A Novel Cubic Convolutional Neural Network for Hyperspectral Image Classification

Jinwei Wang, Xiangbo Song, Le Sun, *Member, IEEE*, Wei Huang, and Jin Wang

***Abstract*—Recently, the hyperspectral image (HSI) classification methods based on convolutional neural networks (CNN) have developed rapidly with the advance of deep learning (DL) techniques. In order to more efficiently extract spatial and spectral features, we propose an end-to-end cubic CNN (Cubic-CNN) in this article. The proposed Cubic-CNN is a supervised DL framework that significantly improves classification accuracy and shortens training time. Specifically, Cubic-CNN employs the dimension reduction method combined with principal component analysis and 1-D convolution to remove redundant information from HSIs. Then, convolutions are performed on the planes in different directions of the feature cube data to fully extract spatial and spatial–spectral features and fuse the features from different dimensions. In addition, we performed batch normalization on the data cube after each convolutional layer to improve the performance of the network. Extensive experiments and analysis on standard datasets show that the proposed algorithm can outperform the existing state-of-the-art DL-based methods.**

***Index Terms*—Cubic convolutional neural network (Cubic-CNN), dimensionality reduction, hyperspectral image (HSI) classification, spatial–spectral features.**

I. Introduction

With the rapid development of remote sensor technology, hyperspectral image (HSI) acquisition has become easier, and HSI analysis is now one of the most promising technologies for many practical applications [1]–[3]. HSIs are obtained by high-altitude hyperspectral imaging spectrometers that collect spectral data of different bands reflected by various substances on the ground [4], [5]. In contrast to traditional remote sensing technology, HSIs have hundreds of bands and so each pixel can contain more spectral information. At the same time, the continuous improvement of HSI spaces and spectral resolution promotes the development of hyperspectral applications [6].

The classification of HSIs has been one of the hot topics in the field of signal processing [7]–[11]. In the last decades, many methods based on traditional machine learning [14], [15] have been proposed for hyperspectral classification. For instance, the k-nearest neighbor method [16] uses Euclidean distance to calculate the similarity between training samples and test samples to classify HSIs. Camps-Valls *et al.* [17] proposed the hybrid kernel support vector machine (SVM) method to transform spatial and spectral information into different kernel spaces to fuse their features. The Markov random field method [18] can be used in HSI classification problems through correlation modeling of adjacent pixels in the spatial domain of the image. Sparse representation-based classification [19] using dictionary learning to construct sparse vectors for target pixels has also proved effective. The graph-based semisupervised HSI classification method is another research hotspot. Camps-Valls *et al.* [20] proposed a graph-based method to establish a graph relationship between labeled and unlabeled samples. Daniel *et al.* [21] proposed an improved semisupervised classification algorithm based on neighborhood graphs to solve the problem that the original algorithm could not measure nonlinear relationships between samples when Euclidean distance was used in the measurement process.

However, these methods only excavate the shallow features of HSIs and do not fully capture the deep information. Deep learning (DL) is widely used to learn image features in deep layers and improve HSI classification accuracy [22]. The structure of DL is generally realized through additional layers. Due to its hierarchical and distributed ability to represent features [23], [24], DL can strongly extract global features that represent context information. Therefore, due to its strong learning ability and feature expression, DL has great potential for image classification and target detection [25]. In [26], Chen *et al.* proposed an autoencoder network (SAE) to automatically learn the deep features of HSIs and then use the acquired features for classification. The autoencoder has many parameters and so it requires many labeled training samples. The autoencoder results are unsatisfactory when there are too few labeled samples. As a mainstream DL framework, the convolutional neural network (CNN) has proved to be effective in learning the abstract features of HSIs through a series of hidden layers. Hu *et al.* [27] used

Manuscript received April 18, 2020; revised June 19, 2020; accepted July 9, 2020. Date of publication July 16, 2020; date of current version July 30, 2020. This work was supported in part by the Natural Science Foundation of China under Grant 61971233, Grant 61672291, Grant 61972206, and Grant 61672293, in part by the Engineering Research Center of Digital Forensics, in part by the Ministry of Education, and in part by PAPD fund. *(Corresponding author: Le Sun.)*

Jinwei Wang and Xiangbo Song are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: wjwei_2004@163.com; sunlecncom@163.com).

Le Sun is with the School of Computer and Software, the Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, the Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: sunlecncom@163.com).

Wei Huang is with the School of Computer and Communication Engineering, the Zhengzhou University of Light Industry, Zhengzhou 450002, China (e-mail: hnhw235@zzuli.edu.cn).

Jin Wang is with the School of Computer and Communication Engineering, the Changsha University of Science and Technology, Changsha 410114, China (e-mail: jinwang@csust.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3008949

classical CNN to complete HSI classification by extracting spectral features. Chen *et al.* [28] constructed a framework that includes a principal component analysis (PCA) logistic regression. Alipourfard *et al.* [29] proposed a framework that combines the CNN structure and subspace reduction methods, extracts features by training samples, and designs an optimized network. Yue *et al.* [30] proposed a feature map generation algorithm to generate spectral and spatial feature maps and then train the classifier to obtain useful high-level features. A framework [31] proposed by Chen *et al.* uses basic CNN operations to make multiple combinations to find the best classification model. Xu *et al.* [32] proposed a band-grouping-based long short-term memory (LSTM) model and a multiscale CNN for extracting spectral and spatial features. Gong *et al*. [33] proposed a multiscale convolution and diversified metric CNN that can obtain discriminative features for HSI classification. The method proposed by Zhang *et al.* [34] uses various region-based inputs to learn contextual interaction features and inputs the joint features into the full connection layers. Mei *et al.* [35] constructed a network to extract spatial and spectral fusion features for classification in both supervised and unsupervised modes. Recently, Zhong *et al.* [36] used spatial and spectral residual blocks to capture rich spectral information and spatial contexts in HSIs.

Even though these methods provide good results for HSI classification, it is still a big challenge to obtain more accurate spatial and spectral information. To overcome the challenge, two aspects may be considered: dimension reduction and deep feature extraction. First, dimensionality reduction can significantly reduce the training time of the network. In addition, effective dimensionality reduction methods can improve the accuracy of the final classification. Second, as the network continues to deepen, multiscale and multilayer networks may not achieve the best performance [37], [38], so it is necessary to design a network with limited branches and layers that extracts more effective spatial and spectral information [39]–[41].

To achieve the goal, in this article, we propose a supervised cubic CNN (Cubic-CNN) that takes into account the data redundancy and spatial–spectral integration of HSIs. The architecture of our network is divided into two main parts: data dimension reduction and cubic convolution. First, PCA combined with 1D convolution is used for data dimension reduction. A comparison analysis reveals that these two techniques have a complementary relationship to achieve better performance in combination. Second, cubic convolution is performed on the cube data after dimension reduction. Unlike the 3D convolution, cubic convolution operates on cube data from the nonparallel planes. We employ this convolution to effectively extract spatial context information and spectral–spatial information.

In summary, this article aims to discuss the generalization of the proposed Cubic-CNN under the conditions of irregular ground object distribution. The main contributions of this article are described as follows.

1) A joint global and local-dimension reduction strategy is proposed for extracting more accurate spatial–spectral information. Specifically, PCA extracts the global spectral features and 1D convolution extracts the local spectral features. The strategy of combining the two techniques effectively retains features with stronger characterization ability. Meanwhile, it can save the computational cost of subsequent processing.
2) The cubic convolution proposed in this article extracts deep spatial and spectral information. Three sets of feature maps are generated by convolution from each side of the data cube, respectively. By characterizing the features in three different sides of the cube, it is more flexible for each branch of the network to update the parameters.
3) Cubic convolution has a smaller convolution kernel than the well-known 3D-CNN. Moreover, with the adoption of the dimension reduction strategy, the training speed of Cubic-CNN is faster than most 3D-CNN architectures.
4) The proposed Cubic-CNN has been proved to deliver good experimental results on four common standard datasets.

The rest of the article is divided into the following sections: Section II introduces the CNN framework and describes it in detail, Section III describes our experiments and analysis using four well-known HSI datasets, and Section IV presents our conclusions about Cubic-CNN.

II. Proposed Method

CNN is a kind of neural network with a special structure inspired by biological research. Neurons in the visual cortex respond to stimuli from neurons in a small area called the receptive field which exists due to the strong local correlation in the image. Inspired by this structure, we propose the Cubic-CNN for HSI classification. The architecture of Cubic-CNN consists of two parts: dimension reduction and cubic convolution.

The flow chart of HSI classification based on Cubic-CNN is illustrated in Fig. 1. Suppose there are N labeled pixels $\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{1\times1\times b}$ contained in the HSI dataset X and $Y = \{y_1, y_2, \ldots, y_N\} \in \mathbb{R}^{1\times1\times C}$ is a set of one-hot vectors representing the labels of corresponding pixels, where b and C represent the number of HSI channels and pixel categories, respectively. In this network, all tagged data are divided into three groups: the training group, the validation group, and the testing group. The pixel-centric data cube in X form a new data group $G = \{g_1, g_2, \ldots, g_N\} \in \mathbb{R}^{w\times w\times b}$, where w is the spatial size of the patch centered at the target pixel. We randomly divided G into a training set G^1, a validation set G^2, and a test set G^3. Their corresponding label sets are Y^1, Y^2, and Y^3. For example, the size of the cube data input from a pixel in the image of the Indian Pines is $9 \times 9 \times 200$. Therefore, the Cubic-CNN continuously updates the parameters throughout the training process until an accurate prediction $\hat{Y}^2$ is obtained compared to Y^2 on G^2 [42], [43].

