Measurement 136 (2019) 429–437

Contents lists available at ScienceDirect

Measurement

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

A Sparse Corruption Non-Negative Matrix Factorization method and application in face image processing & recognition

Zhibo Guo *, Ying Zhang

College of Information Engineering, Yangzhou University, Yangzhou 225009, PR China

ARTICLE INFO

Article history:
Received 5 November 2018
Received in revised form 20 December 2018
Accepted 25 December 2018
Available online 27 December 2018

Keywords:
Image processing
Dimensionality reduction
L_1-norm
Matrix factorization
Face recognition

ABSTRACT

Non-negative Matrix Factorization (NMF) has attracted widely attentions in the areas of data analysis, image processing & measurement and noise separation. NMF can obtain the non-negative low-dimensional representation of the data and low-rank matrix is of great importance to classification and recognition. Classic NMF assumes that noise is subject to Gaussian noise or Poisson noise, while it is inapplicable to some other noise. In addition, corrupted data in the test set cannot be restored by NMF. Many scholars have studied NMF and proposed a variety of improved algorithms so far, such as Robust Non-Negative Matrix Factorization (RNMF), which is designed to deal with sparse corruption. Our paper presents Sparse Corruption Non-Negative Matrix Factorization (SCNMF). SCNMF separates a sparse noise matrix out of the corrupted input matrix, and the rest of the input matrix is represented as the product of two low-dimensional matrices. The product of two low-dimensional matrices approximates the non-corrupted input matrix. The base matrix calculated by the proposed method is tolerant to noise. It can reconstruct new data well regardless of whether the new data is corrupted and non-corrupted. Experiments on specific face databases verifies the validity of SCNMF. Reconstructed faces by the proposed method are clearer and more recognizable and the recognition rate is three percentage points higher by comparison.

1. Introduction

Nowadays, image processing & measurement technology is developing rapidly, which has gradually occupied an increasingly important position in people's daily life. The NMF algorithm is widely used in various fields including the areas mentioned above. Specifically, background separation, image denoising, image classification, etc can be processed by NMF algorithm. In real world, we are often confronted with high-dimensional data [1,2], especially in the realm of data analysis, signal processing and pattern recognition, which has posed serious problems to our calculation and analysis. Dimension reduction has become an important preprocessing method in most data processing. Therefore, researchers have been devoted themselves to finding a way to describe high-dimensional data. Usually, a satisfactory method should have the following two characteristics: (1) Potential characteristics of the original data can be discovered; (2) Data can be dimensionality reduced [3].

Taking face database for example, in the early 1990s, how to extract the geometric structure of face was the main research content of face recognition. Mid-to-late 1990s was an important period for face recognition, during which period reducing the dimensionality of data to make it easier to recognize was studied. The classic algorithms of Principal Component Analysis (PCA) [4–7], Linear Discriminant Analysis (LDA) [8,9] appeared during this period. After 1998, statistical techniques were applied to face recognition and become one of the mainstream algorithms. The scholars were committed to solving the problem of illumination change, expression effect and face occlusion in face recognition field during that period.

Non-Negative Matrix Factorization (NMF) [10] was first presented by Lee et al in 1999. It is a subspace method. The NMF method divides the given original input data into the product of two matrices. The two derived matrices are non-negative and dimensionally lower than the input matrix. A number of improved approaches appeared and attracted great attention after the NMF algorithm was proposed [11–15]. Local Non-Negative Matrix Factorization (LNMF) [17] extracts local characteristics of the input data. Sparse Non-Negative Matrix Factorization (SNMF) [11,18] imposed sparse-constrained on the squared error based objective function. Improved Non-Negative Matrix Factorization (INMF) [19,20] adds more restrictions to the objective function. Discriminant Non-Negative Matrix Factorization (DNMF) [12] take into account image class information. Orthogonal Non-Negative Matrix Factorization (ONMF) [13,14] strengthen localized features and

* Corresponding author.
E-mail addresses: zbguo@yzu.edu.cn (Z. Guo), mx120170419@yzu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.measurement.2018.12.087
0263-2241/

impose orthonormal characteristic of NMF. Kernel Non-Negative Matrix Factorization (KNMF) [15,25] extracts nonlinear non-negative features of images successfully.

There are disadvantages of NMF. Firstly, NMF assumes that noise is subject to Gaussian noise or Poisson noise, which is not always true in practical problems. Secondly, data matrix in the training set can be reconstructed by $\boldsymbol{W}$ and $\boldsymbol{H}$, but the corrupted ones in the test set cannot be dealt with in the same way.

Experiment shows NMF can't reconstruct corrupted samples well due to its structure. To solve the problems mentioned above, this article proposes a new matrix factorization algorithm. The algorithm is called Sparse Corruption Non-Negative Matrix Factorization, by which initial input data is decomposed into a sparse error matrix and a reconstructed data matrix. More remarkable, L_1-norm and soft threshold operator are used in solving the problem. Furthermore, corrupted testing samples can be reconstructed by $\boldsymbol{H}$ that be computed. Our experiments show that SCNMF performs well in image de-noise, image reconstruction, and image classification on specific face databases.

The structure of this paper is organized as follows. Section 2 introduces NMF and its optimization method, especially the improved algorithm RNMF is introduced. Section 3 gives the detailed algorithms of SCNMF, optimization methods, algorithm flow as well as the method of using $\boldsymbol{H}$ to reconstruct the test data. Experimental results and discussion are shown in Section 4. The last part summarizes this paper.

2. Related work

A large number of improved NMF methods have been proposed for data representing and processing so far. We only provide a brief review of NMF and RNMF to ease your reading in this section.

2.1. NMF

NMF is an efficient matrix low-rank approximation method [16]. To obtain the potential characters of the input matrix, we represent it by the multiplication of a base matrix and a coefficient matrix. Supposing the input data is matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n]^{\mathrm{T}} \in R^{n\times m}$, where vector $\boldsymbol{d}_i \in R^m (i = 1, 2, \ldots, n)$ is one piece of input data. The purpose of NMF is to find two new matrices, multiplying the two new matrices can obtain the original matrix $\boldsymbol{D} \approx \boldsymbol{WH}$.

To get the optimal solutions, we usually adopt an iterative method. Before that, the objective function should be determined. Two different objective functions are constructed by Lee and Seung [10]. Different objective functions and two different solutions are proposed in document [10]. The proof of the convergence theorem is given meanwhile.

1. One objective function of NMF calculates the square of the euclidean distance between $\boldsymbol{D}$ and $\boldsymbol{WH}$, which is expressed as follows.

$$\min_{\boldsymbol{W},\boldsymbol{H}} \| \boldsymbol{D} - \boldsymbol{WH} \|_F^2 \qquad \text{s.t. } \boldsymbol{W} \geq 0,\ \boldsymbol{H} \geq 0 \tag{1}$$

where $\| \boldsymbol{A} \|_F$ is the sum of the squares of each element of matrix $\boldsymbol{A}$, whose lower bounds is zero. The problem is figured out by the maximum likelihood method under the Gaussian noise assumption.

