Combining t-distributed stochastic neighbor embedding with convolutional neural networks for hyperspectral image classification.

GAO, L., GU, D., ZHUANG, L., REN, J., YANG, D. and ZHANG, B.

2020

Combining t-Distributed Stochastic Neighbor Embedding with Convolutional Neural Networks for Hyperspectral Image Classification

Lianru Gao, *Senior Member, IEEE,* Daixin Gu, Lina Zhuang, Jinchang Ren, *Senior Member, IEEE,* Dong Yang, and Bing Zhang, *Senior Member, IEEE* .

***Abstract*—Hyperspectral images (HSIs), featured by a high spectral resolution over a wide range of the electromagnetic spectrum, have been widely used to characterize materials with subtle difference in the spectral domain. However, a large number of bands and insufficient number of sample pixels for each class are challenging for traditional machine learning-based classifiers. As alternative tools for feature extraction, neural networks have received extensive attention. This letter proposes to combine t-distributed stochastic neighbor embedding (t-SNE) with convolutional neural network (CNN) for HSI classification. Our framework is designed to automatically capture potential assembly features which are extracted from both the dimension-reduced CNN (DR-CNN) and multiscale-CNN. Experimental results show that the proposed classification framework out-performs several state-of-the-art techniques on three real datasets.**

***Index Terms*—hyperspectral image classification, t-distributed stochastic neighbor embedding, convolutional neural network, dimensionality reduction, assembly fusion.**

I. Introduction

With the development of hyperspectral imaging, increasing spatial-spectral resolution provides rich information for classification. In existing works, random forest (RF) [1], support vector machine (SVM) [2] and sparse representations (SR) [3] have been considered as efficient algorithms for feature extraction and classification in HSIs. However it is difficult to obtain high accuracy from classifying directly original hyperspectral data, due mainly to the Hughes phenomenon, i.e. insufficient number of pixel based samples in comparison to the high dimensionality of the spectral data [4]. Meanwhile, although high spectral resolution leads to increased inter-class variation, the high dimensional data also increases the computational complexity and limits the separability of traditional methods, leading to relatively poor classification performances. This is mainly due to the extract shallow features fails to represent the essential taxonomic features of each class in HSI [5].

In order to address the challenge of high-dimensionality in HSI, manifold learning becomes a hot topic in HSI classification since 2000 [6]. By projecting HSI into low-dimensional spaces, manifold learning can find the intrinsic structure of differently distributed data and suppress the noise. The machine-learning community has demonstrated the potential of manifold-based approaches for nonlinear dimensionality reduction [7]. While, manifold learning inevitably lost some crucial information in data projecting.

For HSI classification, the most important step is to extract high-quality features. Recently, deep learning has received extensive attention in hyperspectral classification due to its capability to learn and represent more meaningful features hierarchically. Compared with state-of-the-art traditional methods, deep learning can extract higher-level and more robust features [8]. A number of deep learning networks have been successfully applied in hyperspectral classification. [9] proposed promising and novel auto-encoder based deep neural networks, which for the first time introduced stacked convolutional denoising and auto-encoder mechanism into high-dimension data featrue representation. Different network structures can extract various discriminate features for classification of HSI. In [10], a CNN with five convolutional layers was used to extract spectral features. However, spectral features carry great intra-class variation and inter-class similarity and ignore specific spatial information to some degree. As a result, the work in [11] used 2-D convolutional kernels to extract the spectral-spatial features for effectively HSI classification. [12] proposed multiscale features fusion based on different spatial structures containing various texture features due to plentiful neighbourhood association. While HSI has noise in the acquired datasets, with fully training, the CNN can not only extract spatial-spectral features to improve accuracy without information loss, it may also learn noise information hence decrease classification results.

