Deep Latent Low-Rank Fusion Network for Progressive Subspace Discovery

Zhao Zhang[1,2], **Jiahuan Ren**[2], **Zheng Zhang**[3] and **Guangcan Liu**[4]
[1] Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, China
[2] School of Computer Science and Technology, Soochow University, China
[3] Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, China
[4] School of Information and Control, Nanjing University of Information Science and Technology, China
cszzhang@gmail.com, hmzry10086@outlook.com, darren.zhang@uq.edu.au, gcliu@nuist.edu.cn

Abstract

Low-rank representation is powerful for recovering and clustering the subspace structures, but it cannot obtain deep hierarchical information due to the single-layer mode. In this paper, we present a new and effective strategy to extend the single-layer latent low-rank models into multiple-layers, and propose a new and progressive Deep Latent Low-Rank Fusion Network (DLRF-Net) to uncover deep features and structures embedded in input data. The basic idea of DLRF-Net is to refine features progressively from the previous layers by fusing the subspaces in each layer, which can potentially obtain accurate features and subspaces for representation. To learn deep information, DLRF-Net inputs shallow features of the last layers into subsequent layers. Then, it recovers the deeper features and hierarchical information by congregating the projective subspaces and clustering subspaces respectively in each layer. Thus, one can learn hierarchical subspaces, remove noise and discover the underlying clean subspaces. Note that most existing latent low-rank coding models can be extended to multilayers using DLRF-Net. Extensive results show that our network can deliver enhanced performance over other related frameworks.

1 Introduction

Representation learning is always a fundamental problem to obtain the underlying explanatory factors and features for the subsequent data classification or clustering tasks [Chen *et al.*, 2019 and 2020] [Zhang *et al.*, 2016 and 2019]. Representation learning is still challenging in reality due to the complexity and diversity of data [Lu *et al.*, 2019] [Su *et al.*, 2019] [Acharya *et al.*, 2019] [Ding *et al.*, 2018].

Since most real data can be characterized using low-rank subspaces, low-rank coding methods can recover the underlying subspaces and obtain notable features [Liu *et al.*, 2019] [Ren *et al.*, 2019]. *Low-Rank Representation* (LRR) [Liu *et al.*, 2013] is one of the most classical algorithms to discover multi-subspaces, but it is essentially a transductive method failing to handle new data efficiently. To address the out-of-sample issue, *Inductive Robust Principal Component Analysis* (IRPCA) [Bao *et al.*, 2012] was recently proposed seeking a low-rank projection to map samples into underlying subspaces. To enable a solution for subspace segmentation and feature extraction, *Latent LRR* (LatLRR) [Liu *et al.*, 2011], was proposed, which decomposed data into a principal feature part, a salient feature part and a sparse error. Although LatLRR resolves the insufficient sampling issue and obtains enhanced performance over LRR, it still suffers from a high computational cost due to using Nuclear-norm to approximate the rank function to constrain the subspaces, while the computation of the Nuclear-norm needs the time-consuming Singular Value Decomposition of matrices at each iteration, especially for large-scale datasets. To improve the efficiency, a *Frobenius-norm based LatLRR* (FLLRR) [Yu *et al.*, 2018] was proposed, which approximates the rank function using Frobenius-norm. But the Frobenius-norm is sensitive to noise and outliers, which may produce inaccurate representations.

It is noteworthy that the above algorithms have a common drawback, i.e., they are "shallow" models using single-layer structures. As a result, they cannot obtain deep information and subspaces. But due to the strong representation ability of the deep neural networks [Kim *et al.*, 2019], deep low-rank coding models equipped with carefully designed hierarchical structures should be able to obtain the enhanced performance. In fact, researchers have also designed some deep low-rank coding models, such as *Weakly-supervised Deep Nonnegative Low-rank Model* (WDNL) [Li *et al.*, 2017a], which finds the intrinsic relations between images and tags by removing noise or irrelevant tags, but it is unclear how to handle images directly and the results are usually incomplete. Another deep model is *Deep Low-Rank Subspace Ensemble* (DLRSE) [Xue *et al.*, 2019], where the Frobenius-norm is used as a low-rank constraint. DLRSE uses the deep matrix factorization to learn the diverse hierarchical structures and obtains the low-rank representations from extracted factors. But DLRSE is originally proposed for multi-view clustering, which clearly differs from our task. To cluster big data effectively, a *Projective Low-rank Sub-space Clustering via Learning Deep Encoder* (PLrSC) [Li *et al.*, 2017] has been recently proposed.

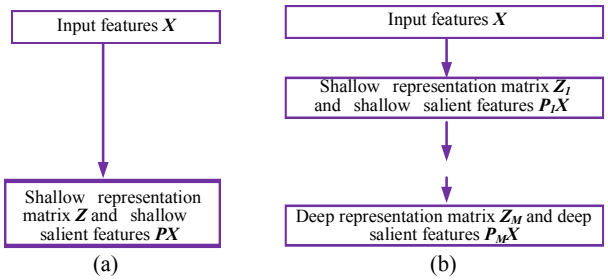


Figure 1: Comparison between the mechanisms of traditional latent low-rank coding methods (a) and our DLRF-Net (b).

PLrSC uses a small database randomly sampled from a big dataset and train a deep encoder, and then a deep encoder is used to compute the low-rank representations of all samples. However, it will be not easy to determine the sampling frequency based on different real datasets in practice.

In this paper, we mainly propose a general and progressive deep low-rank fusion network that can unfold existing latent low-rank methods into multilayers for the hierarchical representation and deep subspace discovery. The major contributions of this paper are summarized as follows:

(1) Technically, a simple yet effective progressive deep representation learning model termed *Deep Latent Low-Rank Fusion Network* (DLRF-Net) is derived. DLRF-Net can learn deep hierarchical information and subspaces from input data. The advantage of this practice is that multiple layer low-rank coding structures can deliver rich and useful hidden hierarchical information that has a great potential in learning more powerful deep representation and subspace structures. To be specific, DLRF-Net in each layer aims to refine the features and subspaces progressively from previous layers by fusion, i.e., it recovers deep hierarchical information by respectively congregating the projective and clustering subspaces in each layer to produce accurate results. We compare the traditional single-layer low-rank models with DLRF-Net in Figure 1.

