Hyperspectral Image Classification Using Deep Pixel-Pair Features

Wei Li, *Member, IEEE*, Guodong Wu, *Student Member, IEEE*, Fan Zhang, *Member, IEEE*, and Qian Du, *Senior Member, IEEE*

***Abstract*— The deep convolutional neural network (CNN) is of great interest recently. It can provide excellent performance in hyperspectral image classification when the number of training samples is sufficiently large. In this paper, a novel pixel-pair method is proposed to significantly increase such a number, ensuring that the advantage of CNN can be actually offered. For a testing pixel, pixel-pairs, constructed by combining the center pixel and each of the surrounding pixels, are classified by the trained CNN, and the final label is then determined by a voting strategy. The proposed method utilizing deep CNN to learn pixel-pair features is expected to have more discriminative power. Experimental results based on several hyperspectral image data sets demonstrate that the proposed method can achieve better classification performance than the conventional deep learning-based method.**

***Index Terms*— Convolutional neural network (CNN), deep learning, feature extraction, hyperspectral imagery, pattern classification.**

I. Introduction

HYPERSPECTRAL imagery consists of hundreds of narrow contiguous wavelength bands carrying a wealth of spectral information. Taking advantage of the rich spectral information, classification using hyperspectral data has been developed for a variety of applications [1]–[6], such as land-use land-cover mapping, mineral exploration, water pollution detection, etc.

Among numerous classification methods, k-nearest neighbor (k-NN) [7], [8] can be viewed as the simplest classifier that employs the Euclidean distance to measure the similarity between a testing sample and available training samples. Support vector machine (SVM) [9], [10] is an efficient and stable method for hyperspectral classification tasks, especially for the small training sample sizes. An SVM seeks to separate classes by learning an optimal decision hyperplane that best separates the training samples in a kernel-induced high-dimensional feature space. In [11], one-against-one with decision fusion enables the use of binary kernel local Fisher discriminant analysis [12] for multiclass classification. Some other state-of-the-art pixelwise classifiers, such as relevance vector machine [13] and extreme learning machine (ELM) [14], [15], have been investigated to improve the performance.

Recently, deep learning-based methods have drawn increasing attention in remote sensing image analysis [16], [17]. In [18], an effective deep network was designed for scene classification, which explores the saliency features in the remote sensing scenes for networks construction. Zhang *et al.* [19] systematically survey the general way to construct deep networks for remote sensing images and is the first attempt to evaluate the performance of all the state-of-the-art deep learning methods on remote sensing images. In [20], a deep learning architecture with multilayer stacked autoencoder was proposed to extract high-level features in an unsupervised manner using hyperspectral images. In [21], spectral–spatial features were obtained via deep belief network and the classification procedure was implemented by logistic regression. In [22], spatial updated deep autoencoder was presented to extract spectral–spatial information by adding a regularization term in the energy function. Furthermore, convolutional neural network (CNN) [23] was employed to exploit deep representation based on spectral signatures and the performance proved to be superior to that of SVM; unsupervised sparse features were learned via deep CNN in a greedy layerwise fashion for pixel classification in [24]; and CNN was utilized to automatically find spatial-related features at high levels from a subspace after local discriminant embedding [25].

In this paper, a novel classification framework based on pixel-pair features (PPFs) learned by deep CNN is proposed. In the proposed method, training samples are first paired with any two selected samples using the following criteria—a pair of samples from the same class is labeled with no change while that of samples selected from different classes is labeled as 0. For the training procedure, paired samples with new labels are fed into deep CNN, whose architecture is well designed; during the testing process, for each testing pixel, neighboring pixel-pairs constructed using its surroundings are classified by the trained CNN, and the final label is then determined via a voting strategy based on joint classification results. The reason we chose deep CNN is due to the fact that CNN has been proved to effectively classify hyperspectral data after building an appropriate layer architecture [23]–[25].

For CNN-based feature extraction or classification tasks [23]–[25], the common approach is to learn object features directly from an original hyperspectral data set. A deep model used in this pipeline needs enormous amounts

Manuscript received June 11, 2016; revised August 12, 2016; accepted October 2, 2016. Date of publication November 4, 2016; date of current version December 29, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant NSFC-61571033, by the Fundamental Research Funds for the Central Universities under Grant BUCTRC201401, Grant BUCTRC201615, and Grant XK1521, and by the Higher Education and High-Quality and World-Class Universities (PY201619).

W. Li, G. Wu, and F. Zhang are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: liwei089@ieee.org).

Q. Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: du@ece.msstate.edu).

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.2016.2616355

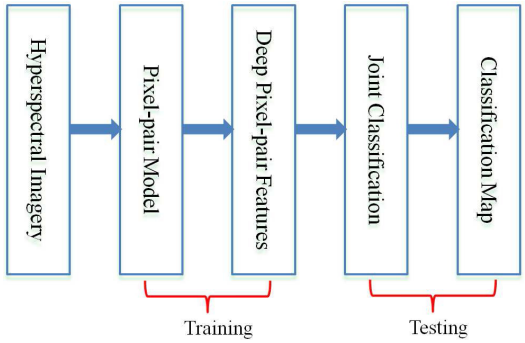


Fig. 1. Flowchart of the proposed classification framework based on deep PPFs.

of training data for tuning a large number of parameters. However, often only a few labeled samples of hyperspectral data are available in practice. To solve this issue, the proposed framework operates on pixel-pair model where a new data combination is constructed via pairing with any two selected samples from the available labeled data and the data entry is relabeled. In doing so, the amount of input data for training exhibits quadratic growth, ensuring the setting of well-tuned parameters. Furthermore, the proposed method fully utilizes the internal correlation of neighbors in hyperspectral imagery, which is ignored by the original CNN.

The main contributions of this paper can be summarized as follows.

1) Deep-learning models are usually heavily parameterized and enormous amounts of training data are required to ensure the performance; however, through reorganizing the available training samples, the proposed pixel-pair strategy is able to overcome this problem.
2) A deep CNN architecture is designed with multiple layers, and then employed to learn deep PPFs, which tend to be more discriminative and reliable.
3) The proposed testing procedure is implemented by a voting strategy based on the fact that neighboring pixels belong to the same class with high probability; the voting fashion that determines the final label makes classification performance more robust, particularly in heterogeneous regions.

This paper is organized as follows. In Section II, the proposed classification framework is described in detail. In Section III, the experimental results and the corresponding analysis are presented. Section IV makes some concluding remarks.

II. Proposed Classification Framework

The proposed classification framework illustrated in Fig. 1 mainly includes three steps: organizing a pixel-pair model using available training samples, constructing a deep CNN architecture to learn PPFs, and determining the label of a testing sample via a voting strategy based on joint classification results.

