

# A Simplified 2D-3D CNN Architecture for Hyperspectral Image Classification Based on Spatial–Spectral Fusion

Chunyan Yu <sup>1</sup>, Rui Han, Meiping Song <sup>2</sup>, Caiyu Liu, and Chein-I Chang <sup>3</sup>, *Life Fellow, IEEE*

**Abstract**—Convolutional neural networks (CNN) have led to a successful breakthrough for hyperspectral image classification (HSIC). Due to the intrinsic spatial-spectral specificities of a hyperspectral cube, feature extraction with 3-D convolution operation is a straightforward way for HSIC. However, the overwhelming features obtained from the original 3-D CNN network suffers from the overfitting and more training cost problem. To address this issue, in this article, a novel HSIC framework based on a simplified 2D-3D CNN is implemented by the cooperation between a 2-D CNN and a 3-D convolution layer. First, the 2-D convolution block aims to extract the spatial features abundantly involved spectral information as a training channel. Then, the 3-D CNN approach primarily concentrates on exploiting band co-relation data by using a reduced kernel. The proposed architecture achieves the spatial and spectral features simultaneously based on a joint 2D-3D pattern to achieve superior fused feature for the subsequent classification. Furthermore, a deconvolution layer intends to enhance the robustness of the deep features is utilized in the proposed CNN network. The results and analysis of extensive real HSIC experiments demonstrate that the proposed light-weighted 2D-3D CNN network can effectively extract refined features and improve the classification accuracy.

**Index Terms**—Convolution, convolutional neural networks (CNN), feature extraction, hyperspectral image classification (HSIC).

## I. INTRODUCTION

**H**YPERSPECTRAL remote sense imaging continuously furnishes a variety of information, such as radiation, space, and spectrum of features, which has an insurmountable

Manuscript received December 26, 2019; revised March 6, 2020; accepted March 22, 2020. Date of publication April 27, 2020; date of current version June 5, 2020. The work was supported in part by the National Nature Science Foundation of Liaoning Province under Grant 20170540095, in part by the Fundamental Research Funds for Central Universities under Grant 3132019341, in part by the Recruitment Program of Global Experts for National Science and Technology Major Project, State Administration of Foreign Experts Affairs funded by ZD20180073, and in part by the National Nature Science Foundation of China under Grant 61601077, Grant 61801075, Grant 61971082, and Grant 41801231. (Corresponding author: Meiping Song.)

Chunyan Yu, Rui Han, Meiping Song, and Caiyu Liu are with the Center of Hyperspectral Imaging in Remote Sensing (CHIRS) Information and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: yucy@dlnu.edu.cn; 269899266@qq.com; smping@163.com; liu\_caiyu@163.com).

Chein-I Chang is with the National Yunlin University of Science and Technology, Yunlin 64002, Taiwan, with the Remote Sensing Signal and Image Processing Laboratory Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, MD 21250 USA, and also with the Department of Computer Science and Information Management, Providence University, Taichung 02912, Taiwan (e-mail: cchang@umbc.edu).

Digital Object Identifier 10.1109/JSTARS.2020.2983224

advantage in the domain of feature recognition and classification [1]–[6]. Hyperspectral image classification (HSIC) aims to divide the multiple dimensional feature space into different regions with the same terrain, and has become one of the most rapid developments in earth science and remote sensing field. In the last two decades, CNNs automatically extract deep features with a hierarchical architecture which have been proven to be very successful on a series of visual application and tasks, such as image denoising [7]–[9], image detection [10]–[12], and classification [13], [14]. Nowadays, the existing classification methods based on the CNN framework provide rich solutions for HSIC tasks [4], [15]–[35]. In general, there are three categories of the convolution operation in the existed CNN HSIC frameworks including 1-D CNN, 2-D CNN, and 3-D CNN, respectively.

The network architecture of 1-D CNN is designed to use the pixel vector along the radiometric dimension as a training sample to extract deep feature [17]–[19], which is called spectral-based classification approach conceptually. In [17], it was the first time to employ CNN with multiple layers for HSIC directly in the spectral domain. A novel RNN model [18] was proposed to effectively analyze hyperspectral pixels as sequential data to capture the intrinsic feature, which designed a new activation function to train the network without the risk of divergence.

The 2-D CNN model for HSIC is called spatial-based classification approach that tried to learn spatial features [20]–[24] by utilizing the similar approach for traditional images of RGB, which brought out an inevitable drawback caused by the ignore the united spectral–spatial attributes of the specific hypercube. In [21], Hao and Wang designed the super-resolution aided with class-wise loss (SRCL) model for HSIC, which explored a super-resolution-aided way to construct a spatially enhanced image. In [22], a CNN-MRF model was proposed to integrate spectral and spatial information in a unified Bayesian framework by learning the posterior class distributions. Li and Xie [23] introduced a CNN model to reconstruct an enhanced image cube by bands selection with a new spatial feature-based strategy.

Since the hyperspectral image is originally 3-D hypercube with the spectral and spatial continuity, the HSIC methods integrated both spectral and spatial information have gained more popularity [25]–[29]. Handling the hyperspectral CNN classification with 3-D convolutions is a straightforward way, which is also called the spectral–spatial classification approach. In this way, 3D- regions with joint spatial–spectral information can be processed simultaneously. An automatic design of CNN

for the HSIC framework is explored in [27], which designed 1-D Auto-CNN and 3-D Auto-CNN to automatically extract spectral and spatial information from the original cube. Feng and Yu proposed a multiclass spatial-spectral GAN method to utilize generators for the samples production and the discriminator for the joint spatial-spectral feature extraction. In [29], a semi-supervised 3-D convolutional neural network (CNN) for the spectral-spatial HSIC is proposed by engaging adaptive dimensionality reduction to deal with the problem of the curse of dimensionality. Recently, a series of popular deep learning-based methods have been exploited for spatial-spectral classification. In [30], generative adversarial networks (GAN) was employed for discriminative features extraction, 1-D GAN as a spectral classifier and a robust 3-D GAN as a spectral-spatial classifier were proposed for HSI classification respectively. In [31], a cascaded RNN model was designed to explore the redundant and complementary information of HSIs by utilizing two RNN layers. In [32], the multiscale hierarchical recurrent neural network was proven to be efficient in hyperspectral image classification, which learns the multiscale local feature by 3-D CNNs and learns the spatial dependency of nonadjacent image patches in the spatial domain by RNN. CSA-MSO3DCNN [33] was developed to optimize the discriminative features with attention modules and reduce the redundancy with a three-layer's octave 3-D convolution. However, there are two main limitations revealed with the 3D convolution model. On one hand, with the increasing number of the 3-D kernels, the complexity and time cost get higher, on the other hand, the overwhelming deep features bring out the overfitting problem and the classification accuracy is not high actual as expected.

To alleviate the mentioned issue, in this article, a new deep 2D-3D CNN network with the spectral-spatial fusion is explored by simplifying the original 3-D network structure, which extracted the fusion feature by jointing the spatial and spectral information simultaneously. The proposed deep network is mainly composed of two convolutional blocks for feature extraction. The first block is denoted as 2-D CNN which aims to spatial information extraction with the spectral channels involved. The second block is called the 3-D CNN model which focuses on integrating band information through spectral convolution to generate spectral-spatial representations of the HSI cubes with a reduced size of  $1 \times 1 \times L$  filter kernel. In this framework, the proposed model extracted both spectral features and spatial neighborhood information simultaneously to improve the distinguishing ability of the fusion feature. Besides, we also adopted a deconvolution layer to solve the drawbacks of the different feature sizes and limited representation.

According to the investigation, this article introduced the new 2D-3D structure for the HSIC for the first time, which adopted the efficient model to realize the classification with spectral-spatial feature fusion. The specific contributions are listed as follows.

- 1) The proposed framework performs joint learning of 3-D spatial-spectral representations with the flexibility of two types of kernels to achieve refined characteristics for classification. Specifically, the 3-D convolution with the kernel size of  $m \times m \times L$  is utilized to mine hidden feature

effectively in the situation for the improvement of the classification accuracy, whereas, the simplified kernel size of  $1 \times 1 \times L$  is adopted to reduce the spatial redundancy with less parameters. The relationship between the two models is the first time to clarify that the classification performance is related to both the size of the kernel and the experimental parameters, generally, the standard kernel performs well with more parameters and the simplified kernel is advantage for the efficiency.

- 2) To our knowledge, the network is enabled to supply an implementation with the depth-wise separable convolution way for HSIC equipped with the simplified 3-D kernel. Specifically, a 3-D convolution with the kernel size of  $1 \times 1 \times L$  is explored for the first time to implement the fusion extraction of spectral information between bands. In this way, the spectral information is extracted by the merge of the neighbor band data, and the spatial and spectral feature is fused simultaneously at the same time.
- 3) To improve the efficiency of the proposed model, we employed only two convolution blocks instead of a very deep network structure. The spatial information convolution is implemented in the 2-D convolution layer with a bunch of filter kernels to expand the spatial features, and the 3-D block relies on only one 3-D convolution kernel to increase the convolution speed and overcome the overfitting problems.
- 4) Especially, we reveal the intrinsic relationship between the proposed model with simplified 3-D kernel and the 1-D CNN on HSIC, and the comparisons between our model and 1D-2D CNN network are explained in this article, which will offer a systematic reference for the learning of the 1-D, 2-D, and 3-D CNN HSIC. Besides, we conduct experiments to exhaustively exploit the performance of different structures based on the proposed network, which will inspire the design of other CNN framework for hyperspectral applications.

The rest of this article is organized as follows. The detail of the proposed 2D-3D CNN classification network is given in Section II. Experimental results and analysis are illustrated in Section III and conclusion is drawn in Section IV.

## II. CLASSIFICATION METHOD BASED ON 2D-3D CNN NETWORK

### A. Network Structure of the Proposed Framework

The main task of hyperspectral image processing of CNN architecture is handling spatial and spectral information simultaneously via adjacent layers. In this article, we proposed the spatial-spectral joint CNN network which contained seven layers, Fig. 1 showed the hierarchical structure diagram of the proposed 2D-3D-D CNN classification framework with the specific parameters. Specifically, the proposed CNN architecture consists of two convolution blocks (2-D CNN, 3-D CNN), the rich spatial convolutional features are obtained by the 2-D CNN model, the 3-D CNN model accomplished the spectral feature processing by fusing with the neighbor band information, an

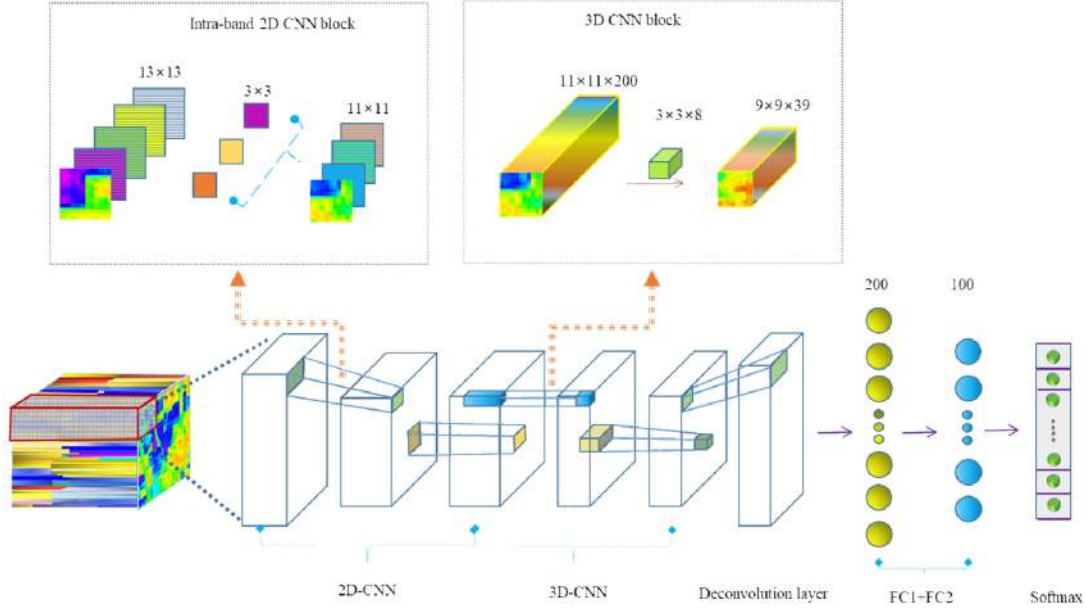


Fig. 1. Hierarchical structure diagram of the convolution neural network.

activator is implemented by the pooling layer in each convolution part which aims to reduce data variance and improve the nonlinearity of the feature. To maintain a good performance of the fusion of the extracted feature, a deconvolution layer is followed in the 3-D CNN part to reconstruct the feature map. Two full connection layers (FC1, FC2) are used separately to reassemble the obtained local features for the output layer. The proposed 2D-3D CNN method has the two following distinctive advantages. First, the 2-D CNN extracted rich spatial feature maps with the channel information involved. Second, we utilized a 3-D CNN to mine the integrated spectral-spatial characteristic. Specially, we design a simplified one filter with the size of  $1 \times 1 \times L$  to exploit the spectral information in the 3-D CNN network effectively. The proposed network integrated the spatial and spectral information simultaneously with the strategy of 3-D convolution. The more details of the CNN framework are introduced in the following sections.