DL can capture the details of real data and achieve the best balance between discernibility and robustness. In order to fully demonstrate the advantages of the CNN, we adopt a 3D convolutional layer for feature capture, and a batch normalization (BN) layer is added after the convolutional layer [44]. BN normalizes the data after convolution, eliminating the effect of zoom in and zoom out caused by w, and solving the problem of gradient disappearance and explosion. As shown in Fig. 2, suppose the

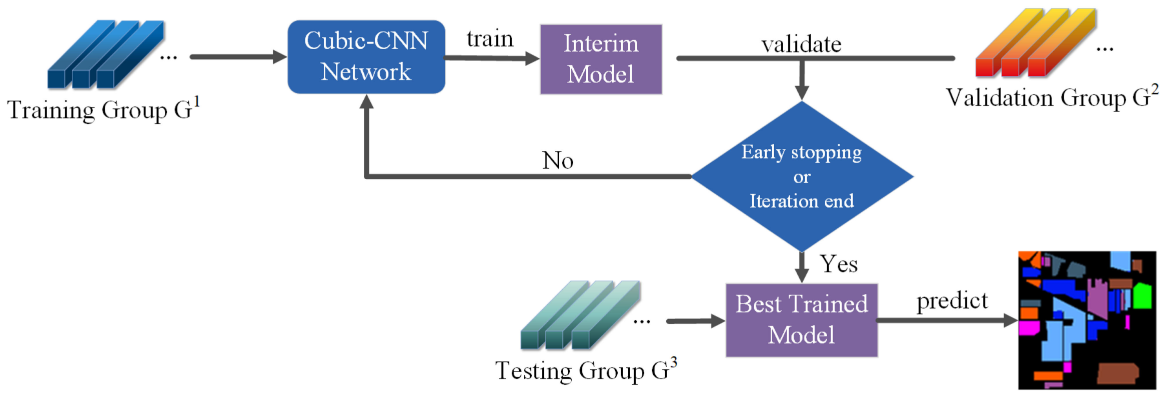


Fig. 1. Flowchart of supervised HSI classification based on Cubic-CNN.

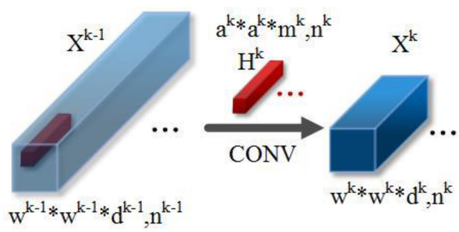


Fig. 2. 3-D convolution.

size of the input n^{k-1} feature cubes of the kth 3D convolutional layer is $w^{k-1} \times w^{k-1} \times d^{k-1}$, the kth convolutional layer has n^k 3D convolutional kernels of size $a^k \times a^k \times m^k$ and the stride is (s_1^k, s_1^k, s_2^k), then the output of the kth convolutional layer is n^k feature cubes of size $w^k \times w^k \times b^k$, where the feature map width is $w^k = \lfloor 1 + (w^{k-1} - a^k)/s_1^k \rfloor$ and the feature map depth is $d^k = \lfloor 1 + (d^{k-1} - m^k)/s_2^k \rfloor$. Therefore, the ith output of the kth 3D convolutional layer can be formulated as

$$X_i^k = \sum_{j=1}^{n^k} X_j^{k-1} * W_i^k + b_i^k \tag{1}$$

where $*$ is the 3D convolutional operation, W_i^k and b_i^k denote the weight and bias of ith convolutional kernel in n^k kernels, respectively. The output of BN and the activation function can be formulated as follows:

$$\hat{X}^k = f\left(\frac{X^k - E\left(X^k\right)}{\mathrm{Var}\left(X^k\right)}\right) \tag{2}$$

where $E(\cdot)$ and $\mathrm{Var}(\cdot)$ denote the functions of expectation and variance, respectively. $f(\cdot)$ denotes the nonlinear activation function. In the network, we use two activation functions, ReLU and Softmax, which are expressed by (3) and (4), respectively

$$f(x) = \max(0, x) \tag{3}$$

$$y_j = \frac{e^{x_i}}{\sum_{i=1}^{D} e^{x_i}}. \tag{4}$$

TABLE I
NUMBER OF TRAINING SAMPLES AND TESTING SAMPLES FOR INDIAN PINES

#	Class	Training	Testing
1	Corn-notill	200	1228
2	Corn-mintill	200	630
3	Grass-pasture	200	283
4	Hay-windrowed	200	278
5	Soybean-notill	200	772
6	Soybean-mintill	200	2255
7	Soybean-clean	200	393
8	Woods	200	1065
-	Total	1600	6904

In (4), the input has a total of D dimensions and x_i is the ith dimension of the input. The output has C dimensions, and y_j is the jth output, where $1 \leq j \leq C$.

A. Dimension Reduction

It is well known that HSIs have many bands and have strong correlations between channels. This causes channel-to-channel information redundancy in the spectral dimension. Therefore, the dimension reduction of HSIs has become a key issue that cannot be ignored. Effective dimension reduction not only removes noise in the dataset but also reduces the time and space complexity of network training and saves the cost of extracting features [45]. Therefore, we combine PCA and 1D convolution to reduce the dimensionality of HSIs. The dimension reduction strategy is shown in Fig. 3.

PCA is the most commonly used multivariate statistical technique. It has two main functions: dimension reduction and feature extraction [46]. PCA uses a mathematical orthogonal transformation to analyze data. Linearly related variables in the data are converted into unrelated variables. The components are ordered according to the variance of all variables, from the highest to the lowest. Therefore, it can be regarded as a method to extract the features from the global channels. Meanwhile, in the network, we also use 3D convolutional kernels with the size of $1 \times 1 \times d$ to reduce the dimension from the local channels. As there is no calculation between adjacent pixels, the essence

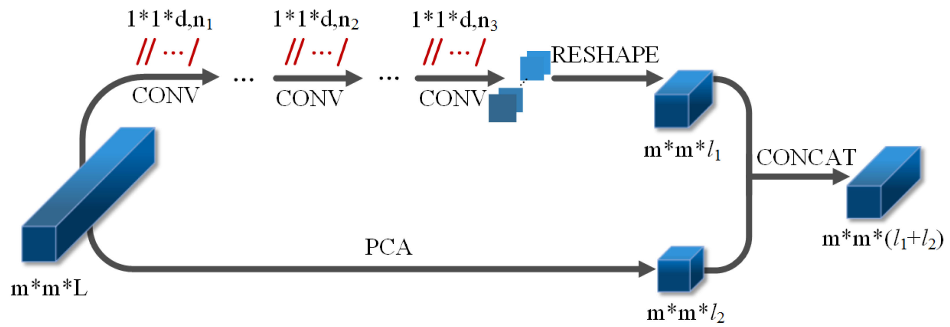


Fig. 3. Dimension reduction combined with 1D convolution and PCA.

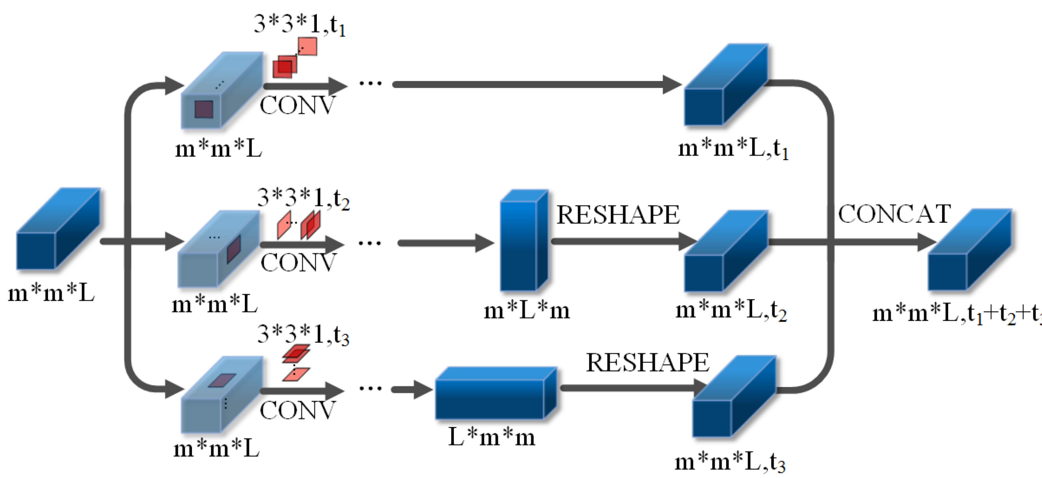


Fig. 4. Illustration of cubic convolution that convolves the data cube in three unique directions.

