EasyChair Preprint

№ 5370

MLLC-Net: A Progressive Multilayer Latent Low-Rank Coding Network for Deep Subspace Discovery

Ren Jiahuan, Wang Chong and Zhang Ruyue

April 24, 2021

MLLC-Net: A Progressive Multilayer Latent Low-Rank Coding Network for Deep Subspace Discovery

Authors

Abstract **— Low-rank representation is a powerful and popular algorithm for discovering the subspaces from samples and has obtained the impressive performance, however it cannot capture deep hierarchical information hidden in data due to the essence of single-layer structures. In this paper, we explore the deep image representation in a progressive way by presenting a new strategy to extend existing single-layer latent low-rank models into multiple layers. Technically, we propose a *Multilayer Latent Low-rank Coding Network* termed MLLC-Net to uncover deep features and the clustering structures embedded in the latent subspace. The basic idea of MLLC-Net in each layer is to refine the principal and salient features progressively from the previous layers by fusing the subspaces, which can potentially learn more accurate features and subspaces for image representation learning and clustering. To learn deep hidden information, MLLC-Net inputs the shallow features from the last layers into the subsequent layers. Then, it recovers hierarchical information and deeper features by respectively congregating the projective subspaces and representation subspaces in each layer. As such, one can learn deeper subspaces and can also ensure the representation learning of deeper layers to remove the noise and discover the underlying clean subspaces, which will be versified by the simulations. It is noteworthy that the framework of our MLLC-Net is applicable to most existing latent low-rank representation models, i.e., existing latent models can be easily extended to multilayer scenario using our MLLC-Net. Extensive results on real databases show that our models can deliver enhanced performance over existing related techniques.**

Index Terms **— *Deep subspace discovery; progressive multilayer latent low-rank coding network; image representation; clustering***

I. INTRODUCTION

Representation learning from the observation data to discover the explanatory factors and the underlying subspace structures for subsequent tasks (such as classification or clustering) is the core of both traditional and deep learning methods [1-6][29-38]. Although lots of low-rank/sparse subspace recovery based and matrix factorization based representation learning methods can be used [5-15][21-22], the task of representation learning is still challenging in reality, because most of real data usually contain various noise, high-dimensional features, and increasing complexity and diversity of attributes [9][25][46-48].

It is noteworthy that real data can usually be characterized by using low-rank or low-dimensional subspaces, low-rank coding methods have been playing an important role in many emerging image processing applications (for instance image representation, image restoration and background/foreground separation). Specifically, low-rank coding methods can recover the underlying subspace structures and obtain the notable features [5-7], among which *Low-Rank Representation* (LRR) [5] is one of the most classical methods to recover mult-subspaces jointly. But it is essentially a transductive method, i.e., it fails to handle new data and extract low-rank features efficiently. To address the out-of-sample problem, *Inductive Robust Principal Component Analysis* (IRPCA) [8] was recently proposed, which computes a low-rank projection so that the input data can be mapped into underlying subspaces and the errors can be recovered. As such, IRPCA can handle the outside new data efficiently and extract low-rank features from input data, but note that it cannot obtain block-diagonal subspace structures for clustering, i.e., it cannot handle the subspace segmentation issue as LRR. To enable a solution for both subspace segmentation and feature extraction, *Latent Low-Rank Representation* (LatLRR) [9], as a combination of LRR and IRPCA, was then proposed, which seamlessly integrates the subspace segmentation and feature extraction into a unified framework. Technically, LatLRR decomposes the input data into a principal feature part, a salient feature part and a sparse error part. More importantly, LatLRR represents the input data by both observed and hidden data, so the insufficient sampling issue can be resolved. Although LatLRR obtained the enhanced clustering results than LRR, it suffers from higher-computational cost, since it applies the Nuclear-norm to approximate the rank function to preserve the low-rank properties of the projection and subspace structures as LRR, while the computation of the Nuclear-norm needs the Singular Value Decomposition (SVD) of matrices at each iteration, which is usually time-consuming especially for large-scale datasets [12]. To improve the computational time efficiency, *Frobenius-norm based LatLRR* (FLLRR) [12] that approximates the rank function using the Frobenius-norm was proposed. However, Frobenius-norm is very sensitive to noise and outliers in input data, which may produce inaccurate representations in theory.

It is easy for us to conclude from the above analyses that each low-rank coding model is defined based on specific properties, and they have some complementary properties. But note that they also have a common drawback, i.e., all above-mentioned methods are “shallow” frameworks that only have single-layer structures. As a result, they cannot capture deep semantic information and deep subspace structures hidden in the complexvisual data. However, due to the strong representation learning ability of deep networks, those deep low-rank coding methods equipped with carefully designed hierarchical structures should also deliver enhanced performance and attractive properties inlearning effective feature representations. As such, researchers have also devoted to designing the low-rank coding based deep network models, such as *Weakly-supervised Deep Nonnegative Low-rank Model* (WDNL) [24], which focused on finding the intrinsic relationships between images and tags by removing the noise or irrelevant tags. However, it fails to handle images directly and the results may also contains negative values and are usually incomplete. Another deep model is *Deep Low-Rank Subspace Ensemble* (DLRSE) [23], where the Frobenius-norm is used as a low-rank constraint. DLRSE uses the

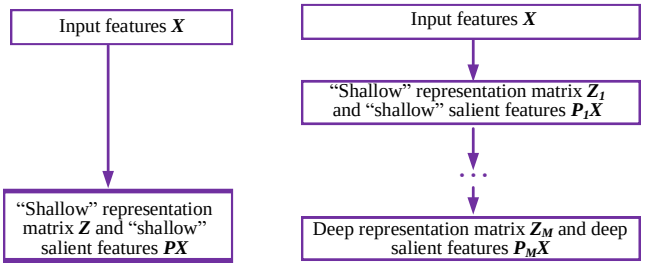


Fig. 1: Comparison between our MLLC-Net (b) and the traditional latent low-rank coding methods (a) .

deep matrixfactorization to capture the diverse and hierarchical structures of data, and obtains the low-rank representation from extracted factors. Note that DLRSE is originally designed for multi-view clustering, which clearly differs from the task of this work. To handle and cluster big data effectively, a *Projective Low-rank Subspace Clustering via Learning Deep Encoder* (PLrSC) [45] was recently proposed. PLrSC used a small dataset randomly sampled from a big dataset for training a deep encoder by the proposed predictive low-rank decomposition and then the deep encoder is used to compute the low-rank representations of all samples. However, it will be difficult and tricky to choose the sampling frequency based on different real datasets in practice.

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

(1) Technically, a simple yet effective deep representation learning framework called *Multilayer Latent Low-rank Coding Network* (MLLC-Net) is proposed. MLLC-Net can mine deep hierarchical information and subspace structures hidden in data. The advantage of this practice is that multiple layer low-rank coding structures can obtain rich and useful hidden hierarchical information that has a great potential in learning more powerful deep representations and deep subspace structures. To be specific, our MLLC-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 subspaces and representation spaces in each layer to produce more accurate results. It is noteworthy that we compare the mechanism of the traditional single-layer low-rank coding models with our MLLC-Net in Fig.1.

(2) The framework of our MLLC-Net is simple, general and easy to extend. More specifically, most existing latent low-rank coding models (e.g., LatLRR and FLLRR) can be easily extended from single layer to multiple layers using the framework of MLLC-Net. In this paper, we mainly explain our basic idea rather than deriving a complex formulation, two simple deep network models are constructed basesd on embedding LatLRR and FLLRR into our framework as the examples for multi-layer low-rank coding to learn deep features, which we call *Nuclear-norm based MLLC-Net* (nMLLC-Net) and *Frobenius-norm based fast MLLC-Net* (fMLLC-Net), respectively.

(3) Extensive simulations on real databases demonstrate that both nMLLC-Net and fMLLC-Net are able to obtain enhanced performance than other related single-layer models. That is, the multi-layer idea of MLLC-Net is feasible and effective.

The paper is outlined as follows. In Section II, we review the related work briefly. In Section III, we present the problem and optimization of MLLC-Net. Section IV shows the experimental setting and results. Finally, the paper is concluded in Section V.

II. Related Work

In this section, we review the closely-related single-layer and multi-layer low-rank coding algorithms briefly.

A. LatLRR and FLLRR

Given a set of training samples $X=[x_1,x_2\cdots,x_N]\in\mathbb{R}^{n\times N}$, 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 by using unobserved hidden data X_H to extend the dictionary and overcome the insufficient data sampling issue. Specifically, LatLRR considers the following coding problem:

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

where $rank(\bullet)$ is the rank function and X_O is observed data matrix. Suppose 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 the sparse L1-norm on the error term E, LatLRR can recover the hidden effects by minimizing the following objective function:

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

where $\|Z\|_*$ denotes the Nuclear-norm of Z [5] [22], i.e., the sum of its singular values, XZ and LX are principal features and salient features respectively, and λ is a positive parameter.

Note that LatLRR uses the Nuclear-norm constraint on Z and L, thus the SVD process will be involved in each iteration of optimization, which is usually time-consuming. It is also worth noting that Frobenius-norm $\|\bullet\|_F$ can also be used as the convex surrogate of the rank function [12]. Besides, the optimization of the Frobenius-norm is efficient compared with Nuclear-norm. As such, by using the Frobenius-norm to approximate the rank function, the criterion 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 the subspace can be similarly recovered by Z and the notable features can be extracted by L.

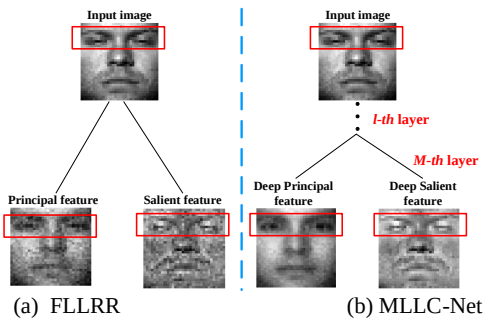


Fig.3: Visual comparison of the learned principal and salient features by the traditional FLLRR and our MLLC-Net.

