Neural Networks 94 (2017) 115–124

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A novel deep learning algorithm for incomplete face recognition: Low-rank-recovery network

Jianwei Zhao, Yongbiao Lv, Zhenghua Zhou, Feilong Cao *

Department of Mathematics and Information Sciences, China Jiliang University, Hangzhou 310018, Zhejiang Province, PR China

ARTICLE INFO

Article history:
Received 13 October 2016
Received in revised form 18 June 2017
Accepted 27 June 2017
Available online 20 July 2017

Keywords:
Deep learning
Convolutional neural networks
Face recognition
Recovery of low-rank matrix
ADMM

ABSTRACT

There have been a lot of methods to address the recognition of complete face images. However, in real applications, the images to be recognized are usually incomplete, and it is more difficult to realize such a recognition. In this paper, a novel convolution neural network frame, named a low-rank-recovery network (LRRNet), is proposed to conquer the difficulty effectively inspired by matrix completion and deep learning techniques. The proposed LRRNet first recovers the incomplete face images via an approach of matrix completion with the truncated nuclear norm regularization solution, and then extracts some low-rank parts of the recovered images as the filters. With these filters, some important features are obtained by means of the binaryzation and histogram algorithms. Finally, these features are classified with the classical support vector machines (SVMs). The proposed LRRNet method has high face recognition rate for the heavily corrupted images, especially for the images in the large databases. The proposed LRRNet performs well and efficiently for the images with heavily corrupted, especially in the case of large databases. Extensive experiments on several benchmark databases demonstrate that the proposed LRRNet performs better than some other excellent robust face recognition methods.

1. Introduction

With the rapid development of computer technology, a great deal of face recognition approaches have been proposed. Generally, these methods can be classified into two categories. The first category is based on the theory of sparse representation (SR). Its basic idea is to train an over-complete dictionary from abundant face images firstly, and then to classify the test image with the coefficients of SR according to the dictionary. The SR-based methods do not need the process of explicit feature extraction. There are some face recognition methods based on SR, such as Deng, Hu, and Guo (2012), Jiang and Lai (2015), Ma, Wang, Xiao, and Zhou (2012), Wagner et al. (2012), Wang et al. (2014), Wright, Yang, Ganesh, Sastry, and Ma (2009), Yang, Zhang, Yang, and Zhang (2011), and Zhao, Cheung, Hu, and Wu (2016), Zhao, Glotin, Xie, Gao, and Wu (2012) and so on. The other category is based on feature extraction. One first extracts some efficient and robust features from each training image and then uses a classifier to distinguish them. The works, such as Ahonen, Hadid, and Pietikainen (2006), He, Yan, Hu, Niyogi, and Zhang (2005), Turk and Pentland (1991), Wiskott, Fellous, Kuiger, and Von Der Malsburg (1997), Yang, Zhang, Frangi, and Yang (2004), and Zhao, Krishnaswamy, Chellappa, Swets, and Weng (1998), are some representational approaches. However, in these methods, the features extraction is usually in handcraft or based on experience. Obviously, this empirical features extraction will undoubtedly affect the accuracy and robustness of the recognition results. In recent years, learning features from the data of interest is regarded as a plausible way to make up the shortcoming of hand-crafted features. A standing example of such methods is learning feature via deep neural networks (DNNs). The main merit to utilize DNNs is that the multiple levels of representation can be discovered with the assumption that higher-level features can represent more abstract semantics of the data. Those features learned from a deep network are expected to provide more invariance to intra-class variability. There is no doubt that DNNs are class of new and effective learning methods, and more and more attention is being drawn (Bruna & Mallat, 2013; Chan et al., 2015; Girshick, 2015; Hinton, Osindero, & Teh, 2006; Krizhevsky, Sutskever, & Hinton, 2012; Mobahi, Collobert, & Weston, 2009; Sifre & Mallat, 2013; Sun, Wang, & Tang, 2014).

1.1. Motivations

In real applications, face images are inevitably corrupted by all kinds of noise in the process of formation, transmission and storage, such as impulse noise caused by malfunctioning pixels in camera sensors, fault memory locations in hardware, and transmission in a noisy channel. Image noise can deteriorate the quality of

* Corresponding author.
E-mail addresses: zhaojw@amss.ac.cn (J. Zhao), 631480592@qq.com (Y. Lv), zzhzjw2003@163.com (Z. Zhou), feilongcao@gmail.com (F. Cao).

http://dx.doi.org/10.1016/j.neunet.2017.06.013
0893-6080/

image significantly. Then it makes the incomplete face recognition be a difficult task. In this paper, we will discuss face recognition for the corrupted image with some percentage of pixels missing randomly. As to the position of corrupted pixels, there are many methods to detect based on the color, statistical noise model, or the underlying image patch structures (Ji, Liu, Shen, & Xu, 2010).

Although the sparse representation based classification (SRC) method in Wright, Yang et al. (2009) has gained great success for frontal face recognition with varying expression and illumination, as well as occlusion and disguise, it needs the assumption that the training images are sufficient and uncorrupted. To overcome the limitations, Jiang and Lai (2015) proposed the sparse and dense hybrid representation (SDR) method that divides the training dictionary into a class-specific dictionary and a non-class-specific dictionary to make all the used information valid, so that it could deal with the case of corruption. These mentioned methods are all robust to face images with little corruption and occlusion. However, when the images are heavily corrupted, their performances are unsatisfactory. Moreover, although SDR can alleviate the deficiencies of SRC, the computational cost is very high when the data is large. In particular, as the size of data grows up, the training dictionary becomes larger, and the time cost is out of our tolerance.

This paper will provide a novel method to solve face recognition with corruption by constructing a class of new convolution neural networks (CNNs), i.e., low-rank-recovery network (LRRNet). The proposed method integrates matrix completion methods, a new artificial neural network, and support vector machines (SVMs). First, we define the corrupted images as images with some percentage of random pixels missing, the corruption parts of image is regarded as the missing parts of matrix, and the missing values are estimated from a small subset of known entries. Obviously, this completion is an ill-posed problem. Fortunately, face image has a low-rank or approximately low-rank structure, and many methods of matrix completion (MC), such as Cai, Candès, and Shen (2010), Candès and Recht (2009), Candès and Tao (2010), Cao, Cai, and Tan (2015), Haeffele, Young, and Vidal (2014), Hu, Zhang, Ye, Li, and He (2013), Jacob, Francis, Theodoros, and Jean-Philippe (2006), Lin, Chen, and Ma (2010), Tara Brian, Vikas, Ebru, and Bhuvana (2013), Wright, Ganesh, Rao, Peng, and Ma (2009), and Zaiwen, Wotao, and Yin (2012), can be used for reference. Here we will choose a completion method called the truncated nuclear norm regularization by alternating direction method of multipliers (TNNR-ADMM) used in Hu et al. (2013). The reasons for this choice are as follows: Different from the nuclear norm based approaches that minimizes the summation of all the singular values, TNNR-ADMM only minimizes the smallest $\min(m, n) - r$ singular values since the rank r of a matrix only corresponds to the first r nonzero singular values, which can guarantee to get a more accurate and robust approximation to the rank function (Cao, Chen, Ye, Zhao, & Zhou, 2016; Hong, Wei, Hu, Cai, & He, 2016; Hu et al., 2015; Liu, Lai, Zhou, Kuang, & Jin, 2016; Wang & Su, 2014). Secondly, to extract high-level features for classification we will employ the idea of deep learning. Different with some popular deep learning methods, such as wavelet scattering networks (ScatNet) (Bruna & Mallat, 2013; Sifre & Mallat, 2013) and principal components analysis networks (PCANet), Chan et al. (2015), a new artificial neural network (named as LRRNet) will be constructed by using a low-rank matrix factorization method (Lin et al., 2010), which can be used to extract the low-rank part as filters and get rid of the sparse noise, and thus the aim to improve recognition effect is achieved.