(2) The network of DLRF-Net is simple, general and easy to extend. Specifically, many existing latent low-rank coding models such as LatLRR and FLLRR can be easily extended from single-layer to multiple-layers using DLRF-Net. In this paper, we mainly explain our basic idea rather than deriving a complex formulation, two simple deep network models are constructed based on embedding LatLRR and FLLRR into DLRF-Net as examples for the multi-layer low-rank coding, which we call *Nuclear-norm based DLRF-Net* (nDLRF-Net) and *Frobenius-norm based fast DLRF-Net* (fDLRF-Net).

(3) Extensive simulations on public databases demonstrate that both nDLRF-Net and fDLRF-Net can deliver enhanced performance than the related single-layer models. That is, the multilayer idea of DLRF-Net is feasible and effective.

2 Related Work

We describe the closely-related low-rank coding algorithms.

2.1 LatLRR and FLLRR

Given a data matrix $X=[x_1,x_2\cdots,x_N]\in\mathbb{R}^{n\times N}$, where $x_i\in\mathbb{R}^n$ is a sample represented using an n-dimensional vector and N is the number of samples, then LatLRR improves LRR using the unobserved hidden data X_H to extend the dictionary and overcome the insufficient data sampling issue. Specifically, LatLRR considers the following coding formulation:

$$\min_{Z} rank(Z), s.t.\ X_O=[X_O,X_H]Z\ , \tag{1}$$

where $rank(\bullet)$ is rank function and X_O is the observed data matrix. Supposing that X_O and X_H are sampled from the same collection of low-rank subspaces, by using the Nuclear-norm to approximate the rank function and using sparse L_1-norm on error term E, LatLRR recovers the hidden effects by

$$\min_{Z,P,E}\|Z\|_*+\|L\|_*+\lambda\|E\|_1,\ s.t.\ X=XZ+LX+E\ , \tag{2}$$

where $\|Z\|_*$ is the Nuclear-norm of Z [Liu *et al.*, 2013] [Xie *et al.*, 2017], i.e., the sum of its singular values, XZ and LX are principal features and salient features respectively, and λ is a positive scalar. Since LatLRR uses the Nuclear-norm constraints on Z and L, the SVD process is involved, which is time-consuming. Note that Frobenius-norm $\|\bullet\|_F$ can also be used as the convex surrogate of the rank function [Yu *et al.*, 2018]. Besides, the optimization of Frobenius-norm is very efficient. The objective function of FLLRR is defined as

$$\min_{Z,P,E}\frac{1}{2}\left(\|Z\|_F^2+\|L\|_F^2\right)+\lambda\|E\|_1,\ s.t.\ X=XZ+LX+E\ , \tag{3}$$

from which multi-subspace structures can be recovered by Z and the notable features can be extracted using L.

2.2 Projective Low-rank Subspace Clustering via Learning Deep Encoder (PLrSC)

Assuming that $Y=\left[Y^1,\cdots Y^i,\cdots Y^k\right]\in\mathbb{R}^{n\times N}$ is a big dataset and over-sufficiently drawn from a union of k subspaces, where N is the number of samples in all subspaces. PLrSC assumes that $X=\left[X^1,\cdots X^i,\cdots X^k\right]\in\mathbb{R}^{n\times N}$ is a small dataset sampled randomly from Y and X is still sufficient. First, PLrSC learns a non-iterative deep encoder $f_{de}(X;\theta)$, where θ is the learning parameter to approximate low-rank representations. Then, the deep encoder is utilized to obtain the low-rank codes for replacing the costly non-linear Singular Value Thresholding (SVT) [*Cai et al.*, 2010] operations. Thus, the predictive low-rank decomposition of PLrSC can be written as follows:

$$\begin{aligned}&\min_{Z}\|Z\|_*+\lambda\|E\|_{2,1}+\gamma\left\|Z-f_{de}(X;\theta)\right\|_F^2\ ,\\ &s.t.\ X=XZ+E\end{aligned} \tag{4}$$

where $E\in\mathbb{R}^{n\times N}$ denotes a sparse error, λ is the regularization parameter for E, γ is a control parameter for approximation term $\left\|Z-f_{de}(X;\theta)\right\|_F^2$. $f_{de}(X;\theta)=g\left(W^M\cdots g\left(W^i\cdots g\left(W^2X\right)\right)\right)$ is a deep encoder with M layers, where $g(\bullet)$ is an activation function (e.g., sigmoid or *ReLU*). θ is a learning parameter set, where $\theta=\left\{W^2,\cdots,W^M\right\}\in\mathbb{R}=\left\{\mathbb{R}^{l_2\times l_1},\cdots,\mathbb{R}^{l_M\times l_{M-1}}\right\}$. l_i denotes the number of the units in the *i-th* layer ($l_1=d$ and $l_M=N$). Then, PLrSC employs the alternating direction algorithm (ADM) [Liu *et al.*, 2013] and a gradient descent algorithm (GD) [Li et al., 2015] to optimize the above problem. Finally, PLrSC applies the landmark-based spectral clustering (LSC) algorithm to cluster the big dataset Y [Li *et al.*, 2017].

