Deep Multilayer Fusion Dense Network for Hyperspectral Image Classification

Zhaokui Li, Tianning Wang, Wei Li, *Senior Member, IEEE*, Qian Du, *Fellow, IEEE*, Chuanyun Wang, Cuiwei Liu, and Xiangbin Shi

***Abstract*—Deep spectral–spatial features fusion has become a research focus in hyperspectral image (HSI) classification. However, how to extract more robust spectral–spatial features is still a challenging problem. In this article, a novel deep multilayer fusion dense network (MFDN) is proposed to improve the performance of HSI classification. The proposed MFDN simultaneously extracts the spatial and spectral features based on different sample input sizes, which can extract abundant spectral and spatial correlation information. First, the principal component analysis algorithm is performed on hyperspectral data to extract low-dimensional HSI data, and then the spatial features are extracted from the low-dimensional 3-D HSI data through 2-D convolutional, 2-D dense block, and average-pooling layers. Second, the spectral features are extracted directly from the raw 3-D HSI data by means of 3-D convolutional, 3-D dense block, and average-pooling layers. Third, the spatial and spectral features are fused together through 3-D convolutional, 3-D dense block, and average-pooling layers. Finally, the fused spectral–spatial features are sent into two full connection layers to extract high-level abstract features. Furthermore, densely connected structures can help alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and improve the HSI classification accuracy. The proposed fusion network outperforms the other state-of-the-art methods especially with a small number of labeled samples. Experimental results demonstrate that it can achieve outstanding hyperspectral classification performance.**

***Index Terms*—Deep learning, densely connected convolutional neural network, hyperspectral image (HSI) classification, multilayer feature fusion.**

I. Introduction

HYPERSPECTRAL sensors can capture hundreds of narrow spectral channels with very high spectral resolution. With the abundant spatial and spectral information, hyperspectral images (HSIs) have been applied in many fields, such as military [1], agriculture [2], and environment monitoring [3].

Due to the complex characteristics of HSI data, HSI classification is still very challenging. During the last decade, the HSI classification method based on spectral features [4]–[6] has become a very active research topic in the remote sensing. However, the large number of spectral bands may bring noise to HSI, and the high dimensionality of HSI may produce the Hughes phenomenon [7]. Therefore, using spectral features directly may not be suitable for HSI classification tasks [8].

To further improve the classification performance, many classification frameworks based on spectral–spatial features have been proposed [9] recently. Benediktsson *et al.* [10] utilized multiple morphological operations to construct spectral–spatial features of HSIs. Khodadadzadeh *et al.* [11] proposed a spectral–spatial classifier for HSI classification that addresses the issue of mixed pixel characterization. In [12], multiple kernel learning based on spectral–spatial information is designed to improve the SVM classifier.

More recently, many studies have shown that HSI classification framework based on deep spectral–spatial features can deliver state-of-the-art results. In [13], spatial features extracted by CNN were integrated with spectral features obtained by balanced local discriminate embedding to finish HSI classification. Li *et al.* [14] proposed a CNN-based feature extractor by learning discriminative representations from pixel pairs. Li *et al.* [15], [16] proposed a deep network based on multiscale spectral–spatial fusion for HSI classification. Yang *et al.* [17] designed a Two-CNN model to learn the spectral features and spatial features jointly. But in this framework, the input of spectral data is a one-dimensional (1-D) dimension, which leads to the lack of neighborhood information in the spatial dimension. And the classification accuracy of these deep learning models will decrease when the network is deeper. In addition, 3D-CNN was used to directly extract deep spectral–spatial features from raw HSIs, and provided promising classification results [18]. Li *et al.* [19] further studied 3D-CNN for spectral–spatial classification using input cubes of HSIs with a smaller spatial size. These models generate thematic maps using an approach that can directly process the raw HSIs, whereas the classification accuracy of the CNN models decreases as the network gets deeper. Song *et al.* [20] proposed a deep fusion feature network for classification. In this network, the features from the lower layers, intermediate layers, and higher layers are, respectively, extracted

Manuscript received August 14, 2019; revised December 3, 2019; accepted March 18, 2020. Date of publication March 30, 2020; date of current version April 16, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61922013, 61902339, and 61703287, in part by the Liaoning Provincial Natural Science Foundation of China under Grants 20180550337, 2019-MS-254 and 20180550664, in part by the Foundation of Liaoning Educational Committee under Grant JYT19029, in part by the Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data in Yanan University under Grant IPBED14, and in part by the Doctoral Starting up Foundation of Yan'an University under Grant No. YDBK2019-06. *(Corresponding author: Zhaokui Li.)*

Zhaokui Li, Tianning Wang, Chuanyun Wang, Cuiwei Liu, and Xiangbin Shi are with the School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China (e-mail: lzk@sau.edu.cn; 1723060570011@email.sau.edu.cn; wangcy0301@sau.edu.cn; liucuiwei@sau.edu.cn; sxb@sau.edu.cn).

Wei Li Wang is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China (e-mail: liwei089@ieee.org).

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

Digital Object Identifier 10.1109/JSTARS.2020.2982614

by the residual network [21], and the features of different layers are fused in a fully convoluted layer to classify the images. Although the network fuses the outputs of different hierarchical layers, it fuses these outputs only in a fully connected (FC) layer, which does not enable the entire network to make full use of these outputs to learn more discriminative features. Zhong *et al.* [22] proposed a supervised spectral–spatial residual network (SSRN) and the idea of identity mapping in residual blocks mitigates the decreasing-accuracy phenomenon. Inspired by the SSRN, Wang *et al.* [23] proposed an end-to-end fast dense [24] spectral–spatial convolution (FDSSC) framework for HSI classification. The SSRN and FDSSC treat spectral features and spatial features separately in two consecutive blocks, and the spectral and spatial features are also fused only in the FC layer. In addition, the input of the spatial block is based on the spectral block in the SSRN and FDSSC, and the spatial learning will lose spatial information.

To solve these problems and extract more discriminative fusion features, we propose a novel deep multilayer fusion dense network (MFDN) for HSI classification. The MFDN simultaneously extracts the spatial and spectral features based on different sample input sizes, and then the spatial and spectral features are fused together through multilayer fusion strategy with a densely connected structure. For spatial feature extraction, in order to reduce the cost of computation, the principal component analysis (PCA) algorithm is first performed on hyperspectral data to extract low-dimensional HSI data. Then, the spatial features are extracted from the low-dimensional 3-D HSI data through 2-D convolutional, 2-D dense block, and average-pooling layers. For spectral feature extraction, the spectral features are extracted from the raw 3-D HSI data by means of 3-D convolutional, 3-D dense block, and average-pooling layers. For spectral–spatial feature extraction, the spatial and spectral features are fused together through 3-D convolutional, 3-D dense block, and average-pooling layers. Then, the fused spectral–spatial features are fused in two full connection layers to extract high-level abstract features.

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

1) To extract rich spectral and spatial correlation information, MFDN simultaneously extracts spatial and spectral features based on different sample input sizes.
2) MFDN simultaneously fuses the spectral and spatial features in the convolutional layers and the FC layers, which can make full use of complementary spatial–spectral correlation information among different layers.
3) MFDN adopts dense connection structures to extract the spatial and spectral features and fuse the spectral–spatial features, which can help alleviate the vanishing-gradient problem, strengthen feature propagation and encourage feature reuse. Therefore, MFDN can learn more discriminative deep spectral–spatial features to improve classification accuracy.

The rest of this article is organized as follows. In Section II, the proposed MFDN is described in detail. The experimental results and analysis are presented in Section III. Finally, Section IV concludes this article.

