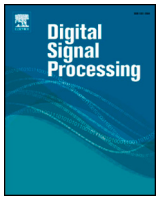
Digit. Signal Process. 145 (2024) 104331

Contents lists available at ScienceDirect

Digital Signal Processing

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



# Neighborhood preserving embedding with autoencoder

Ruisheng Ran [a], Jinping Wang [a], Bin Fang [b], Weiming Yang [a,*]

[a] The College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
[b] The College of Computer Science, Chongqing University, Chongqing 400044, China

ARTICLE INFO

*Keywords:*
Neighborhood preserving embedding
Autoencoder
Dimensionality reduction
Manifold learning

ABSTRACT

Neighborhood preserving embedding (NPE) is a classical method for dimensionality reduction (DR), and it is a linear version of the locally linear embedding method. However, NPE and all its variants only consider the one-way mapping from high-dimensional space to low-dimensional space. The projected low-dimensional data may not accurately and effectively "represent" the original samples. To address this problem, we improve NPE based on linear autoencoder. The conventional projection of NPE is considered as the encoding stage, and the decoder stage is a reconstruction from the low-dimensional space to the original high-dimensional space, which is the key to maintaining more significant information. Based on this, we propose a new NPE method called NPEAE (neighborhood preserving embedding with autoencoder) in this paper. NPEAE performs excellently in face recognition, handwritten character categorization, object classification, etc. The experiments on MNIST, COIL-20, the Extended Yale B, Olivetti Research Laboratory (ORL), and FERET show that NPEAE has a higher recognition accuracy than other comparative methods.

## 1. Introduction

Dimensionality reduction (DR) [1,2] is a popular research subject in machine learning because original data is often high-dimensional. Especially in the field of image processing, such as $m \times n$ images that are commonly converted into $m \times n$-dimensional vectors, resulting in more complex and time-consuming training processes. Redundant information and noise are frequently present in the original data, which can negatively impact certain tasks like image classification [3–6]. In such cases, noise can cause inaccurate model building and classification results. Therefore, many advanced feature extraction methods are proposed to address these issues. DR aims to address the curse of dimensionality [7,8] by identifying optimal representative features or extracting low-dimensional fusion features from high-dimensional data. Especially in deep learning algorithms [9–11], the utilization of data preprocessing methods represented by data dimensionality reduction can make the trained model more stable and also reduce the training time.

DR techniques can be broadly categorized as either linear or nonlinear methods [12]. Linear techniques, such as principal component analysis (PCA) [13,14] and linear discriminant analysis (LDA) [15,16], have been widely used in various applications. PCA, also known as Eigenface in face recognition, was proposed by Turk and Pentland [13]. The main idea behind PCA is to reduce the dimensionality of the data using as few components as possible. On the other hand, LDA, also known as Fisherface in face recognition, was proposed by Fisher in 1936 [17]. However, linear DR methods such as PCA and LDA are not suitable for processing nonlinear data.

Various manifold-based dimensionality reduction methods [18] have been developed to address nonlinear problems. Examples include laplacian eigenmap (LE) [19], locally linear embedding (LLE) [20], and Isomap [21]. LE leverages the Laplacian matrix of graphs to identify an optimal low-dimensional representation that preserves the local neighborhood structure of the original manifold. LLE assumes that each data point and its nearest neighbors form a locally linear patch of the manifold [22].

Manifold-based dimensionality reduction methods have been shown to be effective for nonlinear problems, but they are often not applicable to the test set, leading to the "out-of-sample" problem [23]. To address this issue, linearization methods have been developed. For instance, locality preserving projection (LPP) [24] is a linearization of LE, and neighborhood preserving embedding (NPE) [25] is a linearization of LLE. NPE first constructs an adjacency graph using the $k$-nearest neighbor (KNN) or epsilon-neighborhood method and calculates edge weights. It then maps high-dimensional data into a low-dimensional embedding using a linear mapping that preserves the structure of the

* Corresponding author.
*E-mail address:* 20130956@cqnu.edu.cn (W. Yang).

https://doi.org/10.1016/j.dsp.2023.104331

Available online 23 November 2023

original manifold data points. NPE retains the advantages of LLE and overcomes the limitation of providing an embedding only for the training samples, neighborhood preserving projections (NPP) was proposed [26] whose main idea is similar to NPE is proposed immediately after NPE. The difference between them is only in the way they avoid matrix singularities. Then orthogonal neighborhood preserving projections (ONPP) [27] was proposed as a generalized NPP.

The proposed of the manifold learning not only improves the performance of feature extraction but also enhances the data visualization. [28,29]. However, since the performance of previous manifold learning methods is not satisfactory enough, a series of improved methods have been proposed for visualization. As they are mainly used for visualization, their development direction is focused on how to better maintain the relationship between different data points by reducing the data points to two or three dimensions in order to better show the relationship between them. The representative ones include t-SNE [30], which combines the Student-t distribution with the symmetric version of stochastic neighbor embedding (SNE) [31] to achieve better visualization results. Recently, another dimensionality reduction tool for visualization, uniform manifold approximation and projection (UMAP) [32], has been proposed to solve some problems of t-SNE, such as the crowding problem and long computation time. In addition, kernel t-SNE [33] was proposed to solve the non-linear problem, which makes it applicable to new data.

NPE has many improved methods. The eigenmatrix of NPE is singular in many cases, leading to the difficulty of directly implementing NPE, which is called the small-sample-size (SSS) problem [34]. Usually, the original data is preprocessed by PCA or singular value decomposition (SVD) [35] to avoid the SSS problem, but it also loses some vital information sometimes. Some other methods have also been proposed to solve this problem, for example, exponential neighborhood preserving embedding (ENPE) [34], complete neighborhood preserving embedding (CNPE) [36], and improved CNPE (ICNPE) [37], neighborhood preserving embedding on Grassmann manifold (GNPE) [38]. ENPE introduces the matrix exponential to NPE, and then avoids the SSS problem, and reduces the sensitivity to the neighborhood parameter $k$. CNPE simplifies the objective function of NPE, which computes two easy-solving eigenvalue decompositions instead of a direct singular generalized eigensystem computation. Weighted neighborhood preserving ensemble embedding (WNPEE) [39] is a recently improved method for NPE that constructs an ensemble of adjacent graphs with the number of nearest neighbors varying so that NPE can obtain a more representative low-dimensional projection in a joint optimal manner.

The methods mentioned above can be classified into two categories according to the application scenarios: the DR methods represented by NPE, including its varieties such as CNPE [36] and ENPE [34], are mainly used to remove noise and extract the classification features of the samples, and they generally reduce the dimensions to tens or hundreds of dimensions. As well as methods for visualization represented by t-SNE, including its kernelized version [33], their goal is to reduce the data to 2 or 3 dimensions for visual observation of data relationships. However, the algorithms mentioned above have a common limitation: they only consider a one-way mapping from the high-dimensional space to the low-dimensional space and do not evaluate the accuracy of the low-dimensional embedding. This may result in inaccurate representations of the data in the reduced dimensionality.

Motivated by the encoder-decoder paradigm, we find that the current data dimensionality reduction methods represented by the NPE algorithm only consider a one-way mapping from the high-dimensional space to the low-dimensional space. So the obtained low-dimensional data often cannot accurately represent the original data. Thus, based on the encoder-decoder paradigm, we present a novel NPE method named NPEAE (NPE with Autoencoder) in this paper. Specifically, under the condition of maintaining the neighborhood structure information of the samples, the data points in high-dimensional manifold space are encoded into data points in low-dimensional space by using the conventional NPE projection algorithm. We also consider using the decoder to reconstruct the original high-dimensional data points from the embedded low-dimensional data points. Furthermore, to balance these two items and obtain better performance, the adaptive parameter strategy is applied to them. That is, compared with conventional NPE, the new NPE method has an additional reconstruction stage and a strategy to balance it. This enables the embedded low-dimensional data to retain as much information as possible from the original high-dimensional data, so the embedded low-dimensional data "represents" the original samples more accurately and effectively. The main contribution of this paper can be summarized as follows.

1. A novel NPE method based on an autoencoder named NPEAE is proposed to overcome the one-way mapping problem of the original NPE algorithm. The proposed method not only considers the local structure of the original NPE algorithm but also the constructive features, which makes it perform superior to other methods.
2. An adaptive strategy is applied to the NPEAE algorithm to obtain the optimal balance parameters. The algorithm can perform at its best with the optimal parameters obtained in such a strategy.
3. Extensive experiments were conducted on seven datasets for both classification and visualization tasks, and the results demonstrate the superiority of NPEAE.

The rest of this paper proceeds as follows: in the second section, we review the NPE method and the autoencoder. In the third section, we propose the novel NPE method with the encoder-decoder paradigm. In the fourth section, we conduct experiments to evaluate the new methods. The conclusion and future work are given in the fifth section.