The update rules for $\boldsymbol{W}$ and $\boldsymbol{H}$ are given by:

$$\boldsymbol{W}_{ih} = \boldsymbol{W}_{ih} \frac{\left(\boldsymbol{DH}^{\mathrm{T}}\right)_{ih}}{\left(\boldsymbol{WHH}^{\mathrm{T}}\right)_{ih}} \tag{2}$$

$$\boldsymbol{H}_{hj} = \boldsymbol{H}_{hj} \frac{\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{D}\right)_{hj}}{\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{WH}\right)_{hj}} \tag{3}$$

2. Another objective function of NMF is to use the Kullback-Leibler (K-L) distance as a measure of the dissimilarity between $\boldsymbol{D}$ and $\boldsymbol{WH}$.

$$\min_{\boldsymbol{W},\boldsymbol{H}} D(\boldsymbol{D} \| \boldsymbol{WH}) = \sum_{ij} \left(\boldsymbol{D}_{ij} \ln \frac{\boldsymbol{D}_{ij}}{(\boldsymbol{WH})_{ij}} - \boldsymbol{D}_{ij} + (\boldsymbol{WH})_{ij} \right)$$
$$\text{s.t. } \boldsymbol{W} \geq 0,\ \boldsymbol{H} \geq 0 \tag{4}$$

Same to square Euclidean distance, K-L distance reach the minimal value only when $\boldsymbol{D} = \boldsymbol{WH}$. The objective function is solved under the Poisson noise assumption.

The following multiplicative updates are the optimal solution of the above problem.

$$\boldsymbol{W}_{iq} = \boldsymbol{W}_{iq} \frac{\sum_{j=1}^{n} \boldsymbol{H}_{qj} \boldsymbol{D}_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{j=1}^{n} \boldsymbol{H}_{qj}} \tag{5}$$

$$\boldsymbol{H}_{qj} = \boldsymbol{H}_{qj} \frac{\sum_{i=1}^{m} \boldsymbol{W}_{iq} \boldsymbol{D}_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{i=1}^{m} \boldsymbol{W}_{iq}} \tag{6}$$

The NMF algorithm has been widely used in various fields after it was proposed and a number of improved approaches appeared. Local Non-Negative Matrix Factorization (LNMF) [17] extracts local characteristics of the input data. Sparse Non-Negative Matrix Factorization (SNMF) [18] imposed sparseness constraints on the squared error based objective function. Improved non-negative matrix factorization (INMF) [19,20] adds more restrictions to the objective function.

These methods [17,18] acquire good results with little change in gestures and expressions, while they can't work well on corrupted faces [18]. The reason for the decline in recognition rate is that gestures, facial expressions or corruption are improperly disposed, therefore, some images are misclassified to other classes that are further away from the center of the class.

2.2. RNMF

Two common objective functions mentioned above are both assumed that noise is Gaussian noise or Poisson noise [21]. While in practical image processing, the errors in the data are not limited to Gaussian or Poisson noise. The traditional assumptions of NMF are meaningless under this case, so literature [22] put forward Robust Non-Negative Matrix Factorization (RNMF), which provides a solution to this problem.

The model of NMF is as follows.

$$\boldsymbol{D} \approx \boldsymbol{WH} + \boldsymbol{S} \tag{7}$$

Input matrix $\boldsymbol{D}$ is corrupted arbitrarily. Matrix $\boldsymbol{S} \in R^{n\times m}$ is adopted to represent the sparse noise. $\boldsymbol{W} \in R^{n\times r}$ as well as $\boldsymbol{H} \in R^{r\times m}$ are non-negative. Matrix $\boldsymbol{S}$ can separate the noise, which makes this method has better robustness than the traditional NMF.

The objective function of RNMF is as following, which calculates the square of the euclidean distance.

$$\| \boldsymbol{D} - \boldsymbol{WH} - \boldsymbol{S} \|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \| \boldsymbol{D}_{ij} - (\boldsymbol{WH})_{ij} - \boldsymbol{S}_{ij} \|^2 \tag{8}$$

Since the matrix $\boldsymbol{S}$ is sparse, the L_0-norm of $\boldsymbol{S}$ is considered to be limited.

$$\| \boldsymbol{S} \|_0 \leq \nu \tag{9}$$

As a result, one will need to solve the following minimization problem:

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{S}} \| \boldsymbol{D} - \boldsymbol{WH} - \boldsymbol{S} \|_F^2 \qquad \text{s.t. } \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0, \| \boldsymbol{S} \|_0 \leq \nu \tag{10}$$

L_1-norm is often used to replace L_0-norm, because L_0-norm is non-convex and inconvenient to calculate. Consequently, the convex relaxation of Eq. (10) is reformulated as

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{S}} \| \boldsymbol{D}-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_F^2+\lambda\|\boldsymbol{S}\|_1 \qquad \text{s.t. } \boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0 \tag{11}$$

In Eq. (11) $\lambda>0$ is an adjustable regularization parameter.

3. SCNMF method

Inspired by the traditional NMF and RNMF in document [22], we proposed Sparse Corruption Non-Negative Matrix factorization (SCNMF) to deal with sparse noise. In this part, the details of the proposed algorithm will be introduced. We explain why the method is proposed and adopted, and then give the model and solutions to the algorithm.

3.1. Motivation

The goal of SCNMF is to obtain trained $\boldsymbol{W}$ and $\boldsymbol{H}$. Sparse corruption is added to the non-corrupted data matrix $\boldsymbol{D}$ to get corrupted data matrix $\boldsymbol{D}'$, and both of matrix $\boldsymbol{D}$ and $\boldsymbol{D}'$ are used during training. The particularity of the input matrix allows the algorithm to reconstruct both corrupted and non-corrupted test set, which means the ideal $\boldsymbol{W}$ and $\boldsymbol{H}$ will be tolerant of specific corrupted data in the test set.

3.2. Objective function

Same to RNMF, sparse matrix represents an error matrix. Non-corrupted input matrix $\boldsymbol{D}$ can be given by:

$$\boldsymbol{D}\approx\boldsymbol{W}\boldsymbol{H} \tag{12}$$

Corrupted input matrix $\boldsymbol{D}'$ can be given by:

$$\boldsymbol{D}'\approx\boldsymbol{W}\boldsymbol{H}+\boldsymbol{S} \tag{13}$$

For $\boldsymbol{S}$ is a sparse matrix, so the SCNMF method uses sparse character as objective function. One will need to settle the minimization problem in Eq. (14):

$$\begin{aligned}&\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{S}\|_0\\ &\text{s.t. } \boldsymbol{D}\approx\boldsymbol{W}\boldsymbol{H},\ \boldsymbol{D}'\approx\boldsymbol{W}\boldsymbol{H}+\boldsymbol{S}\\ &\boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0\end{aligned} \tag{14}$$

Considering L_0-norm is NP-hard, thus to convert Eq. (14) into L_1-norm by convex relaxation:

$$\begin{aligned}&\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{S}} \|\boldsymbol{S}\|_1+\alpha_1\|\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_F^2+\alpha_2\|\boldsymbol{D}-\boldsymbol{W}\boldsymbol{H}\|_F^2\\ &\text{s.t. } \boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0\end{aligned} \tag{15}$$

