

# Feature Extraction With Multiscale Covariance Maps for Hyperspectral Image Classification

Nanjun He, *Student Member, IEEE*, Mercedes E. Paoletti, *Student Member, IEEE*,  
 Juan Mario Haut<sup>id</sup>, *Student Member, IEEE*, Leyuan Fang<sup>id</sup>, *Senior Member, IEEE*,  
 Shutao Li<sup>id</sup>, *Senior Member, IEEE*, Antonio Plaza<sup>id</sup>, *Fellow, IEEE*,  
 and Javier Plaza<sup>id</sup>, *Senior Member, IEEE*

**Abstract**—The classification of hyperspectral images (HSIs) using convolutional neural networks (CNNs) has recently drawn significant attention. However, it is important to address the potential overfitting problems that CNN-based methods suffer when dealing with HSIs. Unlike common natural images, HSIs are essentially three-order tensors which contain two spatial dimensions and one spectral dimension. As a result, exploiting both spatial and spectral information is very important for HSI classification. This paper proposes a new hand-crafted feature extraction method, based on multiscale covariance maps (MCMs), that is specifically aimed at improving the classification of HSIs using CNNs. The proposed method has the following distinctive advantages. First, with the use of covariance maps, the spatial and spectral information of the HSI can be jointly exploited. Each entry in the covariance map stands for the covariance between two different spectral

bands within a local spatial window, which can absorb and integrate the two kinds of information (spatial and spectral) in a natural way. Second, by means of our multiscale strategy, each sample can be enhanced with spatial information from different scales, increasing the information conveyed by training samples significantly. To verify the effectiveness of our proposed method, we conduct comprehensive experiments on three widely used hyperspectral data sets, using a classical 2-D CNN (2DCNN) model. Our experimental results demonstrate that the proposed method can indeed increase the robustness of the CNN model. Moreover, the proposed MCMs+2DCNN method exhibits better classification performance than other CNN-based classification strategies and several standard techniques for spectral-spatial classification of HSIs.

**Index Terms**—Data augmentation, deep convolutional neural networks (CNNs), hyperspectral image (HSI) classification, multiscale covariance maps (MCMs).

Manuscript received May 25, 2018; revised June 27, 2018; accepted July 20, 2018. This paper was supported in part by the National Natural Science Fund of China for International Cooperation and Exchanges under Grant 61520106001, in part by the National Natural Science Foundation for Young Scientist of China under Grant No. 61501180, in part by the National Natural Science Foundation under Grant 61771192 and Grant 61471167, in part by the Fund of Hunan Province for Science and Technology Plan Project under Grant 2017RS3024, in part by China Scholarship Council, in part by the Ministerio de Educación (Resolución de 26 de diciembre de 2014 y de 19 de noviembre de 2015, de la Secretaría de Estado de Educación, Formación Profesional y Universidades, por la que se convocan ayudas para la formación de profesorado universitario, de los subprogramas de Formación y de Movilidad incluidos en el Programa Estatal de Promoción del Talento y su Empleabilidad, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016), in part by the Junta de Extremadura (decreto 297/2014, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR15005), and in part by MINECO under Project TIN2015-63646-C5-5-R. (*Corresponding author: Leyuan Fang.*)

N. He is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, also with the Key Laboratory of Visual Perception and Artificial Intelligence of Hunan Province, Changsha 410082, China, and also with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, PC-1003 Cáceres, Spain (e-mail: henanjun@hnu.edu.cn).

M. E. Paoletti, J. M. Haut, A. Plaza, and J. Plaza are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, PC-10003 Cáceres, Spain (e-mail: mpaoletti@unex.es; juanmariohaut@unex.es; aplaza@unex.es; jplaza@unex.es).

L. Fang and S. Li are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the Key Laboratory of Visual Perception and Artificial Intelligence of Hunan Province, Changsha 410082, China (e-mail: fangleyuan@gmail.com; shutao\_li@hnu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2018.2860464

## I. INTRODUCTION

RECENTLY, convolutional neural network (CNN)-based hyperspectral image (HSI) classification has become an active topic in the hyperspectral community [1]–[6]. This is mainly due to the fact that CNN models can achieve state-of-the-art performance on many different computer vision tasks [7] (e.g., image classification, object detection, and semantic segmentation).

As it is well known, it is necessary to obtain adequate training data for a CNN model to achieve powerful generalization and good classification performance [8]. However, HSI data sets usually suffer from the limited availability of labeled samples [1], [5], [9]. In general, the training sets available for HSIs are much smaller than those available for other image databases used in the computer vision community (e.g., ImageNet [10] contains millions of samples for training). Under this context, it is necessary to develop effective methods for CNN-based classification that can alleviate the potential overfitting problems that these architectures exhibit when dealing with HSI data [11]. Moreover, different from common natural images, HSIs are essentially three-order tensors which contain two spatial dimensions and one spectral dimension [12], [13]. It is important to integrate the spatial and spectral information for HSI classification [14]–[16].

With the aforementioned ideas in mind, this paper proposes a new hand-crafted feature extraction method based on multiscale covariance maps (MCMs) for CNN-based HSI classification. The proposed method has the following two

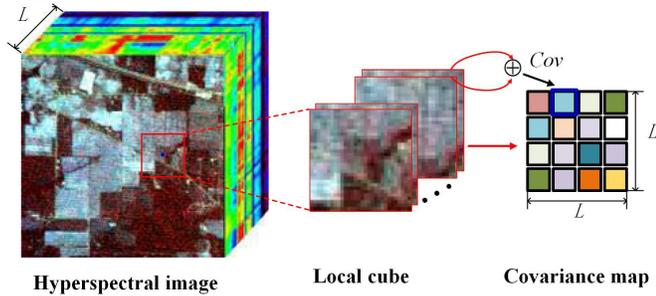


Fig. 1. Graphical illustration of the construction of a covariance map for a center pixel, where  $L$  is the number of spectral bands.  $Cov$  denotes the covariance operation. For two given vectors  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  and  $\mathbf{y} = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$ , we have  $Cov(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1} \sum_{i=1}^n z_i - (\frac{1}{n-1} \sum_{i=1}^n x_i) \times (\frac{1}{n-1} \sum_{i=1}^n y_i)$ , where  $z_i = x_i y_i$ .

distinctive characteristics. First, the covariance maps can fully exploit the spatial–spectral information present in HSIs. Fig. 1 shows an illustrative diagram explaining the construction of the covariance map (on one scale) for an HSI. As it can be observed, each entry in the covariance map stands for the covariance of two different spectral bands within a local spatial window, which offers a natural way to simultaneously exploit the spatial–spectral information in the HSI [17]. Second, with our multiscale strategy, each sample can be enhanced with the information coming from different scales, which is a natural way to increase the information comprised by training samples effectively. We also note that the size of the extracted covariance map is only determined by the number of spectral bands (see Fig. 1), which indicates that the covariance maps coming from different scales share the same size and can be used to train a CNN model uniformly without any dimensionality or scale obstacles. The MCMs resulting from this operation are then used in our context to train a classic CNN model for HSI interpretation. A majority voting strategy is adopted on the test stage to merge the label information coming from different scales.

The remainder of this paper is organized as follows. A review of related works is given in Section II. Section III presents the proposed approach in detail, introducing our MCM-based feature extraction method and the posterior 2-D CNN (2DCNN) data classification approach. Section IV describes our experimental results on three benchmark HSI data sets and conducts a comprehensive comparison with several popular CNN-based classification methods and classic spectral–spatial techniques for HSI classification. Some conclusions and hints at plausible future research lines are given in Section V.

## II. RELATED WORK

Over the past decades, many unsupervised and supervised techniques have been proposed for HSI classification, including traditional pixel-based classifiers such as clustering methods [18], logistic regression (LR) [19], [20], extreme learning machines [21], support vector machines (SVMs) [22], and sparse coding [23], [24]. Due to the complex characteristics of HSIs (high dimensionality, presence of noise/data redundancy,

and intraclass/interclass variability), supervised techniques usually achieve higher classification accuracy and thus have gained significant attention (e.g., SVM). On the other hand, the incorporation of spatial information is a very important strategy for improving HSI classification performance. Techniques for this purpose have addressed the intraclass variability and the interclass data correlations [9], [16]. For instance, Benediktsson *et al.* [14] introduced the extended morphological profiles (EMPs), which consist of the sequential application of opening and closing operations and thus provide an effective way to characterize the multiscale variability of spatial structures in the HSI. Inspired by the success of the EMP, a considerable number of related methods have been developed for spectral–spatial classification of HSIs [15], [25]–[27]. To further accelerate the speed of processing and reducing the storage space for HSI classification, a series of parallel and distributed approaches has been proposed in [28], [29].

Recently, deep learning methods have shown great potential in the field of remote sensing image classification [30], [31], as compared to traditional neural network-based classification methods such as the multilayer perceptron (MLP) [7]. The main idea behind deep learning methods is to extract abstract semantic features from the original data using a hierarchical representation architecture. Chen *et al.* [32] use stacked autoencoders (SAEs) to extract high-level features for HSI classification with a greedy layerwise pretraining strategy. In [33], deep-belief networks (DBNs) are also introduced for feature extraction and classification of HSIs. In [34], diversity-promoting priors are introduced into the pretraining (unsupervised) and fine-tuning (supervised) of the DBN to enhance classification performance. In [35], the recurrent neural network is first applied for HSI classification. However, most of these models [32], [33] need to flatten HSI patches in one dimension to satisfy their input requirements and may not effectively utilize the spatial information [2]. A good attempt to overcome this limitation is the work in [36], in which Ma *et al.* implemented a spatially updated deep autoencoder that jointly exploits spectral and spatial features, replacing each feature with the weighted average computed from the surrounding samples.

As a typical deep model, CNNs have also been extended for HSI classification purposes [2], [37]. Compared to SAEs, CNNs [38] allow using hyperspectral patches as the input, providing a natural way to incorporate the spatial-contextual information through their local receptive fields to enhance classification performance. Several CNN-based models have been proposed for HSI classification using spatial features. Mei *et al.* [39] present a CNN in which spectral signatures and spatial context are integrated by a preprocessing strategy that accounts for neighboring pixels, taking into account three types of feature vectors per pixel: the hyperspectral pixel, the mean of neighboring pixels, and the mean and standard deviation per spectral band of neighboring pixels. In [5], a deeper CNN model with shortcut connections is proposed to perform end-to-end classification. To further utilize the spatial information effectively, a diverse region-based CNN model is proposed in [3]. Zheng *et al.* [40] applied principal component analysis (PCA) [41] to extract the spectral features from HSIs,

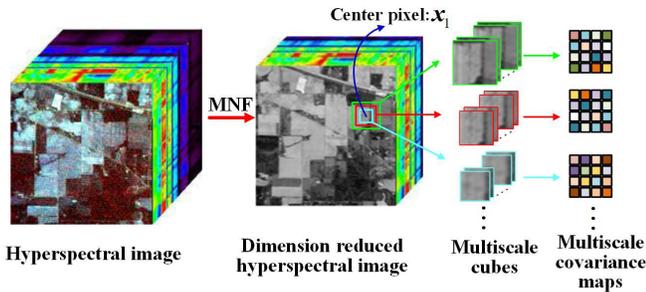


Fig. 2. Graphical illustration of the proposed approach for feature extraction considering a pixel  $x_1$  in an HSI.

and then, fed these features to a 2DCNN (i.e., spatial) to conduct an end-to-end classification.

A similar approach for feature extraction was followed by Liang and Li [42], in which neighborhood areas are first extracted from the resulting bands after applying a PCA transformation on the original HSI, and then, fed to a 2DCNN for feature extraction. Finally, sparse coding is applied to the resulting features for classification purposes. Another example is the work in [43], in which Zhao and Du combine the CNN-based spatial feature extraction with balanced local discriminant embedding-based spectral feature extraction, stacking the obtained features and then feeding them to an LR or SVM classifier. Yang *et al.* [44] and Zhang *et al.* [45] separately apply a 1-D CNN (1DCNN) to extract hierarchical spectral features and a 2DCNN to extract hierarchical space-related features. Then, they used a softmax regression classifier to obtain the final classification results. In addition, Romero *et al.* designed an unsupervised CNN model for unsupervised sparse features learning in [46]. To alleviate the problem of overfitting [8], a pixel-pairs-based data augmentation method has been recently proposed where new samples are constructed by pairing the labeled training pixels [1], with which the amount of input data used for training purposes increases quadratically. In [47], a convolutional-deconvolutional network is proposed for unsupervised feature learning and HSI classification. In [48], a band-adaptive spectral-spatial net is proposed for simultaneous dimensionality reduction and discriminative spectral-spatial feature extraction. All of these methods have been shown to achieve encouraging classification performance.

### III. PROPOSED METHOD

The proposed method consists of two main steps. First, we conduct dimensionality reduction on the original HSI using the maximum noise fraction (MNF) fraction. Then, we construct MCMs for feature extraction purposes. Fig. 2 provides a graphical illustration of the proposed approach, in which we can see how the MNF is first applied to the original HSI for dimensionality reduction. Then, a series of progressively larger subcubes is constructed by considering increasingly larger spatial neighborhoods around each pixel (in the figure, we consider a given pixel  $x_1$  as the center pixel), leading to the construction of MCMs (capturing multiple scales) for each pixel that are then used as features for classification purposes.