A. Pixel-Pair Model of Training Samples

Consider a hyperspectral data set with M labeled samples denoted as $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{M}$ in an $\mathbb{R}^{d\times 1}$ feature space and class labels $y_i \in \{1, 2, \ldots, C\}$, where d is the number of bands and C is the number of classes. Let m_l be the number of available labeled samples in the lth class, and $\sum_{l=1}^{C} m_l = M$. In the pixel-pair model, a combined set of two training samples is expressed as $\mathbf{S}_{ij} = [\mathbf{x}_i \quad \mathbf{x}_j]$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two selected samples from $\mathbf{X}$. The label of $\mathbf{S}_{ij}$ is defined using the following criteria: if the two samples are from the same class, the label of $\mathbf{S}_{ij}$ does not change; if the two samples are from different classes, the label of $\mathbf{S}_{ij}$ is set to be 0. That is

$$\text{Label}(\mathbf{S}_{ij}) = \begin{cases} l, & \text{if } y_i = y_j = l \\ 0, & \text{if } y_i \neq y_j. \end{cases} \tag{1}$$

Note that the new set of labeled samples $\mathbf{X}^{\text{new}} = \{\mathbf{S}_{ij}\}_{i,j=1}^{M}$ has $C+1$ classes with $\mathbf{S}_{ij} \in \mathbb{R}^{d\times 2}$. Fig. 2 illustrates the process of constructing $\mathbf{X}^{\text{new}}$ in $C+1$ classes. For the lth class ($l = 1, 2, \ldots, C$), the number of pairs (denoted as n_l) can be calculated using all permutations (the order is relevant), $n_l = P_2^{m_l} = (((m_l)!)/((m_l-2)!))$. For the class with $\text{Label}(\mathbf{S}_{ij}) = 0$, n_l (i.e., $l = C+1$) is larger since samples can be chosen from any two different classes; however, to keep data balanced, only an approximately equal number of pairs is determined. Thus, the number of total samples in $\mathbf{X}^{\text{new}}$ is calculated as $\sum_{l=1}^{C+1} n_l$, which is much larger than the original size M.

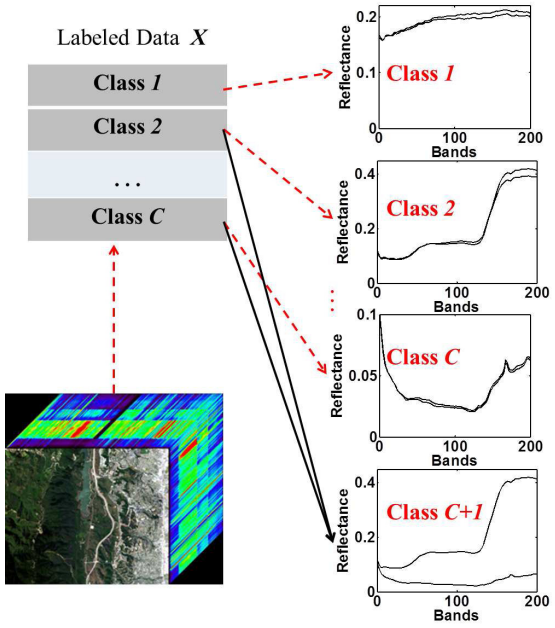


Fig. 2. Process of organizing a pixel-pair model using available training samples. Note that for the $(C+1)$th class, samples can be chosen from any two different classes.

B. Feature Extraction Using CNN Architecture

After obtaining the input samples, a deep CNN architecture is used to extract PPFs, as illustrated in Fig. 3. CNNs represent feedforward neural networks that consist of various combinations of convolutional layers, max-pooling layers, and fully connected layers, and exploit spatially local correlation by enforcing local connectivity between neurons in

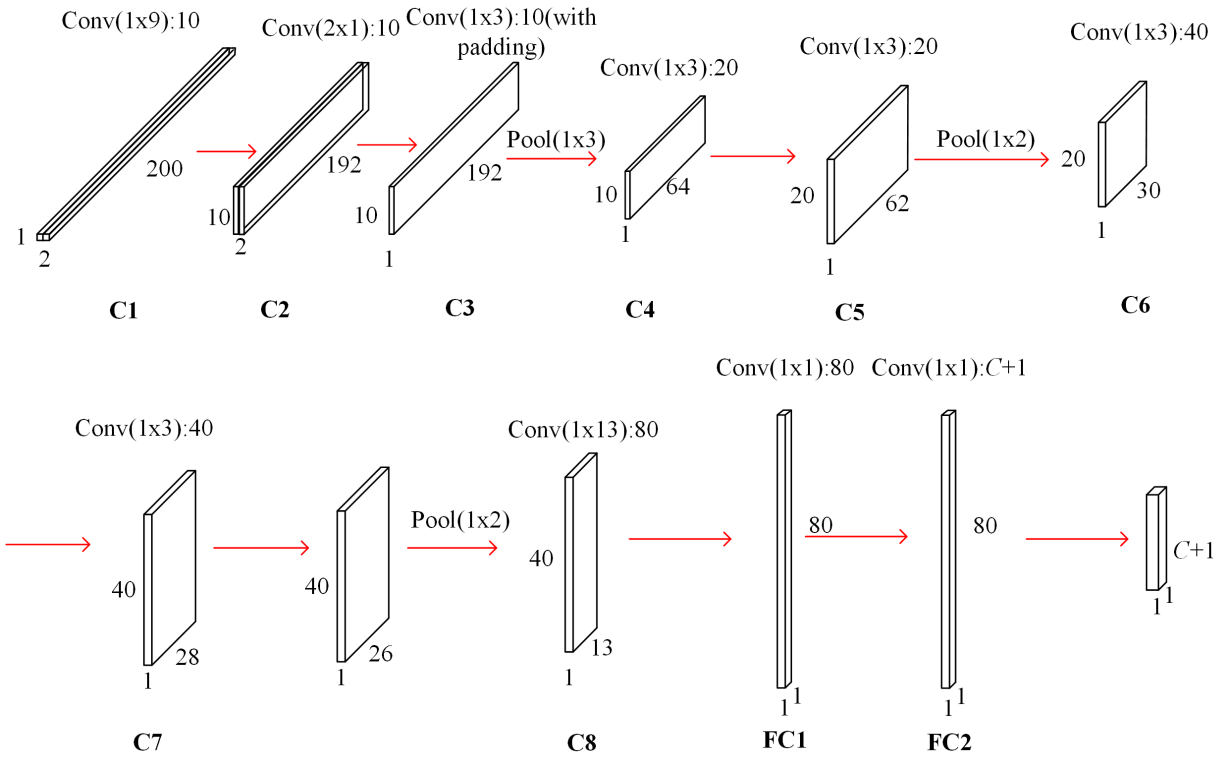


Fig. 3. PPF extraction from a deep CNN architecture with convolutional layers and max-pooling layers. Input data are the pairs of available training samples (assume $d = 200$).

adjacent layers. The designed CNN architecture contains ten learnable convolutional layers, three max-pooling layers, and rectified linear unit (ReLU) layers [26] after each convolutional layer. Following are the details of the proposed framework.