1.2. Contributions

The proposed LRRNet provides a method to solve corrupted face recognition with good performance. The contributions of this paper can be summarized as follows.

- The proposed LRRNet recovers the corrupted parts via a matrix completion method with truncated nuclear norm regularization (TNNR) to get rid of the influence of the corruption. Therefore, more information about the corrupted parts can be extracted for the subsequent recognition, and the recognition rate for corrupted face images is improved due to the use of matrix completion. So the proposed LRRNet should be an effective approach and a new path to deal with the corrupted face recognition.
- The idea of artificial neural network is incorporated into the proposed LRRNet. The proposed LRRNet can extract higher-level features and improve the recognition rate. Different to the construction of filters in PCANet (Chan et al., 2015) with principal components analysis (PCA), the low-rank matrix factorization method is used to extract the low-rank part as filters and to get rid of the sparse noise well, which also provide a method for constructing good filter of convolution neural network.
- Extensive experiments on several benchmark databases demonstrate that the proposed LRRNet performs better than some other excellent robust face recognition methods. As shown in experiments, when the images contain more corruption, the recognition rate of some other methods, such as PCANet, SRC, and SDR, cut down quickly, but our LRRNet still keeps good performance. Furthermore, the training time of the proposed LRRNet method is less than some current methods, especially when the dataset is large.
- When the images are corrupted, the recognition accuracy of the proposed LRRNet method just decreases a little. Especially, when the images are heavily corrupted, the LRRNet method can still reach relative high recognition accuracy compared with some other state-of-art methods.

1.3. Organization

The remaining part of this paper is organized as follows. Section 2 proposes a novel artificial neural network for the face recognition of the heavily corrupted images based on the idea of matrix completion and deep learning. In Section 3, some comparison experiments of our proposed method with some state-of-art methods are carried out on some standard and widely used face image databases, and the illustrations of good performances of the proposed method are exhibited. At last the paper is finished with a conclusion in Section 4.

2. The proposed low-rank-recovery network

In this section, we will propose a novel CNN, named low-rank-recovery network (LRRNet), to deal with the face recognition in the case of corruption, which is mainly based on the idea of low-rank matrix recovery. The proposed LRRNet firstly recovers the heavily corrupted information of the original images based on matrix completion, then constructs the filters with some row-lank parts of the recovered image based on the idea of robust principal components analysis (RPCA). With these filters, some important features are obtained by means of the binaryzation and histogram algorithms. Finally, these features are classified with the classical support vector machines (SVMs). The main flow diagram of our proposed LRRNet is shown in Fig. 1. The next subsections will describe the main components of the network in detail.

2.1. Image recovery with a matrix completion

Let $\{I_i\}_{i=1}^N$ be the set of original face images with some corruption of the size of $m \times n$. In order to get a good recognition rate,

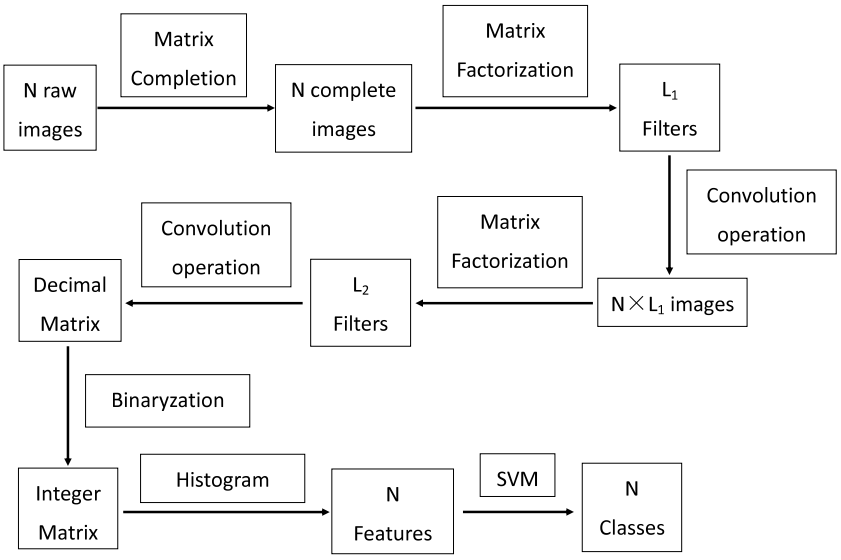


Fig. 1. The main flow diagram of our proposed LRRNet.

the proposed LRRNet method first recovers the corrupted parts via matrix completion method to get rid of the bad influence of the corruption. That is, the corrupted pixels are regarded as some missing elements and a matrix completion method is applied to recover them. If a patch of image is missing, like a blob, not just random pixels missing, we can also address this problem using matrix completion method with a less satisfactory performance. Now for each image, its recovery problem can be formulated as the following low-rank matrix approximation problem:

$$\min_X \ \operatorname{rank}(X) \quad \text{s.t. } P_\Omega(X) = P_\Omega(I), \tag{2.1}$$

where $P_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is the orthogonal projection operator onto the span of matrices vanishing outside of Ω as

$$(P_\Omega(I))_{ij} = \begin{cases} I_{ij}, & (i,j) \in \Omega; \\ 0, & (i,j) \notin \Omega, \end{cases}$$

and $\Omega \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ consists of indices of observed pixels in I which are not corrupted.

The rank minimization problem for low-rank matrix completion is NP-hard due to the nonconvexity and discontinuous nature of the rank function. Therefore, a widely used approach is to apply the nuclear norm, i.e., the sum of singular values of a matrix, as a convex surrogate of the nonconvex matrix rank function (Ma, Goldfarb, & Chen, 2011; Yonatan, Michael, Nathan, & Ullman, 2007). Some theoretical studies showed that the nuclear norm is the tightest convex lower bound of the rank function of matrices (Cai et al., 2010; Jonathan & Dimitri, 1992; Recht, Fazel, & Parrilo, 2010; Toh & Yun, 2010; Wright, Ganesh et al., 2009). However, one major shortcoming of the methods based on nuclear norm minimization is that all the singular values are simultaneously minimized, which results that the rank may not be well approximated in practice. In our proposed LRRNet method, we will apply a better method, TNNR-ADMM (Hu et al., 2013), to approximate the rank of a matrix via the TNN. In fact, for a matrix $X \in \mathbb{R}^{m\times n}$, the TNN, $\|X\|_r$, is defined as the sum of $(\min\{m, n\} - r)$ minimum singular values, i.e.,

$$\|X\|_r = \sum_{i=r+1}^{\min\{m,n\}} \sigma_i(X),$$

where σ_i is the ith largest singular value of X. Then the optimization problem (2.1) can be approximated with the following formula:

$$\min_X \ \|X\|_r \quad \text{s.t. } P_\Omega(X) = P_\Omega(I). \tag{2.2}$$

Since $\|X\|_r$ is nonconvex, it is not easy to solve (2.2) directly. But it can be rewritten as

$$\min_X \|X\|_* - \max_{AA^\top = I, BB^\top = I} \operatorname{Tr}(AXB^\top) \quad \text{s.t. } P_\Omega(X) = P_\Omega(I), \tag{2.3}$$

where $\|X\|_*$ is nuclear norm, $\operatorname{Tr}(\cdot)$ is the trace of a matrix, and $A \in \mathbb{R}^{r\times m}$, $B \in \mathbb{R}^{r\times n}$. Then a simple but efficient iterative scheme, TNNR-ADMM (Hu et al., 2013), is applied to approximate a local minimum of (2.3) based on ADMM.