In the following, we describe in more detail that the two main steps of the proposed approach.

#### A. Dimensionality Reduction

First, in order to reduce the noise and computational complexity, the MNF method [49], [50] is applied to the original HSI. Specifically, given an HSI defined by  $\mathbf{Z} \in \mathbb{R}^{I \times J \times K}$ , a reduced image  $\mathbf{X} \in \mathbb{R}^{I \times J \times L}$  can be obtained by means of the MNF approach, where  $I$  and  $J$  denote the size of the two spatial dimensions,  $K$  is the size of the spectral dimension, and  $L$  represents the number of MNF principal components. The components produced by the MNF transformation are ordered in terms of signal-to-noise ratio (SNR), with the first components retaining the maximum information in terms of SNR. In this paper, we fix  $L$  to 20, since we have empirically found that the most useful information of the considered HSI data sets can be comprised by the first 20 principal components.

#### B. Feature Extraction via Multiscale Covariance Maps

For each pixel, a series of progressively larger cubes is constructed by considering increasingly larger spatial neighborhoods around the original pixel. For each of these cubes, a covariance matrix is further calculated to represent the information contained in the central pixel, which is enhanced by considering progressively larger neighborhoods. As a result, a series of covariance maps obtained from different spatial scales and called MCMs is used to represent the information comprised by the central pixel, which can be then used to train the classifier more effectively. Specifically, for the central pixel  $x_1$ , the covariance map of  $x_1$  on one scale  $T \times T$  is extracted as follows:

$$\mathbf{C} = \frac{1}{T^2 - 1} \sum_{i=1}^{T^2} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \in \mathbb{R}^{L \times L} \quad (1)$$

where  $\boldsymbol{\mu}$  denotes the mean of the set of feature vectors  $\{\mathbf{x}_i\}_{i=1, \dots, T}$  and the  $\{\mathbf{x}_i\}_{i=2, \dots, T^2}$  denote the corresponding neighboring pixels within a window of  $T \times T$  pixels. Moreover,  $M$  scales  $T_k$ ,  $k = 1 \dots M$  are taken into account. Thus,  $M$  covariance maps are extracted with (1) to represent the center pixel  $x_1$ , denoted by  $\mathbf{C}_k$ ,  $k = 1 \dots M$ , respectively. In this paper, the value of  $M$  is set to 15, which will be further discussed in the experimental results section.

The proposed MCMs method has the following distinctive advantages. First, with the use of covariance maps, the spatial and spectral information are simultaneously exploited. Each entry in the covariance map represents the covariance between two different spectral bands within a local spatial window, which can naturally integrate the spatial and spectral information. Second, with the proposed multiscale strategy, each sample can be enhanced using spatial information from different scales, thus increasing the information comprised by the samples effectively. Note that the covariance maps from different scales share the same size, which allows us to train a uniform CNN model without any dimensionality or scale obstacles. This is an important property of the proposed approach. To sum up, the proposed strategy based on

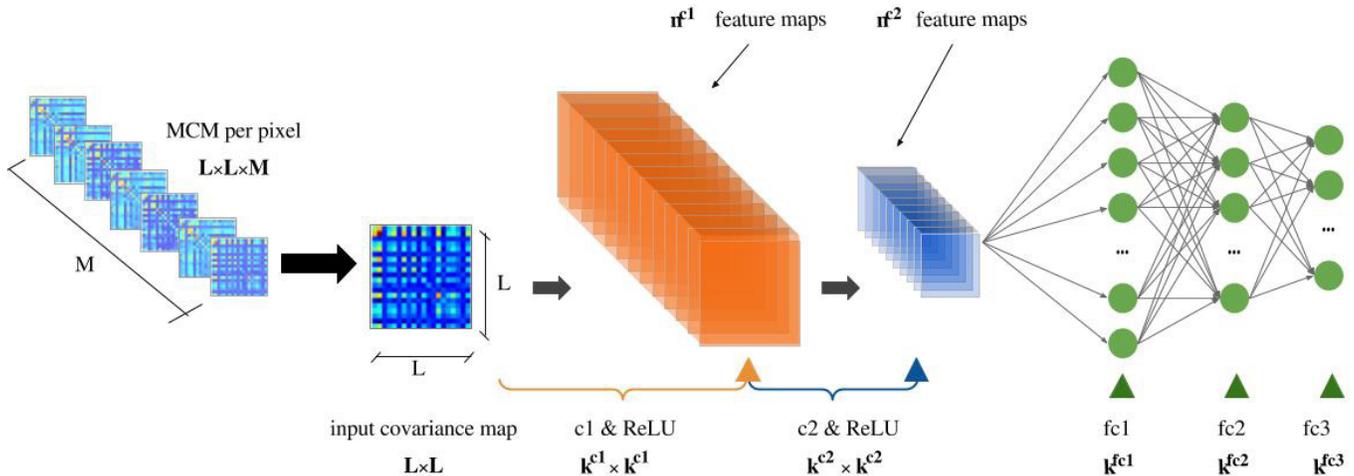


Fig. 3. Proposed 2DCNN architecture model. The input multiscale covariance block of each pixel is fed to the network, map by map, through two convolutional layers with ReLU activation function, followed by a max-pooling layer. The output of the 2DCNN is flattened and fed to three fully connected MLP layers that perform the final classification.

MCMs can naturally integrate the spatial–spectral information contained in the HSI.

#### IV. EXPERIMENTAL RESULTS

This section is organized as follows. In Section IV-A, we describe the neural network classifier that we have used in this paper to perform classification based on the extracted MCMs features. We note that the proposed feature extraction approach can be combined with other classifiers, as will be shown in experiments reported in this section, but we have specifically selected a 2DCNN classifier due to the fact that the proposed MCMs approach is specifically designed to increase the robustness of CNN models. In Section IV-B, we describe our experimental configuration. Section IV-C describes the considered hyperspectral data sets. Finally, Section IV-D describes a series of experiments designed in order to validate the performance of the proposed approach and to compare it with other state-of-the-art techniques for HSI classification.

##### A. 2-D Convolutional Neural Network Used for Classification

The architecture of the 2DCNN that we use for classification purposes is described in Fig. 3 and Table I. Specifically, the 2DCNN contains two convolutional layers and three fully connected layers. Each convolutional layer is followed by max-pooling layers. In addition, the rectified linear unit (ReLU) activation function is adopted. The minibatch gradient descent has been selected as the optimizer, whose goal is to minimize the cross-entropy reached by the 2DCNN. As each pixel  $x_i$  in the HSI  $\mathbf{X}$  has  $M$  associated matrices  $\mathbf{B}_{20 \times 20}$ , the 2DCNN model will perform  $M$  classifications for  $x_i$  on the test phase. The final label is obtained via majority voting from these  $M$  classification results. Note that the 2DCNN model is implemented using the TensorFlow open source library.<sup>1</sup>

<sup>1</sup><https://www.tensorflow.org>

TABLE I

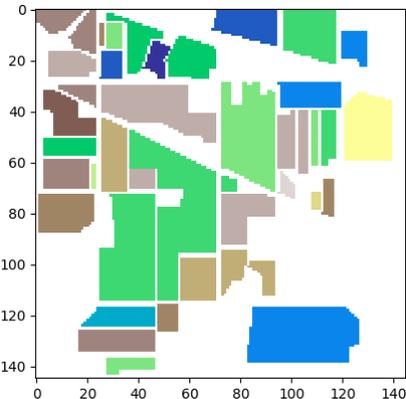
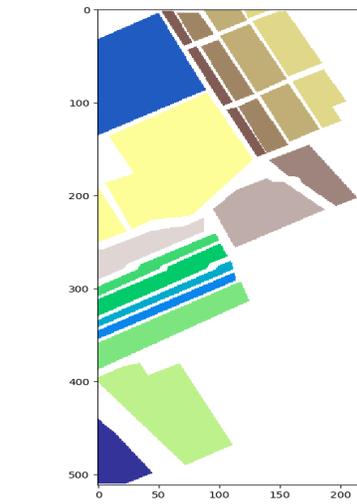
PARAMETERS USED WHEN DESIGNING THE TOPOLOGY OF THE PROPOSED 2DCNN ARCHITECTURE FOR ALL THE HSIS CONSIDERED IN EXPERIMENTS. THE KERNEL SIZES REPRESENT DESIGN CHOICES FOR THE PROPOSED DEEP ARCHITECTURE. NONE OF THE LAYERS USE PADDING

	Indian Pines	Pavia University	Salinas
Layer c1	$128 \times 3 \times 3$ , stride=1	$128 \times 5 \times 5$ , stride=1	$128 \times 3 \times 3$ , stride=1
$n^{c1} \times k^{c1} \times k^{c1}$			
Pooling	$2 \times 2$ , stride=2	$2 \times 2$ , stride=2	$2 \times 2$ , stride=2
Activation	ReLU	ReLU	ReLU
Layer c2	$64 \times 3 \times 3$ , stride=1	$64 \times 5 \times 5$ , stride=1	$64 \times 3 \times 3$ , stride=1
$n^{c2} \times k^{c2} \times k^{c2}$			
Pooling	$2 \times 2$ , stride=2	$2 \times 2$ , stride=2	$2 \times 2$ , stride=2
Activation	ReLU	ReLU	ReLU
Layer fc1	128	512	128
$k^{fc1}$			
Activation	ReLU	ReLU	ReLU
Layer fc2	128	512	128
$k^{fc2}$			
Activation	ReLU	ReLU	ReLU
Layer fc3	16	9	16
$k^{fc3}$			
Activation	Softmax	Softmax	Softmax

##### B. Experimental Configuration

Our experiments have been conducted on a hardware environment composed by a sixth Generation Intel Core i7-6700K processor with 8 M of Cache and up to 4.20 GHz (4 cores/8 way multitask processing), 40 GB of DDR4 RAM with a serial speed of 2400 MHz, a graphic processing unit (GPU) NVIDIA GeForce GTX 1080 with 8 GB GDDR5X of video memory and 10 Gb/s of memory frequency, a Toshiba DT01ACA HDD with 7200 rpm and 2 TB of capacity, and an ASUS Z170 programming motherboard. On the other hand, the software environment is composed of Ubuntu 16.04.4  $\times$  64 as operating system, TensorFlow 1.3.0 and compute device unified architecture 8 for GPU functionality. The so-called Xavier [51] is used for initialization of all weights, while the bias is initialized with zeros. The Adagrad optimizer [52] is adopted for training. The learning rate is set to 0.001 and remains unchanged during the whole procedure. The batch size is set to 100, and the weight decay is set to 0.0005 for  $L_2$  regularization. The above configuration has been used for all the experiments conducted with different HSIs. We do not

TABLE II  
NUMBER OF SAMPLES OF THE THREE TEST HSIS

INDIAN PINES			PAVIA UNIVERSITY			SALINAS		
								
Color	Land-cover type	Samples	Color	Land-cover type	Samples	Color	Land-cover type	Samples
	Background	10776		Background	164624		Broccoli-green-weeds-1	2009
	Alfalfa	46		Asphalt	6631		Broccoli-green-weeds-2	3726
	Corn-notill	1428		Meadows	18649		Fallow	1976
	Corn-min	830		Gravel	2099		Fallow-rough-plow	1394
	Corn	237		Trees	3064		Fallow-smooth	2678
	Grass/Pasture	483		Painted metal sheets	1345		Stubble	3959
	Grass/Trees	730		Bare Soil	5029		Celery	3579
	Grass/pasture-mowed	28		Bitumen	1330		Grapes-untrained	11271
	Hay-windrowed	478		Self-Blocking Bricks	3682		Soil-vinyard-develop	6203
	Oats	20		Shadows	947		Corn-senesced-green-weeds	3278
	Soybeans-notill	972					Lettuce-romaine-4wk	1068
	Soybeans-min	2455					Lettuce-romaine-5wk	1927
	Soybean-clean	593					Lettuce-romaine-6wk	916
	Wheat	205					Lettuce-romaine-7wk	1070
	Woods	1265					Vinyard-untrained	7268
	Bldg-Grass-Tree-Drives	386					Vinyard-vertical-trellis	1807
	Stone-steel towers	93						
	Total samples	21025		Total samples	207400		Total samples	111104

split any validation set from the training set or test set for hyperparameters selection.

### C. Hyperspectral Data Sets

Our experiments have been carried out using three different and well-known hyperspectral data sets, described as follows.

- 1) *AVIRIS Indian Pines*: This scene (see Table II) covers an agricultural site in Northwestern Indiana and was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor [53] in 1992. The data set is of size  $145 \times 145 \times 220$ , with spatial resolution of 20 m per pixel and spectral range from 0.2 to 2.4 microns. Before classification, 20 spectral bands (i.e., 104th–108th, 150th–163rd, and 220th) are discarded due to low SNR. This image contains 16 land-cover classes.
- 2) *ROSIS Pavia University*: This scene (see Table II) was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) [54] sensor over the campus of the University of Pavia, Italy. This data set contains

103 spectral bands after the noise-corrupted bands are discarded, while the scene contains  $610 \times 340$  pixels. The spatial resolution of this data set is 1.3 m, and the spectral coverage ranges from 0.43 to 0.86 microns. The ground-truth is composed by nine land-cover classes in total, covering an urban area with solid structures (asphalt, gravel, metal sheets, bitumen, bricks), natural objects (trees, meadows, soil), and shadows.