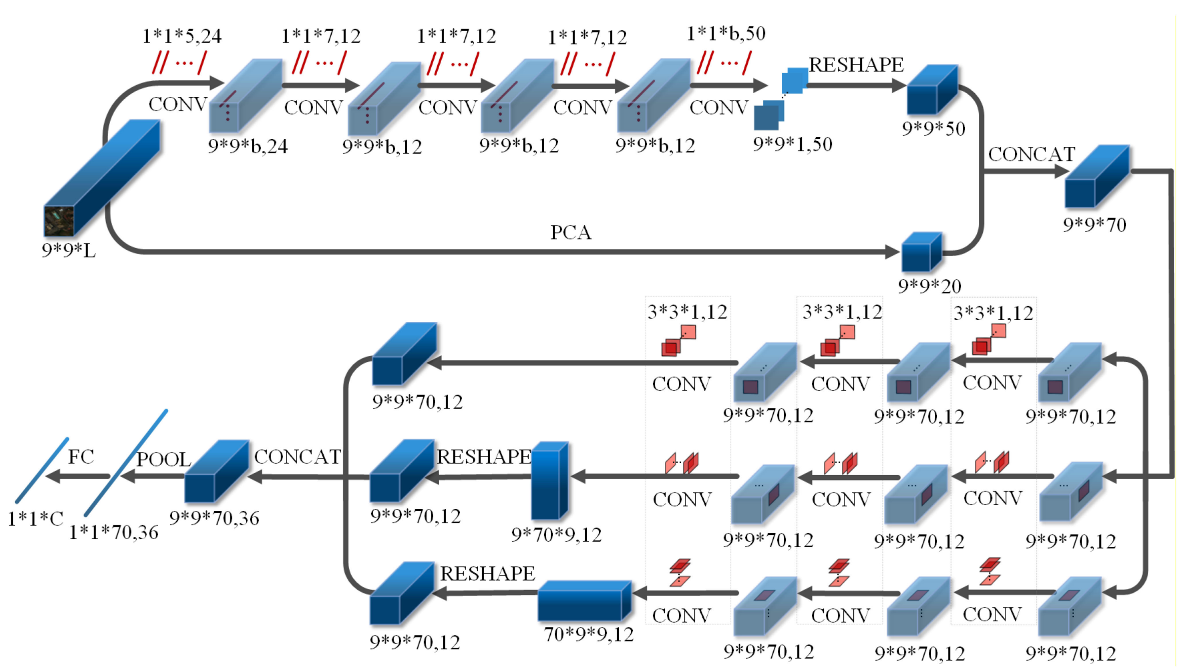
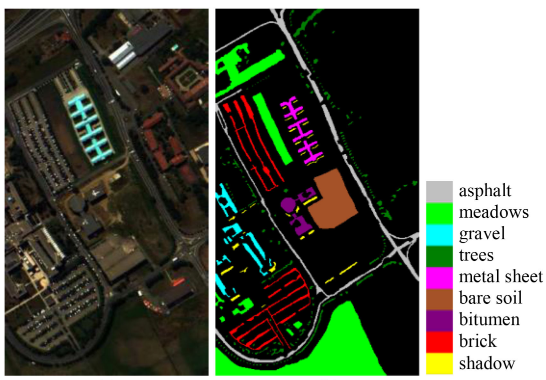


Fig. 5. Detailed structural diagram of Cubic-CNN.

(a) (b)

Fig. 6. (a) Pseudocolor image of Indian Pines dataset. (b) Ground-truth classification map of Indian Pines dataset.



(a) (b)

Fig. 7. (a) Pseudocolor image of the University of Pavia dataset. (b) Ground-truth classification map of the University of Pavia dataset.

TABLE II
NUMBER OF TRAINING SAMPLES AND TESTING SAMPLES FOR UNIVERSITY OF PAVIA

#	Class	Training	Testing
1	Asphalt	200	6431
2	Meadows	200	18449
3	Gravel	200	1899
4	Trees	200	2864
5	Sheets	200	1145
6	Baresoil	200	4829
7	Bitumen	200	1130
8	Bricks	200	3482
9	Shadows	200	747
-	Total	1800	40976

of this operation is 1D convolution in the spectral domain. Considering that the neural network has the mechanisms of gradient descent and backpropagation, using 1D convolution to reduce the dimension can fully explore the correlation information between channels in HSIs. Both PCA and 1D convolution can be used for dimension reduction, and they are complementary in function. The principal features extracted from PCA are decorrelated, which means that the data after PCA does not have the characteristics of channel relevance. In contrast, the features extracted by 1D convolution using linear summation have channel correlation.

(a) (b)

Fig. 8. (a) Pseudocolor image of Salinas dataset. (b) Ground-truth classification map of the Salinas dataset.

B. Cubic Convolution

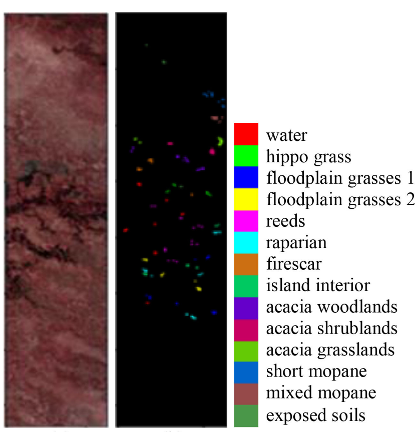
Since the spatial and spectral data of HSIs contain a lot of information, using them simultaneously can improve classification results. In view of this, we chose to perform the convolutional operation on cube data from the three nonparallel planes. The size of the convolution kernel is $n \times n \times 1$. Although this is a 3D convolutional kernel, it is actually a 2D convolution because the data between the channels are not calculated together. As shown in Fig. 4, among the three branches, the calculation of the convolution in the spatial domain involves no spectral dimension, and so the first branch only extracts the characteristics of the spatial domain. The second and third branches perform convolution operations on nonparallel sides of the data cube [47]. The convolution plane is composed of the spectral dimension and one side of the spatial domain. After features are extracted from the convolution plane, locally connected features in the spatial and spectral directions can be extracted simultaneously. After convolution on each plane, the feature cubes can be spliced together by dimensional transformation as the input of the next layer [48] .

The convolution operations on three nonparallel planes are performed in order to achieve real 3D convolution. Compared with 3D convolution, cubic convolution has the following advantages.

1) Cubic convolution captures spatial features and spatial–spectral features separately, while 3D convolution mixes all spatial-spectral features.
2) After 3D convolution, a data cube is generated, but cubic convolution generates three data cubes. Different mixed spatial and spectral features are saved in each of these data cubes. Moreover, cubic convolution is more flexible for updating parameters and has a greater capability to represent information than the 3D convolution.
3) Compared with the commonly used 3D convolution kernel with a size of $3 * 3 * \mathrm{L}$ (for example, the 3D convolution kernel used by S S R N is $3 * 3 * 128$), the $3 * 3$

TABLE III
NUMBER OF TRAINING SAMPLES AND TESTING SAMPLES FOR SALINAS

#	Class	Training	Testing
1	Broccoli green weeds 1	200	1809
2	Broccoli 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 vineyard develop	200	6003
10	Corn senesced weeds	200	3078
11	Lettuce romaines, 4 wk	200	868
12	Lettuce romaines, 5 wk	200	1727
13	Lettuce romaines, 6 wk	200	716
14	Lettuce romaines, 7 wk	200	870
15	Vineyard untrained	200	7068
16	Vineyard vertical trellis	200	1607
-	Total	3200	50929



(a) (b)

Fig. 9. (a) Pseudocolor image of Botswana dataset. (b) Ground-truth classification map of the Botswana dataset.

TABLE IV
NUMBER OF TRAINING SAMPLES AND TESTING SAMPLES FOR BOTSWANA

#	Class	Training	Testing
1	Water	20	250
2	Hippo grass	20	81
3	Floodplain grasses 1	20	231
4	Floodplain grasses 2	20	195
5	Reeds	20	249
6	Raparian	20	249
7	Firescar	20	239
8	Island interior	20	183
9	Acacia woodlands	20	294
10	Acacia shrublands	20	228
11	Acacia grasslands	20	285
12	Short mopane	20	161
13	Mixed mopane	20	248
14	Exposed soils	20	75
-	Total	280	2968

TABLE V
OVERALL ACCURACY (%) FOR INPUT PATCHES WITH DIFFERENT SPATIAL SIZES FOR THREE DATA SETS

Spatial size	IN	UP	Salinas
3×3	95.21	96.69	93.8
5×5	98.64	98.89	96.06
7×7	99.11	99.38	97.88
9×9	99.40	99.65	98.93
11×11	99.44	99.74	98.97

$*$ 1 convolution kernel we used is smaller in the cubic convolution, which makes our network training faster. [49].

C. Cubic-CNN Structure

The detailed structure of the Cubic-CNN proposed is shown in Fig. 5. Cubic-CNN has a dimension reduction process, a cubic convolution process, an average pooling layer, a dropout layer, and a full connection layer. Cubic-CNN is a dual input network with original data and PCA data as input. We take an original data cube with an input size of $9 \times 9 \times b$ and a PCA data cube with an input size of $9 \times 9 \times 20$ as an example to illustrate Cubic-CNN. In the convolutional dimension reduction process, for the first convolution layer, the kernel size is $1 \times 1 \times 5$, the kernel number is 24, the stride is $(1, 1, 2)$. For the next convolution operations, the kernel size is $1 \times 1 \times 7$, the kernel number is 12, the stride is $(1, 1, 1)$. The last convolution layer with 50 kernels of $1 \times 1 \times b$ and a stride of $(1, 1, 1)$ convolves the data cubes. Then 50 feature maps of $9 \times 9 \times 1$ are generated. The size of these feature maps is changed to $9 \times 9 \times 50$ by dimensional transformation. We splice the data after PCA and the feature maps after 1D convolution into a feature map of the size of $9 \times 9 \times 70$ as the final dimension reduction result. In the cubic convolution section, for the convolution operation in the spatial domain, the kernel size is $3 \times 3 \times 1$, the kernel number is 12, and the stride is $(1, 1, 1)$. For the convolution operation on each side plane, the kernel size is $3 \times 3 \times 1$, the kernel number is 12, and the stride is $(1, 1, 1)$. Then, three sets of feature cubes are generated with the sizes $12 * 9 \times 9 \times 70$, $12 * 9 \times 9 \times 70$, and$12 * 9 \times 9 \times 70$. Feature maps of the size $36 * 9 \times 9 \times 70$ are generated by combining the data cubes after dimensional transformation. The input is converted into $36 * 1 \times 1 \times 70$ vectors by averaging the pooling layer. The vectors are stretched to one dimension. After passing through the dropout layer, the vectors are classified in the full connection layer [50]. In order to obtain a model with good performance, we use the early stopping mechanism, which stops training process when the loss of validation set continues to be constant to prevent overfitting. The details of the HSI classification framework using Cubic-CNN are shown in Algorithm 1.