## 2. The related works

### 2.1. Locally linear embedding

NPE is based on locally linear embedding (LLE), let us review the procedure of LLE first. Suppose there are $l$ original samples $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l$ in $\mathbb{R}^n$ space and denote $\boldsymbol{X} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l\right]$. Firstly, construct a neighborhood for each point. In the original high-dimensional input space, the $k$ nearest neighbors of each point are found based on Euclidean distance. Secondly, to calculate the best linear reconstruction weights of the nearest neighbors for each point, let $w_{ij}$ be the weight from node $i$ to node $j$. $w_{ij}$ can be obtained by minimizing the following reconstruction error:

$$\sum_i \left\| \boldsymbol{x}_i - \sum_j w_{ij} \boldsymbol{x}_j \right\|_2^2, \tag{1}$$

with the constraint

$$\sum_j w_{ij} = 1. \tag{2}$$

Thirdly, to find low-dimensional embedding coordinate $\boldsymbol{y}_i$ of the original data point $\boldsymbol{x}_i$, where $\boldsymbol{y}_i$ is the $m$-dimensional ($m << n$) data point, let $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_l]$ be the coordinate matrix, by fixing the linear reconstruction weight matrix $\boldsymbol{W}$, the matrix $\boldsymbol{Y}$ can be obtained by minimizing the reconstruction error of each point in the embedding space:

$$\boldsymbol{Y} = \arg \min_{\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{I}} \sum_i \left\| \boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_j \right\|_2^2, \tag{3}$$

where $\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{I}$ is a constraint imposed to the Eq. (3) to remove an arbitrary scaling factor in the projection.

### 2.2. Neighborhood preserving embedding

However, LLE is a nonlinear dimensionality reduction method and it is only defined on the training dataset. The NPE method is designed to address the above problem. The purpose of the NPE algorithm is to obtain a projection matrix $\boldsymbol{A}$ and use it to project the original high-dimensional data to a low-dimensional space by the following transformation:

$$\boldsymbol{y}_i = \boldsymbol{A}^T \boldsymbol{x}_i \tag{4}$$

From the high-dimensional manifold space into the low-dimensional space, considering the linear mapping (4) in Eq. (3), the objective function of NPE is:

$$\boldsymbol{A}_{opt} = \underset{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{A} = \boldsymbol{I}}{\arg\min} \; tr\left(\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^T \boldsymbol{A}\right), \tag{5}$$

where $\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{A} = \boldsymbol{I}$ is a constraint, $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^T (\boldsymbol{I} - \boldsymbol{W})$. The objective function Eq. (5) can be solved with Lagrange multiplier methods, which reduce Eq. (5) to a generalized eigenvalue problem:

$$\boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^T \boldsymbol{a} = \lambda \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{a}. \tag{6}$$

With the solution of Eq. (6), the low-dimensional embedding coordinate $\boldsymbol{y}_i$ of the original data point $\boldsymbol{x}_i$ can be gotten with Eq. (4).

The NPE algorithm can be considered a linearized variant of the LLE algorithm. This characteristic of NPE allows it to address the limitation of LLE, which is solely available on the training data. In contrast, the NPE algorithm can be employed on both the training and test datasets inside a high-dimensional space, hence enhancing its usability.

### 2.3. Autoencoder

The autoencoder can be categorized into two models based on the number of nodes in the hidden layer compared to that of the input layer: the under-complete model and the over-complete model. When the activation function of the hidden layer is linear, the autoencoder is referred to as a linear autoencoder. The proposed approach in this paper is based on a single hidden layer, linear, under-complete autoencoder.

There are many varieties of autoencoder, which include Stacked Autoencoder [40,41], Sparse Autoencoder [42], Convolutional Autoencoder [43], Variational Autoencoder [44], etc.

## 3. Neighborhood preserving embedding with autoencoder

### 3.1. Framework

In this section, the framework of the new NPE method is presented. As the structure of the encoder-decoder paradigm, the new NPE method can be divided into two stages.

The initial stage of the approach involves the conventional projection model of NPE. In this model, a linear mapping $\boldsymbol{y}_i = \boldsymbol{A}^T \boldsymbol{x}_i$ is used to map high-dimensional data points $\boldsymbol{x}_i$ to low-dimensional data points $\boldsymbol{y}_i$. This mapping can be viewed as encoding each high-dimensional data point into a low-dimensional data point, from the perspective of the linear autoencoder. Furthermore, the mapping preserves the local neighborhood information of the original samples, meaning that if the original data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the $k$-nearest neighborhood, then the embedded points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ will also be in the $k$-nearest neighborhood.

The second stage is a reconstruction process that recovers the low-dimensional embedding point $\boldsymbol{y}_i$ to the original space with the linear decoder. Define $\hat{\boldsymbol{x}}_i$ as the reconstruction of the original data point $\boldsymbol{x}_i$, $\boldsymbol{A}^*$ as the weight matrix of the decoder. Thus, the procedure of decoding can be formulated as follows:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{A}^* \boldsymbol{y}_i. \tag{7}$$

Moreover, the model can be simplified by applying the weights of the encoder and decoder in Ref [45], i.e., $\boldsymbol{A}^* = \left(\boldsymbol{A}^T\right)^T = \boldsymbol{A}$. Thus, Eq. (7) can be rewritten as:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{A} \boldsymbol{y}_i. \tag{8}$$

To preserve more information about the original high-dimensional space, we minimized the error between the original high-dimensional data point $\boldsymbol{x}_i$ and the reconstructed high-dimensional data point $\hat{\boldsymbol{x}}_i$. By considering the error, the low-dimensional embedding is more representative.

The framework of the proposed NPEAE method is shown in Fig. 1.

Based on the above description, the advantages of new NPE can be concluded as follows:

1. Since we continue to use NPE in the first stage, the low-dimensional embedding data still retain the neighborhood relationship between the original high-dimensional data.
2. The low-dimensional embedding data can "represent" the original data more accurately and comprehensively due to the additional process of reconstruction.

The new NPE method is named NPEAE (NPE with Autoencoder) as the NPE method is based on the encoder-decoder paradigm.

### 3.2. The objective function

After the analysis of the framework, the objective function of NPEAE is presented in this section.

As stated above, the first stage of NPEAE is the conventional NPE algorithm, which projects the high-dimensional data to the low-dimensional space, and Eq. (5) is the original formulation of the NPE method. In this paper, the first item of the objective function of NPEAE is the Eq. (9).

$$\mathcal{L}_{1st} = tr\left(\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^T \boldsymbol{A}\right). \tag{9}$$

Besides, the constraint imposed on Eq. (5), i.e., $\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{A} = \boldsymbol{I}$ can be relaxed as:

$$\mathcal{L}_{2nd} = tr\left(\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{A}\right) - m. \tag{10}$$

Eq. (10) is regarded as the second term of the objective function of NPEAE. The parameter $m$ is the dimensionality of the low-dimensional space, and the description is presented in Section 2.1.

The second stage is the decoding stage which preserves the information of the original high-dimensional space. To be specific, NPEAE reconstructs the high-dimensional data points and minimizes the error between the reconstructed high-dimensional data points $\hat{\boldsymbol{x}}_i$ and the original high-dimensional data points $\boldsymbol{x}_i$.

$$\mathcal{L}_{3rd} = \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i \right\|_2^2. \tag{11}$$

The reconstructed data point $\hat{\boldsymbol{x}}_i$ can be denoted as follows by considering the linear mapping.

$$\hat{\boldsymbol{x}}_i = \boldsymbol{A} \boldsymbol{y}_i = \boldsymbol{A} \boldsymbol{A}^T \boldsymbol{x}_i. \tag{12}$$

Then, Eq. (11) can be further presented as follows by substituting Eq. (12):

$$\begin{aligned} \mathcal{L}_{3rd} &= \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \boldsymbol{A} \boldsymbol{A}^T \boldsymbol{x}_i \right\|_2^2 \\ &= \sum_{i=1}^{N} \left\| \left(\boldsymbol{I} - \boldsymbol{A} \boldsymbol{A}^T\right) \boldsymbol{x}_i \right\|_2^2 \\ &= tr\left( \left(\boldsymbol{I} - \boldsymbol{A} \boldsymbol{A}^T\right) \boldsymbol{X} \boldsymbol{X}^T \left(\boldsymbol{I} - \boldsymbol{A} \boldsymbol{A}^T\right)^T \right). \end{aligned} \tag{13}$$