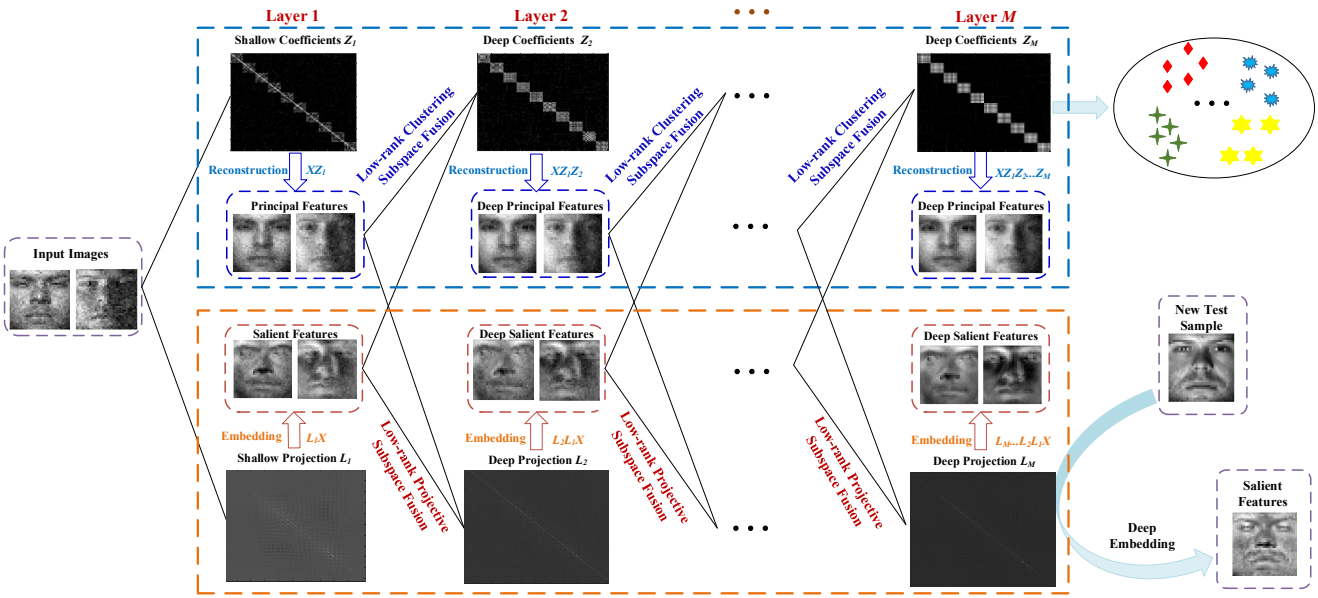


Figure 2: The framework and flow-chart of our proposed DLRF-Net algorithm.

3 Proposed DLRF-Net

3.1 Objective Function

DLRF-Net designs a hierarchical and progressive approach, i.e., the representation in the uncovered subspaces are learnt layer by layer. That is, the deep principal features $XZ_0Z_1\ldots Z_{l-1}$ and deep salient features $L_{l-1}\ldots L_1L_0X$ from the (l-1)-th layer are fed into the l-th layer, which are further decomposed into a deep principal feature part, a deep salient feature part and a deep sparse error. The whole framework of our DLRF-Net is shown in Figure 2. Assuming that DLRF-Net has M layers, the decomposition process of our DLRF-Net framework in the l-th layer can be presented as follows:

$$\begin{aligned} XZ_0Z_1\cdots Z_{l-1} &= XZ_0Z_1\cdots Z_{l-1}Z_l^1 + L_l^1XZ_0Z_1\cdots Z_{l-1} + E_l^1 \\ L_{l-1}\cdots L_1L_0X &= L_{l-1}\cdots L_1L_0XZ_l^2 + L_l^2L_{l-1}\cdots L_1L_0X + E_l^2 \end{aligned}, \quad (5)$$

where Z_l^1 and Z_l^2 are the deep coefficient matrices, L_l^1 and L_l^2 denote the deep projection matrices that are learned from $XZ_0Z_1\ldots Z_{l-1}$ and $L_{l-1}\ldots L_1L_0X$ in the l-th layer, respectively. Note that Z_0 and L_0 are included to simplify the descriptions, which are set to the identity matrices, i.e., the input of the first layer is the original data. It should be noted that for the optimization in the l-th layer, Z_0, Z_1, …, Z_{l-1}, L_0, L_1, …, L_{l-1} are known variables that are updated in the last layer. As such, intuitively from the multilayer leaning process, deep principal features $XZ_0Z_1\ldots Z_{l-1}$ and deep salient features $L_{l-1}\ldots L_1L_0X$ are learnt progressively from different layers, i.e., extracting fine-grained features from layer to layer.

Finally, deep principal features $XZ_0Z_1\ldots Z_l$ and salient features $L_l\ldots L_1L_0X$ in the l-th layer can be obtained as

$$\begin{aligned} XZ_0Z_1\cdots Z_{l-1}Z_l &= XZ_0Z_1\cdots Z_{l-1}\left(Z_l^1 + Z_l^2\right)/2 \\ L_lL_{l-1}\cdots L_1L_0X &= \left(L_l^1 + L_l^2\right)L_{l-1}\cdots L_1L_0X/2 \end{aligned}. \quad (6)$$

The above subspace fusion operation can potentially make the learned representations more accurate by fusing feature information from deep principal and salient features in previous layers. The above averaging operation can also prevent the feature information loss and balance the information from deep principal and salient features in each layer. As such, we have the following model for DLRF-Net in the l-th layer:

$$\begin{aligned} &\min_{Z_l,L_l,E_l} \frac{1}{2}\left(\left\|Z_l^1\right\|_p + \left\|Z_l^2\right\|_p + \left\|L_l^1\right\|_p + \left\|L_l^2\right\|_p\right) + \lambda_l\left(\left\|E_l^1\right\|_1 + \left\|E_l^2\right\|_1\right) \\ &s.t.\, XZ_0\ldots Z_{l-1} = \left(XZ_0\ldots Z_{l-1}\right)Z_l^1 + L_l^1\left(XZ_0\ldots Z_{l-1}\right) + E_l^1 \\ &\quad L_{l-1}\ldots L_0X = \left(L_{l-1}\ldots L_0X\right)Z_l^2 + L_l^2\left(L_{l-1}\ldots L_0X\right) + E_l^2 \end{aligned}, \quad (7)$$

where $\|\bullet\|_p$ is the matrix p-norm, which can be Nuclear-norm or squared Frobenius-norm. λ_l is a positive tunable parameter that replies on the noise level of data [Liu *et al.*, 2011]. We name the Nuclear-norm based DLRF-Net as *nDLRF-Net* and name the squared Frobenius-norm based DLRF-Net as *fDLRF-Net*. The objective function of fDLRF-Net can then be defined as follows for deep subspace discovery:

$$\begin{aligned} &\min_{Z_l,L_l,E_l} \sum_{l=1}^{M}\left\{\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2 + \left\|Z_l^2\right\|_F^2 + \left\|L_l^1\right\|_F^2 + \left\|L_l^2\right\|_F^2\right) + \lambda_l\left(\left\|E_l^1\right\|_1 + \left\|E_l^2\right\|_1\right)\right\} \\ &s.t.\, XZ_0\ldots Z_{l-1} = \left(XZ_0\ldots Z_{l-1}\right)Z_l^1 + L_l^1\left(XZ_0\ldots Z_{l-1}\right) + E_l^1 \\ &\quad L_{l-1}\ldots L_0X = \left(L_{l-1}\ldots L_0X\right)Z_l^2 + L_l^2\left(L_{l-1}\ldots L_0X\right) + E_l^2 \end{aligned}. \quad (8)$$

Next, we show the optimization procedures of DLRF-Net.

