Expert Systems With Applications 144 (2020) 113089

Contents lists available at ScienceDirect

Expert Systems With Applications

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

M^{3}DNet: A manifold-based discriminant feature learning network for hyperspectral imagery

Zhengying Li, Hong Huang*, Yuan Li, Yinsong Pan

Key Laboratory of Optoelectronic Technology and Systems of the Education Ministry of China, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 9 August 2019
Revised 13 November 2019
Accepted 14 November 2019
Available online 23 November 2019

Keywords:
Hyperspectral imagery
Feature extraction
Manifold learning
Graph embedding
Adaptive optimization

ABSTRACT

Feature extraction (FE) is an effective method for learning discriminant features from hyperspectral image (HSI). Recently, graph embedding (GE) framework has been widely applied in FE of HSI data. GE unifies many classical FE methods and explores the low-dimensional embedding of high-dimensional data by a projection matrix generated from undirected weighted graphs. However, GE is unable to adaptively optimize projection matrix due to the absence of an iterative strategy in a single mapping process. To address this issue, a unified optimization method termed manifold-based maximization margin discriminant network (M^{3}DNet) was proposed to improve the performance of traditional FE methods. In M^{3}DNet, an initial projection matrix is obtained from original FE method, and then a maximal manifold margin criterion (M^{3}C) is proposed to maximize the margins among different classes, which enhances the discriminative ability of embedding features. After that, an iterative strategy is designed to optimize the projection matrix. Experiments on real-world HSI data sets indicate that the proposed M^{3}DNet performs significantly better than some state-of-the-art methods.

1. Introduction

Nowadays, hyperspectral imagery (HSI) has become one of the most active research fields in the remote sensing community (Hong, Yokoya, Chanussot, & Zhu, 2019; Huang, Duan, Shi, & Lv, 2018; Liang et al., 2018; Peng, Sun, & Du, 2019; Song, Li, Fang, & Lu, 2018; Su, Zhao, Du, & Du, 2019; Zhong et al., 2018). Each pixel in HSI data contains hundreds of continuous spectral bands through dense spectral sampling from visible to near-infrared wavelength, and these bands provide useful information for the fine classification of land covers (Jiao et al., 2018; Sellami, Farah, Farah, & Solaiman, 2019; Tang, Zhang, Marinoni, Gao, & Gamba, 2018; Zhang, Ke, Du, Zhang, & Hu, 2017). HSI has been successfully used in many fields such as resource exploration, precision agriculture, environmental science, and urban planning (Adao et al., 2017; He, Li, Liu, & Li, 2018; Maxwell, Warner, & Fang, 2018). The classification of each pixel in HSI plays a crucial role for these applications. However, high-dimensional characteristics of spectral features in HSI often lead to the curse-of-dimensionality, which easily causes the Hughes phenomena when traditional methods are introduced for classification. Therefore, it is urged to learn intrinsic low-dimensional features from high-dimensional HSI data (Dian, Li, Guo, & Fang, 2018; Huang, Li, & Pan, 2019; Kang, Duan, Li, & Benediktsson, 2018; Luo, Du, Zhang, Zhang, & Tao, 2018; Qian, Xiong, Zeng, Zhou, & Tang, 2017; Wang, Du, Zhang, Zhang, & Jia, 2017).

Feature extraction (FE) is commonly explored to extract low-dimensional embedding features, and a large number of FE methods have been proposed for processing high-dimensional data (Chen et al., 2019; Li, Li, & Zhang, 2019; Zhang, Gong, Mao, Li, & Wu, 2019a). Principal component analysis (PCA) is one of the typical FE methods, and it aims to preserve maximum variance through orthogonal projection (Tyo, Konsolakis, Diersen, & Olsen, 2003). Independent component analysis (ICA) is applied to find the desired characteristics of high-dimensional data by statistical independent components (Bayliss, Gualtieri, & Cromp, 1998). However, their unsupervised nature limits their discriminative ability for classification. Linear discriminant analysis (LDA) is a traditional supervised method, which uses the label information of samples to maximize the variance of original data in low-dimensional space (Du & Chang, 2001). Maximum margin criterion (MMC) has been proposed to maximize the trace of the difference between an interclass scatter matrix and an intraclass scatter matrix (Datta, Ghosh, & Ghosh, 2014). However, the above methods are based on statistics theory, and they ignore the underlying manifold structure in HSI.

Studies over the past two decades have discovered that HSI data have an intrinsic manifold structure (Bachmann, Ainsworth, &

* Corresponding author.
E-mail addresses: zhengying_li@cqu.edu.cn (Z. Li), hhuang@cqu.edu.cn (H. Huang), yuan_li@cqu.edu.cn (Y. Li), panys@cqu.edu.cn (Y. Pan).

https://doi.org/10.1016/j.eswa.2019.113089
0957-4174/

Fusina, 2005; Zhang, Wei, Bai, Gao, & Zhang, 2018). Many manifold learning methods have been designed to reveal the manifold structure that lies in a high-dimensional space. Such methods include locally linear embedding (LLE), isometric mapping (Isomap), Laplacian eigenmaps (LE), neighborhood preserving embedding (NPE), and locality preserving projections (LPP) (W. Li, Zhang, Zhang, & Du, 2017; X. R. Li, Pan, He, & Liu, 2015; Lin & Niu, 2014; Lu, Jin, & Zou, 2012; Roweis & Saul, 2000). However, these unsupervised methods cannot exploit the label information of training samples to obtain discriminant features for classification. Some supervised manifold learning methods were designed to address this problem. Sugiyama, Ide, Nakajima, and Sese (2010) proposed the local Fisher discriminate analysis (LFDA) to combine Fisher discriminate analysis (FDA) and LPP, and it utilized the prior knowledge to explore the geometric properties of HSI data. Yan et al. 2007 presented a marginal Fisher analysis (MFA) method that constructed an intrinsic graph and a penalty graph to reveal the discriminative manifold structure in data. Luo, Huang, Duan, Liu, and Liao (2017) proposed a local geometric structure Fisher analysis (LGSFA) method to learn the geometric characteristics of hyperspectral data by training samples and their corresponding reconstruction points, which compacts the intraclass neighbors with its reconstruction points while separating the interclass samples and its reconstruction points. Zhang, He, and Gao (2019b) proposed a supervised learning framework based on manifold learning that utilizes an explicit polynomial mapping to learn a compact low-dimensional feature space, in which fast feature extraction and classification for test samples can be achieved. However, these manifold learning methods only perform a single mapping process, which lacks an iterative optimization process to obtain a more optimized projection matrix. In the meanwhile, the aforementioned traditional FE methods depend on shallow-based feature descriptors that cannot deal with the nonlinear relationship in complex scenes.

Deep learning (DL) methods have achieved great success in various computer version tasks including object detection and scene classification. In recent years, DL has been explored to extract deep information from the spectral bands via the hierarchical network, and the deeper layers are able to handle the complex nonlinear relationship in HSI data (Liu, Gong, & He, 2019; Ma et al., 2019; Mostafa, 2017; Zhou, Xue, & Du, 2019). Ratle, Camps-Valls, and Weston (2010) proposed utilizing artificial neural networks (ANN) to extract intrinsic features in hyperspectral images and introduced a regularizer to the loss function for training neural networks. Chen, Zhao, & Jia, 2015 investigated the deep belief network (DBN) to learn the deep features of HSI data and achieved high classification accuracy by combining spectral-spatial information learning with deep feature extraction. Chen, Lin, Zhao, Wang, and Gu (2014) introduced a deep learning model based on stacked autoencoder (SAE) to classify HSI data through inputting spectral information directly into the DL framework. The above DL methods try to extract deep features of HSI data for improving the performance of land cover classification. However, these methods only focus on how to extract deep features and neglect to discover the intrinsic manifold structure in hyperspectral images.

To overcome the drawbacks of above methods, we proposed a novel unified framework termed manifold-based maximization margin discriminant network (M^3DNet) to improve the performance of FE methods. A projection matrix with initial weights is obtained by original FE methods, and then an adaptive optimization framework is designed to further optimize the projection matrix. Under the framework, a maximal manifold margin criterion (M^3C) is proposed to explore the geometric structure of HSI data by a graph-based strategy, and compact the intraclass samples while separating the interclass data through an adaptive process, which further improves the performance of the FE methods.

The main characteristics of the proposed method are listed as follows: 1) M^3DNet can be applied to different types of FE methods, and it enhances the general applicability of these methods by an adaptive optimization process. 2) M^3DNet combines the exploration of manifold structure in HSI with the iterative optimization process to realize the adaptive optimization of projection matrix. 3) An adaptive optimization framework with M^3C was designed to maximize the distance among different classes by an iteration process. As a result, the discriminative ability of extracted features is significantly improved.

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 introduces the details of the proposed method. Section 4 presents experimental results and gives a relevant analysis to demonstrate the effectiveness of our proposed M^3DNet method. Section 5 summarizes this paper and provides some suggestions for future work.

2. Related works

Let us assume a HSI data set $X = [x_1, x_2, x_3, \cdots, x_N] \in \Re^{D\times N}$, where $x_i \in \Re^D$ $(i = 1, 2, \cdots, N)$, N and D are the numbers of pixels and bands, respectively. Suppose there are c classes of land covers, $l(x_i) = \{1, 2, \cdots, c\}$ indicates the class label of x_i. The purpose of feature extraction is to extract low-dimensional features $Y = [y_1, y_2, y_3, \cdots, y_N] \in \Re^{d\times N}$, where $y_i \in \Re^d$, $d(d \ll D)$ is the dimension of embedding features. For convenience, the symbols used in the paper are summarized in Table 1.

2.1. Feature extraction

Feature extraction techniques have been widely used to extract valuable information from high-dimensional data. The FE algorithms can be categorized into unsupervised, semi-supervised and supervised learning methods (He, Liu, Wang, & Hu, 2017; Zabalza et al., 2015). Unsupervised learning methods learn low-dimensional features without utilizing the prior knowledge of training samples, and supervised learning methods exploit the label information of training set to enhance the discriminative ability of extracted features, while semi-supervised learning methods explore both unlabeled and labeled data for feature extraction.

Table 1
Notation and definitions.