We can borrow approach from non-negative quadratic programming problem when calculating the value of $\boldsymbol{W}$ and $\boldsymbol{H}$. By setting $\boldsymbol{W}$ and $\boldsymbol{S}$ in Eq. (15) to fixed values, the update rule of $\boldsymbol{H}$ can be obtained by solving the following equation.

$$\begin{aligned}&\alpha_1\|\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_F^2+\alpha_2\|\boldsymbol{D}-\boldsymbol{W}\boldsymbol{H}\|_F^2\\ &=\alpha_1\|\boldsymbol{S}-\boldsymbol{D}'+\boldsymbol{W}\boldsymbol{H}\|_F^2+\alpha_2\|\boldsymbol{D}-\boldsymbol{W}\boldsymbol{H}\|_F^2\\ &=\alpha_1\mathrm{Tr}\Big[\Big((\boldsymbol{S}-\boldsymbol{D}')^{\mathrm{T}}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\Big)(\boldsymbol{S}-\boldsymbol{D}'+\boldsymbol{W}\boldsymbol{H})\Big]\\ &\quad+\alpha_2\mathrm{Tr}\Big[\Big(\boldsymbol{D}^{\mathrm{T}}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\Big)(\boldsymbol{D}-\boldsymbol{W}\boldsymbol{H})\Big]\\ &=\alpha_1\mathrm{Tr}\Big(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)+2\alpha_1\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}')^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)\\ &\quad+\alpha_1\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}')^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}')\Big)+\alpha_2\mathrm{Tr}\Big(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\Big)\\ &\quad-2\alpha_2\mathrm{Tr}\Big(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)+\alpha_2\mathrm{Tr}\Big(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)\end{aligned} \tag{16a}$$

Parameter α_1 and α_2 are used to adjust the weight. The corrupted training sample is as much as the non-corrupted sample in our experiment, so we considering $\alpha_1=\alpha_2=\alpha$. While changing the proportion of non-corrupted and corrupted samples in the training sample, the parameters should be adjusted in the same proportion. Then we get

$$\begin{aligned}&\alpha_1\|\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_F^2+\alpha_2\|\boldsymbol{D}-\boldsymbol{W}\boldsymbol{H}\|_F^2\\ &=2\alpha\mathrm{Tr}\Big(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)\\ &\quad+2\alpha\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)+\alpha\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}')^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}')\Big)\\ &\quad+\alpha\mathrm{Tr}\Big(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\Big)=2\alpha\Big[\mathrm{Tr}\Big(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)+\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)\Big]\\ &\quad+\alpha\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}')^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}')\Big)+\alpha\mathrm{Tr}\Big(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\Big)\end{aligned} \tag{16b}$$

Considering $\boldsymbol{D}\approx\boldsymbol{W}\boldsymbol{H}$ and $\boldsymbol{D}'\approx\boldsymbol{W}\boldsymbol{H}+\boldsymbol{S}$, the formulas degenerate to Eq. (17) to solve $\boldsymbol{H}$:

$$\min_{\boldsymbol{H}} 2\mathrm{Tr}\Big(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)+\mathrm{Tr}\Big((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big) \text{ s.t. } \boldsymbol{H}>0 \tag{17}$$

The form of Eq. (17) is closer to the following problem:

$$\begin{aligned}&\min_{\boldsymbol{y}} \quad \boldsymbol{y}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}\\ &\text{s.t.} \quad \boldsymbol{y}\geq 0\end{aligned} \tag{18}$$

According to document [23], Eq. (18) can be updated updates by the form:

$$y_i=\left[\frac{-b_i+\sqrt{b_i^2+16(\boldsymbol{M}^{+}\boldsymbol{y})_i(\boldsymbol{M}^{-}\boldsymbol{y})_i}}{4(\boldsymbol{M}^{+}\boldsymbol{y})_i}\right]y_i \tag{19}$$

where $\boldsymbol{M}^{+}$ and $\boldsymbol{M}^{-}$ are assigned the following value, it means $\boldsymbol{M}$ is non-negative:

$$\begin{aligned}\boldsymbol{M}_{ij}^{+}&=\begin{cases}\boldsymbol{M}_{ij}, & \text{if } \boldsymbol{M}_{ij}\geq 0\\ 0, & \text{otherwise}\end{cases}\\ \boldsymbol{M}_{ij}^{-}&=\begin{cases}|\boldsymbol{M}_{ij}|, & \text{if } \boldsymbol{M}_{ij}<0\\ 0, & \text{otherwise}\end{cases}\end{aligned} \tag{20}$$

The update rule of Eq. (19) can ensure that the cost function is Monotone non increasing. Applying Eq. (19) to solving the optimization problem (17), $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}$ is expressed by $\boldsymbol{M}$. In this case, we can obviously get:

$$\boldsymbol{M}^{+}=\boldsymbol{M}=2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W},\quad \boldsymbol{M}^{-}=0 \tag{21}$$

Therefore, we get the solution to problem (17):

$$\boldsymbol{H}_{ij}=\left[\frac{\Big|\Big(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\Big)_{ij}\Big|-\Big(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\Big)_{ij}}{8\Big(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\Big)_{ij}}\right]\boldsymbol{H}_{ij} \tag{22}$$

So similarly, we can get the following update rules of $\boldsymbol{W}$:

$$\boldsymbol{W}_{ij}=\left[\frac{\Big|\Big((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\boldsymbol{H}^{\mathrm{T}}\Big)_{ij}\Big|-\Big((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\boldsymbol{H}^{\mathrm{T}}\Big)_{ij}}{8\Big(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\Big)_{ij}}\right]\boldsymbol{W}_{ij} \tag{23}$$

The definition of soft threshold operator is:

$$\boldsymbol{S}_{ij}=\begin{cases}(\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H})_{ij}-\theta, & (\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H})_{ij}>\theta\\ (\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H})_{ij}+\theta, & (\boldsymbol{D}'-\boldsymbol{W}\boldsymbol{H})_{ij}<-\theta\\ 0, & \text{otherwise}\end{cases} \tag{24}$$

So far, we have obtained the update rules of our objective function, which is shown in Algorithm 1.

Algorithm 1 Proposed SCNMF Algorithm

Input: Data matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n]^{\mathrm{T}} \in \mathrm{R}^{n\times m}$, the rank of $\boldsymbol{W}$ and $\boldsymbol{H}$ (k), the iterations number (*times*)

1: $\boldsymbol{W} \leftarrow \boldsymbol{W}^0 \in \mathrm{R}^{n\times k}$

2: $\boldsymbol{H} \leftarrow \boldsymbol{H}^0 \in \mathrm{R}^{k\times m}$

3: for $i = 1$ to *times* do

4: $\boldsymbol{S} \leftarrow \boldsymbol{D} - \boldsymbol{WH}$

5: $\theta_{ip} \leftarrow \mathrm{mean}(\boldsymbol{S}_{ik}), \quad (p, k = 1, 2, \cdots, m)$

6: $$\boldsymbol{S}_{ij} \leftarrow \begin{cases} (\boldsymbol{D}' - \boldsymbol{WH})_{ij} - \theta_{ij}, & (\boldsymbol{D}' - \boldsymbol{WH})_{ij} > \theta_{ij} \\ (\boldsymbol{D}' - \boldsymbol{WH})_{ij} + \theta_{ij}, & (\boldsymbol{D}' - \boldsymbol{WH})_{ij} < -\theta_{ij} \\ 0, & \text{otherwise} \end{cases}$$