- 3) *AVIRIS Salinas*: This scene (see Table II) was also collected by the AVIRIS sensor over the Salinas Valley, CA. The data set is of size  $512 \times 217 \times 224$ , and it has spatial resolution of 3.7 m per pixel with 16 land-cover classes. Before classification, 20 bands were removed (i.e., 108th–112th, 154th–167th, and 224th).

### D. Experiments

- 1) *Experiment 1*: In our first experiment, we compare the proposed approach (hereinafter, MCMs+2DCNN) with a traditional approach for dimensionality reduction followed by

TABLE III

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY THE MNF+SVM, THE STANDARD 2DCNN, AND THE PROPOSED MCMs+2DCNN METHODS FOR THE AVIRIS INDIAN PINES SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE

Class name	1% training			2% training			3% training		
	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN
Alfalfa	43.11(27.54)	12.61(4.64)	<b>71.11</b> (3.72)	53.56(11.11)	13.04(2.38)	<b>94.67</b> (1.78)	73.41(12.68)	28.26(9.53)	<b>80.00</b> (3.64)
Corn-notill	71.68(8.32)	24.09(6.81)	<b>76.23</b> (1.86)	72.81(4.96)	36.67(10.49)	<b>88.35</b> (0.94)	78.84(4.11)	55.90(3.03)	<b>95.61</b> (0.53)
Corn-min	48.31(11.24)	31.40(10.08)	<b>77.08</b> (1.39)	56.08(4.23)	39.93(13.65)	<b>87.80</b> (0.96)	62.30(7.42)	54.48(2.81)	<b>74.31</b> (3.03)
Corn	22.44(10.72)	11.73(3.88)	<b>94.36</b> (1.65)	40.73(10.52)	18.14(4.14)	<b>95.95</b> (1.14)	50.48(7.66)	35.19(4.41)	<b>94.59</b> (2.80)
Grass/Pasture	79.37(10.75)	14.78(3.74)	<b>65.82</b> (1.63)	88.52(9.08)	40.58(6.91)	<b>92.73</b> (0.76)	<b>90.43</b> (5.53)	54.37(2.15)	85.13(0.98)
Grass/Trees	<b>94.03</b> (4.23)	82.82(6.43)	88.39(2.21)	<b>95.51</b> (2.41)	67.04(2.96)	90.71(1.59)	<b>97.32</b> (1.93)	76.88(4.37)	93.53(1.09)
Grass/pasture-mowed	56.67(36.96)	22.14(7.63)	<b>100.00</b> (0.00)	60.00(30.83)	30.71(13.67)	<b>100.00</b> (0.00)	74.07(8.55)	35.00(6.93)	<b>100.00</b> (0.00)
Hay-windrowed	91.56(8.81)	<b>91.67</b> (6.34)	80.93(5.08)	96.62(4.80)	93.89(4.72)	<b>99.06</b> (0.37)	97.37(3.60)	96.74(1.78)	<b>100.00</b> (0.00)
Oats	42.63(27.34)	22.00(6.78)	<b>90.53</b> (2.11)	55.79(28.20)	28.00(6.78)	<b>100.00</b> (0.00)	53.16(24.49)	27.00(5.10)	<b>84.21</b> (9.42)
Soybeans-notill	47.61(11.95)	34.36(10.47)	<b>68.21</b> (1.04)	58.39(12.61)	34.88(12.95)	<b>81.58</b> (0.92)	67.37(12.12)	47.51(2.79)	<b>94.59</b> (0.86)
Soybeans-min	76.06(7.03)	72.59(14.86)	<b>86.55</b> (1.43)	80.33(4.83)	72.64(7.13)	<b>94.37</b> (0.73)	79.95(3.28)	77.48(3.68)	<b>95.18</b> (1.96)
Soybean-clean	<b>46.25</b> (8.37)	20.91(9.78)	42.25(1.26)	66.13(6.88)	38.99(9.20)	<b>67.99</b> (0.49)	75.88(3.73)	53.46(1.85)	<b>87.17</b> (3.51)
Wheat	<b>97.48</b> (1.75)	29.17(10.57)	86.44(1.84)	<b>97.70</b> (3.73)	64.49(13.58)	97.50(2.07)	<b>99.04</b> (1.10)	85.76(4.95)	90.61(2.18)
Woods	93.79(3.32)	80.74(2.66)	<b>90.37</b> (1.85)	93.78(2.41)	89.49(5.71)	<b>97.51</b> (0.67)	94.64(3.23)	91.83(0.32)	<b>97.23</b> (0.30)
Bldg-Grass-Tree-Drives	40.81(14.11)	27.25(9.74)	<b>65.29</b> (4.43)	52.33(9.32)	42.90(10.75)	<b>91.01</b> (1.93)	62.09(3.95)	58.19(5.55)	<b>93.53</b> (1.63)
Stone-steel towers	57.07(29.40)	8.17(2.85)	<b>71.30</b> (4.22)	72.42(24.39)	19.35(6.20)	<b>74.51</b> (2.54)	81.56(8.72)	16.34(1.43)	<b>93.56</b> (1.30)
Overall Accuracy	70.55(2.00)	50.71(2.27)	<b>78.62</b> (0.59)	76.38(0.97)	57.12(2.98)	<b>90.35</b> (0.23)	80.24(1.12)	67.38(1.10)	<b>92.66</b> (0.50)
Average Accuracy	63.05(5.42)	36.65(3.07)	<b>78.43</b> (0.78)	71.29(3.61)	45.67(1.90)	<b>90.86</b> (0.24)	77.37(2.65)	55.90(1.67)	<b>91.20</b> (0.73)
Kappa	65.96(2.36)	42.85(2.63)	<b>75.60</b> (0.65)	72.83(1.13)	50.58(3.16)	<b>89.01</b> (0.26)	77.35(1.65)	62.56(1.60)	<b>91.64</b> (0.56)
Time(seconds)	-	<b>3.27</b> (0.47)	3.43(0.19)	-	<b>6.26</b> (0.53)	6.27(0.28)	-	<b>7.99</b> (0.47)	8.08(0.31)

Class name	5% training			10% training			15% training		
	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN
Alfalfa	77.91(9.82)	38.70(7.95)	<b>99.53</b> (0.93)	73.66(10.21)	58.26(8.41)	<b>100.00</b> (0.00)	77.44(5.90)	66.09(3.53)	<b>96.92</b> (1.92)
Corn-notill	82.07(1.96)	52.31(10.84)	<b>95.38</b> (0.52)	85.49(1.27)	72.17(8.47)	<b>97.96</b> (0.53)	88.90(1.63)	81.79(3.03)	<b>99.19</b> (0.45)
Corn-min	69.39(5.16)	52.55(9.14)	<b>95.86</b> (0.58)	71.86(3.68)	75.52(5.64)	<b>98.45</b> (0.64)	79.28(2.99)	79.93(18.02)	<b>99.06</b> (0.40)
Corn	57.20(9.15)	35.44(8.58)	<b>96.27</b> (1.34)	66.85(6.40)	55.19(8.27)	<b>97.56</b> (1.82)	74.18(4.69)	86.16(2.94)	<b>98.81</b> (1.90)
Grass/Pasture	<b>93.38</b> (1.27)	56.27(7.99)	87.95(1.78)	94.93(1.42)	62.07(13.81)	<b>98.71</b> (0.78)	95.44(1.32)	89.15(9.77)	<b>99.56</b> (0.45)
Grass/Trees	<b>98.33</b> (0.79)	79.70(6.49)	94.43(0.37)	<b>98.83</b> (0.49)	90.60(4.45)	97.41(0.43)	98.85(0.43)	97.64(0.24)	<b>99.90</b> (0.08)
Grass/pasture-mowed	81.92(18.04)	44.29(5.80)	<b>100.00</b> (0.00)	87.20(14.34)	61.43(9.69)	<b>100.00</b> (0.00)	90.00(8.21)	83.57(15.42)	<b>100.00</b> (0.00)
Hay-windrowed	99.19(0.59)	96.90(1.34)	<b>99.96</b> (0.09)	98.51(1.60)	98.12(0.96)	<b>100.00</b> (0.00)	99.29(0.43)	99.62(0.41)	<b>100.00</b> (0.00)
Oats	52.11(24.87)	25.00(13.78)	<b>76.84</b> (7.88)	72.22(19.42)	52.00(9.27)	<b>83.33</b> (7.86)	84.12(12.42)	72.00(12.88)	<b>97.65</b> (2.88)
Soybeans-notill	74.12(6.25)	43.85(19.05)	<b>89.32</b> (0.49)	79.74(5.23)	65.04(14.86)	<b>83.16</b> (3.10)	99.29(0.43)	71.81(13.89)	<b>99.61</b> (0.37)
Soybeans-min	83.68(3.49)	67.25(21.24)	<b>97.74</b> (0.42)	87.13(1.62)	84.31(5.69)	<b>99.36</b> (0.07)	89.04(1.03)	93.43(2.44)	<b>97.97</b> (1.21)
Soybean-clean	82.18(4.68)	57.03(8.68)	<b>94.99</b> (0.58)	88.74(3.02)	69.98(7.23)	<b>98.16</b> (0.22)	90.93(1.43)	77.23(11.43)	<b>98.45</b> (0.69)
Wheat	<b>98.97</b> (1.11)	72.00(4.37)	95.98(0.60)	<b>99.35</b> (0.50)	93.07(3.17)	91.20(3.80)	<b>99.48</b> (0.33)	97.95(1.76)	98.97(0.76)
Woods	93.90(2.89)	82.86(14.42)	<b>100.00</b> (0.00)	96.61(1.34)	89.52(8.81)	<b>99.58</b> (0.21)	96.90(0.43)	87.78(12.61)	<b>99.81</b> (0.13)
Bldg-Grass-Tree-Drives	66.89(4.21)	67.36(13.00)	<b>93.66</b> (0.98)	73.83(5.05)	86.48(7.13)	<b>98.39</b> (0.99)	76.65(3.01)	86.01(4.37)	<b>99.94</b> (0.12)
Stone-steel towers	82.05(7.28)	15.48(3.88)	<b>89.32</b> (1.16)	83.73(7.19)	46.67(3.16)	<b>91.08</b> (5.53)	82.28(4.22)	53.76(11.58)	<b>98.73</b> (1.60)
Overall Accuracy	83.74(1.06)	63.47(5.48)	<b>95.66</b> (0.06)	87.11(0.62)	78.94(3.35)	<b>98.61</b> (0.30)	89.51(0.35)	86.51(2.71)	<b>99.07</b> (0.25)
Average Accuracy	80.83(3.06)	55.44(2.77)	<b>94.20</b> (0.50)	84.92(1.50)	72.53(2.47)	<b>96.94</b> (0.78)	87.87(0.96)	82.75(3.52)	<b>99.04</b> (0.27)
Kappa	81.40(1.20)	58.51(5.58)	<b>95.05</b> (0.07)	85.26(0.73)	75.97(3.81)	<b>98.42</b> (0.34)	88.01(0.41)	84.56(3.24)	<b>98.94</b> (0.29)
Time(seconds)	-	<b>13.80</b> (0.37)	19.38(0.63)	-	<b>25.73</b> (0.26)	36.89(0.69)	-	<b>38.03</b> (0.59)	55.46(0.68)

classification (MNF+SVM) that is widely used for hyperspectral data interpretation. It should be noted that, in our method, the MNF is also used as preprocessing. As a result, we adopt it for dimensionality reduction prior to the application of the standard SVM classifier for a fair comparison. In all cases, the first 20 MNF components are selected for classification purposes.

Table III shows the classification results obtained by the aforementioned methods using 1%, 2%, 3%, 5%, 10%, and 15% randomly selected samples for each land-cover type in the AVIRIS Indian Pines data set. Each experiment has been repeated 10 times, and the average results (together with the standard deviation) are reported. As it can be seen from Table III, the proposed approach (MCMs+2DCNN) exhibits obvious improvements according to different classification metrics, such as the overall accuracy (OA), average accuracy (AA), and Kappa statistic, when compared to the MNF+SVM approach. For example, when only 2% training samples are selected per class, the OA, AA, and Kappa of the MNF+SVM approach are 76.38%, 71.29%, and 72.83%, respectively, while the results obtained by our MCMs+2DCNN approach are 90.35%, 90.86%, and 89.01%, respectively. This means that the average improvement observed in the three quantitative metrics is over 10%.

Similar results can also be observed for the other two tested data sets, which are, respectively, shown in Tables IV (ROSIS Pavia University) and V (AVIRIS Salinas). The main reason is that the MNF+SVM method only considers the spectral information, while the proposed approach integrates both the spatial and the spectral information contained in the scene. Note that the running time of MNF+SVM is omitted because it is implemented in the central processing unit while our method is implemented on a GPU.