The concrete process of image recovery via TNNR-ADMM is shown in Algorithm 2.1.

Algorithm 2.1. Image recovery via TNNR-ADMM.
Input: an original corrupted image I, the set Ω of indices of the observed pixels, the truncated rank r, and the tolerance ϵ.
Initialize: $X_1 = P_\Omega(I)$.
Repeat
Step 1. Given a matrix X_l, make a singular value decomposition on X_l as $[U_l, \Sigma_l, V_l] = SVD(X_l)$, where $U_l = (u_1, \ldots, u_m) \in \mathbb{R}^{m\times m}$, $V_l = (v_1, \ldots, v_n) \in \mathbb{R}^{n\times n}$. Take two matrices A_l and B_l as $A_l = (u_1, \ldots, u_r)^\top$, $B_l = (v_1, \ldots, v_r)^\top$.
Step 2.
Initialize: $Z_1 = X_l$, $W_1 = Z_1$, $Y_1 = Z_1$.
Repeat
$Z_{k+1} = \mathcal{D}_1(W_k - Y_k)$, where $\mathcal{D}_1$ is the singular value shrinkage operator; $W_{k+1} = Z_{k+1} + (A_l^\top B_l + Y_k)$; Fix values at observed pixels $W_{k+1} = P_{\Omega^c}(W_{k+1}) + P_\Omega(M)$; $Y_{k+1} = Y_k + Z_{k+1} - W_{k+1}$
until $\|Z_{k+1} - Z_k\|_F \le \epsilon$.
$X_{l+1} = Z_{k+1}$.
until $\|X_{l+1} - X_l\|_F \le \epsilon$.
Return the recovered matrix, denoted as I'.

Generally, a singular value shrinkage operator $\mathcal{D}_\tau : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is defined as follows: for a matrix $Q \in \mathbb{R}^{m\times n}$, let $Q = U\Sigma V^\top$, where $\Sigma = \operatorname{diag}(\{\sigma_i\}_{1\le i\le \min\{m,n\}})$, then

$$\mathcal{D}_\tau(Q) = U\mathcal{D}_\tau(\Sigma)V^\top,$$

where $\mathcal{D}_\tau(\Sigma) = \operatorname{diag}(\max\{\sigma_i - \tau, 0\})$. In our proposed LRRNet method, τ is taken to be 1 for saving time within an acceptable decrease in accuracy through many experiments. And with experience, the general range of truncated rank r is [5, 25]. After large numbers of tests, $r = 7$ is a suitable value in our method. The $P_{\Omega^c}(\cdot)$ is the projection operator on the complementary set of Ω,

and the $\|\cdot\|_F$ in Algorithm 2.1 denotes the Fubini norm of a matrix. Considering the computational cost, we set the tolerance ϵ to be 0.01.

Fig. 2 demonstrates the recovery results of Algorithm 2.1 on some face images, where the upper row is some original images without missing pixels, the middle row is the corresponding images with 50% pixels off randomly, and the bottom row is their recovered images with TNNR-ADMM algorithm.

2.2. Extract filters via low-rank parts

With the recovered image I' from the corrupted one via Algorithm 2.1, the remaining work is to classify them. In this subsection, we will propose an effective method to construct the filters for the convolution operation.

Most face recognition methods are trying to extract some efficient features to distinguish different classes fast and accurately, e.g. principal components analysis (PCA) method (Cavalcanti, Ren, & Pereira, 2013), linear discriminant analysis (LDA) technique (Lu, Zou, & Wang, 2012), independent component analysis (ICA) method (Secchi, Vantini, & Zanini, 2016), Gabor features (Yu, He, & Cao, 2010), wavelet sub-bands (Huang, Li, Shang, Wang, & Zhang, 2015), the Zernike moments method (Singh, Mittal, & Walia, 2011), local binary patterns (LBP) feature (Yu et al., 2014), and SIFT feature (Bicego, Lagorio, Grosso, & Tistarelli, 2006). These low-level features can be hand-crafted with great success for some specific data, but designing effective features for new data usually requires new domain knowledge as most hand-crafted features cannot be simply applied into new conditions. Therefore, it is significant to find another way of learning features from the data of interest. A representative example of such methods is DNNs (Hinton et al., 2006), which attracts significant attention recently. In fact, the main idea of deep learning is to find a hierarchical representation for the potential use of both combined low level and high level features in the classification layer. One successful application of deep learning in image classification is the use of CNN architecture that consists of several trainable stages and a supervised classifier. Each stage generally comprises of "two layers", i.e., a convolutional layer and a feature pooling layer (Goodfellow, Warde-Farley, Mirza, Courville, & Bengio, 2013; Kavukcuoglu et al., 2010; Kevin, Koray, Marc'Aurelio, & Yann, 2009; Krizhevsky et al., 2012). The filters in each convolution layer of those methods are learned with a variety of techniques, such as restricted Boltzmann machines (RBM), regularized auto-encoders, and stochastic gradient descent (SGD) method. As we know, learning a good network for classification mostly depends on parameter tuning and some ad hoc tricks. Recently, some variations of CNNs, e.g. wavelet scattering networks (ScatNet) (Bruna & Mallat, 2013; Sifre & Mallat, 2013) and PCANet (Chan et al., 2015), have been proposed for different tasks. The main difference is that their convolutional filters are prefixed, that is, they are simply wavelet operators or obtained with PCA. Therefore, they do not need learning at all, which saves computational cost. But different with ScatNet and PCANet, we here will adopt the low-rank matrix factorization method (Lin et al., 2010) to extract the low-rank part as filters in network, and thus a new artificial neural network named LRRNet is constructed.

Now we begin to show the concrete process of extracting some low-rank parts of images to learn multistage filters based on the idea of RPCA. For each recovered image $I'_i \in \mathbb{R}^{m\times n}$, we take all the patches of size $k_1 \times k_2$ from left to right and top to down, then we can obtain $q := (m-k_1+1)\times(n-k_2+1)$ patches. Let $p_{i,j}$ denote the jth vectorized patch, $j = 1, 2, \ldots, q$, then we get a matrix $P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,q}] \in \mathbb{R}^{k_1k_2\times q}$ for each image I'_i, $i = 1, 2, \ldots, N$. Subtract patch mean from each patch to make the differences of patches more easier to be detected, then obtain $\overline{P}_i = [\overline{p}_{i,1}, \overline{p}_{i,2}, \ldots, \overline{p}_{i,q}] \in \mathbb{R}^{k_1k_2\times q}$, where $\overline{p}_{i,j}$ is the mean-removed patch $j = 1, 2, \ldots, q$. With the same construction for all input images and putting them together, we obtain a matrix

$$P = [\overline{P}_1, \overline{P}_2, \ldots, \overline{P}_N] \in \mathbb{R}^{k_1k_2\times Nq}. \tag{2.4}$$

Assuming that the number of filters in i-layer is L_i, $i = 1, 2$, the remaining work is to give a method of constructing filters. In our proposed LRRNet, we will apply the robust principal components analysis via inexact augmented Lagrange multipliers (RPCA-IALM) method (Lin et al., 2010) to construct the filters. For the matrix P in (2.4), its application of low-rank matrix factorization can be described as

$$\min_{X,E} \|X\|_* + \lambda\|E\|_1 \quad \text{s.t. } P = X + E, \tag{2.5}$$

where X is a low-rank matrix which contains most information of P, E is a noise matrix that is sufficiently sparse, $\|\cdot\|_1$ denotes the sum of absolute values of all entries in a matrix, and λ is a positive weighting parameter.