With the consideration of these advantages and disadvantages of manifold learning and CNN, we propose a combination of dimension-reduced CNN features (DR-CNN features), learned by a manifold learning method and CNN called dimension-reduced CNN (DR-CNN) [13], and spatial-

This paper was submitted in XXXX, 2019. This work was supported by the National Natural Science Foundation of China under Grant No. 41722108 and No. 91638201. (Corresponding author: *Bing Zhang)*

L. Gao is with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: gaolr@radi.ac.cn).

D. Gu and B. Zhang are with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China. D. Gu is with School of Eletronic, Electrical and Communication Engineering, the University of Chinese Academy of Sciences, Beijing 100049, China. And B. Zhang is with College of Resources and Environment, the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: gudx@radi.ac.cn; zb@radi.ac.cn)

L. Zhuang is with Department of Mathematics, Hong Kong Baptist University, HongKong, China (email: linazhuang@hkbu.edu.hk).

J. Ren is with Strathclyde Hyperspectral Imaging Centre, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK (e-mail: jinchang.ren@strath.ac.uk).

D. Yang is with Institute of Spacecraft System Enginering (CAST), No.104 Youyi Road, Haidian, Beijing, China (email: qbdyzy@sina.com).

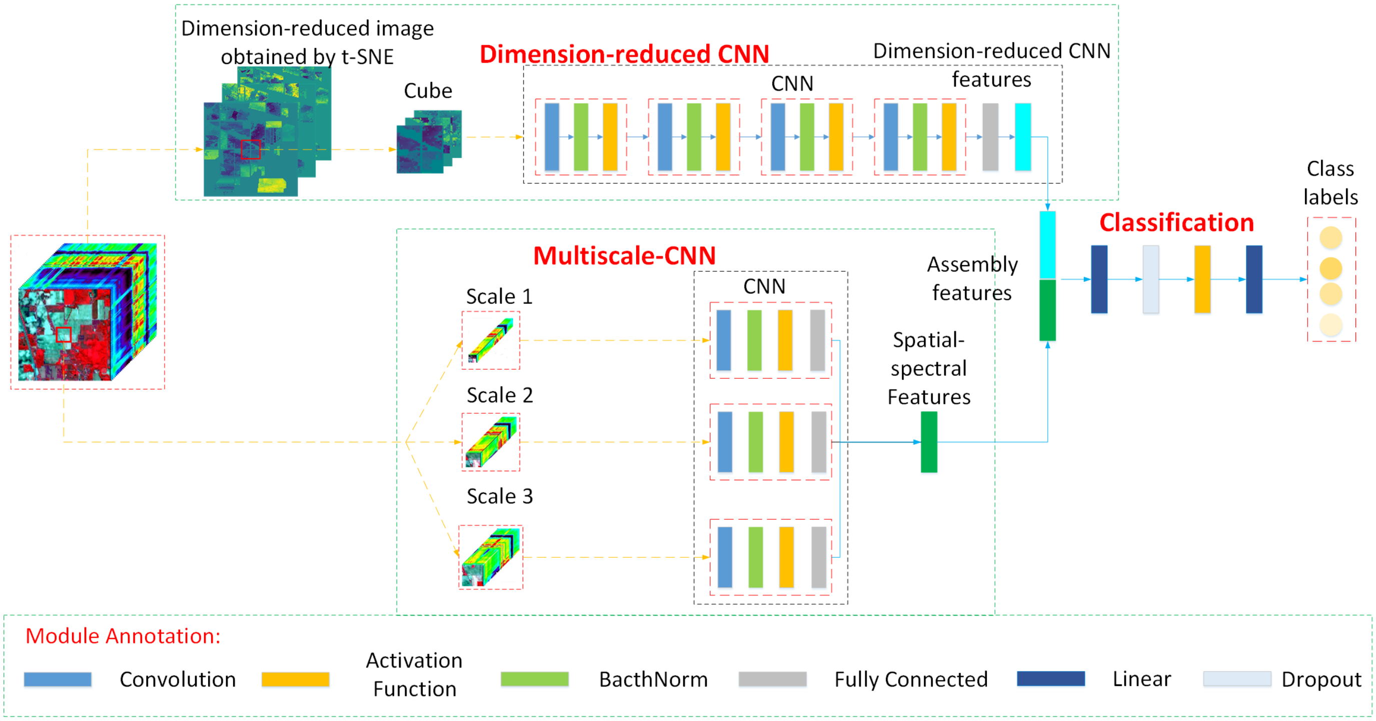

Fig. 1. t-SNE-CNN. The feature fusion based classification framework.

spectral features, learned by a deep learning newtowrk called multiscale-CNN, to improve the classification accuracy of HSI. Firstly, we implement t-SNE to obtain a dimension-reduced hyperspectral image. Then CNNs are designed to extract high-level features from the dimension-reduced image and also from the original HSI with multiscale scheme. Finally, assembly features extracted by CNNs are used for classification.