4 Optimization

We mainly describe the optimization of fDLRF-Net in detail, as the optimization of nDLRF-Net is similar. Since Z_l, P_l and E_l depend on each other, we update them alternately. We use the inexact Augmented Lagrange Multiplier (Inexact ALM) algorithm [Lin *et al.*, 2009] for efficiency.

Algorithm 1 Solving Eq.(9) by Inexact ALM (*l*-th layer)

Inputs: Reconstructed data A_{l-1}, tunable parameters λ_l, α_l.

Initialization: $t=0, \left(Z_l^1\right)^0=0, \left(Z_l^2\right)^0=0, \left(L_l^1\right)^0=0, \left(L_l^2\right)^0=0, \left(E_l^1\right)^0=0$, $\left(E_l^2\right)^0=0, \left(Y_l^1\right)^0=0, \left(Y_l^2\right)^0=0, \mu_{\max}=10^6$, $\mu^0=10^{-6}, \eta=1.12$, $\varepsilon=10^{-7}$.

While *not converged* do

1. Update the coefficients sub-matrices $\left(Z_l^1\right)^{t+1}$ and $\left(Z_l^2\right)^{t+1}$ by using Eq.(12-1e), and obtain $Z_l^{t+1}=\left(\left(Z_l^1\right)^{t+1}+\left(Z_l^2\right)^{t+1}\right)/2$;
2. Update the projection sub-matrices $\left(L_l^1\right)^{t+1}$ and $\left(L_l^2\right)^{t+1}$ by using Eq.(14-15), and obtain $L_l^{t+1}=\left(\left(L_l^1\right)^{t+1}+\left(L_l^2\right)^{t+1}\right)/2$;
3. Update the sparse errors $\left(E_l^1\right)^{t+1}$ and $\left(E_l^2\right)^{t+1}$ by Eq.(16-17);
4. Update the Lagrange multipliers $\left(Y_l^1\right)^{t+1}$ and $\left(Y_l^2\right)^{t+1}$;
5. Update the parameter μ_l by $\mu_l^{t+1}=\min\left(\eta\mu_l^t, \mu_{\max}\right)$;
6. Check for convergence: Suppose $\left\{\left\|P_{l-1}-P_{l-1}Z_l^1-L_l^1\ P_{l-1}-E_l^1\right\|_\infty, \left\|S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\|_\infty\right\}\le\varepsilon$, stop; else $t=t+1$.

End while

Output: $Z_l^*=Z_l^{t+1}, P_l^*=P_l^{t+1}$.

To simplify the descriptions of optimization, we train the model layer by layer. To learn features in the *l*-th layer (*l*=1, 2,…, *M*), the target function can be defined as

$$\begin{aligned}&\min_{Z_l,L_l,E_l}\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2+\left\|Z_l^2\right\|_F^2+\left\|L_l^1\right\|_F^2+\left\|L_l^2\right\|_F^2\right)+\lambda_l\left(\left\|E_l^1\right\|_1+\left\|E_l^2\right\|_1\right)\\&s.t.\ XZ_0\ldots Z_{l-1}=\left(XZ_0\ldots Z_{l-1}\right)Z_l^1+L_l^1\left(XZ_0\ldots Z_{l-1}\right)+E_l^1\\&\quad L_{l-1}\ldots L_0X=\left(L_{l-1}\ldots L_0X\right)Z_l^2+L_l^2\left(L_{l-1}\ldots L_0X\right)+E_l^2\end{aligned}. \quad (9)$$

Denote by $P_{l-1}=XZ_0\ldots Z_{l-1}$ and $S_{l-1}=L_{l-1}\ldots L_0X$ two auxiliary matrices, the Lagrange function of Eq.(9) can be obtained as

$$\begin{aligned}&\wp\left(Z_l,L_l,E_l\right)=\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2+\left\|Z_l^2\right\|_F^2+\left\|L_l^1\right\|_F^2+\left\|L_l^2\right\|_F^2\right)+\lambda\left(\left\|E_l^1\right\|_1+\left\|E_l^2\right\|_1\right)\\&+\left\langle Y_l^1,P_{l-1}-P_{l-1}Z_l^1-L_l^1\ P_{l-1}-E_l^1\right\rangle+\left\langle Y_l^2,S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\rangle,\\&+\frac{\mu_l}{2}\left(\left\|P_{l-1}-P_{l-1}Z_l^1-L_l^1\ P_{l-1}-E_l^1\right\|_F^2+\left\|S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\|_F^2\right)\end{aligned} \quad (10)$$

where Y_l^1 and Y_l^2 are Lagrange multipliers, and μ_l denotes a positive parameter. Then, fDLRF-Net updates the variables by solving $\wp$. Note that the optimization procedures of our fDLRF-Net in the *l*-th layer can then be detailed as follows:

Fix others, update Z_l: For the optimization of Z_l, we need to solve Z_l^1 and Z_l^2. By removing the irrelevant terms, we can update Z_l^1 and Z_l^2 by the following reduced problem:

$$\begin{aligned}&\wp\left(Z_l^1,Z_l^2\right)=\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2+\left\|Z_l^2\right\|_F^2\right)+\left\langle Y_l^1,\Xi_{l-1}-P_{l-1}Z_l^1\right\rangle\\&+\left\langle Y_l^2,\Omega_{l-1}-S_{l-1}Z_l^2\right\rangle+\frac{\mu_l}{2}\left(\left\|\Xi_{l-1}-P_{l-1}Z_l^1\right\|_F^2+\left\|\Omega_{l-1}-S_{l-1}Z_l^2\right\|_F^2\right)\end{aligned}, \quad (11)$$

where $\Xi_{l-1}=P_{l-1}-L_l^1\ P_{l-1}-E_l^1$ and $\Omega_{l-1}=S_{l-1}-L_l^2S_{l-1}-E_l^2$. We first show the optimization of Z_l^1. By taking the derivative of $\wp\left(Z_l^1,Z_l^2\right)$ w.r.t. Z_l^1 and zeroing the derivative, we can infer the coefficients matrix Z_l^1 at the *(t+1)*-th iteration as follows:

$$\left(Z_l^1\right)^{t+1}=\left(I+\mu_l^tP_{l-1}^TP_{l-1}\right)^{-1}\mu_l^tP_{l-1}^T\left(\Xi_{l-1}^t+\left(Y_l^1\right)^t/\mu_l^t\right), \quad (12)$$

where $\Xi_{l-1}^t=P_{l-1}-\left(L_l^1\right)^tP_{l-1}-\left(E_l^1\right)^t$. Similar to the optimization of Z_l^1, we can infer $\left(Z_l^2\right)^{t+1}$ in the *(t+1)*-th iteration as

$$\left(Z_l^2\right)^{t+1}=\left(I+\mu_l^tS_{l-1}^TS_{l-1}\right)^{-1}\mu_l^tS_{l-1}^T\left(\Omega_{l-1}^t+\left(Y_l^2\right)^t/\mu_l^t\right), \quad (13)$$

where $\Omega_{l-1}^t=S_{l-1}-\left(L_l^2\right)^tS_{l-1}-\left(E_l^2\right)^t$. After optimizing the $\left(Z_l^1\right)^{t+1}$ and $\left(Z_l^2\right)^{t+1}$, we can obtain $Z_l^{t+1}=\left(\left(Z_l^1\right)^{t+1}+\left(Z_l^2\right)^{t+1}\right)/2$.