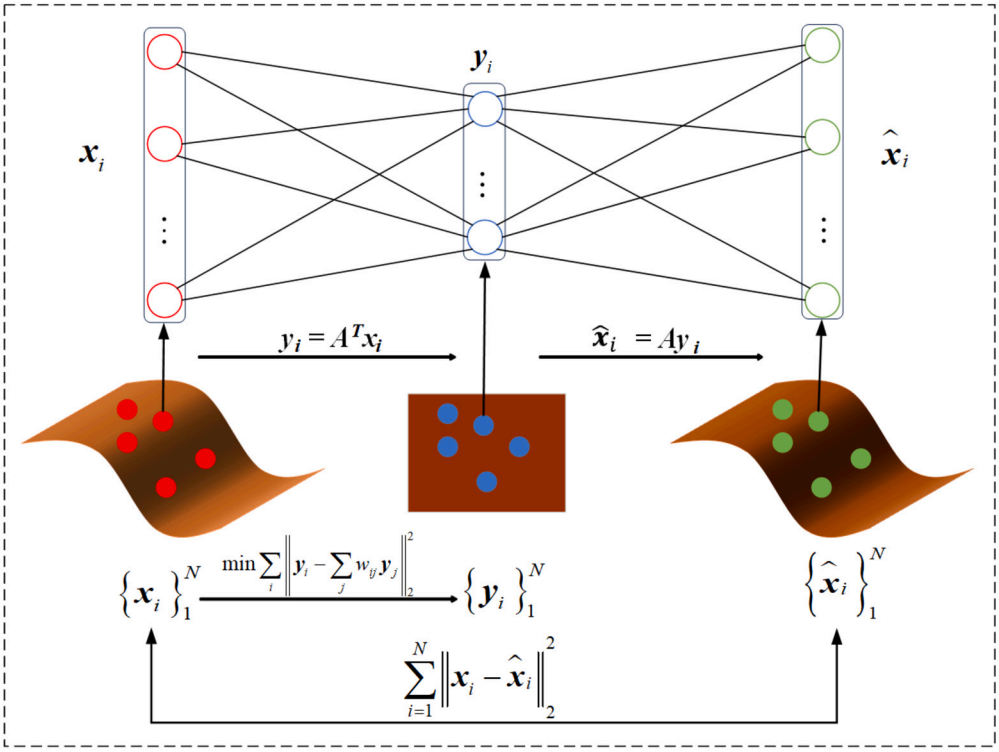



**Fig. 1.** The block diagram of the proposed NPEAE method.

Finally, the objective function of the proposed NPEAE can be obtained by combining the Eq. (9), Eq. (10), and Eq. (13), and the optimal projection matrix $\boldsymbol{A}$ can be obtained by minimizing the following loss function.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{1st} + \lambda\mathcal{L}_{2nd} + \gamma\mathcal{L}_{3rd} \\ &= tr\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T\boldsymbol{A}\right) + \lambda\left(tr\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\right) - m\right) \\ &+ \gamma tr\left(\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T\right)^T\right),\end{aligned} \tag{14}$$

where $\lambda$ and $\gamma$ are the balance parameters. They control the relative importance of the term $\mathcal{L}_{2nd}$ (the term that removes an arbitrary scaling factor) and $\mathcal{L}_{3rd}$ (the term denotes the reconstruction error between the original high-dimensional space and the reconstructed space), allowing for flexible adjustment of optimization focus between different objectives. The details of the parameters can be seen in section 3.4.

To clarify, we did not include $\|\boldsymbol{A}\|_F^2$ regularization in our model because the weights of the encoder and decoder are tied, meaning that $\boldsymbol{A}^* = \boldsymbol{A}$, rendering it unnecessary. This is because if the norm $\|\boldsymbol{A}\|_F^2$ is large, the low-dimensional projection produced by the encoder will have large values. Subsequently, during the decoding stage, the multiplication of the low-dimensional projection with the matrix $\boldsymbol{A}$ produces inferior reconstruction. However, the reconstruction constraints automatically handle the $\|\boldsymbol{A}\|_F^2$ regularization [45].

### 3.3. Optimization

Because the problem of NPE is a linear optimization problem, it can be computed directly by converting it to an eigenvalue problem. But the objective function of NPEAE, Eq. (14), is a combination of three items, which is non-linear. Although it is difficult to be solved directly, we can obtain the optimal projection matrix $\boldsymbol{A}$ with the gradient descent method. As there are three unknown variables: $\boldsymbol{A}$, $\lambda$, and $\gamma$, we used an alternating optimization method to obtain the optimal projection.

The optimization algorithm consists of three main steps and loops through these three steps until the objective function converges:

1. Update the variable $\boldsymbol{A}$ with the fixed variables $\lambda$ and $\gamma$.
The partial derivative of the objective function (14) with respect to the pair $\boldsymbol{A}$ can be calculated by the following equation:

$$\frac{d\mathcal{L}}{d\boldsymbol{A}} = \frac{d\mathcal{L}_{1st}}{d\boldsymbol{A}} + \lambda\frac{d\mathcal{L}_{2nd}}{d\boldsymbol{A}} + \gamma\frac{d\mathcal{L}_{3rd}}{d\boldsymbol{A}}, \tag{15}$$

where,

$$\frac{d\mathcal{L}_{1st}}{d\boldsymbol{A}} = 2\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T\boldsymbol{A},$$
$$\frac{d\mathcal{L}_{2nd}}{d\boldsymbol{A}} = 2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A},$$
$$\frac{d\mathcal{L}_{3rd}}{d\boldsymbol{A}} = -4\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T\right)\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}.$$

Then, with the gradient (15), update the matrix $\boldsymbol{A}$ using the following formula

$$\boldsymbol{A}_{t+1} = \boldsymbol{A}_t - \alpha\frac{d\mathcal{L}}{d\boldsymbol{A}_t}, \tag{16}$$

where the parameter $\alpha$ is the searching step size toward the gradient descend direction.

2. Update the variable $\lambda$ with the fixed variables $\gamma$ and $\boldsymbol{A}$.
The partial derivative of the objective function with respect to the pair $\lambda$ is calculated by the following equation:

$$\frac{d\mathcal{L}}{d\lambda} = tr\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\right) - m. \tag{17}$$

Then, $\lambda$ can be updated by following formula with the gradient calculated by (17):

$$\lambda_{t+1} = \lambda_t - \alpha\frac{d\mathcal{L}}{d\lambda_t} \tag{18}$$

3. Update the variable $\gamma$ with the fixed variables $\lambda$ and $\boldsymbol{A}$.
The partial derivative of the objective function with respect to B is calculated according to the following equation:

$$\frac{d\mathcal{L}}{d\boldsymbol{\gamma}} = tr\left(\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T\right)^T\right). \tag{19}$$

Then, $\gamma$ is updated by following formula with the gradient (19):

$$\gamma_{t+1} = \gamma_t - \alpha \frac{d\mathcal{L}}{d\gamma_t}. \tag{20}$$

The algorithm of NPEAE is expressed as follows.

**Algorithm 1** Gradient descent method for training the optimal matrix $\boldsymbol{A}$.

**Input:** the data matrix $\boldsymbol{X}$

**Initialize:** initialize $\boldsymbol{A}$, $\lambda$, $\gamma$ randomly, set a step size $\alpha$ of gradient descent, and set a threshold $\epsilon$ for the change of loss function.

**Pre-calculation:** calculate the weight matrix $\boldsymbol{W}$, the matrix $\boldsymbol{M}$, and the objective function $\mathcal{L}$ using Eq. (14).

**Repeat:**

1. Calculate the gradient $d\mathcal{L}/d\boldsymbol{A}$ using Eq. (15)
2. Update the matrix $\boldsymbol{A}$ using Eq. (16)
3. Calculate the gradient $d\mathcal{L}/d\lambda$ using Eq. (17)
4. Update $\lambda$ using Eq. (18)
5. Calculate the gradient $d\mathcal{L}/d\gamma$ using Eq. (19)
6. Update the matrix $\gamma$ using Eq. (20)
7. Calculate the objective function $\mathcal{L}$ using Eq. (14).

**Until** the change of objective function $< \epsilon$

**Output:** the projection matrix $\boldsymbol{A}$

The reconstruction stage is just considered a constraint to the whole computation process to make the embedding more representative. So, when we obtain the optimal projection matrix $\boldsymbol{A}$, the data point $\boldsymbol{x}_i$ in high-dimensional manifold space can be embedded as the data point $\boldsymbol{y}_i$ in low-dimensional space with the linear mapping: $\boldsymbol{y}_i = A^T \boldsymbol{x}_i$, and the low-dimensional embedded point $\boldsymbol{y}_i$ "represents" the high-dimensional data point $\boldsymbol{x}_i$.

### 3.4. The parameter analysis

There are two free parameters contained in the objective function of NPEAE: $\lambda$ and $\gamma$ (see Eq. (14)).

In Eq. (14), if $\gamma = 0$, Eq. (14) is reduced to Eq. (21)

$$\mathcal{L} = tr\left(\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^T \boldsymbol{A}\right) + \lambda\left(tr\left(\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{A}\right) - m\right). \tag{21}$$

Eq. (21) is the expression by using the Lagrange multiplier method to Eq. (5). According to Eq. (5) and Eq. (6), NPE is to seek out a series of eigenvectors that correspond to the smallest eigenvalues. Therefore, the parameter $\lambda$ is supposed to take small values. Empirically, the initial value of the parameter $\lambda$ is obtained randomly from the range [0-3].