The remaining parts of this letter are organized as follows. Section II introduces the framework of t-SNE-CNN. Section III presents experimental results including comparisons with the state-of-the-art. Section IV summarizes some concluding remarks.

II. The Proposed t-SNE Deep Learning Framework

A. DR-CNN: Feature Extraction Based on t-SNE Dimensionality Reduction and CNN

Given a high-dimensional hyperspectral image matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{C \times N}$ with N spectral vectors (the columns of $\mathbf{X}$) of size C, the t-SNE algorithm converts $\mathbf{X}$ into a low-dimensional matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] \in \mathbb{R}^{D \times N}(D < C)$. Two conditional probability distributions are defined as follows:

$$P(\mathbf{x}_i|\mathbf{x}_j) = \frac{S(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k \neq i}^{N} S(\mathbf{x}_i, \mathbf{x}_k)} \tag{1}$$

$$Q(\mathbf{y}_i|\mathbf{y}_j) = \frac{S(\mathbf{y}_i, \mathbf{y}_j)}{\sum_{k \neq i}^{N} S(\mathbf{y}_i, \mathbf{y}_k)} \tag{2}$$

where $S(\cdot)$ denotes the Euclidean distance between two vectors of sample pixels. To satisfy the distribution of conditional probability P and Q being as equal as possible for all sample pixels, the Kullback-Leibler (KL) divergence in (3) need be as small as possible:

$$KL = \sum_i \sum_j P(\mathbf{x}_i, \mathbf{x}_j) \log \frac{P(\mathbf{x}_i|\mathbf{x}_j)}{Q(\mathbf{y}_i|\mathbf{y}_j)}. \tag{3}$$

The t-SNE algorithm obtains the optimal dimensionality reduction by calculating the minimum value of KL divergence between the joint conditional probability of the original space and the embedded space. The non-linear dimension-reduction algorithm t-SNE finds the structure of data by identifying the pattern based on the similarity of data points with multiple characteristics, focusing on the local structure of data. Actually, in order to speed up the calculation, we use “Barnes-Hut t-SNE” instead of the traditional “t-SNE” method. Parameters are set as follows:

- The dimension of reduced data $\mathbf{Y}$ is set to $D = 3$, which is the same as that in [14]. This is because larger D can hardly improve the classification accuracy as validated in our experiments.
- The parameter of perplexity, denoted as α, is set as 50. Larger datasets usually require a larger perplexity to use more nearest neighbors information.
- The remaining parameters are consistent with the default setting of “TSNE” in the sklearn codebase (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html).

The DR-CNN feature extraction model is shown in Figure 1 of the block dimension-reduced CNN. The t-SNE algorithm maps hyperspectral data from the original high-dimensional manifold space to a low-dimensional space, followed by four convolutional modules. Considering that the t-SNE algorithm is focusing more on the local structure of the hyperspectral image, we extract large scale data cubes of size $41 \times 41 \times D$ as input data, where D represents the number of reduced dimension. The size of extracted cubes is the same as that in [14]. Meanwhile, the size of convolutional kernels is set to 3×3. Finally, the network outputs DR-CNN features. The network operates in a similar manner as Multiscale-CNN which are elabrated in the next subsection.