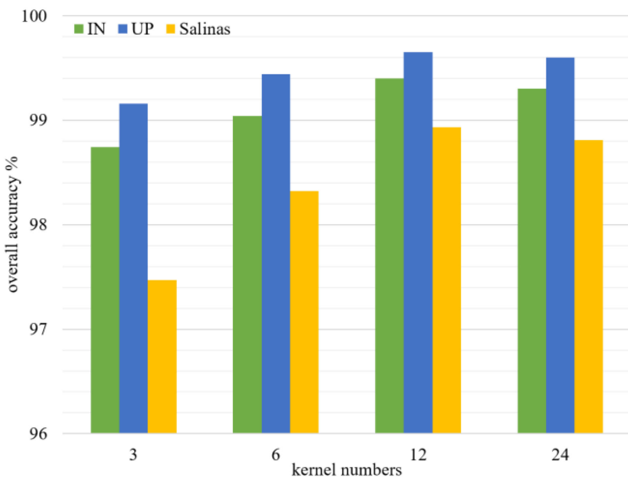


Fig. 10. Overall accuracy histogram of different cubic convolution kernels on the first three datasets.

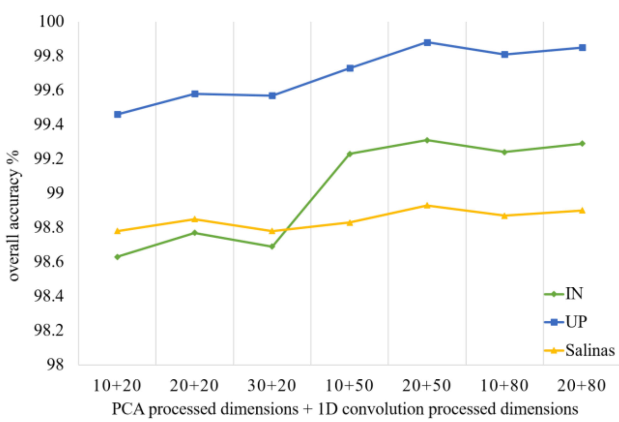


Fig. 11. Line chart of the overall accuracy of different search dimensions on the first three datasets.

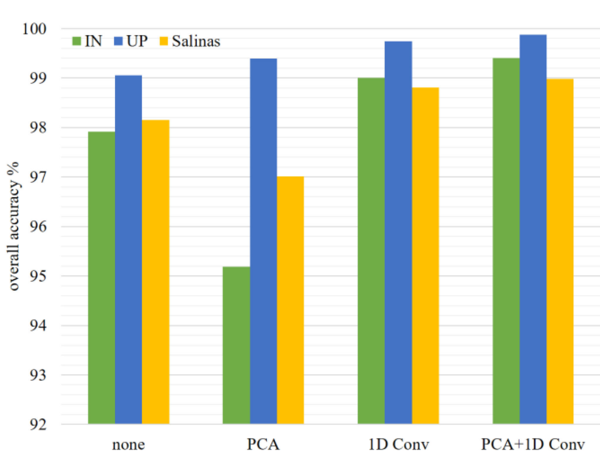


Fig. 12. Overall accuracy histogram of different dimension reduction strategies.

Algorithm 1: Framework of Cubic-CNN for HSI Classification.

Input: Input an HSI X with ground-truth Y; maximum number of iterations $\mathcal{T} = 80$; learning rate $\eta = 0.0003$; PCA dimension $l_2 = 20$;

1: Obtain the $X - \mathrm{pca}$ after PCA preprocessing.
2: Divide X and $X - pca$ into G^1, $G^2$and G^3according to the index.
3: // Train the Cubic-CNN model
4: **for** $t = 1$ to $\mathcal{T}$ **do**
5: Perform 1D convolution to reduce the dimension of HSI, then splice it with the PCA processed data
6: Perform cubic convolution on feature maps.
7: Update parameters with RMSProp.
8: If satisfied the early stopping (the loss of G^2 continues to be constant):
9: break;
10: **end for**
11: Test G^3 with the trained model.

Output:

1: $\hat{Y}^3$and the acc of each class;
2: OA, AA, Kappa.

III. Experiment and Analysis

All the experiments are conducted on a desktop with Intel Core I7-4790K, GeForce 1080Ti GPU, and 24G RAM. We adopt Python 3.5.1, TensorFlow 1.14, and Keras 2.15 to implement the program of the proposed Cubic-CNN.

A. Datasets

The proposed Cubic-CNN was evaluated on four common standard datasets: Indian Pines, University of Pavia, Salinas, and Botswana. For the first three datasets, we randomly selected 200 samples of each class as the training set and used the remaining data as the validation set and testing set. For the Botswana dataset, we randomly selected 20 samples of each class as the training set and used the remaining data as the validation set and testing set. The Indian Pines dataset was collected by the AVIRIS sensor from an experimental field in Indiana, USA, in 1992. It consists of 145×145 pixels, each pixel has a resolution of about 20 m, and there are 200 available spectral channels. Sixteen classes of ground objects were labeled in this data and eight of the subclasses were discarded using statistical methods. Fig. 6 shows a spatial pseudocolor map of Indian Pines with ground object distribution labels. Table I shows the number of training samples and testing samples in each category.

The University of Pavia dataset is an HSI collected by ROSIS sensors in Pavia, Italy, in 2003. Its ground objects can be divided into nine categories marking 42 776 samples. There are 610×340 pixels in this dataset, each pixel has a resolution of about 1.3 m, and there are 103 spectral channels. Fig. 7 shows a spatial pseudocolor map of the University of

TABLE VI
CLASSIFICATION RESULTS WITH CUBIC-CNN AND COMPARED METHODS ON INDIAN PINES DATASET

	SVM	SSUN	DR-CNN	DPP-MS-CNN	SSRN	SPA-SPA	SPA-SPE	Cubic-CNN
1	77.97	81.13	98.20	99.03	99.44	97.39	99.35	99.08±0.40
2	72.19	75.23	99.79	99.74	96.16	99.45	99.29	99.46±0.21
3	87.38	83.59	100	100	99.60	99.59	98.08	**100±0.00**
4	98.23	99.64	100	100	100	99.62	100	**100±0.00**
5	62.97	80.12	99.78	99.61	97.28	99.25	98.49	97.29±0.33
6	98.46	90.27	96.69	97.80	99.79	99.60	99.29	**99.89±0.01**
7	52.79	74.44	99.86	100	99.71	99.02	100	99.14±0.25
8	98.33	99.33	99.99	100	99.47	99.89	99.08	99.89±0.05
OA	79.97	86.07	98.54	99.08	99.04	99.13	99.18	**99.40±0.15**
AA	79.91	85.47	99.29	99.52	98.93	99.23	99.20	99.34±0.19
Kappa	75.88	83.05	98.75	98.83	98.83	98.92	99.00	**99.27±0.18**

The bold numbers indicate the highest accuracies of the proposed algorithm among all comparison algorithms.

TABLE VII
CLASSIFICATION RESULTS WITH CUBIC-CNN AND COMPARED METHODS ON UNIVERSITY OF PAVIA DATASET

	SVM	SSUN	DR-CNN	DPP-MS-CNN	SSRN	SPA-SPA	SPA-SPE	Cubic-CNN
1	94.62	94.95	98.43	99.38	100	99.78	99.75	99.97±0.02
2	98.25	97.62	99.45	99.59	99.94	99.94	99.96	**99.98±0.01**
3	62.89	67.19	99.14	97.33	98.38	99.56	100	98.88±0.05
4	69.38	79.89	99.50	99.31	100	99.22	99.43	99.86±0.10
5	94.85	96.19	100	100	100	100	100	99.91±0.52
6	70.48	79.91	100	99.99	97.45	97.00	98.60	99.96±0.
7	42.07	69.52	99.70	99.85	99.27	98.57	98.05	**100±0.00**
8	89.68	84.56	99.55	99.02	97.69	98.85	98.71	**99.73±0.14**
9	97.52	99.07	100	100	100	99.87	99.46	99.04±0.26
OA	84.19	89.54	99.56	99.46	99.37	99.36	99.56	**99.88±0.04**
AA	79.97	85.43	99.53	99.39	99.19	99.19	99.32	**99.70±0.07**
Kappa	79.54	86.16	99.49	99.27	99.15	99.14	99.41	**99.53±0.14**

The bold numbers indicate the highest accuracies of the proposed algorithm among all comparison algorithms.

Pavia with ground object distribution labels. Table II shows the number of training samples and testing samples in each category.

The Salinas dataset was also captured by the AVIRIS imaging spectrometer in the Salinas Valley in California, USA. It has a spatial resolution of 3.7 m. The original image has 224 spectral channels, and we used the 204 spectral channels remained after we eliminated the channels that could not be reflected by water. The size of the image is 512×217 and there are 16 available types with a total of 54 129 pixels. Fig. 8 shows a spatial pseudocolor map of Salinas with ground object distribution labels. Table III shows the number of training samples and testing samples in each category.

The Botswana dataset was derived from the NASA EO-1 satellite obtained at Botswana in 2001. After removing the noise band, there are 145 bands remained. The size of the image is 1476×256 and there are 14 available types with a total of 3298 pixels. Fig. 9 shows a spatial pseudocolor map of Botswana with ground object distribution labels. Table IV shows the number of training samples and testing samples in each category.

