole network with all previously mentioned modules.

1) Pre-train Stage. In order to obtain good enough latent variables to represent the input data, and reduce the reconstruction difficulty in the later fine-tune stage, we only to minimize the reconstruction loss (1) in this stage.

The coefficient matrices $\mathbf{C}_l$ are set as an identity matrix, which equals to train the whole network without the self-expression layer. The following is the loss function used in pre-train stage:

$$\mathcal{L}_{pre} = \mathcal{L}_0, \tag{8}$$

where $\mathcal{L}_0 = \frac{1}{2}\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$.

2) Fine-tune Stage. Benefit from the pre-trained DAE, we could constrain the coefficient matrices by modifying the reconstruction loss (1), *i.e.*, feed $\mathbf{CZ}$ instead of $\mathbf{Z}$ into the decoder of the DAE.

The total loss function in fine-tune stage is:

$$\mathcal{L}_{fine} = \hat{\mathcal{L}}_0 + \lambda_1\hat{\mathcal{L}}_1 + \lambda_2\hat{\mathcal{L}}_2 + \lambda_3\mathcal{L}_3 + \lambda_4\mathcal{L}_4, \tag{9}$$

Algorithm 1 Fine-tune Stage of Training SC-MSFSC Network, *i.e.,* optimize the loss function (9).

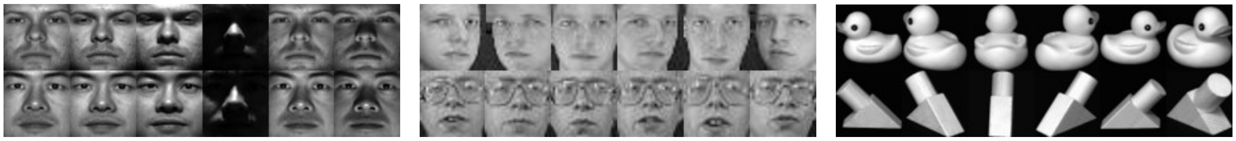
```
Input: Input data, tradeoff parameters, maximum iteration
    $T$, $t = 1$, and pre-trained DAE.
Initialization the weights of the multi self-expression
    layers with constant $10^{-4}$.
where $t < T$ :
    Obtain latent variable matrix $\mathbf{Z}_l, l = 1, \cdots, L$ from
    different layers of DAE.
    Obtain multi-scale self-expression coefficient matrix $\mathbf{C}_l, l = 1, \cdots, L$ from different layers of DAE
    through optimizing the rewrite reconstruction loss
    (3) and self-expression loss (4).
    Stack the obtained multi-scale self-expression coefficient matrices as $\mathbf{C}_S$.
    Obtain the fused self-expression coefficient matrix
    $\mathbf{C}_F$ by applying convolutional kernel on $\mathbf{C}_S$.
    Obtain the similarity matrix $\mathbf{C}_{De}$ by denoising and
    averaging over $\mathbf{C}_S$.
    Supervise $\mathbf{C}_F$ by $\mathbf{C}_{De}$, see similarity constraint loss
    (7). Optimize the fused self-expression loss (5) and
    reconstruction loss (6) to constrain $\mathbf{C}_F$. Then set $t =$
    $t + 1$.
end while
Output: $\mathbf{C}_F$.
```

where $\hat{\mathcal{L}}_0 = \frac{1}{2}\sum_l^L \|\mathbf{X}_l - \hat{\mathbf{X}}_l\|_F^2 + \frac{1}{2}\|\mathbf{X}_L - \hat{\mathbf{X}}_F\|_F^2$ is the modified reconstruction losses of the multi-scale scenarios (3) and the fused self-expression coefficient matrix (6), $\hat{\mathcal{L}}_1 = \sum_l^L \|\mathbf{C}_l\|_p + \|\mathbf{C}_F\|_p$ is the regularization loss, $\hat{\mathcal{L}}_2 = \frac{1}{2}\sum_l^L \|\mathbf{Z}_l - \mathbf{C}_l\mathbf{Z}_l\|_F^2$ is the multi-scale self-expression loss, $\mathcal{L}_3 = \frac{1}{2}\|\mathbf{Z} - \mathbf{C}_F\mathbf{Z}\|_F^2$ is the self-expression loss about the fused coefficient matrix $\mathbf{C}_F$ and $\mathcal{L}_4 = \|\mathbf{C}_{De} - \mathbf{C}_F\|_F^2$ is the similarity constraint loss (7). $\lambda_1, \lambda_2, \lambda_3,$ and λ_4 are the tradeoff parameters for the above loss function.