B. Multiscale-CNN: Spatial-spectral Feature Extraction from the Original HSI

In order to fully extract spatial-spectral features from the original HSIs [15], we propose to adopt multiscale convolution kernels. The CNN computes robust spatial-spectral features by applying various filters in multiple hidden layers. Considering different ground objects have different spectral reflection profiles, the 2D convolutional kernel size is 1×1 to extract spectral features. Meanwhile, 3×3-sized convolutional kernels are exploited to effectively extract spatial information based on local correlation. In the flowchart of multiscale-CNN, the input data are calculated by multiple convolution kernels, then activated by an activation function. Finally, features are concatenated in a fully connected layer. A typical hyperspectral convolution calculation formula is given by:

$$\mathbf{v}_{i,j} = f_{fc}(f_{ac}(w_c * \mathbf{X}_{i,j})) \tag{4}$$

where $\mathbf{v}_{i,j}$ represents the feature vector of the input data at position (i, j) in the HSI, f_{fc} is the fully connected function which transforms the data into one-dimensional vectors, and f_{ac} is an activation function. w_c are multiple convolutional kernels, combining with the input data $\mathbf{X}_{i,j}$ through the operation of convolution denoted as $*$. The parametric rectified linear unit is applied as a non-linear activation function, which is defined by:

$$f_{ac} = \begin{cases} x_i, & if \quad x_i > 0, \\ a_i x_i, & if \quad x_i \leq 0. \end{cases} \tag{5}$$

where a_i is automatically updated in the process of training CNN.

The structure of multiscale-CNN is shown in Figure 1. It can be divided into two main components, including extraction of spectral features and spatial features. The spectral features are extracted by utilizing $1 \times 1 \times L$ convolutional kernels in scale 1 where L represents the number of convolutional kernels. The framework designs $3 \times 3 \times L$ convolutional kernels to extract spatial information from $3 \times 3 \times C$-sized cube and $9 \times 9 \times C$-sized cube which composed by center pixel and its neighbourhoods pixels from original hyperspectral image. This network adopts $3 \times 3 \times C$-sized cube in scale 2, and uses a $9 \times 9 \times C$-sized cube in scale 3. In the final step, different scale features are concatenated as spatial-spectral features.

C. t-SNE-CNN: Classification by Combining the Two Kinds of Features

As shown in Figure 1, spatial-spectral features and t-SNE features are combined together to classify HSI. The network predicts class of assembly features by applying linear layers and an activation function layer. In this work, linear layers mean features transformation which combines two kinds of features and adjusts the number of assembly features. To enhance the robustness of classification results, dropout is introduced to force randomly some nodes of each layers to zero in every training session. The final layer outputs class labels.

This letter proposes a deep learning classification method to extract assembly features which combined spatial-spectral features of multiscale-CNN and dimensionality reduction features of t-SNE algorithm. The next section details the parameter settings of the proposed approach and the classification results on three real hyperspectral images.

III. Experiment Result

A. Datasets

There are three datasets used in evaluating the performance of the proposed method for classification of HSI, including the Indian Pines dataset, the University of Pavia dataset, and the Salinas dataset.

1) Indian Pines: We use remaining 200 bands in 0.4-2.5 um with a spatial resolution of 20 m. It is composited by 16 different land-cover classes with 10249 labelled pixels. Each class is divided randomly into 10% for training the network and 90% for testing the performance of t-SNE-CNN.

2) University of Pavia: We use 103 bands in 0.43-0.86 um with a spatial resolution of 1.3 m. It is composited by 9 different land-cover classes with 42776 labelled pixels. The individual class is divided randomly into 3% for training data and 97% for testing data.

3) Salinas: We use 204 bands in 0.4-2.4 um with a spatial resolution of 3.7 m. It is composited by 16 different land-cover classes with 54129 labelled pixels. Each class is divided randomly into 1% training data and 99% testing data.

B. Baselines

To evaluate the performance of the proposed t-SNE-CNN, we compare it with two widely-used conventional machine learning methods, namely SVM [16] and RF-200 [1] with 200 trees. Meanwhile, deep learning methods, 1D-CNN [11] exploiting spectral information and DC-CNN [14] exploiting spatial-spectral information, are also compared. These experiments just follow default parameters in cited documents to make the results more consistently for comparison. Furthermore, in order to see the individual effect of DR-CNN features and spatial-spectral features extracted by the proposed method, we show their classification results, called 'DR-CNN' and 'multiscale-CNN', respectively, when using only one kind of features.