B. Framework Settings

In this article, the three indicators of overall accuracies (OA), average accuracies (AA), and Kappa are utilized to evaluate the performance of the algorithm. General speaking, the larger the values of the three indicators, the better the classification performance. Let $M \in \mathbb{R}^{C\times C}$ represent the confusion matrix of classification results, where *C* is the number of ground object categories, $M(i, j)$ denotes the number of $i\text{th}$ category samples that have been classified to $j\text{th}$ category. OA, AA, and Kappa can be calculated as follows:

$$\text{OA} = \text{sum}(\text{diag}(M)/\text{sum}(M)) \tag{5}$$

$$\text{AA} = \text{mean}(\text{diag}(M)./\text{sum}(M, 2)) \tag{6}$$

$$\text{Kappa} = \frac{\text{OA} - (\text{sum}(M, 1)\text{sum}(M, 2))/\text{sum}(M)^2}{1 - (\text{sum}(M, 1)\text{sum}(M, 2))/\text{sum}(M)^2} \tag{7}$$

where $\text{diag}(\cdot) \in \mathbb{R}^{C\times 1}$ denotes the vector of diagonal elements of the matrix, $\text{sum}(\cdot)$denotes the sum of all elements, $\text{sum}(\cdot, 1) \in \mathbb{R}^{1\times C}$ denotes the sum of the elements in each

TABLE VIII
CLASSIFICATION RESULTS WITH CUBIC-CNN AND COMPARED METHODS ON SALINAS

	SVM	SSUN	DR-CNN	DPP-MS-CNN	SSRN	SPA-SPA	SPA-SPE	Cubic-CNN
1	96.48	99.83	100	100	100	100	100	**100±0.00**
2	99.91	99.57	100	100	100	100	100	**100±0.00**
3	98.05	96.97	99.98	100	100	100	99.94	99.95±0.02
4	98.59	99.33	99.89	99.25	100	99.67	98.80	99.26±0.13
5	99.59	98.75	99.83	99.44	99.96	99.53	99.71	**99.96±0.01**
6	99.79	99.89	100	100	100	100	100	**100±0.00**
7	99.38	97.85	99.96	99.87	100	94.30	100	**100±0.00**
8	84.64	74.89	94.14	95.36	96.59	100	98.47	99.19±0.10
9	99.53	99.48	99.99	100	99.93	99.10	99.98	99.92±0.01
10	87.24	94.66	99.20	98.85	99.76	99.08	99.33	98.42±0.25
11	96.29	96.32	99.99	99.77	97.21	99.53	100	99.17±0.14
12	99.08	98.74	100	100	100	100	100	**100±0.00**
13	98.20	99.02	100	99.86	100	100	100	**100±0.00**
14	96.49	94.00	100	99.77	100	99.33	96.86	99.78±0.11
15	53.07	77.13	95.52	90.50	96.57	97.17	95.40	94.81±0.34
16	95.25	97.43	99.72	98.94	100	99.82	100	**100±0.00**
OA	86.06	90.31	98.33	97.51	98.71	98.40	98.82	**98.93±0.27**
AA	93.85	95.24	99.26	98.85	99.37	99.21	99.28	**99.40±0.14**
Kappa	84.53	89.11	98.29	97.88	98.56	98.21	98.68	**98.80±0.16**

The bold numbers indicate the highest accuracies of the proposed algorithm among all comparison algorithms.

TABLE IX
CLASSIFICATION RESULTS WITH CUBIC-CNN AND COMPARED METHODS ON BOTSWANA DATASET

	SVM	SSUN	SSRN	SPA-SPA	SPA-SPE	Cubic-CNN
1	99.21	99.20	99.87	98.93	99.22	99.67±0.31
2	91.67	94.12	99.87	100	100	**100±0.00**
3	90.91	95.36	100	100	100	**100±0.00**
4	90.43	85.99	97.59	98.70	98.61	99.31±0.53
5	78.36	85.40	97.64	98.27	98.54	98.17±1.66
6	53.82	69.62	97.41	99.14	98.94	98.93±0.67
7	99.57	97.84	100	100	100	**100±0.00**
8	80.09	91.04	99.48	100	100	**100±0.00**
9	90.68	90.76	99.85	100	100	**100±0.00**
10	79.74	89.62	99.07	99.82	99.85	**99.86±0.29**
11	96.81	92.23	100	99.74	99.93	**100±0.00**
12	83.91	89.71	99.93	100	100	**100±0.00**
13	85.15	93.19	100	100	100	**100±0.00**
14	90.54	96.00	100	100	100	**100±0.00**
OA	85.07	89.99	99.23	99.56	99.60	**99.67±0.27**
AA	86.49	90.72	99.34	99.62	99.64	**99.71±0.23**
Kappa	83.82	89.15	99.16	99.53	99.57	**99.63±0.29**

The bold numbers indicate the highest accuracies of the proposed algorithm among all comparison algorithms.

column, $\mathrm{sum}(\cdot, 2) \in \mathbb{R}^{C\times 1}$denotes the sum of the elements in each row, $\mathrm{mean}(\cdot)$ denotes the mean of all elements, and $./(\cdot)$ denotes the division of element.

Cubic-CNN updates the network parameters with backpropagation. The chain rule is employed to introduce intermediate variables to iteratively calculate the gradient for each layer, which accelerates the calculation of the gradient. We analyzed several factors that affect the training process and training results: learning rate, patch size, number of convolution layer filters, and dimension of the data cube after dimension reduction.

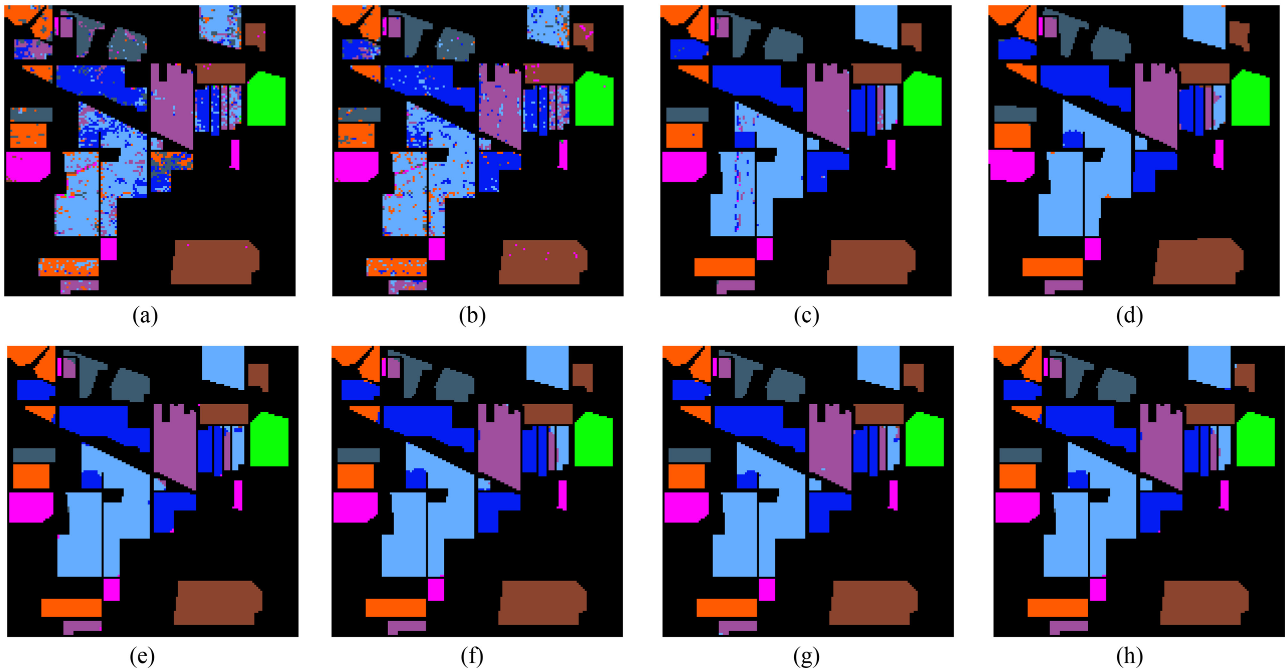


Fig. 13. Classification maps of Indian pines obtained by (a) SVM, (b) SSUN, (c) DR-CNN, (d) DPP-MS-CNN, (e) SSRN, (f) SPA-SPA, (g) SPA-SPE, and (h) Cubic-CNN.

TABLE X
COMPARISON OF TRAINING AND TESTING TIME BETWEEN THE CONTRAST MODELS AND PROPOSED MODEL ON THE FIRST THREE DATASETS

		IN	UP	Salinas
DR-CNN	Train.(m)	30	36	60
	Test.(s)	34	105	240
SSRN	Train.(m)	10	9.7	20.75
	Test.(s)	2.74	12.43	22.11
SPA-SPA	Train.(m)	3.52	2.93	6.87
	Test.(s)	3.18	13.58	24.89
SPA-SPE	Train.(m)	4.52	4.02	8.93
	Test.(s)	3.63	15.76	28.04
Cubic-CNN	Train.(m)	5	3.77	9.73
	Test.(s)	3.94	18.15	30.18

Since each patch is small, we chose 32 as the batch size. The RMSProp and category cross-entropy function were used as the optimizer and loss function of the network, respectively. For each training process, we used 80 epochs to generate the final model. Throughout the training process, we preserved the models with the best classification performance and analyzed their results [51].