### B. 2-D CNN With Intra-band Information

The training of the 2-D CNN block includes two layers, the first layer is a 2-D convolution layer to mine the given HSI data slice in a local perception mechanism. Assume that the sample of a hyperspectral image is denoted as  $X$  with dimension of  $m \times n \times u$ , here  $X \in R_0^{m \times n \times u}$  is a 3-D tensor,  $m \times n$  means the size of the training sample,  $u$  represents the number of spectral bands,  $X(i)$  denotes the spatial domain on the  $i$ th band of  $X$ . The spatial-spatial feature extraction using 2-D CNN is obtained by the following formula:

$$X_{2D} = C_{2D}(X(i)) = \frac{1}{u} \sum_{i=1}^u \sum_{j=1}^{t1} X(i) \Theta w_j^{2D} + b_j^{2D}. \quad (1)$$

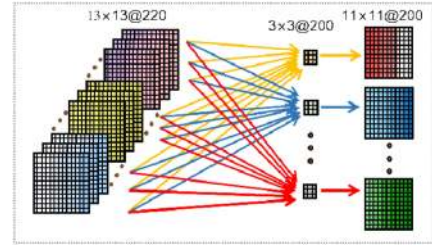


Fig. 2. Illustration of intraband 2-D CNN approach.

Here  $w_j^{2D}$  and  $b_j^{2D}$  represent the parameters and bias of the  $j$ th filter of size  $a_1 \times b_1 \times t_1$ ,  $\Theta$  means the 2-D convolution operator.

The 2-D convolution operator is usually applied to extract features in the spatial domain. To strength the feature representation ability, the spectral feature is utilized as channel information to participate in the production of feature extraction. Fig. 2 shows the illustration of the intraband 2-D CNN approach, we can observe that all the spectral bands are utilized to participate in the calculation of convolution, which can yield rich spatial features for the subsequent layer.

The second layer is called pooling layer, after the convolution layer, the nonlinear activation function is utilized as an activator to generate nonlinear feature in the 2-D CNN model, in this article, we use zero-padding policy to keep the same size of next input feature for the following convolution layer, the specific equation of activator is listed as follows:

$$X'_{2D} = \max \left( 0, \sum_{j=1}^{t2} X_{2D} \Theta w'_j + b'_j \right). \quad (2)$$

Here the size of the filter kernel is  $a_2 \times b_2 \times t_2$ ,  $w'_j$  and  $b'_j$  represents the parameter and bias of the filter. In this module,

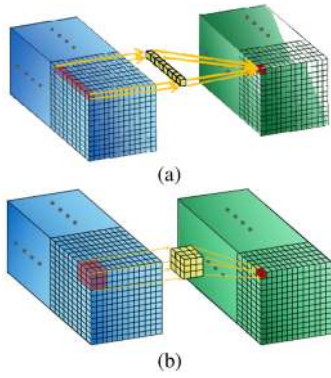


Fig. 3. Illustration of the 3-D CNN. (a) With  $1 \times 1 \times L$  ( $L = 8$ ) kernel. (b) With  $3 \times 3 \times 3$  kernel.

the spatial feature is captured with the intraband information involved, which is expressed in the feature quantification not only in the spatial region but also in the band domain.

### C. 3-D CNN for Spectral Feature Fusion

The 3-D CNN block is designed to complete spatial–spectral information fusion after the 2-D CNN part. In this section, we consider the regular convolution layer for a 3-D hyperspectral image represented by a 4-D tensor  $Y \in R_1^{n_h \times n_w \times l_h \times k_c}$  in the Tensorflow framework, where  $n_h \times n_w \times l_h$  is the size of the input data,  $k_c$  denotes the number of the channel, respectively, in this article, we set the channel number  $k_c$  to 1 in the implementation framework. The refined feature through 3-D CNN is denoted as  $Y_{3D}$  which is obtained by the 3-D filter with the size  $k_h \times k_w \times k_n$  applied to the input data by the following equation:

$$Y_{3D} = C_{3D}(X'_{2D}) = \sum_{j=1}^N X'_{2D} \odot w^{3D} + b^{3D}. \quad (3)$$

Here the symbol  $\odot$  denotes the 3-D convolution operator,  $w^{3D}$  and  $b^{3D}$  represent the parameter and bias of the 3-D filter,  $N$  is the number of the 3-D kernel. The 3-D convolution operation is illustrated in Fig. 3, which is applied to a 3-D block for the capture of the integration of spatial–spectral patch with a 3-D kernel, the calculation of convolution is implemented in a sequential manner, e.g., from top to bottom, from left to right. Clearly, the complexity becomes progressively large as the parameters increase, which caused more training time and overfitting problems.

Different from the original 3-D convolution, we designed a simplified 3-D CNN with a filter kernel size of  $1 \times 1 \times L$  to reduce approximately  $k_h \times k_w$  times of the original parameters. According to the simplified structure, the 3-D CNN block focuses on integrating the characteristic information of adjacent bands with the strike step. In this case, the 3-D CNN model captured the spectral fusion features with faster speed. The key roles of the  $1 \times 1 \times L$  kernel reflected in the two aspects, on one side, the implementation achieves an action to reduce the band dimension with a fusion way, on the other side, the operation performs exceptionally more nonlinear information

for the feature extraction. Next, the activator function of *ReLU* to generate the nonlinear and sparse features of the CNN network is listed as follows:

$$Y'_{3D} = \max(0, Y_{3D} \odot w'_{3D} + b'_{3D}). \quad (4)$$

Here  $w'_{3D}$  and  $b'_{3D}$  represents the parameter and bias of the pooling layer.

Compared with the regular CNN implementation, the proposed classification network for hyperspectral image consists of two steps separately, the 2-D CNN performs spatial convolution independently including band information, followed by the 3-D CNN convolution projecting the output of the spectral channels onto a new fusion feature to generate a powerful representation capability.

### D. Remarks on the $1 \times 1 \times L$ kernel

In the 3-D CNN part, it is worth noting that the 3-D convolution operation with the kernel size fixed as  $1 \times 1 \times L$  becomes to the 1-D convolution conducted in 3-D space. In this section, the analysis of the 3-D operation with  $1 \times 1 \times L$  kernel is conducted in two aspects. On one hand, due to the fact that the traditional 1-D CNN approaches for HSIC deal with the classification application in a pixel-wise way, the 1-D CNN HSIC model usually needs to transform the sample training cube into a vector in a pixel-wise way first, and all the refined features have to combine to an integrated 3-D patch after the convolution operation, which increased the time cost obviously with the data conversion back and forth. While with the  $1 \times 1 \times L$  kernel, the 3-D convolution operation can easily be done by the setting of the filters and the channel number. On the other hand, compared with the existing 1D-2D CNN architectures, the key to our proposed HSIC framework is to use 2D-1D networks to explicitly learn spectral and spatial feature jointly. Specific, the traditional 1D-2D CNN extracted the features from 1-D CNN and 2-D CNN in parallel part and the features are fused in a fully connected layer usually, which means that the implementation of feature extraction actually is produced separately. Instead of producing feature maps independently, we employ the 2-D CNN to extract abundant spatial features first and the subsequent 1-D CNN block (3-D with  $1 \times 1 \times L$  kernel) aims to further refine the feature in the spectral domain. The proposed CNN attempts to perform the feature extraction with spectral and spatial information simultaneously to enhance the feature representation.

### E. Optimization Details of the Proposed CNN Architecture

Furthermore, we adopted a series of optimization policy to improve the performance of the CNN Network in this article. First, in the preprocessing period, the mirroring strategy is used to enhance the classification accuracy of the pixels around the borders. The specific operation means to keep the border pixel be the center of the modified sample by expanding the original image. Second, we used the Root Mean Square prop (RMSprop) [36] as the optimization algorithm for the proposed network training, which is attenuated through a certain ratio by introducing an attenuation coefficient. Lastly, in the CNN network, the detailed structure of the feature is usually weakened or smoothed

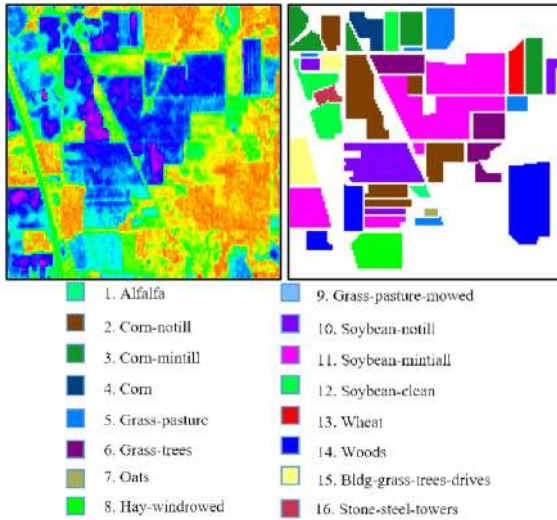


Fig. 4. Ground truth image of Purdue Indiana Indian Pines Scene.

out, in order to enhance the expression of the extracted feature, we utilize deconvolution operation to expand the feature map as identical as the size of the input sample. A deconvolution operation aims to increase the size of the feature map, which is inverse to convolution operation and plays an important role in FCN and GAN networks to map low-resolution input into a high-resolution feature. The option can be implemented by upsampling to enlarge the feature map with convolution kernel and padding, which has better effectiveness because it is a learning-based upsampling operation. For example, the size of input data is  $5 \times 5$ , the extended feature map is  $7 \times 7$  after zero-padding and the deconvolution procedure with a kernel size of  $3 \times 3$ . In our article, the deconvolution layer has a kernel filter with a size of  $3 \times 3$ , and the length of the deconvolution filter is set to 39.

### III. EXPERIMENT AND RESULT ANALYSIS

#### A. Data Description

In this article, we have employed four well-known hyperspectral image datasets in the experiments.

1) *Purdue Indiana Indian Pines Scene*: The first one namely Purdue Indiana Indian Pines Scene is an AVIRIS image, which was collected over North-western Indiana. The image is characterized by  $145 \times 145$  pixels with 220 spectral bands in the range of  $0.4\text{--}2.5 \mu\text{m}$ . As shown in Fig. 4, the Purdue Indian Pines Scene consists of 16 classes available in the ground truth image.