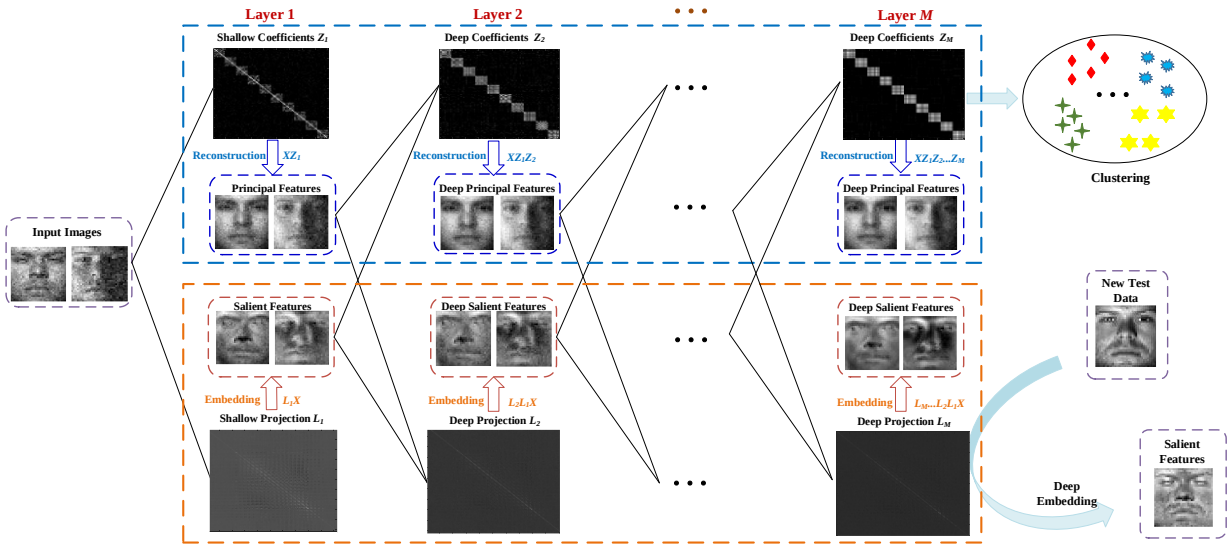


Fig. 2: The framework and flow-chart of our proposed MLLC-Net algorithm.

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

Supposing 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 denoted by using S_i, where N' is the number of samples in all subspaces. Let m_i be the number of samples in Y^i of S_i, i.e., $N'=\sum_{i=1}^{k}m_i$, then PLrSC assumes that the set $X=\left[X^1,\cdots X^i,\cdots X^k\right]\in\mathbb{R}^{n\times N}$ is a small dataset sampled randomly from the big dataset and X is still sufficient. Then, PLrSC learns a non-iterative deep encoder $f_{de}(X;\theta)$ firstly, where θ is the learning parameter to approximate the low-rank representations. Then, the deep encoder is employed to obtain the low-rank codes for replacing the costly non-linear Singular Value Thresholding (SVT) [26] operations. Thus, the predictive low-rank decomposition of PLrSC can be written as the following nonconvex optimization problem:

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

where $E\in\mathbb{R}^{n\times N}$ is a sparse error term, λ is the regularization parameter for E, γ is a control parameter and $\left\|Z-f_{de}(X;\theta)\right\|_F^2$ is an approximation term. $f_{de}(X;\theta)=g\left(W^M\cdots g\left(W^i\cdots g\left(W^2X\right)\right)\right)$ 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 is the number of the units in the *i-th* layer ($l_1=d$ and $l_M=N$). Then, PLrSC employs an alternating direction algorithm (ADM) [5] and a gradient descent algorithm (GD) [43] to optimize the problem. Finally, PLrSC employs the landmark-based spectral clustering (LSC) to cluster the original big dataset Y.

III. Progressive Multilayer Latent Low-rank Coding Network (MLLC-Net)

A. Motivation

We describe the motivation of our method using an illustration. We mainly compare the principal features and salient features obtained by the traditional single-layer methods, e.g., FLLRR, and our MLLC-Net framework in Fig.3. We can observe that the learnt principal features of FLLRR still contain some distinguishing features, and the salient features may also contain some principal features in the red rectangle for the single-layer structure of FLLRR. While the deep principal features of our MLLC-Net are different from that of FLLRR, since it contains less notable features. It can also be found that the learned deep features by our MLLC-Net contain less noise, which means that our MLLC-Net can recovery the subspace more accurately.

B. Objective Function

To improve the abilities of learning representation and features, our MLLC-Net designs a hierarchical and progressive approach, i.e., in the proposed deep architecture, the image representation in the uncovered subspaces are learned 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 subsequent l-th layer, which are further decomposed into a deep low-rank principal feature part, a deep low-rank salient feature part and a deep sparse error. The whole framework of our MLLC-Net is illustrated in Fig.2. Assuming that MLLC-Net has M layers, the decomposition process of our proposed MLLC-Net framework in the l-th layer can be presented as follows:

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

where Z_l^1 and Z_l^2 denote the deep coefficient matrices that are further learned from $XZ_0Z_1\ldots Z_{l-1}$ and $L_{l-1}\ldots L_1L_0X$ respectively, 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 mainly added to simplify the description, which are initialized to the identity matrix in our model, i.e., the input of the first layer is the original input. 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 (l-1)-th layer. As such, intuitively from the proposed multilayer leaning process, the deep principal features $XZ_0Z_1\ldots Z_{l-1}$ and deep sali-

ent features $L_{l-1}\ldots L_1L_0X$ are learnt progressively from layers, i.e., extracting fine-grained features from layer to layer.

Finally, from the l-th layer, we compute the deep coefficients and projection matrix by fusion as follows:

$$Z_l = \left(Z_l^1 + Z_l^2\right)/2,\ L_l = \left(L_l^1 + L_l^2\right)/2. \tag{6}$$

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

$$\begin{aligned} XZ_0Z_1\cdots Z_{l-1}Z_l &= XZ_0Z_1\cdots Z_{l-1}\left(Z_l^1 + Z_l^2\right)/2 \\ L_lL_{l-1}\cdots L_1L_0X &= \left(L_l^1 + L_l^2\right)L_{l-1}\cdots L_1L_0X/2 \end{aligned}. \tag{7}$$

It is noteworthy that the above fusion operation can potentially make the learnt feature representations more accurate by fusing the feature information from the deep principal and deep salient features in the previous layers. In addition, the above averaging operation can prevent the feature information loss and can also balance information from deep principal features and deep salient features in each layer.

Based on the above descriptions, we can have the following formulation for our MLLC-Net in the l-th layer:

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

where $\|\bullet\|_p$ is the matrix p-norm, which can be Nuclear-norm or squared Frobenius-norm and λ_l is a positive tunable parameter that replies on the noise level of data [9]. In this paper, we name the Nuclear-norm based MLLC-Net as *nMLLC-Net* and name squared Frobenius-norm based MLLC-Net as *fMLLC-Net*.

More specifically, the objective function of our nMLLC-Net can be defined as follows for deep subspace discovery:

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

Similarly, the objective function of our proposed fMLLC-Net can be presented as follows for deep representations:

$$\begin{aligned} &\min_{Z_l,L_l,E_l} \sum_{l=1}^{M}\left\{\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2 + \left\|Z_l^2\right\|_F^2 + \left\|L_l^1\right\|_F^2 + \left\|L_l^2\right\|_F^2\right) + \lambda_l\left(\left\|E_l^1\right\|_1 + \left\|E_l^2\right\|_1\right)\right\} \\ &s.t.\ XZ_0\ldots Z_{l-1} = \left(XZ_0\ldots Z_{l-1}\right)Z_l^1 + L_l^1\left(XZ_0\ldots Z_{l-1}\right) + E_l^1 \\ &\quad L_{l-1}\ldots L_0X = \left(L_{l-1}\ldots L_0X\right)Z_l^2 + L_l^2\left(L_{l-1}\ldots L_0X\right) + E_l^2 \end{aligned}. \tag{10}$$

In Fig.4, we illustrate the detailed decomposition process of our MLLC-Net in each layer, where we can see that MLLC-Net can not only capture deep hidden information and features layer by layer in a progressive way, but also remove those redundant information and noise in data progressively. Next, we describe the optimization procedures of our MLLC-Net.

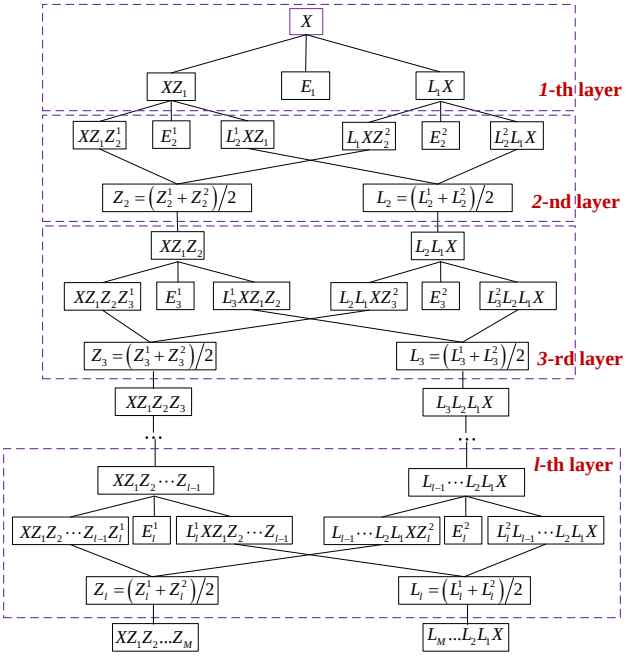


Fig.4: The detailed decomposition process of our MLLC-Net.

IV. Optimization of MLLC-Net

We mainly describe the optimization procedures of MLLC-Net. It is easy to check that the variables Z_l, P_l and E_l depend on each other, so they cannot be solved directly. Following the common procedures, we solve the variables Z_l, P_l and E_l alternately, i.e., solving one of them by fixing the others. For efficiency, we also use the inexact Augmented Lagrange Multiplier (Inexact ALM) method [13]. Due to the page limitation, we mainly detail the optimization process of our fMLLC-Net, since the optimization of nMLLC-Net is similar as solving fMLLC-Net.

To simplify the descriptions of the optimization, we train the model layer by layer. Specifically, fMLLC-Net with M layers can be divided into M sub-problems. To learn features in the l-th layer (l=1, 2,…, M), the target function can be defined as

$$\begin{aligned} &\min_{Z_l,L_l,E_l} \frac{1}{2}\left(\left\|Z_l^1\right\|_F^2 + \left\|Z_l^2\right\|_F^2 + \left\|L_l^1\right\|_F^2 + \left\|L_l^2\right\|_F^2\right) + \lambda_l\left(\left\|E_l^1\right\|_1 + \left\|E_l^2\right\|_1\right) \\ &s.t.\ XZ_0\ldots Z_{l-1} = \left(XZ_0\ldots Z_{l-1}\right)Z_l^1 + L_l^1\left(XZ_0\ldots Z_{l-1}\right) + E_l^1 \\ &\quad L_{l-1}\ldots L_0X = \left(L_{l-1}\ldots L_0X\right)Z_l^2 + L_l^2\left(L_{l-1}\ldots L_0X\right) + E_l^2 \end{aligned}, \tag{11}$$

Algorithm 1: Solving Eq.(8) 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.(15-16), 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.(18-19), 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.(20-21);
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 $\left\{\left\|P_{l-1}-P_{l-1}Z_l^1-L_l^1P_{l-1}-E_l^1\right\|_\infty, \left\|S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\|_\infty\right\}\le\varepsilon$, stop; else $t=t+1$.