2) *Experiment 2*: In this experiment, we provide a comprehensive comparison between the 2DCNN trained with and without MCMs. Specifically, the 2DCNN models trained without MCMs take patches from the original hyperspectral scenes as the input data, i.e., the networks are fed with a neighborhood window centered around each pixel in the scene. In this way, the input layer accepts volumes of size  $d \times d \times n$ , where  $d$  is the width and height of the input volume and  $n$  is the number of components extracted from MNF. This requires a preprocessing stage in order to: 1) reduce the spectral dimensionality with the MNF method and 2) divide the reduced HSI into patches of size  $d \times d \times n$ , being  $d$  an odd number because the desired label to be reached by the network will be the one that owns the central position of the pixel in the patch of size  $[d/2+1, d/2+1, n]$ . In our case,  $d$  has been fixed

TABLE IV

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY THE MNF+SVM, THE STANDARD 2DCNN, AND THE PROPOSED MCMS+2DCNN FOR THE ROSIS PAVIA UNIVERSITY SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE

Class name	1% training			2% training			3% training		
	MNF+SVM	2DCNN	MCMS+2DCNN	MNF+SVM	2DCNN	MCMS+2DCNN	MNF+SVM	2DCNN	MCMS+2DCNN
Asphalt	86.23(4.88)	83.13(2.32)	<b>94.88</b> (0.85)	88.44(1.51)	86.51(1.39)	<b>99.02</b> (0.16)	90.54(1.69)	86.54(3.21)	<b>98.85</b> (0.69)
Meadows	94.12(1.35)	91.61(1.56)	<b>98.93</b> (0.35)	95.89(0.69)	94.49(0.27)	<b>99.95</b> (0.03)	96.66(1.08)	91.73(3.08)	<b>99.99</b> (0.01)
Gravel	57.25(6.49)	55.17(1.38)	<b>92.10</b> (7.83)	61.96(3.66)	67.45(8.26)	<b>94.92</b> (3.00)	64.72(4.21)	67.95(7.86)	<b>97.04</b> (0.39)
Trees	85.08(3.01)	<b>85.75</b> (3.98)	77.16(4.17)	<b>87.06</b> (2.24)	86.42(2.09)	82.09(1.26)	86.97(2.70)	87.98(4.74)	<b>89.04</b> (2.21)
Painted metal sheets	<b>99.68</b> (0.20)	93.73(3.07)	98.33(0.85)	<b>99.79</b> (0.22)	90.47(2.60)	94.32(4.22)	<b>99.84</b> (0.13)	95.64(1.70)	99.75(0.09)
Bare Soil	71.19(3.87)	36.92(1.96)	<b>99.73</b> (0.44)	76.67(2.57)	43.63(0.90)	<b>98.77</b> (0.45)	77.03(3.05)	54.22(4.43)	<b>99.96</b> (0.05)
Bitumen	36.64(19.83)	67.93(5.97)	<b>99.35</b> (1.02)	53.58(8.22)	69.16(2.68)	<b>99.71</b> (0.55)	52.94(10.49)	50.98(19.70)	<b>99.44</b> (0.25)
Self-Blocking Bricks	64.04(6.07)	79.70(5.59)	<b>98.15</b> (0.82)	77.61(4.23)	82.13(2.15)	<b>95.90</b> (0.81)	80.36(2.86)	83.20(2.50)	<b>93.71</b> (0.91)
Shadows	<b>97.58</b> (1.63)	74.55(6.14)	42.13(7.24)	<b>99.16</b> (1.26)	78.46(3.48)	64.35(5.93)	<b>99.62</b> (0.24)	75.29(5.21)	76.64(2.09)
Overall Accuracy	83.62(0.96)	79.59(1.55)	<b>95.17</b> (0.50)	87.48(0.55)	83.04(0.54)	<b>96.82</b> (0.29)	88.54(0.58)	82.84(1.24)	<b>97.67</b> (0.20)
Average Accuracy	76.87(1.57)	74.28(1.72)	<b>88.97</b> (0.96)	82.24(1.05)	77.64(1.08)	<b>92.12</b> (0.72)	83.19(1.50)	77.06(1.86)	<b>94.50</b> (0.43)
Kappa	78.09(1.20)	72.72(1.95)	<b>93.60</b> (0.66)	83.25(0.73)	77.10(0.72)	<b>95.78</b> (0.39)	84.65(0.78)	77.06(1.45)	<b>96.91</b> (0.27)
Time (seconds)	-	<b>7.37</b> (0.30)	10.23(0.65)	-	<b>14.59</b> (0.25)	20.10(0.66)	-	<b>21.48</b> (0.30)	28.34(0.65)

Class name	5% training			10% training			15% training		
	MNF+SVM	2DCNN	MCMS+2DCNN	MNF+SVM	2DCNN	MCMS+2DCNN	MNF+SVM	2DCNN	MCMS+2DCNN
Asphalt	92.50(1.15)	93.65(0.70)	<b>99.73</b> (0.23)	93.22(0.77)	96.73(1.62)	<b>100.00</b> (0.01)	93.94(0.46)	98.20(1.67)	<b>100.00</b> (0.00)
Meadows	96.98(0.34)	95.14(3.44)	<b>100.00</b> (0.01)	97.26(0.37)	97.51(1.56)	<b>99.98</b> (0.01)	97.44(0.29)	99.17(0.62)	<b>99.98</b> (0.02)
Gravel	68.89(4.03)	80.52(2.42)	<b>98.49</b> (0.31)	71.52(3.27)	82.51(12.79)	<b>99.84</b> (0.17)	74.87(1.80)	95.50(1.07)	<b>99.93</b> (0.09)
Trees	91.09(1.56)	<b>93.77</b> (1.08)	91.65(3.46)	92.26(0.57)	<b>97.71</b> (1.42)	97.14(0.39)	92.98(0.93)	<b>98.45</b> (0.52)	97.12(0.21)
Painted metal sheets	<b>99.75</b> (0.09)	98.66(0.57)	99.56(0.20)	99.89(0.10)	99.66(0.40)	<b>99.95</b> (0.07)	<b>99.91</b> (0.08)	99.76(0.25)	99.88(0.24)
Bare Soil	81.29(0.94)	72.95(2.52)	<b>100.00</b> (0.00)	85.58(1.48)	81.48(14.28)	<b>100.00</b> (0.00)	88.45(0.80)	93.30(5.31)	<b>100.00</b> (0.00)
Bitumen	59.87(2.58)	74.35(1.90)	<b>99.95</b> (0.10)	71.99(2.60)	83.61(9.09)	<b>99.98</b> (0.03)	74.46(2.95)	92.93(3.03)	<b>100.00</b> (0.00)
Self-Blocking Bricks	81.69(2.40)	93.21(1.66)	<b>98.47</b> (0.46)	84.71(1.73)	97.93(0.86)	<b>99.59</b> (0.26)	85.68(1.47)	97.93(0.87)	<b>99.42</b> (0.07)
Shadows	<b>99.73</b> (0.19)	90.29(2.32)	83.74(1.01)	<b>99.82</b> (0.16)	93.81(1.55)	85.49(1.01)	<b>99.88</b> (0.16)	97.42(0.43)	95.85(0.95)
Overall Accuracy	90.32(0.35)	90.67(1.42)	<b>98.78</b> (0.27)	91.91(0.18)	94.37(2.15)	<b>99.42</b> (0.06)	92.82(0.17)	97.78(0.77)	<b>99.64</b> (0.03)
Average Accuracy	85.75(0.65)	88.06(0.77)	<b>96.84</b> (0.48)	88.47(0.50)	92.33(4.06)	<b>98.00</b> (0.15)	89.73(0.37)	96.96(0.98)	<b>99.13</b> (0.13)
Kappa	87.06(0.47)	87.57(1.81)	<b>98.38</b> (0.36)	89.22(0.25)	92.48(2.99)	<b>99.23</b> (0.07)	90.44(0.22)	97.05(1.03)	<b>99.52</b> (0.04)
Time (seconds)	-	<b>36.02</b> (0.48)	46.85(0.70)	-	<b>70.87</b> (0.66)	93.50(0.48)	-	<b>104.31</b> (0.81)	138.66(0.57)

to 21 (a number that is close to the MCMS+2DCNN input volume with  $L = 20$ ) and  $n$  has been set to 1, so the 2DCNN input data uses patches of size  $21 \times 21$ . Also, following the strategy in [55], a border mirroring strategy has been implemented in order to take into account the pixels at the border of the HSI.

Tables III–V (already discussed in Experiment 1) show the corresponding results comparison between the 2DCNN trained with and without MCMS using various numbers of samples selected per class, ranging from 1% to 15%. From Table III, we can observe that the proposed method achieves significantly better OAs than the classical 2DCNN on the Indian Pines data set. This is mainly due to the fact that, with MCMS, the 2DCNN model can be trained sufficiently, integrating spatial–spectral information from different scales, thus leading to satisfactory classification results. Also, it should be noted that the proposed MCMS+2DCNN exhibits similar execution time than the classical 2DCNN. At first sight, we suppose that classical 2DCNN should be faster than MCMS+2DCNN, because the classical 2DCNN is trained with  $n$  pixels (this number is calculated according to the considered training percentage), while the MCMS+2DCNN is trained with  $M$  covariance matrices per pixel, so in the end, the MCMS+2DCNN is trained with  $n \times M$  samples, which should make it slower. However, the runtime is eventually very similar because the MCMS+2DCNN needs less epochs than 2DCNN to converge, although its epochs are slower than those of the 2DCNN due to the amount of data involved, as we can observe in Fig. 4, where the test accuracy evolution is displayed across the epochs. Fig. 5 illustrates the aforementioned results in a graphical way. Specifically, the figure displays the classification maps obtained by the 2DCNN trained without MCMS (center) and the classification maps obtained by the 2DCNN trained with

MCMS (right). In both cases, 10% of the available samples have been randomly selected from each class to train both networks. As it can be seen, the map obtained by 2DCNN trained with MCMS exhibits a very clean result, with less misclassifications at the borders of the classes than the one obtained by the 2DCNN trained without MCMS.

A similar situation can be observed in Tables IV and V for the ROSIS Pavia University and the AVIRIS Salinas scene, respectively. As it can be seen, although the proposed MCMS+2DCNN method is slightly slower than the classical 2DCNN, it reaches better OA values. Figs. 6 and 7 present the corresponding classification maps obtained by the 2DCNN model, trained with and without MCMS and using 10% of the available training samples per class on these two data sets. As it can also be observed, the 2DCNN trained with the MCMS obtains a classification map that is more similar to the ground truth on both data sets.

In addition, Fig. 4 shows the convergence of the test classification performance for the 2DCNN and the MCMS+2DCNN on the three considered data sets (i.e., Indian Pines, Pavia University and Salinas). As it can be observed from Fig. 4, the proposed MCMS+2DCNN method converges to satisfactory classification performance after the first few epochs. In contrast, without MCMS, the 2DCNN needs more epochs to reach acceptable results and still has the risk of divergence, even after 50 epochs. Note that, in this experiment, 10% of the available samples per class are randomly selected to train the CNN model.

The above-mentioned experiments demonstrate that the proposed method can indeed alleviate the problem of overfitting and thus improve the robustnesses of the CNN model for HSI classification purposes. This also reveals that the main benefits of the proposed method are twofold. First, with the use of

TABLE V

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY THE MNF+SVM, THE STANDARD 2DCNN, AND THE PROPOSED MCMs+2DCNN FOR THE AVIRIS SALINAS SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE

Class name	1% training			2% training			3% training		
	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN
Broccoli-green-weeds-1	98.73(0.70)	63.44(13.14)	<b>99.70(0.30)</b>	<b>99.69(0.25)</b>	69.61(28.96)	99.01(0.64)	99.58(0.31)	78.47(20.76)	<b>100.00(0.00)</b>
Broccoli-green-weeds-2	99.71(0.21)	55.07(25.64)	<b>100.00(0.00)</b>	99.75(0.25)	53.61(37.37)	<b>99.72(0.37)</b>	99.81(0.13)	89.36(7.79)	<b>100.00(0.00)</b>
Fallow	99.54(0.34)	66.02(26.39)	<b>100.00(0.00)</b>	99.46(0.55)	77.37(35.23)	<b>100.00(0.00)</b>	99.63(0.10)	86.35(8.55)	<b>100.00(0.00)</b>
Fallow-rough-plow	<b>96.11(3.43)</b>	64.81(10.30)	98.99(0.82)	<b>98.86(0.55)</b>	76.03(18.30)	92.93(1.41)	<b>99.22(0.26)</b>	95.58(1.96)	96.52(0.90)
Fallow-smooth	<b>98.37(0.58)</b>	76.09(17.74)	95.97(0.58)	<b>99.14(0.39)</b>	76.73(13.89)	97.52(1.47)	<b>98.97(0.35)</b>	91.57(11.23)	99.88(0.25)
Stubble	<b>99.75(0.12)</b>	96.42(0.92)	97.16(1.65)	<b>99.86(0.13)</b>	87.41(21.13)	99.29(0.58)	<b>99.81(0.09)</b>	99.49(0.92)	98.95(0.36)
Celery	99.67(0.20)	85.60(9.06)	<b>100.00(0.00)</b>	99.72(0.27)	86.35(7.54)	<b>99.95(0.04)</b>	99.66(0.22)	83.00(20.81)	<b>99.98(0.02)</b>
Grapes-untrained	87.31(3.22)	57.71(26.49)	<b>99.41(0.34)</b>	89.09(1.33)	69.27(30.68)	<b>99.13(0.84)</b>	89.98(1.64)	75.45(11.96)	<b>99.91(0.05)</b>
Soil-vinyard-develop	99.71(0.21)	98.46(2.07)	<b>99.96(0.08)</b>	99.78(0.17)	98.61(1.14)	<b>100.00(0.00)</b>	99.81(0.14)	98.76(0.76)	<b>100.00(0.00)</b>
Corn-senesced-green-weeds	97.24(1.48)	60.36(11.13)	<b>99.61(0.35)</b>	97.65(1.18)	75.71(8.62)	<b>98.59(0.24)</b>	97.98(0.79)	74.07(8.91)	<b>99.99(0.03)</b>
Lettuce-romaine-4wk	94.10(2.55)	90.54(2.73)	<b>98.32(0.45)</b>	97.29(2.82)	90.00(11.68)	<b>99.98(0.04)</b>	<b>98.89(0.98)</b>	92.29(7.14)	98.16(0.27)
Lettuce-romaine-5wk	<b>99.75(0.30)</b>	73.36(24.60)	97.88(1.18)	<b>99.88(0.15)</b>	95.14(4.38)	99.19(0.18)	<b>99.94(0.08)</b>	97.96(1.26)	99.73(0.38)
Lettuce-romaine-6wk	99.14(0.48)	85.54(7.12)	<b>98.17(1.45)</b>	98.52(0.69)	80.04(6.44)	<b>99.26(0.32)</b>	99.01(0.60)	96.19(1.49)	<b>99.91(0.18)</b>
Lettuce-romaine-7wk	91.68(4.44)	67.48(9.01)	<b>93.16(1.09)</b>	95.82(1.20)	85.86(8.64)	<b>99.81(0.30)</b>	96.55(1.03)	92.05(1.39)	<b>99.77(0.10)</b>
Vinyard-untrained	68.40(3.33)	48.11(28.48)	<b>98.07(1.13)</b>	75.79(3.90)	51.66(34.64)	<b>99.41(0.12)</b>	77.24(2.39)	72.57(16.66)	<b>99.71(0.17)</b>
Vinyard-vertical-trellis	98.39(0.65)	44.92(17.88)	<b>100.00(0.00)</b>	98.61(0.40)	51.82(21.55)	<b>99.98(0.03)</b>	98.92(0.69)	51.87(13.59)	<b>100.00(0.00)</b>
Overall Accuracy	92.25(0.42)	68.79(3.45)	<b>98.87(0.30)</b>	93.95(0.47)	74.14(5.97)	<b>99.19(0.17)</b>	94.42(0.27)	83.72(2.57)	<b>99.72(0.05)</b>
Average Accuracy	95.48(0.43)	70.87(3.54)	<b>98.52(0.25)</b>	96.81(0.30)	76.58(7.02)	<b>98.99(0.11)</b>	97.19(0.17)	85.94(2.97)	<b>99.53(0.08)</b>
Kappa	91.36(0.46)	65.32(3.63)	<b>98.74(0.34)</b>	93.26(0.52)	71.19(6.59)	<b>99.10(0.19)</b>	93.78(0.31)	81.87(2.86)	<b>99.68(0.06)</b>
Time (seconds)	-	<b>9.51(0.67)</b>	12.38(0.68)	-	<b>18.38(0.71)</b>	25.08(0.64)	-	<b>26.87(0.62)</b>	36.68(0.69)

Class name	5% training			10% training			15% training		
	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN	MNF+SVM	2DCNN	MCMs+2DCNN
Broccoli-green-weeds-1	99.87(0.13)	92.55(1.93)	<b>100.00(0.00)</b>	99.89(0.10)	86.66(14.30)	<b>100.00(0.00)</b>	99.92(0.14)	99.24(0.41)	<b>100.00(0.00)</b>
Broccoli-green-weeds-2	99.91(0.05)	94.03(3.05)	<b>99.99(0.01)</b>	99.81(0.23)	79.01(23.47)	<b>100.00(0.00)</b>	99.90(0.12)	98.49(1.53)	<b>100.00(0.00)</b>
Fallow	99.76(0.10)	88.17(2.41)	<b>100.00(0.00)</b>	99.73(0.11)	99.39(0.69)	<b>100.00(0.00)</b>	99.87(0.14)	98.84(0.99)	<b>100.00(0.00)</b>
Fallow-rough-plow	99.14(0.47)	91.13(6.80)	<b>99.95(0.09)</b>	99.43(0.27)	98.74(0.46)	<b>99.71(0.28)</b>	99.36(0.28)	99.83(0.14)	<b>99.54(0.17)</b>
Fallow-smooth	99.02(0.41)	97.96(1.44)	<b>99.72(0.13)</b>	99.26(0.31)	98.05(1.35)	<b>99.89(0.07)</b>	99.31(0.31)	99.45(0.20)	<b>99.99(0.02)</b>
Stubble	<b>99.87(0.10)</b>	97.68(4.43)	99.60(0.80)	<b>99.88(0.11)</b>	98.46(3.09)	<b>99.88(0.12)</b>	99.93(0.07)	<b>99.99(0.01)</b>	99.67(0.42)
Celery	99.85(0.13)	96.17(1.83)	<b>99.96(0.03)</b>	99.75(0.22)	98.33(0.56)	<b>99.86(0.15)</b>	99.87(0.14)	98.92(0.68)	<b>100.00(0.00)</b>
Grapes-untrained	91.01(1.10)	76.01(28.08)	<b>100.00(0.00)</b>	92.67(0.64)	62.70(25.20)	<b>100.00(0.00)</b>	92.82(0.54)	78.94(17.83)	<b>100.00(0.00)</b>
Soil-vinyard-develop	99.70(0.46)	85.31(22.55)	<b>100.00(0.00)</b>	99.91(0.08)	76.90(25.18)	<b>100.00(0.00)</b>	99.87(0.09)	80.76(23.18)	<b>100.00(0.00)</b>
Corn-senesced-green-weeds	98.25(0.46)	82.56(10.51)	<b>99.79(0.15)</b>	98.93(0.26)	90.03(4.77)	<b>100.00(0.00)</b>	98.86(0.30)	92.01(8.56)	<b>100.00(0.00)</b>
Lettuce-romaine-4wk	<b>99.48(0.95)</b>	96.80(1.49)	99.11(0.41)	<b>99.77(0.30)</b>	98.40(0.37)	99.63(0.39)	99.83(0.29)	99.98(0.04)	<b>99.91(0.11)</b>
Lettuce-romaine-5wk	99.92(0.17)	95.89(3.61)	<b>99.98(0.03)</b>	<b>99.99(0.02)</b>	97.28(2.65)	99.92(0.06)	99.98(0.04)	<b>100.00(0.00)</b>	99.99(0.02)
Lettuce-romaine-6wk	99.36(0.53)	95.06(1.56)	<b>99.72(0.24)</b>	99.24(0.47)	97.40(1.22)	<b>100.00(0.00)</b>	99.56(0.24)	99.18(0.30)	<b>100.00(0.00)</b>
Lettuce-romaine-7wk	97.47(0.71)	91.87(2.45)	<b>99.35(0.40)</b>	98.09(0.65)	98.01(0.80)	<b>100.00(0.00)</b>	98.66(0.62)	98.81(0.27)	<b>99.96(0.05)</b>
Vinyard-untrained	79.90(1.58)	65.98(25.54)	<b>99.99(0.02)</b>	83.76(1.03)	92.48(11.49)	<b>100.00(0.00)</b>	83.76(1.03)	98.08(2.22)	<b>100.00(0.00)</b>
Vinyard-vertical-trellis	99.22(0.45)	67.81(15.46)	<b>100.00(0.00)</b>	99.36(0.31)	92.07(8.85)	<b>100.00(0.00)</b>	99.51(0.28)	94.62(1.68)	<b>100.00(0.00)</b>
Overall Accuracy	95.08(0.23)	84.30(3.69)	<b>99.90(0.06)</b>	96.04(0.14)	85.18(5.64)	<b>99.96(0.01)</b>	96.30(0.12)	92.17(5.24)	<b>99.96(0.00)</b>
Average Accuracy	97.61(0.12)	88.44(1.48)	<b>99.82(0.04)</b>	98.09(0.09)	91.49(2.36)	<b>99.93(0.02)</b>	98.28(0.07)	96.07(1.90)	<b>99.94(0.00)</b>
Kappa	94.51(0.26)	82.55(3.95)	<b>99.89(0.06)</b>	95.59(0.15)	83.66(6.12)	<b>99.95(0.02)</b>	95.88(0.14)	91.35(5.75)	<b>99.96(0.00)</b>
Time(seconds)	-	<b>44.42(0.37)</b>	59.03(0.57)	-	<b>86.93(0.34)</b>	117.09(0.60)	-	<b>130.17(0.50)</b>	175.43(0.49)

covariance matrices, the spatial and spectral information are jointly integrated in a natural way. Second, with the adopted multiscale strategy, training samples can be enhanced with abundant spatial-spectral information to train the CNN model more effectively.

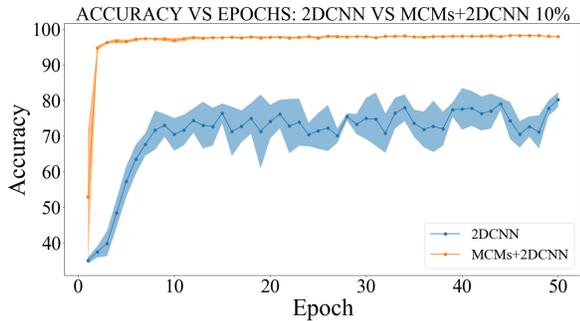
3) *Experiment 3*: In our third experiment, we compared the proposed model with another method, i.e., the pixel-pairs model [1]. In this paper, Li *et al.* proposed a deep CNN architecture which is employed to learn deep pixel-pair features (PPFs). Specifically, on the training stage, the input data of the PPF+2DCNN method is composed by pairs of samples  $\mathbf{S}_{ij} = [\mathbf{x}_i, \mathbf{x}_j]$ , where  $x_i \in \mathbb{R}^{d \times 1}$  and  $x_j \in \mathbb{R}^{d \times 1}$  are the two samples from the training set of hyperspectral scene  $\mathbf{X}$  with  $d$  spectral bands, and categorized with  $y_i \in \{1, 2, \dots, C\}$  and  $y_j \in \{1, 2, \dots, C\}$ , where  $y_i$  and  $y_j$  are the labels and  $C$  is the number of different classes. Each pair  $\mathbf{S}_{ij}$  is labeled as follows:

$$\text{Label}_{\mathbf{S}_{ij}} = \begin{cases} l & \text{if } y_i = y_j = l \\ 0 & \text{if } y_i \neq y_j. \end{cases} \quad (2)$$

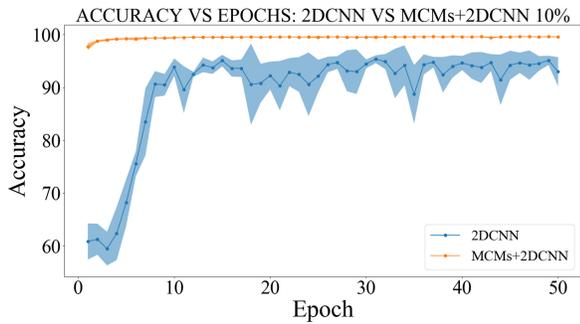
As a result, the method creates  $n_l^{\text{new}} = (((n_l)!)/((n_l - 2)!))$  samples for each class, where  $n_l$  is the number of available labeled training samples in the  $l$ th class. The final number of samples is  $N^{\text{new}} = \sum_{l=1}^{C+1} n_l^{\text{new}}$ , which is much larger than original number of samples. Then, the new pixel-pairs are fed

to a deep 2DCNN (with six convolutional layers and three fully connected layers) for training. On the test stage, they construct the pixel pairs within a local window for each test sample, and the final classification result is also obtained by majority voting. Moreover, the results of a classical 1DCNN classification method [56], which is composed by one convolution layer and two fully connected layers, are also reported here for comparison.

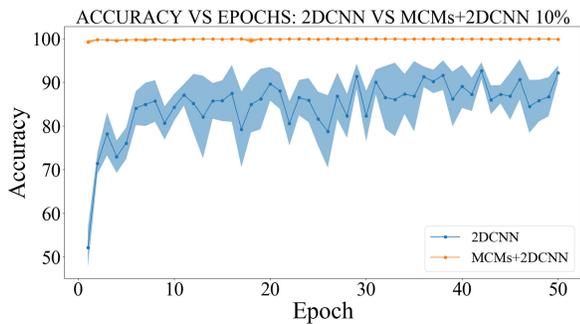
Tables VI–VIII show the classification results obtained for the three considered scenes by the 1DCNN in [56], the PPF+2DCNN in [1], and the proposed MCMs+2DCNN method, trained with 200 randomly selected samples per class. In Table VI, we can see the obtained classification accuracies for the AVIRIS Indian Pines scene, where the OA achieved by the MCMs+2DCNN is better than the OA achieved by the 1DCNN and PPF+2DCNN models. For instance, the OA achieved by the proposed method is 99.10%, while the OA obtained by the PPF is 94.34%. The improvement in classification accuracy is more than 4.5%. Similar results can be observed in Tables VII (ROSIS Pavia University scene) and VIII (AVIRIS Salinas scene). This is mainly due to the fact that MCMs can take into account the spatial-spectral information contained in the HSI for classification purposes, while the PPF only takes the spectral information into consideration. As such, the proposed MCMs method can gain



(a)



(b)



(c)

Fig. 4. Curve of accuracy for the 2DCNN (blue) and the proposed 2DCNN+MCMs (orange) using 10% randomly selected training samples for the classification of (a) Indian Pines data set, (b) Pavia University data set, and (c) Salinas data set. As it can be seen, the proposed MCMs+2DCNN can obtain high and stable test accuracy with less epochs, while without MCMs, the 2DCNN needs more epochs to converge.