Assume the input dimensionality $d = 200$. The first convolutional layer (C1) primarily filters the $2 \times 200 \times 1$ prepared data with ten $1 \times 9 \times 1$ kernels, producing a $2 \times 192 \times 10$ tensor (i.e., $192 = 200 - 9 + 1$, without padding). The second layer (C2) combines the features obtained in the C1 layer with ten $2 \times 1 \times 10$ kernels, resulting in a $1 \times 192 \times 10$ tensor. The tensor obtained in the C2 layer mainly measures the similarity between the PPFs. A convolutional layer is usually followed by a pooling layer. In order to produce high-level features, the network goes deeper with more convolutional layers and pooling layers.

The following layers include 1×3 convolutional layers, max-pooling layers, and ReLU. A pooling layer is added on top of the convolution layer to compute a lower resolution representation of the convolution layer activations through subsampling. In Fig. 3, the third layer (C3) filters the tensor with ten $1 \times 3 \times 10$ kernels (for numerical consideration), producing the tensor with the same shape as the C2 layer, then a 1×3 max-pooling layer is used to reduce the spectral dimension. It is worth mentioning that two 1×3 convolutional layers are used instead of one 1×5 layer in order to increase the nonlinearity of the model and reduce the number of parameters [26]. Furthermore, the number of features is doubled after each pooling layer (e.g., there are twenty $1 \times 3 \times 10$ convolutional kernels in the C4 layer, generating a $1 \times 62 \times 20$ tensor). There are three pooling layers in the model with kernel sizes 1×3, 1×2, and 1×2, respectively. A 1×2 max-pooling is adopted instead of 1×3 kernel to preserve more information as the deeper the layer is, the more the useful features used for the following classification.

In the designed deep CNN architecture, three are three max-pooling layers used to reduce the spectral dimensionality. Once the spectral dimension is reduced to a desired value (e.g., 13), the C8 tensor is fed into two fully connected layers (i.e., FC1 and FC2). The fully connected layer in a traditional neural network is transformed into a convolutional layer in the proposed architecture, e.g., eighty $1 \times 13 \times 40$ kernels are operated on the C8 layer to form a $1 \times 1 \times 80$ tensor. To be used for classification, the chain of the CNN architecture ends in a fully connected network with softmax and label information (layer).

Similar to the traditional neural network, the training process of CNN contains two steps: forward-propagation and back-propagation. The former aims at computing the classification performance of the input data with current parameters, while the latter is employed to update the trainable parameters. In Fig. 3, suppose $d = 200$, $C = 9$, and 200 training samples per class are available. The total number of trainable parameters in the designed architecture can be estimated to be approximately 50 000, which is much larger than that of all the training data (i.e., $9 \times 200 = 1800$); however, based on the proposed pixel-pair model, the number of input pairs for the network becomes 398 000 (i.e., $200 \times 199 \times 10$). This example verifies that the pixel-pair model can generate sufficient input data to learn the parameters in the CNN architecture.

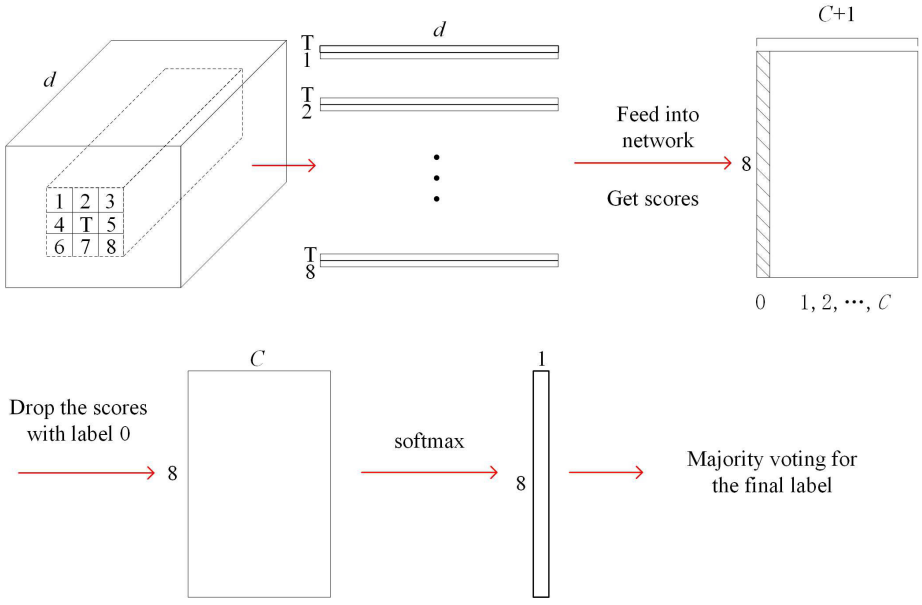


Fig. 4. Joint classification with voting strategy based on deep PPFs.

C. Joint Classification With Voting Strategy

In hyperspectral imagery, neighboring pixels tend to belong to the same class with high probability, which inspires us to build a joint classification with voting strategy during the testing process. For a testing pixel, we also construct pixel-pairs with surrounding samples, which are then fed into the trained CNN architecture as illustrated in Fig. 4. The output of neural network is a tensor with $C+1$ dimensionality, where each row means the probability scores of each pair belonging to these classes (i.e., classes with label from 0 to C, and class $C+1$ with label 0 will not be considered further).

In Fig. 4, a block window is employed to neighboring pixels (the size can be 3×3, 5×5, 7×7, etc.), and we take 3×3 for example. The center pixel is expressed as T; thus, there are eight pairs (i.e., $\{T,1\},\{T,2\},\ldots,\{T,8\}$) fed into CNN. The output of the network is an $8\times(C+1)$ matrix with scores. Because the first column exhibits pairs of two samples from different classes (the label of pairs is 0), it can be removed. After dropping the first column, the resulting $8\times C$ matrix is followed by a softmax layer, yielding an 8×1 vector, which indicates the labels for all the pairs. Subsequently, the final label of the center pixel is determined with majority voting strategy based on the 8×1 vector.