For quantitative assessment, the overall accuracy (OA) of all classes, the average accuracy (AA) of each class, and the Kappa (consistency of classification results based on confusion matrix) are calculated.

The optimization function of t-SNE-CNN network is Adam [17] with an initial learning rate of 0.005. For each 10 training epochs, the learning rate decays to 90% of the previous one. Meanwhile, the batch size of the training data is set as 100 in

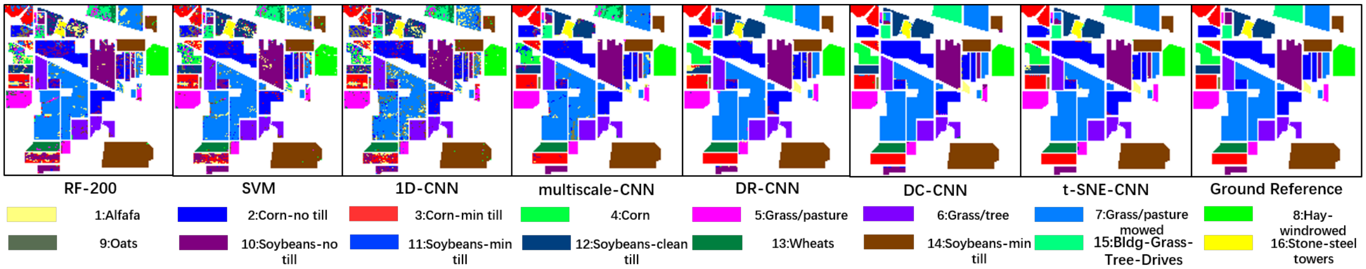

Fig. 2. Indian Pines. Classification result graphs of different methods and a standard classification result graph of Indian Pines.

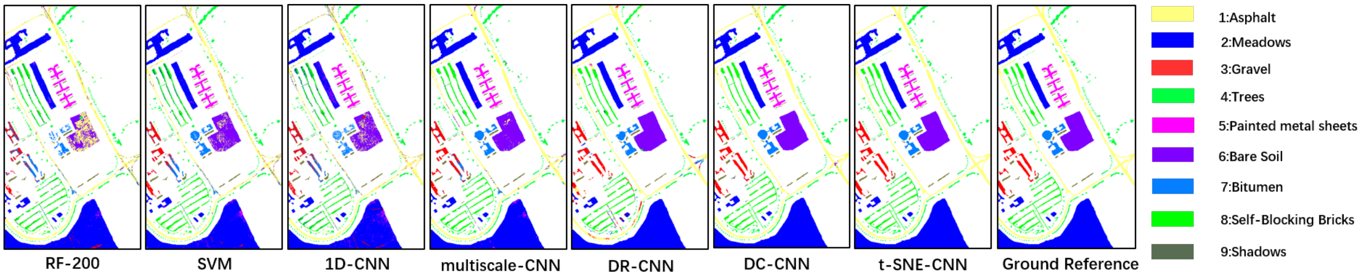

Fig. 3. University of Pavia. Classification result graphs of different methods and a standard classification result graph of University of Pavia.