II. Proposed Framework

The main procedure of the proposed MFDN is shown in Fig. 1, including deep spectral and spatial features extraction, multilayer deep spectral–spatial features fusion, and a softmax classifier. Generally, a hyperspectral data can be denoted as $I \in \Re^{H\times W\times B}$, where H, W, B denote that the hyperspectral data have $H \times W$ pixels, and B bands, respectively. In the MFDN, due to the high spectral resolution and high spatial correlation of HSI, we first design a spatial extraction network substructure to extract spatial features from the low-dimensional 3-D HSI data obtained by PCA, and design a spectral extraction network substructure to extract spectral features from the raw 3-D HSI data. Then, in order to exploit better spectral–spatial features, a multilayer fusion network is designed to fuse spatial and spectral features. Among the proposed network, the spatial contexts are exploited by 2-D convolutional operation, whereas the spectral correlations and spectral–spatial contexts are exploited by a 3-D convolutional operation. For the proposed framework, batch normalization (BN) [25] and PReLU [26] are added before the convolutional layer. PReLU introduces a small number of parameters based on ReLU [27], and its formula is defined as follows:

$$\mathrm{PReLU}\,(v_i) = \begin{cases} v_i, & \text{if } v_i > 0 \\ \alpha_i v_i & \text{if } v_i \le 0 \end{cases} \tag{1}$$

where v_i is the input of the nonlinear activation on the *i*th channel and α_i is a learnable parameter that determines the slope of the negative part. In PReLU, the momentum method is adopted to update α_i

$$\Delta\alpha_i := \mu\Delta\alpha_i + \gamma\frac{\partial\varepsilon}{\partial\alpha_i} \tag{2}$$

where μ is the momentum and γ is the learning rate, and $\alpha_i = 0.25$ is used as the initial value.

A. 2-D Convolutional Dense Block

In the 2-D convolutional operation, input data are convolved with 2-D kernels before going through the activation function to produce the output data (i.e., feature maps).

As shown in Fig. 2, if the (*n* + 1)th 2-D convolutional layer has k^n input feature squares of size $r^n \times r^n$, a convolutional filter bank that contains k^{n+1} convolutional filters of size $a^{n+1} \times a^{n+1}$, and the subsampling strides of (s, s) for the convolutional operation, then, this layer generates k^{n+1} output feature squares of size $r^{n+1} \times r^{n+1}$, where the spatial width $r^{n+1} = \lfloor 1 + (r^n - a^{n+1})/s\rfloor$. The value of a neuron v_{ixy}^{n+1} at position (x, y) of the *i*th feature map in the $(n + 1)$th layer is denoted as follows:

$$v_{ixy}^{n+1} = F\left(\sum_{j=1}^{k^n}\sum_{m}\sum_{p=0}^{a^{n+1}-1}\sum_{q=0}^{a^{n+1}-1} v_{jm(x+p)(y+q)}^{n} w_{impq}^{n+1} + b_i^{n+1}\right) \tag{3}$$

where *m* indexes the feature map in the *n*th layer connected to the *i*th feature map in the $(n + 1)$th layer, w_{impq}^{n+1} is the weight of

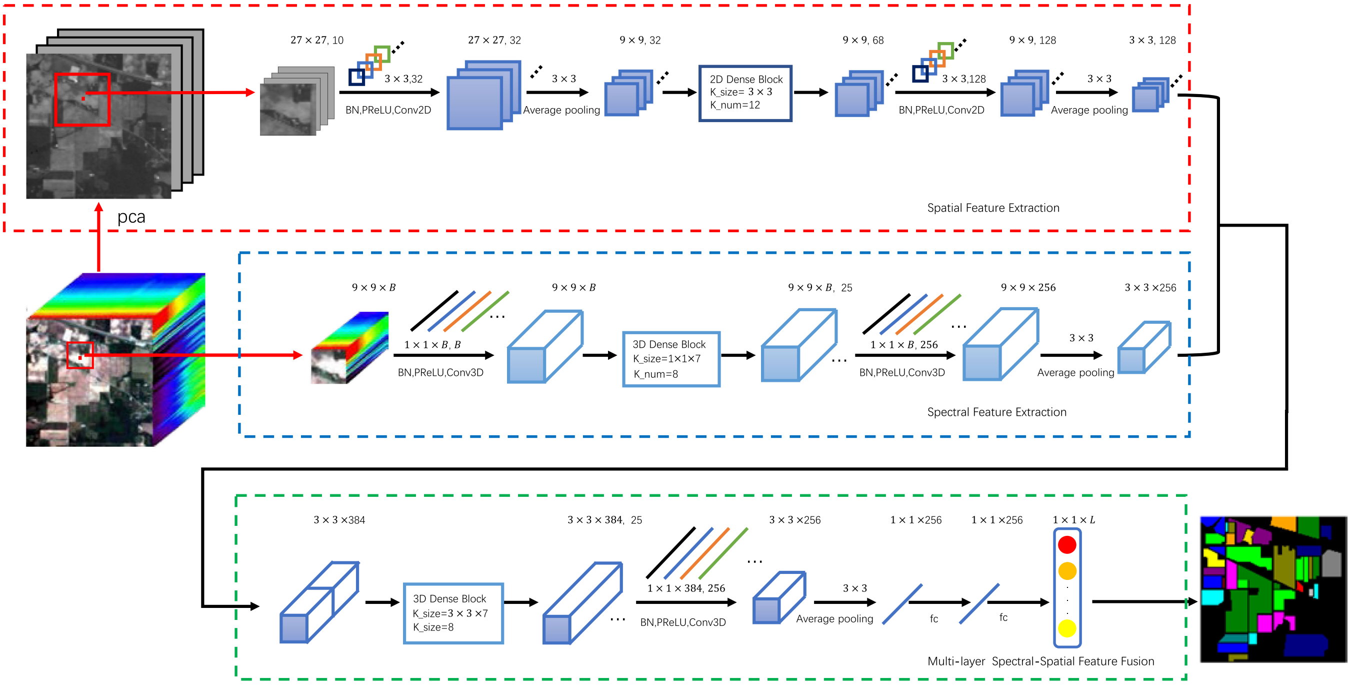


Fig. 1. Overall flowchart of HSI classification based on the MFDN.

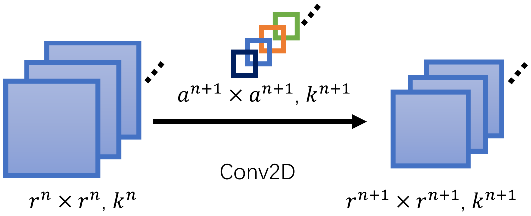

$a^{n+1} \times a^{n+1}$, k^{n+1}
Conv2D
$r^n \times r^n$, k^n
$r^{n+1} \times r^{n+1}$, k^{n+1}

Fig. 2. 2-D convolutional operator.

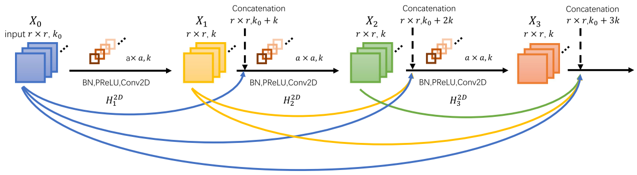


Fig. 3. 2-D convolutional dense block with three composite layers ($l = 4$).

position (p, q) connected to the mth feature map in $(n + 1)$th layer, a^{n+1} is the width of the spatial convolutional kernel, b_i^{n+1} is the bias of the ith feature map in the $(n + 1)$th layer, j indexes the input feature square in the nth layer, and $F(\cdot)$ is the parametric rectified linear unit activation function that sets elements.

Fig. 3 illustrates the layout of 2-D convolutional dense block. As shown in Fig. 3, the input of the lth layer receives the feature maps of all preceding layers $(X_0, X_1, \ldots, X_{l-1})$, and the output of the lth layer is calculated as follows:

$$X_l = H^{2D}\left([X_0, X_1, \ldots, X_{l-1}]\right) \tag{4}$$

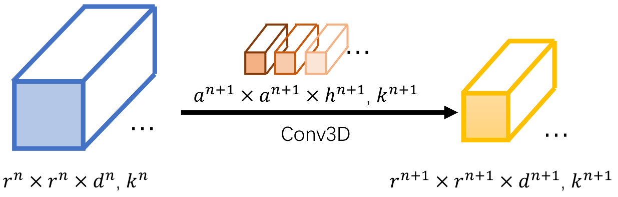

$a^{n+1} \times a^{n+1} \times h^{n+1}$, k^{n+1}
Conv3D
$r^n \times r^n \times d^n$, k^n
$r^{n+1} \times r^{n+1} \times d^{n+1}$, k^{n+1}

Fig. 4. 3-D convolutional operator.

where $[X_0, X_1, \ldots, X_{l-1}]$ represents the concatenation operation of the feature maps produced in layers $(0, 1, \ldots, l - 1)$, $H^{2D}(\cdot)$ is defined as consecutive operations: BN, followed by PReLU, a 3×3 same convolution. Such connectivity pattern strongly encourages feature reuse throughout the network and makes all layers in the architecture receive direct supervision signal from the loss function. If each layer produces k feature maps, thus, the number of input feature maps in layer l can be formulated as follows:

$$k_l = k_0 + (l - 1) \times k \tag{5}$$

where k_0 is the number of channels in the input layer, and the k (generally set a smaller value, e.g., $k = 12$) is referred as growth rate of the 2-D dense block.