Fix others, update L_l: By removing the irrelevant terms from $\wp$, taking the derivatives of $\wp\left(L_l^1,L_l^2\right)$ w.r.t L_l^1 and L_l^2, and zeroing the derivatives, we can similarly obtain

$$\left(L_l^1\right)^{t+1}=\mu_l^t\left(\Xi_{l-1}^{t+1}+\left(Y_l^1\right)^t/\mu_l^t\right)\left(I+\mu P_{l-1}^TP_{l-1}\right)^{-1}, \quad (14)$$

$$\left(L_l^2\right)^{t+1}=\mu_t\left(\Omega_{l-1}^{t+1}+\left(Y_l^2\right)^t/\mu_t\right)\left(I+\mu S_{l-1}{}^TS_{l-1}\right)^{-1}, \quad (15)$$

where $\Xi_{l-1}^{t+1}=P_{l-1}-P_{l-1}\left(Z_l^1\right)^{t+1}-\left(E_l^1\right)^t$ and $\Omega_{l-1}^{t+1}=S_{l-1}-S_{l-1}Z_{l(t+1)}^2-E_{l(t)}^2$. After optimizing the projection matrices $\left(L_l^1\right)^{t+1}$ and $\left(L_l^2\right)^{t+1}$, we can obtain $L_l^{t+1}=\left(\left(L_l^1\right)^{t+1}+\left(L_l^2\right)^{t+1}\right)/2$.

Fix others, update E_l^1 and E_l^2: By taking the derivative of Lagrange function w.r.t. E_l^1 and E_l^2 respectively and zeroing the derivatives, we can infer E_l^1 and E_l^2 as

$$\left(E_l^1\right)^{t+1}=\arg\min_{E_l^1}\frac{\lambda}{\mu_l^t}\left\|E_l^1\right\|_1+\frac{1}{2}\left\|E_l^1-\left(P_{l-1}-\Delta_l^1\right)\right\|_F^2, \quad (16)$$

$$\left(E_l^2\right)^{t+1}=\arg\min_{E_l^2}\frac{\lambda}{\mu_l^t}\left\|E_l^2\right\|_1+\frac{1}{2}\left\|E_l^2-\left(S_{l-1}-\Delta_l^2\right)\right\|_F^2, \quad (17)$$

which can be easily solved by the shrinkage operator [Lin *et al.*, 2009], where Δ_l^1 and Δ_l^2 are auxiliary matrices defined as $\Delta_l^1=P_{l-1}\left(Z_l^1\right)^{t+1}-\left(L_l^1\right)^{t+1}P_{l-1}+\left(Y_l^1\right)^t/\mu_l^t$ and $\Delta_l^2=S_{l-1}\left(Z_l^2\right)^{t+1}-\left(L_l^2\right)^{t+1}S_{l-1}+\left(Y_l^2\right)^t/\mu_l^t$. For complete presentation of our model, we summarized the optimization procedures of solving the sub-problem of Eq.(9) in the *l-th* layer in Algorithm 1.

5 Discussion

5.1 Relationship Analysis

We mainly discuss the relations of our DLRF-Net to LatLRR and FLLRR. To facilitate the analysis, we consider the special case that *l*=1.We first express this special case as

$$\begin{aligned}&\min_{Z_1,L_1,E_1}\frac{1}{2}\left(\left\|Z_1\right\|_p+\left\|L_1\right\|_p\right)+\lambda_1\left\|E_1\right\|_1\\&s.t.\ XZ_0=XZ_0Z_1+L_1XZ_0+E_1,\ L_0X=L_0XZ_1+L_1L_0X+E_1\end{aligned}. \quad (18)$$

Since Z_0 and L_0 are initialized to the identity matrices in the optimization, the two constraints are the same. As such, it is clear that when we use the Frobenius-norm to constrain the matrices Z_1 and L_1, the problem identifies FLLRR; while we use the Nuclear-norm as constraints, the resulting problem is identical to LatLRR. That is, both FLLRR and LatLRR are the special causes of our DLRF-Net framework.

5.2 Computational Time Complexity

We analyze the complexity of each layer of our deep models. For fDLRF-Net, SVD is not used and the major computation is updating the matrices Z_l and L_l. Thus, the time complexity of Algorithm 1 is equal to that of FLLRR. Thus, it is easy to

Dataset Name	# Samples	# Dim	# Classes
CMU PIE	11554	1024	68
UMIST face	1012	1024	20
Fashion MNIST	70000	784	10

Table 1: Descriptions of used image datasets.

infer that the total time complexity of fDLRF-Net is M times that of each layer, where M is the number of layers, which is usually a small value. For nDLRF-Net, the time complexity of each layer is the same as that of regular LatLRR and the total time complexity is M times that of each layer.

6 Experimental Results and Analysis

We conduct experiments to evaluate the effectiveness of our fDLRF-Net and nDLRF-Net, and show the comparison results with other related methods, including FLLRR, LatLRR, LRR, Robust LatLRR (rLatLRR) [Zhang *et al.*, 2014b], Laplacian Regularized LRR (rLRR) [Zhang *et al.*, 2014a], Sa-LatLRR [Wang *et al.*, 2018] and PLrSC [Li *et al.*, 2017b]. Three real image databases are involved, including two face datasets (i.e., CMU PIE [Sim *et al.*, 2003], UMIST [Graham *et al.*, 1998]) and the Fashion MNIST database [Xiao *et al.*, 2017]. The details of used databases are described in Table 1. We follow the common procedure to resize each face image into 32×32 pixels and images of the Fashion MNIST dataset are resized into 28×28 pixels. We perform all experiments on a PC with Intel (R) Core (TM) i7-7700 CPU @ 3.6 GHz 8G.