Fig. 5 illustrates an example of joint classification using neighboring pixels in a 3×3 window. The pair label matrix demonstrates the results after the softmax process, and the final label of the central testing pixel (i.e., T) can be determined via a majority voting strategy. It is worth mentioning that if the window-size is changed to 5×5, there will be 24 pairs for decision.

III. Experimental Results

For the proposed deep CNN with pixel-pair features (denoted as CNN-PPF), all the programs are implemented using Python language and TensorFlow[1] library. TensorFlow is an open source software library for numerical computation using data flow graphs. Computation can be easily deployed to one or more CPUs or GPUs with TensorFlow.

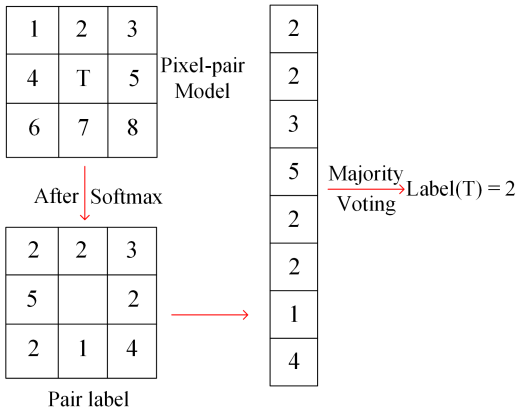


Fig. 5. Example of joint classification with a 3×3 window.

TABLE I
NUMBER OF TRAINING AND TESTING SAMPLES USED IN THE INDIAN PINES DATA SET

No.	*Class*	*Training*	*Testing*
1	Corn-notill	200	1228
2	Corn-mintill	200	630
3	Grass-pasture	200	283
4	Grass-trees	200	530
5	Hay-windrowed	200	278
6	Soybean-notill	200	772
7	Soybean-mintill	200	2255
8	Soybean-clean	200	393
9	Woods	200	1065
	TOTAL	1800	7434

[1] http://tensorflow.org/

TABLE II

NUMBER OF TRAINING AND TESTING SAMPLES USED IN THE SALINAS DATA SET

No.	*Class*	*Training*	*Testing*
1	Brocoli_green_weeds_1	200	1809
2	Brocoli_green_weeds_2	200	3526
3	Fallow	200	1776
4	Fallow_rough_plow	200	1194
5	Fallow_smooth	200	2478
6	Stubble	200	3759
7	Celery	200	3379
8	Grapes_untrained	200	11071
9	Soil_vinyard_develop	200	6003
10	Corn_senesced_green_weeds	200	3078
11	Lettuce_romaine_4wk	200	868
12	Lettuce_romaine_5wk	200	1727
13	Lettuce_romaine_6wk	200	716
14	Lettuce_romaine_7wk	200	870
15	Vinyard_untrained	200	7068
16	Vinyard_vertical_trellis	200	1607
	TOTAL	3200	50929

TABLE III

NUMBER OF TRAINING AND TESTING SAMPLES USED IN THE UNIVERSITY OF PAVIA DATA SET

No.	*Class*	*Training*	*Testing*
1	Asphalt	200	6431
2	Meadows	200	18449
3	Gravel	200	1899
4	Trees	200	2864
5	Sheets	200	1145
6	Bare Soil	200	4829
7	Bitumen	200	1130
8	Bricks	200	3482
9	Shadows	200	747
	TOTAL	1800	40976

TABLE IV

CLASSIFICATION (%) PERFORMANCE OF DIFFERENT NUMBERS OF FEATURES AND WINDOWS-SIZE USING THE UNIVERSITY OF PAVIA DATA

	Window Size		
No. of Features	3×3	5×5	7×7
5	95.62	96.03	96.29
8	95.55	96.16	96.38
10	95.66	96.48	96.51
12	95.28	95.85	96.04
15	95.13	95.82	96.01

TABLE V

CLASSIFICATION ACCURACY (%) WITH AND WITHOUT FULLY CONNECTED LAYER USING THE UNIVERSITY OF PAVIA DATA

WindowSize	3×3	5×5	7×7
With FC	95.66	96.48	96.51
Without FC	94.85	95.46	95.77

TABLE VI

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE INDIAN PINES DATA

	k-NN	SVM	ELM	SVM-RFS	CNN	CNN-PPF
1	61.83	83.61	86.06	88.73	78.58	92.99
2	72.65	87.23	88.19	91.20	85.23	96.66
3	95.65	98.34	96.07	97.52	95.75	98.58
4	98.90	99.73	99.73	99.86	99.81	100
5	100	100	100	100	99.64	100
6	80.76	88.17	90.02	91.67	89.63	96.24
7	59.39	76.58	71.00	78.79	81.55	87.80
8	75.72	94.94	95.62	93.76	95.42	98.98
9	94.86	98.89	98.66	98.74	98.59	99.81
OA	76.24	88.26	87.33	89.83	86.44	**94.34**

TABLE VII

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE SALINAS DATA

	k-NN	SVM	ELM	SVM-RFS	CNN	CNN-PPF
1	98.71	99.60	99.75	99.55	97.34	100
2	99.65	100	99.87	99.92	99.29	99.88
3	99.09	99.65	99.60	99.44	96.51	99.60
4	99.78	99.64	99.64	99.86	99.66	99.49
5	95.29	98.39	98.81	98.02	96.97	98.34
6	99.49	99.70	99.67	99.70	99.60	99.97
7	99.55	99.72	99.66	99.69	99.49	100
8	63.53	84.38	84.04	84.85	72.25	88.68
9	95.94	99.65	99.89	99.58	97.53	98.33
10	91.98	96.74	95.03	96.49	91.29	98.60
11	98.41	98.31	96.82	98.78	97.58	99.54
12	99.84	99.95	100	100	100	100
13	98.69	99.24	98.25	99.13	99.02	99.44
14	97.38	98.88	97.94	98.97	95.05	98.96
15	65.66	74.59	72.96	76.38	76.83	83.53
16	99.00	99.39	99.06	99.56	98.94	99.31
OA	86.29	92.85	92.42	93.15	89.28	**94.80**

TABLE VIII

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE UNIVERSITY OF PAVIA DATA

	k-NN	SVM	ELM	SVM-RFS	CNN	CNN-PPF
1	77.70	86.46	81.32	87.95	88.38	97.42
2	75.30	90.17	90.91	91.17	91.27	95.76
3	77.27	85.04	85.09	86.99	85.88	94.05
4	92.46	96.64	96.61	95.50	97.24	97.52
5	99.63	99.78	99.63	99.85	99.91	100
6	79.50	94.89	94.33	94.31	96.41	99.13
7	92.86	95.19	95.94	94.74	93.62	96.19
8	76.45	85.36	82.65	85.89	87.45	93.62
9	99.62	99.89	99.79	99.89	99.57	99.60
OA	79.45	90.62	89.86	91.10	92.27	**96.48**

A. Experimental Data

Three hyperspectral data,[2] including Indian Pines, Salinas, and University of Pavia scenes, are employed to evaluate the effectiveness of the proposed CNN-PPF. For all the data, we randomly select 200 labeled pixels per class for training and all the other pixels in the ground-truth map for testing. Development data are derived from the available training data by further dividing them into training and testing samples for tuning the parameters of the proposed method. Furthermore, each pixel is uniformly scaled to the range of 0–1.