B. 3-D Dense Block

In the 3-D convolutional operation, input data are convolved with 3-D kernels before going through the activation function to produce the output data (i.e., feature maps).

As shown in Fig. 4, if the $(n + 1)$th 3-D convolutional layer has k^n input feature cubs of size $r^n \times r^n \times d^n$, a convolutional filter bank that contains k^{n+1} convolutional filters of size

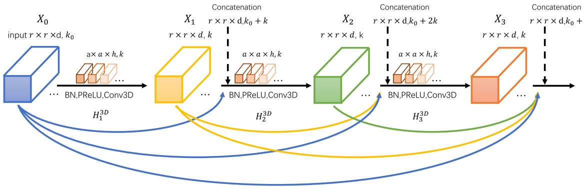


Fig. 5. 3-D convolutional dense block with three composite layers ($l = 4$).

$a^{n+1} \times a^{n+1} \times h^{n+1}$, and the subsampling strides of (s, s, s_1) for the convolutional operation, then, this layer generates k^{n+1} output feature cubs of size $r^{n+1} \times r^{n+1} \times d^{n+1}$, where the spatial width $r^{n+1} = \lfloor 1 + (r^n - a^{n+1})/s \rfloor$ and the spectral depth $d^{n+1} = \lfloor 1 + (d^n - a^{n+1})/s_1 \rfloor$. The value of a neuron v_{ixyz}^{n+1} at position (x, y, z) of the *i*th feature map in the $(n + 1)$th layer is denoted as follows:

$$v_{ixyz}^{n+1} = F\left(\sum_{j=1}^{k^n} \sum_{m} \sum_{p=0}^{a^{n+1}-1} \sum_{q=0}^{a^{n+1}-1} \times \sum_{t=0}^{h^{n+1}-1} v_{jm(x+p)(y+q)(z+t)}^{n} w_{impqt}^{n+1} +, b_i^{n+1}\right) \quad (6)$$

where *m* indexes the feature map in the *n*th layer connected to the *i*th feature map in the $(n + 1)$th layer, w_{impqt}^{n+1} is the weight of position (p, q, t) connected to the *m*th feature map in $(n + 1)$th layer, a^{n+1} is the width of the spatial convolutional kernel, h^{n+1} is the depth of the spatial convolutional kernel, b_i^{n+1} is the bias of the *i*th feature map in the $(n + 1)$th layer, *j* indexes the input feature square in the *n*th layer, and $F(\cdot)$ is the parametric rectified linear unit activation function that sets elements.

Fig. 5 illustrates the layout of 3-D convolutional dense block. As shown in Fig. 5, the input of the *l*th layer receives the feature maps of all preceding layers $(X_0, X_1, \ldots, X_{l-1})$, and the output of the *l*th layer is calculated as follows:

$$X_l = H^{3D}([X_0, X_1, \ldots, X_{l-1}]) \quad (7)$$

where $[X_0, X_1, \ldots, X_{l-1}]$ represents the concatenation operation of the feature maps produced in layers $(0, 1, \ldots, l - 1)$, $H^{3D}(\cdot)$ is defined as consecutive operations: BN, followed by PReLU, a 3×3 same convolution. If each layer produces k feature maps, thus, the lth layer has $k_0 + (l - 1) \times k$ input feature maps, where k_0 is the number of channels in the input layer. Here, the k (generally set a smaller value, e.g., $k = 8$) is referred as growth rate of the 3-D dense block.

C. Spatial Feature Extraction

We take the Indian Pines (IN) dataset, the low-dimensional 3-D samples of which have the size of $27 \times 27 \times 10$, as an example to explain the designed spatial feature extraction substructure.

For spatial features extraction, a PCA algorithm is the first performed on hyperspectral data $I \in \Re^{H \times W \times B}$ to extract the most informative components, which can reduce the cost of computation. The data after executing PCA is denoted as $T \in \Re^{H \times W \times b}$, $b < B$. Then, the spatial neighboring cube patch $T_i \in \Re^{r \times r \times b}$ (r is the patch size, b is the most informative components, and we set r to 27 and b to 10 in the experiment) of the ith pixel is used as the input for the spatial features extraction.

To better exploit spatial structure, and texture features, 2-D convolutional operator is adopted as the basic element of spatial features extraction. In addition, BN is conducted at every convolutional layer in spatial features extraction.

In the red dashed box in Fig. 1, the spatial features extraction section includes two 2-D convolutional layers, a 2-D dense block, and two average pool layers.

In the first convolutional layer, each 3×3 spatial kernel with a subsampling stride of $(1, 1)$ convolves 10 27×27 feature tensors to generate a 27×27 feature tensor. All 32 3×3 spatial kernels generate 32 27×27 feature tensors. Next, an average pooling layer transforms the extracted 32 27×27 spatial feature tensors to 32 9×9 feature tensors.

Then, in order to extract and reuse spatial features effectively, a four-layer 2-D spatial dense block, which contains three convolutional layers and six direct connections, uses 12 3×3 vector kernels with a subsampling stride of $(1, 1)$ at each convolutional layers to extract deep spatial features, and finally produces 68 9×9 feature tensors. In the 2-D spatial dense block, all convolutional layers use padding to keep the sizes of output feature maps the same as input.

Following the 2-D spatial dense block, the next convolutional layer in this feature extraction section, which includes 128 3×3 spatial kernels with a subsampling stride of $(1, 1)$ for keeping discriminative spatial features, convolves the 68 9×9 feature tensors to produce 128 9×9 feature tensor. Next, an average pooling layer transforms the extracted 128 9×9 spatial feature tensors to a 128 3×3 feature tensors.

D. Spectral Feature Extraction

We take the IN dataset, the 3-D samples of which have the size of $9 \times 9 \times B$ and $B = 200$, as an example to explain the designed spectral feature extraction substructure.

For spectral features extraction, the 3-D convolutional operation is employed to capture spectral correlations from HSI data in spectral dimension. Specially, a spectral kernel of size $1 \times 1 \times b$ $(1 < b \leq B)$ is utilized to learn the spectral features from a HIS.

Though the 3-D convolutional operation with a kernel size of $1 \times 1 \times b$ can exploit the spectral correlations, it does not consider the relationship between pixels and their neighbors in the spatial field. However, the convolutional of a kernel size of 1×1can make linear combinations or integrate spatial information for each pixel in a small spatial patch. Therefore, in a small spatial patch, the 3-D convolutional operation with a kernel size of $1 \times 1 \times b$ can extract spectral correlations and perfectly retains the spatial correlations.

In the blue dashed box in Fig. 1, the spectral features extraction section includes two 3-D convolutional layers, a 3-D dense block, and an average pool layer. In addition, BN is conducted at every convolutional layer in spectral features extraction.

In the first convolutional layer, B $1 \times 1 \times B$ spectral kernels with a subsampling stride of $(1, 1, 0)$ convolve the input HSI volume of size $9 \times 9 \times B$ to generate a $9 \times 9 \times B$ feature cube. Because the raw input data contain redundant spectral information, $1 \times 1 \times B$ vector kernels are used in these blocks. This layer reduces the high dimensionality of input cubes and extracts low-level spectral features of HSI.

Then, in order to extract and reuse spectral features effectively, a four-layer 3-D spectral dense block, which contains three convolutional layers and six direct connections, uses 8 $1 \times 1 \times 7$ vector kernels with a subsampling stride of $(1, 1, 1)$ at each convolutional layers to extract deep spectral features, and finally produces 25 $9 \times 9 \times B$ feature cubes. In the 3-D spectral dense block, all convolutional layers use padding to keep the sizes of output feature cubes the same as input.