2) *Salinas Valley*: The second dataset is the Salinas Valley scene captured by the AVIRIS sensor over the Salinas Valley in Southern California. The data contain 224 bands and the spatial resolution is  $512 \times 217$ . According to the ground truth image shown in Fig. 5, there are 16 categories of classes labeled in different colors.

3) *Kennedy Space Center (KSC)*: The third dataset used for the experiment is called KSC data, which was collected by AVIRIS in the range of  $0.4\text{--}2.5 \mu\text{m}$  of Kennedy Space Center

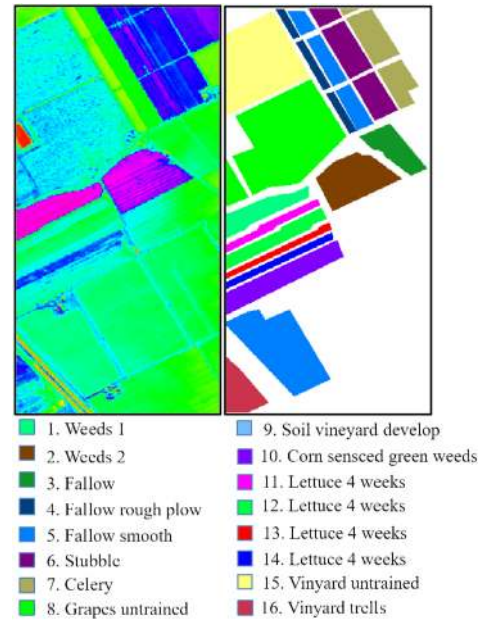


Fig. 5. Ground truth image of Salinas valley.

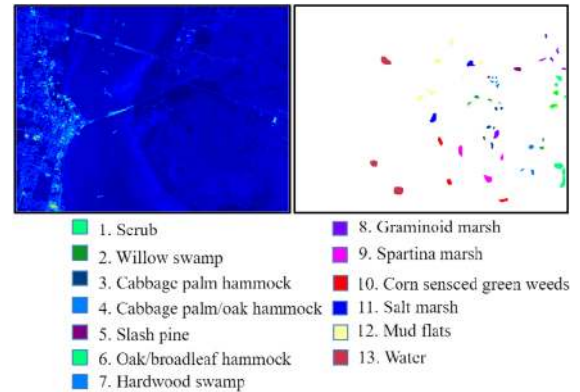


Fig. 6. Ground truth image of KSC.

located in Florida. After removing water absorption and low SNR bands, the subimage with the size of  $512 \times 614$  remains 176 bands for the analysis. The scene contains various land cover types represented by 13 classes labeled 1–13 as shown in Fig. 6.

4) *University of Pavia*: The last dataset used in the following experiment is the University of Pavia acquired by the ROSIS-03 satellite sensor. This scene has 103 spectral bands and  $610 \times 340$  pixels with a spatial resolution of 1.3 m. Fig. 7 shows the Ground truth image of the University of Pavia with nine classes of interest.

#### B. Experimental Configuration

In this section, the hyperspectral image classification methods based on the three datasets were performed to evaluate the proposed CNN model. The experiments were run on a computer with Intel (R) Core (TM) i7-7820X CPU, 3.60 Ghz, Nvidia Geforce GTX 1050Ti, RAM 32.0 GB for the Salinas data, GTX

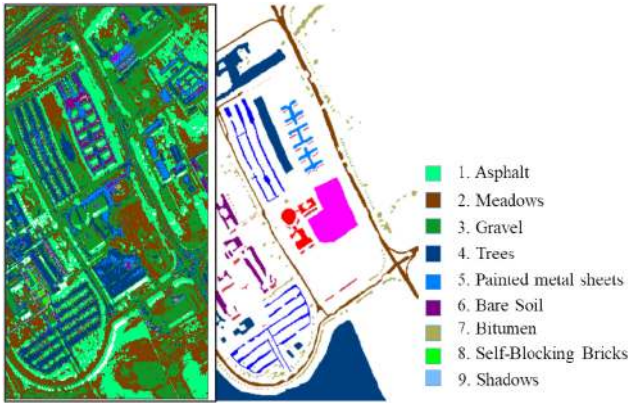


Fig. 7. Ground truth image of the University of Pavia.

TABLE I  
HYPERPARAMETERS SETTING OF THE PROPOSED MODELS

Type	2D-3D-D-S	2D-3D-D
Batch size	100	100
LR	0.001	0.001
2D CNN	3*3*200 stride=[1,1]	3*3*200 stride=[1,1]
3D CNN	1*1*8*1 stride=[1,1,5]	3*3*3*1 stride=[1,1,1]
Deconvolution	3*3, 200 stride=[1,1]	3*3, 200 stride=[1,1]
FC 1	200	600
FC 2	100	150

1060Ti, RAM 16.0 GB for the other three datasets, the platform is python 3.6 of the Tensorflow framework. We evaluate the proposed architecture with classification performances in terms of Overall Accuracy (OA), here OA measures the number of samples is classified correctly.

We also exploited a series of other CNN models for comparison with the proposed architecture, here denoted the abbreviation symbols of the compared CNN networks. The network proposed in the article is called as 2D-3D-D model, the 3-D part with simplified kernel is denoted 2D-3D-D-S, the 2-D CNN methods has only one 2-D convolution layer, one pooling layer, and two fully connected layers, the 2D+D CNN network plus one deconvolution layer and the 2-D+3-D CNN model include one 2-D convolution layer, one 3-D convolution layer, one pooling layer, and two fully connected layers. The hyperparameters setting of the proposed 2D-3D-D models are listed in Table I. The other compared existing methods include SVM [37], EPF [38], LCMV [39], MFASR [40]. In addition, in our experiment for the CNN networks, the batch size is initialized to 100, the iteration number of training is fixed to 5000. All the other CNN networks mentioned for comparison have the same parameters as 2D-3D-D architecture. Each execution of all the CNN networks has been repeated 5 times and the classification accuracy reported in our experiment is averaged by the results, which is represented in the form of mean  $\pm$  standard deviation.

Furthermore, it is noted that the existing challenging problems of HSIC are the few-shot learning and the unbalanced sample, therefore the data augmentation is usually a necessary operation to generate the samples before the training launch. In the preprocessing phase, we expand the training dataset by augmentation in this article. The specific types of addition are listed as follows:

TABLE II  
NUMBER OF SAMPLES IN THE TRAINING SET OF THE PURDUE INDIANA INDIAN PINES SCENE USED IN 2D-3D-D CNN METHODS

Class No	Class Name	Sample Number			
		Total	Training	Augmentation	Testing
1	Alfalfa	46	5	20	41
2	Corn-notill	1428	143	428	1285
3	Corn-mintill	830	83	249	747
4	Corn	237	24	71	213
5	Grass-pasture	483	49	145	434
6	Grass-trees	730	73	219	657
7	Grass-pasture-mowed	28	3	12	25
8	Hay-windrowed	478	48	143	430
9	Oats	20	2	8	18
10	Soybean-notill	972	98	292	874
11	Soybean-mintill	2455	246	737	2209
12	Soybean-clean	593	60	178	533
13	Wheat	205	21	62	184
14	Woods	1265	127	380	1138
15	Buildings-Grass-Trees-Dr	386	39	116	347
16	Stone-Steel-Towers	93	10	41	83

TABLE III  
NUMBER OF SAMPLES IN THE TRAINING SET OF THE SALINAS VALLEY USED IN 2D-3D-D CNN METHODS

Class No	Class Name	Sample Number			
		Total	Training	Augmentation	Testing
1	Broccoli_green_weeds_1	2009	101	502	1908
2	Broccoli_green_weeds_2	3726	187	932	3539
3	Fallow	1976	99	494	1877
4	Fallow_rough_plow	1394	70	348	1324
5	Fallow_smooth	2678	134	669	2544
6	Stubble	3959	198	989	3761
7	Celery	3579	179	894	3400
8	Grapes_untrained	11271	564	2818	10707
9	Soil_vinyard_develop	6203	311	1551	5892
10	Corn_senesced_green_w	3278	164	819	3114
11	Lettuce_romaine_4wk	1068	54	267	1014
12	Lettuce_romaine_5wk	1927	97	482	1830
13	Lettuce_romaine_6wk	916	46	229	870
14	Lettuce_romaine_7wk	1070	54	268	1016
15	Vinyard_untrained	7268	364	1817	6904
16	Vinyard vertical trellis	1807	91	452	1716

- 1) reverse the original data from up to down;
- 2) reverse the training sample from left to right,
- 3) increase data by adding random Gaussian noise to the training sample.

For a specific class, we gather the data augmentation according to the number of the total samples of the ground truth image, specifically, when the number of total samples is less than 200, the percentage number of augmentation is set to 1/3 and the augmentation percentage is set to 1/5 for other situation. During the preprocessing phase, the training data is randomly selected from the original datasets and is expanded with the above ways randomly, where the mean value of Gaussian noise is set to 0 and the variance is 0.05. The specific numbers of the training sample of the three data sets are reported in Tables II–V, which included the number selected randomly from every class and the number generated by the augmentation method, and the number in the column of *Augmentation* denotes the sum number of selected and extended sample

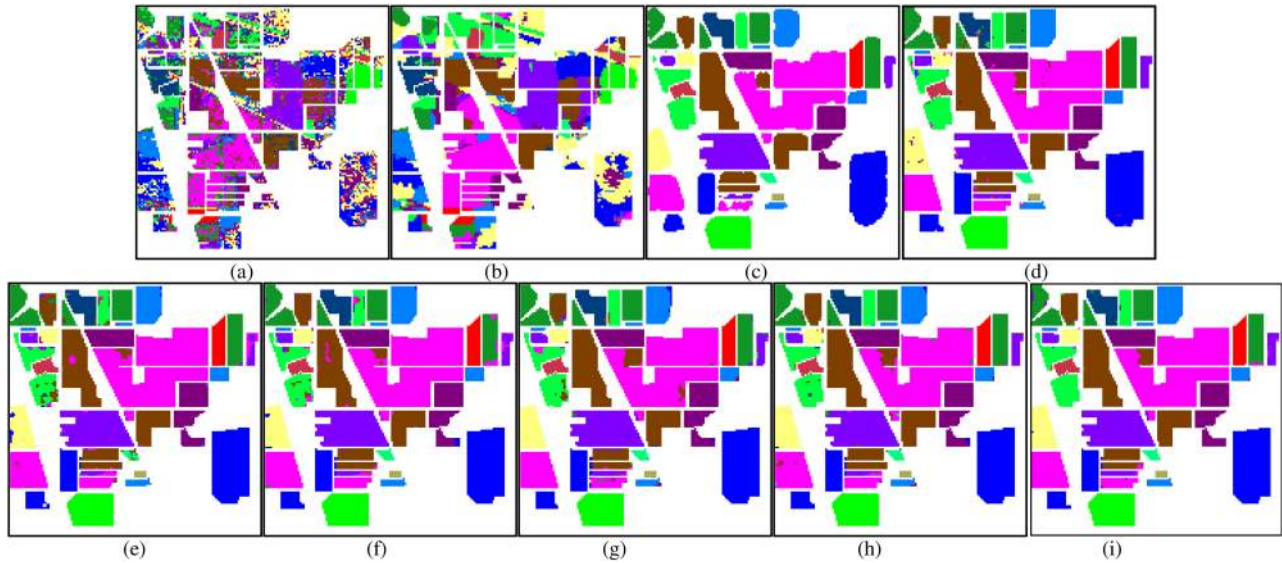


Fig. 8. Classification results of the Purdue Indiana Indian Pines Scene with compared methods. (a) SVM. (b) EPF. (c) LCMV. (d) MFASR. (e) 2DCNN. (f) 2D+D. (g) 2D+3D. (h) 2D-3D-D-S. (i) 2D-3D-D.