For the convex optimization problem (2.5) which involves minimizing a combination of both the nuclear norm and ℓ_1-norm, we will apply the scalable and fast RPCA-IALM method to approximate its optimal solution. The RPCA-IALM method is based on the inexact augmented Lagrange multipliers (ALM), where the Lagrangian function is

$$\begin{aligned} L(X, E, Y) = {} & \|X\|_* + \lambda\|E\|_1 + \langle Y, P - A - E\rangle \\ & + \frac{\mu}{2}\|P - A - E\|_F^2. \end{aligned} \tag{2.6}$$

The IALM consists of the following iterations:

$$\begin{cases} X_{k+1} = \arg\min_X L(X, E_k, Y_k); \\ E_{k+1} = \arg\min_E L(X_{k+1}, E, Y_k); \\ Y_{k+1} = Y_k + \mu(P - X_{k+1} - E_{k+1}). \end{cases} \tag{2.7}$$

In RPCA-IALM method, the sub-problems $\arg\min_X L(X, E_k, Y_k)$ and

$$\arg\min_E L(X_{k+1}, E, Y_k)$$

are solved via the iterative thresholding approach. For example, for $\arg\min_X L(X, E_k, Y_k)$, let $(U_k, \Sigma_k, V_k) = SVD(P - E_k + \mu^{-1}Y_k)$, the iteration formula is

$$X_{k+1} = U_k S_{\mu^{-1}}[\Sigma_k]V_k^\top, \tag{2.8}$$

where $S_{\mu^{-1}}[\Sigma_k]$ is the extension of soft-thresholding (shrinkage) operator on a matrix via element-wise. Generally, a soft-thresholding (shrinkage) operator S_ε can be defined as

$$S_\varepsilon[x] = \begin{cases} x - \varepsilon, & \text{if } x > \varepsilon; \\ x + \varepsilon, & \text{if } x < -\varepsilon; \\ 0, & \text{if } -\varepsilon \le x \le \varepsilon, \end{cases} \tag{2.9}$$

where $x \in \mathbb{R}$ and $\varepsilon > 0$. On the other hand, for $\arg\min_E L(X_{k+1}, E, Y_k)$, the iteration formula is

$$E_{k+1} = S_{\lambda\mu^{-1}}[P - X_{k+1} + \mu^{-1}Y_k]. \tag{2.10}$$

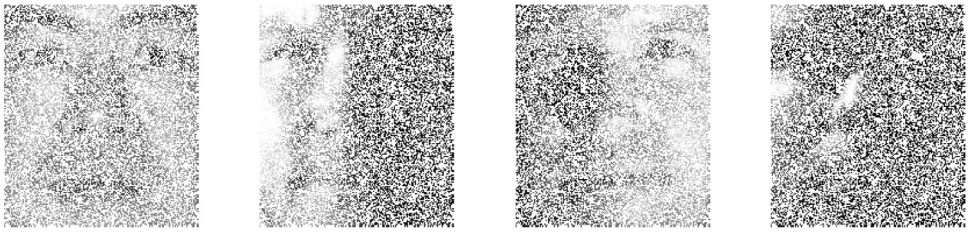
Put the iterations (2.8) and (2.10) into (2.7) and loop them until they are convergent, then we denote the outputs X_k, E_k, U_k, Σ_k and V_k as X^*, E^*, U^*, Σ^* and V^*, respectively.

For the obtained matrix X^* with the low rank r, it has the relation as

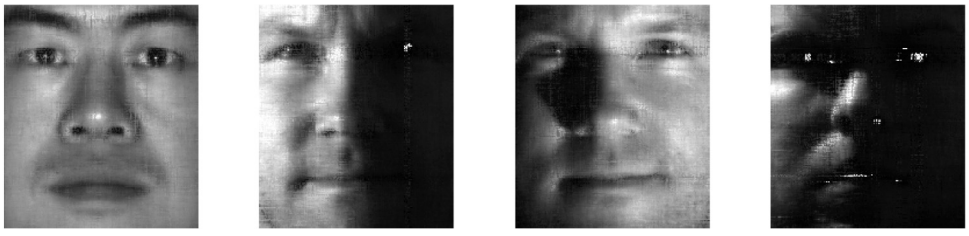
$$X^* = U^* S_{\mu^{-1}}[\Sigma^*]V^{*\top}, \tag{2.11}$$

where $U^* \in \mathbb{R}^{k_1k_2\times r}$, $\Sigma^* \in \mathbb{R}^{r\times r}$, $V^* \in \mathbb{R}^{Nq\times r}$. Take L_1 column vectors in U^* that are corresponding to the L_1 maximal singular values, and transform them into matrices with size of $k_1 \times k_2$, then we get L_1 filters $W_\ell^1 \in \mathbb{R}^{k_1\times k_2}$, $\ell = 1, 2, \ldots, L_1$ for the first stage of our proposed LRRNet. Since these filters are reshaped by

(a) Some original images.



(b) The corresponding images of the original images with 50% pixels are corrupted with noise randomly.



(c) The corresponding recovered images with TNNR-ADMM algorithm.

Fig. 2. Recovery results of TNNR-ADMM algorithm on some face images.

the singular value vectors, and the matrix X^* is low-rank, they can capture the "main information" of P.

For the ℓth filter W_ℓ^1, its output of the first stage is

$$I_i'^{\ell} = I_i' * W_\ell^1, \quad i = 1, 2, \ldots, N, \tag{2.12}$$

where $*$ denotes 2D convolution, and the boundary of I_i is zero-padded before convolving with W_ℓ^1 so as to make $I_i'^{\ell}$ having the same size of I_i'.

Suppose that the number of filters in stage II is L_2, then the similar process can be applied on the derived image $\{I_i'^{\ell}\}_{L_1}^N$ to get L_2 filters $W_\ell^2 \in \mathbb{R}^{k_1\times k_2}$. For each input $I_i'^{\ell}$ of the second stage, we will have L_2 outputs, each convolves $I_i'^{\ell}$ with W_ℓ^2 for $\ell = 1, 2, \ldots, L_2$:

$$O_i^{\ell} = \{I_i'^{\ell} * W_\ell^2\}_{\ell=1}^{L_2}. \tag{2.13}$$

Now the stages of convolution in our proposed LRRNet can be described in Algorithm 2.2 as follows.