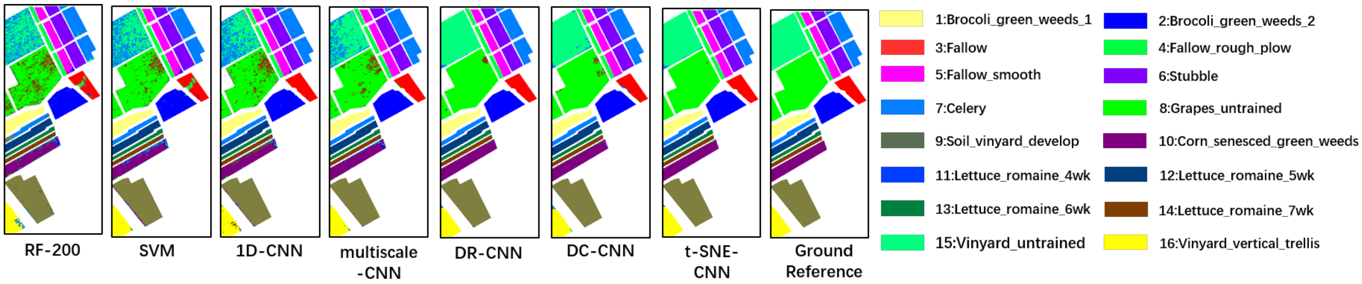

Fig. 4. Salinas. Classification result graph of different methods and a standard classification result graph of Salinas.

every training epoch. For optimum parameters, t-SNE-CNN network chooses CrossEntropyLoss as the loss function and initializes network parameters by 'Kaiming_normal' method. In Table I, network configuration are specified with the parameters.

TABLE I
PARAMETERS OF PROPOSED T-SNE-CNN NETWORK

t-SNE-CNN Network	Indain Pines	University of Pavia	Salinas
Scale 1	Convolutional kernel size: 1×1, number of kernels: 16, stride=1 Activation function: PRelu()		
Scale 2	Convolutional kernel size: 3×3, number of kernels: 16, stride=1 Activation function: PRelu()		
Scale 3	Convolutional kernel size: 3×3, number of kernels: 16, stride=2 Activation function: PRelu()		
t-SNE	Convolutional kernel size: 3×3, number of kernels: 40, stride=2 Activation function: PRelu()		
Classification	Dropout: 0.5 Activation function: PRelu()		

C. HSI classification

As shown in the Figure 2, Figure 3 and Figure 4, the classification maps of t-SNE-CNN are visually better than those from other methods. Obviously, t-SNE-CNN with spatial-spectral information from dimensionality reduced HSI and the original HSI provides higher feature extraction capability in classification. As illustrated in detail in Table II, several observations are summarized as follows concisely: (1) In three HSIs, t-SNE-CNN yields uniformly the best performance in terms of OA, AA, and Kappa indexes; (2) t-SNE-CNN outperforms DR-CNN and multiscale-CNN, implying the fusion of two kinds of features (i.e., DR-CNN features used in DR-CNN method and spatial-spectral features used in multiscale-CNN) can improve classification accuracy than using only one of them; (3) In particular, the t-SNE employing dimension-reduced HSI boosts the separability between categories compared with SVM, RF-200 and 1D-CNN using original HSI; (4) Basically, neural network-based methods, except 1D-CNN, achieve higher classification accuracy than two conventional machine learning methods, i.e., SVM and RF-200.

The classification accuracy of each class is compared in Figure 5. As seen, t-SNE-CNN (in red color) uniformly achieves higher accuracy in each class and has great improvement on some classes with low accuracy in contrast methods, e.g. class 9 in Indian Pines, class 3 in University of Pavia data, and class 11 in Salinas data. Furthermore, the red lines exceed the blue lines and the pink lines, implying that t-SNE-CNN can fully integrate two features to improve classification performance.

IV. CONCLUSION

In order to deal with Hughes phenomenon and to take full advantage of spectral-spatial information in HSIs, we propose

TABLE II
CLAFFISICATION OF THREE DATA SETS FOR DIFFERENT CLASSIFICATION METHODS. (REPEATING TEN TIMES TO CALCULATE MEAN VALUE.)

Data set	Indian Pines			University of Pavia			Salinas		
Evaluation Index	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
t-SNE-CNN	**98.94%**	**97.89%**	**98.79%**	**99.74%**	**99.60%**	**99.65%**	**99.26%**	**99.35%**	**99.18%**
DC-CNN [14]	98.07%	95.50%	97.80%	99.14%	98.71%	98.86%	97.84%	98.39%	97.59%
DR-CNN	97.38%	94.93%	97.02%	96.33%	92.50%	95.12%	98.73%	98.65%	98.58%
multiscale-CNN	92.87%	87.42%	91.86%	98.32%	97.06%	97.78%	93.44%	96.14%	92.70%
1D-CNN [11]	74.53%	66.36%	70.69%	91.00%	88.30%	88.01%	88.99%	92.71%	87.70%
SVM [16]	80.59%	74.81%	77.75%	92.24%	89.39%	89.62%	89.27%	92.54%	88.01%
RF-200 [1]	75.71%	61.89%	71.97%	85.41%	80.59%	80.16%	86.56%	90.65%	85.01%