X	Set of training samples
Y	Set of embedding features extracted from training samples
l	Labels of samples
D	Number of spectral bands
N	Number of training samples
c	Number of classes for land covers
d	Dimensionality of embedding features
h_i^L	Feature of the ith training sample extracted by original FE methods
sig	Nonlinear active function
G_w	Intraclass graph
G_b	Interclass graph
k_w	Number of intraclass neighbor points
k_b	Number of interclass neighbor points
w_{ij}^w	Intraclass weight between h_i^L and h_j^L
w_{ij}^b	Interclass weight between h_i^L and h_j^L
t_i	Heat kernel parameter
A	Projection matrix
A_i	Initial projection matrix (Input matrix)
A_o	Optimal projection matrix (Output matrix)
tr	Trace of the matrix
$\frac{\partial S}{\partial A}$	Gradient of loss function S with respect to projection matrix A
η	Learning rate
λ	Tradeoff parameter
T	Number of iterations

2.2. Graph embedding

The graph embedding (GE) framework is exploited to unify many classical FE methods (Yan et al., 2007), such as PCA, LDA, ISOMAP, LE, NPE, and LPP. Under this framework, the desirable geometrical or statistical properties are characterized through building an intrinsic graph $G(X, W)$, and some properties that should be avoided are represented by constructing a penalty graph $G_P(X, W^P)$, where X is the vertex set of a graph. G and G_P are undirected weighted graphs, $W \in \Re^{n \times n}$ and $W^P \in \Re^{n \times n}$ are the weight matrices of G and G_P. w_{ij} calculates the similarity of vertices x_i and x_j within graph G, and w^P_{ij} measures the dissimilarity between vertices x_i and x_j in graph G_P.

The purpose of graph embedding is to map vertexes of graphs into low-dimensional space while preserving the similarity of vertex pairs, and the objective function can be defined as

$$\arg\min_{Y^THY=k} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|y_i - y_j\right\|^2 w_{ij} = tr(Y^T(D-W)Y) = tr(Y^T L_G Y) \tag{1}$$

where D is a diagonal matrix, $D_{ii} = \sum_{j=1}^{N} w_{ij}$, $L_G = D - W$ is the Laplacian matrix of graph G, k is a constant, H is a constraint matrix for scale normalization, and it can be a Laplacian matrix of graph G_P, that is $H = L_P = D^P - W^P$, $D^P_{ii} = \sum_{j=1}^{N} w^P_{ij}$.

3. Proposed method

3.1. Motivation

Many FE methods try to tackle the curse-of-dimensionality in HSI. However, the methods based on GE framework, such as LDA, LFDA, LPP, and LGSFA, only depend on a single mapping process and they fail to introduce an iterative optimization process to further improve the performance of feature extraction. Some deep learning methods, DBN and SAE, cannot learn the manifold structure in HSI to enhance the discriminability of extracted deep features. To address the issues as discussed above, a unified adaptive optimization framework is proposed to improve the performance of classical FE methods.

3.2. Manifold-based margin maximization discriminant network

The goal of the M^{3}DNet is to adaptively optimize the projection matrix obtained by original FE methods. Based on graph embedding theory, M^{3}C is designed to explore the discriminative manifold structure of HSI data, which tries to enhance the separability of embedding features from different classes. The projection matrix is updated iteratively by the Back Propagation (BP) algorithm to realize the process of adaptive optimization. The flowchart of the proposed algorithm is shown in Fig. 1.

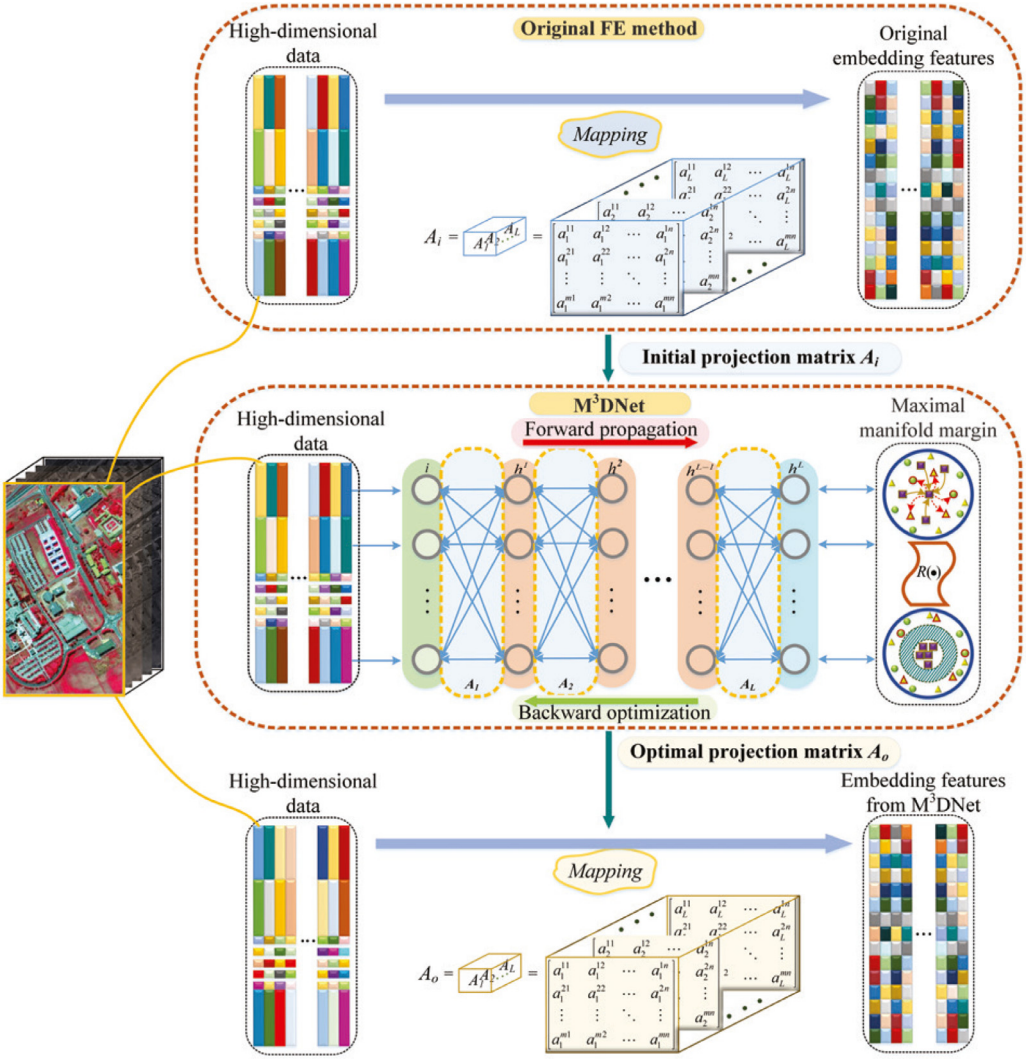

Fig. 1. Flowchart of the proposed M^{3}DNet method.

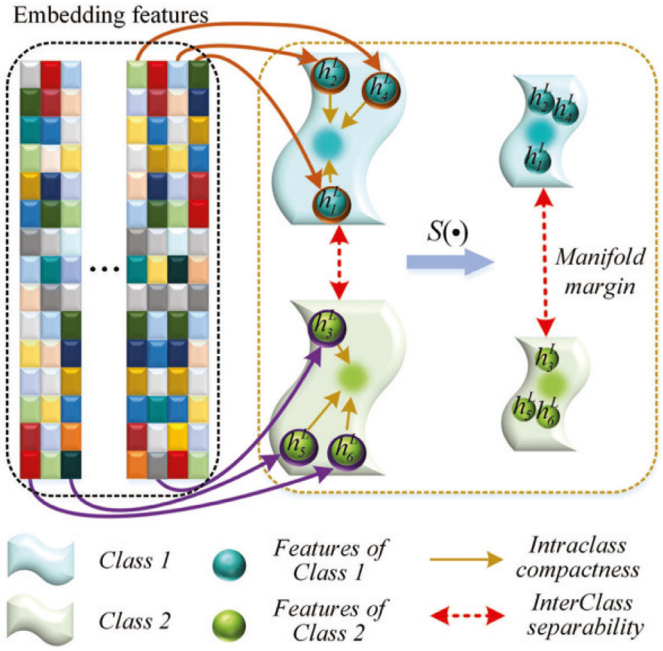

Fig. 2. Schematic diagram of M^{3}C.

3.2.1. Initial projection matrix

The original FE methods can obtain a projection matrix from hyperspectral data through feature learning. This projection matrix is used as an input matrix for M^{3}DNet and it can be further optimized by the proposed method.

As illustrated in Fig. 1, the projection matrix $A_i = [A_1, \cdots, A_l, \cdots, A_L]$ obtained from original FE methods is introduced into the adaptive optimization model as the initial projection matrix, where L denotes the number of layers in the projection matrix. For linear subspace methods and manifold learning methods, there is only one layer, i.e. $L = 1$. For deep learning methods, the value of L varies with the number of layers in deep learning networks. The artificial neural network is explored as the basic network and it extracts features from HSI data through a fully connected layer. The embedding feature of the ith training sample extracted by the FE method is defined as

$$h_i^L = sig(A_L h_i^{L-1} + b^L) \tag{2}$$

where $h_i^L \in \Re^d$, h_i^{L-1} is the output of layer $L-1$, b^L is the bias vector that is learned in the Lth layer, $sig = (\bullet)$ is a nonlinear activation function widely used in deep learning (Lu, Wang, Deng, Moulin, & Zhou, 2015).

3.2.2. Optimal projection matrix

The traditional deep learning networks exploit the cross entropy loss function to measure the difference between the predictive value and the actual value by an iterative process to minimize the loss function and optimize the network parameters. M^{3}DNet not only designs M^{3}C as the loss function of the network using label information to explore the geometric properties of HSI data and maximize the margins among different classes, but also proposes a different iterative optimization strategy to compact the intraclass samples and separate the interclass samples. After the iterative optimization, M^{3}DNet can get the optimal projection matrix. Then, we put HSI data into the projection matrix to get the discriminant embedding features. The diagram of M^{3}C is displayed in Fig. 2.

Under the M^{3}C, an intraclass graph G_w and an interclass graph G_b are constructed to explore the manifold structure in original embedding features. In G_w, each feature h_i^L is connected to corresponding intraclass neighbors, and the similarity weight w_{ij}^w between h_i^L and h_j^L is represented as

$$w_{ij}^w = \begin{cases} exp\left(-\dfrac{\left\|h_i^L - h_j^L\right\|^2}{2(t_i)^2}\right), & h_i^L \in N_w(h_j^L) \text{ or } h_j^L \in N_w(h_i^L) \\ 0, & otherwise \end{cases} \tag{3}$$

where $N_w(h_i^L)$ indicates the k_w intraclass neighbors of h_i^L, and the heat kernel parameter t_i is defined as $t_i = \frac{1}{n}\sum_{j=1}^{n}\|h_i^L - h_j^L\|$.