End while

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

Algorithm 2: Optimization procedures of fMLLC-Net

Input: Input data matrix X, layer number M, tunable parameter λ_1. Initialize Z_0 and L_0 to be the identity matrices.

For l=1 **to** M, **do**

1. Compute the reconstructed data P_{l-1} and mapped data S_{l-1} as $P_{l-1} = XZ_0...Z_{l-1}$ and $S_{l-1}=L_{l-1}...L_0X$;
2. Solve Z_l, L_l, E_l^1, E_l^2 by optimizing Eq.(11) in Algorithm 1.

End

Output: Z_l and L_l, where l=1,2,…M.

Let $P_{l-1} = XZ_0...Z_{l-1}$ and $S_{l-1}=L_{l-1}...L_0X$ be two auxiliary matrices, the Lagrange function of Eq.(11) can then be defined as

$$\begin{aligned}\wp(Z_l,L_l,E_l) &= \frac{1}{2}\left(\left\|Z_l^1\right\|_F^2+\left\|Z_l^2\right\|_F^2+\left\|L_l^1\right\|_F^2+\left\|L_l^2\right\|_F^2\right)+\lambda\left(\left\|E_l^1\right\|_1+\left\|E_l^2\right\|_1\right)\\ &+\left\langle Y_l^1, P_{l-1}-P_{l-1}Z_l^1-L_l^1\,P_{l-1}-E_l^1\right\rangle+\left\langle Y_l^2, S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\rangle,\\ &+\frac{\mu_l}{2}\left(\left\|P_{l-1}-P_{l-1}Z_l^1-L_l^1\,P_{l-1}-E_l^1\right\|_F^2+\left\|S_{l-1}-S_{l-1}Z_l^2-L_l^2S_{l-1}-E_l^2\right\|_F^2\right)\end{aligned} \tag{12}$$

where $\langle A,B\rangle = Tr\left(A^TB\right)$ is the inner product between matrices A and B, Y_l^1 and Y_l^2 are Lagrange multipliers, and μ_l denotes a positive weighting parameter.

By using the inexact ALM, fMLLC-Net updates the variables by solving the augmented Lagrange function $\wp(Z_l,L_l,E_l)$:

$$\begin{aligned}&\left(Z_l^{t+1},L_l^{t+1},E_l^{t+1}\right)=\arg\min_{Z_l,L_l,E_l}\wp^{t+1}\left(Z_l^t,L_l^t,E_l^t\right)\\ &\left(Y_l^1\right)^{t+1}=\left(Y_l^1\right)^t+\mu_l\left(P_{l-1}^{t+1}-P_{l-1}^{t+1}\left(Z_l^1\right)^{t+1}-\left(L_l^1\right)^{t+1}P_{l-1}^{t+1}-\left(E_l^1\right)^{t+1}\right)\\ &\left(Y_l^2\right)^{t+1}=\left(Y_l^2\right)^t+\mu_l\left(S_{l-1}^{t+1}-S_{l-1}^{t+1}\left(Z_l^2\right)^{t+1}-\left(L_l^2\right)^{t+1}S_{l-1}^{t+1}-\left(E_l^2\right)^{t+1}\right)\\ &\mu_l^{t+1}=\min\left(\eta\mu_l^t,\max_{\mu_l}\right)\end{aligned} \tag{13}$$

Then, the optimization procedures of our fMLLC-Net in the l-th layer can be detailed as follows:

Fix others, update the coefficients Z_l: For the optimization of Z_l, we need to calculate both Z_l^1 and Z_l^2. By removing the irrelevant terms from the Lagrange function, we can update Z_l^1 and Z_l^2 by the following reduced problem:

$$\begin{aligned}\wp\left(Z_l^1,Z_l^2\right)=&\frac{1}{2}\left(\left\|Z_l^1\right\|_F^2+\left\|Z_l^2\right\|_F^2\right)+\left\langle Y_l^1,\Xi_{l-1}-P_{l-1}Z_l^1\right\rangle\\ &+\left\langle Y_l^2,\Omega_{l-1}-S_{l-1}Z_l^2\right\rangle+\frac{\mu_l}{2}\left(\left\|\Xi_{l-1}-P_{l-1}Z_l^1\right\|_F^2+\left\|\Omega_{l-1}-S_{l-1}Z_l^2\right\|_F^2\right)\end{aligned}, \tag{14}$$

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:

$$\left(Z_l^1\right)^{t+1}=\left(I+\mu_l^tP_{l-1}^TP_{l-1}\right)^{-1}\mu_l^tP_{l-1}^T\left(\Xi_{l-1}^t+\left(Y_l^1\right)^t/\mu_l^t\right), \tag{15}$$

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

$$\left(Z_l^2\right)^{t+1}=\left(I+\mu_l^tS_{l-1}^TS_{l-1}\right)^{-1}\mu_l^tS_{l-1}^T\left(\Omega_{l-1}^t+\left(Y_l^2\right)^t/\mu_l^t\right), \tag{16}$$

where $\Omega_{l-1}^t=S_{l-1}-\left(L_l^2\right)^tS_{l-1}-\left(E_l^2\right)^t$. After optimizing $\left(Z_l^1\right)^{t+1}$ 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 the projection matrix L_l: In this step, similar to the optimization of Z_l, we need to calculate L_l^1 and L_l^2. By removing irrelevant terms from the Lagrange function, we can update L_l^1 and L_l^2 by the following problem:

$$\begin{aligned}\wp\left(L_l^1,L_l^2\right)=&\frac{1}{2}\left(\left\|L_l^1\right\|_F^2+\left\|L_l^2\right\|_F^2\right)+\left\langle Y_l^1,\Xi_{l-1}-L_l^1\,P_{l-1}\right\rangle\\ &+\left\langle Y_l^2,\Omega_{l-1}-L_l^2S_{l-1}\right\rangle+\frac{\mu_l}{2}\left(\left\|\Xi_{l-1}-L_l^1\,P_{l-1}\right\|_F^2+\left\|\Omega_{l-1}-L_l^2S_{l-1}\right\|_F^2\right)\end{aligned}, \tag{17}$$

where $\Xi_{l-1}=P_{l-1}-P_{l-1}Z_l^1-E_l^1$ and $\Omega_{l-1}=S_{l-1}-S_{l-1}Z_l^2-E_l^2$. By taking the derivative of $\wp\left(L_l^1,L_l^2\right)$ w.r.t. L_l^1 and L_l^2 respectively, and zeroing the derivatives, we can easily have

$$\left(L_l^1\right)^{t+1}=\mu_l^t\left(\Xi_{l-1}^{t+1}+\left(Y_l^1\right)^t/\mu_l^t\right)\left(I+\mu P_{l-1}^TP_{l-1}\right)^{-1}, \tag{18}$$

$$\left(L_l^2\right)^{t+1}=\mu_t\left(\Omega_{l-1}^{t+1}+\left(Y_l^2\right)^t/\mu_t\right)\left(I+\mu S_{l-1}{}^TS_{l-1}\right)^{-1}, \tag{19}$$

where $\Xi_{l-1}^{t+1}=P_{l-1}-P_{l-1}\left(Z_l^1\right)^{t+1}-\left(E_l^1\right)^t$, $\Omega_{l-1}^{t+1}=S_{l-1}-S_{l-1}Z_{l\,(t+1)}^2-E_{l\,(t)}^2$. After optimizing the projection sub-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 the sparse errors E_l^1 and E_l^2 : When the other variables are updated in the $(t+1)$-th iteration, we can easily solve E_l^1 and E_l^2. Specifically, by taking the derivative of the Lagrange function w.r.t. E_l^1 and E_l^2 respectively, and then zeroing the derivatives, we can infer E_l^1 and E_l^2 as follows:

$$\left(E_l^1\right)^{t+1}=\arg\min_{E_l^1}\frac{\lambda}{\mu_l^t}\left\|E_l^1\right\|_1+\frac{1}{2}\left\|E_l^1-\left(P_{l-1}-\Delta_l^1\right)\right\|_F^2, \tag{20}$$

$$\left(E_l^2\right)^{t+1}=\arg\min_{E_l^2}\frac{\lambda}{\mu_l^t}\left\|E_l^2\right\|_1+\frac{1}{2}\left\|E_l^2-\left(S_{l-1}-\Delta_l^2\right)\right\|_F^2, \tag{21}$$

which can be easily solved by using the shrinkage operator [13], where Δ_l^1 and Δ_l^2 are two auxiliary matrices defined as $\Delta_l^1=P_{l-1}\left(Z_l^1\right)^{t+1}-\left(L_l^1\right)^{t+1}P_{l-1}+\left(Y_l^1\right)^t/\mu_l^t$, $\Delta_l^2=S_{l-1}\left(Z_l^2\right)^{t+1}-\left(L_l^2\right)^{t+1}S_{l-1}+\left(Y_l^2\right)^t/\mu_l^t$.

For complete presentation of our fMLLC-Net, we summarize the optimization procedures of solving the sub-problem of Eq.(11) in the l-th layer in Algorithm 1. We also summarize the whole procedures of our fMLLC-Net algorithm in Algorithm 2.

V. Discussion

A. Relationship Analysis

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

$$\begin{aligned}&\min_{Z_1,L_1,E_1}\frac{1}{2}\left(\left\|Z_1\right\|_p+\left\|L_1\right\|_p\right)+\lambda_1\left\|E_1\right\|_1\\ &s.t.\,XZ_0=XZ_0Z_1+L_1XZ_0+E_1,\,L_0X=L_0XZ_1+L_1L_0X+E_1\end{aligned}. \tag{22}$$

Since Z_0 and L_0 are initialized to the identity matrices in the optimization, i.e., the two constraints are the same, the above formulation can be further reduced to

$$\min_{Z_1,L_1,E_1}\frac{1}{2}\left(\left\|Z_1\right\|_p+\left\|L_1\right\|_p\right)+\lambda_1\left\|E_1\right\|_1,\ s.t.\,X=XZ_1+L_1X+E_1. \tag{23}$$

It is clear that when we use the Frobenius-norm to constrain the matrices Z_1 and L_1, the above problem identifies the problem of FLLRR; while we use the Nuclear-norm as constraints,

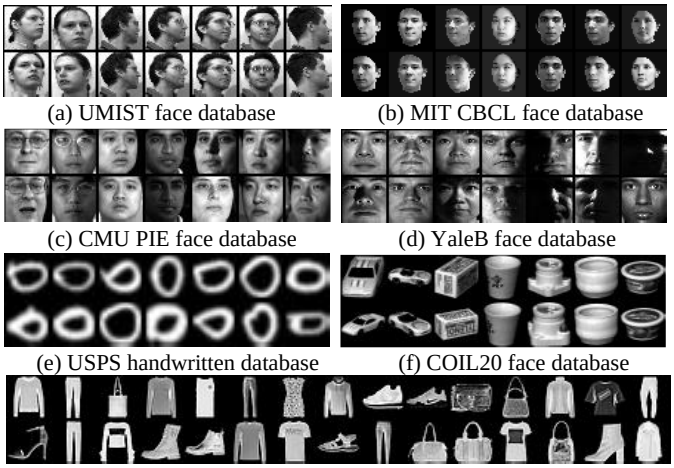
the resulting problem is identical to the objective function of LatLRR. That is, both FLLRR and LatLRR are special causes of our proposed MLLC-Net framework. Since both FLLRR and LatLRR are single-layer models, and they cannot extract deep feature information, our proposed MLLC-Net will be superior to them for subspace discovery and clustering.