The first term in Eq. (14), denoted by $\mathcal{L}_{1st}$, can be viewed as a term that preserves local structural information. Its coefficient can be set to 1. On the other hand, the third term $\mathcal{L}_{3rd}$ can be seen as a reconstruction term, with the parameter $\gamma$ indicating the relative importance of the reconstruction. If $0 < \gamma < 1$, the reconstruction term is less important than the local structural information preserving term, while if $\gamma \geq 1$, the reconstruction term is equally or more important than the local structural information preserving term. Empirically, the initial value of the parameter $\lambda$ is obtained randomly from the range [0-5].

## 4. Experiments

In this section, we conduct experiments using several datasets, including MNIST, COIL-20 object, Extended Yale B, Olivetti Research Laboratory (ORL) face datasets, and Face Recognition Technology (FERET) datasets. For all datasets, the image format is grayscale, i.e., it contains only one channel and each pixel point is between 0 and 255.

The proposed NPEAE method is compared with other methods, such as LPP, NPP, NPE, ONPP, CNPE, ENPE, and WNPEE.

PCA is utilized in the LPP and NPE methods to prevent singular problems. During this step, 98% of the principal components are retained for all datasets.

The goal of the classification task is to assign the right class label to a sample using the obtained low-dimensional features. For example, in the case of the face dataset, the aim is to determine exactly which person that sample is by using the low-dimensional features.

The nearest neighbor (NN) classifier is employed in the experiments. The basic idea behind the NN classifier is to classify a test data point by comparing it with the labeled samples in a training dataset, identifying the nearest neighbor based on a chosen distance metric, and then assigning the class label of its nearest neighbor to the test data point. Euclidean distance was chosen as the measure of distance between samples in this experiment.

The evaluation indicator used is recognition accuracy, which is represented as the ratio of correctly classified samples to all samples. The definition of accuracy in the experiment can be described as:

$$ACC = \frac{N^{CC}}{N}$$

Where $N$ is the total number of classified subjects, $N^{CC}$ is the total number of correctly classified subjects, and $ACC$ is the accuracy.

The experiments follow a five-step procedure. Firstly, we randomly select $p$ samples from each subject for the training set and use the rest for testing. Secondly, we fix $p$ and vary the reduced dimensions from 10 to 100 in step of 5. Thirdly, we fix $p$ and the reduced dimension and vary the neighborhood size $k$ from 5 to 25 in step of 5. Fourthly, we use the recognition accuracy corresponding to the best $k$ value as the recognition accuracy for the current $p$ and reduced dimension. Finally, we repeat this process for 10 cycles, obtaining 10 recognition accuracy for each subspace dimension. We calculate the average recognition accuracy for each subspace dimension and $p$, and for training samples $p$, the best recognition accuracy and the corresponding reduced dimension are viewed as the final result.

### 4.1. Experimental datasets

MNIST dataset [46]. There are 10 classes in the dataset, and every class contains 10 digits from '0' to '9'. 5000 images are chosen, and every image is 28×28 pixels in the experiment.

COIL-20 object dataset [47]. The Columbia University Image Library (COIL-20) contains 20 objects. There are 72 samples of each object class which are taken from 0° to 360° by the step of 5°. In the experiment, the image is 32×32 pixels.

The Extended Yale B [48] contains 39 human subjects, and each subject consists of 64 image samples. The image of everyone is taken from 9 postures and 64 illumination conditions. The images are cropped and resized into 24×24 and aligned based on the eye coordinates.

ORL face dataset contains 40 distinct individuals, and there are 10 different images of each individual. The images were taken at different times, varying the lighting, facial expressions, etc. All images are resized into 32×32 pixels in the experiment.

FERET dataset [49] was made by the Army Research Laboratory. There are up to 200 human subjects and each person contains 7 images. Notably, the images of each person were taken at multiple times. In the experiment, the size of the images is also adjusted to 32×32 pixels.

Some sample images of the MNIST, COIL-20, the Extended Yale B, ORL, and, FERET are shown in the following Fig. 2. In detail, the first ten samples of the first class from each dataset (if sufficient) were used for the examples.

### 4.2. Experimental results and analysis

#### 4.2.1. Classification task

Tables 1–5 report the best recognition accuracy, standard deviation, and optimal subspace dimension for the eight aforementioned methods across the five datasets.

Based on the above experimental results, the recognition results of each method in different reduced dimensions are shown in Figs. 3–7.

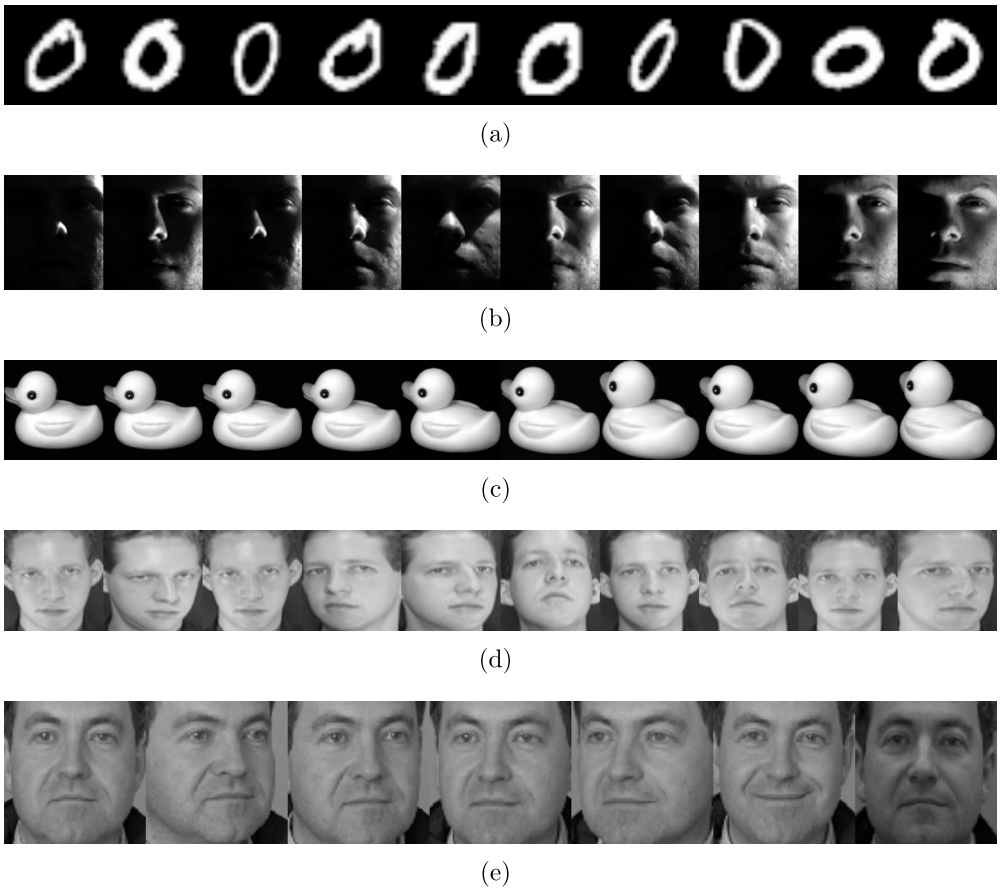



**Fig. 2.** Some sample images of the five datasets used in the experiments (a) MNIST (b) the Extended Yale B (c) COIL-20 (d) ORL (e) FERET.

**Table 1**
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses)on MNIST dataset.

| Method | 30 trains | 40 trains | 50 trains |
|---|---|---|---|
| LPP | 76.54±0.88(25) | 78.84±0.73(40) | 80.83±0.82(30) |
| NPP | 77.93±0.73(40) | 79.29±0.91(60) | 81.76±0.73(50) |
| NPE | 70.84±1.35(75) | 72.42±1.03(55) | 75.51±0.88(70) |
| ONPP | 82.20±0.92(45) | 83.04±0.56(25) | 84.91±0.62(40) |
| CNPE | 75.27±1.10(45) | 76.91±0.82(50) | 79.42±0.90(55) |
| ENPE | 71.26±0.72(60) | 73.71±0.62(70) | 77.05±0.63(85) |
| WNPEE | 78.69±0.72(50) | 82.29±0.81(35) | 83.29±0.46(45) |
| **NPEAE** | **85.30±0.52(30)** | **87.18±0.49(40)** | **88.57±0.53(25)** |

**Table 2**
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses)on COIL-20 dataset.

| Method | 6 trains | 8 trains | 10 trains |
|---|---|---|---|
| LPP | 76.50±4.33(40) | 78.89±2.65(70) | 81.62±1.67(60) |
| NPP | 75.32±2.10(65) | 76.90±2.82(75) | 80.83±1.99(80) |
| NPE | 84.47±2.20(65) | 87.07±2.93(100) | 88.03±1.95(100) |
| ONPP | 80.52±1.50(80) | 83.00±1.22(65) | 85.47±1.63(70) |
| CNPE | 86.91±2.56(75) | 89.10±2.77(100) | 90.04±1.49(90) |
| ENPE | 85.62±1.91(70) | 87.91±2.28(100) | 88.81±4.81(90) |
| WNPEE | 87.47±2.22(50) | 90.27±4.90(40) | 92.10±1.30(50) |
| **NPEAE** | **91.61±2.89(40)** | **93.36±2.37(70)** | **96.21±2.70(80)** |