Following the 3-D spectral dense block, the last convolutional layer in this feature extraction section, which includes 256 $1 \times 1 \times B$ spectral kernels with a subsampling stride of $(1, 1, 0)$ for keeping discriminative spectral features, convolves the 25 $9 \times 9 \times B$ feature cubes to produce a $9 \times 9 \times 256$ feature cube. Next, an average pooling layer transforms the extracted $9 \times 9 \times 256$ spectral feature volume to a $3 \times 3 \times 256$ feature volume.

E. Spectral–Spatial Feature Extraction

The PCA algorithm is first performed on hyperspectral data to extract low-dimensional HSI data. 2-D convolutional and 2-D dense block can exploit perfectly spatial correlations in low-dimensional HSI data.

3-D convolution and 3-D dense block with a kernel size of $1 \times 1 \times b$ can extract spectral correlations and perfectly retains the spatial correlations.

In order to fuse the spatial and spectral features, a spectral–spatial feature extraction substructure is designed. As shown in the green dashed box in Fig. 1, the spatial and spectral features are first concatenated to cascade features, which are characterized by both the spatial and spectral dimensions. 3-D dense block and 3-D convolutional operation are then applied to cascade features to extract spatial-spectral features simultaneously. In addition, an average pool layer and two FC layers can exploit more abstract spatial-spectral features at high levels, which are generally robust and invariant [28].

The 3-D dense block contains three 3-D convolutional layers and six direct connections. 8 $3 \times 3 \times 7$ vector kernels with a subsampling stride of $(1, 1, 1)$ at each convolutional layers are used to extract deep spectral–spatial features, and finally produce25 $3 \times 3 \times 384$ feature tensors.

Following the 3-D spectral–spatial dense block, the 3-D convolutional layer in this feature extraction section, which includes 256 $1 \times 1 \times 384$ spectral kernels with a subsampling stride of $(1, 1, 0)$ for keeping discriminative spectral–spatial features, convolves the 25 $3 \times 3 \times 384$ feature cubes to produce a $3 \times 3 \times 256$ feature cube. Then, an average pooling layer transforms the extracted $3 \times 3 \times 256$ spectral–spatial feature volume to a $1 \times 1 \times 256$ feature volume. Next, two FC layers adapt the MFDN to HSI dataset according to the number of land-cover categories and generates an output vector $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L]$. The truth label vector $y = [y_1, y_2, \ldots, y_L]$ is the number of land-cover categories. The loss function of the MFDN is defined as

$$\text{Loss} = -\frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \left[y_i \log\ (\hat{y}_i) + (1 - y_i) \log\ (1 - \hat{y}_i) \right] \tag{8}$$

where $\hat{y}_i$ is the corresponding predicted labels for the ith training sample, y_i is the true label, and n_{train} is the size of training set. The whole network is trained in an end-to-end manner, where all the parameters are optimized by the Adam [29] at the same time.

TABLE I
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN THE IN DATASET

No.	Class Name	Numbers of Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-tree	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
	Total	10,249

III. RESULTS AND DISCUSSION

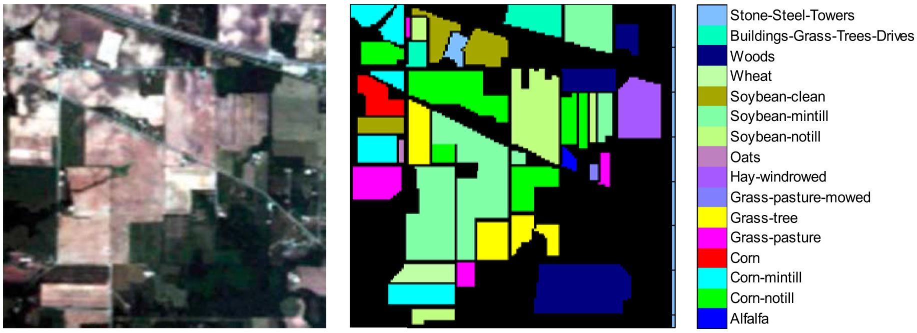
A. Experimental Datasets

In our experiments, the effectiveness of our method is proved in three real-world hyperspectral remote sensing datasets, which contain the IN, the University of Pavia (UP), and the Kennedy Space Center (KSC) datasets.

The IN dataset was collected by AVIRIS in 1992 in northwestern Indiana. This commonly used dataset has 16 vegetation classes and 224 bands. The spatial size is 145×145 and the spatial resolution is 20 m per pixel. To avoid the negative influence on classification due to water absorption and noise, some bands are discarded and the remaining 200 bands are adopted for analysis. Fig. 6 shows the false-color image and the ground-truth map, and the samples are listed in Table I.

The UP was captured by a Reflective Optics System Imaging Spectrometer optical sensor over an urban area surrounding the UP. The image is of size $610 \times 340 \times 115$ with a resolution of 1.3 m per pixel and nine urban land-cover classes are considered in this experiment. The number of remaining bands is 103 after discarding the useless bands. Fig. 7 shows the false-color image and the ground-truth map, and the samples are listed in Table II.

The KSC dataset was collected by AVIRIS in 1996 in Florida, and contains 512×614 pixels with spatial resolution of 18 m per pixel and the ground-truth classes are 13. After removing the noise bands, 176 bands are retained and used for our experiments. Fig. 8 shows the false-color image and the ground-truth map, and the samples are listed in Table III.



(a) (b)

Fig. 6. (a) False-color image of the IN dataset. (b) Ground truth of the IN dataset.

TABLE II
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN THE UP DATASET

No.	Class Name	Numbers of Samples
1	Asphalt	6631
2	Meadows	18,649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947
	Total	42,776

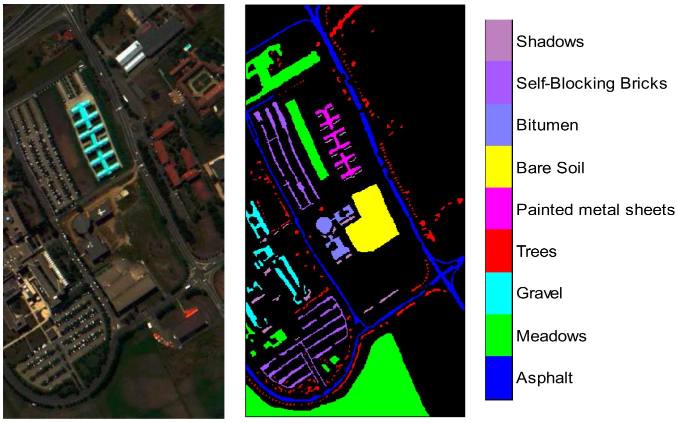
TABLE III
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN THE KSC DATASET

No.	Class Name	Numbers of Samples
1	Scrub	761
2	Willow swamp	243
3	CP hammock	256
4	Slash pine	252
5	Oak/Broadleaf	161
6	Hardwood	229
7	Grass-pasture-mowed	105
8	Graminoid marsh	431
9	Spartina marsh	520
10	Cattail marsh	404
11	Salt marsh	419
12	Mud flats	503
13	Water	927
	Total	5211

B. Experimental Settings

In our implementation, the learning rate was set to 0.0001, the training epoch was 150 for the IN dataset, 80 for the UP dataset, and 200 for the KSC dataset. The optimizer adopted the Adam method and the batch size was set to 30.

All experiments were conducted on a Lenovo ThinkCentre with NVIDIA P106-100 GPU, Intel i3-7100 CPU, and 16 GB RAM. The software environment of the workstation is python3.6.3, tensorflow1.9.0, cuda9.0, and keras2.2.6. The proposed method was compared with some state-of-the-art methods including the Two-CNN [17], 3D-CNN [19], SSRN [22], DFFN [20], and FDSSC [23]. In the above-compared methods, the input sizes of FDSSC and SSRN are (9,9, B) and (7,7, B), respectively, where B is the number of bands of the raw hyperspectral data. In DFFN, the PCA algorithm is first applied to the hyperspectral data. Different input sizes are set for different datasets, where the IN dataset is (25,25,3), the KSC dataset is (27,27,9), and the UP dataset is (23,23,5). The input size of 3D-CNN is set to (5,5, B). Two-CNN has Two branches and the input size of spatial branch is (21, 21) and the input size of spectral branch is (1,1,B). FDSSC and SSRN first extract the spectral information and then extract the spatial information. Their spatial input sizes are (9,9) and (7,7). The overall accuracy (OA), the average accuracy (AA), and d kappa coefficient (k) are the classification metrics used to assess the classification performance of all the methods. We ran experiments for ten times with randomly selected training data and reported the mean and standard deviation of main classification metrics. We evaluated the performance of all methods on the small training samples to prove that our proposed MFDN has strong robustness and generalization.