Fig. 5. Classification maps obtained for the AVIRIS Indian Pines scene with 10% randomly selected training samples per class. (Left) Original ground truth. (Center) Classical 2DCNN model. (Right) Proposed MCMs+2DCNN model.

higher classification accuracy. The above experiments demonstrate that the proposed method is an effective strategy when compared to another established approach such as the PPF.

4) *Experiment 4:* In this experiment, we compared the proposed method with another classical CNN-based

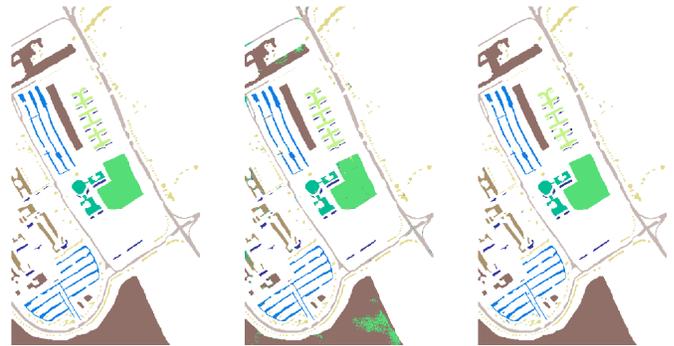


Fig. 6. Classification maps obtained for the ROSIS Pavia University scene with 10% randomly selected training samples per class. (Left) Original ground truth. (Center) Classical 2DCNN model. (Right) Proposed MCMs+2DCNN model.

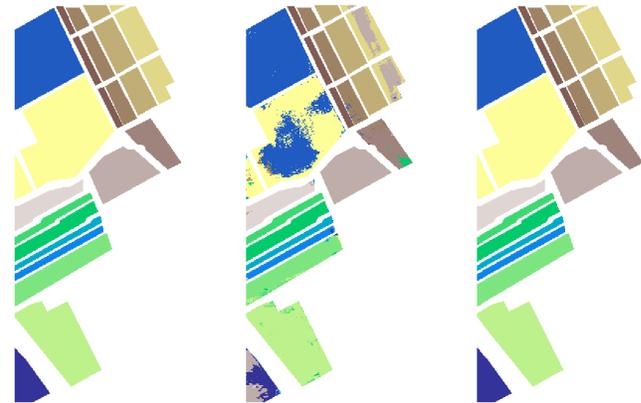


Fig. 7. Classification maps obtained for the AVIRIS Salinas scene with 10% randomly selected training samples per class. (Left) Original ground truth. (Center) Classical 2DCNN model. (Right) Proposed MCMs+2DCNN model.

TABLE VI

COMPARISON BETWEEN THE 1DCNN IN [56], THE PPF+2DCNN IN [1], AND THE PROPOSED MCMs+2DCNN USING THE AVIRIS INDIAN PINES SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. SYMBOL “\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER

Class name	Samples	1DCNN* [56]	PPF+2DCNN* [1]	MCMs+2DCNN Proposed
Corn-notill	200	78.58	92.99	<b>97.61</b> (0.17)
Corn-min	200	85.23	96.66	<b>99.62</b> (0.08)
Grass/Pasture	200	95.75	98.58	<b>100.00</b> (0.00)
Grass/Trees	200	99.81	<b>100.00</b>	99.96(0.08)
Hay-windrowed	200	99.64	100.00	100.00(0.00)
Soybeans-notill	200	89.63	96.24	<b>99.64</b> (0.13)
Soybeans-min	200	81.55	87.80	<b>98.57</b> (0.20)
Soybean-clean	200	95.42	98.98	<b>100.00</b> (0.00)
Woods	200	98.59	99.81	<b>99.98</b> (0.04)
Overall Accuracy	-	86.44	94.34	<b>99.10</b> (0.09)

classification method [2]. Specifically, in [2], the 1DCNN and 2DCNN are used as a feature extraction method followed by the LR classifier. Tables IX and X show the classification results obtained by the 1DCNN and 2DCNN in [2] and our method with the Indian Pines and Pavia University data sets, respectively. As it can be seen from Tables IX and X, the proposed method can obtain better classification performance on both Indian Pines and Pavia University data sets. This is mainly due to the fact that, with the proposed method, the information coming from training samples can be enhanced significantly, which is helpful to train the CNN model more

TABLE VII

COMPARISON BETWEEN THE 1DCNN IN [56], THE PPF+2DCNN IN [1], AND THE PROPOSED MCMs+2DCNN, USING THE ROSIS PAVIA UNIVERSITY SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. SYMBOL “\*\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER

Class name	Samples	1DCNN* [56]	PPF+2DCNN* [1]	MCMs+2DCNN Proposed
Asphalt	200	88.38	97.42	<b>97.85</b> (0.66)
Meadows	200	91.27	95.76	<b>99.20</b> (0.25)
Gravel	200	85.88	94.05	<b>99.91</b> (0.08)
Trees	200	97.24	<b>97.52</b>	94.66(0.72)
Painted metal sheets	200	99.91	<b>100.00</b>	99.72(0.14)
Bare Soil	200	96.41	99.13	<b>99.97</b> (0.04)
Bitumen	200	93.62	96.19	<b>100.00</b> (0.00)
Self-Blocking Bricks	200	87.45	93.62	<b>98.87</b> (0.26)
Shadows	200	99.57	<b>99.60</b>	99.06(0.53)
Overall Accuracy	-	92.27	96.48	<b>98.80</b> (0.18)

TABLE VIII

COMPARISON BETWEEN THE 1DCNN IN [56], THE PPF+2DCNN IN [1], AND THE PROPOSED MCMs+2DCNN, USING THE AVIRIS SALINAS SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. SYMBOL “\*\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER

Class name	Samples	1DCNN* [56]	PPF+2DCNN* [1]	MCMs+2DCNN Proposed
Brocoli-green-weeds-1	200	97.34	<b>100.00</b>	99.99(0.02)
Brocoli-green-weeds-2	200	99.29	99.88	<b>100.00</b> (0.00)
Fallow	200	96.51	99.60	<b>100.00</b> (0.00)
Fallow-rough-plow	200	99.66	99.49	<b>99.83</b> (0.11)
Fallow-smooth	200	96.97	98.34	<b>99.89</b> (0.04)
Stubble	200	99.60	<b>99.97</b>	98.96(0.29)
Celery	200	99.49	100.00	100.00(0.00)
Grapes-untrained	200	72.25	88.68	<b>99.30</b> (0.53)
Soil-vinyard-develop	200	97.53	98.33	<b>100.00</b> (0.00)
Corn-senesced-green-weeds	200	91.29	98.60	<b>100.00</b> (0.00)
Lettuce-romaine-4wk	200	97.58	99.54	<b>99.98</b> (0.05)
Lettuce-romaine-5wk	200	100.00	100.00	99.98(0.03)
Lettuce-romaine-6wk	200	99.02	99.44	<b>100.00</b> (0.00)
Lettuce-romaine-7wk	200	95.05	98.96	<b>100.00</b> (0.00)
Vinyard-untrained	200	76.83	83.53	<b>99.78</b> (0.18)
Vinyard-vertical-trellis	200	98.94	99.31	<b>100.00</b> (0.00)
Overall Accuracy	-	89.28	94.8	<b>99.73</b> (0.04)

effectively. Also, the proposed method can converge faster than the other methods, using less epochs and, therefore, less computation time.

5) *Experiment 5*: To further evaluate the effectiveness of the proposed MCMs+2DCNN, the classification results obtained by combining MCMs with several classical classifiers (i.e., K nearest neighbors (KNN), RF, and SVMs trained with radial basis function (RBF) kernel) are reported in this experiment. Tables XI–XIII show the corresponding classification results comparison on the tested HSI data sets. In addition, to make a more comprehensive comparison, the classification results obtained by combining original spectral features with the KNN, RF, and SVM classifiers are also reported. From Tables XI–XIII, we highlight the following observations. First, MCMs can improve the classification performance of the different classifiers by a large margin, which demonstrates that the proposed method is an effective spatial–spectral feature extraction method for HSIs. Second, the proposed MCMs+2DCNN can outperform the combination of MCMs with other classifiers, which suggests the effectiveness of the MCMs+2DCNN classification framework. Note that, for the KNN, parameter  $K$  is set to 1 with the Euclidean distance metric. For the RF, the number of trees is set to 200, which

TABLE IX

CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CNNs ON THE INDIAN PINES DATA SET. SYMBOL “\*\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER

Class	Neural networks		CNNs* in [2]		Proposed
	Samples	1-D	2-D	MCMs+2DCNN	
Alfalfa	30	89.58	99.65	<b>100.00</b> (0.00)	
Corn-notill	150	85.68	90.64	<b>98.45</b> (0.25)	
Corn-min	150	87.36	<b>99.11</b>	98.32(0.76)	
Corn	100	93.33	100.00	100.00(0.00)	
Grass/Pasture	150	96.88	98.48	<b>99.40</b> (1.06)	
Grass/Trees	150	98.99	97.95	<b>99.90</b> (0.21)	
Grass/pasture-mowed	20	91.67	100.00	100.00(0.00)	
Hay-windrowed	150	99.49	100.00	100.00(0.00)	
Oats	15	100.00	100.00	100.00(0.00)	
Soybeans-notill	150	90.35	95.33	<b>98.52</b> (0.14)	
Soybeans-min	150	77.90	78.21	<b>97.48</b> (0.59)	
Soybean-clean	150	95.82	<b>99.39</b>	98.37(0.48)	
Wheat	150	98.59	<b>100.00</b>	99.64(0.73)	
Woods	150	98.55	97.71	<b>99.89</b> (0.10)	
Bldg-Grass-Tree-Drives	50	87.41	99.31	<b>99.64</b> (0.12)	
Stone-steel towers	50	98.06	<b>99.22</b>	99.07(1.14)	
Overall Accuracy	-	87.81	89.99	<b>98.65</b> (0.20)	
Average Accuracy	-	93.12	97.19	<b>99.29</b> (0.17)	
Kappa	-	85.30	87.95	<b>98.43</b> (0.23)	
Time (seconds)	-	457.8	357.0	<b>80.60</b> (0.44)	

TABLE X

CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CNNs ON THE PAVIA UNIVERSITY DATA SET. SYMBOL “\*\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER

Class	Neural networks		CNNs* in [2]		Proposed
	Samples	1-D	2-D	MCMs+2DCNN	
Asphalt	548	92.06	97.11	<b>99.88</b> (0.18)	
Meadows	540	92.80	87.66	<b>99.80</b> (0.08)	
Gravel	392	83.67	99.69	<b>99.87</b> (0.17)	
Trees	542	93.85	98.49	<b>98.55</b> (0.22)	
Painted metal sheets	256	98.91	100.00	100.00(0.00)	
Bare Soil	532	94.17	98.00	<b>100.00</b> (0.00)	
Bitumen	375	92.68	99.89	<b>100.00</b> (0.00)	
Self-Blocking Bricks	514	89.09	99.70	<b>99.91</b> (0.08)	
Shadows	231	97.84	97.11	<b>97.91</b> (0.54)	
Overall Accuracy	-	92.28	94.04	<b>99.74</b> (0.05)	
Average Accuracy	-	92.55	97.52	<b>99.55</b> (0.07)	
Kappa	-	90.37	92.43	<b>99.64</b> (0.07)	
Time (seconds)	-	994.80	607.19	<b>130.57</b> (0.41)	

follows the settings in [35]. Five-fold cross-validation is adopted to determine the parameters of the SVM classifier.

6) *Experiment 6*: In this experiment, we compare the proposed MCMs+2DCNN method with several popular spectral–spatial classification methods such as EMPs [14], extended morphological attribute profiles (EMAPs) [15], edge-preserving filter (EPF) [16], multiscale adaptive sparse representation (MASR) [9], and the CNN-based classification method in [5]. The corresponding classification results are reported in Tables XIV and XV. As it can be seen, the proposed MCMs+2DCNN is superior to other spatial–spectral methods.