For interclass graph G_b, h_i^L is connected with the interclass neighbors from different classes. The weight w_{ij}^b between h_i^L and h_j^L is defined as follows:

$$w_{ij}^b = \begin{cases} exp\left(-\dfrac{\left\|h_i^L - h_j^L\right\|^2}{2(t_i)^2}\right), & h_i^L \in N_b(h_j^L) \text{ or } h_j^L \in N_b(h_i^L) \\ 0, & otherwise \end{cases} \tag{4}$$

where $N_b(h_i^L)$ indicates the k_b interclass neighbors of h_i^L.

The purpose of the M^{3}C is to explore the structure information for neighbor points. Suppose pixels x_i and x_j are neighbor points, the relationship should be preserved between h_i^L and h_j^L. The corresponding objective functions can be represented as follows:

$$J_1(h^L) = \frac{1}{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|h_i^L - h_j^L\right\|^2 w_{ij}^w\right) \tag{5}$$

$$J_2(h^L) = \frac{1}{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|h_i^L - h_j^L\right\|^2 w_{ij}^b\right) \tag{6}$$

With some mathematical operations, Eqs. (5) and (6) can be reduced as

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|h_i^L - h_j^L\right\|^2 w_{ij}^w \\ &= tr\left(\frac{1}{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left(h_i^L w_{ij}^w (h_i^L)^T - 2h_i^L w_{ij}^w (h_j^L)^T - h_j^L w_{ij}^w (h_j^L)^T\right)\right)\right) \\ &= tr\left(h^L(D^w - W^w)(h^L)^T\right) \\ &= tr\left(h^L L_w (h^L)^T\right) \end{aligned} \tag{7}$$

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|h_i^L - h_j^L\right\|^2 w_{ij}^b \\ &= tr\left(\frac{1}{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left(h_i^L w_{ij}^b (h_i^L)^T - 2h_i^L w_{ij}^b (h_j^L)^T - h_j^L w_{ij}^b (h_j^L)^T\right)\right)\right) \\ &= tr\left(h^L(D^b - W^b)(h^L)^T\right) \\ &= tr\left(h^L L_b (h^L)^T\right) \end{aligned} \tag{8}$$

where $W^w = [w_{ij}^w]_{i,j=1}^N$, $D^w = diag([\sum_{j=1}^N w_{ij}^w]_{i=1}^N)$, $W^b = [w_{ij}^b]_{i,j=1}^N$, $D^b = diag([\sum_{j=1}^N w_{ij}^b]_{i=1}^N)$.

As discussed above, the proposed algorithm should be explored to minimize the distance among the features extracted from the same class and maximize the distance among the features extracted from different classes. Therefore, the following optimization problem is formulated as

$$S_1 = min(J_1(h^L) - J_2(h^L)) \tag{9}$$

According to Eq. (9), the distance between each feature h_i^L and its k_w intraclass neighbors can be minimized and that between h_i^L and its k_b interclass neighbors can be maximized, which effectively increases the manifold margin of features obtained from the Lth layer.

By introducing the regularized parameter term, the objective function is defined as follows:

$$\begin{aligned} S &= S_1 + \frac{\lambda}{2} S_2 \\ &= (J_1(h^L) - J_2(h^L)) + \frac{\lambda}{2} \sum_{l=1}^{L} (\|A_l\|_F^2 + \|b_l\|_2^2) \end{aligned} \tag{10}$$

where S_2 regularizes the parameters of the networks, λ is a tradeoff parameter.

Based on the above analysis, the iteration strategy is introduced to achieve an optimal projection matrix. The BP algorithm and stochastic gradient descent method are adopted to update the projection matrix by Eqs. (2) and (10) until convergence. Assume $l(1 \le l \le L)$ represents the lth layer of the projection matrix, the gradient of loss function S with respect to the projection matrix can be calculated as

$$\begin{aligned} \frac{\partial S}{\partial A_l} &= \frac{\partial S_1}{\partial A_l} + \frac{\lambda}{2} \frac{\partial S_2}{\partial A_l} \\ &= \frac{\partial S_1}{\partial h^L} \cdot \frac{\partial h^L}{\partial A_l} + \lambda A_l \\ &= \frac{\partial (J_1(h^L) - J_2(h^L))}{\partial h^L} \cdot \frac{\partial h^L}{\partial A_l} + \lambda A_l \\ &= \frac{\partial (tr(h^L L_w (h^L)^T) - tr(h^L L_b (h^L)^T))}{\partial h^L} \cdot \frac{\partial h^L}{\partial A_l} + \lambda A_l \\ &= \frac{\partial (tr(h^L (L_w - L_b)(h^L)^T))}{\partial h^L} \cdot \frac{\partial h^L}{\partial A_l} + \lambda A_l \\ &= 2h^L (L_w - L_b) \cdot \frac{\partial h^L}{\partial A_l} + \lambda A_l \\ &= 2h^L (L_w - L_b) \cdot h^L \cdot (1 - h^L) \cdot A_L \cdot \cdots \cdot h^l \cdot (1 - h^l) \cdot h^{l-1} + \lambda A_l \end{aligned} \tag{11}$$

The projection matrix is updated by BP algorithm depending on these gradients. Let η indicates the learning rate, and the projection matrix is updated by

$$A = A - \eta \frac{\partial S}{\partial A} \tag{12}$$

The proposed M^{3}DNet algorithm is summarized in Algorithm 1.

Algorithm 1 M^{3}DNet.

Input: hyperspectral data set $X = [x_1, x_2, x_3, \cdots, x_n] \in \Re^{D \times N}$, corresponding class labels $l(x_i) \in \{1, 2, \ldots, c\}$, initial projection matrix A_i, the number of neighbors k_w and k_b, learning rate η, tradeoff parameter λ, iteration number T, and convergence error ε.

1: for t = 1 to T do
2: Find k_w intraclass neighbor points and k_b interclass neighbor points of the input data.
3: Calculate the weights of intraclass graph G_w and the interclass graph G_b by Eqs. (3) and (4).
4: Compute D^w and D^b by $D^w = diag([\sum_{j=1}^{N} w_{ij}^w]_{i=1}^N)$ and $D^b = diag([\sum_{j=1}^{N} w_{ij}^b]_{i=1}^N)$, respectively.
5: Calculate S_t through Eq. (10).
6: If $t > 1$ and $|S_t - S_{t-1}| < \varepsilon$, go to Return.
7: Achieve the gradients according to Eq. (11).
8: Update A by Eq. (12).
9: end for
10: Obtain the optimal projection matrix A_o.

Output: $Y = A_o^T X \in \Re^{d \times N}$

4. Experimental results and analysis

4.1. Data sets description

Indian Pines data set: This data set was captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in Northwest Indiana. The scene possesses a spatial size of 145 × 145 pixels and the spatial resolution is 20 m. There are 200 bands used for experiments after removing 20 bands influenced by water vapor and atmosphere effects. The data contains sixteen classes of land covers. This scene in false color and its ground truth are shown in Fig. 3, and the values in brackets indicate the number of samples in each class.

PaviaU data set: The second data set is a scene of the Pavia University (PaviaU) collected by the reflective optics system imaging spectrometer (ROSIS) sensor. The image size is 610 × 340 pixels with the spatial resolution is 1.3 m. The set is composed of 103 spectral channels after the 12 bands were removed due to the noise. Nine ground truth types are used in this data set. More detailed information about this image can be found in Fig. 4.

4.2. Experimental setup

In each experiment, HSI data was randomly divided into a training set and a test set. Some classes have a small number of samples like *Alfalfa, Oats* and *Grass/pasture – mowed* in Indian Pines data set, 10 samples were chosen from each class for training. The FE methods utilize training samples to learn a low-dimensional embedding space, and then all test samples were mapped into the embedding space. After that, K-nearest neighborhood (KNN) classifier was utilized for classification. Overall

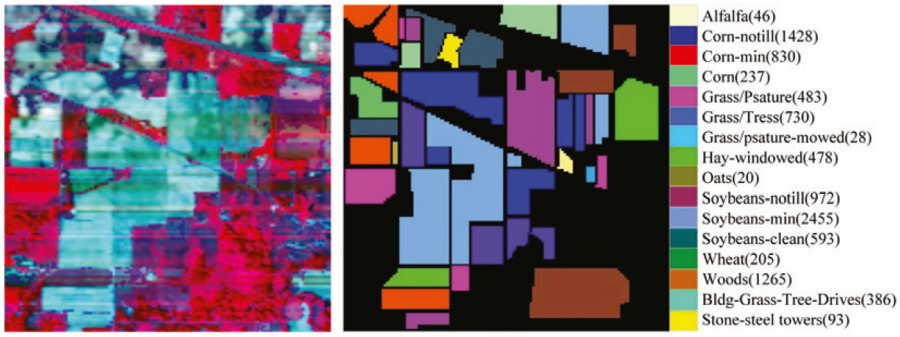

Fig. 3. Indian Pines hyperspectral image. (a) HSI in false-color; (b) Ground-truth map.

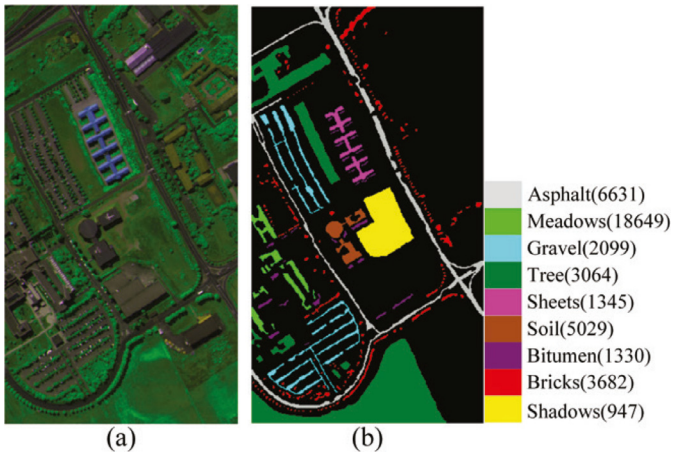

Fig. 4. PaviaU hyperspectral image. (a) HSI in false-color; (b) Ground-truth map.

accuracies (OAs), the classification accuracies for each class, average accuracies (AAs) and kappa coefficient (Kappa) (Thompson & Walter, 1988) were explored to evaluate the classification results of all FE methods. For M^{3}DNet, the λ was empirically set as 0.00001 (Lu et al., 2015). In order to achieve robust results, we repeated the experiments 10 times in each condition, and the results were presented in the form of mean with standard deviation (STD). All experiments were carried out on a personal computer with 32-G memory, i7-7800X central processing unit and 64-bit Windows 10 using MATLAB 2014b.

4.3. Two-dimension embedding

In this subsection, the Indian Pines data set is used to illustrate the discriminant capability of the M^{3}DNet. In the experiments, we only chose five classes of land covers including *Corn, Grass/Tress, Hay – windowed, Wheat* and *Woods*, and they were denoted as 1, 2, 3, 4 and 5. Fifty samples in each class were randomly selected as the training set, and the remaining samples were projected into the two-dimension embedding space to facilitate data visualization. Fig. 5 shows the two-dimensional distributions of the original FE methods and these methods optimized by M^{3}DNet.