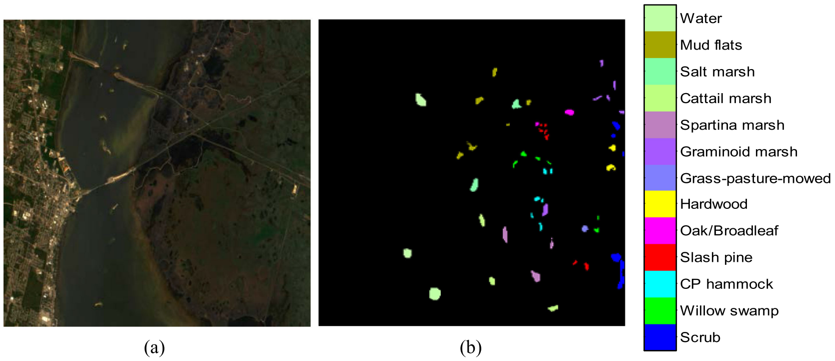


(a) (b)

Fig. 7. (a) False-color image of the UP dataset. (b) Ground truth of the UP dataset.

C. Analysis of Parameters

For the proposed MFDN method, the different sample input sizes are set for the spectral and spatial feature extraction, respectively. In addition, for spatial feature extraction, the PCA algorithm is performed on the original HSIs with the purpose of extracting first several principal components. The corresponding experiments are performed on the IN, UP, and KSC datasets, respectively. For the IN, UP, and KSC datasets, 3% of labeled



(a) (b)

Fig. 8. (a) False-color image of the ground-truth map of the KSC data. (b) Ground truth of the KSC data.

TABLE IV
OA of Our Methods on the Three Datasets With Different Sample Input Sizes

Data Set	Spectral Input Size Spatial Input Size	5×5	7×7	9×9	11×11	13×13
IN	23×23	90.31	93.77	93.76	93.45	93.75
	25×25	92.99	94.71	94.76	94.22	94.11
	27×27	93.77	94.98	**96.08**	95.26	95.09
	29×29	92.21	95.67	95.77	95.19	94.62
	31×31	91.02	95.3	95.54	95.04	94.1
UP	23×23	98.05	98.55	98.67	98.42	98.25
	25×25	98.16	98.31	98.39	98.30	98.23
	27×27	98.56	98.70	**98.89**	98.86	98.85
	29×29	98.65	98.83	98.85	98.79	98.83
	31×31	98.6	98.69	98.82	98.80	98.81
KSC	23×23	90.57	92.1	95.66	95.51	95.45
	25×25	92.26	94.7	96.57	96.21	96.02
	27×27	92.73	94.2	**97.55**	96.82	96.53
	29×29	90.37	93.74	97.42	96.79	97.06
	31×31	92.45	93.82	96.81	96.57	96.63

pixels are randomly selected as training samples, and the rest of samples are utilized for testing. For the UP dataset, we only use 0.5% samples per class to train classifiers, and the rest of samples are used as the test samples.

1) Effect of the Sample Input Size on Classification Accuracies: For the CNN used for HSI classification, the sample input size is an important factor affecting the HSI classification. In the MFDN, we set different input sizes for spectral and spatial feature extraction, respectively.

For spectral feature extraction, the sample input size is set to 5×5, 7×7, 9×9, 11×11 and 13×13. For spatial feature extraction, the sample input size is set to 23×23, 25×25, 27×27, 29×29, 31×31.

We measured the OA for each dataset. Table IV lists the classification results (OA%) of our methods on the three datasets with different input sizes.

As can be seen from Table IV, when the sample input size was set to 9×9 (for spectral feature extraction) and 27×27 (for spatial feature extraction), respectively, the MFDN achieved the best overall classification accuracies on all three datasets.

2) Effect of the Number of Principal Components on Classification Accuracies: For the proposed MFDN method, the PCA algorithm was first performed on the original HSIs in the extraction of spatial features. In this analysis, the spatial size of the input sample is empirically set to 9×9 and 27×27 for spectral feature extraction and spatial feature extraction, respectively.

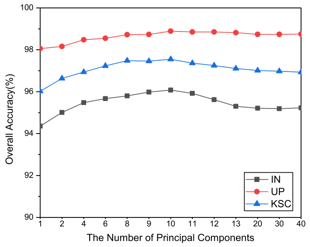


Fig. 9. Effect of the number of principal components on classification accuracies in the three datasets.

Fig. 9 shows the overall accuracies with the different number of principal components on three datasets. As can be seen from Fig. 9, the overall accuracies on the IN dataset UP dataset and KSC dataset generally increase and then become comparatively stable as the number of principal components increases. When the number of principal components was 10, the overall accuracies on the three datasets reached a higher accuracy. We also

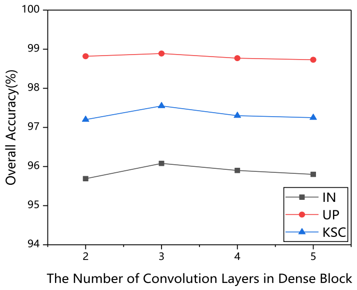


Fig. 10. Effect of the number of convolution layers in a dense block on classification accuracies in the three datasets.

performed an experiment without PCA. The overall accuracies without PCA were, respectively, 95.37%, 98.77%, and 96.96% on the IN dataset UP dataset and KSC dataset. However, when the number of principal components was 10, the overall accuracies of the three sets of data were 96.08%, 98.89%, and 97.55% on the IN dataset UP dataset and KSC dataset, respectively, which is slightly better than the OA of the analysis without principal components.

3) Effect of the Number of Convolution Layers in the Dense Block: In a dense block, the output of each convolutional layer is a part of the input of all subsequent convolutional layers. Therefore, the number of convolutional layers in the dense block determines the degree of feature reuse in the dense block. Fig. 10 shows the overall accuracies with the different number of convolution layers in the dense block on three datasets. It can be seen from Fig. 10 that when the number of convolution layers in the dense block was set to 3, the overall accuracies of the three datasets reach a higher accuracy.

D. Experiment Results and Analysis

In order to prove the superiority of the proposed network MFDN in the case of small label samples, we compared MFDN with other state-of-the-art methods on the three datasets. To verify the effectiveness of the spectral and spatial feature extraction parts in this framework, we also tested the network containing only the spectral feature extraction part (Spectral) and the network containing only the spatial feature extraction part (Spatial). The classification results are shown in Fig. 11.

The influence of different training and test sets on several methods is first analyzed on the IN, UP, and KSC datasets, respectively. For the IN, UP, and KSC datasets, different percentages (from 1% to 5% for the IN and KSC datasets, 0.1%–0.7% for the UP dataset) of labeled pixels per class are randomly selected as training samples, and the rest of samples are used as test samples. Specifically, all experimental results are averaged ten times with different randomly selected training data. Fig. 11 shows the overall classification accuracy of each method under different numbers of training samples. As can be seen from the curve, as the number of training samples increases, the performance of all methods generally increases.

As can be also seen from Fig. 11, the MFDN, DFFN, FDSSC, and SSRN achieved higher overall accuracies than the CNN and Two-CNN in most cases. It can be seen from the above analysis that the residual connections or the dense connections can achieve a better effect.

In all three cases, the MFDN achieved the highest classification accuracies than other methods. Compared with the Two-CNN, which only fused spectral and spatial features in the FC layer, the MFDN fused spectral and spatial features by 3-D convolutional, 3-D dense block, and FC layers, so it achieved higher overall accuracies. Compared with the SSRN and FDSSC, which do not consider the original spatial correlation information, the MFDN simultaneously learns spectral and spatial features and fuses them together, so it has obvious advantages in most cases.