Algorithm 2.2. Convolution operation of LRRNet.
Step 1. Given the recovered image $\{I_i'\}_{i=1}^N \subseteq \mathbb{R}^{m\times n}$, construct a matrix $P = [\overline{P}_1, \overline{P}_2, \ldots, \overline{P}_N] \in \mathbb{R}^{k_1k_2\times Nq}$ as in (2.4).
Step 2. Solve the optimization problem $\min_{X,E} \|X\|_* + \lambda\|E\|_1$ s.t. $P = X + E$ with the RPCA-via-Inexact-ALM method:
Input: λ, penalty parameter μ.
Initialize: $Y_0 = P/J(P)$; $E_0 = 0$; $k = 0$.
Loop: While not converged do: $(U_k, \Sigma_k, V_k) = SVD(P - E_k + \mu^{-1}Y_k)$; $X_{k+1} = U_k S_{\mu^{-1}}[\Sigma_k]V_k^\top$; $E_{k+1} = S_{\lambda\mu^{-1}}[P - X_{k+1} + \mu^{-1}Y_k]$; $Y_{k+1} = Y_k + \mu(P - X_{k+1} - E_{k+1})$.
End while.
Output: $X^* = X_k$, $E^* = E_k$, $U^* = U_k$,$V^* = V_k$, and $\Sigma^* = \Sigma_k$.
Step 3. Take L_1 column vectors in U^* that are corresponding to the L_1 maximal singular values, and transform them into matrices with size of $k_1 \times k_2$ to get L_1 filters $W_\ell^1 \in \mathbb{R}^{k_1\times k_2}$, $\ell = 1, 2, \ldots, L_1$.
Step 4. For the ℓth filter W_ℓ^1, output the derived images $I_i'^{\ell} = I_i' * W_\ell^1$, $i = 1, 2, \ldots, N$.
Step 5. Apply the similar process Step 1 to Step 3 on the derived image $\{I_i'^{\ell}\}_{i=L_1}^N$ to get L_2 filters $W_\ell^2 \in \mathbb{R}^{k_1\times k_2}$, $\ell = 1, 2, \ldots, L_2$.
Step 6. For the ℓth filter W_ℓ^2, its output of the second stage is $O_i^{\ell} = \{I_i'^{\ell} * W_\ell^2\}_{\ell=1}^{L_2}$.

The operator J in Algorithm 2.2 is defined as

$$J(I) = \max\{\|I\|_2, \lambda^{-1}\|I\|_\infty\},$$

where $\|\cdot\|_\infty$ is the maximum absolute value of the entries of a matrix.

2.3. Feature generation and output

For the L_2 real-valued outputs $\{I_i'^{\ell} * W_\ell^2\}_{\ell=1}^{L_2}$ in (2.13), we binarize them and get $\{H(I_i'^{\ell} * W_\ell^2)\}_{\ell=1}^{L_2}$, where $\mathcal{H}(\cdot)$ is an extension of Heaviside step function $H(\cdot)$ as follows from $\mathbb{R}$ to matrices:

$$H(x) = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

Around each pixel, we take the vector of L_2 binary bits as a decimal number by means of converting the L_2 outputs in O_i^{ℓ} back into a single integer-valued "image":

$$T_i^{\ell} = \sum_{\ell=1}^{L_2} 2^{\ell-1}\mathcal{H}(I_i'^{\ell} * W_\ell^2). \tag{2.15}$$

It is easy to see that each pixel is an integer in the range $[0, 2^{L_2} - 1]$.

For each of the L_1 feature images $\{T_i^{\ell}\}_{\ell=1}^{L_1}$, partition it into B blocks. For each block, compute the histogram of the decimal values with 2^{L_2} bins. Then we concatenate all the B histograms into

Table 1
The parameters' set in our proposed LRRNet and PCANet.

Parameters	Our LRRNet	PCANet
Patch size	5 × 5	5 × 5
Number of filters L_1	7	8
Number of filters L_2	7	8
Block size	5 × 5	5 × 5

a vector, denoted as $Bh(T_i^{\ell})$. Hence, for every original image I_i, it has L_1 vectors. We concentrate the L_1 vectors in one, then the feature of the input image I_i' is given as

$$f_i = [Bh(T_i^1), \ldots, Bh(T_i^{L_1})]^{\top} \in \mathbb{R}^{(2^{L_2})L_1 B}. \tag{2.16}$$

The data size of feature turns out to be very large when the number of filters we set becomes bigger, so we usually adjust the number of histogram bins to decrease the size at a cost of a little accuracy.

Different from PCANet, our proposed LRRNet method constructs filters with the low-rank part of the matrix which can get rid of the sparse noise to improve the effect. Furthermore, we can adjust the bins of histograms to make our features more suitable for large number of images.

3. Experimental results and analysis

In this section, we apply our proposed LRRNet for face recognition. Firstly, we test its performance on completed images without corruption. Then different performance on corrupted images on several datasets is analyzed. Lastly, the influence of parameters is discussed. All the programs are carried out in Matlab 7.11.0.584 environment running in Intel® Xeon® W3503 processor with the speed of 2.40 GHz, double kernel, memory of 8 GB.

3.1. Recognition comparison of our LRRNet with PCANet, SRC and SDR on standard MultiPIE and AR database without corruption

Now we display the performance of LRRNet and other excellent approaches like PCANet, SRC and SDR on standard MultiPIE and AR face images without corruption. MultiPIE database (Gross, Matthews, Cohn, Kanade, & Baker, 2010) is a common evaluation database for face recognition. It contains 337 subjects and more than 750,000 images are captured under various illumination conditions. In our experiment, we select 6936 images from 68 subjects equivalently, i.e. 102 images from each subject. Then we resize these images to 80 × 60. We divide these 6936 images into training set and testing set equivalently in a random way, i.e. 50% images as training samples and 50% images as testing samples. Of course, the images are converted into gray ones for speed. AR face database (Martinez, 1998) contains more than 4000 color face images corresponding to 126 subjects (70 male and 56 female). All the images are captured under different facial expressions and illumination conditions. In our experiment, 2600 images from 50 men and 50 women are chosen, i.e. 26 images for each person. These images are resized to 165 × 120. We also divide these 2600 images randomly into training set and testing set equivalently, i.e. 50% images as training samples and 50% images as testing samples, and convert the images into gray ones. The parameters of LRRNet and PCANet are listed in Table 1, where the parameters are the optimal parameters for LRRNet and PCANet via numbers of experiments, respectively. In this paper, the experimental results are taken the mean value of 5 times of experiments. Then the results of Experiment 3.1 are shown in Table 2.

Table 2 shows that our LRRNet, PCANet, and SDR both have excellent recognition for original images without corruption. Although our method has the best accuracy among approaches on AR database, SDR has the best one on MultiPIE database.

Table 2
Face recognition comparison of our LRRNet with some approaches like PCANet, SRC and SDR on MultiPIE and AR databases for original images without corruption.

Dataset	PCANet	SRC	SDR	Our LRRNet
MultiPIE	99.80%	80.54%	**99.88**%	99.54%
AR	97.92%	81.34%	96.46%	**98.31**%

Table 3
Recognition comparison of our LRRNet with PCANet, SDR, and SRC methods on MultiPIE dataset with different corrupted images.

Methods	10%	20%	30%	40%	50%
LRRNet	**99.05**%	**98.04**%	**96.63**%	**94.46**%	**87.37**%
PCANet	19.09%	10.67%	7.76%	5.85%	4.79%
SDR	59.49%	32.96%	6.23%	4.01%	3.69%
SRC	23.88%	15.22%	8.91%	6.92%	5.19%

Table 4
Recognition comparison of our LRRNet with PCANet, SDR, and SRC methods on MultiPIE dataset with different corrupted images denoised via matrix completion.