B. Computational Time Complexity

We analyze the time complexity of each layer in Algorithm 1. For our proposed fMLLC-Net, no SVD is needed 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 infer that the total time complexity of our fMLLC-Net is M times that of each layer, where M is the number of layers, which is usually a small value. For our nMLLC-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.

VI. Experimental Results And Analysis

In this section, we conduct experiments to evaluate the effectiveness of our fMLLC-Net and nMLLC-Net algorithms, along with illustrating the visual and quantitative comparison results with other related representation learning algorithms, including FLLRR [12], LatLRR [9], Laplacian Regularized LRR (rLRR) [10], Similarity-Adaptive LatLRR (SA-LatLRR) [33], Robust LatLRR (rLatLRR)[14], LRR [5] and PLrSC [45]. Seven public real image databases are involved, including four face image databases (i.e., CMU PIE [41], MIT CBCL [42], UMIST [39] and YaleB [40]), COIL20 object database [28], USPS handwritten digits database [32] and Fashion MINIST database [44]. The details about the evaluated datasets are described in Table I and some image examples of the datasets are shown in Fig.5. In this study, we follow the common procedure to resize each face or object image into 32×32 pixels for efficiency, the digit image is resized into 16×16 pixels and the images of Fashion MINIST are resized into 28×28 pixels. We perform experiments on a PC with Intel (R) Core (TM) i7-7700 CPU @ 3.6 GHz 8G.



(a) UMIST face database (b) MIT CBCL face database

(c) CMU PIE face database (d) YaleB face database

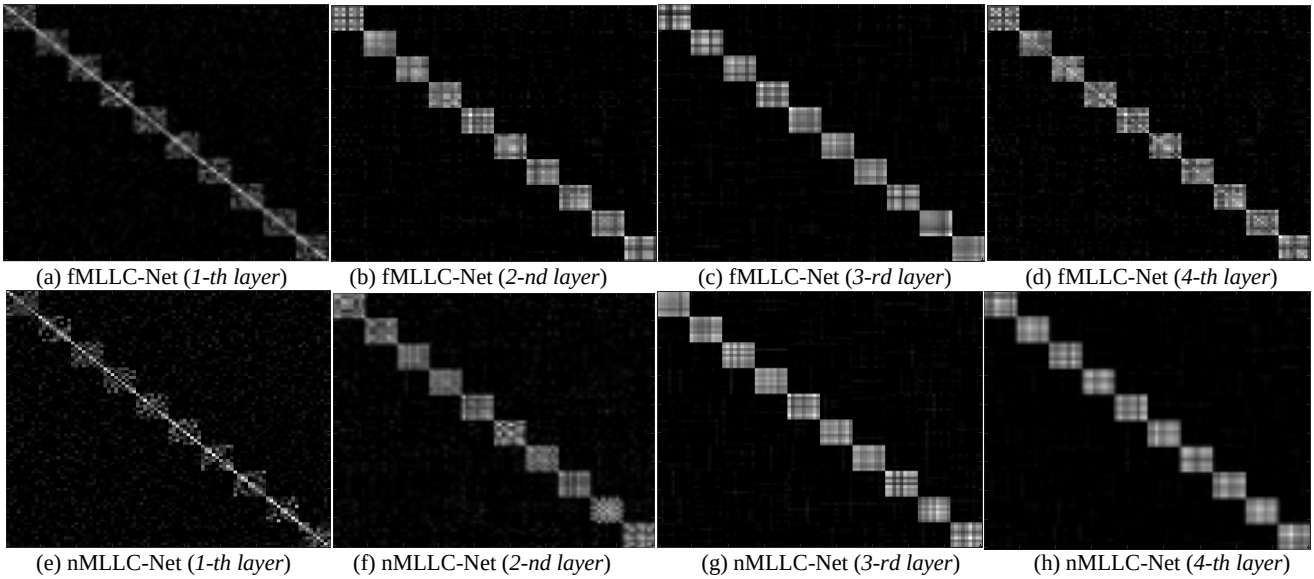
(e) USPS handwritten database (f) COIL20 face database

(g) Fashion MINIST database

Fig. 5: Image examples of the evaluated image databases.

TABLE I. Descriptions of Used Image Datasets.

Dataset Name	*# Samples*	*# Dim*	*# Classes*
CMU PIE	11554	1024	68
UMIST face	1012	1024	20
MIT CBCL face	3240	1024	10
YaleB face	2414	1024	38
COIL20 object	1440	1024	20
USPS digits	9298	256	10
Fashion MINIST	70000	784	10



(a) fMLLC-Net (*1-th layer*) (b) fMLLC-Net (*2-nd layer*) (c) fMLLC-Net (*3-rd layer*) (d) fMLLC-Net (*4-th layer*)

(e) nMLLC-Net (*1-th layer*) (f) nMLLC-Net (*2-nd layer*) (g) nMLLC-Net (*3-rd layer*) (h) nMLLC-Net (*4-th layer*)

Fig.6: Visual comparison of the representation coefficients matrix Z of MLLC-Net based on FLLRR and LatLRR.

A. Visual Image Analysis by Visualization

In this study, we provide some visualization results to show the block-diagonal structures of the representation coefficients Z in different layers, since the quality of the coefficients determines the performance for subspace clustering. We also compare the recovered principal features XZ in different layers.

Visualization of coefficients matrix *Z*. To represent given data accurately, the learnt coding coefficients matrix Z should be block-diagonal, where 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. In this study, we follow [9] to construct 10 independent subspaces $\{S_i\}_{i=1}^{10}$, whose bases $\{H_i\}_{i=1}^{10}$ are computed by $H_{i+1} = GH_i, 1 \le i \le 9$, where G is a random rotation and H_i is a random column orthogonal matrix whose dimension is 200×10 and each subspace has a fixed rank of 10. We construct a data matrix $X = [X_1, X_2, \ldots X_{10}] \times \mathbb{R}^{200 \times 10}$ by sampling 9 (that is smaller than the rank of subspace) data vectors from each subspace by $X_i = H_i C_i, 1 \le i \le 9$ with C_i is a 10×9 *i.i.d.* $N(0,1)$ data matrix. Then, we employ this artificial data matrix for the coefficient coding by our MLLC-Net. The visualization of the coefficient matrices Z of the first four layers of our fMLLCC-Net and nMLLCC-Net methods are shown in Fig.6, respectively. We observe that all computed coefficients matrices Z have block-diagonal structures. But compared with the *1-th* layer, the results of the *2-nd* layer, *3-rd* layer and *4-th* layer have less noise and have less number of wrong inter-class connections. The major reason is that the proposed framework of our MLLC-Net can mine the deep information and structure hidden in the subspaces by multi-layer representation learning. It can also be found that the discovered 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 Z in the *3-rd* and *4-th* layers is small, i.e., our MLLC-Net can remove the noise contained in the features and recovery the subspace structures of Z by using small number of layers. In other words, the structures of the coefficient matrix Z will not become better, even though we use more layers involving higher computational cost. It should be noted that in all the simulations the representation results of our fMLLC-Net and nMLLC-Net in the *1-th* layer corresponds to the results of FLLRR and LatLRR, respectively.

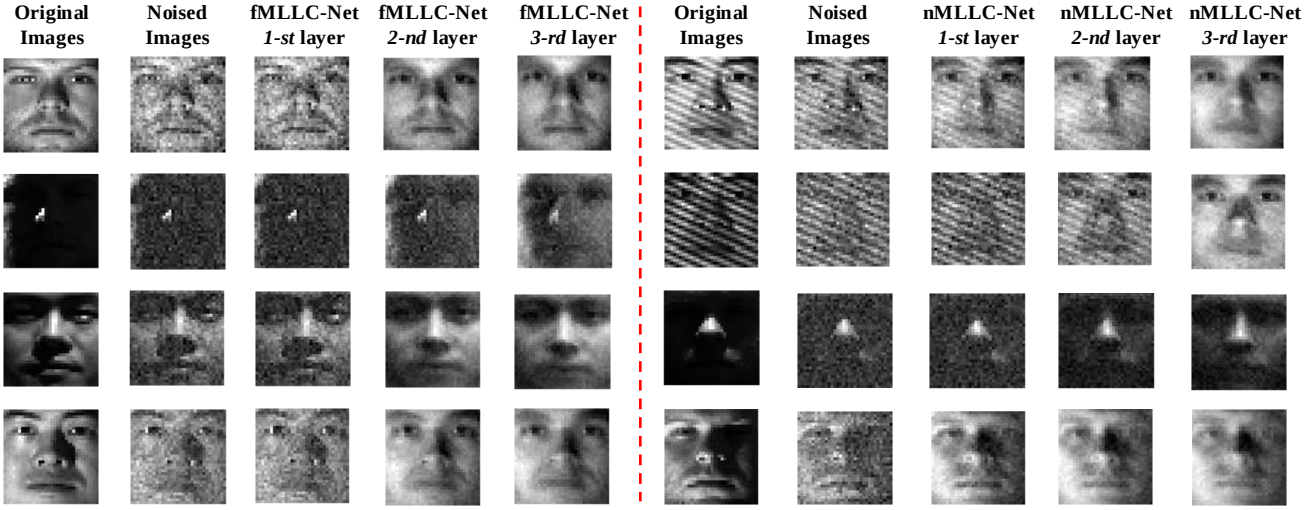


Fig.7: Recovered principal features of our fMLLC-Net and nMLLC-Net on the YaleB face image database.