The Indian Pines data set was gathered by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor in northwestern Indiana. There are 220 spectral channels in the 0.4–45 μm region of the visible and infrared spectrum with a spatial resolution of 20 m. There are 16 different land-cover classes in the original ground truth; however, only 9 classes are used in this paper so as to avoid a few classes that have

[2]http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

TABLE IX
STATISTICAL SIGNIFICANCE FROM THE STANDARDIZED McNEMAR'S TEST ABOUT THE DIFFERENCE BETWEEN METHODS

Indian Pines	Salinas	University of Pavia
Z/significant?	Z/significant?	Z/significant?
CNN-PPF *vs* SVM-RFS		
15.70/yes	19.26/yes	30.95/yes
CNN-PPF *vs* CNN		
24.33/yes	36.84/yes	24.59/yes
CNN-PPF *vs* ELM		
21.29/yes	20.69/yes	39.30/yes
CNN-PPF *vs* SVM		
19.27/yes	21.93/yes	36.03/yes
CNN-PPF *vs* k-NN		
37.81/yes	54.45/yes	75.10/yes

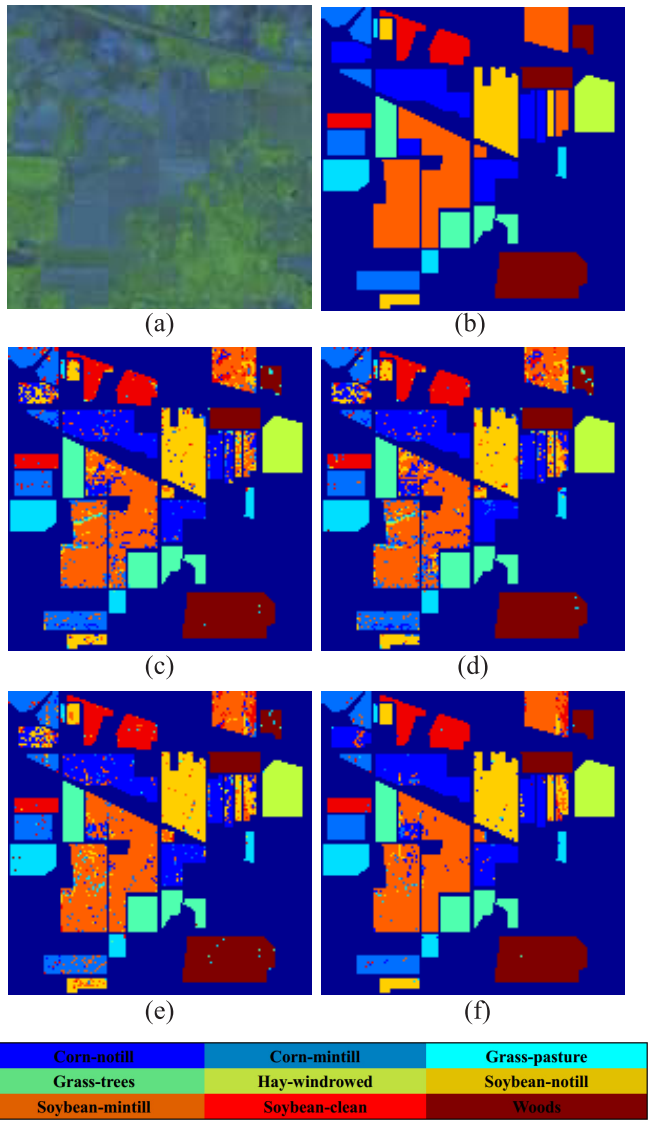


Fig. 6. Thematic maps resulting from classification for the Indian Pines data set with nine classes. (a) Pseudocolor image. (b) Ground-truth map. (c) SVM: 88.26%. (d) ELM: 87.33%. (e) CNN: 86.44%. (f) CNN-PPF: 94.34%.