7: $$\boldsymbol{W}_{ij} = \left[\frac{\left|((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\boldsymbol{H}^{\mathrm{T}})_{ij}\right| - ((\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D})\boldsymbol{H}^{\mathrm{T}})_{ij}}{8(\boldsymbol{WHH}^{\mathrm{T}})_{ij}}\right]\boldsymbol{W}_{ij}$$

8: $$\boldsymbol{H}_{ij} = \left[\frac{\left|(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D}))_{ij}\right| - (\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{D}'-\boldsymbol{D}))_{ij}}{8(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{WH})_{ij}}\right]\boldsymbol{H}_{ij}$$

9: end for

Output: matrix $\boldsymbol{W}$ and matrix $\boldsymbol{H}$

The soft threshold operator is aimed at reducing each entry in $\boldsymbol{D}' - \boldsymbol{WH}$ (that is, the sparse corruption $\boldsymbol{S}$) Intuitively. The entries in $(-\theta_j, \theta_j)$ change to 0, those to the right of the interval decrease and left increase. So that the noise matrix tends to be sparse gradually. Fig. 1 presents the renderings before and after one iteration.

For non-square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, they are called generalized inverse matrix if $\boldsymbol{ABA} = \boldsymbol{A}$ and $\boldsymbol{BAB} = \boldsymbol{B}$. For $\boldsymbol{H}$, the formula $\boldsymbol{H}\mathrm{pinv}(\boldsymbol{H})\boldsymbol{H} = \boldsymbol{H}$ is true. So (25) is considered to reconstruct faces in the test set.

$$\boldsymbol{d}_i' \leftarrow \boldsymbol{d}_i \mathrm{pinv}(\boldsymbol{H})\boldsymbol{H} \tag{25}$$

where $\boldsymbol{d}_i$ is corrupted face, $\boldsymbol{d}_i'$ is reconstructed face.

4. Experimental results and discussion

We perform experiments on the Yale face database [26] and the ORL face database [27] to test the effectiveness of SCNMF. The algorithms compared are tuned to achieve their best performance by choosing the optimal parameters. The rank of SCNMF is chosen $k = 15$, the iterations number of SCNMF is chosen as 200 and threshold matrix θ is chosen as the mean of the corresponding row vectors in matrix $\boldsymbol{S}$. All experiments are done on the original face database without any image preprocessing. We conduct these trials fifty times independently and record the average accuracy finally.

The YALE database created by the Computing Vision and Control Center, Yale University, US. There are 165 images with facial expression changes or light changes in the database. These images are taken from fifteen volunteers with each one eleven images. Each image is a 112×92 pixel array and is converted into a vector. We validate the proposed algorithm on this database, in which the faces have expressions changes and lighting changes. Fig. 2 is partial sample from the Yale face database.

White blocks are used to simulate the noise of the picture in the experiment. We added three 10×10 white block noise on each image. Fig. 3 shows the samples with white block noise in the Yale face database.

The ORL face dataset has a total of 400 images, which consist of 40 people and 10 images per person and each image is a 112×92 pixel array. These images are collected during different periods, so facial gestures, facial proportion changes and details of these faces are different. For example, happy, sad, eyes-opened, eyes-closed, w/glasses, w/no glasses, Plane rotation can reach 20 degrees, and the scale changes as much as 10%.

Minimum Euclidean distance is used for classification. The training set consists of $k(k = 1, 2, 3, \cdots, 7)$ images randomly selected from per person, and the test set consists of the remaining images.

For the experiment of SCNMF, we choose $rank(rank = 10, 15, 20)$ for dimensionality reduction. The accuracy varies with different numbers of training samples per person as shown in Table 1. From Table 1, we can see that the dimensionality of basis matrix has effect on classification accuracy. When the dimensionality of basis matrix is too low, the accuracy is low as well, so the dimensionality of basis matrix is set 15 in the following experiment.

Figs. 4–6 show the denoising and reconstruction results of the corrupted images on Yale by NMF, RNMF and SCNMF respectively. Among them, (a) shows the images in the training set; (b) is the reconstructed image by $\boldsymbol{WH}$; (c) is the separated error matrix. The dimensionality of the faces is reduced from 10,304 to 15. The special noise assumption of the traditional NMF leads the algorithm unable to remove sparse corruption (Fig. 4). By comparison, SCNMF and RNMF obtain satisfying results, and the reconstructed faces of SCNMF are clearer and more recognizable. For a more objective evaluation, we use Structural Similarity Index Measure (SSIM) to evaluate the reconstruction quality. SSIM is an index for calculating structural similarity proposed by Wang et al. [24], which takes characteristics of the Human Visual System (HVS) into consideration. Experimental results prove that the image reconstructed by our proposed method is the most effective.

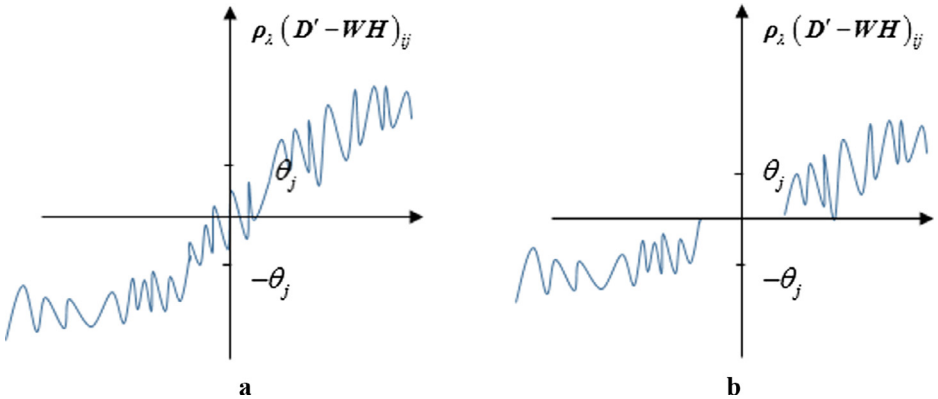

Fig. 1. The sketches of the soft threshold operator. a is the rendering before one iteration; b is the rendering after one iteration;