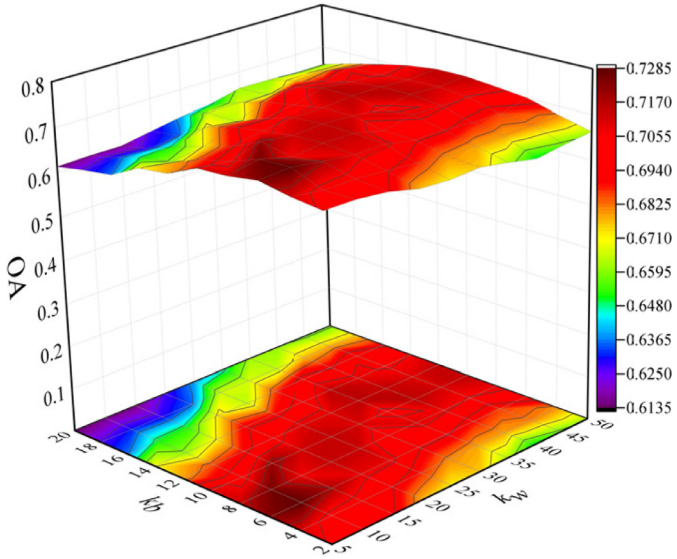

Fig. 6. The OAs of LGSFA-M^{3}DNet versus the different number of k_w and k_b on the PaviaU data set.

As illustrated in Fig. 5, original FE methods produce a large number of overlapped points from different classes, because these methods cannot further optimize the projection matrix obtained from a single mapping process. The FE methods optimized by M^{3}DNet achieved better performance through iteratively enhancing the compactness of intraclass samples and the separation of interclass samples.

4.4. Evaluation of the number of neighbor points

To investigate the relationship between the number of neighbors in M^{3}DNet and classification accuracy, we randomly chose 40 samples from each class for training, and the rest were used for testing. In the experiment, parameters k_w and k_b are tuned with a set of {5, 10, 15, ⋯, 50} and a set of {2, 4, 6, ⋯, 20}, respectively. The OAs of LGSFA-M^{3}DNet versus the different number of neighbors on the PaviaU data set are displayed in Fig. 6.

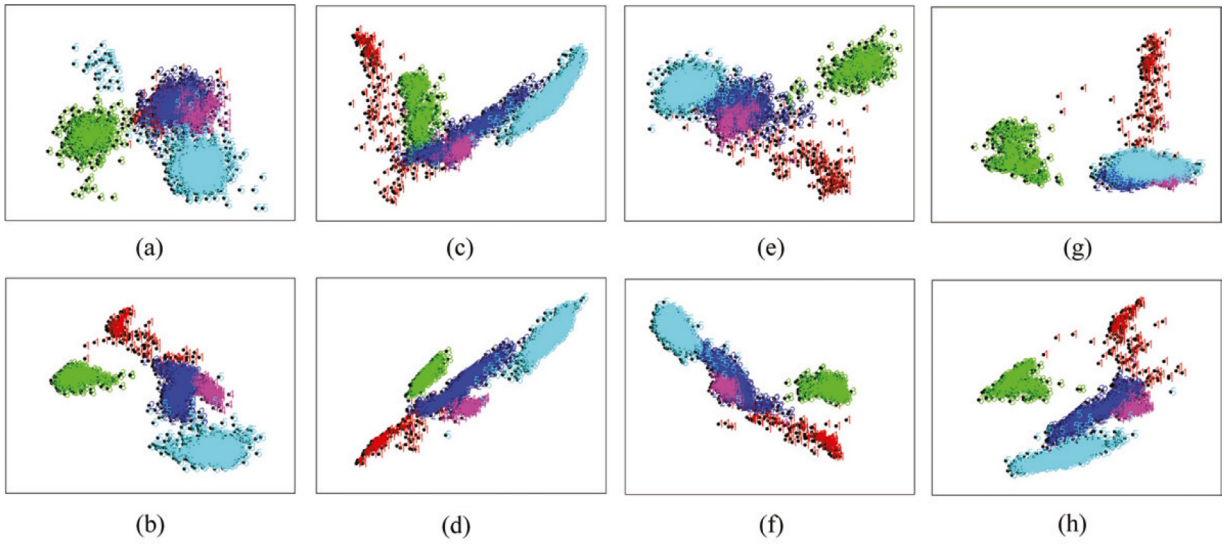

Fig. 5. Two-dimension embedding of different FE methods on the Indian Pines data set. (a) LDA-Original; (b) LDA-M^{3}DNet; (c) LFDA-Original; (d) LFDA-M^{3}DNet; (e) LPP-Original; (f) LPP-M^{3}DNet; (g) LGSFA-Original; (h) LGSFA-M^{3}DNet.

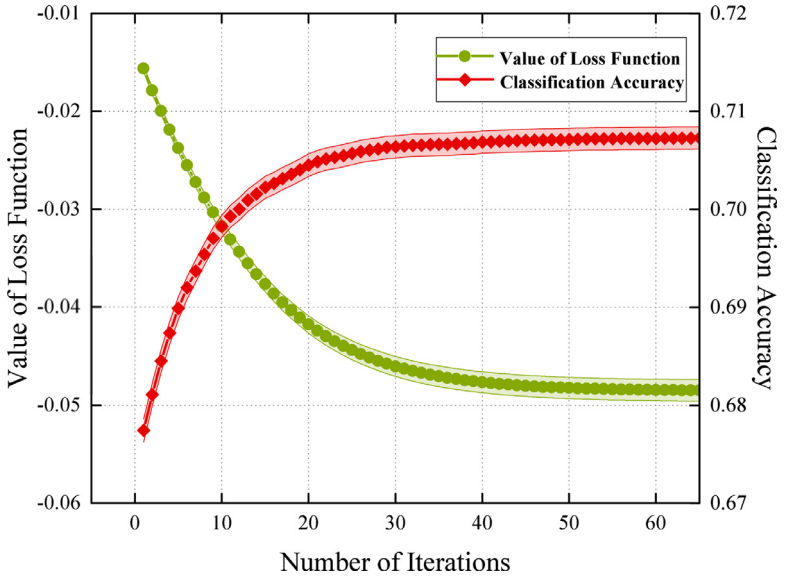

Fig. 7. Convergence curve and classification accuracy versus a different number of iterations for LGSFA-M^{3}DNet on the PaviaU data set.

Table 2
The classification results with different FE methods on PaviaU data set (Overall Accuracy ± Std(%)).

Algorithm		$n_i = 20$	$n_i = 30$	$n_i = 40$	$n_i = 50$	$n_i = 80$	$n_i = 100$
LDA	Original	57.26 ± 4.12	63.65 ± 3.09	65.83 ± 1.40	67.61 ± 1.86	71.30 ± 0.94	72.81 ± 1.17
	M^{3}DNet	63.49 ± 3.49	66.08 ± 1.72	67.63 ± 1.59	70.11 ± 2.06	72.57 ± 1.33	73.85 ± 1.34
LFDA	Original	59.44 ± 2.16	61.98 ± 1.72	62.05 ± 2.33	62.33 ± 0.99	63.40 ± 0.89	64.02 ± 1.54
	M^{3}DNet	59.84 ± 2.12	62.15 ± 1.75	62.23 ± 2.40	62.95 ± 1.13	63.92 ± 1.04	64.98 ± 1.54
LPP	Original	55.07 ± 3.16	62.96 ± 2.42	65.73 ± 2.50	67.34 ± 1.19	72.04 ± 1.19	73.16 ± 1.56
	M^{3}DNet	57.37 ± 3.12	64.36 ± 2.19	67.45 ± 1.81	69.49 ± 1.49	73.29 ± 1.30	73.69 ± 1.23
LGSFA	Original	62.85 ± 4.53	67.08 ± 2.55	68.46 ± 1.55	68.97 ± 2.24	73.62 ± 1.00	75.13 ± 1.04
	M^{3}DNet	67.46 ± 3.93	69.98 ± 1.54	70.80 ± 2.31	71.94 ± 2.10	74.44 ± 1.02	76.20 ± 1.00
ANN	Original	71.12 ± 5.10	73.69 ± 2.13	73.94 ± 3.86	77.82 ± 4.00	78.80 ± 2.07	79.34 ± 2.38
	M^{3}DNet	72.93 ± 3.49	74.15 ± 2.16	75.56 ± 2.22	78.13 ± 2.35	79.61 ± 3.47	81.51 ± 1.22
DBN	Original	51.52 ± 10.1	61.78 ± 4.60	62.72 ± 3.19	64.29 ± 6.15	72.24 ± 4.06	74.86 ± 3.18
	M^{3}DNet	49.96 ± 12.2	61.00 ± 5.26	63.43 ± 4.68	66.94 ± 3.73	73.21 ± 5.47	76.29 ± 3.77
SAE	Original	70.47 ± 5.06	73.40 ± 5.22	74.91 ± 1.44	75.30 ± 2.89	77.82 ± 3.24	79.60 ± 1.70
	M^{3}DNet	72.06 ± 4.02	74.36 ± 3.09	75.67 ± 1.12	78.04 ± 1.12	79.33 ± 3.12	81.71 ± 1.44

Table 3
The classification results with different FE methods on Indian Pines data set (Overall Accuracy ± Std(%)).