A proper learning rate determines when the objective function converges to a local minimum in an appropriate time. {0.01, 0.003, 0.001, 0.0003, and 0.0001} is selected as a set of alternative learning rates through a grid search, then we trained and tested each dataset based on these learning rates. According to the classification results, 0.0003 is the best learning rate for the first three datasets.

The number of filters in the convolutional layer determines the ability of the network to characterize features. An improper number of convolution kernels have a bad effect on classification accuracy and training time. Therefore, Cubic-CNN uses the same number of convolution kernels in dimension reduction and cubic convolution. We experimented with different numbers of kernels ranging from 3 to 24 in each convolutional layer to find a general framework. As shown in Fig. 10, when the network has 12 convolution kernels, the classification result is the best. And when the network has 24 convolution kernels, it exhibits equivalent or slightly worse performance.

Appropriate dimensions can reduce data size as much as possible and make enough information retained at the same time. The dimension reduction strategy combined the PCA and 1D convolution is used to conduct a grid search for the optimal dimension in the network. We selected 10, 20, and 30 as the PCA's search dimensions, and chose 20, 50, and 80 as the channel numbers to be searched for 1D convolution. As shown in Fig. 11, we used {(10, 20), (20, 20), (30, 20), (10, 50) (20, 50), (10, 80), and (20, 80)} as the search list. It can be seen that the accuracy converged when the dimension reached (20, 50) through several test comparisons based on the dimension reduction strategy.

In order to verify the effectiveness of dimension reduction, we carried out ablation experiments. Fig. 12 indicates that the proposed dimension reduction strategy (PCA+1D Conv) has higher overall accuracy compared with PCA-only (PCA), 1D convolution-only (1D Conv), and no dimension reduction (none) strategies.

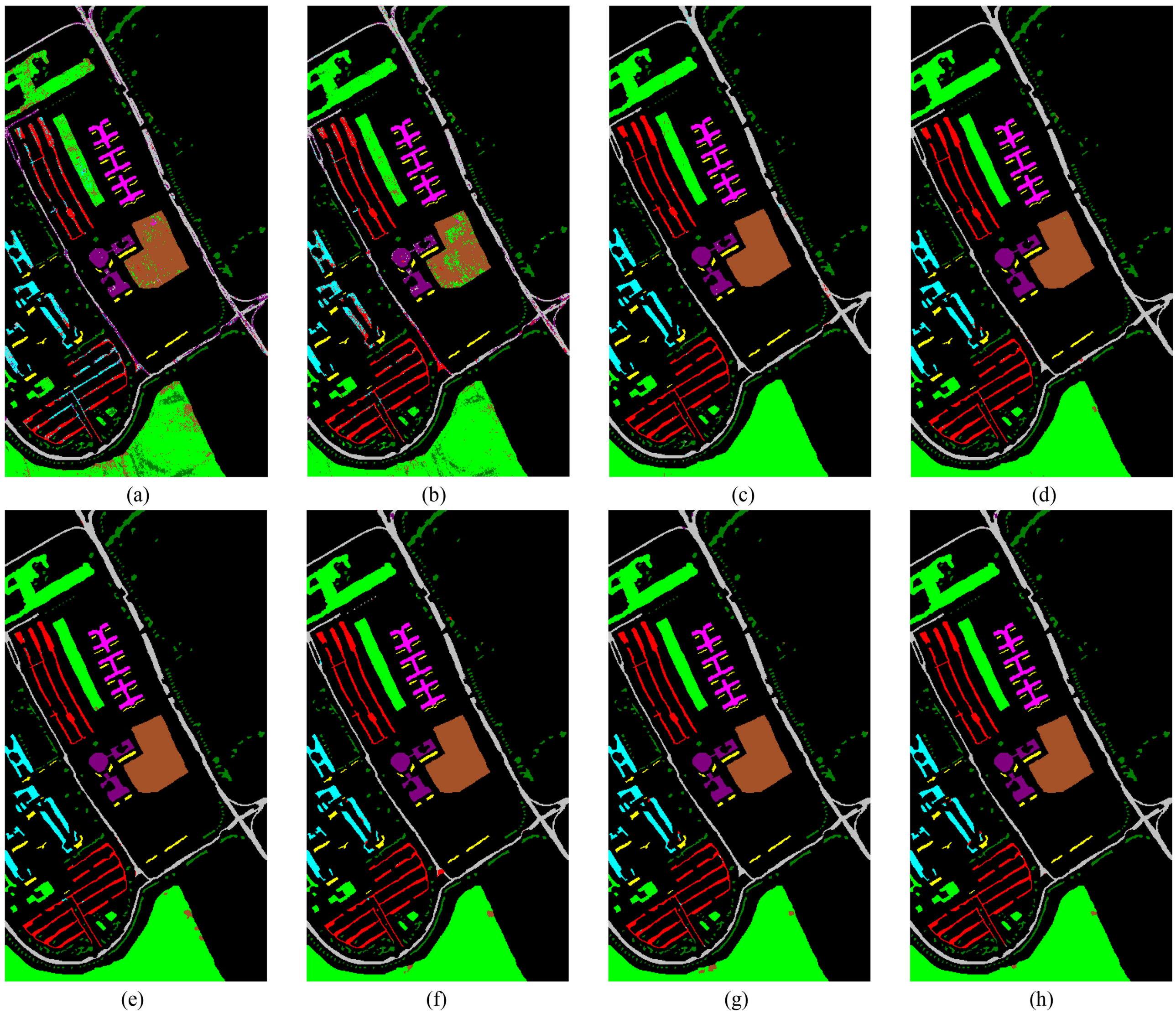


Fig. 14. Classification maps of the University of Pavia obtained by (a) SVM, (b) SSUN, (c) DR-CNN, (d) DPP-MS-CNN, (e) SSRN, (f) SPA-SPA, (g) SPA-SPE, and (h) Cubic-CNN.

Since ground objects tend to be spatially continuous, adjacent pixels are likely to belong to the same category. The classification accuracy can be enhanced by using spatial information effectively. We use patches of different spatial sizes to experiment for Cubic-CNN to determine a proper data cube size. In Table V, classification accuracy increases with the increased patch size. When the spatial size of input data reaches a certain value, our proposed model has strong robustness. So we verify the stability of our proposed model compared with other methods by using a 9×9 size patch.

C. Classification Performance

This section compares the performance of SVM [13], SSUN [32], DR-CNN [34], DPP-MS-CNN [33], SSRN [36], and our proposed methods on four datasets. SVM is a classic machine learning method. SSUN is a DL method based on LSTM. DR-CNN performs 2D convolution from different regions of HSI to extract local features and global features. DPP-MS-CNN is a multiscale framework with a determinantal point process prior to enhance the diversity of samples. And SSRN uses 3D convolutional residual blocks to extract features in the spectral domain and spatial domain successively. Due to the lack of source code, the results of DR-CNN and DPP-MS-CNN are from the corresponding references. To verify the effectiveness of the cubic convolution part of the framework, we also tested a branch network containing only the spatial domain as the convolution plane (SPA-SPA) and a branch network containing the other two nonparallel surfaces as the convolution plane (SPA-SPE). In order to ensure fairness, we chose 9×9 as the spatial size of the input patch. We randomly selected 200 cubes for each class as training samples for Indian Pines, University of Pavia, and Salinas.

Tables VI–IX show the classification accuracy for each class and the OAs, AAs, and Kappas of all algorithms for each of the four datasets. The SVM results were worse than all DL-based