**Table 3**
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses)on the Extended Yale B face dataset.

| Method | 10 trains | 15 trains | 20 trains |
|---|---|---|---|
| LPP | 82.43±2.97(100) | 86.56±1.98(100) | 87.16±2.90(95) |
| NPP | 81.75±2.21(100) | 84.82±2.30(100) | 86.53±2.94(90) |
| NPE | 76.00±3.58(100) | 77.00±4.73(100) | 81.12±4.88(100) |
| ONPP | 85.35±4.72(100) | 87.25±3.80(100) | 83.35±1.93(90) |
| CNPE | 78.83±2.62(100) | 79.16±2.13(95) | 75.87±2.38(95) |
| ENPE | 76.92±2.01(100) | 78.61±1.76(90) | 83.32±3.18(80) |
| WNPEE | 82.35±3.26(80) | 82.37±4.24(65) | 85.91±1.37(95) |
| **NPEAE** | **87.52±4.82(65)** | **90.14±2.15(90)** | **92.26±1.9(90)** |

**Table 4**
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the ORL face dataset.

| Method | 4 trains | 5 trains | 6 trains |
|---|---|---|---|
| LPP | 83.29±2.33(75) | 84.40±1.84(100) | 86.25±3.35(90) |
| NPP | 83.89±1.71(80) | 85.20±2.88(100) | 87.07±2.00(80) |
| NPE | 85.00±2.67(70) | 90.15±1.68(95) | 90.68±1.75(90) |
| ONPP | 88.74±3.53(90) | 90.17±0.55(100) | 92.32±0.74(85) |
| CNPE | 89.20±3.46(50) | 91.40±1.42(55) | 92.09±0.70(60) |
| ENPE | 88.47±3.40(85) | 90.73±1.38(100) | 91.22±0.87(85) |
| WNPEE | 89.15±1.96(65) | 91.82±3.23(55) | 93.00±1.59(55) |
| **NPEAE** | **90.83±2.41(65)** | **93.90±1.46(60)** | **96.12±0.87(60)** |

The experimental results have shown that NPEAE, as an improved method of NPE, exhibits significantly enhanced performance on all datasets. Specifically, NPEAE achieves recognition accuracy improvements of approximately 13.06, 8.18, 11.14, 5.44, and 12.35% for MNIST, COIL-20, the Extended Yale B, ORL, and FERET, respectively, compared to the original NPE method. This improvement can be attributed to the additional reconstruction item introduced in NPEAE, which helps to capture more information from the high-dimensional space.

**Table 5**
Best average recognition accuracy (in percent), standard deviation, and the optimal dimension (in parentheses) on the FERET face dataset.

| Method | 3 trains | 4 trains | 5 trains |
|---|---|---|---|
| LPP | 46.20±8.24(70) | 50.75±8.38(75) | 59.92±3.54(70) |
| NPP | 45.03±7.14(80) | 48.82±6.52(75) | 58.53±4.08(75) |
| NPE | 43.12±8.56(90) | 46.43±9.12(90) | 56.90±8.94(95) |
| ONPP | 49.25±3.81(90) | 53.11±2.87(90) | 62.91±2.19(85) |
| CNPE | 45.37±6.2(100) | 49.21±7.23(95) | 58.91±5.61(95) |
| ENPE | 45.01±4.15(95) | 46.92±4.06(90) | 57.50±3.74(95) |
| WNPEE | 49.63±6.91(90) | 54.89±8.47(80) | 62.61±5.17(85) |
| **NPEAE** | **59.05±4.47(75)** | **67.03±1.87(90)** | **69.25±1.15(70)** |

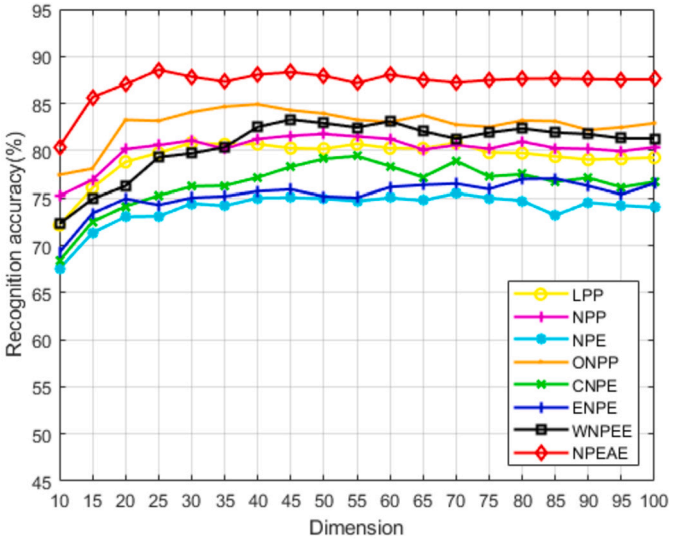



**Fig. 3.** The recognition accuracy versus the subspace dimension on MNIST data set (50 training samples).

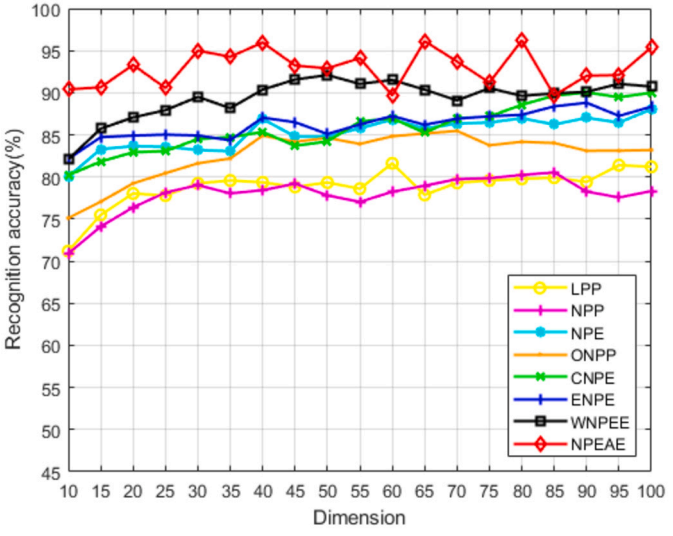



**Fig. 4.** The recognition accuracy versus the subspace dimension on COIL-20 data set (10 training samples).

Furthermore, we compared the performance of NPEAE with some other manifold learning methods, such as LPP, NPP, NPE, ONPP, CNPE, and WNPEE. Our results indicate that NPEAE outperforms all other methods in terms of recognition accuracy, which demonstrates the effectiveness of our proposed approach.

Finally, we analyzed the performance of NPEAE across different subspace dimensions, as shown in Fig. 3-Fig. 7. Our results demonstrate

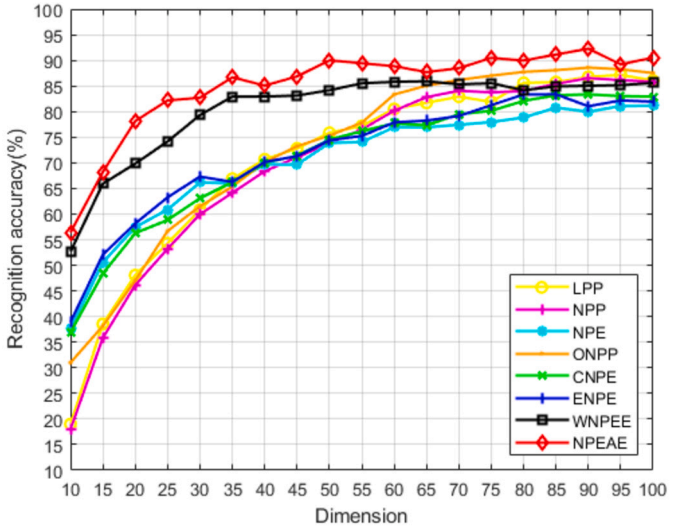



**Fig. 5.** The recognition accuracy versus the subspace dimension on the Extended Yale B (20 training samples).

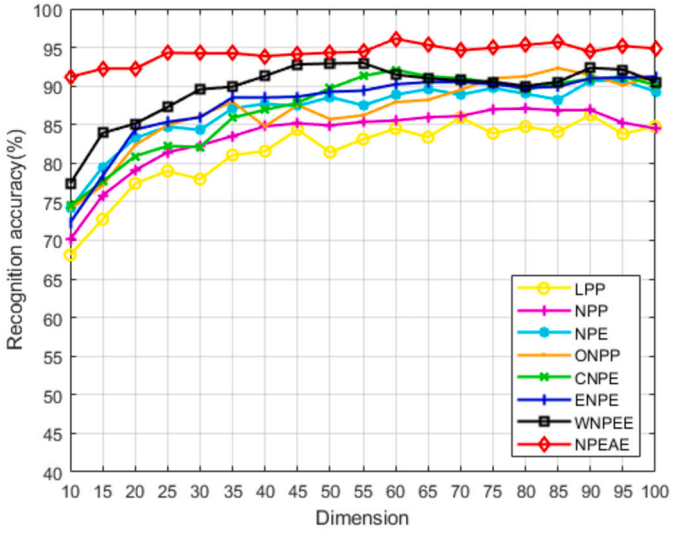



**Fig. 6.** The recognition accuracy versus the subspace dimension on the ORL face dataset (6 training samples).

that NPEAE consistently outperforms other methods across all subspace dimensions, providing strong evidence for its superior performance in subspace learning.