Tables V–VII report the OAs, AAs, kappa coefficients, and the classification accuracies of all classes for HSI classification. The corresponding classification maps on the IN, UP, and KSC datasets are shown in Figs. 12–14, respectively. For the IN and KSC dataset, the training set, validation set, and test set were split into 3%, 5%, and 92%, respectively. For the UP dataset, the training set, validation set, and test set were split into 0.5%, 5%, and 94.5%, respectively. All experimental results are averaged ten times with different randomly selected training data. As can be seen from Figs. 12–14, the MFDN achieved the most accurate and smooth classification maps for all three HSIs.

As can be seen from Tables V–VII, the MFDN is superior to Two-CNN, 3D-CNN, SSRN, DFFN, and FDSSC methods in all three cases. The DFFN and SSRN with residual connections generated obviously better outcomes than the Two-CNN and 3D-CNN in most cases. The MFDN and FDSSC with dense connections also generated obviously better outcomes than the Two-CNN and 3D-CNN. It is worth noting that the Spectral performed better than the Two-CNN and 3D-CNN, and the Spatial performed better than the Two-CNN and 3D-CNN in most cases. These results show that the proposed spectral and spatial dense connection structures alleviate the phenomenon of reduced accuracy. In addition, MFDN always outperforms Spectral and Spatial due to the fusion of complementary space-spectrum correlation information among different layers.

In contrast to the idea of fusing only through the full connection layer in the DFFN and Two-CNN, the MFDN fuses spatial and spectral features through 3-D convolutional, 3-D dense block, and FC layers, which can learn more discriminative features. It can be seen from Tables V–VII that, on the IN, UP, and KSC datasets, the mean overall classification accuracy of the MFDN is 3.52%, 4.17%, and 6.67% higher than that of the DFFN, and 35.8%, 30.36%, and 24.75% higher than that of the Two-CNN. The SSRN and FDSSC adopt consecutive spectral and spatial blocks to learn the spectral–spatial feature, and the input of spatial blocks is based on spectral blocks, which causes spatial learning blocks to lose spatial information. In particular, the SSRN and FDSSC also fuse spectral and spatial features only in the FC layer. Different from the idea of spectral–spatial fusion in the SSRN and FDSSC, the MFDN simultaneously learns spectral and spatial features and sends them into a multilayer structure (including 3-D convolutional, 3-D dense block, and FC layers) for fusion, which can achieve the abundant spectral and spatial structure information and extract more discriminative features. It can be seen from Tables V–VII that, on the IN, UP, and KSC datasets, the mean overall classification accuracy of

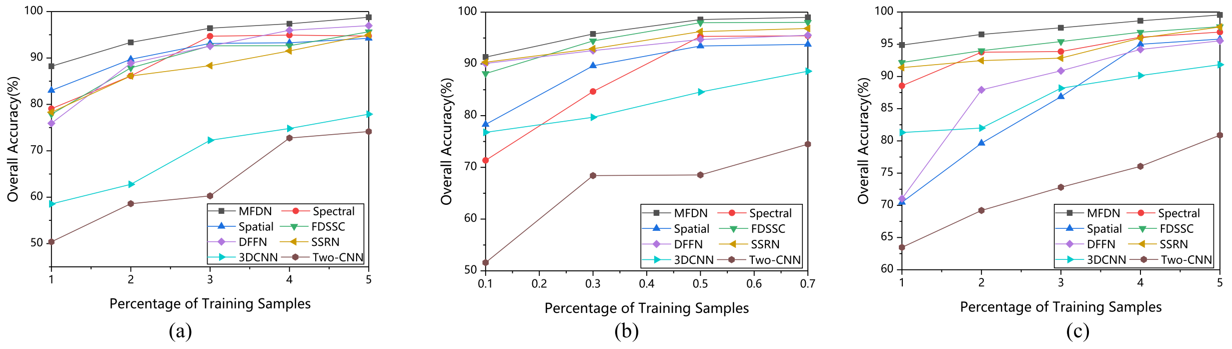


Fig. 11. OA of changing the percentage of training samples by all methods on the three datasets. (a) OA on IN. (b) OA on UP. (c) OA on KSC.

TABLE V
Classification Results of Different Methods for the IN Dataset

	Two-CNN	**3D-CNN**	**SSRN**	**DFFN**	**FDSSC**	**Spatial**	**Spectral**	**MFDN**
1	60.71	30.43	96.87	88.25	55.22	75.75	82.27	**97.2**
2	51.54	71.4	**96.26**	92.28	85.65	94.32	91.92	95.16
3	51.54	50.42	91.67	88.5	94.24	96.06	92.74	**96.57**
4	51.86	25.76	77.33	**90.52**	81.9	89.66	93.18	90.48
5	69.26	73.93	**94.03**	90.53	93.14	90.59	92.86	91.99
6	67.04	92.08	98.58	95.04	**99.18**	96.32	96.55	98.93
7	41.42	16.21	63.32	88.36	16.34	35.8	42.21	**90.9**
8	80.39	98.55	92.85	98.58	99.61	**99.96**	99.3	99.75
9	55	28.89	55.1	68.42	13.88	24.56	14.73	**82.9**
10	45.08	63.57	91.9	91.13	93.48	91.82	93.18	**95.35**
11	58.35	77.23	89.48	94.4	91.17	93.33	93.45	**96.52**
12	39.68	34.57	67.97	83.03	**94.85**	83.7	85.84	90.04
13	82.92	94.82	97.37	95.6	**98.85**	94.27	98.48	98.76
14	77.49	94.63	93.39	95.19	**99.28**	96.14	97.63	98.37
15	60.51	51.6	**99.02**	93.54	91.14	93.84	95.5	96.53
16	70.13	78.01	98.54	87.78	96.34	74.07	**99.87**	99.01
OA	60.28	72.26	88.34	92.56	92.63	93.14	94.7	**96.08**
	±1.44	±1.91	±0.37	±1.21	±1.11	±1.01	±1.87	**±0.53**
AA	60.19	61.38	92.03	90.08	81.52	80.89	85.45	**94.89**
	±1.04	±2.36	±0.88	±1.52	±0.84	±1.75	±2.86	**±1.33**
Kappa×100	52.94	68.07	86.71	91.53	91.91	91.19	91.6	**95.26**
	±1.73	±2.26	±0.42	±1.37	±1.24	±1.1	±2.07	**±0.73**

TABLE VI
Classification Results of Different Methods for the UP Dataset

	Two-CNN	**3D-CNN**	**SSRN**	**DFFN**	**FDSSC**	**Spatial**	**Spectral**	**MFDN**
1	70.69	88.63	97.5	93.09	**98.57**	95.51	97.44	98.34
2	73.39	94.67	98.07	98.71	99.1	99.53	98.2	**99.93**
3	42.52	61.09	91.57	84.8	85.89	86.62	84.53	**92.55**
4	80.51	88.28	**98.89**	92.28	95.72	63.1	87.47	94.63
5	84.81	97.98	99.67	98.12	99.84	99.79	**99.98**	99.9
6	39.88	55.13	97.85	94.43	**99.72**	99.53	91.09	99.6
7	30.49	50.03	92.85	84.1	96.41	98.54	90.96	**99.71**
8	61.98	82.06	84.80	93.14	97.24	91.7	95.11	**97.3**
9	86.82	93.09	**100**	74.95	98.89	31.48	95.32	95.19
OA	68.53	84.57	96.25	94.72	97.96	93.46	95.31	**98.89**
	±3.91	±1.04	±0.34	±1.01	±0.42	±0.66	±1.22	**±0.25**
AA	63.45	78.99	95.7	90.4	96.79	85.09	93.35	**97.46**
	±5.22	±2.38	±1.21	±0.69	±0.88	±1.44	±1.97	**±0.68**
Kappa×100	55.61	79.21	95.24	92.99	97.3	91.31	93.76	**98.1**
	±5.23	±1.57	±0.31	±1.34	±0.56	±0.88	±1.64	**±0.33**