TABLE IV  
NUMBER OF SAMPLES IN THE TRAINING SET OF THE KENNEDY SPACE CENTER  
USED IN 2D-3D-D CNN METHODS

Class No.	Class Name	Sample Number			
		Total	Training	Augmentation	Testing
1	Scrub	761	77	229	684
2	Willow swamp	243	25	73	218
3	Cabbage palm hammock	256	26	77	230
4	Cabbage palm/oak hammock	252	26	76	226
5	Slash pine	161	17	70	144
6	Oak/broadleaf hammock	229	23	68	206
7	Hardwood swamp	105	11	46	94
8	Graminoid marsh	431	44	130	387
9	Spartina marsh	520	52	156	468
10	Cattail marsh	404	41	121	363
11	Salt marsh	419	42	125	377
12	Mud flats	503	51	151	452
13	Water	927	93	278	834

TABLE V  
NUMBER OF SAMPLES IN THE TRAINING SET OF UNIVERSITY OF PAVIA  
USED IN 2D-3D-D METHODS

Class No.	Class Name	Sample Number			
		Total	Training	Augmentation	Testing
1	Asphalt	6631	332	1658	6299
2	Meadows	18649	933	4662	17716
3	Gravel	2099	105	524	1994
4	Trees	3064	154	766	2910
5	Painted metal sheets	1345	68	337	1277
6	Bare Soil	5029	252	1257	4777
7	Bitumen	1330	67	333	1263
8	Self-Blocking Bricks	3682	185	921	3497
9	Shadows	947	48	237	899

### C. Results and Analysis

The training sample percentage of the Purdue Indiana Indian Pines Scene is set to 10% randomly, the dropout rate is initialized to 0.3 for 2D-3D-D-S and 0.8 for 2D-3D-D, the sample size is

fixed as  $13 \times 13$ . Fig. 8 illustrated the classification maps of the dataset with each classifier mentioned above, the performance of the proposed 2D-3D-D CNN network is much better than the compared models. Moreover, the accuracy of each class is demonstrated individually in Table VI and the OA of the Purdue data is shown in the last line. According to the listed value, it can be observed that our proposed 2D-3D-D-S model achieves the overall accuracy of 97.98%, and the second (97.49%) OA is implemented by the MFASR model. It is 0.49% and 0.88% higher than 2D+D and 2D+3D model, respectively, and the best OA is 98.33% with the 2D-3D-D framework. Compared with the OA 94.88% of the 2-D model, it is obtained that the deconvolution layer and 3-D convolution operations have an important positive effect on the feature refined under the same conditions of other network structure parameters. Also, it also can be seen that the proposed method generates the best accuracy of class 4, and class 7. Besides, we can also observe that the accuracy of each class of the proposed model is relatively high, it can be concluded that the overall classification performance is more stable in the proposed framework, and there is no case where the accuracy is much higher or lower for different classes.

The colorful classification maps of each method for the Salinas Valley are illustrated in Fig. 9, in this experiment, the training sample percentage is set to 5%, the setting of the dropout rate and the size of the training data is 0.3 for 2D-3D-D-S and 0.8 for 2D-3D-D and  $13 \times 13$ , respectively. Table VII lists the accuracy of each class and OA of the Salinas data. As the Salinas data have a rich sample size and relatively regular spatial distribution, several network structures generate good performance with almost the same overall accuracy. Objectively, it is clearly to be seen that the 2D-3D-D model obtained the best experimental results with an overall accuracy of 99.07%. In contrast, both the MFASR model and the 2D+D model showed good results with an overall accuracy of 97.91% and 97.90%, respectively. Also, our proposed model is relatively 0.3% higher

TABLE VI  
 OA CALCULATED FROM THE CLASSIFICATION RESULTS OF PURDUE INDIANA INDIAN PINES SCENE WITH ALL THE COMPARED METHODS (10%)

Class P <sub>0</sub> %	SVM	EPF	LCMV	MFASR	2D	2D+D	2D+3D	2D-3D-D-S	<b>2D-3D-D</b>
1	84.78	100	95.65	100	99.56±0.99	100.00±0.0	100.00±0.0	99.55±1.02	100±0
2	70.38	83.68	96.01	96.58	88.26±3.06	97.95±1.66	96.37±2.02	97.61±1.01	98.36±0.62
3	62.05	72.17	96.99	97.59	97.01±1.12	97.38±1.13	96.57±1.19	96.62±1.81	97.80±0.76
4	89.03	99.58	98.73	89.67	97.97±2.27	95.67±6.87	96.41±3.37	98.75±1.55	97.20±1.86
5	90.89	94.20	89.44	97.93	97.53±4.38	99.75±0.17	98.32±2.49	99.67±0.42	99.30±0.50
6	95.48	99.59	97.12	98.63	98.95±1.17	99.70±0.52	99.09±0.43	99.34±0.41	99.07±0.41
7	96.43	96.43	100	96.00	100.00±0.0	98.62±1.89	100.00±0.0	100±0	100±0
8	96.86	100	98.78	100	98.85±1.0	99.46±0.11	99.58±0.33	99.46±0.18	99.83±0.18
9	85.00	95.00	100	94.44	98.18±4.07	99.05±2.13	87.71±22.75	95.42±4.54	92.72±3.73
10	72.94	87.55	93.93	97.71	94.89±2.75	98.10±0.68	95.51±2.49	97.35±1.50	97.34±0.84
11	65.42	88.47	94.70	99.00	93.11±2.73	95.06±3.72	96.26±1.42	97.73±0.95	98.23±0.53
12	67.79	97.64	95.45	95.31	97.46±1.71	98.30±0.87	98.01±1.16	96.79±2.38	97.66±0.93
13	97.56	99.51	98.54	99.46	99.32±0.81	99.61±0.41	99.90±0.22	99.42±0.85	99.32±0.65
14	89.96	98.74	93.52	99.03	98.43±1.45	98.89±0.38	98.33±0.82	98.71±0.48	99.01±0.47
15	71.50	96.89	90.67	95.68	96.27±1.60	98.62±1.42	96.88±2.31	98.23±1.55	98.60±0.86
16	100	100	98.92	97.59	91.27±0.18	89.97±5.41	96.96±3.28	94.92±3.19	92.59±3.08
POA	76.47	90.78	95.09	97.83	94.88±0.89	97.49±1.03	97.10±0.51	97.98±0.19	<b>98.33±0.21</b>

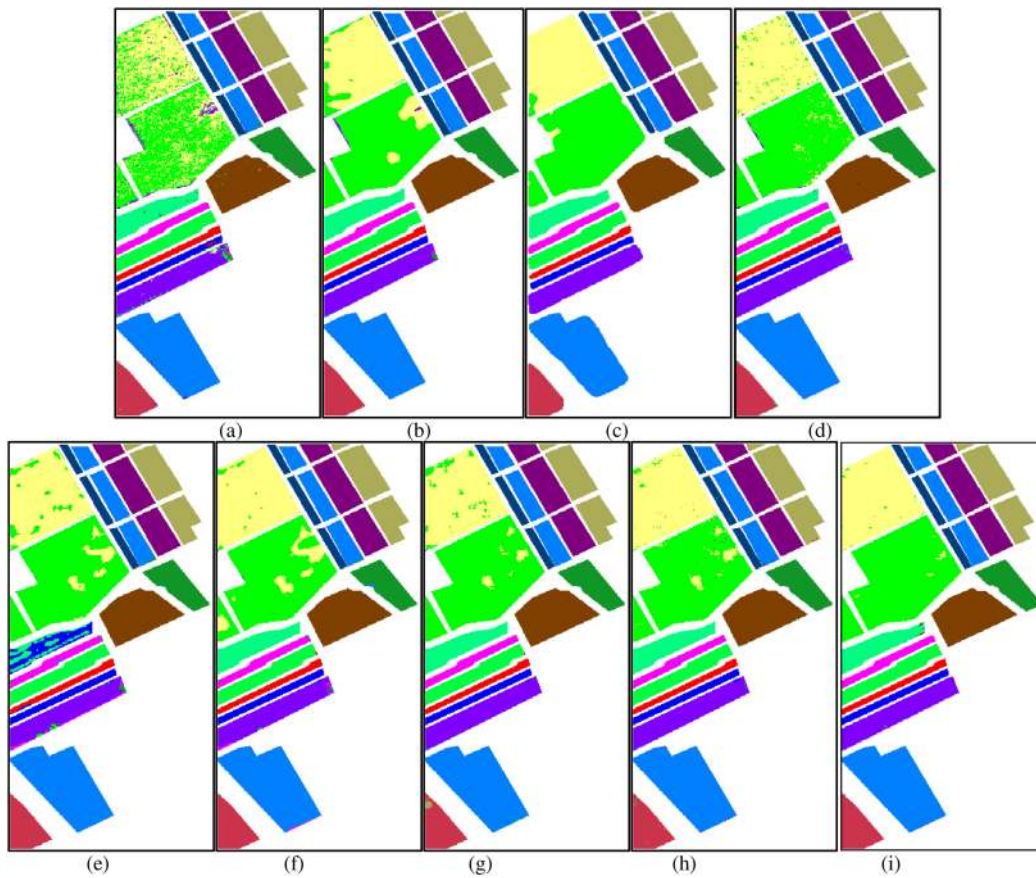


Fig. 9. Classification results of the Salinas valley with compared methods. (a) SVM. (b) EPF. (c) LCMV. (d) MFASR. (e) 2DCNN. (f) 2D+D. (g) 2-D+3-D. (h) 2D-3D-D-S. (i) 2D-3D-D.

than the 2-D+3-D model. According to the obtained value of the different classes, the proposed network can obtain a more stable accuracy.