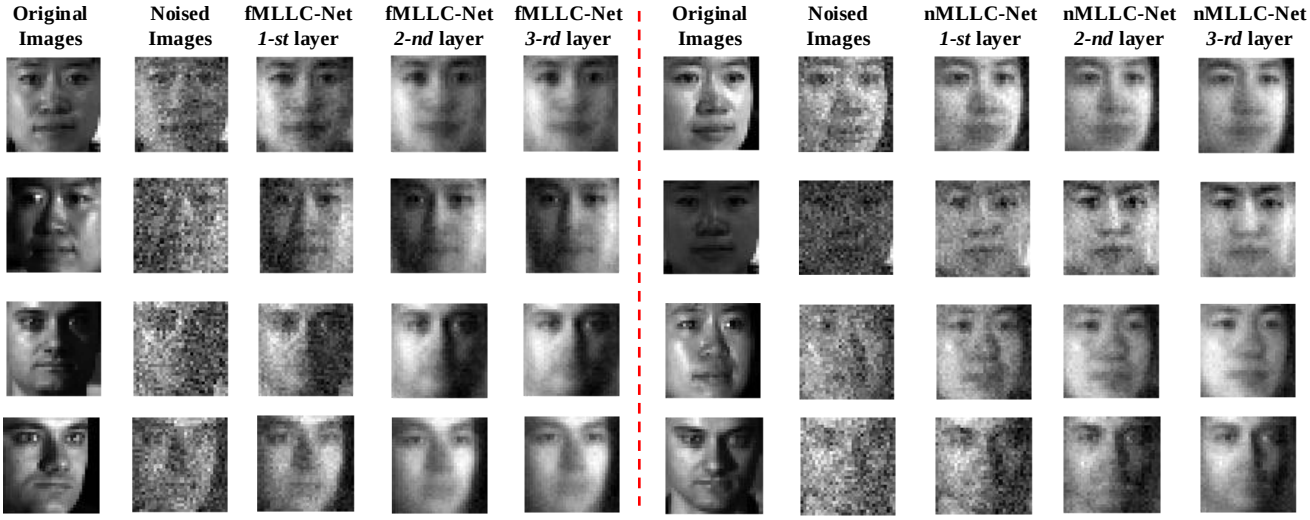


Fig.8: Recovered principal features of our fMLLC-Net and nMLLC-Net on the CMU PIE face image database.

Visualization of recovered features *XZ*. We then evaluate our fMLLC-Net and nMLLC-Net by visualizing the recovered principal features *XZ*. Given a face data matrix *X*, our methods decompose it into principal features *XZ*, salient features *LX* and a sparse error *E*. Then, our methods further input learnt shallow features to the next layer to obtain deep features. In this study, YaleB and CMU PIE face databases are used. YaleB database has more facial variations, including the illumination changes, expressions and occlusions, while CMU PIE face database has 68 persons with 41368 images under varying pose, illumination

and facial expression. In this study, 170 near frontal images per person are used for CMU PIE, which contains five near frontal poses (C05, C07, C09, C27, and C29) and the face images have different illuminations, lighting and expressions. To verify the robustness of our fMLLC-Net nMLLC-Net, random Gaussian noise with variance being 500 is included into the image data. For the YaleB and CMU PIE, some original face images, noisy face images and the recovered principal features of the noisy images in the first three layers are shown in Fig.7 and Fig.8, respectively. We can see that our fMLLC-Net and nMLLC-Net can effectively remove the shadow and noise in the face images in a progressive way, compared with the recovered results of FLLRR and LatLRR, which means that the deep structures of our proposed framework are effective for data representation.

B. Application to Image Recognition

Since the proposed latent low-rank coding network can deliver a projection for extracting the salient features, we also evaluate our fMLLC-Net and nMLLC-Net methods for image recognition and mainly compare the results with those of the regular FLLRR and LatLRR. In this study, the MIT CBCL and UMIST face databases are evaluated. UMIST database has 1012 images from 20 different individuals that changes the poses from profile to frontal views. Images are numbered consecutively as they were taken. MIT CBCL contains the synthetic face images of 10 persons (324 images per person, i.e., 3240 face images totally) rendered from 3D head models. For classification, we split each database into a training set and an unlabeled test set, where the training set is employed for representation learning, and the test set is mainly used to evaluate the accuracies. We mainly compare classification results of the first four layers.

The face classification results on MIT CBCL and UMIST are described in Table II and Table III respectively, where we show the results of our fMLLC-Net and nMLLC-Net methods in the 2-nd to the 4-th layers. Based on the learned projection in each layer, we can extract salient features by embedding the test data onto the obtained projection. Then, the one-Nearest-Neighbor (1NN) classifier is used for classification. The classification result of each method is averaged based on 10 random splits of training/test samples. From the results, we can find that that our fMLLC-Net and nMLLC-Net outperform the regular FLLRR and LatLRR methods by obtaining higher accuracies. One can also find that the best records of our methods are obtained in the 2-nd and 3-rd layers, which implies that the features obtained in

TABLE II
AVERAGED FACE RECOGNITION RATES ON UMIST.

Evaluated Methods	**2 train**	**3 train**	**4 train**	**5 train**
	Acc. *(%)*	**Acc. *(%)***	**Acc. *(%)***	**Acc. *(%)***
FLLRR	67.24	77.83	80.94	85.96
fMLLC-Net (*2 layers*)	68.51	78.21	81.13	86.00
fMLLC-Net (*3 layers*)	**69.78**	**78.52**	**81.95**	**86.81**
fMLLC-Net (*4 layers*)	66.95	77.28	80.61	85.81
LatLRR	69.54	77.84	83.76	88.44
nMLLC-Net (*2 layers*)	69.96	**79.05**	**84.91**	**89.99**
nMLLC-Net (*3 layers*)	**71.67**	77.41	84.51	88.57
nMLLC-Net (*4 layers*)	66.05	76.26	82.05	87.39

TABLE III
AVERAGED FACE RECOGNITION RATES ON MIT CBCL.

Evaluated Methods	**2 train**	**3 train**	**4 train**	**5 train**
	Acc. *(%)*	**Acc. *(%)***	**Acc. *(%)***	**Acc. *(%)***
FLLRR	66.75	76.63	84.78	87.00
fMLLC-Net (*2 layers*)	67.78	77.68	85.84	87.05
fMLLC-Net (*3 layers*)	**67.83**	**77.69**	**85.88**	**87.96**
fMLLC-Net (*4 layers*)	67.51	77.68	84.17	87.51
LatLRR	71.45	76.53	82.31	89.68
nMLLC-Net (*2 layers*)	71.68	**77.78**	**83.61**	**90.94**
nMLLC-Net (*3 layers*)	**73.51**	77.01	82.76	90.68
nMLLC-Net (*4 layers*)	70.40	75.75	80.71	86.02

TABLE IV.
NUMERICAL CLUSTERING EVALUATION RESULTS ON THE UMIST FACE DATABASE.

Method	**Clustering Accuracy (%)**				**Clustering F-Score (%)**			
	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	72.69±21.21	64..7±15.57	48.04±4.52	50.89±12.03
rLatLRR	86.25±15.06	79.44±9.94	69.67±7.40	67.51±6.56	78.60±20.58	66.18±10.68	51.50±7.42	53.67±6.71
SA-LatLRR	84.57±14.30	78.35±12.26	72.08±7.40	67.52±9.29	75.94±20.82	69.09±13.73	59.27±10.07	54.81±10.93
rLRR	82.35±14.41	78.52±6.48	70.67±9.49	66.57±7.52	77.69±14.61	68.49±6.69	59.44±8.59	52.24±3.84
PLrSC	87.89±8.74	79.89±6.77	71.11±10.27	67.79±7.38	79.12±14.83	70.48±8.64	60.54±7.89	51.77±4.71
FLLRR	84.16±17.37	79.25±8.98	71.56±8.18	66.29±7.70	78.58±21.23	69.24±10.59	60.67±8.0	54.59±8.72
fMLLC-Net (*2 layers*)	88.17±16.58	80.42±10.61	73.83±8.20	69.83±7.96	82.93±21.95	71.0±12.58	62.30±8.96	57.66±9.39
fMLLC-Net (*3 layers*)	88.83±16.64	**81.08±10.66**	**76.44±8.19**	**70.88±8.62**	83.94±22.10	**71.58±12.89**	**63.40±8.73**	**58.16±9.78**
fMLLC-Net (*4 layers*)	**90.16±16.49**	77.92±11.14	71.50±8.70	66.67±8.30	**85.86±21.87**	66.36±12.0	59.18±9.05	55.01±8.63
fMLLC-Net (*5 layers*)	88.83±16.05	75.0±13.47	68.06±8.98	65.08±7.40	83.83±21.24	63.61±14.85	54.85±9.94	53.43±8.51
LatLRR	81.50±9.73	77.25±15.02	69.17±10.77	67.00±7.58	80.61±3.73	70.06±16.10	62.09±12.55	54.82±7.87
nMLLC-Net (*2 layers*)	82.52±9.14	79.52±12.41	73.33±7.54	67.38±9.98	81.27±2.01	71.82±14.34	64.16±9.55	56.17±10.38
nMLLC-Net (*3 layers*)	85.05±12.35	**80.75±11.74**	**76.59±12.70**	**71.37±6.07**	81.40±16.11	**73.67±13.84**	**65.18±12.98**	**59.32±5.51**
nMLLC-Net (*4 layers*)	**94.00±7.75**	76.25±12.20	70.16±10.33	65.00±3.46	**89.28±11.95**	66.67±12.17	58.73±12.81	52.17±5.68
nMLLC-Net (*5 layers*)	90.20±7.09	76.71±15.48	68.83±14.03	64.63±6.48	85.02±12.23	65.61±17.53	56.27±15.15	53.38±7.13

TABLE V.
NUMERICAL CLUSTERING EVALUATION RESULTS ON THE USPS HANDWRITTEN DATABASE.

Method	Clustering Accuracy (%)				Clustering F-Score (%)			
	K=2	K=4	K=6	K=8	K=2	K=4	K=6	K=8
LRR	76.21±7.82	63.25±11.65	56.59±11.90	50.25±3.18	70.92±12.33	53.00±14.16	42.96±10.57	35.37±3.05
rLatLRR	82.52±10.29	69.85±14.92	60.67±5.92	54.62±7.02	74.48±13.85	54.98±17.02	46.43±4.11	38.43±5.54
SA-LatLRR	81.51±12.48	68.54±8.35	59.67±5.82	54.75±5.03	75.05±9.91	53.92±7.21	47.42±3.08	36.16±3.35
rLRR	78.36±10.47	66.48.00±7.51	58.47±5.29	54.57±4.16	74.27±5.71	53.12±6.24	44.47±2.12	39.57±8.71
PLrSC	81.89±8.76	66.79±8.59	59.47±6.18	52.84±7.16	73.17±6.47	55.58±7.84	42.55±6.84	38.27±5.65
FLLRR	82.83±15.35	67.67±8.37	60.94±8.14	54.67±4.74	74.61±17.84	54.18±8.30	46.57±7.85	39.25±4.34
fMLLC-Net (*2-nd layer*)	83.17±15.12	70.25±10.01	62.78±6.84	54.75±4.60	74.85±17.67	55.94±11.23	47.65±6.26	39.13±4.67
fMLLC-Net (*3-rd layer*)	84.33±13.44	**71.58±11.35**	**63.28±8.17**	**55.96±4.53**	75.43±16.75	**57.93±13.42**	**47.78±7.65**	**39.33±4.58**
fMLLC-Net (*4-th layer*)	**84.83±13.16**	70.08±9.67	61.94±9.15	53.25±5.34	**76.76±17.21**	57.28±10.24	46.92±9.07	37.54±5.30
fMLLC-Net (5-*th layer*)	79.67±15.02	69.58±10.49	58.89±6.50	51.46±4.86	73.13±17.10	55.95±9.48	42.46±6.42	35.23±4.98
LatLRR	80.11±16.67	65.75±7.98	56.33±8.88	50.38±6.48	70.89±21.39	52.09±10.88	41.87±6.78	34.53±7.66
nMLLC-Net (*2-nd layer*)	81.24±14.10	67.05±8.27	57.0±11.84	51.50±4.56	71.34±20.22	52.83±7.21	42.22±11.80	36.12±5.73
nMLLC-Net (*3-rd layer*)	**84.56±15.24**	**69.54±8.00**	**57.67±8.16**	**52.25±5.39**	**75.00±20.29**	**54.16±8.83**	**43.60±7.71**	**36.68±4.66**
nMLLC-Net (4-*th layer*)	81.17±14.23	66.40±8.27	57.33±9.43	51.63±4.17	70.36±13.85	51.63±7.21	42.03±8.35	35.37±4.22
nMLLC-Net (5-*th layer*)	80.52±13.12	65.52±6.58	56.18±6.28	47.38±6.73	69.84±15.77	50.77±8.31	41.00±6.87	33.18±6.62

TABLE VI.
NUMERICAL CLUSTERING EVALUATION RESULTS ON THE FASHION MINIST DATABASE.