Methods	10%	20%	30%	40%	50%
LRRNet	99.05%	98.04%	96.63%	94.46%	87.37%
PCANet	99.42%	**98.96**%	**97.98**%	**95.85**%	**91.52**%
SDR	**99.79**%	59.51%	58.36%	57.06%	57.64%
SRC	62.82%	62.10%	59.65%	57.93%	54.61%

3.2. Recognition comparison of our LRRNet with PCANet, SRC and SDR on MultiPIE dataset with corrupted images

How about the recognition for the heavily corrupted images? Can those methods keep on the good performance? Now we carry out Experiment 3.2 on the corrupted images in MultiPIE and AR datasets. For consistency, the set of parameters of LRRNet and PCANet are same as those in Experiment 3.1 (see Table 1), except for the occlusion. Here the occlusion is produced randomly by computer, and its proportion ranges from 10% to 50%. Then the results of Experiment 3.2 on MultiPIE dataset are shown in Table 3.

As observed from Table 3, our proposed LRRNet method has the outstanding performance with the increase of corruption, while the recognition rate of other methods decline rapidly. The reason is that when the images are corrupted, PCANet method considers the corrupted values as more important parts and the corrupted areas are randomly distributed. Therefore, its recognition result is certainly poor, just like the over-fitting. While in the LRRNet, original image is recovered before so the corrupted regions are almost completed according to the known pixels. Furthermore, it extracts the row-rank part and the little values that have not been recovered as well as sparse noise are removed and will not affect the performance. However, as the rate of noise increases, the known pixels decreased, and the image recover algorithm will complete the corrupted values with the irrelevant parts, so the recognition rate comes down.

In Table 3, matrix completion is applied in our proposed LRRNet method, not for other methods. In the references that introduce PCANet, SRC and SDR methods, there are also experiments that show its good robustness to noises. In order to look fair, here we also carry out experiments that add the process of denoising via matrix completion for PCANet, SRC and SDR methods, too. Then their corresponding experimental results on MultiPIE database are shown in Table 4.

As observed from Table 4, our proposed LRRNet method and PCANet method have better recognition rate than SDR and SRC do with the increase of corruption. Of course, after adding the step of denoising for PCANet, SDR, and SRC methods, the whole recognition rates are improved than those in Table 3. And it is clearest for PCANet than other methods.

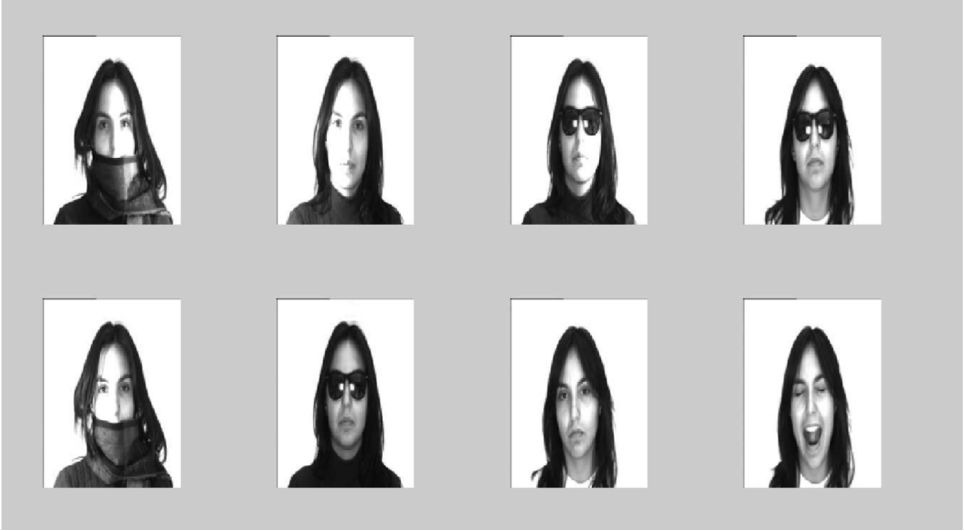


Fig. 3. The samples of AR.

Table 5
Face recognition comparison on AR dataset with different corrupted images.

Method	10%	20%	30%	40%	50%
LRRNet	**97.85**%	**98.23**%	**97.54**%	**96.15**%	**94.46**%
PCANet	87.92%	82.92%	77%	71.54%	68.31%
SDR	79.49%	78.11%	56.91%	44.93%	19.35%
SRC	77.19%	66.13%	58.76%	48.39%	37.79%

Table 6
Recognition comparison of our LRRNet with PCANet, SDR, and SRC methods on AR dataset with different corrupted images denoised via matrix completion.

Methods	10%	20%	30%	40%	50%
LRRNet	**98.15**%	**97.92**%	97.38%	**96.77**%	95.08%
PCANet	97.92%	97.69%	**97.54**%	96.31%	94.31%
SDR	96.08%	96.46%	96.46%	96.69%	**96.85**%
SRC	96.15%	96.00%	96.46%	96.31%	96.46%

3.3. Recognition comparison of our LRRNet with PCANet, SRC and SDR on AR dataset with corrupted images

The set for AR database is same as that in Experiment 3.1. For easier comparison, the images are cropped to size of 165 × 120 similar to the size for PCANet in the paper (Chan et al., 2015). The results of Experiment 3.3 on AR database are shown in Table 5.

As shown in Table 5, when the rate of noise increases, the recognition rate of PCANet decreases as well as SRC and SDR, but our LRRNet still keep good performance. Furthermore, the average performance on AR database is better than on MultiPie. The reason is that AR database is various on each subject. As shown in Fig. 3, the samples are full of expressions, occlusion like sunglasses and scarf. Because of this, the face recognition algorithms learn the intrinsic information of each subject and will not reach such high recognition rate as MultiPie with original images. However, when it comes to corrupted images, it will stay steady as little ruin will not affect the nature of the image. If we make the images heavily corrupted, the precision deteriorated, the best performance is produced by our LRRNet, and the next is PCANet. Even though SDR looks like better than SRC, as the corrupted rate increase, we can find that the SRC is more robust.

Similar to Table 4 shown in Section 3.2, here we also carry out experiments that add the process of denoising via matrix completion for PCANet, SRC and SDR methods on AR database. Then their corresponding experimental results on AR database are shown in Table 6.

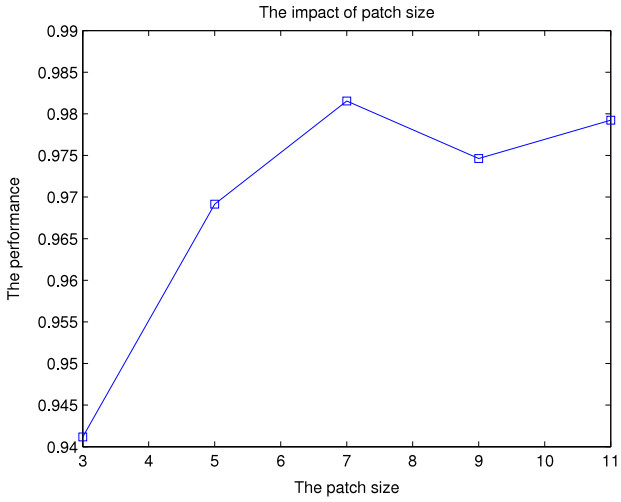


Fig. 4. The impact of patch size to our LRRNet method on MultiPie database.

As observed from Table 6, all the methods have better recognition rate than they do on MultiPIE dataset in Table 4. And our proposed LRRNet method almost has the best results than those for PCANet, SDR and SRC methods, which implies that the part of CNN in our proposed LRRNet is instructive and effective.