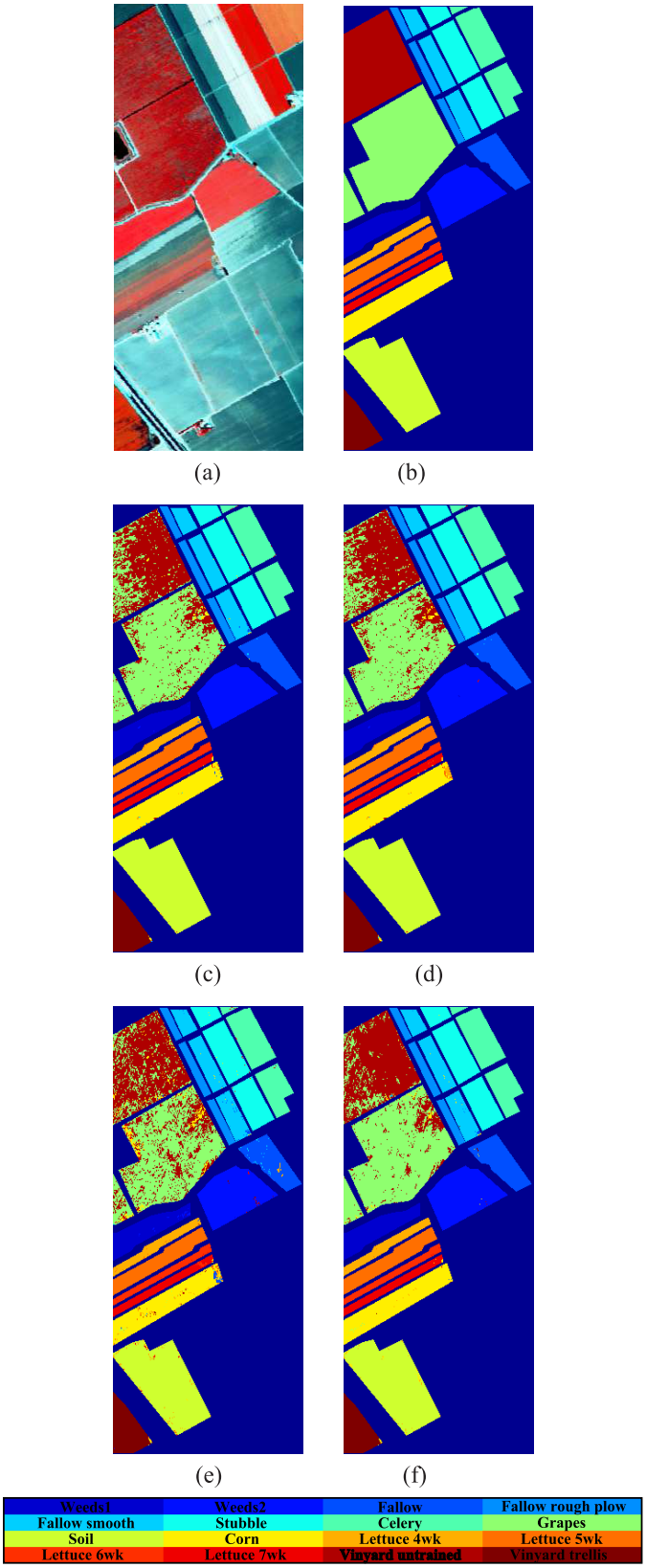


Fig. 7. Thematic maps resulting from classification for the Salinas data set with 16 classes. (a) Pseudocolor image. (b) Ground-truth map. (c) SVM: 92.85%. (d) ELM: 92.42%. (e) CNN: 89.72%. (f) CNN-PPF: 94.80%.

very few training samples [27]. The numbers of training and testing samples are listed in Table I.

The second data were also collected by the AVIRIS sensor over Salinas Valley, California. The image comprises

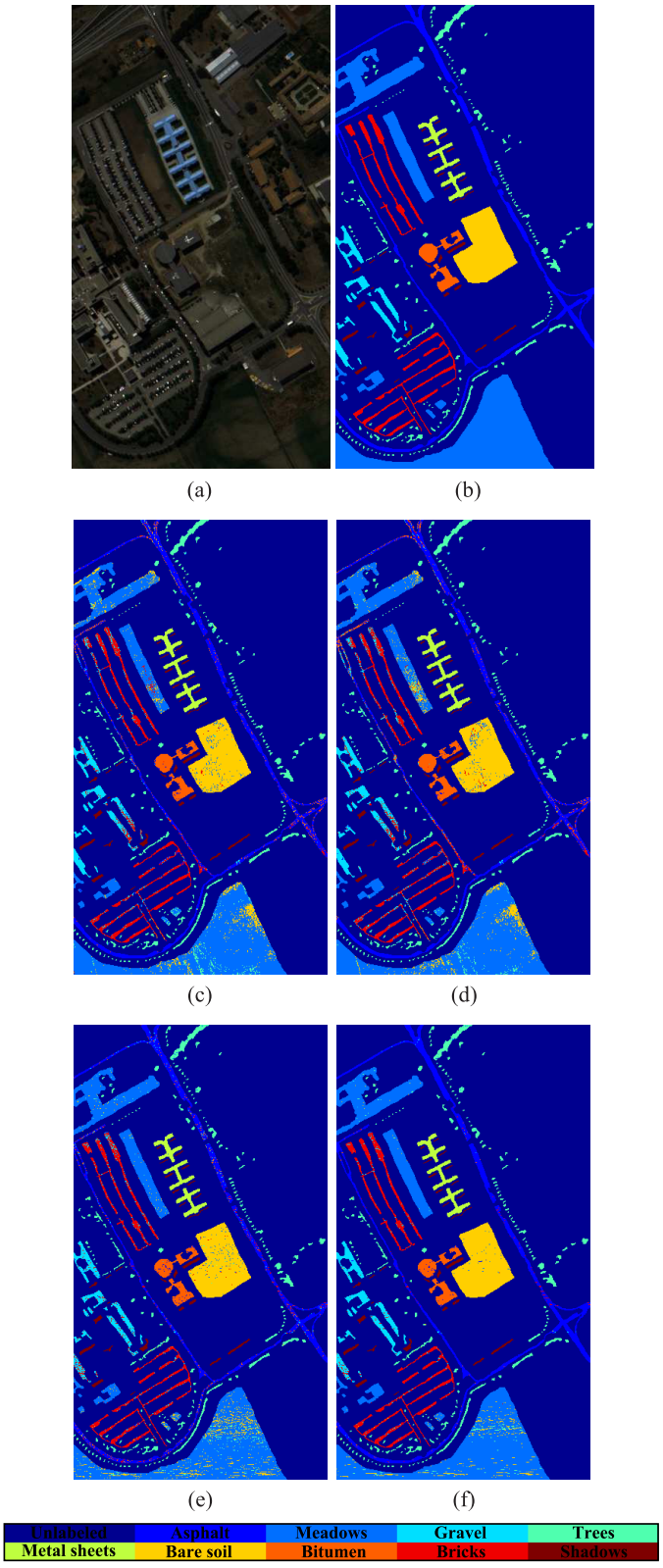


Fig. 8. Thematic maps resulting from classification for the University of Pavia data set with nine classes. (a) Pseudocolor image. (b) Ground-truth map. (c) SVM: 90.62%. (d) ELM: 89.86%. (e) CNN: 92.27%. (f) CNN-PPF: 96.48%.

512 × 217 pixels with a spatial resolution of 3.7 m and 204 bands after 20 water absorption bands are removed. It mainly contains vegetables, bare soils, and vineyard fields.