Algorithm		$n_i = 20$	$n_i = 30$	$n_i = 40$	$n_i = 50$	$n_i = 80$	$n_i = 100$
LDA	Original	50.13 ± 2.04	50.94 ± 1.74	51.99 ± 1.23	52.52 ± 1.39	54.61 ± 0.80	55.90 ± 0.95
	M^{3}DNet	51.84 ± 1.65	52.24 ± 1.91	52.59 ± 1.35	53.09 ± 1.50	61.60 ± 2.20	65.85 ± 0.98
LFDA	Original	49.32 ± 1.77	52.83 ± 2.20	57.52 ± 0.62	58.75 ± 1.12	61.42 ± 1.21	63.11 ± 0.80
	M^{3}DNet	50.25 ± 2.05	53.15 ± 2.32	57.81 ± 0.55	59.05 ± 1.11	61.70 ± 1.21	63.16 ± 1.08
LPP	Original	43.48 ± 1.63	59.08 ± 1.51	62.59 ± 0.97	65.34 ± 0.86	68.85 ± 0.81	70.87 ± 0.71
	M^{3}DNet	43.57 ± 1.63	64.60 ± 1.83	68.49 ± 1.12	70.54 ± 0.89	73.56 ± 0.62	75.03 ± 0.75
LGSFA	Original	56.65 ± 2.07	63.86 ± 1.19	67.21 ± 1.65	69.80 ± 1.09	73.83 ± 0.69	74.62 ± 0.69
	M^{3}DNet	60.77 ± 2.11	67.36 ± 1.14	70.40 ± 1.60	72.22 ± 1.20	75.60 ± 0.52	75.92 ± 0.70
ANN	Original	64.95 ± 1.86	70.14 ± 1.87	71.82 ± 0.92	74.27 ± 0.93	76.92 ± 0.89	78.51 ± 1.12
	M^{3}DNet	65.90 ± 2.60	70.21 ± 2.33	72.46 ± 1.08	75.30 ± 0.67	77.37 ± 1.06	79.16 ± 0.76
DBN	Original	60.18 ± 3.00	64.32 ± 3.62	64.59 ± 3.22	69.75 ± 1.43	72.14 ± 1.60	73.35 ± 3.67
	M^{3}DNet	61.05 ± 3.42	64.43 ± 4.02	64.94 ± 2.60	70.94 ± 1.42	72.83 ± 1.58	74.59 ± 2.07
SAE	Original	64.29 ± 2.58	69.98 ± 1.79	71.37 ± 0.64	74.08 ± 1.61	77.25 ± 0.72	78.97 ± 0.90
	M^{3}DNet	65.49 ± 2.00	70.70 ± 1.12	72.52 ± 0.49	75.68 ± 0.67	78.62 ± 1.14	79.79 ± 0.66

As can be seen in Fig. 6, the OAs improved and then declined with the increase of k_b. It is for the reason that a small value of k_b fails to obtain sufficient information to represent the interclass structure, while a large size of k_b will lead to overfitting phenomenon. At the same time, an appropriate value of k_w can effectively explore the intraclass structure, while a large value of k_w will result in the saturation of discriminant information and cause a decrease in classification performance. To achieve better experiment results, we selected k_w and k_b as 10 and 6, respectively.

4.5. Convergence analysis

In this subsection, we analyze the convergence of M^{3}DNet with a different number of iterations. 40 samples per class were randomly selected as the training set and the remaining samples as the test data. In each iteration, we repeated the experiment 10 times to obtain the mean value and standard deviation of classification accuracy and loss function value under the different number of iterations. Fig. 7 indicates the mean and standard deviation for

Table 4
Classification results of each class samples via different deep learning methods on PaviaU data set (%).

Class	Samples		ANN		DBN		SAE	
	Train	Test	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet
Asphalt	67	6564	84.77 ± 3.54	85.14 ± 3.08	78.74 ± 13.9	83.71 ± 1.85	82.02 ± 3.42	84.76 ± 2.64
Meadows	187	18,462	95.69 ± 1.72	94.27 ± 1.06	95.72 ± 2.60	95.98 ± 2.55	95.76 ± 1.30	92.20 ± 1.91
Gravel	21	2078	57.53 ± 1.42	56.01 ± 0.82	0	0	51.36 ± 1.78	58.23 ± 0.97
Tree	31	3033	80.46 ± 0.94	85.44 ± 0.41	79.06 ± 1.21	80.27 ± 0.40	78.66 ± 1.04	87.29 ± 0.32
Sheets	14	1331	99.20 ± 0.34	99.36 ± 0.21	99.15 ± 0.36	99.24 ± 0.31	89.24 ± 3.14	89.35 ± 3.14
Soil	51	4978	62.26 ± 3.49	65.40 ± 3.18	40.89 ± 1.79	43.27 ± 1.79	61.46 ± 2.65	67.84 ± 3.27
Bitumen	14	1316	8.182 ± 1.98	8.606 ± 2.10	0	0	8.341 ± 2.26	4.811 ± 1.02
Bricks	37	3645	81.23 ± 1.78	88.34 ± 2.66	89.24 ± 4.44	90.97 ± 2.74	86.48 ± 7.88	86.77 ± 2.85
Shadows	10	937	75.42 ± 2.69	79.97 ± 2.84	90.66 ± 0.59	92.04 ± 0.50	73.11 ± 2.82	74.47 ± 2.82
OA			82.79 ± 1.65	83.62 ± 0.99	77.21 ± 3.70	78.64 ± 1.60	81.96 ± 1.25	82.49 ± 1.32
AA			79.74 ± 3.52	80.08 ± 4.37	81.92 ± 5.16	83.64 ± 2.99	76.55 ± 5.36	78.89 ± 3.70
Kappa			76.72 ± 2.19	78.00 ± 1.36	68.82 ± 5.17	70.83 ± 2.44	75.55 ± 1.79	76.63 ± 1.75

Table 5
Classification results of each class samples via different linear subspace methods and manifold learning methods on PaviaU data set (%).

Class	LDA		LFDA		LPP		LGSFA	
	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet
1	84.76 ± 2.04	88.34 ± 2.14	81.18 ± 3.20	82.46 ± 2.35	82.31 ± 3.44	83.13 ± 2.54	85.36 ± 1.94	88.94 ± 2.35
2	89.68 ± 0.63	90.90 ± 1.18	84.30 ± 2.20	86.32 ± 1.99	89.15 ± 2.31	89.75 ± 2.09	96.63 ± 1.11	97.04 ± 0.99
3	38.82 ± 6.10	40.86 ± 7.15	39.67 ± 4.37	48.38 ± 5.28	47.72 ± 6.03	50.00 ± 6.92	36.05 ± 7.60	39.46 ± 7.10
4	80.88 ± 2.06	80.91 ± 3.09	64.33 ± 2.61	65.33 ± 3.50	73.79 ± 3.42	75.35 ± 4.20	83.03 ± 2.66	83.15 ± 2.53
5	99.63 ± 0.13	98.98 ± 0.27	97.96 ± 0.91	98.92 ± 0.30	99.38 ± 0.10	99.55 ± 0.06	99.67 ± 0.12	99.12 ± 0.28
6	53.05 ± 3.62	52.09 ± 3.25	41.49 ± 2.34	43.28 ± 2.23	50.22 ± 3.14	51.65 ± 4.88	37.42 ± 8.36	44.53 ± 6.38
7	20.42 ± 4.09	30.76 ± 4.98	57.23 ± 10.5	61.60 ± 12.1	61.54 ± 9.95	60.92 ± 10.6	11.30 ± 4.02	21.49 ± 4.73
8	62.85 ± 6.70	77.12 ± 4.54	72.43 ± 2.34	76.35 ± 3.12	75.40 ± 3.41	74.63 ± 5.72	66.35 ± 8.26	77.45 ± 6.84
9	86.99 ± 6.06	98.96 ± 0.29	99.68 ± 0.05	99.69 ± 0.03	99.69 ± 0.03	98.76 ± 1.05	85.87 ± 12.7	99.07 ± 0.14
OA	77.27 ± 0.66	80.14 ± 0.57	74.07 ± 0.98	76.36 ± 0.95	78.89 ± 1.26	79.57 ± 1.23	78.56 ± 1.64	81.86 ± 0.84
AA	68.57 ± 1.12	73.21 ± 1.01	70.92 ± 1.47	73.59 ± 1.83	75.47 ± 1.41	75.97 ± 1.78	66.85 ± 3.07	72.25 ± 1.58
Kappa	69.56 ± 0.91	73.31 ± 0.75	65.29 ± 1.18	68.26 ± 1.19	71.68 ± 1.60	72.60 ± 1.59	70.55 ± 2.39	75.17 ± 1.24

Table 6
Classification results of each class samples via different deep learning methods on Indian Pines data set (%).

Class	Samples		ANN		DBN		SAE	
	Train	Test	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet
Alfalfa	10	36	50.56 ± 2.94	43.33 ± 2.81	21.94 ± 2.79	20.28 ± 2.68	58.89 ± 1.15	58.33 ± 1.28
Corn-notill	143	1285	76.05 ± 1.88	79.45 ± 2.73	71.06 ± 5.16	74.03 ± 3.26	75.90 ± 3.18	78.70 ± 1.88
Corn-min	83	747	64.37 ± 3.70	68.84 ± 3.66	50.79 ± 1.04	45.65 ± 1.22	67.53 ± 4.42	70.20 ± 4.41
Corn	24	213	25.30 ± 0.74	30.28 ± 0.87	1.475 ± 3.92	1.429 ± 3.91	24.06 ± 0.75	28.80 ± 0.99
Grass/Psature	49	434	84.55 ± 3.34	87.14 ± 2.77	70.72 ± 1.10	68.89 ± 1.44	85.84 ± 2.85	87.85 ± 2.61
Grass/Tress	73	657	94.76 ± 2.22	96.73 ± 1.61	95.89 ± 2.15	95.70 ± 2.50	95.77 ± 1.25	97.14 ± 1.16
Grass/Psature-mowed	10	18	76.67 ± 2.71	73.33 ± 2.88	0	0	82.22 ± 0.97	86.11 ± 0.54
Hay-windowed	48	430	96.40 ± 1.42	98.29 ± 1.33	93.74 ± 4.47	95.23 ± 4.01	95.58 ± 2.58	97.45 ± 1.60
Oats	10	10	76.00 ± 3.24	85.00 ± 3.06	35.00 ± 4.12	31.00 ± 3.90	62.00 ± 4.29	71.00 ± 3.93
Soybeans-notill	98	874	72.45 ± 3.74	77.25 ± 4.47	67.81 ± 6.38	69.69 ± 5.17	75.11 ± 4.46	77.05 ± 4.98
Soybeans-min	246	2209	85.87 ± 2.13	82.77 ± 1.64	82.54 ± 4.36	81.32 ± 3.33	85.10 ± 1.97	82.93 ± 1.85
Soybeans-clean	60	533	65.44 ± 5.02	74.30 ± 4.30	41.99 ± 1.54	45.01 ± 1.60	63.98 ± 3.10	74.41 ± 2.11
Wheat	21	184	94.76 ± 2.88	95.24 ± 4.61	94.86 ± 0.67	94.59 ± 0.70	94.16 ± 2.57	95.95 ± 2.70
Woods	127	1138	94.87 ± 2.16	94.62 ± 2.16	92.67 ± 4.86	94.17 ± 2.00	94.10 ± 4.19	94.41 ± 1.98
Bldg-Grass-Tree	39	347	61.47 ± 0.92	63.24 ± 0.79	62.51 ± 0.73	61.99 ± 0.75	61.71 ± 1.00	64.71 ± 0.69
Stone-steel towers	10	83	55.78 ± 3.89	57.95 ± 4.01	80.00 ± 0.67	78.43 ± 0.78	74.82 ± 2.69	75.90 ± 2.70
OA			79.86 ± 0.69	81.43 ± 0.84	73.88 ± 1.87	74.04 ± 2.02	80.24 ± 0.80	81.73 ± 0.58
AA			76.79 ± 2.22	78.88 ± 2.18	72.68 ± 3.58	72.99 ± 4.05	76.26 ± 2.47	79.04 ± 1.57
Kappa			76.80 ± 0.80	78.75 ± 0.95	69.86 ± 2.18	70.09 ± 2.32	77.28 ± 0.94	79.10 ± 0.68

loss function value and classification accuracy of LGSFA-M^{3}DNet with a different number of iterations on the PaviaU data set.