3.4. The impact of parameters

As we all know, parameter setting is usually important in experiment. Sometimes a result maybe ruined only because of a little change of them. In this part, we analyze the impact of parameters in our LRRNet. In our experiment, the parameters need to be tuned are little, like patch size, number of filters and block size. We first select the MultiPie database with 10% noise rate, and then we make patch size change from 3 × 3 to 11 × 11, the other parameters are same as Experiment 3.3, we make a plot below (Fig. 4) to show the influence, where the number of across coordinate means the patch size. For example, across coordinate 3 means that the patch size is 3 × 3.

Fig. 4 shows that as the patch size grows, the performance becomes better at first, then it deteriorated. The best patch size is

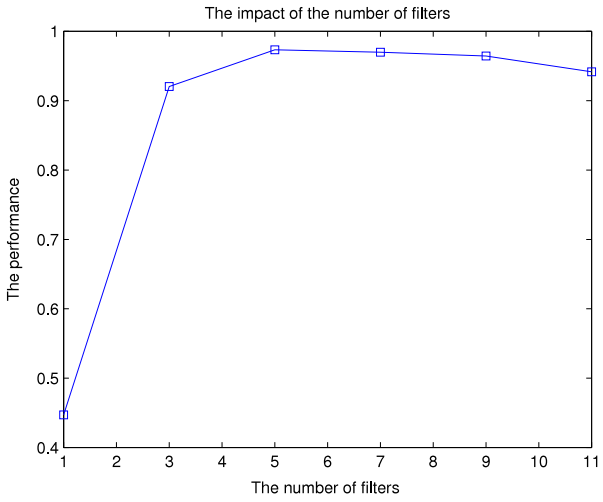


Fig. 5. The impact of numbers of filters to our LRRNet method on MultiPie database.

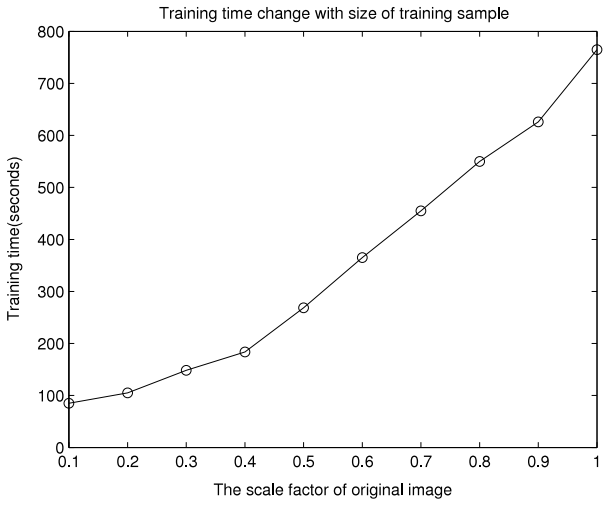


Fig. 6. The relation between training time and the size of training sample for our LRRNet method.

Table 7
The computational cost of different methods.

Method	LRRNet	PCANet	SRC	SDR
Time (h)	0.84	**0.81**	1734	51.28

Table 8
The time for applying trained classifier to an image from testing sample (time per image).

Method	PCANet	SDR	LRRNet
Time (s)	0.49	46	0.37

about 7 × 7. Because if the patch size is too small, the LRRNet will choose nearly all information of image as feature, then we did not know which one is critical to classification. And if the patch size is too big, we will lost much information that is essential for face recognition.

Now we begin to test the impact of number of filters for face recognition since the number of filters determinate the size of features in a way. Here we choose the same model as above, i.e. the patch size used here is 7 × 7, and then we make the number of filters change from 1 to 11. The results are shown in Fig. 5.

As shown in Fig. 5, the trend of curve is similar to the one of patch size. As the number grows, the performance begins with poor, then reach a peak, and deteriorated later. The reason of the phenomenon is that when the filters are few, we cannot describe the image clearly. So each image has its own "sweet point" that is the most suitable number to trace out the image, and other numbers maybe to be also good but still not the best.

3.5. The computational cost

Now we analyze the computational cost of LRRNet. Here we choose the images in MultiPIE database with 30% noise rate, the size of training samples is 80 × 60, and the number for training and testing is both 3468. Without losing generality, we also compare the cost with PCANet, SRC and SDR. First we have a view of the total computational cost as below in Table 7.

Table 7 shows that our LRRNet and PCANet cost less time than SDR and SRC methods. PCANet only costs 0.81 h (48.6 min), and LRRNet only costs 0.84 h (50.4 min). Although our LRRNet takes 2 min than PCANet, but it get a better performance (see Tables 2, 3). In the training stage, LRRNet costs nearly 44.68% additional time of recover as LRRNet use low-rank algorithm. While it needs less filters to achieve the same performance. As a result, it takes less time for test each sample. For example, LRRNet needs 0.37 s per test sample, while PCANet needs 0.49 s. Therefore, even though the LRRNet looks more complex, the computational cost is not far from PCANet. SRC method costs about 32 min for every sample (SRC combines training and testing together, we can only get the time for samples totally). If the number is small such like hundreds of samples, the computational cost of each image is only about 30 s. The time keeps an exponential growth, and it is suit for the small sample data. SDR contains a dictionary decomposition of supervised low-rank (SLR), which brings down the computational cost. In its total 51.28 h, it takes 7.28 h to execute the SLR and then cost 46 s for every sample.

Furthermore, in order to explore how training time of our LRRNet method changes with the size of training sample, we carry out the responding experiment on AR database. We take 1300 images as training samples, and the size of original image is 165 × 120. Then the relation between training time and the size of training sample of our LRRNet method is shown in Fig. 6, where the horizontal axis means the scale factor of original image, and the vertical axis means the training time.

As observed from Fig. 6, we can see that when the scale factor of original image increases, i.e. the size of training sample increases, the training time of our LRRNet method will increase.

The result of applying trained classifier to an image from testing sample (time per image) is shown in Table 8.

Finally, in order to understand the computation cost for each step of our proposed LRRNet, we carry out face recognition with LRRNet method for 10 times on AR database with 30% pixels missing, and take the average time as the final cost time. We find that the step of matrix completion for the corrupted images takes 808.46 s, nearly 13.5 min, the step of feature extraction and encode (binaryzation and histogram) takes 1723.68 s, nearly 28.7 min, and the step of SVM training takes 1300 s, nearly 21.67 min.

4. Conclusions

We have proposed a novel deep learning algorithm, named LRRNet, to solve the corrupted face recognition with high effect based on matrix completion and artificial neural network. The proposed LRRNet first recovers the incomplete input images with an efficient matrix completion algorithm, TNNR-ADMM, then processes it with a scalable and fast low-rank matrix factorization, RPCA-IALM method, to extract some low-rank parts as the filters of artificial neural network. Finally, the operation of convolution,

binaryzation, and histogram are fully utilized to generate the features for classification.

The proposed LRRNet keeps the idea of using deep learning to extract high-level features for classification without learning filters as ScatNet and PCANet do. While different with ScatNet and PCANet, we apply the low-rank matrix factorization to extract the low-rank part as filters, which provides a method for constructing a deep learning frame.

Extensive comparison experiments of our LRRNet with PCANet, SRC, and SDR on several benchmark databases demonstrated that our proposed LRRNet possesses relatively satisfactory performance. Experiment 3.1 showed that our LRRNet does have similar excellent performance as PCANet and SDR do for standard images of MultiPIE and AR database, which implies that our proposed LRRNet provided a way for solving face recognition problem. Experiment 3.2 and Experiment 3.3 showed that for the corrupted images of MultiPIE and AR database, our LRRNet does have a more outstanding recognition rate than PCANet, SRC, and SDR, which supported the goal of our proposed LRRNet that it will provide a way for solving incomplete face recognition problem. Experiment 3.5 showed that our proposed LRRNet not only achieved a high recognition rate, but also costed relatively less time, which is the goal we want to attain.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (61571410 and 61672477).