Once the network is trained, we could use the final fused coefficient matrix $\mathbf{C}_F$ to construct an affinity matrix $\mathbf{A}_F$ for spectral clustering [28]. Before the clustering step, we adopt a heuristic, same to the previous works [4, 12, 13], to further enhance the block-structure and improve the final clustering accuracy.

4. Experiments

Our SC-MSFSC network is implemented by Tensorflow [1] and optimized by ADAM [15]. To assess its performance, we design extensive experiments on four benchmark datasets: two face image datasets, the Extended Yale B [7] and ORL [32]; and two object image datasets, COIL20/100 [27, 26]. The following baselines will be compared against our SC-MSFSC network: Low Rank Representation (LRR) [20], Low Rank Subspace Cluster-



(a) Extended Yale B dataset (b) ORL dataset (c) COIL 20/100 dataset

Figure 3. Example images of the four benchmark datasets

	Extended Yale B		ORL	
Layers	Kernel Size	Channels	Kernel Size	Channels
encoder-1	5×5	10	5×5	5
encoder-2	3×3	20	3×3	3
encoder-3	3×3	30	3×3	3
decoder-3	3×3	30	3×3	3
decoder-2	3×3	20	3×3	3
decoder-1	5×5	10	5×5	5

Table 1. Network structures for Extended Yale B and ORL.

Losses	Extended Yale B (38 subjects)	ORL
$\mathcal{L}_0+\mathcal{L}_1+\mathcal{L}_2$ (DSC)	2.67	14.00
$\hat{\mathcal{L}}_0+\hat{\mathcal{L}}_1+\hat{\mathcal{L}}_2$	4.85	3.00
$\hat{\mathcal{L}}_0+\hat{\mathcal{L}}_1+\hat{\mathcal{L}}_2+\mathcal{L}_3$	**0.29**	3.75
$\hat{\mathcal{L}}_0+\hat{\mathcal{L}}_1+\hat{\mathcal{L}}_2+\mathcal{L}_4$	4.65	2.50
$\hat{\mathcal{L}}_0+\hat{\mathcal{L}}_1+\hat{\mathcal{L}}_2+\mathcal{L}_3+\mathcal{L}_4$	**0.29**	**2.00**

Table 2. Ablation study of SC-MSFSC network.

ing (LRSC) [34], Sparse Subspace Clustering (SSC) [5], Kernel Sparse Subspace Clustering (KSSC) [30], SSC by Orthogonal Matching Pursuit (SSC-OMP) [41], Efficient Dense Subspace Clustering (EDSC) [12], SSC with the pre-trained convolutional auto-encoder features (AE+SSC), EDSC with the pre-trained convolutional auto-encoder features (AE+EDSC), Deep Subspace Clustering Networks (DSC) [13], Deep Adversarial Subspace Clustering (DASC) [45], Self-Supervised Convolutional Subspace Clustering Network (S^{2}CSC) [42] and Distribution Preserving Subspace Clustering (DPSC) [44]. For comparison methods, we directly cite the best results reported in related papers.

The specific structure of SC-MSFSC on each dataset are presented in Table 1 and Table 6. Consistent with the index of previous coefficient matrices, we label the decoder index in descending order. In the DAE, the kernel stride is set as 2, and the activation function is the Rectified Linear Unit (ReLU) [16]. The learning rate in total network is 1.0×10^{-3} over all experiments and the size of the channel fusion convolutional kernel is 3×3 in the multi-scale fusion module and the weights are initialized by Glorot Uniform [8] for ORL and Extended Yale B datasets and all-ones for COIL dataset. The results of different settings about kernel size and initialization are shown in Table 5. For a fair comparison, we adopt the same pre-trained weights of DAE and pre-defined DAE structures with DSC [13].