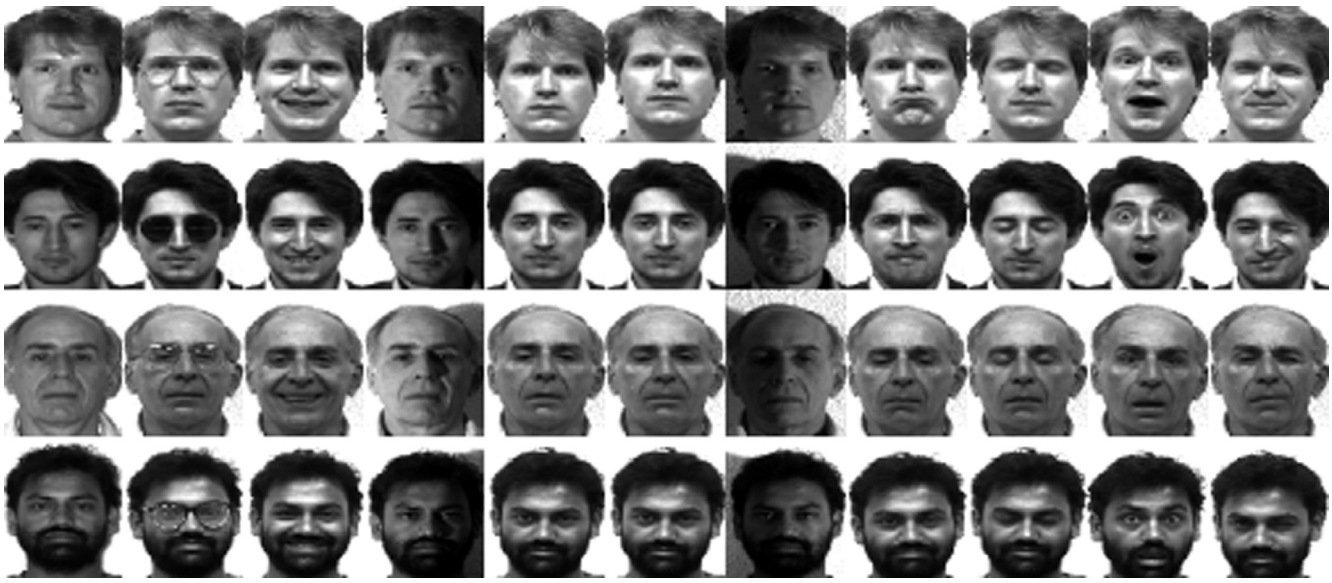

Fig. 2. Images from the Yale face database (Partial Sample).

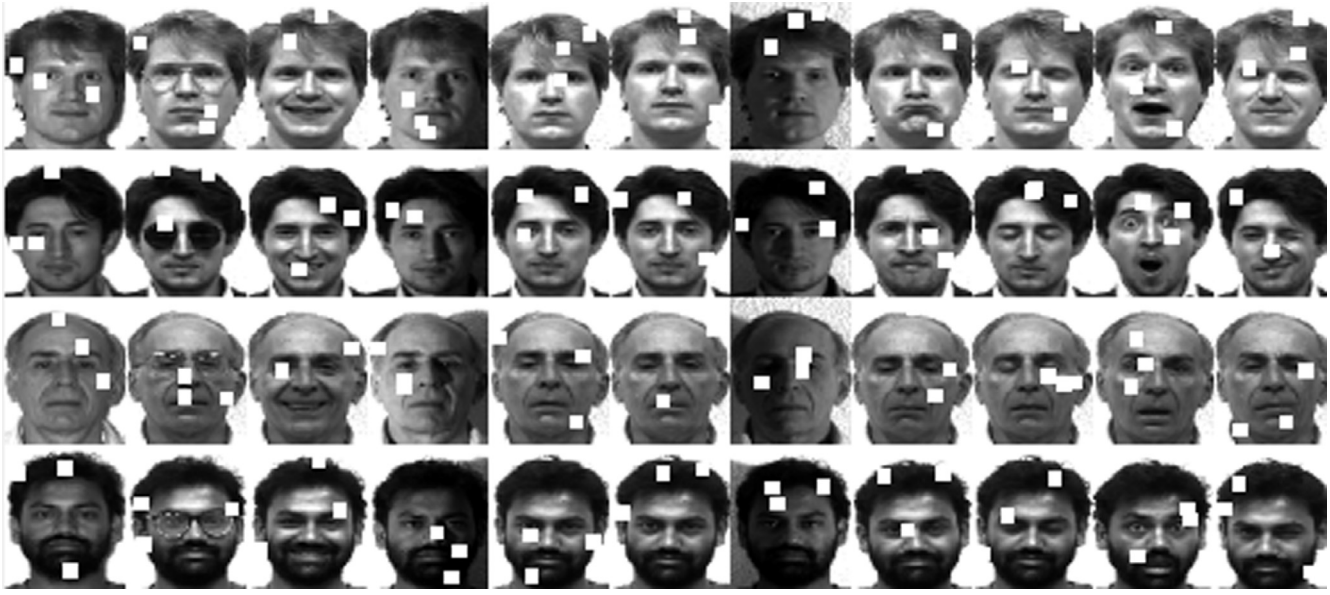

Fig. 3. Images with white block noise from the Yale face database (Partial Sample).

Table 1
Recognition results using SCNMF on the Yale with the value of rank and the numbers of training samples per person changing.

k	1	2	3	4	5	6	7
rank = 10	50.00	74.81	80.00	84.76	84.44	85.33	93.33
rank = 15	65.33	81.48	87.50	88.57	90.00	92.00	98.33
rank = 20	65.33	80.00	87.50	88.57	88.89	92.00	98.33

In addition, we conducted experiments to verify the effect of reconstructing sample in the test set. The subspace we train is compatible, so the algorithm has the ability to restore both corrupted and non-corrupted samples in the test set in theory. The experiment results are shown in Figs. 7 and 8. Faces with sparse noise are as shown in Fig. 7a and faces without noise are in Fig. 8a. Faces in Fig. 7b and Fig. 8b are reconstructed by Eq. (25). As the reconstructed image shows, there is no facial expression change and sparse corruption on reconstructed faces, which means the proposed algorithm has a good reconstruction effect on both corrupted and non-corrupted images, which is consistent with our expected results.

To illustrate the effectiveness of the proposed SCNMF objectively, we use the classification accuracy to evaluate the algorithm.

Fig. 9 is a line chart showing different recognition accuracy of SCNMF, NMF and RNMF methods on non-corrupted Yale face database. Obviously showed in Fig. 8, the SCNMF algorithm works better than the other two algorithms through accurate data comparison. In