Method	Clustering Accuracy (%)				Clustering F-Score (%)			
	K=2	K=4	K=6	K=8	K=2	K=4	K=6	K=8
LRR	85.21±4.37	62.27±9.11	51.83±1.99	45.75±3.24	80.84±5.52	59.94±12.11	43.76±3.52	36.41±2.64
rLatLRR	89.55±13.83	65.50±9.01	63.33±8.47	56.75±4.83	84.58±18.58	62.16±7.85	51.27±6.67	47.37±3.06
SA-LatLRR	91.07±11.74	68.74±4.28	63.54±11.91	56.12±6.41	85.81±16.06	63.50±3.35	54.64±7.66	46.57±5.60
rLRR	89.48±14.49	67.48±8.74	62.17±7.05	56.87±5.47	86.52±13.48	62.48±10.04	54.43±3.19	47.78±1.88
PLrSC	91.42±5.39	72.06±6.47	64.43±8.27	57.61±6.75	86.48±10.19	63.81±8.47	54.09±5.18	47.58±4.29
FLLRR	90.52±10.60	71.58±10.31	62.67±9.51	56.33±5.78	86.25±14.15	63.55±9.40	54.36±8.68	47.74±6.44
fMLLC-Net (*2-nd layer*)	91.67±10.69	73.50±11.06	64.58±9.92	58.38±4.87	86.60±14.37	63.93±10.48	54.55±9.33	48.11±5.51
fMLLC-Net (*3-rd layer*)	92.67±9.07	**73.75±11.51**	**64.61±9.24**	**59.17±5.84**	87.18±13.42	**64.57±10.70**	**55.38±8.95**	**48.85±5.64**
fMLLC-Net (*4-th layer*)	**93.17±8.76**	72.75±11.51	64.67±10.03	57.87±5.61	**87.99±12.98**	63.37±10.56	55.38±8.31	47.16±5.36
fMLLC-Net (5-*th layer*)	92.33±9.54	70.66±11.91	60.78±8.70	55.96±5.48	86.81±14.37	62.01±11.33	51.90±8.41	45.51±6.53
LatLRR	90.17±13.03	70.25±12.91	65.66±8.39	59.83±7.68	85.94±14.47	63.24±9.78	57.05±8.86	49.51±7.28
nMLLC-Net (*2-nd layer*)	92.17±10.64	71.31±11.43	67.00±8.31	61.52±7.37	87.10±15.40	65.62±8.76	57.19±8.12	50.17±6.65
nMLLC-Net (*3-rd layer*)	**93.33±8.54**	**73.03±11.58**	**67.60±9.16**	**61.56±6.68**	**88.28±13.87**	**66.67±9.83**	**57.70±9.56**	**50.99±7.16**
nMLLC-Net (4-*th layer*)	89.33±14.29	73.00±12.27	66.06±9.15	59.58±6.25	85.87±15.29	66.14±12.22	55.50±7.13	49.15±6.87
nMLLC-Net (5-*th layer*)	87.33±15.52	72.11±12.39	63.50±8.18	57.13±5.79	83.07±16.39	65.87±11.19	53.40±7.43	42.21±5.11

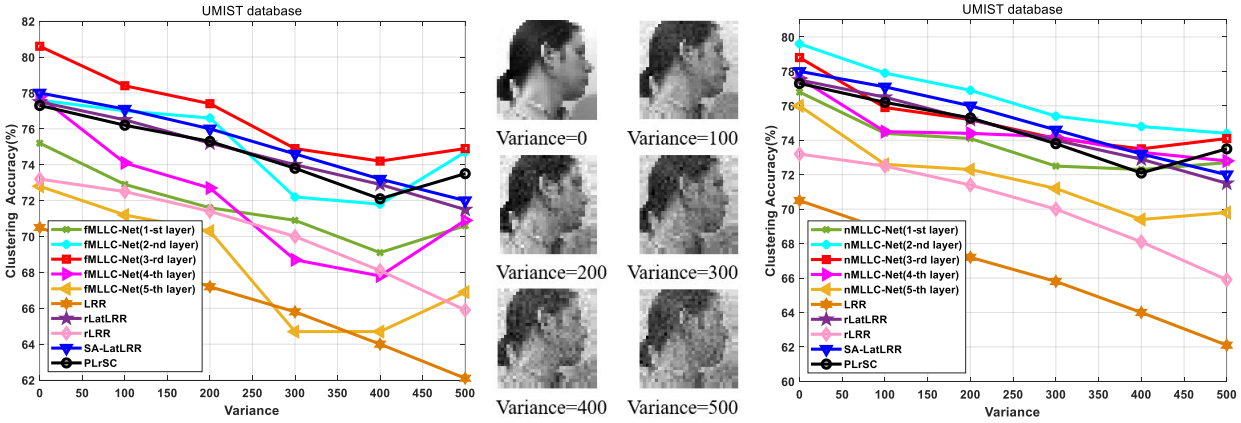
the 2-nd and 3-rd layers may contain more useful discriminative information. Once again, increasing the number of layers to a higher level cannot enhance the classification results, which may even decrease the results to some extent.

C. Quantitative Clustering Evaluations

In this section, we compare our algorithms and other methods for clustering images. Three datasets, i.e., UMIST face, USPS handwritten and Fashion MINIST are applied for the clustering evaluations. The USPS digit database has 9298 handwritten digits ('0'-'9') of 16×16 pixels. Fashion-MNIST database has 10 classes and 70000 unique products, which comes from different gender groups, i.e., men, women, kids and neutral. Every fashion product has a set of image shot by professional photographers, demonstrating different aspects of the product, i.e. front and back looks, details, looks with model and in an outfit. In this study, we use 1000 samples from each class, i.e., 10000 samples in total. The clustering process is described as follows. For each dataset, we use all images for learning the coefficients matrix Z^*. Then, to evaluate the clustering performance, we following common procedures and use the coefficient matrix Z^* of each algorithm to define the edge weights of an undirected graph, i.e., $W = \left(\left|Z^*\right| + \left|Z^{*T}\right|\right)/2$, and then we use the Normalized Cuts (NCut) [27] method to produce the clustering result. For PLrSC and our MLLC-Net, we use the coefficient matrix Z_M^* obtained in the final layer to define the edge weights of graph as $W = \left(\left|Z_M^*\right| + \left|Z_M^{*T}\right|\right)/2$. Specifically, for each fixed number K of clusters, we choose K categories from the dataset randomly and use the K categories to form the data matrix for representation learning. In each setting, the results are averaged based on 30 random initialization for the NCut algorithm.

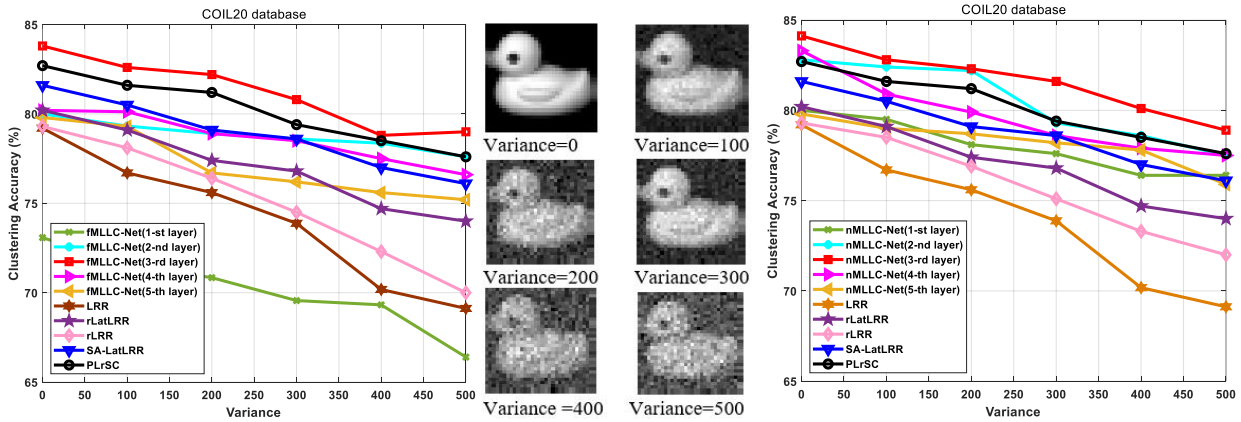
In this study, the clustering accuracy (AC) and F-measure are used as the quantitative evaluation metrics. The values of AC and F-measure based on the evaluated databases are described in Tables IV-VI, respectively. From the results, we can find that: (1) the clustering accuracy of each algorithm goes down when

the number of categories increases, which is because clustering more data is always difficult than clustering less data; (2) Our fMLLC-Net and nMLLC-Net deliver higher values of AC than other competitors, especially compared to regular FLLRR and LatLRR in the one layer case, which means that our MLLC-Net can learn more effective feature representation by mining deep information, compared with the single-layer models.