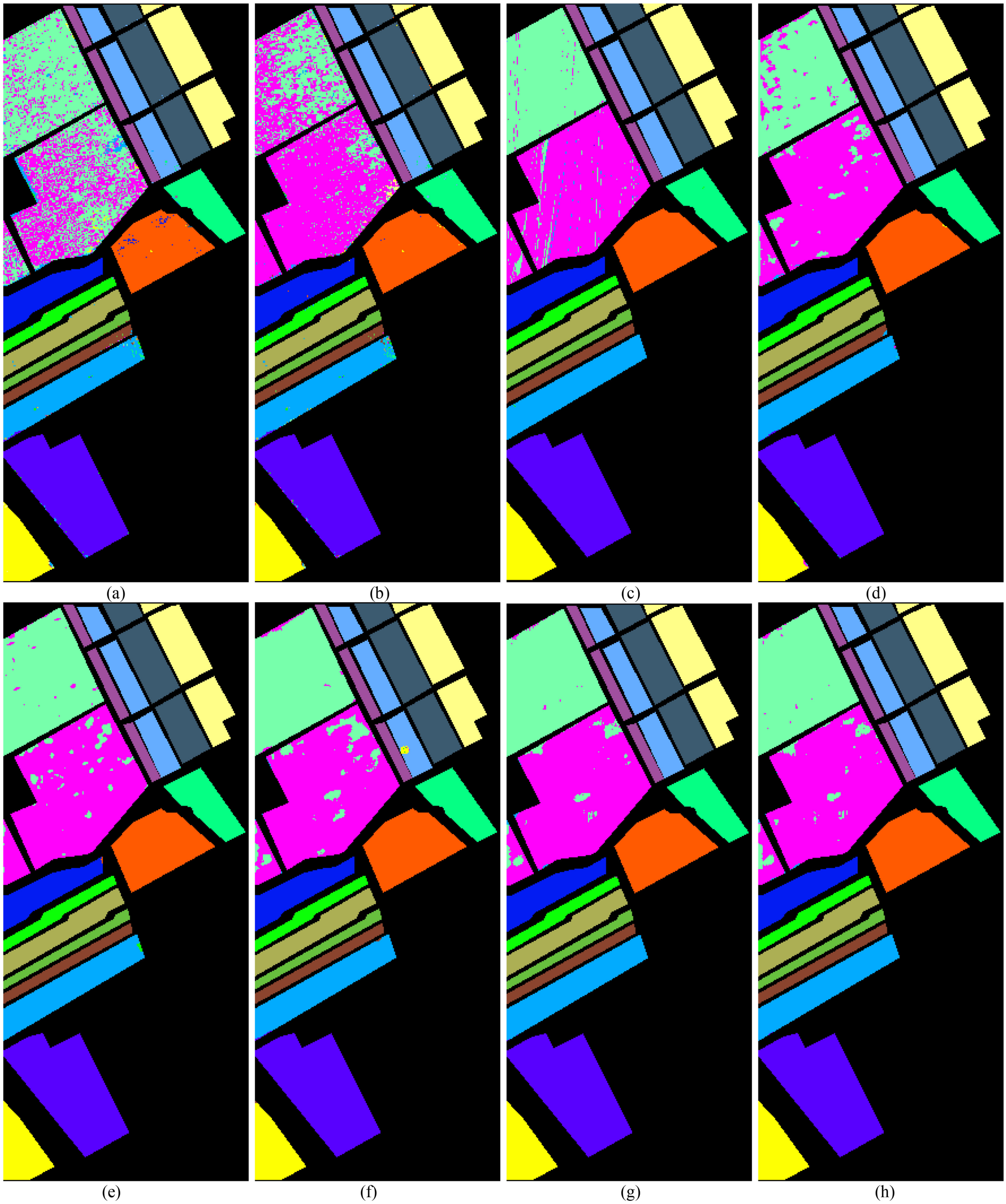


Fig. 15. Classification maps of Salinas obtained by (a) SVM, (b) SSUN, (c) DR-CNN, (d) DPP-MS-CNN, (e) SSRN, (f) SPA-SPA, (g) SPA-SPE, and (h) Cubic-CNN.

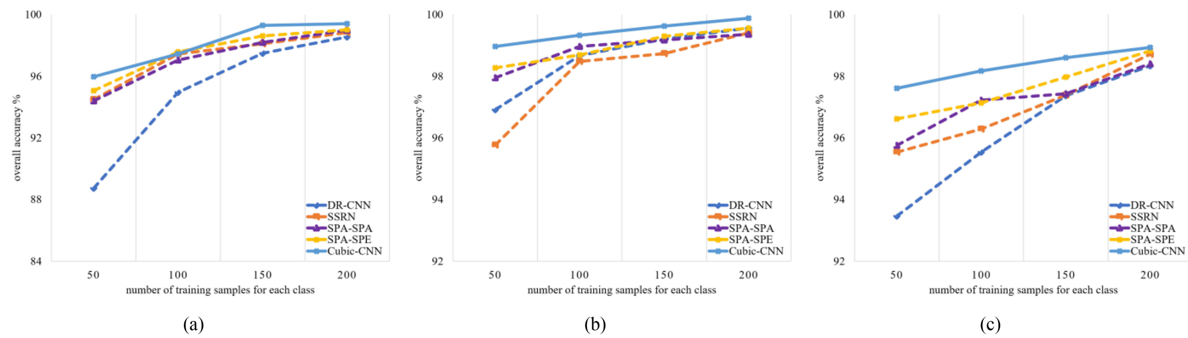


Fig. 16. (a) Line chart of the overall accuracy of different models at different sampling rates for Indian Pines. (b) Line chart of the overall accuracy of different models at different sampling rates for the University of Pavia. (c) Line chart of the overall accuracy of different models at different sampling rates for Salinas.

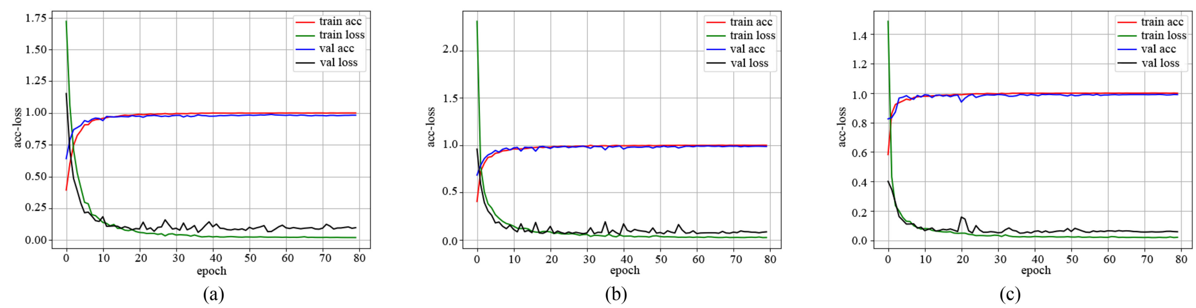


Fig. 17. (a) Accuracy and loss function curves of the training and validation sets for Indian Pines. (b) Accuracy and loss function curves of the training and validation sets for the University of Pavia. (c) Accuracy and loss function curves of the training and validation sets for Salinas.

methods. After we adjusted the sampling rate, the accuracy of SSUN decreased significantly. The results of SPA-SPA and SPA-SPE slightly surpassed the contrasting DL methods. For example, on the Indian Pines and Salinas datasets, the OAs, AAs, and Kappas of our two branch models were higher than those of the compared algorithms. The accuracy of SPA-SPE was slightly higher than that of SPA-SPA, this is because that SPA-SPE can obtain the joint characteristics of space and spectrum, which is more beneficial to classification. It also indicates that almost all indicators of Cubic-CNN surpassed the best algorithms available. For example, on the University of Pavia dataset, the OA of Cubic-CNN (99.88%) was 0.32% higher than the optimal results of SSRN (99.56%) and DR-CNN (99.56%). Furthermore, Cubic-CNN achieved the highest accuracy for some classes on the first three datasets. For example, Cubic-CNN achieved 100% accuracy for the corn class and corn-mintill class in the Indian Pines dataset, the bitumen class in the University of Pavia dataset, and the stubble class and celery class in the Salinas dataset. Table IX shows that our performance on the Botswana dataset is better than the comparison methods.

Figs. 13–15 show pseudocolor images of the classification results of the optimally trained model on the first three datasets. The experimental results of SVM and SSUN are not satisfactory and there is a lot of noise in their classification maps. The classification maps of the remaining methods on the University of Pavia dataset have almost no noise. The classification maps of DR-CNN on the Indian Pines dataset and DPP-MS-CNN on the Salinas dataset are clearly worse than the results of SSRN and our proposed models. Moreover, the performance of the classification maps of SPA-SPA and SPA-SPE on the first three datasets was extraordinary. The classification map of Cubic-CNN for the University of Pavia dataset is better than SPA-SPA and SPA-SPE. On the Salinas dataset, our algorithm significantly reduced noise in the Vineyard untrained class compared to the other algorithms. In short, Cubic-CNN's classification maps are more accurate and their texture is smoother than maps produced by the other algorithms tested.

In order to test the dependence of Cubic-CNN on the number of training samples, we used 50, 100, 150, and 200 patches of each class as the training set for experiments. We used the same sampling rate for comparison on DR-CNN, SSRN, SPA-SPA, and SPA-SPE. Fig. 16 is a line chart of OA for each method at different sampling rates on the first three datasets. Most of the curves are relatively concentrated for the Indian Pines and University of Pavia datasets. As the number of training samples decreases, the accuracy of DR-CNN decreases most significantly. The Cubic-CNN curve has clear advantages for the Salinas dataset. Compared with SSRN, the SPA-SPE curve can achieve the same or better results. From the sampling numbers of each interval, the Cubic-CNN curve is essentially at the highest point.

The time complexity of the training and testing process using DR-CNN, SSRN, SPA-SPA, SPA-SPE, and Cubic-CNN is shown in Table X. The DR-CNN training process and test time are longer due to the large amount of calculation required for various regional structures. SSRN uses a 3D convolution kernel of $3 \times 3 \times 128$ and performs 200 iterations, and so it requires more training time than the algorithm we propose. This shows that our cubic convolution can both obtain very good accuracy and reduce the time complexity during the training process. The training time of SPA-SPA is shorter than that of SPA-SPE. This is because the network structure of SPA-SPE is more complicated. Cubic-CNN can achieve excellent results with lower-time complexity. The accuracy and loss curves of the training and validation sets on the Indian Pines, University of Pavia, and Salinas datasets are shown in Fig. 17. The proposed model quickly converged for the first three datasets at the beginning of the training process. The loss curves for the first three datasets do not have huge fluctuations. The curve converged in 60 iterations. To ensure more stable accuracy, we used 80 epochs.

IV. Conclusion

This article proposed a supervised CNN of deep spatial–spectral learning for HSI classification. The proposed framework includes both dimension reduction and cubic convolution, which improves the accuracy of existing algorithms for HSI classification. The dimension reduction method employs both PCA and 1D convolution to effectively remove the information redundancy in the spectrum. Moreover, PCA and 1D convolution complement each other in preserving the characteristics of the spectrum and reducing subsequent calculation pressure. Cubic convolution obtains features in the spatial domain and spatial–spectral domain of HSIs. Compared to 3D convolution, cubic convolution makes the network more flexible and the training process faster. Cubic-CNN obtained better results in less time for the first three datasets. It is worth mentioning that cubic convolution provides a new strategy for the analysis of cube data.