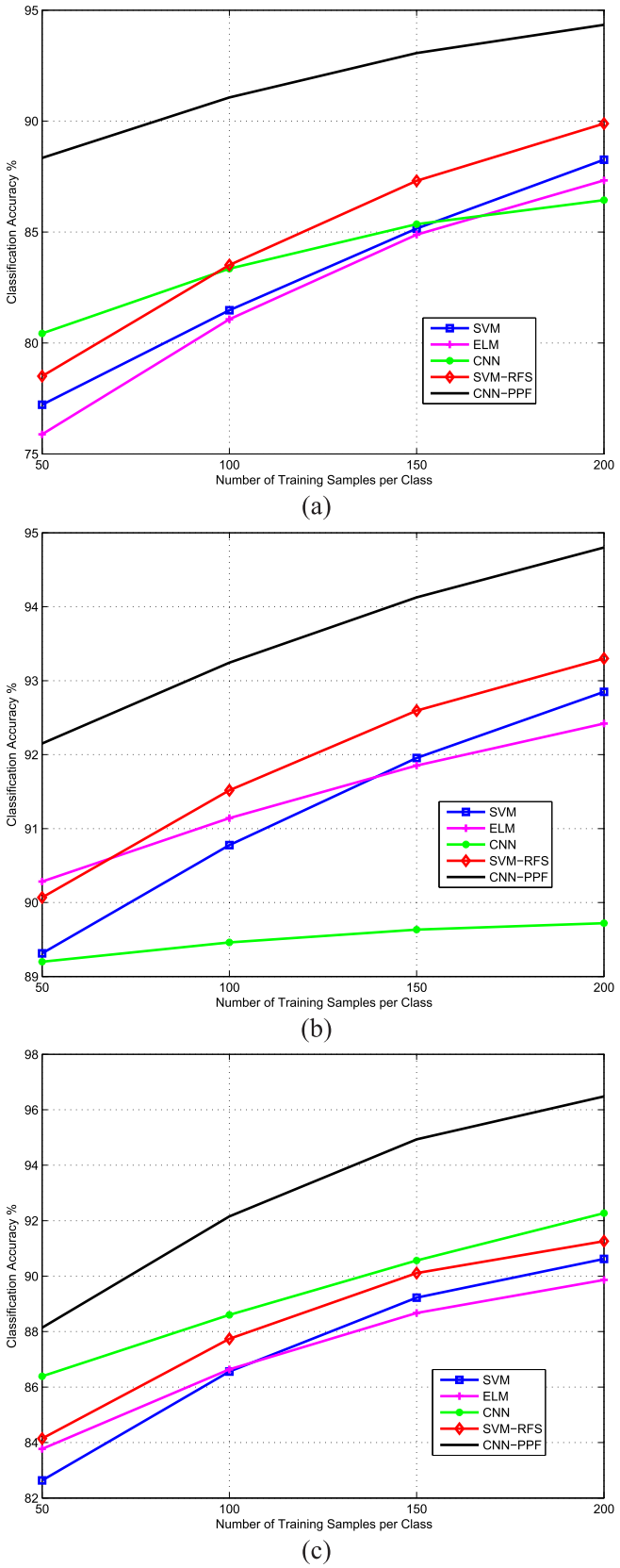


Fig. 9. Classification performance of methods with different numbers of training sample sizes using the experimental data sets. (a) Indian Pines data. (b) Salinas data. (c) University of Pavia data.

There are also 16 classes, and the number of training and testing samples are listed in Table II.

The University of Pavia data set was collected by the reflective optics system imaging spectrometer sensor. The image scene, with 610 × 340 pixels covering the city of Pavia, Italy, was collected under the HySens project managed by DLR (the German Aerospace Agency). The data set has 103 spectral bands prior to waterband removal. It has a spectral coverage from 0.43 to 0.86 μm and a spatial resolution of 1.3 m. Approximately 42 776 labeled pixels with nine classes are from the ground-truth map, and the numbers of training and testing samples are listed in Table III.

B. Parameter Tuning

Except weights that can be automatically learned during training, there are several other important parameters in the designed CNN architecture, such as learning rate, dimensionality of features, and window size. Learning rate determines the convergence speed in the procedure of back-propagation [23], [28], which can significantly affect the training performance. In practical implementation, we initially set the learning rate as 0.1 and decrease it by dividing 10 if the learning curve fluctuates too much. According to our empirical study, the best learning rate is 0.001, 0.01, and 0.001 for the Indian Pines, University of Pavia, and Salinas data, respectively. Take the University of Pavia data for example, we test various learning rate, i.e., 0.1, 0.01, 0.001, and the corresponding accuracies (%) are 1.8, 95.91, and 91.00, respectively. It is found that a large learning rate (e.g., 0.1) may reduce the classification accuracy or even make the network diverge.

In the framework of CNN, the number of features determines the dimensionality of the extracted PPFs in the convolutional layer (C1) as shown in Fig. 3. Furthermore, the window size during testing can affect the final classification accuracy as indicated in Fig. 4. Take the University of Pavia for example, Table IV demonstrates the effect of the number of features and the size of the voting window, which shows that the best number of features is 10. When the window size is 5 × 5 or 7 × 7, the performance seems to be very close, while the latter may cause a higher computational cost; thus, 5 × 5 is chosen to be the optimal window size. According to experiments, the optimal window size is 5 × 5 for both the other data, and 10 and 17 are chosen to be the optimal number of features for the Indian Pines and Salinas data, respectively.

Note that in Fig. 3, the C8 tensor is fed into two fully connected layers (FC1 and FC2) rather than being connected with the final label layer. Although the fully connected layers increase the number of parameters, we find the fully connected layers still help, at least in the proposed CNN architecture. Table V demonstrates the classification performance as a function of different window sizes with (i.e., $with\ FC$) and without (i.e., $without\ FC$) fully connected layers using the University of Pavia data. It is obvious that the results of $with\ FC$ are always better than those of $without\ FC$, which verifies the need for employing fully connected layers in the proposed framework.

C. Classification Performance

To demonstrate the effectiveness of the proposed CNN-PPF, we compare with several traditional classifiers, such as k-NN, SVM, ELM [14], [15], CNN [23], and multiple classifier systems based on SVM and random feature selection (SVM-RFS) [29]. For a fair comparison with CNN, the number of training and testing samples is exactly the same as [23]. SVM with radial basis function kernel is implemented using the *libsvm* toolbox.[3] ELM is downloaded from the Web page.[4] All the classifiers are operated with optimal parameters.

Tables VI–VIII list the class-specific accuracy and overall accuracy (OA) for these three experimental data. From the results of each individual method, the proposed CNN-PPF is obviously superior to CNN and all the other classifiers. For example, in Table VI, CNN-PPF (i.e., 94.34%) yields over 8% higher accuracy than CNN (i.e., 86.44%), and approximately 6% accuracy higher than SVM (i.e., 88.26%). Especially for some classes, such as *Corn-mintill* and *Soybean-mintill*, the class-specific accuracy of the proposed CNN-PPF is even approximately 11% higher than that of CNN. The similar situations happen to the other two experimental data.

The standardized McNemar's test [30] is employed to demonstrate the statistical significance in accuracy improvement of the proposed CNN-PPF as listed in Table IX. The Z values of McNemar's test larger than 1.96 and 2.58 mean that two results are statistically different at the 95% and 99% confidence levels, respectively. The sign Z indicates whether classifier 1 outperforms classifier 2 ($Z > 0$) or vice versa. In the experiment, we design the comparison between CNN-PPF and four other methods, i.e., SVM-RFS, ELM, SVM, and k-NN. In Table IX, all the values are much larger than 2.58, which confirms that the proposed CNN-PPF can significantly outperform these traditional methods.