7) *Experiment 7*: In this experiment, we evaluate the effect of the number of scales on the proposed MCMs+2DCNN. As Fig. 8 shows, there are consistent improvements in terms of OA, AA, and Kappa as the number of scales increases, but the number of scales should be less than 15. The best classification accuracy is achieved when 15 scales are used. In addition, growing further the number of scales (e.g., to 17) does not improve the classification accuracy. As a result, the number

TABLE XI

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY DIFFERENT CLASSIFIERS WITH THE ORIGINAL SPECTRAL FEATURES AND MCMS FOR THE AVIRIS INDIAN PINES SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. 10% OF THE AVAILABLE SAMPLES PER CLASS ARE RANDOMLY SELECTED FOR TRAINING

Class name	KNN	RF	SVM	MCMS+KNN	MCMS+RF	MCMS+SVM	MCMS+2DCNN
Alfalfa	44.88(12.27)	19.27(11.67)	14.63(3.86)	100.00(0.00)	95.61(3.05)	100.00(0.00)	100.00(0.00)
Corn-notill	53.96(2.06)	65.45(3.07)	70.93(2.50)	<b>98.83</b> (0.61)	96.50(2.03)	97.69(0.61)	97.96(0.53)
Corn-min	49.56(3.33)	53.79(3.48)	61.37(4.45)	97.72(1.00)	97.63(2.15)	<b>99.20</b> (0.81)	98.45(0.64)
Corn	30.14(1.86)	34.18(7.66)	41.97(7.81)	95.77(1.50)	96.76(2.15)	<b>99.53</b> (0.72)	97.56(1.82)
Grass/Pasture	82.03(2.06)	82.00(3.09)	87.24(2.43)	98.85(0.61)	95.78(1.95)	<b>99.08</b> (0.61)	98.71(0.78)
Grass/Trees	88.68(2.09)	94.00(2.18)	95.28(2.64)	97.11(0.61)	<b>98.78</b> (0.75)	98.02(0.61)	97.41(0.43)
Grass/pasture-mowed	72.00(0.00)	8.00(9.47)	66.40(6.69)	88.00(6.41)	86.80(9.47)	100.00(0.00)	100.00(0.00)
Hay-windrowed	95.07(1.37)	98.49(1.06)	98.14(0.62)	100.00(0.00)	99.47(0.52)	100.00(0.00)	100.00(0.00)
Oats	31.11(22.43)	4.44(5.44)	16.67(9.62)	83.33(6.41)	66.67(23.83)	77.78(6.41)	83.33(7.86)
Soybeans-notill	64.46(2.52)	67.80(3.30)	76.66(2.68)	93.02(0.71)	<b>96.35</b> (1.22)	95.54(0.71)	83.16(3.10)
Soybeans-min	70.03(1.67)	86.64(2.71)	83.43(2.52)	99.00(0.31)	98.64(0.93)	98.74(0.31)	<b>99.36</b> (0.07)
Soybean-clean	40.94(3.29)	48.01(4.97)	55.68(3.40)	92.87(3.05)	94.84(3.81)	92.47(3.05)	<b>98.16</b> (0.22)
Wheat	91.74(2.76)	92.01(3.77)	94.67(3.25)	98.37(0.93)	97.72(1.35)	<b>98.91</b> (0.93)	91.20(3.80)
Woods	90.54(1.21)	95.44(1.41)	94.62(1.48)	99.74(0.83)	98.89(0.96)	99.38(0.83)	<b>99.58</b> (0.21)
Bldg-Grass-Tree-Drives	33.60(4.64)	41.10(4.90)	45.24(2.70)	97.12(0.73)	97.67(1.65)	97.41(0.73)	<b>98.39</b> (0.99)
Stone-steel towers	87.71(1.98)	82.89(3.69)	67.71(4.77)	96.39(1.22)	84.82(6.71)	<b>100.00</b> (0.00)	91.08(5.53)
Overall Accuracy	67.63(0.58)	75.42(0.60)	77.95(0.57)	97.71(0.35)	97.44(0.60)	98.02(0.35)	<b>98.61</b> (0.30)
Average Accuracy	64.15(1.23)	60.84(1.74)	66.92(1.05)	96.01(1.27)	93.93(1.87)	<b>97.11</b> (1.27)	96.94(0.78)
Kappa	63.03(0.66)	71.62(0.70)	74.71(0.62)	97.39(0.39)	97.08(0.69)	97.74(0.39)	<b>98.42</b> (0.34)

TABLE XII

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY DIFFERENT CLASSIFIERS WITH THE ORIGINAL SPECTRAL FEATURES AND MCMS FOR THE AVIRIS PAVIA PINES SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. 10% SAMPLES ARE RANDOMLY SELECTED FOR TRAINING

Class name	KNN	RF	SVM	MCMS+KNN	MCMS+RF	MCMS+SVM	MCMS+2DCNN
Asphalt	86.57(0.63)	90.66(0.18)	93.74(0.54)	99.95(0.05)	99.78(0.27)	100.00(0.00)	100.00(0.00)
Meadows	94.19(0.32)	97.67(0.35)	98.22(0.11)	99.99(0.01)	99.98(0.01)	<b>100.00</b> (0.00)	99.98(0.01)
Gravel	67.13(1.45)	66.26(3.53)	79.59(2.19)	99.68(0.15)	98.82(0.71)	99.84(0.71)	99.84(0.17)
Trees	84.40(0.56)	89.09(0.56)	94.36(0.37)	93.58(1.08)	94.08(0.71)	95.79(0.71)	<b>97.14</b> (0.39)
Painted metal sheets	99.26(0.26)	98.56(0.49)	99.32(0.24)	100.00(0.00)	99.79(0.22)	<b>100.00</b> (0.00)	99.95(0.07)
Bare Soil	64.49(0.63)	65.70(1.62)	89.60(0.56)	100.00(0.00)	99.94(0.04)	100.00(0.00)	100.00(0.00)
Bitumen	81.17(1.57)	76.17(2.10)	86.72(1.41)	100.00(0.00)	99.82(0.18)	99.82(0.04)	<b>99.98</b> (0.03)
Self-Blocking Bricks	81.25(0.91)	88.80(0.95)	90.34(1.71)	99.40(0.60)	99.61(0.27)	98.69(0.22)	<b>99.59</b> (0.26)
Shadows	99.86(0.21)	99.62(0.47)	<b>99.86</b> (0.10)	89.79(1.66)	83.85(2.05)	89.20(2.05)	85.49(1.01)
Overall Accuracy	86.25(0.29)	89.31(0.22)	94.36(0.09)	99.23(0.07)	99.07(0.06)	99.15(0.06)	<b>99.42</b> (0.06)
Average Accuracy	84.26(0.37)	85.84(0.42)	92.42(0.07)	98.04(0.25)	97.30(0.21)	<b>98.17</b> (0.21)	98.00(0.15)
Kappa	81.58(0.38)	85.59(0.30)	92.50(0.12)	98.99(0.09)	98.76(0.08)	99.13(0.08)	<b>99.23</b> (0.07)

TABLE XIII

CLASSIFICATION RESULTS (IN PERCENTAGE) OBTAINED BY DIFFERENT CLASSIFIERS WITH THE ORIGINAL SPECTRAL FEATURES AND MCMS FOR THE AVIRIS SALINAS SCENE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD TYPEFACE. 10% SAMPLES ARE RANDOMLY SELECTED FOR TRAINING

Class name	KNN	RF	SVM	MCMS+KNN	MCMS+RF	MCMS+SVM	MCMS+2DCNN
Broccoli-green-weeds-1	96.53(1.04)	98.67(1.06)	98.30(0.84)	99.35(0.14)	99.37(0.68)	98.44(2.06)	<b>100.00</b> (0.00)
Broccoli-green-weeds-2	98.27(0.92)	99.65(0.30)	99.20(0.36)	99.72(0.29)	99.51(0.92)	97.15(4.03)	<b>100.00</b> (0.00)
Fallow	89.04(7.43)	93.77(2.58)	95.80(3.15)	99.92(0.11)	99.45(1.03)	96.40(3.72)	<b>100.00</b> (0.00)
Fallow-rough-plow	99.39(0.68)	98.00(1.85)	98.28(1.13)	94.86(1.23)	96.20(2.63)	92.46(4.82)	<b>99.71</b> (0.28)
Fallow-smooth	94.34(2.59)	95.35(0.86)	96.04(1.39)	94.04(2.29)	95.93(2.03)	95.06(0.24)	<b>99.89</b> (0.07)
Stubble	99.36(0.32)	98.99(0.22)	98.87(0.73)	98.89(0.78)	99.16(0.76)	98.90(1.34)	<b>99.88</b> (0.12)
Celery	98.77(0.29)	97.76(1.44)	99.37(0.21)	99.56(0.38)	99.45(0.30)	97.76(2.73)	<b>99.86</b> (0.15)
Grapes-untrained	69.38(2.58)	79.98(4.16)	82.83(4.12)	98.32(0.27)	98.17(1.06)	98.72(0.05)	<b>100.00</b> (0.00)
Soil-vinyard-develop	97.05(0.50)	98.26(1.08)	98.73(0.89)	99.69(0.18)	99.83(0.18)	97.72(0.48)	<b>100.00</b> (0.00)
Corn-senesced-green-weeds	80.11(4.49)	82.78(2.84)	87.68(3.59)	98.21(0.35)	99.01(1.09)	97.26(3.84)	<b>100.00</b> (0.00)
Lettuce-romaine-4wk	86.87(2.39)	86.83(4.31)	88.39(4.72)	98.49(2.14)	97.06(2.31)	96.26(5.28)	<b>99.63</b> (0.39)
Lettuce-romaine-5wk	98.15(1.61)	93.59(6.76)	97.93(1.41)	95.75(0.82)	97.21(3.05)	95.10(1.97)	<b>99.92</b> (0.06)
Lettuce-romaine-6wk	96.51(3.31)	97.42(0.69)	97.58(1.03)	97.02(2.34)	95.49(2.68)	96.14(2.97)	<b>100.00</b> (0.00)
Lettuce-romaine-7wk	88.52(2.95)	91.12(1.32)	90.40(2.42)	94.00(1.27)	96.70(3.04)	88.95(3.61)	<b>100.00</b> (0.00)
Vinyard-untrained	56.61(2.90)	55.12(6.27)	57.54(8.60)	96.35(3.61)	95.69(1.26)	94.43(2.77)	<b>100.00</b> (0.00)
Vinyard-vertical-trellis	89.33(3.72)	92.77(2.02)	94.78(2.12)	99.89(0.16)	99.73(0.38)	90.30(7.40)	<b>100.00</b> (0.00)
Overall Accuracy	84.22(0.36)	86.81(0.48)	88.49(0.75)	98.08(0.66)	98.18(0.43)	96.65(1.80)	<b>99.96</b> (0.01)
Average Accuracy	89.89(0.70)	91.25(0.20)	92.61(0.76)	97.75(0.63)	98.00(0.56)	95.69(2.37)	<b>99.93</b> (0.02)
Kappa	82.43(0.40)	85.29(0.53)	87.15(0.85)	97.86(0.73)	97.97(0.48)	96.25(2.03)	<b>99.95</b> (0.02)

of scales in our experiments is set to 15. This is mainly due to the fact that, within a certain range, more scales indicate that more samples with additional spatial-spectral information

are utilized for the CNN training and thus lead to better classification accuracy. On the other hand, increasing too much the number of scales may not be beneficial to classification.

TABLE XIV

CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CLASSIFICATION METHODS OVER PAVIA DATA SET. 200 SAMPLES PER CLASS ARE SELECTED FOR TRAINING. WITH/WITHOUT SYMBOL “\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER OR OBTAINED BY OURSELVES

Method	EMP [14]	EMAP [15]	EPF [16]	MASR [9]	2DCNN* [5]	MCMs+2DCNN
Asphalt	97.33(0.69)	98.57(0.54)	94.75(1.99)	90.37(2.07)	94.60	<b>97.85</b> (0.66)
Meadows	93.19(1.21)	97.86(0.44)	96.14(1.14)	98.79(0.49)	96.00	<b>99.20</b> (0.25)
Gravel	97.93(0.42)	91.89(2.68)	93.52(1.87)	<b>99.94</b> (0.14)	95.50	99.91(0.08)
Trees	98.58(0.47)	<b>99.02</b> (0.60)	95.43(1.11)	97.58(0.41)	95.90	94.66(0.72)
Painted metal sheets	99.83(0.11)	99.65(0.17)	99.63(0.04)	<b>100.00</b> (0.00)	100.00	99.72(0.14)
Bare Soil	95.12(1.37)	98.55(0.50)	98.48(1.21)	99.76(0.32)	94.10	<b>99.97</b> (0.04)
Bitumen	97.43(0.58)	98.51(0.76)	97.89(0.32)	100.00(0.00)	97.50	100.00(0.00)
Self-Blocking Bricks	98.25(0.45)	95.43(2.00)	93.57(0.63)	98.92(0.62)	88.80	<b>98.87</b> (0.26)
Shadows	99.60(0.31)	<b>99.95</b> (0.07)	99.89(0.11)	93.04(2.19)	99.06	99.06(0.53)
Overall Accuracy	95.51(0.39)	97.76(0.17)	96.02(0.48)	97.53(0.49)	95.97	<b>98.80</b> (0.18)

TABLE XV

CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CLASSIFICATION METHODS OVER SALINAS DATA SET. 200 SAMPLES PER CLASS ARE SELECTED FOR TRAINING. WITH/WITHOUT SYMBOL “\*” MEANS THAT THE RESULTS ARE DIRECTLY OBTAINED FROM THIS PAPER OR OBTAINED BY OURSELVES