*4.2.2. 2-D visualization task*

In order to better analyze the effect of data dimensionality reduction, we perform visualization experiments in this section. Specifically, data points are projected into a two-dimensional space and then plotted out, which is also called the scatter diagram. Since the scatter diagram can visualize the position of each data point, it can evaluate the feature extraction ability of the algorithm to some extent. In this experiment, two different types of data sets are selected. The first one is the classical UCI Iris data set, which contains 150 records in 3 classes, 50 records in each class. Each record has 4 characteristics: calyx length, calyx width, petal length, and petal width, which can be used to predict which species of iris flowers belong to which species: iris setosa, iris versicolor, and iris virginica. The other is the Columbia Object Image Library (COIL-100) Dataset [50], which contains 7200 color images of 100 objects. A fixed color camera took images of each object as it was rotated on a turntable in 5 degree increments until it reached 360 degrees. This resulted in 72 images per object, with varying poses, that

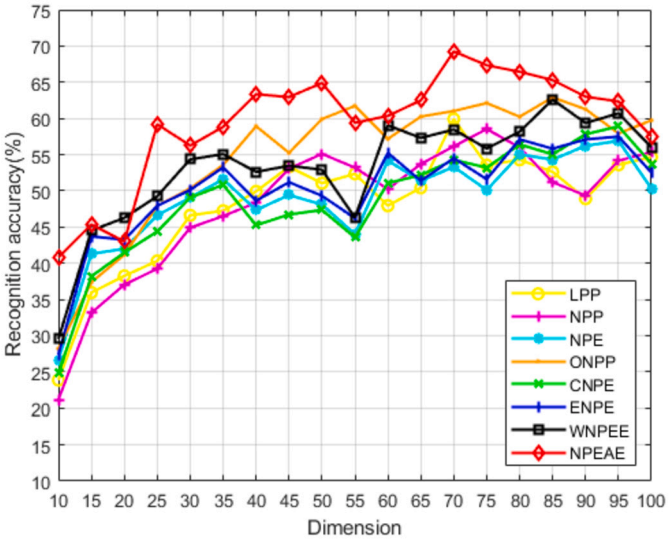



**Fig. 7.** The recognition accuracy versus the subspace dimension on the FERET face dataset (5 training samples).

were then normalized in size. In the following experiment, only the first 5 classes of samples are selected to visualize the results more clearly.

Four algorithms: NPE, ENPE, t-SNE, and the proposed NPEAE are compared. The nearest parameter is set to 10. For the linear algorithms: NPE, ENPE, and NPEAE, we use 10 and 5 training samples for the two data sets, respectively, and the rest as test samples. For t-SNE, a non-linear algorithm, only the test sets are directly projected into two-dimensional space.

The visualization results are shown in Fig. 8 and Fig. 9.

From the above figures, it can be concluded that NPEAE performs well. Specially, on the Iris data set, the data points in both the NPE and ENPE diagrams are comparatively separated and not well clustered in a smaller area. t-SNE and NPEAE all perform well. The distances of the different classes are far enough apart and the data points of the same class are distributed in a smaller area. However, it is obvious to notice that NPEAE is still much more effective than t-SNE. As the distance between data points of the same category is much smaller, which makes it easier to classify correctly during the classification task. In the case of the COIL-100 data set, when compared with that of NPE and ENPE, it is evident that NEPAE exhibits much superior performance. As the samples of the same class were aggregated more compactly. This also means that the feature ability of NPEAE is stronger, which also confirms the results of the classification task above. The green and red data points are hard to classify accurately, despite t-SNE and NPE-based methods adopting two different approaches. t-SNE disperses samples of the same classes into different regions as if they were different classes. As for NPE-based methods, the data points in green and red are distributed in a long strip of area. Overall, both results are not optimal from a visualization perspective. t-SNE does not perform well in putting the same class into the same region, while NPEAE does not separate the two classes of data enough, although it maintains the neighborhood of data points in the same class.

### 4.3. Neighborhood parameter analysis

To evaluate the proposed NPEAE algorithm more comprehensively, we also investigated the influence of the neighborhood parameter $k$ on the algorithm. In detail, we tested the algorithm on five datasets in the classification task, where the target dimension was set to 40. The neighborhood parameter $k$ is taken from 2 to 30 with the step of 2. Five experiments were conducted and their mean values were taken as final results. The test results are shown in Fig. 10

As can be seen in Fig. 10, the proposed NPEAE is not sensitive to the neighborhood parameter $k$, which indicates that it is more robust. Specifically, only on the FERET and ORL datasets, there is a small decrease in the accuracy when the value of k is less than 5. In other cases, the change in accuracy is more stable without large fluctuations. That is, there is no need to determine the accurate neighborhood parameter $k$ before the application, which greatly increases the ease of use of the algorithm.

### 4.4. Convergence analysis

The convergence of the NPEAE method is examined, with Fig. 11 illustrating the convergence curves of NPEAE across five datasets. The datasets consist of MNIST, COIL-20, the Extended Yale B, ORL, and FERET, with training sample sizes of 50, 10, 20, 6, and 5, respectively. The neighborhood size parameter is set to $k = 10$, and the subspace dimension is 40. The convergence plots in Fig. 11 demonstrate that NPEAE can converge in just 30 steps across all five datasets.

## 5. Discussion

In the experimental section, we conducted four experiments on a total of seven datasets. Specifically, we tested the classification task on the MNIST, the Extended Yale B, COIL-20, ORL, and FERET datasets. Then the visualization task was performed on the Iris and COIL-100 datasets. Finally, we perform test of the neighborhood parameter $k$, and convergence analysis based on the classification task. From the above experimental results, we can draw the following conclusions.

First, in the classification task, our method not only performs better than the original NPE algorithm but also performs better than other improved algorithms for NPE. For example, at 5 training samples on the FERET dataset, NPEAE outperforms NPE by 12.35% and outperforms its improved methods, CNPE, ENPE, and WNPEE by 10.34, 11.75, 6.65%. From different datasets, NPEAE outperforms the original NPE by 13.06, 8.18, 11.14, 5.44, and 12.35% in the case of maximum test samples for MNIST, the Extended Yale B, COIL-2, ORL, and FERET, respectively. The experimental results fully demonstrate the superiority of the NPEAE algorithm and confirm that our proposed consideration of bidirectional mapping can make the low-dimensional data more accurate.

Second, in the visualization experiments, NPEAE has a stronger clustering ability compared to the original NPE algorithm. That is, in the low-dimensional space obtained by NPEAE, sample points of the same class are closer together, while sample points of different classes are isolated further away. In addition, we also compare other methods such as ENPE and t-SNE, it is worth noting that NPEAE still achieves superior results even when comparing the t-SNE algorithm, which is specifically designed for visualization.

Thirdly, we also tested the effect of the neighborhood parameter $k$ on the proposed NPEAE. By testing on the above five datasets, we found that the NPEAE is insensitive to changes in the parameter $k$. That is, there is no need to determine the exact neighborhood parameter $k$ before application, which greatly increases the ease of use of the algorithm.

Finally, we perform a convergence analysis based on the classification task. We perform convergence tests on the five previous datasets. From the results, we can conclude that the NPEAE method converges very fast and almost always converges to a relatively low value within 30 steps. This also shows the superiority of our proposed NPEAE algorithm.

Therefore, the superiority of the proposed NPEAE algorithm can be concluded from the above discussion. That is, it excels in the classification task, the visualization task, and convergence.