In the experiment for the KSC data, we fix the training sample percentage as 10%, and the data size and the dropout rate are set to 0.3, the kernel numbers of FC1 and FC2 for 2D-3D-D are 600 and 150. Fig. 10 demonstrates the classification results with the compared methods. The accuracy of each class and OA is

illustrated in Table VIII objectively, it can be seen from the table that our CNN architecture yielded an overall accuracy of 97.14% and 97.47%. The MFASR model based on sparsity generates the results of 98.31%, followed by 2-D+D and 2-D+3-D models with the OA of 96.30% and 93.58%, respectively. Due to the super sparsity of the KSC data itself, the MFASR has the better experimental result than our proposed method, although, for the class 5, the OA is only 79.86% in MFASR, while the OA of



TABLE VII  
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF SALINAS VALLEY WITH ALL THE COMPARED METHODS (5%)

Class P <sub>OA</sub> %	SVM	EPF	LCMV	MFASR	2D	2D+D	2D+3D	2D-3D-D-S	2D-3D-D
1	99.15	100	95.52	99.90	99.91±0.20	95.78±8.53	99.98±0.04	99.43±0.28	99.81±0.42
2	99.30	100	98.42	99.66	93.15±14.83	97.03±6.64	99.70±0.58	99.69±0.50	99.65±0.36
3	98.94	100	93.78	99.95	93.75±7.22	99.55±0.70	96.41±4.97	99.50±0.47	99.75±0.22
4	99.43	100	95.62	99.85	99.61±0.43	99.17±1.08	99.27±0.98	99.06±0.63	99.37±0.68
5	97.80	99.33	96.90	99.17	99.04±0.70	98.38±3.13	99.78±0.21	99.25±0.32	98.68±1.97
6	99.75	100	98.79	99.57	99.78±0.16	99.99±0.01	100.00±0.0	99.87±0.06	99.99±0.02
7	99.75	100	98.63	99.24	95.14±8.77	99.74±0.30	99.90±0.18	99.60±0.50	99.88±0.13
8	71.91	91.08	96.69	94.95	85.40±13.51	95.37±2.70	95.17±3.39	<b>97.17±1.46</b>	98.05±1.13
9	99.21	99.95	95.87	100	99.80±0.26	99.91±0.07	99.47±0.46	99.82±0.17	99.80±0.19
10	91.06	97.47	96.67	99.00	99.63±0.24	99.92±0.18	99.92±0.05	98.85±0.83	99.86±0.08
11	98.78	100	97.75	99.70	76.60±11.25	99.31±0.45	99.83±0.21	96.61±4.14	98.67±0.98
12	99.84	100	97.15	100	93.77±5.15	99.96±0.09	99.96±0.07	98.83±1.83	99.92±0.13
13	99.24	100	96.51	97.59	99.69±0.27	99.96±0.06	99.96±0.10	98.96±0.82	99.89±0.19
14	93.74	99.91	95.89	97.05	84.06±21.13	99.43±0.68	98.62±1.38	98.72±3.68	99.40±1.00
15	70.65	89.97	94.00	94.93	94.36±3.97	96.43±2.21	93.94±5.48	93.52±4.34	97.76±1.82
16	98.56	100	93.30	98.78	99.96±0.06	100.00±0.0	99.96±0.10	99.72±0.22	99.88±0.19
POA	89.09	96.60	96.37	97.91	92.24±5.20	97.90±1.09	97.77±0.45	98.07±0.60	99.07±0.38

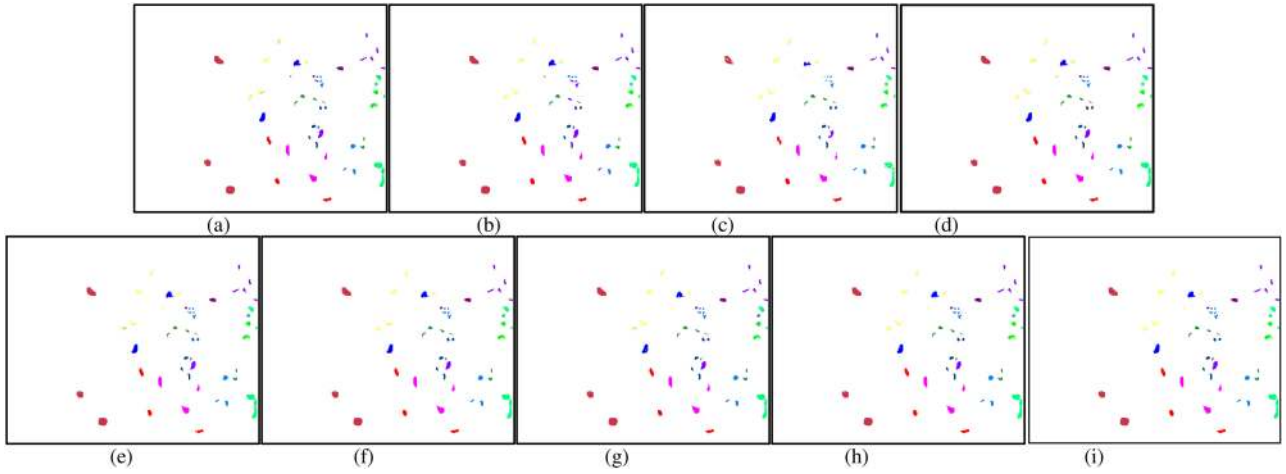


Fig. 10. Classification results of the KSC data with compared methods. (a) SVM. (b) EPF. (c) LCMV. (d) MFASR. (e) 2DCNN. (f) 2-D+D. (g) 2-D+3-D. (h) 2D-3D-D-S. (i) 2D-3D-D.

TABLE VIII  
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF KENNEDY SPACE CENTER WITH ALL THE COMPARED METHODS (10%)

Class P <sub>OA</sub> %	SVM	EPF	LCMV	MFASR	2D	2D+D	2D+3D	2D-3D-D-S	<b>2D-3D-D</b>
1	86.07	100	87.68	97.37	93.19±4.43	95.16±1.78	89.49±3.35	97.06±1.06	98.19±0.70
2	89.71	66.26	88.52	100	96.38±1.74	97.54±1.84	92.66±2.94	95.52±2.43	96.75±2.31
3	91.41	80.47	93.02	100	97.88±1.36	97.19±1.66	96.04±2.67	97.33±1.73	98.31±0.32
4	75.00	53.17	80.24	99.12	82.01±7.46	90.34±5.89	87.83±2.50	89.79±4.14	87.50±1.24
5	83.85	80.12	79.50	79.86	85.06±4.49	87.51±3.85	83.60±3.27	84.59±2.43	94.10±3.22
6	80.35	100	95.20	100	88.17±4.63	99.05±0.74	87.97±6.17	99.61±0.55	98.73±1.19
7	95.24	100	96.33	100	90.36±4.44	93.59±2.35	83.86±8.44	94.82±3.27	95.87±4.33
8	90.95	100	82.60	100	94.63±6.71	98.52±1.37	97.36±2.51	98.29±2.04	96.32±3.75
9	97.69	100	94.23	100	95.00±4.22	93.66±1.36	91.27±3.95	97.44±0.82	97.34±0.84
10	89.11	99.75	91.85	96.42	95.12±2.67	98.19±1.52	98.38±0.97	97.89±1.08	97.35±2.04
11	97.37	100	82.38	100	88.60±1.10	98.96±0.36	97.03±2.80	99.88±0.20	99.48±0.61
12	94.43	97.81	95.83	97.35	86.64±2.69	96.41±1.80	94.65±1.60	97.19±1.72	98.90±1.22
13	99.89	100	91.37	99.40	96.88±2.58	98.49±1.73	98.37±1.83	99.55±0.24	99.70±0.47
POA	91.81	94.36	89.45	98.31	93.04±0.86	96.30±0.41	93.58±1.58	97.14±0.26	<b>97.47±0.39</b>

our proposed network is 84.59% and 94.10%, it can be analyzed that our network still has more stable performance for each class than other classifiers in the KSC data.

For the experiment of the University of Pavia, the training sample percentage is 5%, the training size of the dataset is fixed as  $13 \times 13$ , and the drop out rate is set to 0.3 for 2D-3D-D-S

and 0.8 for 2D-3D-D. Fig. 11 illustrates the classification results with the mentioned HSIC methods, the accuracy of each class and OA of Pavia data is shown in Table IX objectively. As can be observed, the 2D-3D-D model wins the other compared classification methods in terms of the OA criterion, the proposed architecture achieves the best OA of 99.54% and the highest OA

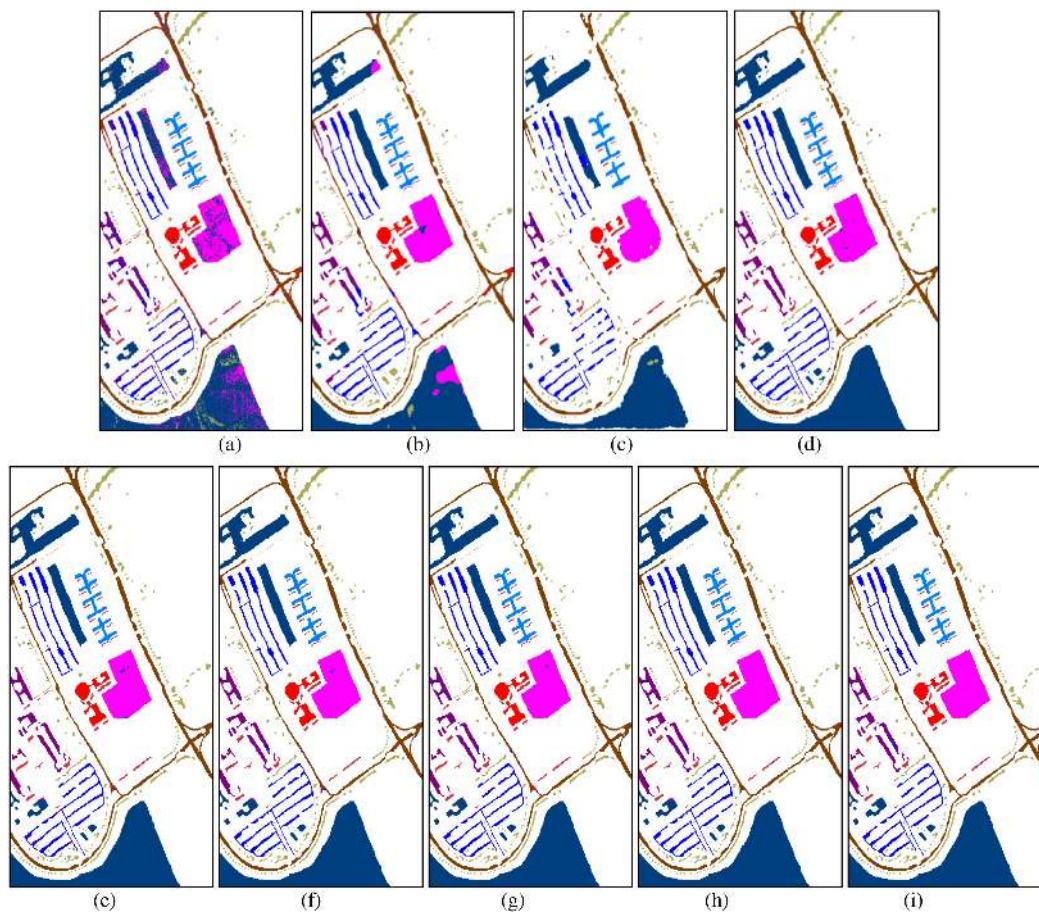


Fig. 11. Classification results of the University of Pavia with compared methods. (a) SVM. (b) EPF. (c) LCMV. (d) MFASR. (e) 2DCNN. (f) 2D+D. (g) 2D+3D. (h) 2D-3D-D-S. (i) 2D-3D-D.

TABLE IX  
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF UNIVERSITY OF PAVIA WITH ALL THE COMPARED METHODS (5%)

Class	SVM	EPF	LCMV	MFASR	2D	2D+D	2D+3D	2D-3D-D-S	2D-3D-D
1	93.75	90.47	77.24	99.68	98.86±0.33	98.90±0.29	99.45±0.08	98.79±0.90	99.42±0.05
2	94.03	91.95	86.42	99.90	99.49±0.43	99.89±0.03	99.92±0.10	99.91±0.06	<b>99.93±0.24</b>
3	67.41	87.18	73.34	99.70	97.57±1.41	96.98±0.88	97.67±1.51	97.60±1.96	98.69±0.32
4	76.52	98.37	79.85	90.24	99.50±0.81	99.86±0.05	99.77±0.31	99.81±0.18	<b>99.88±0.16</b>
5	97.68	99.93	98.81	99.61	99.96±0.03	99.88±0.16	99.91±0.09	99.75±0.20	<b>99.97±0.04</b>
6	62.90	97.93	91.49	98.12	98.58±1.46	99.26±0.11	<b>99.55±0.25</b>	99.18±0.20	99.45±0.32
7	57.29	100	89.10	99.76	95.73±3.05	97.38±1.66	98.12±1.42	99.47±0.32	99.47±0.45
8	83.47	90.77	81.10	99.20	96.70±0.87	97.45±0.53	97.47±0.67	97.48±0.67	97.89±0.58
9	99.86	100	78.46	99.11	99.79±0.15	99.50±0.54	99.75±0.14	99.68±0.31	<b>99.96±0.09</b>
POA	84.13	93.23	84.32	98.87	98.87±0.22	99.22±0.04	99.41±0.14	99.29±0.27	<b>99.54±0.11</b>

of class No. 2, 4, 5, and 9 especially, which are 99.93%, 99.88%, 99.97%, and 99.96%.