4.1. Experiments on Extended Yale B dataset

The Extended Yale B dataset [7] is a popular benchmark for subspace clustering which consists of 38 subjects, with approximately 64 frontal face images per subject acquired under different illumination conditions (seen Figure 3(a)). Following the protocol of experiment in [13, 42], we downsample the original face images from 192×168 to 48×42 and test the robustness with an increasing number of clusters, *i.e.*, $n\in\{10,15,20,25,30,35,38\}$. For a fair comparison, we adopt same network settings as that used in DSC [13], see Table 1. The same tradeoff parameters λ_1,λ_2 with DSC are set as 1 and $3.0\times 10^{n/10-2.0}$ respectively. And the remaining parameters λ_3,λ_4,T are set as 1, 200 and $50+40*n$ respectively.

The clustering performances of different comparison methods on various numbers of subjects are presented in Table 3. For the experiments about n subjects, we report the mean and median clustering errors of $(39-n)$ trials. We observe that our network could significantly reduce the clustering errors and achieve the lowest clustering error in all kinds of n that all listed comparison methods. Note that DASC [45] gains 1.44% clustering error of 38 subjects which is still higher than the result of our network. In particular, for 38 subjects case, our SC-MSFSC obtains a clustering error of 0.29% which improves 1.2% over the best performing baseline S^{2}CSC. Additionally, the best baseline S^{2}CSC adopts the output of spectral clustering to supervise the learning process of other modules, in such situation, it still achieves poorer performance which means our network truly learns more useful information from DAE. It is weird that the results of 38 subjects are the lowest among the ones of all ranges of n and its reason may be that DAE learned the features of all 38 subjects, but for each trial in $n<38$ cases, multi-scale features extracted from DAE would be unstable and then leads to the unstable outcomes.

To further demonstrate the effectiveness of the proposed multi-scale fusion loss ((3)-(6)) and similarity constraint loss (7), we evaluate the impact of adopting multi-scale fusion module and similarity constraint module via an ablation study in Table 2. The baseline is set as the experimental results of DSC [13], whose losses are $\mathcal{L}_0+\mathcal{L}_1+\mathcal{L}_2$. And the second loss $\hat{\mathcal{L}}_0+\hat{\mathcal{L}}_1+\hat{\mathcal{L}}_2$ have the same meanings in total loss function (9), *i.e.,* adding multi-scale fusion loss (except

Methods	LRR	LRSC	SSC	AE+SSC	KSSC	SSC-OMP	EDSC	AE+EDSC	DSC$_{\ell 1}$	DSC$_{\ell 2}$	S^{2}CSC$_{\ell 2}$	S^{2}CSC$_{\ell 1}$	Ours
10 subjects													
Mean	22.22	30.95	10.22	17.06	14.49	12.08	5.64	5.46	2.23	1.59	1.18	1.18	**0.84**
Median	23.49	29.38	11.09	17.75	15.78	8.28	5.47	6.09	2.03	1.25	1.09	1.09	**0.78**
15 subjects													
Mean	23.22	31.47	13.13	18.65	16.22	14.05	7.63	6.70	2.17	1.69	1.14	1.12	**0.88**
Median	23.49	31.64	13.40	17.76	17.34	14.69	6.41	5.52	2.03	1.72	1.14	1.14	**0.78**
20 subjects													
Mean	30.23	28.76	19.75	18.23	16.55	15.16	9.30	7.67	2.17	1.73	1.31	1.30	**0.94**
Median	29.30	28.91	21.17	16.80	17.34	15.23	10.31	6.56	2.11	1.80	1.32	1.25	**0.85**
25 subjects													
Mean	27.92	27.81	26.22	18.72	18.56	18.89	10.67	10.27	2.53	1.75	1.32	1.29	**0.71**
Median	28.13	26.81	26.66	17.88	18.03	18.53	10.84	10.22	2.19	1.81	1.34	1.28	**0.62**
30 subjects													
Mean	37.98	30.64	28.76	19.99	20.49	20.75	11.24	11.56	2.63	2.07	1.71	1.67	**0.96**
Median	36.82	30.31	28.59	20.00	20.94	20.52	11.09	10.36	2.81	2.19	1.77	1.72	**0.67**
35 subjects													
Mean	41.85	31.35	28.55	22.13	26.07	20.29	13.10	13.28	3.09	2.65	1.67	1.62	**1.27**
Median	41.81	31.74	29.04	21.74	25.92	20.18	13.10	13.21	3.10	2.64	1.69	1.60	**1.31**
38 subjects													
Mean	34.87	29.89	27.51	25.33	27.75	23.52	11.64	12.66	3.33	2.67	1.56	1.52	**0.29**
Median	34.87	29.89	27.51	25.33	27.75	23.52	11.64	12.66	3.33	2.67	1.56	1.52	**0.29**

Table 3. Clustering error (%) on Extended Yale B. Best in bold.