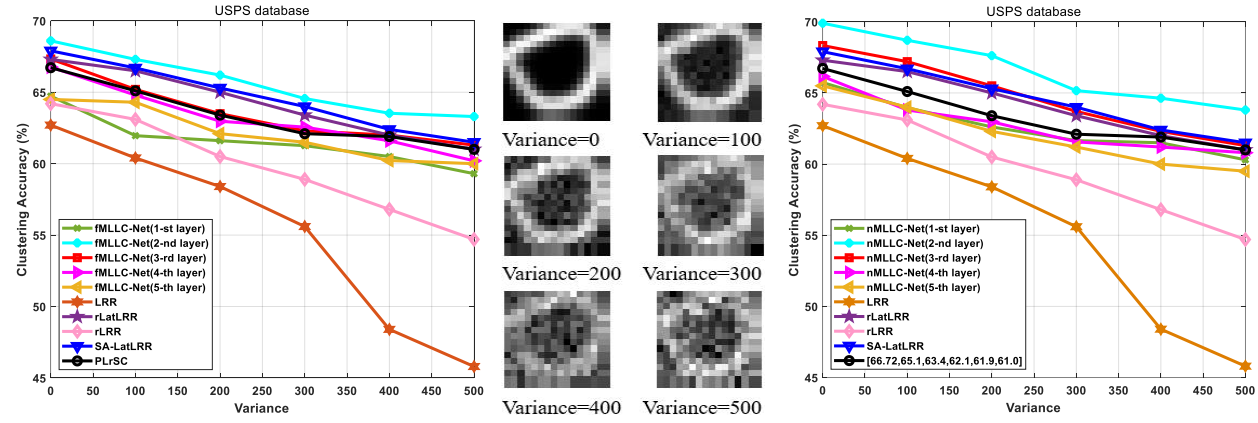
(a) fMLLC-Net (b) Examples (c) nMLLC-Net

Fig. 9: Clustering performance vs. varying variance on the UMIST face database.



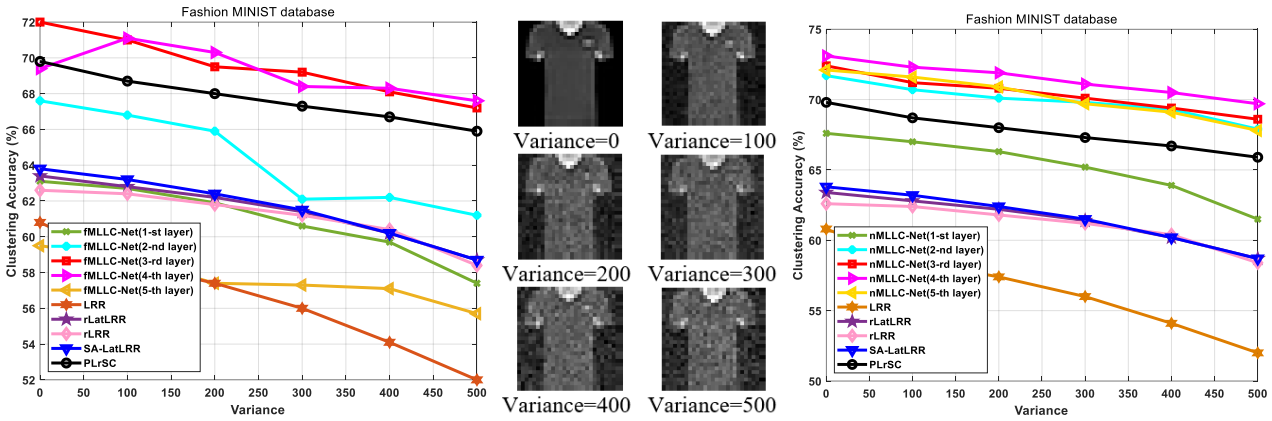
(a) fMLLC-Net (b) Examples (c) nMLLC-Net

Fig. 10: Clustering performance vs. varying variance on the COIL20 object databases.



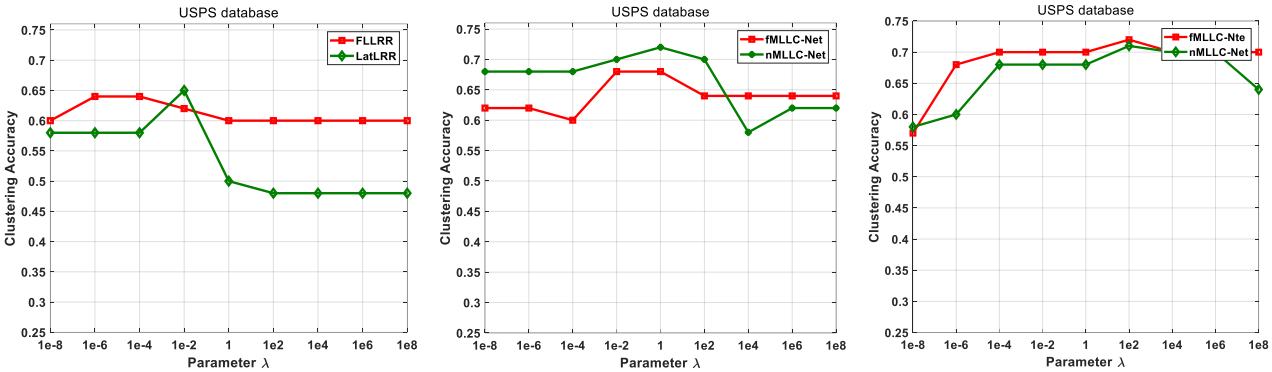
(a) fMLLC-Net (b) Examples (c) nMLLC-Net

Fig. 11: Clustering performance vs. varying variance on the USPS handwritten database.



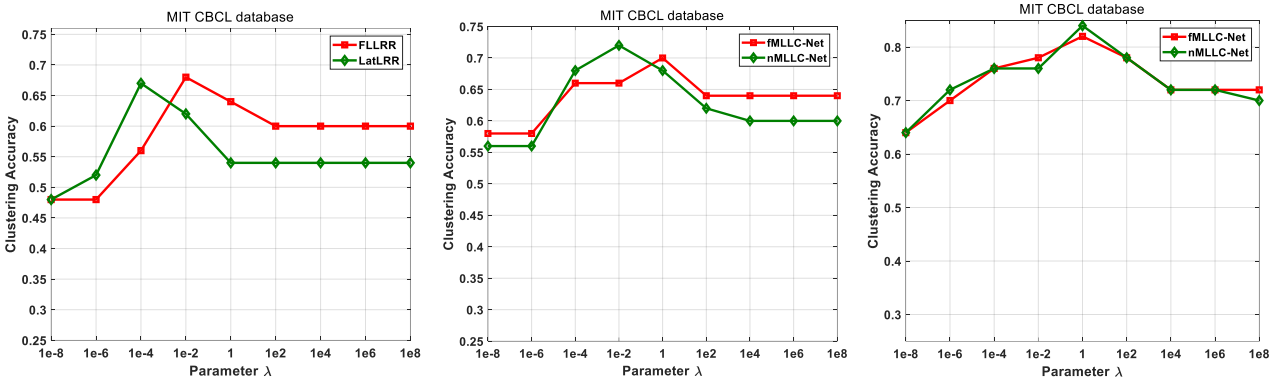
(a) fMLLC-Net (b) Examples (c) nMLLC-Net

Fig. 12: Clustering performance vs. varying variance on the fashion MNIST image database.



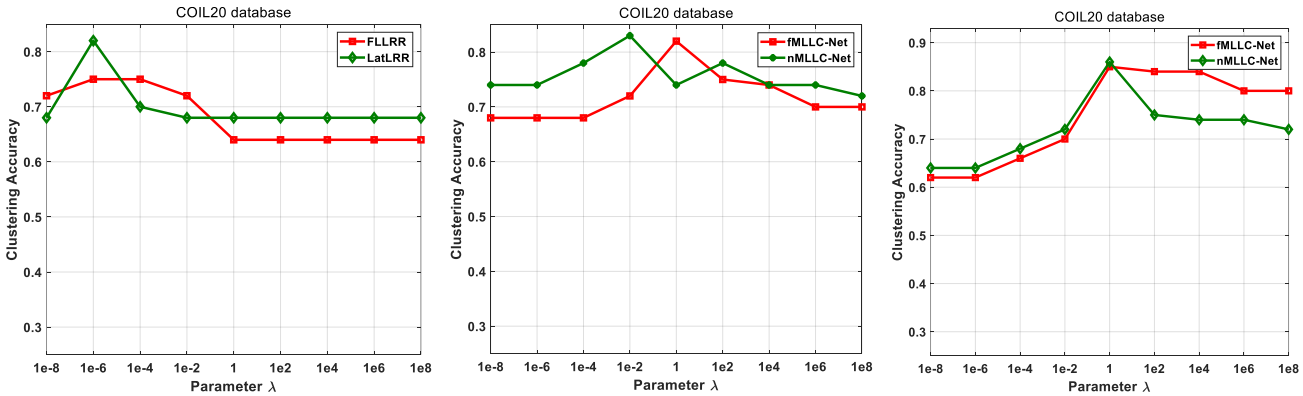
(a) The first layer (b) The second layer (c) The third layer

Fig.13: Parameter sensitivity analysis of MLLC-Net on the USPS digit database.



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

Fig.14: Parameter sensitivity analysis of our MLLC-Net on the MIT CBCL face database.



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

Fig.15: Parameter sensitivity analysis of MLLC-Net on the COIL20 object database.

D. Noisy Image Clustering Against Corruptions

In this study, we mainly investigate the robustness properties of our fMLLC-Net and nMLLC-Net against the noisy case that the images are corrupted. Four image databases, including USPS, MIT CBCL, COIL20 and Fashion MINIST, are used. COIL20 object database contains 1440 images of 20 subjects. To corrupt data, random Gaussian noise with different variance is added to examine the robustness with the *var*=100, 200, …, 500. For each setting, we average the clustering result over 30 random initialization for the NCut clustering method and we set the number K of clusters as 5. The noisy clustering accuracy and some examples of noisy images are shown in Figs.9-12, where the image clustering results of five layers for our fMLLC-Net and nMLLC-Net are described. From the results, we easily find that: (1) generally speaking, the clustering accuracy of each layer and each algorithm goes down with the increasing level of noise, which is not difficult to understand, since clustering data of high noise level is usually difficult than clustering data of low noise level; (2) the best records are usually obtained in t*he* 2-nd layer and 3-rd layer, compared with the other cases, including FLLRR and LatLRR, which means that the learnt deep features contains more useful information. But we once again find that increasing the number of layers cannot improve the representation learning ability further for clustering.

E. Investigation of Parameters

In this part, we analyze the parameter sensitivity of fMLLC-Net and nMLLC-Net. Since they only has one parameter λ, we can easily select the most important ones from the candidate set $\{10^{-8},10^{-6},...,10^{6},10^{8}\}$. Three databases, i.e., USPS, MIT CBCL and COIL20 are used. For each database, we set the number K of clusters to 5 and describe the results of the model parameters used in the first three layers in Figs.13-15. For each layer, we select the best parameters, and then we fix it to calculate deeper features of the next layer. From the reults, we can find that: (1) the value of λ usually becomes a lager one with the increasing number of the layers, which is easy to understand, because the parameter λ depends on the noise level of the datasets and a large value should be used when the level of noise is lower, otherwise a smaller value should be used instead [5]. It is worth noting that our proposed deep low-rank coding model recovers the subspaces progressively, and more importantly the learned susbapces will become clean layer by layer, as can be observed from the above viusal and quantitative evalutions. Note that similar observations can be found from the other datasets, but due to the page limitation, the results will be not presented.