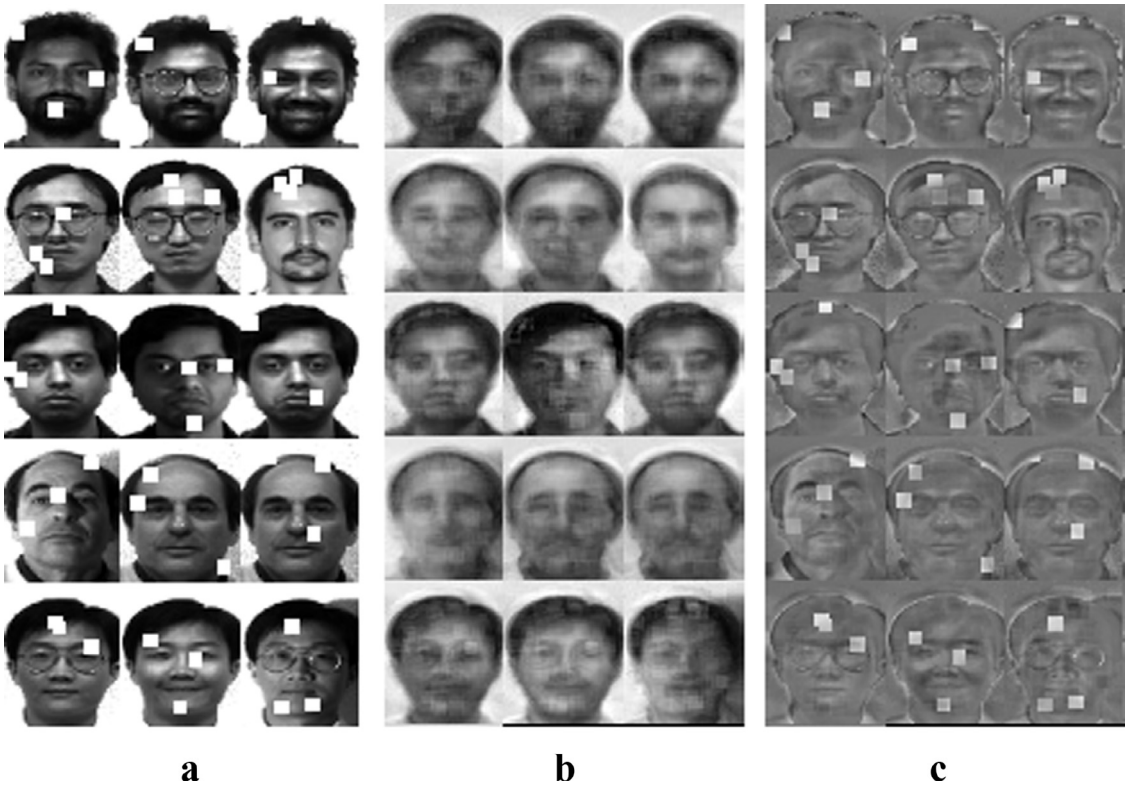

Fig. 4. Experimental results of NMF.

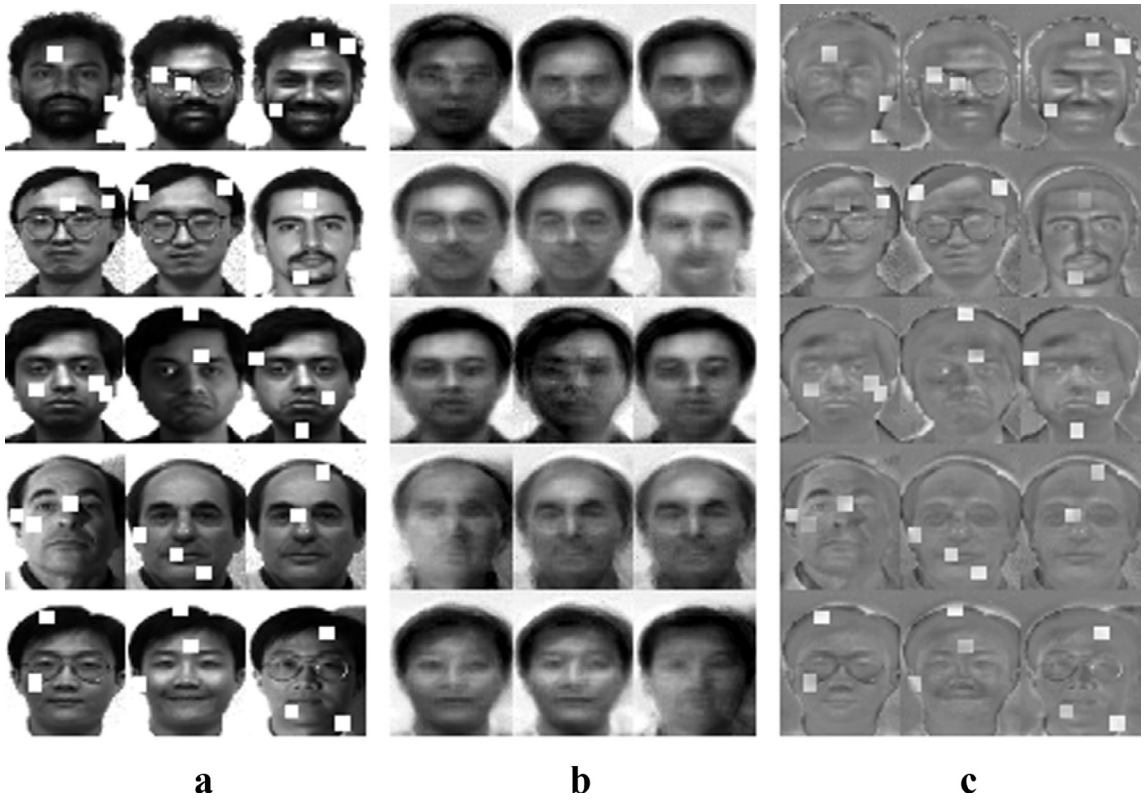

Fig. 5. Experimental results of RNMF.

fact, not all the test samples are corrupted during image reconstruction. The actual situation is that some of the pictures are corrupted and some pictures are non-corrupted. We randomly selected samples in test set to add white block noise. The changes of the recognition accuracy of SCNMF, NMF and RNMF methods on randomly corrupted Yale face database is shown in Fig. 10.

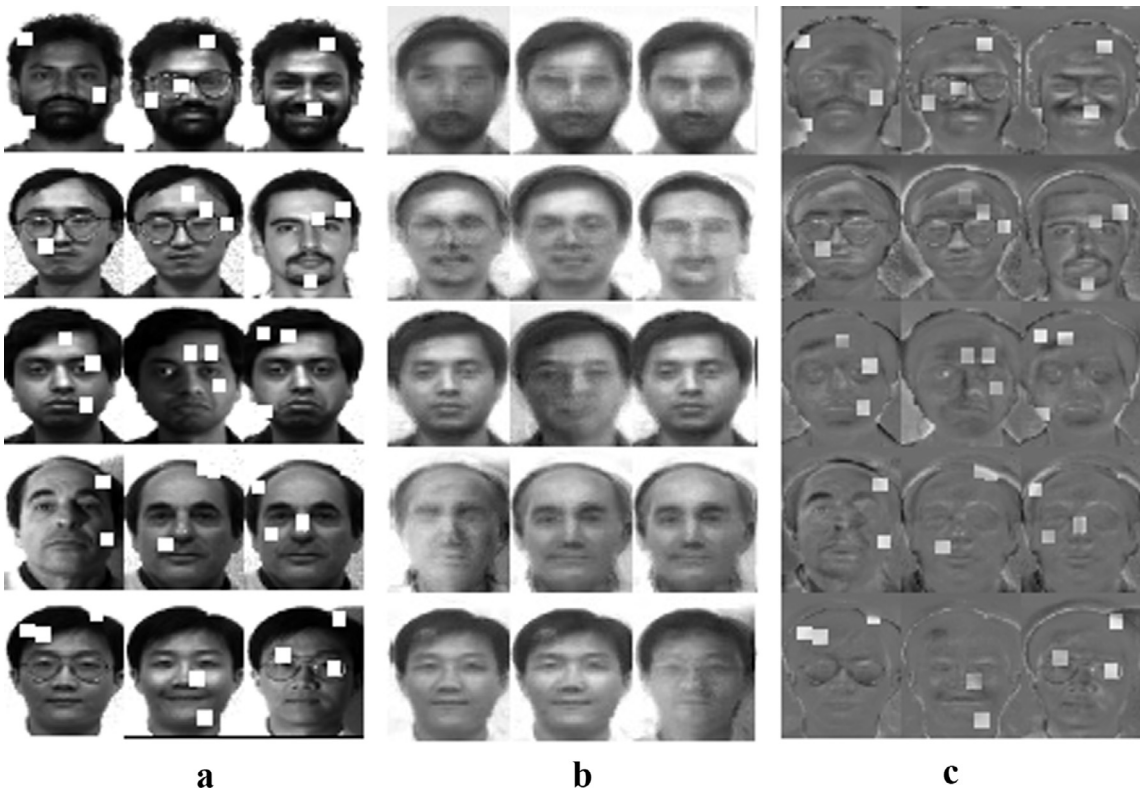

Fig. 6. Experimental results of SCNMF.