6.1 Visual Image Analysis by Visualization

Visualization of coefficient matrix *Z*. To represent data appropriately, Z should have a block-diagonal structure. Each block denotes the coefficients for certain subject so that each sample can be reconstructed by the samples of one class as much as possible. We follow [Liu *et al.*, 2011] to construct 10 independent subspaces $\{S_i\}_{i=1}^{10}$ and apply this artificial data for DLRF-Net. The visualization of Z in the first four layers are illustrated in Figure 3. We see that all coefficient matrices Z have block-diagonal structures. But compared with the *1-th* layer, the results of other layers have less noise and wrong inter-class connections. It can also be found that the subspace structures of Z are improved progressively, i.e., the learned structures from the *3-rd* and *4-th* layers are better that from the *2-nd* layer. But the difference of the structures of Zs in the *3-rd* and *4-th* layers is small, i.e., our DLRF-Net can remove noise contained in features and recovery the subspaces of Z by using small number of layers. That is, the structures of the coefficient matrix Z will not become better, even though we use more layers involving high computational cost. In all the simulations, the results of fDLRF-Net and nDLRF-Net in the *1-th* layer corresponds to FLLRR and LatLRR, respectively.

Visualization of the recovered features *XZ*. We evaluate fDLRF-Net and nDLRF-Net by visualizing recovered deep features XZ. Given a data matrix X, DLRF-Net decomposes it into principal features XZ, salient features LX and a sparse error E in each layer. CMU PIE face dataset is used. This face image database contains 68 persons with 41368 images under varying poses, illuminations and facial expressions. 170 near frontal images per person are employed for CMU PIE, which contains five near frontal poses (C05, C07, C09, C27, and C29). To evaluate the robustness properties, random Gaussian noise with variance 500 is included into the image data. Some original images, noisy images and recovered principal features in the first three layers are shown in Figure 4. We see find that fDLRF-Net and nDLRF-Net can effectively remove the shadow and noise in images in a progressive way, compared with the recovered results of FLLRR and LatLRR.

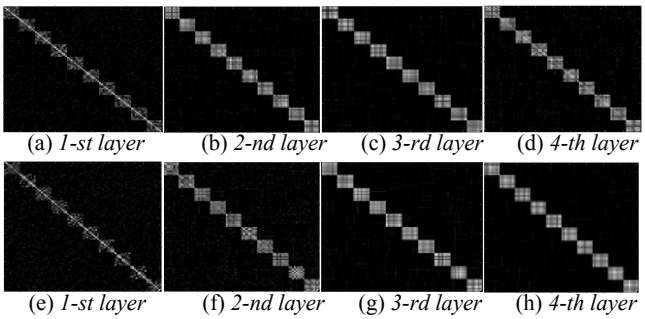


Figure 3: Visualization of the representation coefficient matrices Z of our fDLRF-Net (a-d) and nDLRF-Net(e-h).

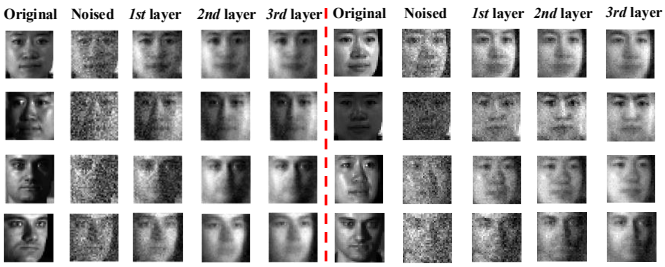


Figure 4: Recovered principal features on CMU PIE face database (**Left**: fDLRF-Net; **Right**: nDLRF-Net).

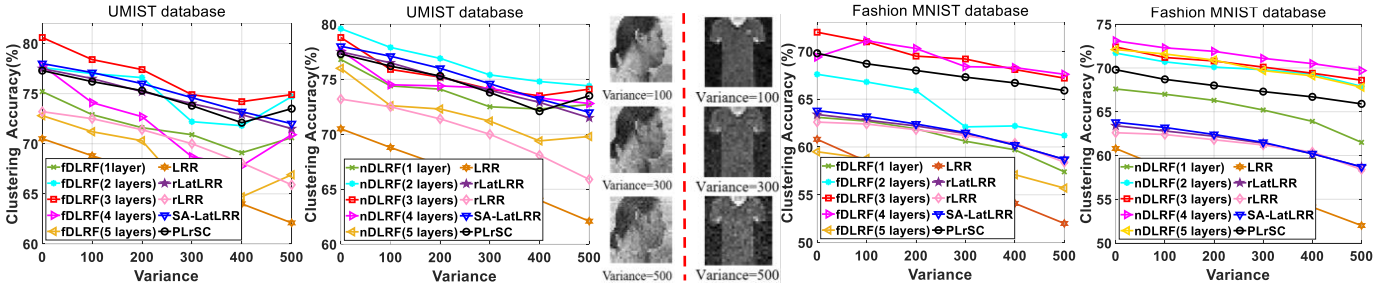
6.2 Quantitative Clustering Evaluations

We compare each model for clustering images. UMIST and Fashion MNIST are evaluated. UMIST has 1012 images from 20 different individuals. Fashion-MNIST has 10 classes and 70000 unique products. In this study on Fashion-MNIST, we choose 1000 samples per class, i.e., totally 10000 samples. To evaluate the performance, we follow the common procedures and use the coefficient matrix Z^* of each method to construct the weights by $W = \left(|Z^*| + |Z^{*T}|\right)/2$ and then use the Normalized Cuts (NCut) [Shi *et al.*, 2000] for clustering. For PLrSC and DLRF-Net, we use the coefficients from the final layer. For each number K of clusters, we choose K categories randomly and the results are averaged over 30 initializations.

The clustering accuracy (AC) [Cai *et al.*, 2017] is used as the quantitative metric. The values of AC on evaluated databases are shown in Table 2. We see that: (1) the clustering accuracy of each method goes down as the number of categories increases, since clustering more data is difficult than clustering less; (2) Our fDLRF-Net and nDLRF-Net deliver higher ACs than the other competitors, especially for FLLRR and LatLRR, implying that DLRF-Net can learn more effective representations by mining deep information.