As shown in Fig. 7, with the number of iterations increases, the value of loss function declines sharply in the first 30 iterations and then tends to converge after 50 iterations, while the classification performance dramatically improves in the first 30 iterations and then maintains a stable value after 50 iterations. Meanwhile, the standard deviation of loss function value and classification accuracy value are stabilized in a small range. Therefore, it is obvious that M^{3}DNet converges within 50 iterations.

4.6. Comparisons between the state-of-the-art FE methods and M^{3}DNet

To demonstrate the effectiveness of the proposed method, a series of experiments were designed to compare the classification accuracy of some state-of-art FE methods with the same methods optimized by M^{3}DNet. These methods include linear space methods, manifold learning methods, and deep learning methods, such as LDA, LFDA, LPP, LGSFA, ANN, DBN, and SAE. For all methods, the parameters were optimized by adopting cross-validation to obtain

Table 7
Classification results of each class samples via different linear subspace methods and manifold learning methods on Indian Pines data set (%).

Class	LDA		LFDA		LPP		LGSFA	
	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet	Original	M^{3}DNet
1	68.06 ± 11.5	71.67 ± 9.15	66.39 ± 7.91	52.78 ± 14.5	50.83 ± 12.5	54.17 ± 13.0	69.72 ± 10.8	74.72 ± 11.9
2	66.29 ± 2.17	66.79 ± 2.17	44.74 ± 2.53	55.61 ± 1.81	49.94 ± 2.41	50.23 ± 3.15	68.70 ± 2.60	68.54 ± 2.47
3	52.91 ± 3.45	60.47 ± 3.19	38.69 ± 2.33	48.91 ± 5.55	47.16 ± 3.19	49.27 ± 3.72	61.92 ± 3.81	65.01 ± 3.70
4	38.71 ± 6.16	44.98 ± 6.17	28.20 ± 4.55	27.37 ± 4.05	31.75 ± 4.72	31.01 ± 4.53	44.52 ± 4.99	48.25 ± 5.69
5	89.03 ± 1.82	88.82 ± 2.09	66.21 ± 4.78	81.69 ± 3.85	79.75 ± 2.02	78.71 ± 3.33	88.66 ± 1.48	88.29 ± 1.95
6	94.38 ± 1.76	94.65 ± 2.04	78.11 ± 4.11	91.80 ± 1.46	87.71 ± 2.76	89.82 ± 2.80	95.24 ± 2.62	94.73 ± 2.60
7	93.89 ± 5.52	90.56 ± 5.27	73.33 ± 9.37	90.56 ± 6.44	85.56 ± 5.37	88.89 ± 4.54	90.56 ± 4.57	90.56 ± 4.57
8	98.41 ± 0.75	98.88 ± 0.98	90.28 ± 2.42	92.10 ± 2.67	92.43 ± 2.85	92.90 ± 2.90	97.62 ± 1.50	97.71 ± 1.49
9	83.00 ± 13.4	85.00 ± 17.2	36.00 ± 10.7	87.00 ± 11.6	51.00 ± 21.3	66.00 ± 15.8	81.00 ± 13.7	83.00 ± 16.4
10	57.36 ± 3.40	71.01 ± 2.40	57.12 ± 3.47	60.00 ± 4.30	62.27 ± 3.35	64.97 ± 4.11	69.99 ± 3.37	75.85 ± 2.99
11	71.33 ± 2.38	77.93 ± 2.15	61.90 ± 2.36	68.71 ± 2.71	67.89 ± 2.20	69.55 ± 1.33	78.35 ± 1.35	80.18 ± 1.46
12	64.33 ± 3.95	65.52 ± 4.47	33.40 ± 2.44	43.47 ± 3.43	39.74 ± 3.09	39.94 ± 3.69	65.76 ± 4.32	65.65 ± 3.86
13	96.86 ± 1.46	96.65 ± 0.98	76.76 ± 5.92	90.38 ± 5.31	88.70 ± 2.90	90.86 ± 3.87	96.92 ± 1.05	96.54 ± 1.47
14	93.00 ± 2.02	92.04 ± 2.11	82.30 ± 3.01	88.69 ± 3.21	88.57 ± 2.51	88.22 ± 2.91	93.08 ± 2.12	93.26 ± 1.60
15	61.27 ± 4.80	56.01 ± 3.61	24.94 ± 3.77	40.58 ± 3.67	34.48 ± 2.83	33.82 ± 2.43	57.49 ± 3.79	54.19 ± 3.30
16	88.67 ± 2.06	92.29 ± 3.42	89.16 ± 3.16	85.06 ± 4.44	83.37 ± 3.40	93.61 ± 2.35	86.02 ± 1.63	90.48 ± 3.61
OA	73.38 ± 0.70	76.91 ± 0.39	59.08 ± 0.81	67.34 ± 1.68	65.60 ± 1.00	66.68 ± 1.05	77.40 ± 0.81	78.61 ± 0.63
AA	76.09 ± 1.13	78.33 ± 1.10	59.22 ± 1.03	69.04 ± 1.86	65.07 ± 2.06	67.62 ± 2.05	77.85 ± 1.51	79.19 ± 1.67
Kappa	69.50 ± 0.80	73.55 ± 0.46	53.33 ± 0.92	62.67 ± 1.92	60.71 ± 1.11	61.97 ± 1.21	74.11 ± 0.94	75.52 ± 0.73

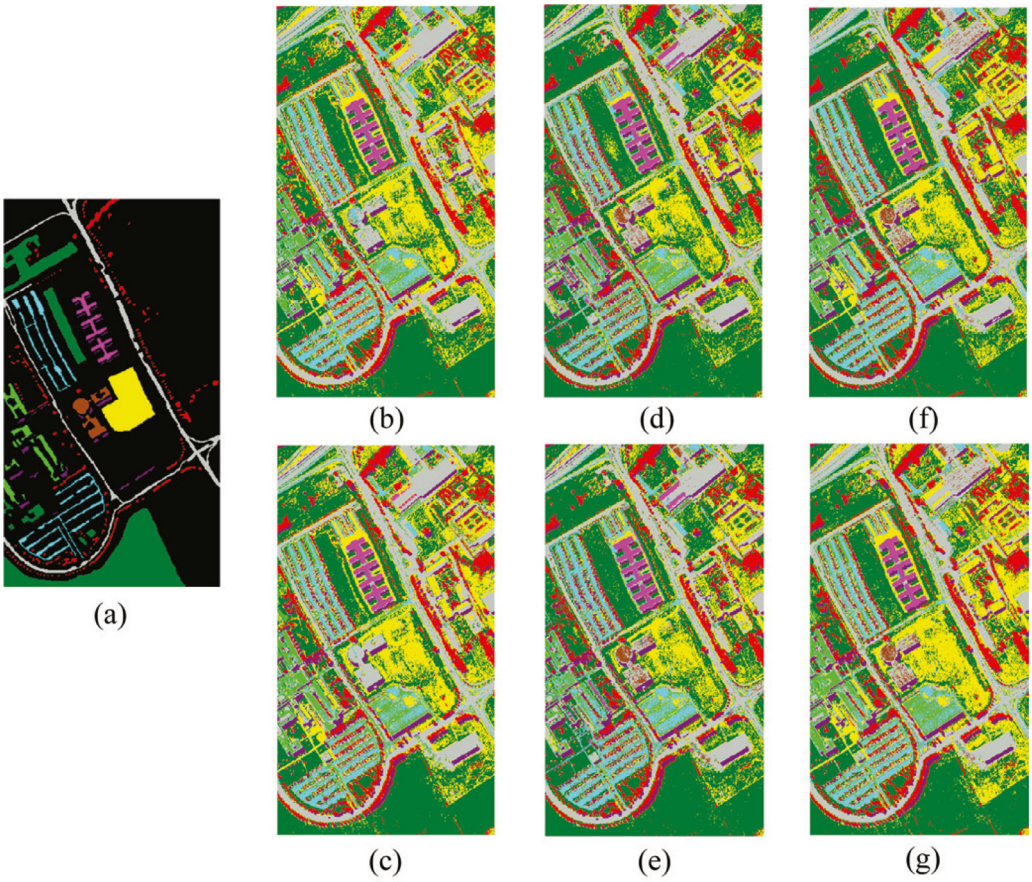

Fig. 8. Classification results of different deep learning algorithms on PaviaU data set. (a) Ground truth; (b) ANN-Original; (c) ANN-M^{3}DNet ; (d) DBN-Original; (e) DBN-M^{3}DNet; (f) SAE-Original; (g) SAE-M^{3}DNet.

a good performance. The number of neighbors for LPP was set to 9. For LGSFA, the numbers of intraclass and interclass neighbors were set as 9 and 180, respectively.

In order to evaluate the classification performance of each algorithm under a different number of training samples, $n_i (n_i = 20, 30, 40, 50, 80, 100)$ samples were randomly selected from each class for training, and the remaining samples were used as test data. Tables 2 and 3 report the average OAs with STD for different FE methods on two HSI data sets.