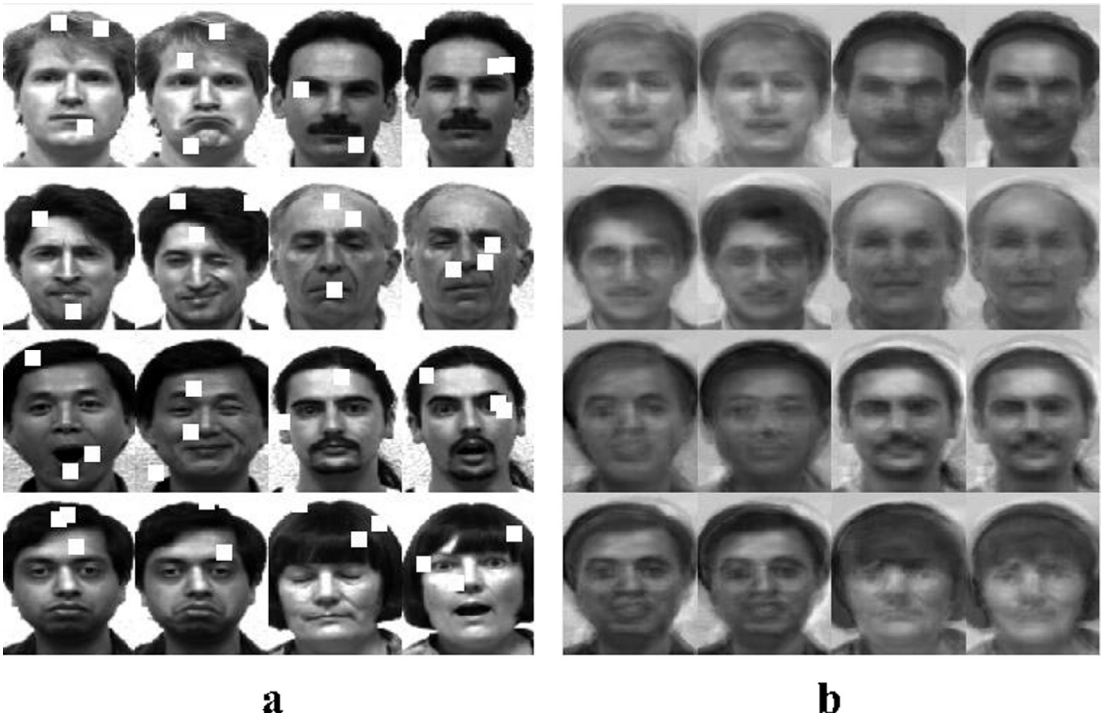

Fig. 7. Reconstructed faces for the corrupted test set.

By analyzing the two line charts, NMF has no tolerance for sparse white block noise so it has the lowest recognition rate in Fig. 10. RNMF can tolerate sparse white block noise, but when the test set is non-corrupted, it does not perform well, as shown in Fig. 9. The proposed method takes both non-corrupted and corrupted data into account during training, so the recognition accuracy of SCNMF is the highest among the three, which confirms the its effectiveness.

Result on corrupted ORL face dataset is shown in Table 2 through comparative analysis of SCNMF, NMF and RNMF. We

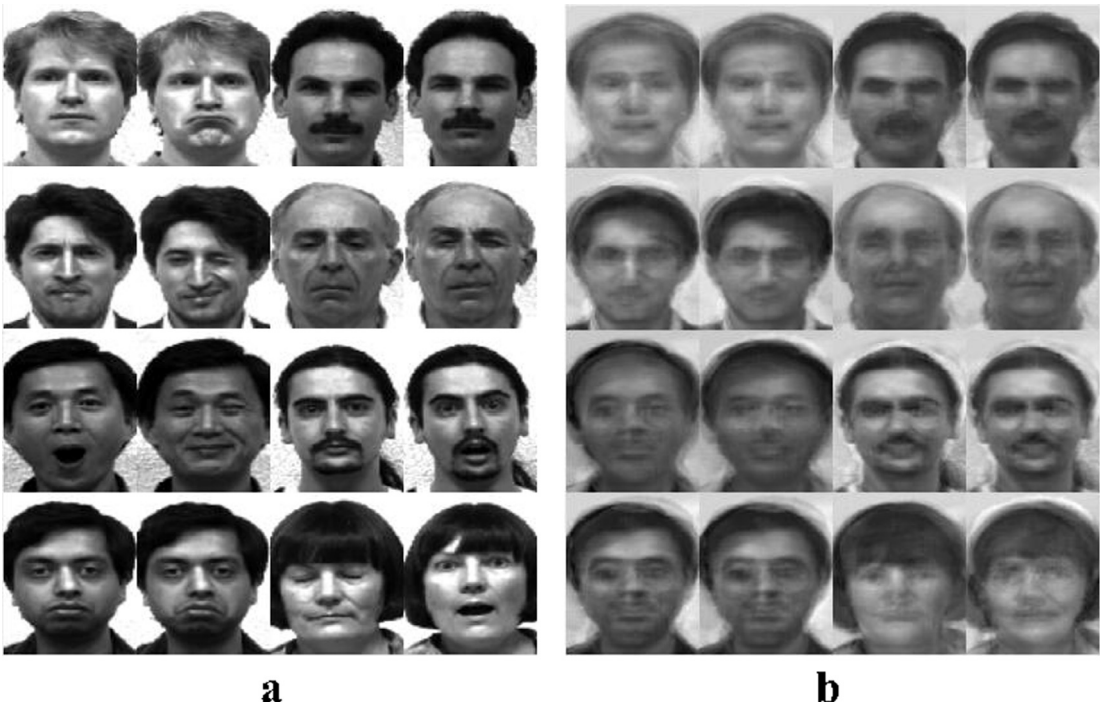

Fig. 8. Reconstructed faces for the non-corrupted test set.

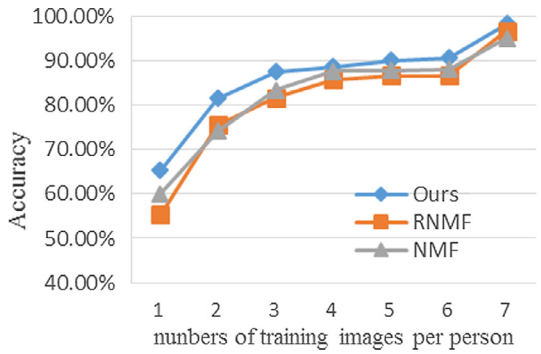

Fig. 9. Performance comparisons on the non-corrupted Yale database with different numbers of training images per person.