## 6. Conclusion

In this paper, a novel NPE method called NPEAE (neighborhood preserving embedding with autoencoder) based on the encoder-decoder

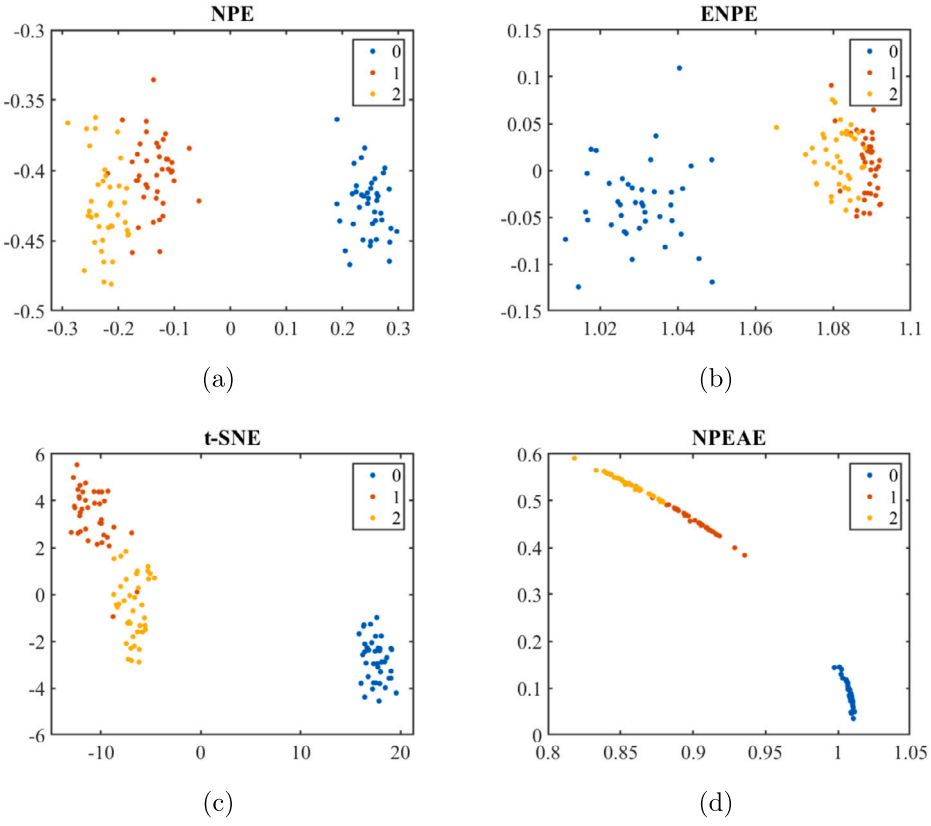



**Fig. 8.** The 2D projections of Iris. (a) NPE (b) ENPE (c) t-SNE (d) NPEAE.

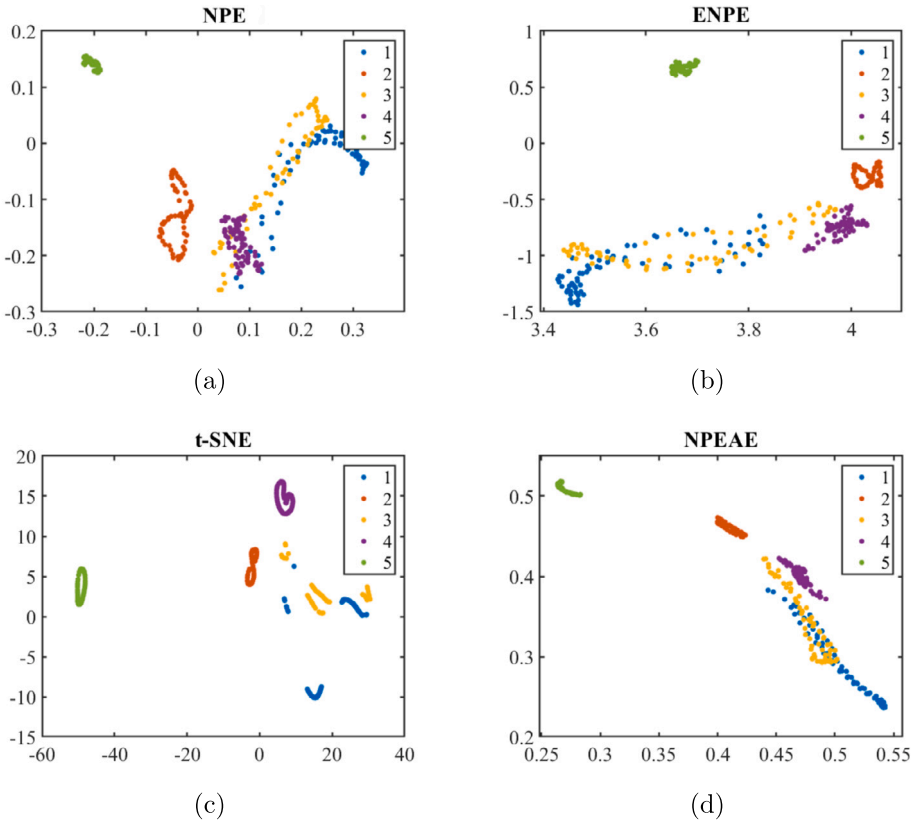



**Fig. 9.** The 2D projections of COIL-100. (a) NPE (b) ENPE (c) t-SNE (d) NPEAE.

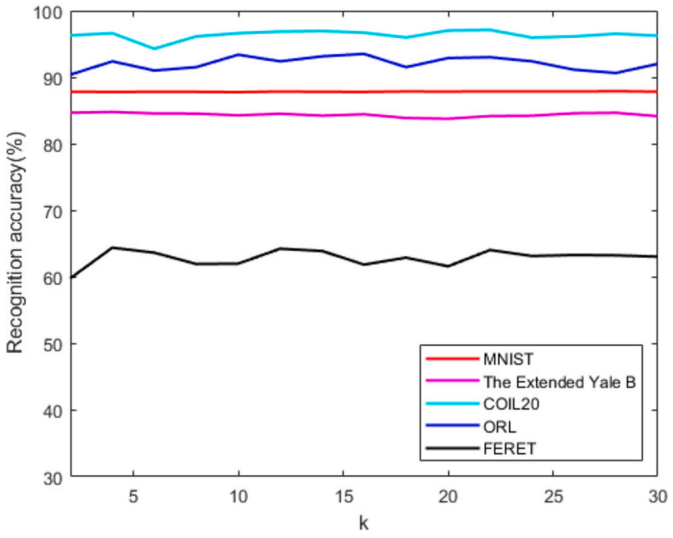



**Fig. 10.** The recognition accuracy of the NPEAE versus the parameter $k$.

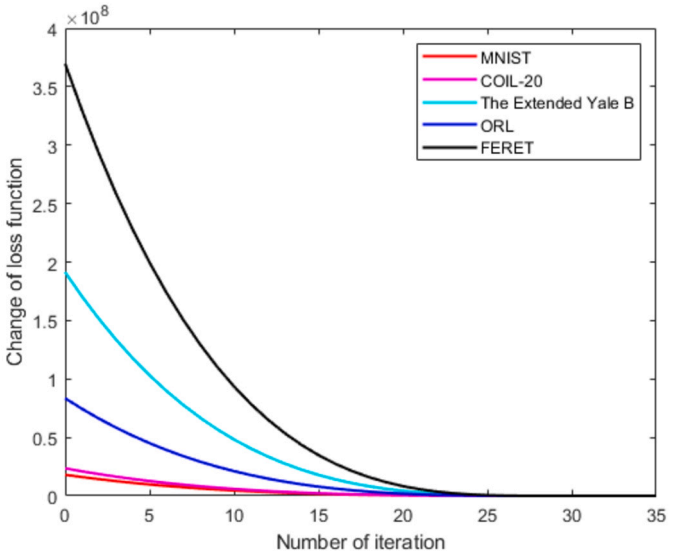



**Fig. 11.** The convergence curve of NPEAE.

paradigm of autoencoder has been proposed. The main idea of NPEAE is to take the conventional projection of NPE as the encoding stage and use the decoder to reconstruct the original data from the projected low-dimensional data. Also, we added adaptive parameters to both items to balance the two parts, which allows the model to automatically adapt to different scenarios for better performance. Compared with NPE, NPEAE makes the low-dimensional features "represent" the original high-dimensional data more accurately and effectively because of the reconstruction stage. The experiments on MNIST, COIL-20, the Extended Yale B, ORL, and FERET show that NPEAE has higher recognition accuracy than other comparative methods. Furthermore, NPEAE has much better performance in visualization experiments compared to the original NPE method and other improved methods.

The idea of NPEAE can be used as a general framework and then extended to other manifold-based dimensionality reduction methods (for example, Locality Preserving Projections (LPP), Isomap Projection (IsoP), linear LTSA (LLTSA) [51], t-SNE, etc.). In addition, the proposed method can also be combined with other feature extraction methods such as wavelet energy methods [52,53] which are excellent feature extraction method to describe the texture of objects in images to achieve better results.

## CRediT authorship contribution statement

**Ruisheng Ran:** Conceptualization, Methodology, Software. **Jinping Wang:** Software, Validation, Writing – original draft. **Bin Fang:** Resources, Supervision, Writing – review & editing. **Weiming Yang:** Validation, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This work was supported by Chongqing Higher Education Teaching Reform Research Project "Research and Practice on the Reform of Software Talent Training Mode for Cross regional Collaborative Innovation" under grant 233178, Humanities and Social Sciences Project of the Ministry of Education of China under grant 20YJAZH084, Chongqing Technology Innovation and Application Development Project under Grant cstc2020jscx-msxmX0190 and cstc2019jscxmbdxX0061, the Key Project for Science and Technology Research Program of Chongqing Municipal Education Commission under grant KJZD-K202100505.