Since the parameters play a significant role in the 3-D part of the model, for fair comparison with the same number of neurons of fully connection layers, a series of experiments are conducted for all the four datasets to show the difference of the proposed 2D-3D-D-S and 2D-3D-D methods. All the classification results by the average of 5-times execution are listed in Table X, where the “FC-A” denotes the number of FC1 and FC2 is 200 and 150, the “FC-B” describes the circumstance with the number of FC1 and FC2 is 600 and 150, respectively. In the last row, the OA with

FC-A is illustrated in the color of light orange and the OA with FC-B is performed in the color of light green. It can be observed that the 2D-3D-D-S has better performance with all the data sets in the situation of FC-A, while 2D-3D-D is more outstanding in the FC-B environment. Due to the fact listed below, all the results are reasonable and plausible. For one thing, the 2D-3D-D is adopted to learn the feature with numerous parameters, which is a disadvantage to the classification performance due to the fewer kernels in the fully connection layers. For the other thing, the complex model performs better and reveals that 2D-3D-D extracted more efficient feature maps with more parameters.

TABLE X  
OA CALCULATED OF THE FOUR DATASETS WITH THE PROPOSED METHOD WITH DIFFERENT FC PARAMETERS

Class P <sub>OA</sub> %	Purdue				Salinas				KSC				Pavia			
	2D-3D-D-s		2D-3D-D		2D-3D-D-s		2D-3D-D		2D-3D-D-s		2D-3D-D		2D-3D-D-s		2D-3D-D	
	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B	FC-A	FC-B
1	99.55	96.96	98.26	100	99.43	93.13	98.24	99.81	97.06	99.33	99.01	98.19	98.79	98.06	97.78	99.42
2	97.61	93.36	95.90	98.36	99.69	99.78	98.99	99.65	95.52	92.33	95.30	96.75	99.91	99.90	99.94	99.93
3	96.62	96.46	94.07	97.80	99.50	99.89	99.55	99.75	97.33	97.70	97.04	98.31	97.60	97.08	94.44	98.69
4	98.75	98.06	98.14	97.20	99.06	99.64	99.49	99.37	89.79	92.91	91.29	87.50	99.81	99.03	98.28	99.88
5	99.67	94.08	94.66	99.30	99.25	96.97	99.34	98.68	84.59	94.24	93.33	94.10	99.75	99.99	99.96	99.97
6	99.34	99.07	99.43	99.07	99.87	100	99.99	99.99	99.61	98.60	99.35	98.73	99.18	99.64	99.54	99.45
7	100	97.14	95.00	100	99.60	99.77	99.62	99.88	94.82	99.09	97.95	95.87	99.47	98.81	97.14	99.47
8	99.46	100	100	99.83	97.17	97.80	95.57	98.05	98.29	93.85	95.96	96.32	97.48	98.04	98.27	97.89
9	95.42	90.00	98.00	92.72	99.82	99.97	99.56	99.80	97.44	94.90	95.42	97.34	99.68	99.93	99.64	99.96
10	97.35	97.41	98.29	97.34	98.85	99.55	99.55	99.86	97.89	95.43	96.00	97.35	-	-	-	-
11	97.73	97.35	98.57	98.23	96.61	99.76	99.94	98.67	99.88	97.87	98.72	99.48	-	-	-	-
12	96.79	92.58	92.92	97.66	98.83	99.74	99.22	99.92	97.19	96.73	96.69	98.90	-	-	-	-
13	99.42	99.22	99.41	99.32	98.96	99.67	99.72	99.89	99.55	99.51	99.30	99.70	-	-	-	-
14	98.71	99.70	99.90	99.01	98.72	99.42	99.68	99.40	-	-	-	-	-	-	-	-
15	98.23	98.45	99.53	98.60	93.52	91.32	92.30	97.76	-	-	-	-	-	-	-	-
16	94.92	100	98.28	92.59	99.72	99.62	98.40	99.88	-	-	-	-	-	-	-	-
POA	97.98	96.94	97.62	98.33	98.07	97.85	97.65	99.07	97.14	96.77	97.04	97.47	99.29	99.20	98.94	99.54

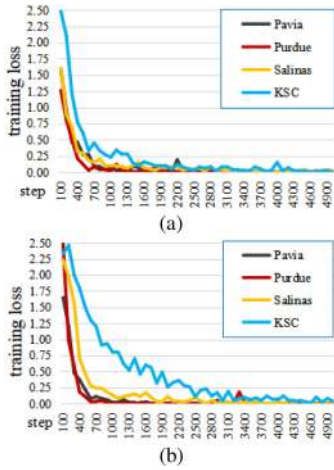


Fig. 12. Training loss of the proposed CNN framework. (a) 2D-3D-D-S. (b) 2D-3D-D.

That is to say, the 2D-3D-D-S in FC-A outperform the 2D-3D-D model, while the 2D-3D-D model is more challenging in the condition of the FC-B.

In order to fully demonstrate all the aspects of the proposed network, we also recorded the convergence curves and the changing trend of the training loss of the four datasets with the above parameters. The loss curve is shown in Fig. 12, respectively, the vertical axis reflects the evolution of the error, the number of the iteration is listed on the horizontal axis, and the shadow represents the stand deviation of the curve fitting. It can be concluded there is a very little difference in the curve between the Purdue Indiana Indian Pines Scene and the KSC data, the loss values tend to be zero after 2000 iteration demonstrates that the proposed CNN model converges well, and the reasonable and similar training loss curves also show that our method is effective with the RMSprop optimization. As mentioned, we chose 5000 times as the final number of training iterations to obtain better precision results in the experiments. In the training phase, the training loss is feedback to the machine for back

TABLE XI  
TRAINING TIME OF THE ABOVE CNN FRAMEWORKS (SECONDS)

Time (Seconds)	CNNS (4 layers)	CNNT (6layers)	CNND (7 layers)	2D-3D-D-S (7 layers)	2D-3D-D (7 layers)
Purdue	363.20	465.60	671.87	317.19	505.88
Salinas	679.16	786.44	995.03	964.67	1497.08
KSC	396.22	430.92	654.7	657.67	1288.15
Pavia	354.58	415.78	632.74	455.23	807.06

propagation, which reflects the training state of the network. The 2D-3D-D model used 3-D kernel for the spectral–spatial feature extraction, which generated more spatial redundancy than the 2D-3D-D-S model especially for the sparse data. Due to fact that the KSC data is too sparse as shown in the ground truth image, it is more sensitive to the feature representation ability, therefore, the convergence of the curve of KSC data is slower than other datasets as shown in Fig. 12(b) for the redundancy caused by the 2D-3D-D architecture.

The training time of the proposed model with a comparison of the other CNN networks on the three hyperspectral data sets is listed in Table XI. The computational cost is related to the size of the image, the numbers of the total layers, and the training sample, we can observe that CNND [41] has the longest time cost for the Salinas data, while the Purdue data have the smallest training time for our proposed framework. Also, it can be seen that with the same seven layers, our proposed 2D-3D-D-S model is quicker than the CNN model for the Purdue data, Pavia data, and Salinas data, and has quite the same cost on the KSC data, the 2D-3D-D network has more cost than the 2D-3D-D-S due to the 3-D kernel in 2D-3D-D is more complex than the 2D-3D-D-S model.

#### D. Comparison With the State-of-the-Art CNN Architectures

To demonstrate the effectiveness of the proposed framework, we compared the 2D-3D-D model with the other five state-of-the-art CNN models on the Purdue Indiana Indian Pines Scene. The approaches included in the comparison are summarized

TABLE XII  
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF THE PURDUE INDIANA INDIAN PINES SCENE WITH THE COMPARED METHODS

Class POA%	CNN-MR F (40 per class)	3-D Auto-CN N-Cutout (5%)	SRCL (98%)	SS-3DCNN (2%)	3D SP-DNNs (1%)	3D-CNN (50%)	FL-CNN (200 per class)	3-D-CNN (70%)	DC-CNN (2 per class)	2D-3D-D-S (10%)	2D-3D-D (10%)
1	90.24	33.14	89.58	97.96	100	95.89	80.43	85.71	67.44	99.55	100
2	96.96	88.28	97.62	96.49	91.11	98.46	95.92	96.46	96.39	97.61	98.36
3	98.26	79.86	97.78	99.53	85.78	98.99	98.22	97.13	98.10	96.62	97.80
4	96.71	56.57	81.91	97.47	24.89	99.14	93.34	98.55	97.78	98.75	97.20
5	97.24	71.13	98.51	97.21	81.57	99.29	97.52	97.90	94.32	99.67	99.30
6	99.24	95.58	98.76	95.24	89.59	99.92	98.89	97.68	98.85	99.34	99.07
7	92.00	48.18	71.02	98.88	100	100	95.63	100	100	100	100
8	99.77	99.77	100.0	98.51	66.32	100	99.86	99.30	100	99.46	99.83
9	0	20.00	3.7	97.73	100	92.31	68.33	100	100	95.42	92.72
10	98.40	91.74	98.55	94.44	77.88	98.12	95.66	98.26	90.25	97.35	97.34
11	98.05	93.70	98.54	97.83	88.43	98.96	97.47	98.77	97.04	97.73	98.23
12	98.31	73.70	94.57	97.70	66.95	98.99	97.86	97.15	92.18	96.79	97.66
13	100	94.75	99.65	97.24	99.51	99.82	98.76	96.72	98.45	99.42	99.32
14	96.49	98.20	99.00	96.47	99.37	99.81	98.46	99.46	99.50	98.71	99.01
15	100	50.99	98.74	95.81	83.68	99.56	89.21	93.80	98.63	98.23	98.60
16	86.75	70.57	94.12	99.83	98.92	99.38	98.21	100	97.73	94.92	92.59
POA	97.10	89.01	97.58	97.89	85.20	<b>99.07</b>	96.99	97.31	96.55	97.98	98.33
AA	90.53	72.88	88.88	97.39	84.63	98.66	93.99	98.92	95.42	98.09	97.94

as follows. CNN-MRF model [22] adopted CNN to learn the posterior class distributions for better spatial information extraction, which included two convolution layers with the kernel size of  $5 \times 5$  and  $3 \times 3$ , the numbers of the kernel of two fully connected layers are set to 200 and 100. The learning rate is 0.001 and the batch size is set as 100. The SS-3DCNN [29] is a semi-supervised 3-D CNN for deep feature extraction, in the model, the size of the input sample is  $25 \times 25 \times 12$ , the learning rate is 0.005 and the training epoch is set to 150, respectively. In SRCL [21], Hao *et al.* proposed deep network architecture for a super-resolution aided HSIC with class-wise loss, the size of the convolution kernel is set to  $9 \times 9$ ,  $1 \times 1$ , and  $5 \times 5$ , respectively, the filter size of the pooling layer is fixed to  $2 \times 2$ . In [27], the 3-D auto model, the author designed 1-D Auto-CNN and 3-D Auto-CNN for spectral and spectral-spatial HSI classifiers,  $3 \times 3$  and  $5 \times 5$  separable convolution,  $3 \times 3$  and  $5 \times 5$  dilated separable convolution,  $3 \times 3$  average pooling and  $3 \times 3$  max-pooling are used in the 3-D auto model. The training epoch for the 1-D Auto-CNN and 3-D Auto-CNN-Cutout is set to 300 and 100. In 3-D SP DNN [26], the model exploits the 3-D spectral-spatial information via super pixel-based neural networks. The sizes of filter are set  $4 \times 4 \times 63$  and  $3 \times 3 \times 62$ , the numbers of the filter are set to 6 and 12, the kernel size is set to  $2 \times 2 \times 2$  for the max-pooling process. The 3-D CNN [42] model is composed of two 3-D convolution blocks (C1 and C2) followed by a fully connection layer (F1), the size of the 3-D convolutional kernel of C1 is  $3 \times 3 \times 7$ , the number is set to 2, and the size of kernel is  $3 \times 3 \times 3$ , the number of filters is 4 for C2, respectively. FLCNN [43] is designed to learn sensor-specific spatial-spectral features with five-layers CNN, which can be implemented the HSIC in both unsupervised and supervised way, the size of input sample is  $3 \times 3$  and  $5 \times 5$ , the number of kernel is 20 in the convolution layer and the number of sigmoid neuron nodes in the fully connection layer is set to 100. A 3-D CNN network [44] captured the spatial and spectral context, which is consisted of seven convolution layers with 3-D kernel and one full connection layer, the kernel sizes of the 3-D operation included  $3 \times 3 \times 3$ ,  $1 \times 1 \times 3$ , and  $1 \times 1 \times 2$  specifically.