Method	Clustering Accuracy (%) on UMIST				Clustering Accuracy (%) on Fashion MNIST			
	K=2	K=4	K=6	K=8	K=2	K=4	K=6	K=8
LRR	78.57±15.67	72.08±14.57	66.67±4.97	62.57±9.56	85.21±4.37	62.27±9.11	51.83±1.99	45.75±3.24
rLatLRR	86.25±15.06	79.44±9.94	69.67±7.40	67.51±6.56	89.55±13.83	65.50±9.01	63.33±8.47	56.75±4.83
SA-LatLRR	84.57±14.30	78.35±12.26	72.08±7.40	67.52±9.29	91.07±11.74	68.74±4.28	63.54±11.91	56.12±6.41
rLRR	82.35±14.41	78.52±6.48	70.67±9.49	66.57±7.52	89.48±14.49	67.48±8.74	62.17±7.05	56.87±5.47
PLrSC	87.89±8.74	79.89±6.77	71.11±10.27	67.79±7.38	91.42±5.39	72.06±6.47	64.43±8.27	57.61±6.75
FLLRR	84.16±17.37	79.25±8.98	71.56±8.18	66.29±7.70	90.52±10.60	71.58±10.31	62.67±9.51	56.33±5.78
fDLRF-Net (*2 layers*)	88.17±16.58	80.42±10.61	73.83±8.20	69.83±7.96	91.67±10.69	73.50±11.06	64.58±9.92	58.38±4.87
fDLRF-Net (*3 layers*)	88.83±16.64	**81.08±10.66**	**76.44±8.19**	**70.88±8.62**	92.67±9.07	**73.75±11.51**	**64.61±9.24**	**59.17±5.84**
fDLRF-Net (*4 layers*)	**90.16±16.49**	77.92±11.14	71.50±8.70	66.67±8.30	**93.17±8.76**	72.75±11.51	64.67±10.03	57.87±5.61
fDLRF-Net (*5 layers*)	88.83±16.05	75.0±13.47	68.06±8.98	65.08±7.40	92.33±9.54	70.66±11.91	60.78±8.70	55.96±5.48
LatLRR	81.50±9.73	77.25±15.02	69.17±10.77	67.00±7.58	90.17±13.03	70.25±12.91	65.66±8.39	59.83±7.68
nDLRF-Net (*2 layers*)	82.52±9.14	79.52±12.41	73.33±7.54	67.38±9.98	92.17±10.64	71.31±11.43	67.00±8.31	61.52±7.37
nDLRF-Net (*3 layers*)	85.05±12.35	**80.75±11.74**	**76.59±12.70**	**71.37±6.07**	**93.33±8.54**	**73.03±11.58**	**67.60±9.16**	**61.56±6.68**
nDLRF-Net (*4 layers*)	**94.00±7.75**	76.25±12.20	70.16±10.33	65.00±3.46	89.33±14.29	73.00±12.27	66.06±9.15	59.58±6.25
nDLRF-Net (*5 layers*)	90.20±7.09	76.71±15.48	68.83±14.03	64.63±6.48	87.33±15.52	72.11±12.39	63.50±8.18	57.13±5.79

Table 2: Numerical clustering evaluation results on the UMIST and Fashion MNIST databases.



(a) fDLRF-Net (b) nDLRF-Net (c) Examples (d) fDLRF-Net (e) nDLRF-Net

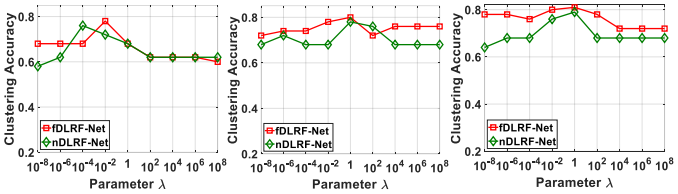
Figure 5: Clustering performance vs. varying variance on the UMIST (a-b) and Fashion MNIST (d-e) database.

6.3 Noisy Image Clustering Against Corruptions

We investigate the robustness properties against noisy cases that images are corrupted. UMIST and Fashion MNIST are used. Random Gaussian noise with different variance (100, 200, …, 500) is added to examine the robustness. For each setting, we average the result over 30 random initialization for NCut. We set the number K of clusters as 5. Some noisy images and clustering accuracy are shown in Figure 5, where the clustering results of five layers of our fDLRF-Net and nDLRF-Net are described. We see clearly that: (1) generally speaking the clustering accuracy of each method goes down with the increasing level of noise, as clustering data of high noise level is more difficult than clustering that of low noise level; (2) the best records are usually obtained in the 2-nd layer and the 3-rd layer, by comparing with the other cases.

6.4 Investigation of Parameters

UMIST database is used. DLRF-Net has one parameter λ, so we can select the most important one by a linear search from $\{10^{-8},10^{-6},...,10^{6},10^{8}\}$. We set the number K of clusters to 5 and show the analysis results of the first three layers in Figure 6. For each layer, we select the best parameter, and then we fix it to learn deeper features of next layer. We find that λ usually becomes a lager one with increasing number of layers, which is easy to understand. Since DLRF-Net recovers the subspaces progressively and learnt susbapces become clean layer by layer. Note that similar observations can be found from the other datasets, but the results will be not presented.



(a) The first layer (b) The second layer (c) The third layer

Figure 6: Parameter sensitivity analysis on the UMIST face database.

7 Conclusion

We proposed a new progressive deep latent low-rank fusion network to uncover deep hidden features and deep clustering structures. DLRF-Net discovers the subspaces by refining the principal and salient features from previous layers progressively and then fusing the subspaces. Specifically, DLRF-Net recovers hierarchical features by congregating the projective subspace and subspaces in each layer. Most existing latent low-rank coding models can also be easily extended to multilayer scenario for learning deep features. In future, more effective deep low-rank fusion strategies will be explored.

Acknowledgments

This work is supported by the NSFC (61672365, 61806035 and U1936217) and the Fundamental Research Funds for the Central Universities of China (JZ2019HGPA0102), and New Generation AI Major Project of Ministry of Science and Technology of China (2018AAA0100601). Zhao Zhang and Zheng Zhang are the corresponding authors of this paper.

References

[Acharya *et al.*, 2019] Anish Acharya, Rahul Goel, Angeliki Metallinou, Inderjit S Dhillon. Online Embedding Compression for Text Classification Using Low Rank Matrix Factorization. *In: AAAI*, Honolulu, Hawaii, 2019.