TABLE VII
Classification Results of Different Methods for the KSC Dataset

	Two-CNN	**3D-CNN**	**SSRN**	**DFFN**	**FDSSC**	**Spatial**	**Spectral**	**MFDN**
1	86	92.21	92.43	92.92	99.83	99.83	99.83	**99.84**
2	66.29	91.08	97.26	78.81	**99.4**	59.57	90.35	96.81
3	64.79	91.54	95.57	89.28	**99.67**	68.57	96.5	97.11
4	49.82	38.99	81.85	62.05	69.91	29.1	74.03	**86.33**
5	70.45	54.79	52.25	**92.69**	36.02	90.14	41.64	**86.32**
6	56.04	62.5	63.36	89.28	**97.47**	57.65	81.97	93.32
7	52.64	79.58	60.84	90.3	63.16	57.91	96.36	**99.51**
8	59.26	86.98	91.06	88.47	99.66	85.4	96.96	**99.72**
9	66.29	94.16	98.23	86.03	**99.96**	86.5	94.96	97.35
10	60.22	92.71	98.47	88.13	**99.74**	96.28	97.09	95.78
11	85.66	94.45	99.34	98.13	98.96	99.13	99.75	**99.96**
12	71.3	91.53	98.76	95.61	**99.05**	97.02	94.59	98.6
13	91.11	100	99.52	**100**	**100**	100	99.66	**100**
OA	72.8 ±1.12	88.17 ±0.87	92.85 ±1.43	90.88 ±1.56	95.42 ±0.87	86.86 ±0.76	93.86 ±0.67	**97.55** **±0.75**
AA	67.68 ±2.53	82.35 ±1.24	88.25 ±2.21	88.59 ±1.68	92.43 ±2.59	78.86 ±1.42	89.52 ±1.38	**96.21** **±1.08**
Kappa×100	69.61 ±1.28	86.81 ±0.97	92.01 ±1.16	89.85 ±1.75	94.9 ±0.96	85.28 ±1.11	93.17 ±0.75	**97.27** **±0.84**

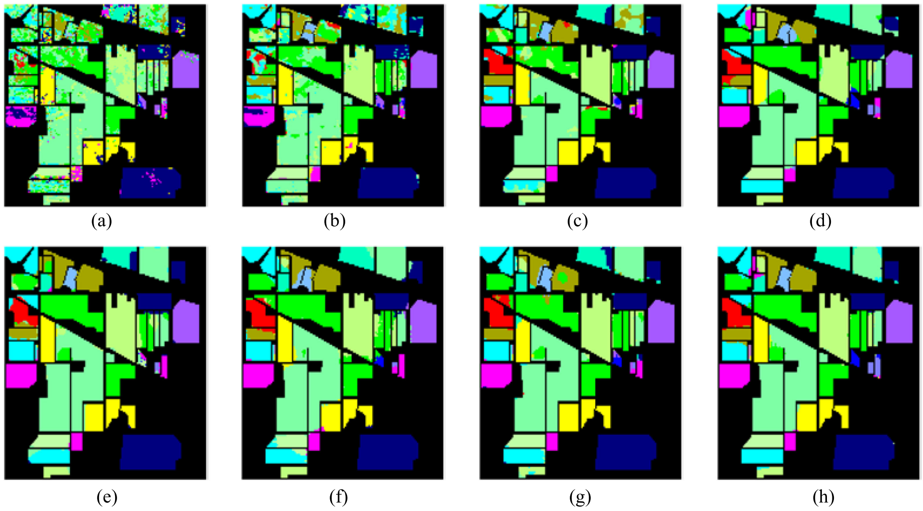

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 12. Classification maps on the IN dataset obtained by (a) Two-CNN, (b) 3D-CNN, (c) SSRN, (d) DFFN, (e) FDSSC, (f) Spatial, (g) Spectral, and (h) MFDN.

the MFDN is 3.45%, 0.93%, and 2.13% higher than that of the FDSSC, and 7.74%, 2.64%, and 4.7% higher than that of the SSRN.

It is worth noting that when training samples are very few (for example, there is only one sample for grass-pasture-mowed and oats classes in the IN dataset), the FDSSC with dense connections is inferior to SSRN with residual connections, and even inferior to Two-CNN. However, compared with SSRN (63.32% and 55.1%), the overall classification accuracy of MFDN (90.9% and 82.9%) in grass-pasture-mowed and oats increased by about 27.58% and 27.8%. These results validated the robustness of the designed models under very difficult conditions and demonstrated the effectiveness of a multilayer fusion strategy.

The training and testing times provide a direct measure of computational efficiency for the MFDN. All experiments were conducted on a Lenovo ThinkCentre with NVIDIA P106-100 GPU, Intel i3-7100 CPU, and 16 GB RAM. The software environment of the workstation is python3.6.3, tensorflow1.9.0, cuda9.0, and keras2.2.6. The training set, validation set, and test set of all methods on the IN and KSC datasets were split into 3%, 5%, and 92% for the IN and KSC datasets, respectively. For the UP dataset, the training set, validation set, and test set were split into 0.5%, 5%, and 94.5%, respectively. Table VIII lists the results of training and test times for all methods on three different datasets. It can be seen from Table VIII that the training time of the SSRN, DFFN, FDSSC, and MFDN is longer than that of the CNN and Two-CNN, which means that the computational cost of residual or dense connections is more expensive. The training time of the MFDN and FDSSC is longer than that of the SSRN and DFFN, which means that the dense connections are more computationally expensive than the residual connections. Although the MFDN has a longer training time in most cases, it

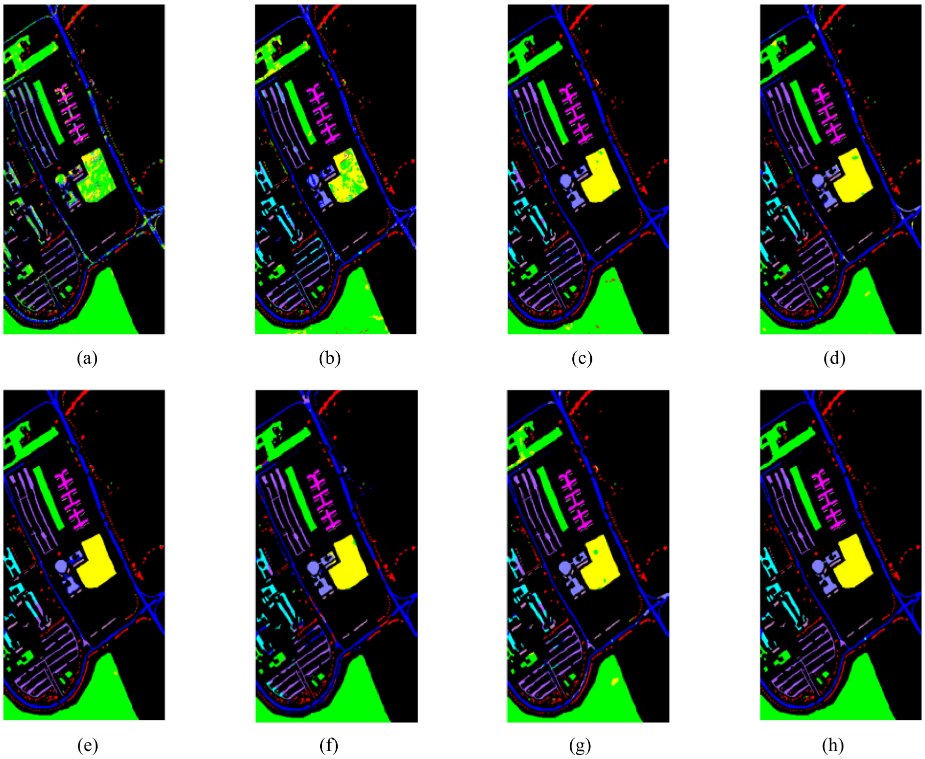


Fig. 13. Classification maps on the UP dataset obtained by (a) Two-CNN, (b) 3D-CNN, (c) SSRN, (d) DFFN, (e) FDSSC, (f) Spatial, (g) Spectral, and (h) MFDN.