By observing the classification maps, we can find that some noise remains at the edges of the areas that belong to disparate categories in the image. This is because there are many pixels of different classes in the patch centered on the pixel to be classified. Methods of limiting the input patch and optimizing the network structure still need to be explored. On the other hand, due to the advent of the big data era, DL may re-emerge with a large number of labeled samples as the training set. However, HSIs present the problem of marked pixel scarcity. Our method can achieve excellent results in 200 training samples of each ground object, which does not mean that the classification problem of HSIs has been solved. The results of existing algorithms at low sampling rates still have upside potential. Our future research will seek to achieve classification accuracy close to that of high sampling under low sampling conditions. Moreover, high-performance implementation of the proposed method will also be considered [52], [53] to accelerate the convergence speed for real-time applications.

References

[1] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, “Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 639–643, Jun. 2017.

[2] L. Sun, W. Ge, Y. Chen, J. Zhang, and B. Jeon, “Hyperspectral unmixing employing l(1)-l(2) sparsity and total variation regularization,” *Int. J. Remote Sens.*, vol. 39, no. 19, pp. 6037–6060, Oct. 2018.

[3] L. Sun, T. Zhan, Z. Wu, L. Xiao, and B. Jeon, “Hyperspectral mixed denoising via spectral difference-induced total variation and low-rank approximation,” *Remote Sens.*, vol. 10, no. 12, Dec. 2018, Art. no. 1956.

[4] A. Hosseini and H. Ghassemian, “Classification of hyperspectral and multispectral images by using fractal dimension of spectral response curve,” in *Proc. 20th Iranian Conf. Electr. Eng.*, May 2012, pp. 1452–1457.

[5] C. Fang, N. Zheng, and L. WenPeng, “A study on methods of describing the figure of reflectance curve,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2005, vol. 6, pp. 3765–3767.

[6] L. Sun, F. Wu, T. Zhan, W. Liu, J. Wang, and B. Jeon, “Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, no. 4, pp. 1–15, Mar. 2020.

[7] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification,” *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[8] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, “Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1905–1919, Jun. 2019.

[9] L. Sun, C. Ma, Y. Chen, Y. Zheng, H. J. Shim, Z. Wu, and B. Jeon, “Low rank component induced spatial-spectral kernel method for hyperspectral image classification,” *IEEE Trans. Circuits Syst. Video Technol*, to be published, doi: 10.1109/TCSVT.2019.2946723.

[10] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, “Classification of hyperspectral images with regularized linear discriminant analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[11] G. Cheng, Z. Li, J. Han, X. Yao, and G. Lei, “Exploring hierarchical convolutional features for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[12] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[13] Y. Chen, J. Xiong, W. Xu, and J. Zuo, “A novel online incremental and decremental learning algorithm based on variable support vector machine,” *Cluster Comput.*, vol. 22, pp. 7435–7445, 2019, doi:10.1007/s10586-018-1772-4.

[14] Y. Chen, W. Xu, J. Zuo, and K. Yang, “The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier,” *Cluster Comput.*, vol. 22, pp. 7665–7675, 2019, doi:10.1007/s10586-018-2368-8.

[15] Y. Song, G. Yang, H. Xie, D. Zhang, and X. Sun, “Residual domain dictionary learning for compressed sensing video recovery,” *Multimedia Tools Appl.*, vol. 76, pp. 10083–10096, 2017.

[16] E. Blanzieri and M. Farid, “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.

[17] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, “Composite kernels for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[18] P. Ghamisi, A. B. Jon, and O. U. Magnus, “Spectral–spatial classification of hyperspectral images based on hidden Markov random fields,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2565–2574, May 2014.

[19] Y. Chen, M. N. Nasser, and D. T. Trac, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[20] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, “Semi-supervised graph-based hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[21] J. I. Daniel and W. T. Graham, “Semisupervised hyperspectral image classification via neighborhood graph learning,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1913–1917, Sep. 2015.

[22] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, “Spatial sequential recurrent neural network for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4141–4155, Nov. 2018.

[23] P. Zhou, J. Han, G. Cheng, and B. Zhang, “Learning compact and discriminative stacked autoencoder for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 60, pp. 1097–1105, 2012.

[25] 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.

[26] Y. Chen, Z. Lin, X. Zhao, and G. Wang, “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.

[27] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, “Deep convolutional neural networks for hyperspectral image classification,” *J. Sensors*, vol. 2015 no. 2, pp. 1–12, Jul. 2015.

[28] 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.
[29] T. Alipourfard, H. Arefi, and S. Mahmoudi, "A novel deep learning framework by combination of subspace-based feature extraction and convolutional neural networks for hyperspectral images classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 4780–4783.
[30] J. Yue, W. Zhao, S. Mao, and Hui Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, 2015.
[31] Y. Chen, K. Zhu, Z. Lin, and X. He, "Automatic design of convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, Sep. 2019.
[32] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.
[33] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.
[34] 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.
[35] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.
[36] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
[37] D. Zhang, T. Yin, G. Yang, M. Xia, L. Li, and X. Sun, "Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies," *J. Visual Commun. Image Representation*, vol. 48, pp. 281–291, 2017.
[38] L. Xiang, X. Shen, J. Qin, and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Process. Lett.*, vol. 49, pp. 1055–1069, 2019, doi:10.1007/s11063-018-9892-7.
[39] Y. Gui and G. Zeng, "Joint learning of visual and spatial features for edit propagation from a single image," *Visual Comput.*, vol. 36, pp. 469–482, 2020.
[40] D. Zeng, Y. Dai, F. Li, and R. S. Sherratt, "Adversarial learning for distant supervised relation extraction," *Comput. Mater. Continua*, vol. 55, no. 1, pp. 121–136, 2018.
[41] W. Liu, J. Chen, Y. Wang, P. Gao, Z. Lei, and X. Ma, "Quantum-based feature selection for multiclassification problem in complex systems with edge computing," *Collaborative Big Data Manage. Analytics Complex Syst. Edge*, vol. 2020, no. 12, 2020, Art. no. 8216874, doi:10.1155/2020/8216874.
[42] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech :The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[43] S. Zhou, M. Ke, and P. Luo, "Multi-camera transfer GAN for person re-identification," *J. Visual Commun. Image Representation*, vol. 59, pp. 393–400, 2019.
[44] M. Long and Z. Yan, "Detecting Iris liveness with batch normalized convolutional neural network," *Comput., Mater. Continua*, vol. 58, no. 2, pp. 493–504, 2019.
[45] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, Jul. 2006.
[46] S. Wold, K. Esbensen, and P. Geladi, "*Principal component analysis,*" *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.
[47] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools Appl.*, vol. 79, no. 21–22, pp. 15095–15115, Jun. 2020.
[48] P. Du, E. Li, J. Xia, A. Samat, and X. Bai, "Feature and model level fusion of pretrained CNN for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2600–2611, Aug. 2018.
[49] D. Zhang, G. Yang, F. Li, J. Wang, and A. K. Sangaiah, "Detecting seam carved images using uniform local binary patterns," *Multimedia Tools Appl.*, vol. 79, pp. 8415–84302018. doi:10.1007/s11042-018-6470-y.
[50] 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.
[51] S. He, Z. Li, Y. Tang, Z. Liao, J. Wang, and H. Kim, "Parameters compressing in deep learning," *Comput., Mater. Continua*, vol. 62, no. 1, pp. 321–336, 2020.
[52] Z. Wu, L. Shi, J. Li, Q. Wang, L. Sun, Z. Wei, J. Plaza, and A. Plaza, "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.
[53] Z. Wu, Y. Li, A. Plaza, J. Li, 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.

Jinwei Wang received the B.S. degree in automatic control from the Inner Mongolia University of Technology, Hohhot, China, in 2000, and the Ph.D. degree in information security from the Nanjing University of Science & Technology, Nanjing, China, in 2007.

He is a Professor with the Nanjing University of Information Science & Technology. He was a Visiting Scholar with Service Anticipation Multimedia Innovation (SAMI) Lab of France Telecom R&D Center, Beijing, China, in 2006. He worked as a Senior Engineer with the 28th research institute, CETC from 2007 to 2010. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2014 to 2015. He has published more than 50 papers, hosted and participated in more than 10 projects. His research interests include multimedia copyright protection, multimedia forensics, multimedia encryption, and data authentication.

Xiangbo Song received the B.S. degree in network engineering from the Nanjing University of Information Science & Technology, Nanjing, China, in 2018, where he is currently working toward the M.S. degree in software engineering.

His research interests include hyperspectral image processing.

Le Sun (Member, IEEE) was born in Jiangsu, China, in 1987. He received the B.S. degree from the School of Science, Nanjing University of Science and Technology, Nanjing, Jiangsu, China, in 2009, where he received the Ph.D. degree at the School of Computer Science and Engineering, in 2014.

He was doing research in the field of multi-images fusion based on sparse dictionary learning and compressive sensing as a post-doctor with the Digital Media Laboratory of the School of Electronic and Electrical Engineering, Sungkyunkwan University, South Korea, from September 2015 to August 2018. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include sparse representation, compressive sensing, especially in the field of hyperspectral image processing.

Wei Huang was born in Henan, China, in 1982. He received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2015.

He is currently a Lecturer with the School of Computer and Communication Engineering, the Zhengzhou University of Light Industry, Zhengzhou, China. His current research interests include pansharpening, hyperspectral classification, image processing, and machine learning.

Jin Wang received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunication, Nanjing, China, in 2002 and 2005, respectively, and the Ph.D. degree from Kyung Hee University, Seoul, South Korea, in 2010.

He is currently a Professor with the Changsha University of Science and Technology, Changsha, China and the Fujian University of Technology, Fuzhou, China. His research interests mainly include routing protocol and algorithm design, and performance evaluation for wireless ad hoc and sensor networks.

Dr. Wang is a member of ACM.