According to Tables 2 and 3, the OAs of all methods raise with the increase in the sample size of the training set. Deep learning methods, ANN, DBN, and SAE, are superior to subspace methods and manifold learning methods in most cases, because the latter two kinds of methods can only extract shallow features that usually fail to describe the complex nonlinear relationship of HSI data. The proposed M^{3}DNet algorithm optimizes original FE methods and achieves better classification performance than the original FE methods under most conditions, especially when the original FE

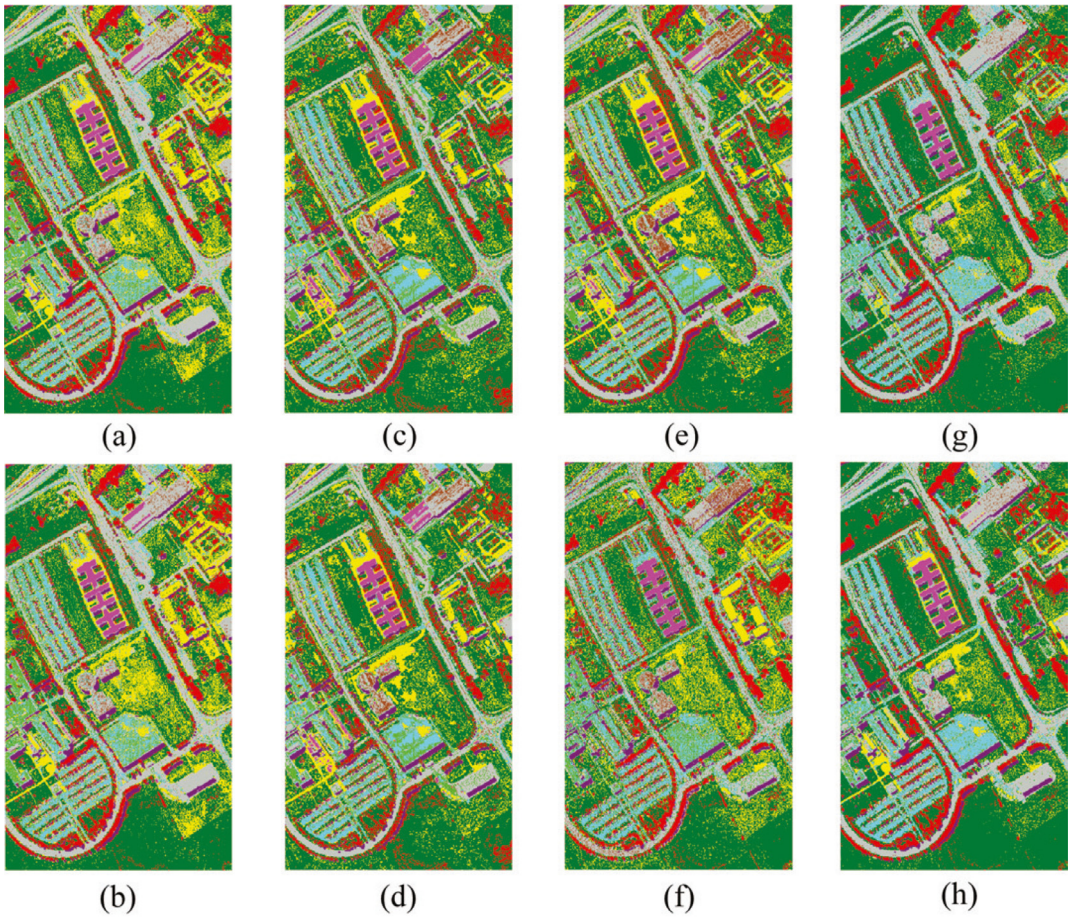

Fig. 9. Classification results of different linear subspace methods and manifold learning methods on PaviaU data set. (a) LDA-Original; (b) LDA-M^{3}DNet; (c) LFDA-Original; (d) LFDA-M^{3}DNet; (e) LPP-Original; (f) LPP-M^{3}DNet; (g) LGSFA-Original; (h) LGSFA-M^{3}DNet.

Table 8
z value in the McNemar's test for M^{3}DNet versus original FE methods.

Method	PaviaU		Indian Pines	
	z	Significant?	*z*	Significant?
LDA	15.42	yes	9.59	yes
LFDA	13.37	yes	14.23	yes
LPP	12.88	yes	4.14	yes
LGSFA	17.77	yes	4.80	yes
ANN	4.90	yes	8.48	yes
DBN	6.72	yes	4.05	yes
SAE	7.64	yes	8.13	yes

methods cannot obtain good classification performance. In Indian Pines data set, it is easily led to non-optimal classification results that the original SAE fails to explore the manifold structure in HSI. M^{3}DNet method is introduced to adaptively optimize the projection matrix obtained by SAE through maximizing manifold margin, which significantly improves the separability of different classes.

To explore the classification performance of M^{3}DNet on each class, 1% and 10% of labeled samples per class were randomly selected from PaviaU and Indian Pines data sets for training, and the remaining data were utilized for testing. Tables 4–7 report the classification accuracy of each class, OA, AA and kappa coefficient in two data sets, respectively. The corresponding classification maps of different methods are displayed in Figs. 8–11.

From Tables 4–7, it indicates that the M^{3}DNet achieved the highest classification accuracy in most classes on the two data sets than some state-of-art FE methods. For linear subspace methods and manifold learning methods, the introduction of M^{3}DNet method overcomes the problem that a single mapping process cannot obtain a better projection matrix in many cases without an iterative optimization strategy. In the process of maximizing the margins among different classes, M^{3}DNet adaptively optimized the projection matrix and enhanced the discriminative ability of low-dimensional embedding features. Meanwhile, for deep learning methods and linear subspace methods, the proposed method can further explore the manifold structure in HSI data under the GE framework. And it uses the M^{3}C to make the samples from the same class as close as possible, while the samples belonging to different classes are more separated. As shown in Figs. 8–11, the numerical results are confirmed by inspecting the classification maps. The M^{3}DNet produces fewer misclassified points and more homogenous areas because the optimized projection matrix can be used to extract more effective discriminant features for classification.

4.7. Statistical significance evaluation

The McNemar's test is employed to evaluate the statistical significance of using the proposed method to improve accuracy (Foody, 2004). Then, the McNemar's test statistic for M^{3}DNet versus original FE method can be defined as

$$z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{13}$$

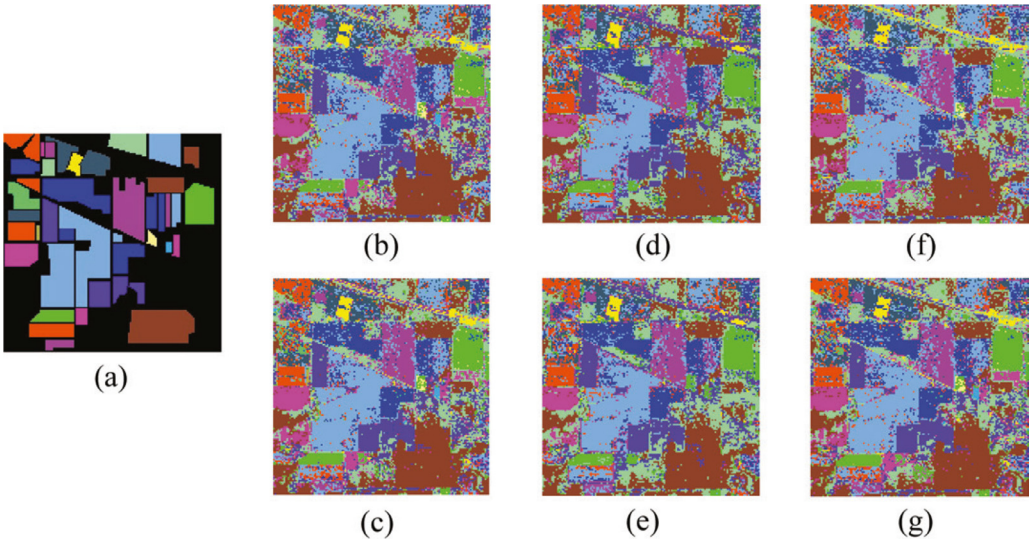

Fig. 10. Classification results of different deep learning algorithms on Indian Pines data set. (a) Ground truth; (b) ANN-Original; (c) ANN-M^{3}DNet; (d) DBN-Original; (e) DBN-M^{3}DNet; (f) SAE-Original; (g) SAE-M^{3}DNet.

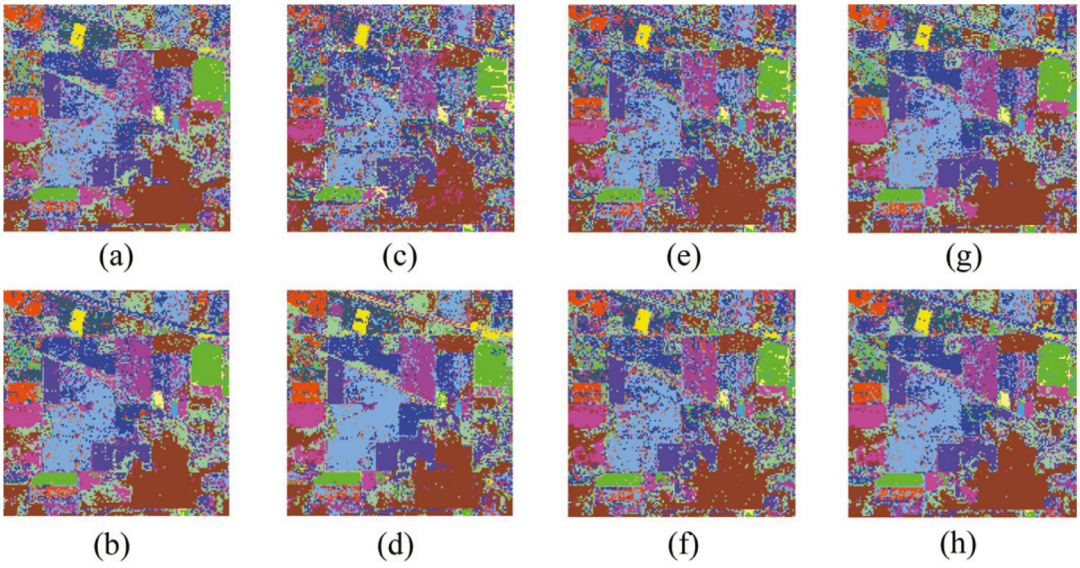

Fig. 11. Classification results of different linear subspace methods and manifold learning methods on Indian Pines data set. (a) LDA-Original; (b) LDA-M^{3}DNet; (c) LFDA-Original; (d) LFDA-M^{3}DNet; (e) LPP-Original; (f) LPP-M^{3}DNet; (g) LGSFA-Original; (h) LGSFA-M^{3}DNet.

where f_{12} denotes the number of samples misclassified via original FE method but not M^{3}DNet, and f_{21} means the number of samples misclassified by M^{3}DNet but not original FE method. The z values of McNemar's test larger than 1.96 and 2.58 mean that two results are statistically different at the 95% and 99% confidence levels, respectively. The sign z indicates whether M^{3}DNet outperforms original FE method ($z > 0$) or vice versa.

In the experiment, we randomly selected 1% and 10% of samples per class from PaviaU and Indian Pines data sets as training samples, and the rest were test samples. Tables 8 depicts z values from PaviaU and Indian Pines, respectively. A "yes" here indicates that the two methods in McNemar's test have significant performance discrepancy. Obviously, the performance of the proposed M^{3}DNet is statistically different from original FE methods, and it is demonstrated that this method can better extract the discriminant features of the land covers and improve the classification accuracy.