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041.

Bicego, M., Lagorio, A., Grosso, E., & Tistarelli, M. (2006). On the use of SIFT features for face authentication. In *2006 conference on computer vision and pattern recognition workshop*. (CVPRW'06), (pp. 35–42). IEEE.

Bruna, J., & Mallat, S. (2013). Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1872–1886.

Cai, J.-F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, *20*(4), 1956–1982.

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, *9*(6), 717–772.

Candès, E. J., & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, *56*(5), 2053–2080.

Cao, F., Cai, M., & Tan, Y. (2015). Image interpolation via low-rank matrix completion and recovery. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(8), 1261–1270.

Cao, F., Chen, J., Ye, H., Zhao, J., & Zhou, Z. (2016). Recovering low-rank and sparse matrix based on the truncated nuclear norm. *Neural Networks*. http://dx.doi.org/10.1016/j.neunet.2016.09.005.

Cavalcanti, G. D., Ren, T. I., & Pereira, J. F. (2013). Weighted modular image principal component analysis for face recognition. *Expert Systems with Applications*, *40*(12), 4971–4977.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). PCANet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, *24*(12), 5017–5032.

Deng, W., Hu, J., & Guo, J. (2012). Extended SRC: Undersampled face recognition via intraclass variant dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(9), 1864–1870.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. C., & Bengio, Y. (2013). Maxout networks. *ICML (3)*, *28*, 1319–1327.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-pie. *Image and Vision Computing*, *28*(5), 807–813.

Haeffele, B. D., Young, E. D., & Vidal, R. (2014). Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proceedings of the 31st international conference on machine learning*, Vol. 105, no. 3–4 (pp. 604–609).

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H.-J. (2005). Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 328–340.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hong, B., Wei, L., Hu, Y., Cai, D., & He, X. (2016). Online robust principal component analysis via truncated nuclear norm regularization. *Neurocomputing*, *175*, 216–222.

Hu, Y., Jin, Z., Shi, D., Zhang, D., Cai, D., & He, X. (2015). Large scale multi-class classification with truncated nuclear norm regularization. *Neurocomputing*, *148*, 310–317.

Hu, Y., Zhang, D., Ye, J., Li, X., & He, X. (2013). Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(9), 2117–2130.

Huang, Z.-H., Li, W.-J., Shang, J., Wang, J., & Zhang, T. (2015). Non-uniform patch based face recognition via 2D-DWT. *Image and Vision Computing*, *37*, 12–19.

Jacob, A., Francis, B., Theodoros, E., & Jean-Philippe, V. (2006). Low-rank matrix factorization with attributes. *Computer Science*, arXiv preprint cs/0611124.

Ji, H., Liu, C., Shen, Z., & Xu, Y. (2010). Robust video denoising using low rank matrix completion. In *2010 IEEE conference on computer vision and pattern recognition*. (CVPR), (pp. 1791–1798). IEEE.

Jiang, X., & Lai, J. (2015). Sparse and dense hybrid representation via dictionary decomposition for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(5), 1067–1079.

Jonathan, E., & Dimitri, P. B. (1992). On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, *55*(1), 293–318.

Kavukcuoglu, K., Sermanet, P., Boureau, Y.-L., Gregor, K., Mathieu, M., & Yann, L. (2010). Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems* (pp. 1090–1098).

Kevin, J., Koray, K., Marc'Aurelio, R., & Yann, L. (2009). What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International conference on computer vision* (pp. 2146–2153). IEEE.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

Lin, Z., Chen, M., & Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv Preprint arXiv:1009.5055.

Liu, Q., Lai, Z., Zhou, Z., Kuang, F., & Jin, Z. (2016). A truncated nuclear norm regularization method based on weighted residual error for matrix completion. *IEEE Trans. Image Process.*, *25*(1), 316–330.

Lu, G.-F., Zou, J., & Wang, Y. (2012). Incremental complete LDA for face recognition. *Pattern Recognition*, *45*(7), 2510–2521.

Ma, S., Goldfarb, D., & Chen, L. (2011). Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, *128*(1–2), 321–353.

Ma, L., Wang, C., Xiao, B., & Zhou, W. (2012). Sparse representation for face recognition based on discriminative low-rank dictionary learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. (CVPR), (pp. 2586–2593). IEEE.

Martinez, A. M. (1998). The AR face database. CVC Technical Report, Vol. 24.

Mobahi, H., Collobert, R., & Weston, J. (2009). Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 737–744). ACM.

Recht, B., Fazel, M., & Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, *52*(3), 471–501.

Secchi, P., Vantini, S., & Zanini, P. (2016). Hierarchical independent component analysis: a multi-resolution non-orthogonal data-driven basis. *Computational Statistics & Data Analysis*, *95*, 133–149.

Sifre, L., & Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1233–1240).

Singh, C., Mittal, N., & Walia, E. (2011). Face recognition using Zernike and complex Zernike moment features. *Pattern Recognition and Image Analysis*, *21*(1), 71–81.

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1891–1898).

Tara, N. S., Brian, K., Vikas, S., Ebru, A., & Bhuvana, R. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *IEEE International conference on acoustics* (pp. 6655–6659).

Toh, K.-C., & Yun, S. (2010). An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, *6*(615–640), 15.

Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings*. CVPR'91, (pp. 586–591). IEEE.

Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., & Ma, Y. (2012). Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(2), 372–386.

Wang, J., Lu, C., Wang, M., Li, P., Yan, S., & Hu, X. (2014). Robust face recognition via adaptive sparse representation. *IEEE Transactions on Cybernetics*, *44*(12), 2368–2378.

Wang, Y., & Su, X. (2014). Truncated nuclear norm minimization for image restoration based on iterative support detection. *Mathematical Problems in Engineering*, *2014*, 1–17.

Wiskott, L., Fellous, J.-M., Kuiger, N., & Von Der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775–779.

Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems* (pp. 2080–2088).

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J.-y. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(1), 131–137.

Yang, M., Zhang, L., Yang, J., & Zhang, D. (2011). Robust sparse coding for face recognition. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*. (CVPR), (pp. 625–632). IEEE.

Yonatan, A., Michael, F., Nathan, S., & Ullman, S. (2007). Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on machine learning* (pp. 17–24).

Yu, W., Gan, L., Yang, S., Ding, Y., Jiang, P., Wang, J., et al. (2014). An improved LBP algorithm for texture and face classification. *Signal, Image and Video Processing*, *8*(1), 155–161.

Yu, L., He, Z., & Cao, Q. (2010). Gabor texture representation method for face recognition using the Gamma and generalized Gaussian models. *Image and Vision Computing*, *28*(1), 177–187.

Zaiwen, W., Wotao, Y., & Yin, Z. (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*,(4), 333–361.

Zhao, Z.-Q., Cheung, Y.-m., Hu, H., & Wu, X. (2016). Corrupted and occluded face recognition via cooperative sparse representation. *Pattern Recognition*, *56*, 77–87.

Zhao, Z.-Q., Glotin, H., Xie, Z., Gao, J., & Wu, X. (2012). Cooperative sparse representation in two opposite directions for semi-supervised image annotation. *IEEE Transactions on Image Processing*, *21*(9), 4218–4231.

Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D. L., & Weng, J. (1998). Discriminant analysis of principal components for face recognition. In *Face Recognition* (pp. 73–85). Springer.