## References

[1] W. Jia, M. Sun, J. Lian, S. Hou, Feature dimensionality reduction: a review, Complex Intell. Syst. (2022) 1–31.
[2] N. Ahmad, A.B. Nassif, Dimensionality Reduction: Challenges and Solutions, ITM Web of Conferences, vol. 43, EDP Sciences, 2022, p. 01017.
[3] N. Aghaei, G. Akbarizadeh, A. Kosarian, Osdes_net: oil spill detection based on efficient_shuffle network using synthetic aperture radar imagery, Geocarto Int. 37 (2022) 13539–13560.
[4] N. Davari, G. Akbarizadeh, E. Mashhour, Corona detection and power equipment classification based on googlenet-alexnet: an accurate and intelligent defect detection model based on deep learning for power distribution lines, IEEE Trans. Power Deliv. 37 (2021) 2766–2774.
[5] N. Aghaei, G. Akbarizadeh, A. Kosarian, Greywolflsm: an accurate oil spill detection method based on level set method from synthetic aperture radar imagery, Eur. J. Remote Sen. 55 (2022) 181–198.
[6] F.M. Ghara, S.B. Shokouhi, G. Akbarizadeh, A new technique for segmentation of the oil spills from synthetic-aperture radar images using convolutional neural network, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 15 (2022) 8834–8844.
[7] K. Kumar, S. Das, S. Bhaumik, A new method for nonlinear state estimation problem, Digit. Signal Process. 132 (2022) 103788.
[8] F. Abboud, M. Stamm, E. Chouzenoux, J.-C. Pesquet, H. Talbot, Distributed algorithms for scalable proximity operator computation and application to video denoising, Digit. Signal Process. 128 (2022) 103610.
[9] F. Sharifzadeh, G. Akbarizadeh, Y. Seifi Kavian, Ship classification in sar images using a new hybrid cnn–mlp classifier, J. Indian Soc. Remote Sen. 47 (2019) 551–562.
[10] F. Samadi, G. Akbarizadeh, H. Kaabi, Change detection in sar images using deep belief network: a new training approach based on morphological images, IET Image Process. 13 (2019) 2255–2264.
[11] Z. Tirandaz, G. Akbarizadeh, H. Kaabi, Polsar image segmentation based on feature extraction and data compression using weighted neighborhood filter bank and hidden Markov random field-expectation maximization, Measurement 153 (2020) 107432.
[12] J. Guo, Y. Sun, J. Gao, Y. Hu, B. Yin, Robust discriminant analysis with feature selective projection and between-classes structural incoherence, Digit. Signal Process. 134 (2023) 103896.
[13] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1991) 71–86.
[14] G.D. Pelegrina, L.T. Duarte, A novel approach for fair principal component analysis based on eigendecomposition, IEEE Trans. Artif. Intell. (2023) 1–12, https://doi.org/10.1109/TAI.2023.3298291.
[15] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (1936) 179–188.
[16] F. Ma, C. Wang, Y. Hao, X. Wu, Feature reduction based transfer structural subspace learning for small-footprint cross-domain keyword spotting via linear discriminant analysis, Digit. Signal Process. 127 (2022) 103594.

[17] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 711–720.
[18] A. Gisbrecht, B. Hammer, Data visualization by nonlinear dimensionality reduction, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5 (2015) 51–73.
[19] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Adv. Neural Inf. Process. Syst. 14 (2001).
[20] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[21] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[22] L.K. Saul, S.T. Roweis, An introduction to locally linear embedding, J. Mach. Learn. Res. 7 (2008).
[23] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. Roux, M. Ouimet, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, Adv. Neural Inf. Process. Syst. 16 (2003) 177–184.
[24] X. He, P. Niyogi, Locality preserving projections, Adv. Neural Inf. Process. Syst. 16 (2004) 153–160.
[25] X. He, D. Cai, S. Yan, H.-J. Zhang, Neighborhood preserving embedding, in: Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 1, vol. 2, 2005, pp. 1208–1213.
[26] Y. Pang, L. Zhang, Z. Liu, N. Yu, H. Li, Neighborhood preserving projections (npp): a novel linear dimension reduction method, in: International Conference on Intelligent Computing, Springer, 2005, pp. 117–124.
[27] E. Kokiopoulou, Y. Saad, Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 2143–2156.
[28] Y. Guo, J. Gao, P.W. Kwan, Kernel Laplacian eigenmaps for visualization of non-vectorial data, in: AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Proceedings, Hobart, Australia, December 4–8, 2006, Springer, 2006, pp. 1179–11831.
[29] V. Jain, L.K. Saul, Exploratory analysis and visualization of speech and music by locally linear embedding, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, IEEE, 2004, pp. iii–984.
[30] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008).
[31] G.E. Hinton, S. Roweis, Stochastic neighbor embedding, Adv. Neural Inf. Process. Syst. 15 (2002).
[32] L. McInnes, J. Healy, J. Melville, Umap: uniform manifold approximation and projection for dimension reduction, arXiv preprint, arXiv:1802.03426, 2018.
[33] A. Gisbrecht, A. Schulz, B. Hammer, Parametric nonlinear dimensionality reduction using kernel t-sne, Neurocomputing 147 (2015) 71–82.
[34] R. Ran, B. Fang, X. Wu, Exponential neighborhood preserving embedding for face recognition, IEICE Trans. Inf. Syst. 101 (2018) 1410–1420.
[35] P. Lv, X. Wu, Y. Zhao, J. Chang, Noise removal for semi-airborne data using wavelet threshold and singular value decomposition, J. Appl. Geophys. 201 (2022) 104622.
[36] Y. Wang, Y. Wu, Complete neighborhood preserving embedding for face recognition, Pattern Recognit. 43 (2010) 1008–1015.
[37] G.-F. Lu, Y. Wang, J. Zou, Improved complete neighbourhood preserving embedding for face recognition, IET Comput. Vis. 7 (2013) 71–79.
[38] D. Wei, X. Shen, Q. Sun, X. Gao, Z. Ren, Neighborhood preserving embedding on Grassmann manifold for image-set analysis, Pattern Recognit. 122 (2022) 108335.
[39] S. Mehta, B.-S. Zhan, X.-J. Shen, Weighted neighborhood preserving ensemble embedding, Electronics 8 (2019) 219.
[40] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[41] Y. Bengio, Learning Deep Architectures for AI, Now Publishers Inc., 2009.
[42] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2006).
[43] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59.
[44] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint, arXiv:1312.6114, 2013.
[45] M. Ranzato, Y.-L. Boureau, Y. Cun, et al., Sparse feature learning for deep belief networks, Adv. Neural Inf. Process. Syst. 20 (2007).
[46] Y. LeCun, The mnist database of handwritten digits, http://yann.lecun.com/exdb/mnist/, 1998.
[47] S.A. Nene, S.K. Nayar, H. Murase, et al., Columbia object image library (coil-20), 1996.
[48] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 643–660.
[49] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The feret evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1090–1104.
[50] S.K.N.S.A. Nene, H. Murase, Columbia object image library (coil-100), 1996.
[51] T. Zhang, J. Yang, D. Zhao, X. Ge, Linear local tangent space alignment and application to face recognition, Neurocomputing 70 (2007) 1547–1553.
[52] G. Akbarizadeh, A new statistical-based kurtosis wavelet energy feature for texture recognition of sar images, IEEE Trans. Geosci. Remote Sens. 50 (2012) 4358–4368.
[53] G. Akbarizadeh, G.A. Rezai-Rad, S.B. Shokouhi, A new region-based active contour model with skewness wavelet energy for segmentation of sar images, IEICE Trans. Inf. Syst. 93 (2010) 1690–1699.

**Ruisheng Ran** received the B.S. degree in mathematics from Chongqing Normal University, Chongqing, China, the M.S. degree in computational mathematics from University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree in computer application technology from the University of Electronic Science and Technology of China, Chengdu, China. He is currently a Professor with the College of computer and information science, Chongqing Normal University, Chongqing, China. He is doing pattern recognition, computer vision research.

**Jinping Wang** is a Master student at the College of Computer and Information science in Chongqing Normal University, Chongqing, China. He is doing machine learning, computer vision research.

**Bin Fang** received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, and the Ph.D. degree in electrical engineering from the University of Hong Kong, Hong Kong.

He is currently a Professor with the Department of Computer Science, Chongqing University, Chongqing, China. His research interests include computer vision, pattern recognition, medical image processing, biometrics applications, and document analysis. He has published more than 100 technical papers and is an Associate Editor of the International Journal of Pattern Recognition and Artificial Intelligence.

**Weiming Yang** received the B.S. degree in computer application from Chongqing Normal University, Chongqing, China, the M.S. degree in computer software and theory from Chongqing University, Chongqing, China. He is currently a Lecturer with the College of computer and information science, Chongqing Normal University, Chongqing, China. He is doing pattern recognition, intelligent learning research.