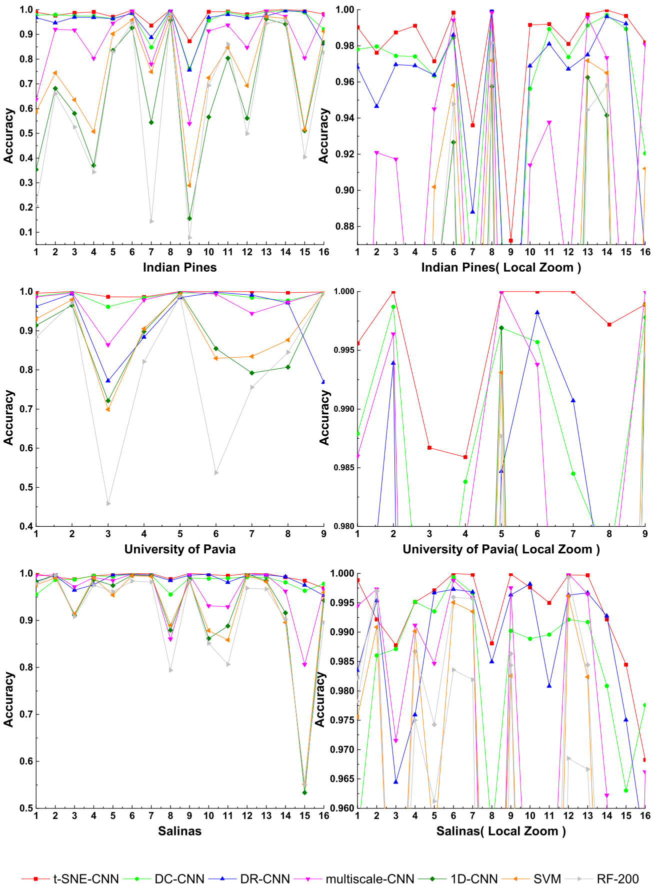

Fig. 5. Classification accuracy of each class : Indian Pines, University of Pavia, Salinas. The right column represents the classification result of zooming in the upper part of the image corresponding to the left column.

in this letter a CNN based classification framework exploiting assembling features extracted from both the dimension-reduced data and the original image with multiscale scheme. The experiments using three real HSI datasets demonstrate that assembly features achieve better classification results than individual features. A comparison of t-SNE-CNN with the state-of-the-art algorithms is conducted, leading to the conclusion that t-SNE-CNN outperform several conventional and deep learning based methodologies.

REFERENCES

[1] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classifiers for hyperspectral data," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 1, July 2005, pp. 160–163.

[2] J. Zabalza, J. Ren, J. Zheng, J. Han, H. Zhao, S. Li, and S. Marshall, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4418–4433, Aug 2015.

[3] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, Oct 2011.

[4] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, January 1968.

[5] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, April 2016.

[6] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 55–66, Jan 2014.

[7] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 441–454, March 2005.

[8] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, June 2016.

[9] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 1017–1027, Apr 2017.

[10] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, pp. 1–12, January 2015.

[11] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct 2016.

[12] Z. Li, L. Huang, D. Zhang, C. Liu, Y. Wang, and X. Shi, "A deep network based on multiscale spectral-spatial fusion for hyperspectral classification," in *Knowledge Science, Engineering and Management*, August 2018, pp. 283–290.

[13] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, pp. 2579–2605, Nov 2008.

[14] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sensing Letters*, vol. 8, no. 5, pp. 438–447, Jan 2017.

[15] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, Sep 2018.

[16] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 1, July 2003, pp. 288–290.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, 2014.