VII. Conclusion

We discussed the deep latent low-rank representation problem, and technically proposed a Multilayer Latent Low-rank Coding Network termed MLLC-Net to uncover deep hidden features and the deep clustering structures embedded in the latent subspace. To improve the representation learning abilities, MLLC-Net discovers the subspaces by refining the principal and salient features from previous layers progressively and then fusing the subspaces. Specifically, our network takes shallow features from the previous layer as the inputs of subsequent layers, and then recovers hierarchical information and deeper features by congregating the projective subspaces and representation subspaces in each layer, which can recover features and correct errors jointly. Moreover, our network can be extended to most existing latent low-rank coding models and extends them to the multilayer scenario for learning powerful deep information.

We have examined the effectiveness of our network on the widely-used image databases by extensive visual image analysis and quantitative evaluations. The quantitative recognition and clustering results over both original and corrupted images demonstrate that superior performance has been obtained by our methods. The visual image analysis also demonstrated the validity of our methods for image representation and recovery. In future, more effective deep latent low-rank coding strategies will be explored. Besides, we will extend the proposed network to other related restoration based low-level vision tasks.

References

[1] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol.40, no.10, pp. 2331-2338, 2002.

[2] X. Wang, K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition*, vol.36, no.10, pp.2429- 2439, 2003.

[3] Z. Zhang, M. B. Zhao and T. Chow, "Binary- and Multi-Class Group Sparse Canonical Correlation Analysis for Feature Extraction and Classification," *IEEE Trans. Knowledge and Data Engineering*, vol.25, no.10, pp.2192-2205, 2013.

[4] J. Wright, Y. Ma, J. Mairal, "Sparse Representation for Computer Vision and Pattern Recognition," *Proceedings of the IEEE*, vol.98, no.6, pp.1031-1044, 2010.

[5] G. Liu, Z. Lin, and S. C. Yan, "Robust Recovery of Subspace Structures by Low-Rank Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.35, no.1, pp.171-184, 2013.

[6] Z. Zhang, C L. Liu, and M. B. Zhao, "A sparse projection and low-rank recovery framework for handwriting representation and salient stroke feature extraction," *ACM TIST*, vol.6, no.1, 2015.

[7] E. J. Candès, X. Li and Y. Ma, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol.58, no.3, pp.1-11, 2011.

[8] B. K. Bao, G. C. Liu, and C. Xu, "Inductive robust principal component analysis," *IEEE Trans.on Image Proc*., vol.21, no.8, pp.3794-3800, 2012.

[9] G. C. Liu, and S. C. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," *Computer Vision (ICCV), IEEE Int. Conf. on. IEEE*, Barcelona, Spain, pp.1615-1622, 2011.

[10] Z. Zhang, S. C. Yan, and M. B. Zhao, "Similarity preserving low-rank representation for enhanced data representation and effective subspace learning," *Neural Networks*, vol.53, pp. 81-94, 2014.

[11] Z. Zhang, F. Li, M. Zhao, L. Zhang, S. Yan, "Joint low-rank and sparse principal feature coding for enhanced robust representation and visual classification," *IEEE Transactions on Image Processing*, vol.25, no.6, pp.2429-2443, 2016.

[12] S.Yu and Y. Wu, "Subspace clustering based on latent low rank representation with Frobenius norm minimization," *Neurocomputing*, vol.275, pp.2479-2489, 2018.

[13] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Tech. Tep. UILU-ENG-09-2215*, 2009.

[14] H. Zhang, Z. C. Lin, C. Zhang and J. Gao, "Robust latent low rank representation for subspace clustering," *Neurocomputing*, vol. 145, pp. 369-373, 2014.

[15] Z. Zhang, S. C. Yan, and M. B. Zhao, "Similarity preserving low-rank representation for enhanced data representation and effective subspace learning," *Neural Networks*, vol.53, pp. 81-94, 2014.

[16] Z. Zhang, W. Jiang, J. Qin, L. Zhang, F. Li, M. Zhang, S. C. Yan, "Jointly Learning Structured Analysis Discriminative Dictionary and Analysis Multiclass Classifier," *IEEE Trans. on Neural Networks and Learning Systems*, vol.29, iss.8, pp.3798-3814, August 2018.

[17] C. P. Hou, F. P. Nie, X. L. Li, D.Y. Yi, and Yi Wu, "Joint Embedding Learning and Sparse Regression: A Framework for Unsupervised Feature Selection," *IEEE Trans. on Cybernetics*, vol.44, no.6, pp.793- 804, 2014.

[18] L. Ma, C.H. Wang, B.H. Xiao, W. Zhou, "Sparse Representation for Face Recognition based on Discriminative Low-Rank Dictionary Learning," *In:*

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, pp. 2586-2593, 2012.
[19] L.Y. Li, S. Li Y. Fu, "Learning low-rank and discriminative dictionary for image classification," *Image and Vision Computing*, vol.32, no.10, pp.814-823, 2014.
[20] Y. Zhang, Z. L. Jiang, and L. Davis, "Learning structured low-rank representations for image classification," *In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, 2013.
[21] J. Chen and Z. Yi, "Sparse representation for face recognition by discriminative low-rank matrix recovery," *J. Vis. Commun. Image R.*, vol.25, no.5, pp.763-773, 2014.
[22] J. Xie, J. Yang, J. Qian, Y. Tai, H. Zhang, "Robust Nuclear Norm-Based Matrix Regression With Applications to Robust Face Recognition," *IEEE Trans. Image Processing*, vol.26, no.5, pp.2286-2295, 2017.
[23] Z. Xue, J. Du, D. Du, and S. Lyu, "Deep low-rank subspace ensemble for multi-view clustering," *Information Sciences*, vol.482, pp.210-227, 2019.
[24] Z. Li, and J. Tang. "Weakly-supervised deep nonnegative low-rank model for social image tag refinement and assignment," *In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[25] J. Zhao, Y. Lv, Z. Zhou, and F. Cao, "A novel deep learning algorithm for incomplete face recognition: Low-rank-recovery network," *Neural Networks*, vol.94, pp.115-124, 2017.
[26] J. Cai, E.J. Candès and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol.20, no.4, pp.1956-1982. 2010.
[27] J. Shi and, J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, no.8, pp.888-905, 2000.
[28] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-20)," Technical Report CUCS-005-96, February 1996.
[29] L. Wang, B. Wang, Z. Zhang, Q. Yue, L. Fu, G. Liu, M. Wang, "Robust Auto-weighted Projective Low-Rank and Sparse Recovery for Visual Representation," *Neural Networks*, vol.117, pp.201-215, Sep 2019.
[30] Z. Zhang, M. Zhao, F. Li, L. Zhang, S. Yan, "Robust Alternating Low-Rank Representation by Joint Lp- and L2,p-norm Minimization," *Neural Networks*, vol.96, pp.55-70, Dec 2017.
[31] Z. Zhang, J.H. Ren,, S. Li, R. Hong, Z. Zha, M. Wang, "Robust subspace discovery by block-diagonal adaptive locality-constrained representation," *In Proceedings of ACM International Conference on Multimedia*, pp. 1569-1577, 2019.
[32] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on pattern analysis and machine intelligence*, no.16, vol.5, pp.550-554, 1994.
[33] Z. Zhang, S. Yan and M. Zhao, "Similarity Preserving Low-Rank Representation for Enhanced Data Representation and Effective Subspace Learning," *Neural Networks*, vol.53, pp.81-94, 2014.
[34] Z. Zhang, L. Wang, S. Li, Y. Wang, Z. Zhang, Z. J. Zha and M. Wang, "Adaptive Structure-constrained Robust Latent Low-Rank Coding for Image Recovery," In: *Proceedings of the 19th IEEE International Conference on Data Mining*, Beijing, China, Oct 2019.
[35] L. Wang, Z. Zhang, S. Li, G. Liu, C. Hou, J. Qin, "Similarity-Adaptive Latent Low-Rank Representation for Robust Data Representation," In: *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, Nanjing, China, pp.71-84, 2018.
[36] Z. Tang, Z. Zhang, X. Ma, J. Qin, and M. Zhao, "Robust Neighborhood Preserving Low-Rank Sparse CNN Features for Classification," In: *Proceedings of the 19th Pacific-Rim Conference on Multimedia*, Hefei, China, pp.357-369, July 2018.
[37] J. Ren, Z. Zhang, S. Li, G. Liu, M. Wang, and S. Yan, "Robust Projective Low-Rank and Sparse Representation by Robust Dictionary Learning," In: *Proceedings of the International Conference on Pattern Recognition*, Beijing, China, April 2018.
[38] L. Wang, Z. Zhang, G. Liu, Q. Ye, J. Qin, M. Wang, "Robust Adaptive Low-Rank and Sparse Embedding for Feature Representation," In: *Proceedings of the 24th International Conference on Pattern Recognition*, Beijing, China, April 2018.
[39] D. Graham and N. Allinson. "Characterizing virtual eigensignatures for general purpose face recognition," in *Face Recognition: From Theory to Application (NATO ASI Series F, Computer and Systems Sciences)*, Springer, Berlin, Heidelberg, vol.163, pp.446-456, 1998.
[40] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643-660, 2001.
[41] T. Sim, S. Baker, M. Bsat, "The CMU pose, illuminlation, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.25, no.12, pp.1615-1618, 2003.
[42] B. Weyrauch, J. Huang, B. Heisele, V.Blanz, "Component-based Face Recognition with 3D Morphable Models," *In: Proc. 1st IEEE workshop on Face Processing in Video*, Washington, D.C, 2004.
[43] J. Li, H. Chang and J. Yang, "Sparse deep stacking network for image classification," *In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.* 2015.
[44] H. Xiao, K. Rasul, R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv:1708.07747v2*, 2017.
[45] J. Li, H. Liu, "Projective low-rank subspace clustering via learning deep encoder," *In Proceedings of International Joint Conferences on Artificial Intelligence* 2017.
[46] X. Chen, J. Li, Y. Song, F. Li, J. Chen, and K. Yang, "Low-Rank Tensor Completion for Image and Video Recovery via Capped Nuclear Norm," *IEEE Access*, vol.7 pp. 112142-112153,2019.
[47] F. Su, H. Y. Shang, J. Y. Wang, "Low-Rank Deep Convolutional Neural Network for Multitask Learning," *Computational Intelligence and Neuroscience*, 2019.
[48] Z. Ding and Y. Fu, "Deep Transfer Low-Rank Coding for Cross-Domain Learning," *IEEE transactions on neural networks and learning systems*, vol.30, no.6, pp.1768-1779, 2018.