In [45], the DC-CNN utilized 2-D CNN the hierarchical feature extraction, the 1-D CNN channel and 2D channel extract the features with the 1-D and 2-D convolution operation. The 1-D unit contains two pairs of convolutional layers, it has 300 and 2 convolutional kernels for the sequential convolutional layer, the 2-D CNN contains two pairs of convolutional layers, the size of the convolutional layer is fixed to  $3 \times 3$ . Tables XII–XV expressed the comparison of classification accuracy obtained by the CNN models for comparison, we compare the detailed classification performance on each class for the four datasets. As can be seen, for the Purdue Indiana Indian Pines Scene data, the 3-D CNN method achieves the highest classification OA than the other models, however, we investigate that the 3-D CNN acquired OA of 99.07% with the training percentage of 70%, while the proposed 2D-3D-D yields OA of 98.33% with only 10% training sample. For fair comparison, we have done the experiment on the KSC dataset with the same ratio, the classification results are listed in the last column in Table XV, it can be seen that the better performance of our model generates the OA of 99.95%. For the Salinas valley and the University of Pavia, the proposed framework generates the best classification performance as shown in Tables XIII and XIV, and Table XV shows that the proposed CNN almost get the best OA with 40% training sample for the Kennedy Space Center. Furthermore, to evaluate the performance of our network objectively, we also have done a series of the experiment of the classification without sample augmentation for all the four datasets, each execution of the CNN network has been repeated 3 times in this part, and we reported the accuracy averaged by the results in Table XVI. It is easy to be observed that the 2D-3D-D is effective without data expansion and the OA is stable for all the datasets. To sum up, from the above results and analysis, we can conclude that the proposed architecture achieves better classification performance compared with state-of-the-art approaches.

#### E. Comparison of the Different Size of the Training Sample

In this part, the impact with different training size of the three datasets is evaluated with the proposed 2D-3D-D CNN network.



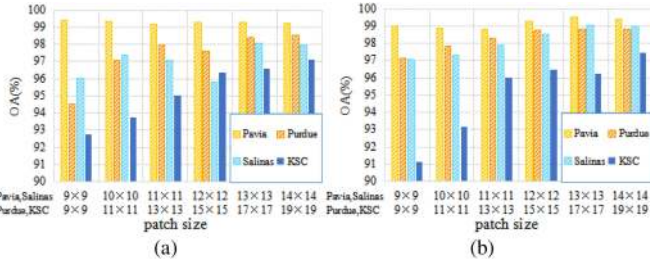


Fig. 13. OA (%) obtained by the proposed network with the different patch size of training samples on three data sets. (a) 2D-3D-D-S. (b) 2D-3D-D.

For the Purdue Indiana Indian Pines Scene and the KSC training sets, the patch sizes of input data are set to  $9 \times 9$ ,  $11 \times 11$ ,  $13 \times 13$ ,  $15 \times 15$ ,  $17 \times 17$ ,  $19 \times 19$ , respectively, and we select  $9 \times 9$ ,  $10 \times 10$ ,  $11 \times 11$ ,  $12 \times 12$ , and  $13 \times 13$ ,  $14 \times 14$  as the size of the input sample for the Salinas data set and the Pavia data. The dropout rate for all the three datasets is fixed to 0.3, and the percentage of the Purdue and KSC data is set to 10%, and the portion for the Salinas data is 5%, Fig. 13 shows the OA expression with the histogram obtained by our proposed network on three datasets. It can be observed that OA varies as the size of the training sample varies. The accuracy of the three datasets generally shows an increasing trend with the size of the space increase, among them, the KSC dataset has the most obvious promotion with the size is enlarged. As shown in Fig. 13(a), it can be observed that the best results are generated with the size of  $19 \times 19$  for the Purdue and KSC datasets in the two 2D-3D-D models. When the sample size is set to  $9 \times 9$ , the lowest value of OA is 94.53% for the Purdue data with the simplified kernel, 96.02% for the Salinas data, and 92.79% for the KSC data separately. For the 2D-3D-D model, the lowest OA value is 97.15%, 97.08%, and 91.12% for the Purdue data, the Salinas data, and the KSC data with the smallest size of the training patch, while the size is  $13 \times 13$ , the proposed model yields the lowest OA is 98.84%. To be noticed, the larger size of the space contributes to the extraction of spatial features of the dataset, and the overall accuracy tends to be stable after  $17 \times 17$ .

#### F. Effect of Different Numbers of Training Samples

Next, we explore the effect of the different percentages of the training sample with the proposed CNN network in this part. For the Purdue and KSC data, 1% to 10% with 1% interval of the total sample is set in this experiment, for the Salinas image, the portion changes from 1% to 5% with the 0.5% step. The dropout rate of the three sets are all set to 0.3, the sample size of the Purdue and KSC datasets is fixed as  $19 \times 19$ , and the sample size of the Salinas data is  $13 \times 13$ . Fig. 14 presents the average OA results of the different portions of the training sample which are executed five times separately. It can be seen that the proposed CNN network generates robust performance even the percentage is set to 1%. The performances of the Purdue and KSC generally improve with the increase of sample percentage, while for the Salinas Valley data, the OA is more stable because

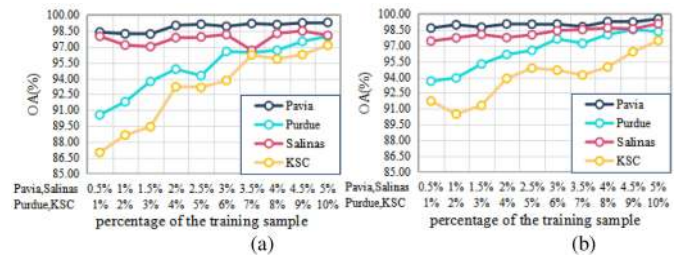


Fig. 14. OA (%) obtained by the proposed network with different percentages of training samples on three data sets. (a) 2D-3D-D-S. (b) 2D-3D-D.

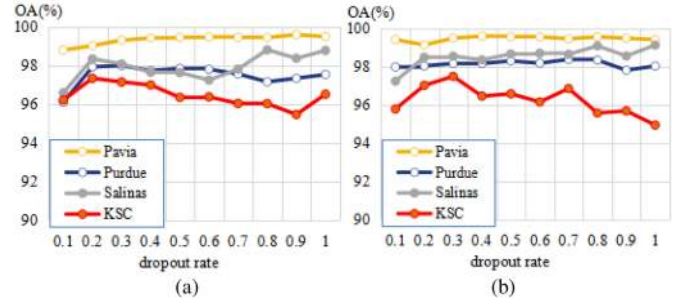


Fig. 15. OA (%) obtained by the 2D-3D-D network with the different dropout rates of training samples on three data sets. (a) 2D-3D-D-S. (b) 2D-3D-D.

of the characteristic of the rich spatial information. Especially, in the simplified 2D-3D-D model, the lowest OA is 96.65% when the percentage is 3.5% for the Salinas dataset, and the best OA can reach 97.98% and 97.14% when the percentage is 10% for the Purdue and the KSC data specific. The 2D-3D-D network generates the best OA of 98.51% with the ratio as 9% for the Purdue data, the more percentage brings the better performance for the rest three datasets with the value of 99.07%, 97.47%, and 99.54%, for the Salinas data, the KSC data, and the Pavia data.

#### G. Comparison of the Dropout Rate

Lastly, the effectiveness of the setting of the dropout rate is analyzed in this section. The proposed 2D-3D-D networks are compared with different dropout percentages. The size and the percentage of the training sample Purdue data and the KSC data is set to  $19 \times 19$  and 10%, and for Salinas Valley, they are set to  $13 \times 13$  and 5%. OA obtained by the 2D-3D-D network with different dropout rates on three datasets is demonstrated in Fig. 15, it can be observed that OA varies with the different dropout rate in the 2D-3D-D-S, it leads the best OA for the Purdue data to be 97.98% when the rate is 0.3, the best OA of Salinas data is 98.80% when the value is set to 0.8, and for the KSC data, the OA of 97.32% when the dropout rate is 0.2. On the contrary, the proposed framework generates the lowest value for Purdue data when the rate is 0.8, for the Salinas data with the rate of 0.6, and for the KSC data when the drop rate is 0.9, it generates the worst OA. The best OA in the 2D-3D-D model for the Purdue data and the Salinas data are 98.33% and 99.07% when the rate is 0.8, and when the dropout rate is 0.3, the model generates the best OA of 97.48%, and the best OA is 99.58% for

TABLE XVII

OA CALCULATED FROM THE CLASSIFICATION RESULTS OF PURDUE INDIANA INDIAN PINES SCENE WITH DIFFERENT MODULES OF THE PROPOSED METHOD WITH DIFFERENT MODULES

Class P <sub>OA</sub> %	2D-3D-D-S			2D-3D-D		
	One Module	Two Modules	Three Modules	One Module	Two Modules	Three Modules
1	99.55	100	1	100	100	100
2	97.61	97.56	96.78	98.36	98.92	95.04
3	96.62	94.99	96.68	97.80	96.75	93.69
4	98.75	98.71	98.33	97.20	97.53	99.16
5	99.67	98.33	97.15	99.30	99.18	96.96
6	99.34	98.64	99.31	99.07	99.45	95.30
7	100	100	96.43	100	100	100
8	99.46	100	99.79	99.83	100	100
9	95.42	95.24	86.96	92.72	94.74	76.98
10	97.35	95.53	96.02	97.34	96.35	97.91
11	97.73	96.10	98.16	98.23	97.75	97.06
12	96.79	94.79	91.97	97.66	96.74	92.80
13	99.42	98.56	100	99.32	100	100
14	98.71	98.44	98.90	99.01	99.68	95.79
15	98.23	95.08	96.73	98.60	98.96	94.18
16	94.92	93.81	97.87	92.59	94.68	91.75
POA	97.98	96.91	97.43	98.33	98.23	96.11
AA	98.10	97.24	90.76	97.94	98.17	95.41

TABLE XVIII

OA CALCULATED FROM THE CLASSIFICATION RESULTS OF THE SALINAS VALLEY WITH DIFFERENT MODULES OF THE PROPOSED METHOD WITH DIFFERENT MODULES