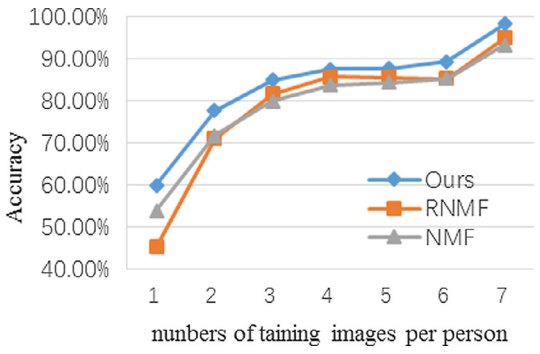

Fig. 10. Performance comparisons on the randomly corrupted Yale database with different numbers of training images per person.

randomly choose $k(k = 1, 2, 3, \cdots, 9)$ images per person as training data, and the rest as test data sets. The corruption is three 10×10 white blocks as before. The data in Table 2 shows that the recognition accuracy increases as the number of training increases and ours is the highest whatever value of k is. The recognition accuracy of the proposed algorithm is slightly higher than that of the contrastive algorithm when the training set is small, but when the training set is gradually expanded, the recognition accuracy of our SCNMF is significantly higher, and the average value can be more than 3 percentage points.

5. Conclusion

A new approach Sparse Corruption Matrix Factorization (SCNMF) is proposed to reduce data dimensions and remove data noise in this paper. The experiments on Yale face database and ORL face database confirm the effectiveness of the proposed

Table 2
Recognition accuracy and standard deviation of SCNMF, RNMF and NMF on the randomly corrupted ORL.

k	1	2	3	4	5	6	7	8	9
SCNMF	56.94 ±3.22	67.81 ±1.31	72.71 ±4.28	78.75 ±2.25	83.00 ±4.70	88.12 ±3.42	91.50 ±3.50	92.50 ±2.01	95.00 ±3.71
RNMF	56.58 ±3.09	65.94 ±1.81	69.64 ±2.97	76.58 ±1.34	80.50 ±2.15	86.25 ±3.75	88.33 ±3.34	89.75 ±1.25	92.00 ±3.50
NMF	50.72 ±2.39	62.38 ±3.62	67.71 ±3.21	71.58 ±1.75	77.50 ±2.36	81.87 ±2.51	84.66 ±3.50	86.25 ±2.75	87.50 ±1.28

method. The SCNMF algorithm can eliminate facial expression and pixel noise corruptions. Regardless of whether the image in the test set is corrupted, the algorithm is effective and has good reconstruction results, which is significant for image processing and image measurement.

There is still one open question that needs to be addressed. In the experiment, we find that the reconstruction of face captured in different illumination conditions is not good, so how to improve the reconstruction and recognition result of these faces is a issue we will investigate in the future.

Acknowledgements

This research was supported by the Prospective Joint Research Project of Jiangsu Province, China (BY201506-01) and National Natural Science Foundation of China (NFSC), grant no. 61503329, China.

Competing interests

The authors declare that they have no competing interests.

References

[1] W. Zhao, R. Chellappa, P.J. Phillips, et al., Face recognition: a literature survey, ACM Comput. Surv. (CSUR) 35 (4) (2003) 399–458.
[2] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2014) 1373–1396.
[3] Y.X. Wang, Y.J. Zhang, Nonnegative matrix factorization: a comprehensive review, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1336–1353.
[4] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A, Opt. Image Sci. 4 (3) (1987) 519–524.
[5] Y. Xia, H. Tong, W.K. Li, et al., An adaptive estimation of dimension reduction space, J. R. Stat. Soc. 64 (3) (2010) 363–410.
[6] J. Yang, D. Zhang, A.F. Frangi, et al., Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 131–137.
[7] Z. Lai, Y. Xu, Q. Chen, et al., Multilinear sparse principal component analysis, IEEE Trans. Neural Netw. Learn. Syst. 25 (10) (2017) 1942–1950.
[8] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data — with application to face recognition, Pattern Recognit. 34 (10) (2001) 2067–2070.
[9] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face recognition using LDA-based algorithms, IEEE Trans. Neural Netw. 14 (1) (2003) 195–200.
[10] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the 2000 Conference Advances in Neural Information Processing Systems (NIPS 2000), 2000, pp. 556–562.
[11] R. Zhi, M. Flierl, Q. Ruan, et al., Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition, IEEE Trans. Syst. Man Cybern. B Cybern. 41 (1) (2011) 38–52.
[12] S. Zafeiriou, A. Tefas, I. Buciu, et al., Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification, IEEE Trans. Neural Netw. 17 (3) (2006) 683–695.
[13] F. Pompili, N. Gillis, P.A. Absil, et al., Two algorithms for orthogonal nonnegative matrix factorization with application to clustering, Neurocomputing 141 (2014) 15–25.
[14] B. Li, G. Zhou, A. Cichocki, Two efficient algorithms for approximately orthogonal nonnegative matrix factorization, IEEE Sig. Process. Lett. 22 (7) (2015) 843–846.
[15] I. Buciu, N. Nikolaidis, I. Pitas, Nonnegative matrix factorization in polynomial feature space, IEEE Trans. Neural Netw. 19 (6) (2008) 1090–1100.
[16] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[17] T. Feng, S.Z. Li, H.Y. Shum, et al., Local non-negative matrix factorization as a visual representation, Icdl (2002) 178–183.
[18] P.O. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (1) (2004) 1457–1469.
[19] A. Cichocki, S.I. Amari, R. Zdunek, et al., Extended SMART algorithms for non-negative matrix factorization, International Conference on Artificial Intelligence & Soft Computing, 2006.
[20] S. Nikitidis, A. Tefas, N. Nikolaidis, et al., Facial expression recognition using clustering discriminant Non-negative Matrix Factorization, in: IEEE International Conference on Image Processing, IEEE, 2011, pp. 3001–3004.
[21] A. Cichocki, R. Zdunek, S.I. Amari, Csiszár's, divergences for non-negative matrix factorization: family of new algorithms, Springer Lncs 3889 (2006) 32–39.
[22] L. Zhang, Z. Chen, M. Zheng, et al., Robust non-negative matrix factorization, Front. Electr. Electron. Eng. 6 (2) (2011) 192–200.
[23] F. Sha, Y. Lin, L.K. Saul, et al., Multiplicative updates for nonnegative quadratic programming, Neural Comput. 19 (8) (2014) 2004–2031.
[24] Z. Wang, A.C. Bovik, H.R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans Image Process. 13 (4) (2004) 600–612.
[25] W.S. Chen, Y. Zhao, B. Pan, et al., Supervised kernel nonnegative matrix factorization for face recognition, Neurocomputing 205.C (2016) 165–181.
[26] Yale University Face Database 2002 ⟨http://cvc.yale.edu/projects/yalefaces/yale-faces.html⟩.
[27] ORL Face Database ⟨https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html⟩.