5. Discussion and conclusion

In view of the fact that most traditional FE methods fail to consider an iterative optimization process for the projection matrix by exploring the manifold structure in HSI data, this paper proposed a unified optimization model termed manifold-based maximization margin discriminant network (M^{3}DNet) to improve the performance of FE methods. The projection matrix obtained by traditional FE methods can be further optimized in many cases, and it is used as the initial projection matrix in M^{3}DNet. Under the GE framework, M^{3}DNet designs a maximal manifold margin criterion (M^{3}C) to enhance the intraclass compactness and the interclass separability of embedding features. Then, an adaptive process is proposed to iteratively optimize the projection matrix. Experiment results on Indian Pines and PaviaU data sets demonstrated that the M^{3}DNet can effectively improve the performance of some state-of-the-art FE methods. Our future work will focus

on how to combine the adaptive optimization process with deep learning to realize the feature extraction under unsupervised learning, which can effectively solve the problem of insufficient labeled samples for hyperspectral data.

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.

Credit authorship contribution statement

Zhengying Li: Conceptualization, Methodology, Formal analysis, Software, Writing - original draft. **Hong Huang:** Supervision, Investigation, Methodology, Validation, Formal analysis, Writing - review & editing. **Yuan Li:** Validation, Data curation, Writing - review & editing. **Yinsong Pan:** Methodology, Validation, Writing - review & editing.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments on this paper. This work was supported in part by the Basic and Frontier Research Programmes of Chongqing under Grant cstc2018jcyjAX0093, the Chongqing University Postgraduates Innovation Project under Grants CYS18035, the Fundamental Research Funds for the Central Universities under Grant 2019CDYGYB008, and the scientific and technological research project of Chongqing Education Commission under Grant KJZD-K201902501.

References

Adao, T., Hruska, J., Padua, L., Bessa, J., Peres, E., Morais, R., & Sousa, J. J. (2017). Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing, 9*, 1110.

Bachmann, C. M., Ainsworth, T. L., & Fusina, R. A. (2005). Exploiting manifold geometry in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 43*(3), 441–454.

Bayliss, J. D., Gualtieri, J. A., & Cromp, R. F. (1998). Analyzing hyperspectral data with independent component analysis. In *26th AIPR workshop: Exploiting new image sources and sensors* (pp. 133–143).

Chen, Y. S., Lin, Z. H., Zhao, X., Wang, G., & Gu, Y. F. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7*(6), 2094–2107.

Chen, Y. S., Zhao, X., & Jia, X. P. (2015). Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8*(6), 2381–2392.

Chen, Z. K., Jiang, J. J., Zhou, C., Jiang, X. W., Fu, S. Y., & Cai, Z. H. (2019). Trilateral smooth filtering for hyperspectral image feature extraction. *IEEE Geoscience and Remote Sensing Letters, 16*(5), 781–785.

Datta, A., Ghosh, S., & Ghosh, A. (2014). Maximum margin criterion based band extraction of hyperspectral imagery. In *Fourth international conference on emerging applications of information technology* (pp. 300–304).

Dian, R. W., Li, S. T., Guo, A. J., & Fang, L. Y. (2018). Deep Hyperspectral Image Sharpening. *IEEE Transactions on Neural Networks and Learning Systems, 29*(11), 5345–5355.

Du, Q., & Chang, C. (2001). A linear constrained distance-based discriminant analysis for hyperspectral image classification. *Pattern Recognition, 34*(2), 361–373.

Foody, G. M. (2004). Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy. *Photogrammetric Engineering and Remote Sensing, 70*(5), 627–633.

He, L., Li, J., Liu, C. Y., & Li, S. T. (2018). Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing, 56*(3), 1579–1597.

He, Z., Liu, H., Wang, Y. W., & Hu, J. (2017). Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sensing, 9*(10), 1042.

Hong, D. F., Yokoya, N., Chanussot, J., & Zhu, X. X. (2019). Cospace: Common subspace learning from hyperspectral-multispectral correspondences. *IEEE Transactions on Geoscience and Remote Sensing, 57*(7), 4349–4359.

Huang, H., Duan, Y. L., Shi, G. Y., & Lv, Z. Y. (2018). Fusion of weighted mean reconstruction and svmck for hyperspectral image classification. *IEEE Access,, 6*, 15224–15235.

Huang, H., Li, Z. Y., & Pan, Y. S. (2019). Multi-feature manifold discriminant analysis for hyperspectral image classification. *Remote Sensing, 11*(6), 651.

Jiao, C. Z., Chen, C., McGarvey, R. G., Bohlman, S., Jiao, L. C., & Zare, A. (2018). Multiple instance hybrid estimator for hyperspectral target characterization and sub-pixel target detection. *ISPRS Journal of Photogrammetry and Remote Sensing, 146*, 235–250.

Kang, X. D., Duan, P. H., Li, S. T., & Benediktsson, J. A. (2018). Decolorization-based hyperspectral image visualization. *IEEE Transactions on Geoscience and Remote Sensing, 56*(8), 4346–4360.

Li, H. Z., Li, H., & Zhang, L. M. (2019). Quaternion-based multiscale analysis for feature extraction of hyperspectral images. *IEEE Transactions on Signal Processing, 67*(6), 1418–1430.

Li, W., Zhang, L. P., Zhang, L. F., & Du, B. (2017). GPU parallel implementation of isometric mapping for hyperspectral classification. *IEEE Geoscience and Remote Sensing Letters, 14*(9), 1532–1536.

Li, X. R., Pan, J., He, Y. Q., & Liu, C. S. (2015). Bilateral filtering inspired locality preserving projections for hyperspectral images. *Neurocomputing, 164*, 300–306.

Liang, M. M., Jiao, L. C., Yang, S. Y., Liu, F., Hou, B., & Chen, H. (2018). Deep multi-scale spectral-spatial feature fusion for hyperspectral images classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(8), 2911–2924.

Lin, Y., & Niu, X. T. (2014). Spectral-angle-based laplacian eigenmaps for nonlinear dimensionality reduction of hyperspectral imagery. *Photogrammetric Engineering and Remote Sensing, 80*(9), 849–861.

Liu, J., Gong, M. G., & He, H. B. (2019). Deep associative neural network for associative memory based on unsupervised representation learning. *Neural Networks, 113*, 41–53.

Lu, G. F., Jin, Z., & Zou, J. (2012). Face recognition using discriminant sparsity neighborhood preserving embedding. *Knowledge-Based Systems, 31*, 119–127.

Lu, J. W., Wang, G., Deng, W. H., Moulin, P., & Zhou, J. (2015). Multi-manifold deep metric learning for image set classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2271–2282).

Luo, F. L., Du, B., Zhang, L. P., Zhang, L. F., & Tao, D. C. (2018). Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image. *IEEE Transactions on Cybernetics, 49*(7), 2406–2419.

Luo, F. L., Huang, H., Duan, Y. L., Liu, J. M., & Liao, Y. H. (2017). Local geometric structure feature for dimensionality reduction of hyperspectral imagery. *Remote Sensing, 9*(8), 6197–6211.

Ma, L., Liu, Y., Zhang, X. L., Ye, Y. X., Yin, G. F., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing, 152*, 166–177.

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing, 39*(9), 2784–2817.

Mostafa, H. (2017). Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 29*(7), 3227–3235.

Peng, J. T., Sun, W. W., & Du, Q. (2019). Self-paced joint sparse representation for the classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing, 57*(2), 1183–1194.

Qian, Y. T., Xiong, F. C., Zeng, S., Zhou, J., & Tang, Y. Y. (2017). Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 55*(3), 1776–1792.

Ratle, F., Camps-Valls, G., & Weston, J. (2010). Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 48*(5), 2271–2282.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science,, 290*(5500), 2323–2326.

Sellami, A., Farah, M., Farah, I. R., & Solaiman, B. (2019). Hyperspectral imagery classification based on semi-supervised 3-d deep neural network and adaptive band selection. *Expert Systems with Applications, 129*, 246–259.

Song, W. W., Li, S. T., Fang, L. Y., & Lu, T. (2018). Hyperspectral image classification with deep feature fusion network. *IEEE Transactions on Geoscience and Remote Sensing, 56*(6), 3173–3184.

Su, H. J., Zhao, B., Du, Q., & Du, P. J. (2019). Kernel collaborative representation with local correlation features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 57*(2), 1230–1241.

Sugiyama, M., Ide, T., Nakajima, S., & Sese, J. (2010). Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine Learning, 78*(12), 35–61.

Tang, M. F., Zhang, B., Marinoni, A., Gao, L. R., & Gamba, P. (2018). Multiharmonic postnonlinear mixing model for hyperspectral nonlinear unmixing. *IEEE Geoscience and Remote Sensing Letters, 15*(11), 1765–1769.

Thompson, W. D., & Walter, S. D. (1988). A reappraisal of the kappa coefficient. *Journal of Clinical Epidemiology,, 41*(10), 949–958.

Tyo, J. S., Konsolakis, A., Diersen, D. I., & Olsen, R. C. (2003). Principal-components-based display strategy for spectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 41*(3), 708–718.

Wang, Z. M., Du, B., Zhang, L. F., Zhang, L. P., & Jia, X. P. (2017). A novel semisupervised active-learning algorithm for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 55*(6), 3071–3083.

Yan, S. C., Xu, D., Zhang, B. Y., Zhang, H. J., Yang, Q., & Lin, S. (2007). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(1), 40–51.

Zabalza, J., Ren, J. C., Zheng, J. B., Han, J. W., Zhao, H. M., Li, S. T., & Marshall, S. (2015). Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging. *IEEE Transactions on Geoscience and Remote Sensing, 53*(8), 4418–4433.

Zhang, L., Wei, W., Bai, C. C., Gao, Y. F., & Zhang, Y. N. (2018). Exploiting clustering manifold structure for hyperspectral imagery super-resolution. *IEEE Transactions on Image Processing, 27*(12), 5969–5982.

Zhang, M. Y., Gong, M. G., Mao, Y. S., Li, J., & Wu, Y. (2019a). Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing, 57*(5), 2669–2688.

Zhang, P., He, H. X., & Gao, L. R. (2019b). A nonlinear and explicit framework of supervised manifold-feature extraction for hyperspectral image classification. *Neurocomputing, 337*(14), 315–324.

Zhang, Y. X., Ke, W., Du, B., Zhang, L. P., & Hu, X. Y. (2017). Hyperspectral target detection via adaptive joint sparse representation and multi-task learning with locality information. *Remote Sensing, 9*(5), 482.

Zhong, Y. F., Wang, X. Y., Xu, Y., Wang, S. Y., Jia, T. Y., Hu, X., … Wei, L. F. (2018). Mini-uav-borne hyperspectral remote sensing: From observation and processing to applications. *IEEE Geoscience and Remote Sensing Magazine, 6*(4), 46–62.

Zhou, S. G., Xue, Z. H., & Du, P. J. (2019). Semisupervised stacked autoencoder with cotraining for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 57*(6), 3813–3826.