Method	EMP [14]	EMAP [15]	EPF [16]	MASR [9]	2DCNN* [5]	MCMs+2DCNN
Broccoli-green-weeds-1	99.91(0.10)	100.00(0.00)	99.91(0.20)	100.00(0.00)	100.00	99.99(0.02)
Broccoli-green-weeds-2	99.79(0.17)	99.90(0.07)	99.96(0.06)	100.00(0.00)	100.00	100.00(0.00)
Fallow	99.64(0.32)	99.86(0.12)	100.00(0.00)	100.00(0.00)	100.00	100.00(0.00)
Fallow-rough-plow	99.65(0.19)	99.65(0.43)	99.85(0.14)	99.79(0.18)	99.30	<b>99.83</b> (0.11)
Fallow-smooth	96.70(0.32)	98.33(0.37)	99.06(0.50)	99.44(0.00)	98.50	<b>99.89</b> (0.04)
Stubble	99.86(0.05)	99.66(0.11)	99.93(0.05)	100.00(0.00)	100.00	99.96(0.29)
Celery	99.37(0.37)	99.72(0.16)	99.76(0.11)	100.00(0.00)	99.80	<b>100.00</b> (0.00)
Grapes-untrained	94.28(1.35)	93.64(0.65)	88.42(2.13)	97.32(0.55)	83.40	<b>99.30</b> (0.53)
Soil-vinyard-develop	99.82(0.25)	99.47(0.74)	99.73(0.16)	99.98(0.02)	99.60	<b>100.00</b> (0.00)
Corn-senesced-green-weeds	98.43(0.75)	99.03(0.30)	97.54(0.99)	99.95(0.07)	94.60	<b>100.00</b> (0.00)
Lettuce-romaine-4wk	98.87(0.54)	98.96(0.85)	99.68(0.30)	<b>100.00</b> (0.00)	99.30	99.98(0.05)
Lettuce-romaine-5wk	100.00(0.00)	100.00(0.00)	99.98(0.05)	99.94(0.08)	100.00	99.98(0.03)
Lettuce-romaine-6wk	99.58(0.26)	99.39(0.23)	99.25(0.34)	100.00(0.00)	100.00	100.00(0.00)
Lettuce-romaine-7wk	98.34(0.88)	99.15(0.22)	99.36(0.46)	99.83(0.24)	100.00	100.00(0.00)
Vinyard-untrained	94.92(0.94)	95.61(0.82)	83.06(2.21)	99.63(0.07)	<b>100.00</b>	99.78(0.18)
Vinyard-vertical-trellis	98.92(0.33)	99.49(0.45)	99.43(0.30)	100.00(0.00)	98.00	100.00(0.00)
Overall Accuracy	97.60(0.30)	97.69(0.11)	94.83(0.31)	99.32(0.12)	95.07	<b>99.73</b> (0.04)

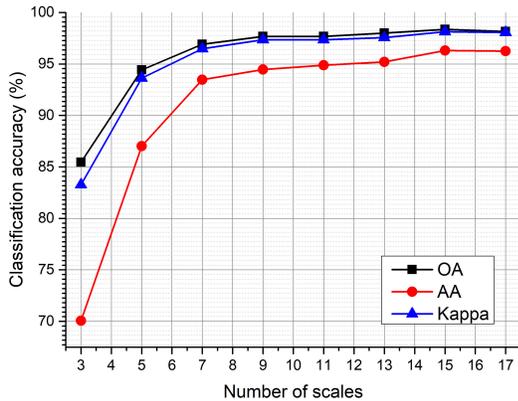


Fig. 8. Effect of the number of scales on the proposed MCMs+2DCNN method. 10% samples are randomly selected per class for training. The average values of 10 experiments are reported, and only the AVIRIS Indian Pines scene is used for illustrative purposes (the results with other scenes are similar).

Note that only the Indian Pines data set is used in this experiment with 10% training samples randomly selected per class for illustrative purposes. The results obtained with other scenes are similar and not displayed here for space considerations.

8) *Experiment 8*: Finally, in this experiment, we report the processing time of the proposed MCMs method when applied to the three considered data sets. As mentioned before, the MCMs method consists of two steps (i.e., MNF-based dimensionality reduction and MCMs calculation). The running

TABLE XVI

IMPLEMENTATION TIME (SECONDS) OF THE PROPOSED MCMs METHOD WHEN APPLIED TO THE THREE CONSIDERED DATA SETS. THE MCMs METHOD CONSISTS OF TWO STEPS (I.E., MNF-BASED DIMENSIONALITY REDUCTION AND MCMs CALCULATION). THE RUNNING TIME OF EACH STEP AND THE TOTAL TIME ARE REPORTED

Procedure	Indian Pines	Pavia	Salinas
MNF dimensionality reduction	0.37	1.64	2.04
MCMs calculation	26.21	103.97	138.63
Total	26.58	105.61	140.67

time of each step and the total processing time are reported in Table XVI. We note that this implementation time can be further reduced via parallel computing, which will be explored in our future developments.

## V. CONCLUSION

In this paper, we propose a novel hand-crafted feature extraction method for the classification of HSIs using CNNs. With our newly developed model, we not only increase the amount of information conveyed by training samples but also naturally integrate the spectral and the spatial-contextual information contained in the original scene, which are both highly beneficial to train a CNN model with powerful generation ability for HSI classification. Experiments on three widely used HSIs indicate that the proposed method provides

classification results that are superior to those obtained by several classical CNN-based and spectral–spatial classification methods. In the future works, we will combine the MCMs with superpixel segmentation approaches in order to explore in more detail the spatial information contained in the original hyperspectral data.

#### ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the three anonymous reviewers for their valuable comments and suggestions, which greatly helped us to improve the technical quality and presentation of this paper.

#### REFERENCES

- [1] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
- [2] 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 Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [3] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.
- [4] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.
- [5] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
- [6] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018. [Online]. Available: <https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=8423220>
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [9] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [11] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, to be published.
- [12] S. Li, R. Dian, L. Fang, and J. M. Bioucas-Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4118–4130, Aug. 2018.
- [13] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: [10.1109/TNNLS.2018.2798162](https://doi.org/10.1109/TNNLS.2018.2798162).
- [14] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
- [15] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.
- [16] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [17] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, to be published.
- [18] J. M. Haut, M. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *J. Supercomput.*, vol. 73, no. 1, pp. 514–529, Jan. 2017.
- [19] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.
- [20] J. M. Haut, M. E. Paoletti, A. Paz-Gallardo, J. Plaza, and A. Plaza, "Cloud implementation of logistic regression for hyperspectral image classification," in *Proc. 17th Int. Conf. Comput. Math. Methods Sci. Eng.*, 2017, pp. 1063–2321.
- [21] J. M. Haut, M. E. Paoletti, J. Plaza, and A. Plaza, "Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines," *J. Real-Time Image Process.*, pp. 1–24, Jun. 2018.
- [22] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
- [23] L. Fang, H. Zhuo, and S. Li, "Super-resolution of hyperspectral image via superpixel-based sparse representation," *Neurocomputing*, vol. 273, pp. 171–177, Jan. 2018.
- [24] L. Fang, S. Li, D. Cunefare, and S. Farsiu, "Segmentation based sparse reconstruction of optical coherence tomography images," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 407–421, Feb. 2017.
- [25] P. Ghamisi, R. Souza, J. A. Benediktsson, X. X. Zhu, L. Rittner, and R. A. Lotufo, "Extinction profiles for the classification of remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5631–5645, Oct. 2016.
- [26] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, "Extinction profiles fusion for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1803–1815, Mar. 2018.
- [27] T. Liu, Y. Gu, J. Chanussot, and M. D. Mura, "Multimorphological superpixel model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 6950–6963, Dec. 2017.
- [28] Z. Wu *et al.*, "GPU parallel implementation of spatially adaptive hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1131–1143, Apr. 2018.
- [29] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao, and Z. Wei, "Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2270–2278, Jun. 2016.
- [30] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
- [31] J. M. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "A new deep generative network for unsupervised remote sensing single-image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, to be published.
- [32] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [33] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5132–5136.
- [34] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.
- [35] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [36] X. Ma, H. Wang, and J. Geng, "Spectral-spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.
- [37] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, May 2015.
- [38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [39] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li, "Integrating spectral and spatial information into deep convolutional Neural Networks for hyperspectral classification," in *Proc. Int. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 5067–5070.
- [40] Z. Zheng *et al.*, "Classification based on deep convolutional neural networks with hyperspectral image," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2017, pp. 1828–1831.

- [41] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
- [42] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sens.*, vol. 8, no. 2, pp. 99–115, Jan. 2016.
- [43] W. Shao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Oct. 2016.
- [44] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 5079–5082.
- [45] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.
- [46] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.
- [47] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral-spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.
- [48] A. Santara *et al.*, "BASS net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.
- [49] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 1, pp. 65–74, Jan. 1988.
- [50] L. Fang, N. He, S. Li, A. J. Plaza, and J. Plaza, "A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3534–3546, Jun. 2018.
- [51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.
- [52] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
- [53] R. O. Green *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.
- [54] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, and H. van der Piepen, "RODIS (Reflective Optics System Imaging Spectrometer)—A candidate instrument for polar platform missions," *Proc. SPIE*, vol. 886, pp. 134–141, Apr. 1988.
- [55] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, to be published, doi: 10.1016/j.isprsjprs.
- [56] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.



**Nanjun He** (S'17) received the B.S. degree from the Central South University of Forestry and Technology, Changsha, China, in 2013. He is currently pursuing the Ph.D. degree with the Laboratory of Vision and Image Processing, Hunan University, Changsha.

From 2017 to 2018, he was a Visiting Ph.D. Student with the Hyperspectral Computing Laboratory, University of Extremadura, Cáceres, Spain, supported by the China Scholarship Council. His research interests include remote sensing image classification and remote sensing object detection.



**Mercedes E. Paoletti** (S'17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, with a University Teacher Training Program from the Spanish Ministry of Education.

Her research interests include remote sensing and analysis of very high-spectral resolution with the current focus on deep learning and high-performance computing.



**Juan Mario Haut** (S'17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Hyperspectral Computing Laboratory, Department of Computers and Communications, with a University Teacher Training Program from the Spanish Ministry of Education.

His research interests include remote sensing and analysis of very high-spectral resolution with the current focus on deep learning and cloud computing

(<https://mhaut.github.io/>)



**Leyuan Fang** (S'10–M'14–SM'17) received the Ph.D. degree from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2015.

From 2011 to 2012, he was a Visiting Ph.D. Student with the Department of Ophthalmology, Duke University, Durham, NC, USA, supported by the China Scholarship Council. From 2016 to 2017, he was a Post-Doctoral Research Fellow with the Department of Biomedical Engineering, Duke University. Since 2017, he has been an Associate Professor with the College of Electrical and Information Engineering, Hunan University. His research interests include sparse representation and multiresolution analysis in remote sensing and medical image processing.

Dr. Fang was a recipient of the Scholarship Award for Excellent Doctoral Student granted by Chinese Ministry of Education in 2011.



**Shutao Li** (M'07–SM'15) received the B.S., M.S., and Ph.D. degrees from Hunan University, Changsha, China, in 1995, 1997, and 2001, respectively.

In 2011, he was a Research Associate with the Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong. From 2002 to 2003, he was a Post-Doctoral Fellow with the Royal Holloway College, University of London, London, U.K., with Prof. J. Shawe-Taylor. In 2005, he was a Visiting Professor with the

Department of Computer Science, Hong Kong University of Science and Technology. In 2001, he joined the College of Electrical and Information Engineering, Hunan University, where he is currently a Full Professor. He has authored or co-authored over 160 refereed papers. His current research interests include compressive sensing, sparse representation, image processing, and pattern recognition.

Dr. Li is currently an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, and an Editorial Board Member of the Information Fusion and the Sensing and Imaging. He was a recipient of two Second-Grade National Awards at the Science and Technology Progress of China in 2004 and 2006.



**Antonio Plaza** (M'05–SM'07–F'15) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 1999 and 2002, respectively.

He is currently the Head of the Hyperspectral Computing Laboratory. He has authored more than 600 publications, including over 200 JCR journal papers (over 160 in the IEEE journals), 23 book chapters, and around 300 peer-reviewed conference proceeding papers. He has guest edited 10 special issues on hyperspectral remote sensing for different journals. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a recipient of the Recognition of Best Reviewers of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010, the Recognition as an Outstanding Associate Editor of the IEEE ACCESS in 2017, the Best Column Award of *IEEE Signal Processing Magazine* in 2015, the 2013 Best Paper Award of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), the most highly cited paper in the *Journal of Parallel and Distributed Computing* from 2005 to 2010, and the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter from 2011 to 2012 and *IEEE Geoscience and Remote Sensing Magazine* in 2013. He was also a member of the Steering Committee of JSTARS. He served as an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2007 to 2012 and currently an Associate Editor for the IEEE ACCESS. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012 and as the President of the Spanish Chapter of IEEE GRSS from 2012 to 2016. He has reviewed more than 500 manuscripts for over 50 different journals. He served as an Editor-in-Chief of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017.



**Javier Plaza** (M'09–SM'15) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 2004 and 2008, respectively.

He is currently a member of the Hyperspectral Computing Laboratory. He has authored more than 150 publications, including over 50 JCR journal papers, 10 book chapters, and 90 peer-reviewed conference proceeding papers. He has guest edited four special issues on hyperspectral remote sensing for different journals. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a recipient of the Outstanding Ph.D. Dissertation Award from the University of Extremadura in 2008, the Best Column Award of *IEEE Signal Processing Magazine* in 2015, the most highly cited paper from 2005 to 2010 in the *Journal of Parallel and Distributed Computing*, and the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He is an Associate Editor of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and an Associate Editor of the IEEE Remote Sensing Code Library.