Figs. 6–8 show the thematic maps. We produced ground-cover maps of entire image scenes (including unlabeled pixels). However, to facilitate comparison between methods, with ground truth are shown in these maps. These maps are consistent with the results listed in Tables VI–VIII, respectively. Some areas in the classification maps produced by the proposed CNN-PPF are obviously less noisy than those of SVM, ELM, and CNN, e.g., the regions of *Meadows* in Fig. 8. Actually, the visual results are consistent with those in Tables VI–VIII.

Fig. 9 illustrates the classification performance with different numbers of training samples. Usually, the number of training samples available may be insufficient to estimate models for each class in practical situations, which is necessary to investigate the sensitivity of the training sizes. As shown in Fig. 9, the number of training samples per class is changed from 50 to 200 with an interval of 50. From the results, CNN-PPF still consistently performs better than the three other methods, i.e., SVM-RFS, SVM, ELM, and CNN. For example, CNN-PPF always has an 8% improvement compared with CNN for the Indian Pines data in Fig. 9(a), which verifies that the pixel-pair method works well. Actually, the

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[4] http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

TABLE X
EXECUTION TIME (h: HOURS; s: SECONDS) OF TRAINING AND TESTING PROCEDURES IN THE THREE EXPERIMENTAL DATA SETS

		Indian Pines	Salinas	University of Pavia
CNN	training (h)	0.5	1.0	0.6
	testing (s)	0.21	0.26	0.37
CNN-PPF	training (h)	6.0	12.0	4.0
	testing (s)	4.76	20.97	16.92

reasons can be summarized as follows: 1) sufficient input data guarantees the network parameters to be well-tuned; 2) deep CNN learns PPFs with more discriminative power; and 3) the voting procedure further ensures the accuracy and reliability.

Table X summarizes the computational complexity of training and testing procedures using the original CNN [23] and the proposed CNN-PPF. All the experiments were carried out using a PC equipped with a single QUADRO K2200 GPU. Note that the execution time of CNN is much less than that of CNN-PPF for both the training and testing procedures. For the training procedure, the reasons can be that the number of convolutional layers and fully connected layers in the former is less than that of the latter; furthermore, the size of input data (to the network) of the former is much smaller than that of the latter (numerical analysis can be found in Section II-B). In the testing procedure, CNN-PPF is more time-consuming due to the computational burden of joint classification based on pixel-pair model.

IV. CONCLUSION

In this paper, a CNN-based classification framework based on deep PPFs was proposed. The pixel-pair model is to exploit the similarity between pixels and ensure a sufficient amount of input data to learn a large number of parameters in the CNN, including ten learnable convolutional layers and three max-pooling layers. In the testing procedure, the surrounding pixels were combined with the central testing pixel to fit the pixel-pair model, and the final label was determined via a majority voting strategy. The experimental results with real hyperspectral images demonstrated that the proposed CNN-PPF can greatly improve the performance of the original CNN, even with a small number of training samples with the sacrifice of increased computational cost.

REFERENCES

[1] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[2] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, 2015.

[3] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 257–272, Jan. 2013.

[4] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6844–6857, Nov. 2014.

[5] W. Li and Q. Du, "Gabor-filtering-based nearest regularized subspace for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1012–1022, Apr. 2014.

[6] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 34–44, Jan. 2014.

[7] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[8] W. Li, Q. Du, F. Zhang, and W. Hu, "Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 389–393, Feb. 2015.

[9] L. Gao *et al.*, "Subspace-based support vector machines for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 349–353, Feb. 2015.

[10] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.

[11] W. Li, S. Prasad, and J. E. Fowler, "Decision fusion in kernel-induced spaces for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3399–3411, Jun. 2014.

[12] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 894–898, Sep. 2011.

[13] F. A. Mianji and Y. Zhang, "Robust hyperspectral classification using relevance vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2100–2112, Jun. 2011.

[14] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[15] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, "E^2LMs: Ensemble extreme learning machines for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, Apr. 2014.

[16] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 20, no. 1, pp. 1–12, 2015.

[17] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[18] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[19] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[20] 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.

[21] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[22] 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.

[23] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, 2015, Art. no. 258619, doi: 10.1155/2015/258619.

[24] 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.

[25] W. Zhao 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, Aug. 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," unpublished paper. [Online]. Available: https://arxiv.org/abs/1409.1556

[27] 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.

[28] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (2012). "Improving neural networks by preventing co-adaptation of feature detectors." [Online]. Available: https://arxiv.org/abs/1207.0580

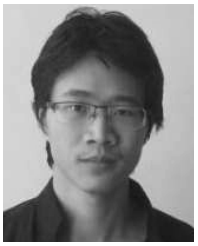
[29] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.

[30] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

Wei Li (S'11–M'13) received the B.E. degree in telecommunications engineering from Xidian University, Xi'an, China, in 2007, the M.S. degree in information science and technology from Sun Yat-sen University, Guangzhou, China, in 2009, and the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2012.

He spent one year as a Post-Doctoral Researcher at the University of California at Davis, Davis, CA, USA. He is currently with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His current research interests include statistical pattern recognition, hyperspectral image analysis, and data compression.

Dr. Li is a Reviewer for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE GEOSCIENCE REMOTE SENSING LETTERS, and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He was a recipient of the 2015 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society for his service for the IEEE JSTARS.



Guodong Wu (S'16) received the B.S. degree from the Beijing University of Chemical Technology, Beijing, China, in 2015, where he is currently pursuing the M.S. degree under the supervision of Dr. W. Li.

Fan Zhang (S'07–M'10) received the B.E. degree in communication engineering from the Civil Aviation University of China, Tianjin, China, in 2002, the M.S. degree in signal and information processing from Beihang University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from Institute of Electronics, Chinese Academy of Science, Beijing, China, in 2008.

He is currently an Associate Professor of electronic and information engineering at the Beijing University of Chemical Technology, Beijing, China. His research interests are synthetic aperture radar signal processing, high performance computing and scientific visualization.

Dr. Zhang has been a Reviewer for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and the *International Journal of Antennas and Propagation*.

Qian Du (S'98–M'00–SM'05) received the Ph.D. degree in electrical engineering from the University of Maryland at Baltimore County, Baltimore, MD, USA, in 2000.

Currently, she is the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du is a Fellow of the SPIE—International Society for Optics and Photonics. She was a recipient of the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. She was a Co-Chair for the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society from 2009 to 2013, and the Chair for Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She served as an Associate Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the *Journal of Applied Remote Sensing*, and the IEEE SIGNAL PROCESSING LETTERS. Since 2016, she has been the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She is the General Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing in Shanghai, China, in 2012.