$\mathcal{L}_3$) into the original loss $\mathcal{L}_0 + \mathcal{L}_1 + \mathcal{L}_2$. From the results in Table 2, $\mathcal{L}_3$ is the key to successfully reduce the clustering errors and adopt more information from the fused coefficient matrix, when $\hat{\mathcal{L}}_0 + \hat{\mathcal{L}}_1 + \hat{\mathcal{L}}_2$ could not provide better fused coefficient matrix. Additionally, $\mathcal{L}_4$ could improve the performance to a certain extent, but the improvement is limited, which is in line with its positioning, *i.e.,* guiding the fused coefficient matrix.

4.2. Experiments on ORL dataset

The ORL dataset [32] is composed of face images of 40 subjects, where each subjects having 10 face images taken under varying lighting conditions, with different facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses) (see Figure 3(b)). Since the face images were taken under the various facial expressions and details, the ORL dataset becomes more challenging for subspace clustering due to the more non-linearity of subspace and the smaller dataset size compared to Extended Yale B.

Consistent with the experiments in [13, 42], the face images of ORL dataset are down-sampled from 112 × 92 to 32 × 32. The specific network structure of ORL is shown in Table 1. As for the trade-off parameters, we set $\lambda_1 = 1, \lambda_2 = 0.2, \lambda_3 = 0.2, \lambda_4 = 100$, and $T = 800$. The experimental results of our network SC-MSFSC are presented in Table 4. In such a difficult dataset, our proposed network still yields up to 8% ahead of the best baseline S^{2}CSC, which shows the superiority of our network again. Since the DPSC [44] neither not evaluate on ORL dataset nor also provide the official code, therefore, we miss the ORL results of DPSC and denote it as '–'.

Methods	ORL	COIL20	COIL100
LRR	33.50	30.21	53.18
LRSC	32.50	31.25	50.67
SSC	29.50	14.83	44.90
AE+SSC	26.75	22.08	43.93
KSSC	34.25	24.65	47.18
SSC-OMP	37.05	29.86	67.29
EDSC	27.25	14.86	38.13
AE+EDSC	26.25	14.79	38.88
DSC$_{\ell 1}$	14.25	5.65	33.62
DSC$_{\ell 2}$	14.00	5.42	30.96
DASC	11.75	3.61	27.85
S^{2}CSC$_{\ell 2}$	11.25	2.33	27.83
S^{2}CSC$_{\ell 1}$	10.50	2.14	26.67
DPSC	–	2.46	24.60
Ours	**2.00**	**0.62**	**23.90**

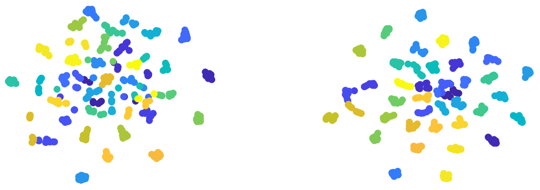
Table 4. Clustering error (%) on ORL, COIL20/100. Best in bold.

As read from the ablation study about ORL in Table 2, if we only use the multi-scale loss, the clustering error still reduces significantly. The introduction of $\mathcal{L}_3$ maybe brings more information, but at the same time brings some risks. Additionally, similarity constraint loss $\mathcal{L}_4$ successfully reduces the risks $\mathcal{L}_3$ brings and further improves the final performance. The reason why $\mathcal{L}_4$ has different behaviors in both ORL and Extended Yale B may be that different size of the dataset leads to the different representation of DAE, and then results in the different outcomes of $\hat{\mathcal{L}}_0 + \hat{\mathcal{L}}_1 + \hat{\mathcal{L}}_2$.