[Bao *et al.*, 2012] Bingkun Bao, Guangcan Liu, Changsheng Xu, Shuicheng Yan. Inductive robust principal component analysis. *IEEE TIP*, 21(8):3794-3800, 2012.

[Cai *et al.*, 2010] Jianfeng Cai, Emmanuel J. Cand`es, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4): 1956-1982. 2010.

[Cai *et al.*, 2017] Deng Cai, Xiaofei He, Jiawei Han. Document clustering using locality preserving indexing. *IEEE TKDE*, 17(12): 1624-1637, 2005.

[Chen *et al.*, 2019] Xi Chen, Jie Li, Yun Song, Feng Li, Jianjun Chen, Kun Yang. Low-Rank Tensor Completion for Image and Video Recovery via Capped Nuclear Norm. *IEEE Access*, 7:112142-112153, 2019.

[Chen *et al.*, 2020] Yongyong Chen, Xiaolin Xiao, Yicong Zhou. Low-Rank Quaternion Approximation for Color Image Processing. *IEEE TIP*, 29:1426-1439, 2020.

[Ding *et al.*, 2018] Zhengming Ding, Yun Fu. Deep Transfer Low-Rank Coding for Cross-Domain Learning. *IEEE TNNLS,* 30(6):1768-1779, 2018.

[Graham *et al.*, 1998] Daniel B. Graham, Nigel M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. *In: NATO ASI Series F, Computer and Systems Sciences*, Berlin, Heidelberg, 1998.

[Kim *et al.*, 2019] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, Junmo Kim. Learning Not to Learn: Training Deep Neural Networks With Biased Data. In: *IEEE CVPR*, pp.9012-9020, 2019.

[Li *et al.*, 2015] Jun Li, Heyou Chang, Jian Yang. Sparse deep stacking network for image classification. *In: AAAI*, Hyatt Regency in Austin, Texas, USA, pp.3804-3810, 2015.

[Li *et al.*, 2017a] Zechao Li, Jinhui Tang. Weakly-supervised deep nonnegative low-rank model for social image tag refinement and assignment. *In: AAAI*, San Francisco, California, USA, pp.4154-4160, 2017.

[Li *et al.*, 2017b] Jun Li, Hongfu Liu, Handong Zhao, Yun Fu. Projective low-rank subspace clustering via learning deep encoder. *In: IJCAI*, Melbourne, Australia, pp.2145-2151, 2017.

[Lin *et al.*, 2009] Zhouchen Lin, Minming Chen, Leqin Wu, Yi Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Tech. Tep.,* 2009.

[Liu *et al*., 2011] Guangcan Liu and Shuicheng Yan. Latent low-rank representa- tion for subspace segmentation and feature extraction. *In: ICCV,* Barcelona, Spain, 2011.

[Liu *et al*., 2013] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, Yi Ma. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE TPAMI*, 35(1):171-627, 2013.

[Liu *et al.*, 2019] Guangcan Liu, Zhao Zhang, Qingshan Liu, Hongkai Xiong. Robust Subspace Clustering with Compressed Data. *IEEE TIP*, 28(10): 5161-5170, 2019.

[Lu *et al.*, 2019] Canyi Lu, Xi Peng and Yunchao Wei. Low-Rank Tensor Completion With a New Tensor Nuclear Norm Induced by Invertible Linear Transforms. In: *IEEE CVPR*, Long Beach, USA, pp.5996-6004, 2019.

[Ren *et al.*, 2019] Jiahuan Ren, Zhao Zhang, Sheng Li, Yang Wang, Guangcan Liu, Shuicheng Yan and Meng Wang. Learning Hybrid Representation by Robust Dictionary Learning in Factorized Compressed Space. *IEEE TIP*, 29(1): 3941-3956, 2020.

[Shi *et al.*, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8): 888-905, 2000.

[Sim *et al.*, 2003] Terence Sim, Simon Baker, Maan Bsat. The CMU pose, illuminlation, and expression database. *IEEE TPAMI*, 25(12): 1615- 1618, 2003.

[Su *et al.*, 2019] Fang Su, Haiyang Shang, Jingyan Wang. Low-Rank Deep Convolutional Neural Network for Multitask Learning. *Computational Intelligence and Neuroscience*, 7410701:1-7410701:10, 2019.

[Wang *et al.*, 2018] Lei Wang, Zhao Zhang, Sheng Li, Guangcan Liu, Chenping Hou and Jie Qin. Similarity-Adaptive Latent Low-Rank Representation for Robust Data Representation. *In*: *PRICAI*, Nanjing, China, 2018.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, Roland Vollgraf. Fashion-Mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747v2*, 2017.

[Xie *et al.*, 2017] Jianchun Xie, Jian Yang, Jianjun Qian, Ying Tai, Hengmin M Zhang. Robust Nuclear Norm-Based Matrix Regression With Applications to Robust Face Recognition. *IEEE TIP*, 26(5):2286-2295, 2017.

[Xue *et al.*, 2019] Zhe Xue, Junping Du, Dawei Du, Siwei Lyu. Deep low-rank subspace ensemble for multi-view clustering. *Information Sciences*, 482: 210-227, 2019.

[Yu *et al.*, 2018] Yu Song, Yiquan Wu. Subspace clustering based on latent low rank representation with Frobenius-norm. *Neurocomputing*, 275:2479-2489, 2018.

[Zhang *et al.*, 2014a] Zhao Zhang. Shuicheng Yan, Mingbo Zhao. Similarity preserving low-rank representation for enhanced data representation and effective subspace learning. *Neural Networks*, 53: 81-94, 2014.

[Zhang *et al.*, 2014b] Hongyang Zhang, Zhouchen Lin, Chao Zhang, Junbin Gao. Robust latent low rank representation for subspace clustering. *Neurocomputing*, 145:369-373, 2014.

[Zhang *et al.*, 2016] Zhao Zhang, Fanzhang Li, Mingbo Zhao, Li Zhang, Shuicheng Yan. Joint Low-rank and Sparse Principal Feature Coding for Enhanced Robust Representation and Visual Classification. *IEEE TIP*, 25(6): 2429-2443, June 2016.

[Zhang *et al.*, 2019] Zhao Zhang, Jiahuan Ren, Sheng Li, Richang Hong, Zhengjun Zha, Meng Wang. Robust subspace discovery by block-diagonal adaptive locality-constrained representation. *In*: *ACM MM*, 2019.