Class P <sub>OA</sub> %	2D-3D-D			2D-3D-D		
	One Module	Two Modules	Three Modules	One Module	Two Modules	Three Modules
1	99.43	99.96	99.00	99.81	96.81	100
2	99.69	99.78	99.52	99.65	99.05	98.65
3	99.50	98.21	98.85	99.75	99.95	100
4	99.06	99.00	95.22	99.37	99.35	96.94
5	99.25	99.45	98.66	98.68	98.40	99.58
6	99.87	99.97	99.86	99.99	100	99.97
7	99.60	99.38	99.78	99.88	99.52	99.25
8	97.17	95.01	97.54	98.05	96.28	96.35
9	99.82	99.49	99.86	99.80	99.37	99.60
10	98.85	99.35	98.60	99.86	99.51	99.75
11	96.61	97.83	97.13	98.67	96.21	93.36
12	98.83	99.80	99.23	99.92	98.77	97.32
13	98.96	99.62	99.60	99.89	100	99.13
14	98.72	99.02	98.71	99.40	99.26	99.91
15	93.52	95.86	87.03	97.76	98.43	90.82
16	99.72	99.05	99.50	99.88	98.62	99.21
POA	98.07	97.99	96.84	99.07	98.40	97.39
AA	98.66	98.89	98.01	99.40	98.72	98.12

TABLE XIX

OA CALCULATED FROM THE CLASSIFICATION RESULTS OF KENNEDY SPACE CENTER WITH DIFFERENT MODULES OF THE PROPOSED METHOD WITH DIFFERENT MODULES

Class P <sub>OA</sub> %	2D-3D-D-S			2D-3D-D		
	One Module	Two Modules	Three Modules	One Module	Two Modules	Three Modules
1	97.06	89.74	88.14	98.19	96.57	95.84
2	95.52	98.24	93.09	96.75	93.53	97.77
3	97.33	96.88	98.15	98.31	96.36	98.07
4	89.79	91.60	89.50	87.50	84.53	79.73
5	84.59	85.39	79.12	94.10	89.44	85.14
6	99.61	97.24	96.97	98.73	98.10	99.07
7	94.82	86.87	93.06	95.87	100	80.95
8	98.29	97.76	93.19	96.32	90.85	92.44
9	97.44	91.95	91.18	97.34	99.41	99.60
10	97.89	94.87	95.30	97.35	91.69	93.80
11	99.88	97.19	98.53	99.48	99.29	96.77
12	97.19	93.68	96.25	98.90	97.62	95.56
13	99.55	98.10	96.56	99.70	99.25	97.17
POA	97.14	93.84	93.32	97.47	95.85	94.68
AA	96.07	93.81	93.00	96.81	95.13	93.22

TABLE XX

OA CALCULATED FROM THE CLASSIFICATION RESULTS OF UNIVERSITY OF PAVIA WITH DIFFERENT MODULES OF THE PROPOSED METHOD WITH DIFFERENT MODULES

Class P <sub>OA</sub> %	2D-3D-D-S			2D-3D-D		
	One Module	Two Modules	Three Modules	One Module	Two Modules	Three Modules
1	98.79	98.81	97.91	99.42	98.16	95.20
2	99.91	99.81	99.27	99.93	99.79	98.91
3	97.6	91.29	91.72	98.69	93.18	79.29
4	99.81	99.51	99.50	99.88	100	97.33
5	99.75	98.66	99.19	99.97	99.70	99.85
6	99.18	98.91	98.72	99.45	99.66	98.99
7	99.47	94.90	94.56	99.47	98.41	86.72
8	97.48	95.12	94.72	97.89	92.93	92.45
9	99.68	98.77	99.32	99.96	99.89	99.05
POA	99.29	98.45	98.07	99.54	98.56	96.36
AA	99.074	97.31	97.21	99.41	97.97	94.20

TABLE XXI

TRAINING TIME OF THE PROPOSED CNN FRAMEWORKS WITH DIFFERENT MODULES OF THE FOUR DATA SETS (SECONDS)

Time (Seconds)	2D-3D-D-S			2D-3D-D		
	One module	Two modules	Three modules	One module	Two modules	Three modules
Purdue	270	305	352	530	629	681
Salinas	352	471	589	557	704	892
KSC	474	564	668	980	1257	1440
Pavia	192	375	358	354	518	669

the Pavia data with a rate of 0.4. It also can be concluded that the dropout rate will not always improve the performance of the training procedure. We also noted that, due to the simplification of the model, the proposed model can achieve relatively great overall accuracy with the bigger dropout rates.

*H. Analysis of Number of the Modules*

To further analyze the influence of the number of the 2D-3D-D module, we complete a series of experiments to compare the classification performance and the running time of the proposed method with different modules, the programming environment is Nvidia Geforce GTX 1060Ti, RAM 32.0 GB for the four datasets. In this section, one 2-D block and one 3-D block is described as one module. To conduct a fair evaluation, the experiments are implemented on all the four datasets and Tables XVII–XX list the classification accuracy of each class.

According to the reported results in Tables XVII–XX, the proposed networks with different modules achieve stable performance for every dataset. Table XXI shows the training time of the proposed frameworks (both simplified 3-D kernel and 3-D kernel) with three different modules separately, and it can be seen that the more modules consume more time cost. Moreover, as can be seen in Figs. 16 and 17, with the same setting of the hyperparameters, the proposed architecture with one module obtains the best OA value with the four datasets. It can be concluded that the proposed CNN network with one module is superior to the other modules in terms of classification performance and quantitative time cost.

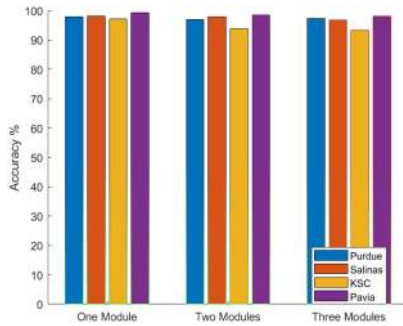


Fig. 16. OA (%) obtained by the proposed 2D-3D-D-S networks with different modules of the four datasets.

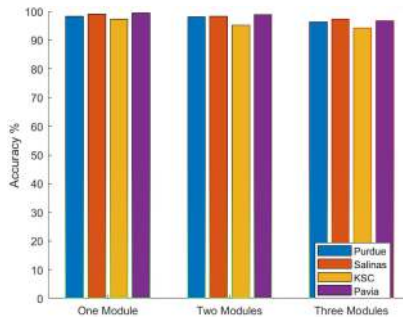


Fig. 17. OA (%) obtained by the proposed 2D-3D-D networks with different modules of the four datasets.

## V. CONCLUSION

In this study, we propose a new network model for hyperspectral image classification based on the cooperation between 2-D CNN and simplified 3-D convolution layer. The 2-D CNN part focuses on the extraction of rich spectral-spatial features from the available HSI. Subsequently, the 3-D block mainly deals with the reconstruction of the refined spectral feature with neighbor bands information involved. In this way, the proposed model can generate the fusion refined feature to enhance the representation and description of the deep map. The key role of this framework is that the 3-D block depends on only one convolution kernel with a size of  $1 \times 1 \times L$  to increase the convolution speed and overcome the fitting problem. The proposed model explored an implementation with the depth-wise separable convolution way for HSIC for the first time. The real hyperspectral image experimental results proved that the proposed architecture achieves a great performance on the four popular testing datasets. Notice that the band information is crucial for feature refinement in the CNN network for HSIC, one of our future works will focus on design band selection and augmentation network to extract contextual feature hierarchically.

## REFERENCES

- [1] A. Plaza *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, no. 1, pp. S110–S122, 2009.
- [2] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.
- [3] Y. Zhao *et al.*, "Application of hyperspectral imaging and chemometrics for variety classification of maize seeds," *Rsc Adv.*, vol. 8, no. 3, pp. 1337–1345, 2018.
- [4] 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.
- [5] H. Yu, L. Gao, W. Li, Q. Du, and B. Zhang, "Locality sensitive discriminant analysis for group sparse representation-based hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1358–1362, Aug. 2017.
- [6] H. Yu, L. Gao, W. Liao, B. Zhang, A. Pižurica, and W. Philips, "Multiscale superpixel-level subspace-based support vector machines for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2142–2146, Nov. 2017.
- [7] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.
- [8] N. Divakar and R. V. Babu, "Image denoising via CNNs: An adversarial approach," in *Proc. Comput. Vision Pattern Recognit. Workshops*, 2017, pp. 1076–1083.
- [9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [10] M. Nan *et al.*, "Salient object detection using a covariance-based CNN model in low-contrast images," *Neural Comput. Appl.*, vol. 29, no. 8, pp. 181–192, 2018.
- [11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [12] H. C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1–17, May 2016.
- [13] Z. Yan *et al.*, "HD-CNN: Hierarchical deep convolutional neural network for image classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015.
- [14] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.
- [15] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018.
- [16] N. He, M. E. Paoletti, J. M. Haut, S. Li, A. Plaza, and J. Plaza, "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 755–769, Feb. 2019.
- [17] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Art. no. 258619.
- [18] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [19] F. Ratle, G. Camps-Valls, and J. Weston, "Semi supervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, May 2010.
- [20] L. Fang, G. Liu, S. Li, P. Ghamisi, and J. A. Benediktsson, "Hyperspectral image classification with squeeze multibias network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1291–1301, Mar. 2019.
- [21] S. Hao, W. Wang, and Y. Ye, "A deep network architecture for super-resolution-aided hyperspectral image classification with classwise loss," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4650–4663, Aug. 2018.
- [22] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image segmentation with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.
- [23] Y. Li, W. Xie, and H. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," *Pattern Recognit.*, vol. 63, pp. 371–383, 2017.
- [24] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.
- [25] J. Zhu, J. Hu, S. Jia, X. Jia, and Q. Li, "Multiple 3-D feature fusion framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1873–1886, Apr. 2018.
- [26] C. Shi and C. Pun, "Superpixel-based 3D deep neural networks for hyperspectral image classification," *Pattern Recognit.*, vol. 74, pp. 600–616, 2018.



- [27] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Design of convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, Sep. 2019.
- [28] J. Feng, H. Yu, L. Wang, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial-spectral generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5329–5343, Aug. 2019.
- [29] A. Sellami, M. Farah, I. R. Farah, and B. Solaiman, "Hyperspectral imagery classification based on semi-supervised 3-D deep neural network and adaptive band selection," *Expert Syst. Appl.*, vol. 129, pp. 246–259, 2019.
- [30] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.
- [31] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.
- [32] C. Shi and C. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, pp. 82–93, 2018.
- [33] Q. Xu, Y. Xiao, D. Wang, and B. Luo, "CSA-MSO3DCNN multiscale octave 3D CNN with channel and spatial attention for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 1, pp. 1–24, 2020.
- [34] 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.
- [35] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.
- [36] Y. N. Dauphin, H. D. Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1504–1512.
- [37] M. Adankon and M. Cheriet, "Support vector machine," *Comput. Sci.*, vol. 1, no. 4, pp. 1–28, 2002.
- [38] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [39] C. Yu, Y. Wang, M. Song, and C.-I. Chang, "Class signature-constrained background-suppressed approach to band selection for classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 14–31, Jan. 2019.
- [40] L. Fang, W. Cheng, S. Li, and J. A. Benediktsson, "Hyperspectral image classification via multiple-feature-based adaptive sparse representation," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1646–1657, Jul. 2017.
- [41] C. Yu *et al.*, "Hyperspectral image classification method based on CNN architecture embedding with hashing semantic feature," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1866–1881, Jun. 2019.
- [42] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, pp. 67–88, 2017.
- [43] 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.
- [44] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.
- [45] C. Chen, J. Zhang, C. Zheng, Q. Yan, and L. Xun, "Classification of hyperspectral data using a multi-channel convolutional neural network," in *Proc. Int. Conf. Intell. Comput.*, 2018, pp. 81–92.