Besides that, we show the clustering errors in different experiment settings about the multi-scale, fusion methods

	Baseline (DSC)	14.00
Multi-Scale	Layer 3	26.75
	Layer 2,3	12.25
	Layer 1,2,3	**2.00**
Fusion Methods	Common Matrix $\mathbf{C}_c$	16.00
	Sum	14.25
	Mean	16.50
	1×1 kernel	25.50
	3×3 kernel	**2.00**
	5×5 kernel	10.25
Kernel Initialization	Glorot Normal	11.50
	Random Normal	42.25
	Glorot Uniform	**2.00**
	Random Uniform	4.50
	Ones	4.50

Table 5. Different experiments settings about multi-scale, fusion methods and kernel initialization on ORL dataset.



(a) DSC

(b) Ours

Figure 4. t-SNE visualization to show the discriminative capability of the self-expression matrix on ORL dataset.

and kernel initialization in Table 5. Note that we adopt the same trade-off parameters in these experiments. For the multi-scale section, more scales could lead to better performance in clustering. For the fusion methods section, the convolutional kernel with 3×3 size yields the best performance. These results indicate that on the one hand, 3×3 kernel size is suitable well for the task of fusing coefficient matrices and proper kernel size would influence the final performance, on the other hand, simply adding or averaging over coefficient matrices could not improve the clustering performance. In addition, we also evaluate the clustering errors of the method to learn a common coefficient matrix $\mathbf{C}_c$ which has discussed in Section 3.3. As Table showed, its clustering results even lower than the one of DSC which demonstrates the correctness of our statement. As for the initialization of convolutional kernel, we could state that glorot uniform [8] achieves the best performance and the normal distribution seems unfit to initialize this kind of fusing strategy. To illustrate the effectiveness of our proposed network, we display the discriminative capability of the obtained self-expression matrix from DSC network and our SC-MSFSC by using t-SNE visualization [23] in Figure 4. The results demonstrate that the self-expression matrix of our network achieves more clear and discriminative latent mapping, which leads to better clustering performance.

	COIL20		COIL100	
Layers	Kernel Size	Channels	Kernel Size	Channels
encoder-1	3×3	15	5×5	50
decoder-1	3×3	15	5×5	50

Table 6. Network structures for COIL20/100.

4.3. Experiments on COIL20/100 dataset

To further demonstrate the performance of our proposed network SC-MSFSC, we also conduct some experiments on object image datasets: COIL20/100 [27, 26]. COIL 20/100 consists of object images (such as duck, see Figure 3(c)) of 20/100 subjects, where each subject having 72 gray-scale images taken under varying poses.

Since the original structure of DSC about COIL experiments only consist of one layer in the encoder, the multi-scale fusion module cannot work well, therefore, we stack the only one coefficient matrix twice along channel dimension. Note that in this experiment, we initialize the weights of the convolutional kernel with 1 and set the convolutional kernel size of the COIL 100 dataset as 9×9. The trade-off parameters used in COIL20 dataset are $\lambda_1 = 1, \lambda_2 = 150, \lambda_3 = 30, \lambda_4 = 100, T = 50$ and that used in COIL 100 dataset are $\lambda_1 = 1, \lambda_2 = 180, \lambda_3 = 360, \lambda_4 = 400, T = 270$. Although the network structure adopted in the COIL dataset is not ideal, our SC-MSFSC still achieves satisfying results, particular in COIL 20 dataset, our network yields the best performance and leads by 1.52% over the best baseline S^2SCN. These results also illustrate that the convolutional kernels are well suited to the coefficient matrix of subspace clustering and our network successfully adopts the latent information embedded in the DAE.

5. Conclusion

We have proposed a novel deep subspace learning framework, Multi-Scale Fusion Subspace Clustering Using Similarity Constraint (SC-MSFSC) to fully adopting the latent information embedded in the DAE. Two novel modules, *i.e.,* multi-scale fusion module and similarity constraint module are devised to learn a more discriminative self-expression coefficient matrix. Benefiting from these two modules, our network on four benchmark datasets outperforms state-of-the-art methods.

Acknowledgment

ZD, CD, XY were supported in part by the Key R&D Program-The Key Industry Innovation Chain of Shaanxi under Grant 2018ZDXM-GY-176 and 2019ZDLGY03-02-01, and in part by the National Key R&D Program of China under Grant 2017YFE0104100 and 2016YFE0200400.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[3] Guangliang Chen and Gilad Lerman. Spectral curvature clustering (scc). *Int. J. Comput. Vis.*, 81(3):317–330, 2009.

[4] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797. IEEE, 2009.

[5] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013.

[6] Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, pages 3818–3825, 2014.

[7] Athinodoros S Georghiades, Peter N Belhumeur, and David J Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):643–660, 2001.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[9] Amit Gruber and Yair Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *CVPR*, volume 1, pages I–I. IEEE, 2004.

[10] Trevor Hastie and Patrice Y Simard. Metrics and models for handwritten character recognition. *Stat. Sci.*, pages 54–65, 1998.

[11] Weitian Huang, Ming Yin, Jianzhong Li, and Shengli Xie. Deep clustering via weighted k-subspace network. *IEEE Signal Process. Lett.*, 26(11):1628–1632, 2019.

[12] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *WACV*, pages 461–468. IEEE, 2014.

[13] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *NeurIPS*, pages 24–33, 2017.

[14] Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 2, pages 586–591. IEEE, 2001.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[18] Chun-Guang Li, Chong You, and René Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Trans. Image Process.*, 26(6):2988–3001, 2017.

[19] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):171–184, 2012.

[20] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, volume 1, page 8, 2010.

[21] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360. Springer, 2012.

[22] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1546–1562, 2007.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[24] Brian McWilliams and Giovanni Montana. Subspace clustering of high-dimensional data: a predictive approach. *Data Min. Knowl. Discov.*, 28(3):736–772, 2014.

[25] Quanyi Mo and Bruce A Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *ECCV*, pages 402–415. Springer, 2012.

[26] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). *Technical report CUCS-006-96*, 1996.

[27] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). *Technical report CUCS-005-96*, 1996.

[28] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, pages 849–856, 2002.

[29] Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse subspace clustering. In *ICCV*, pages 225–232, 2013.

[30] Vishal M Patel and René Vidal. Kernel sparse subspace clustering. In *ICIP*, pages 2849–2853. IEEE, 2014.

[31] Pulak Purkait, Tat-Jun Chin, Alireza Sadri, and David Suter. Clustering with hypergraphs: the case for large hyperedges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1697–1711, 2016.

[32] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *WACV*, pages 138–142. IEEE, 1994.

[33] René Vidal. Subspace clustering. *IEEE Signal Process. Mag.*, 28(2):52–68, 2011.

[34] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *Pattern Recogn. Lett.*, 43:47–61, 2014.

[35] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, pages 94–106. Springer, 2006.

[36] Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry. Unsupervised segmentation of natural images via lossy data compression. *Comput. Vis. Image Understand.*, 110(2):212–225, 2008.

[37] Xu Yang, Cheng Deng, Xianglong Liu, and Feiping Nie. New l 2, 1-norm relaxation of multi-way graph cut for clustering. In *AAAI*, 2018.

[38] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *CVPR*, June 2019.

[39] Ming Yin, Yi Guo, Junbin Gao, Zhaoshui He, and Shengli Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *CVPR*, pages 5157–5164, 2016.

[40] Ming Yin, Shengli Xie, Zongze Wu, Yun Zhang, and Junbin Gao. Subspace clustering via learning an adaptive low-rank graph. *IEEE Trans. Image Process.*, 27(8):3716–3728, 2018.

[41] Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *CVPR*, pages 3918–3927, 2016.

[42] Junjian Zhang, Chun-Guang Li, Chong You, Xianbiao Qi, Honggang Zhang, Jun Guo, and Zhouchen Lin. Self-supervised convolutional subspace clustering network. In *CVPR*, pages 5473–5482, 2019.

[43] Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. *arXiv preprint arXiv:1904.10596*, 2019.

[44] Lei Zhou, Bai Xiao, Xianglong Liu, Jun Zhou, Edwin R Hancock, et al. Latent distribution preserving deep subspace clustering. In *IJCAI*, 2019.

[45] Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *CVPR*, pages 1596–1604, 2018.