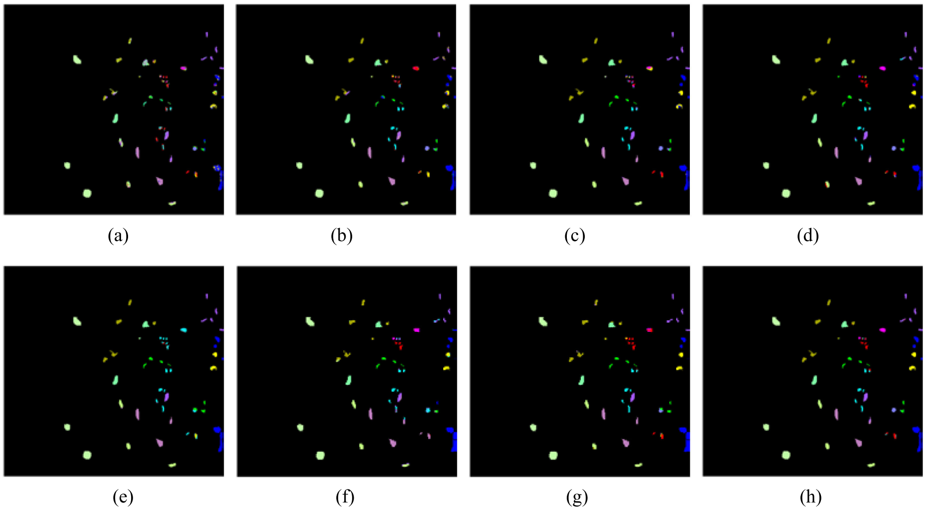


Fig. 14. Classification maps on the KSC dataset obtained by (a) Two-CNN, (b) 3D-CNN, (c) SSRN, (d) DFFN, (e) FDSSC, (f) Spatial, (g) Spectral, and (h) MFDN.

TABLE VIII
TRAINING AND TESTING TIMES OF DIFFERENT MODELS FOR THREE HSI DATASETS

		Two-CNN	3D-CNN	SSRN	DFFN	FDSSC	Spatial	Spectral	MFDN
IN	Train(s)	60.7	64.5	291.6	215.1	402.0	99.51	343.42	699.3
	Test(s)	7.5	5.0	5.1	7.9	19.0	4.35	14.64	27.1
UP	Train(s)	65.51	73.46	147.44	306.8	616.15	98.98	219.84	739.26
	Test(s)	2	3.63	13.37	22.61	55.65	13.2	36.3	64.33
KSC	Train(s)	12.8	32.5	208.1	327.7	537.9	101.85	237.58	533.5
	Test(s)	5.0	1.0	2.2	10.7	12.8	3.52	7.65	16.1

has a higher classification accuracy, especially in the IN dataset that is difficult to classify.

E. Discussions

First, compared with other deep learning methods, the MFDN extracts spatial and spectral features based on different sample input sizes. Spatial features are extracted in a large neighborhood to extract more spatial correlation information, whereas spectral features are extracted in a relatively small neighborhood to extract more spectral correlation information. For spatial feature extraction, the abundant spatial correlation information can be obtained in large neighborhood. For spectral feature extraction, in small neighborhood, spectral feature extraction can exploit spectral correlation information and perfectly retains the spatial correlation information.

Second, compared with other deep learning methods, the MFDN adopts multilayer fusion strategy to fuse the spatial and spectral features. The Two-CNN, SSRN, DFFN, and FDSSC only fuse spectral and spatial features in the FC layer, which cannot achieve more discriminating features. The SSRN and FDSSC treat spectral features and spatial features separately in two consecutive blocks, and the input of spatial blocks is based on spectral blocks, which causes spatial learning blocks to lose spatial information. The MFDN adopts dense connections to simultaneously extract spectral and spatial features, and fuses them through 3-D convolutional, 3-D dense block, and FC layers. On the one hand, the abundant spatial and spectral correlation information can be exploited, and on the other hand, the spatial and spectral features can be better fused.

Third, the MFDN and FDSSC adopt dense connections that strengthen feature propagation, encourage feature reuse, and improve the classification accuracy. It is worth noting that when training samples are very few, the FDSSC with dense connections is inferior to SSRN with residual connections, and even inferior to Two-CNN. However, the MFDN achieved very high classification accuracy in this case, which demonstrated the effectiveness of multilayer fusion strategy.

Finally, the shortcoming of the MMFN model is that the training time is relatively long, which is mainly because the network consists of spatial and spectral branches, and the spatial and spectral features are fused by a multilayer fusion strategy. In addition, the densely connected structure also increases the corresponding time. Fortunately, however, the adoption of GPU has largely alleviated the extra computational costs and reduced the training times.

IV. Conclusion

In this article, a deep MFDN is proposed for HSI classification. Compared with previous deep networks, the proposed MFDN simultaneously extracts the spatial and spectral features based on different sample input sizes, which can extract the abundant spectral and spatial correlation information. In addition, a multilayer fusion strategy with a densely connected structure is exploited to fuse the spatial and spectral features, which can extract more discriminative spectral–spatial features. Finally, the dense connection-based network model can strengthen feature propagation, encourage feature reuse, and improve the classification accuracy. The experimental results demonstrate that the proposed MFDN can obtain the state-of-the-art performance with small labeled samples on the three data, and can be easily generalized to other remote sensing scenarios due to its uniform structural design and deep feature learning ability.

The MFDN is still a time-consuming model compared to traditional methods. In future work, we will focus on further simplifying the network structure while improving classification accuracy.

References

[1] R. Tao, X. Zhao, W. Li, H. Li, and Q. Du, "Hyperspectral anomaly detection by fractional Fourier entropy," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4920–4929, Dec. 2019.

[2] B. Luo, C. Yang, J. Chanussot, and L. Zhang, "Crop yield estimation based on unsupervised linear unmixing of multidate hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 162–173, Jan. 2013.

[3] X. Yang and Y. Yu, "Estimating soil salinity under various moisture conditions: An experimental study," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2525–2533, May 2017.

[4] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[5] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[6] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.

[7] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.

[8] X. Zhang, Y. Liang, Y. Zheng, and J. An, "Hierarchical discriminative feature learning for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 594–598, Apr. 2016.

[9] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[10] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[11] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li, "Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6298–6314, Oct. 2014.

[12] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.

[13] W. Shao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Oct. 2016.

[14] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[15] Z. Li, L. Huang, D. Zhang, C. Liu, Y. Wang, and X. Shi, "A deep network based on multiscale spectral-spatial fusion for hyperspectral classification," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, 2018, pp. 283–290.

[16] Z. Li, L. Huang, and J. He, "A multiscale deep middle-level feature fusion network for hyperspectral classification," *Remote Sens.*, vol. 11, no. 6, pp. 695–794, 2019.

[17] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[18] 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.

[19] 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. 67–87, 2017.

[20] 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.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[22] 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.

[23] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral–spatial convolution network framework for hyperspectral images classification," *Remote Sens.*, vol. 10, no. 7, pp. 1068–1086, 2018.

[24] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2261–2269.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[28] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

Zhaokui Li received the M.S. degree in computer application from Liaoning University, Shenyang, China, in 2003, and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2014.

He is currently a Professor with the School of Computer, Shenyang Aerospace University, Shenyang, China. His research interests include remote sensing, computer vision, and machine learning.

Tianning Wang received the B.S. degree in 2017 from Shenyang Aerospace University, Shenyang, China, where he is currently working toward the master's degree with the School of Computer.

His research interests include hyperspectral image processing and deep learning.

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

Subsequently, he spent one year as a Postdoctoral Researcher with the University of California, Davis, CA, USA. He is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology, Beijing, China. His research interests include hyperspectral image analysis, pattern recognition, and data compression.

Dr. Li is an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.

Qian Du (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland, Baltimore, MD, USA, in 2000.

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

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

Chuanyun Wang received the Ph.D. degree in pattern recognition and intelligent system from Beihang University, Beijing, China, in 2017.

He is currently an Associate Professor with the School of Computer Science, Shenyang Aerospace University, Shenyang, China. His research interests include machine vision, pattern recognition, and Internet of Things.

Cuiwei Liu received the B.S. and Ph.D. degrees from Beijing Institute of Technology, Beijing, China, in 2009 and 2015, respectively.

She is currently an Associate Professor with the School of Computer Science, Shenyang Aerospace University, Shenyang, China. Her research interests include computer vision, machine learning, and video content analysis.

Xiangbin Shi received the B.S. degree in computer application from the Shenyang University of Technology, Shenyang, China, in 1985, and the M.S. degree in computer application and Ph.D. degree in computer software and theory from the Northeastern University, Shenyang, China, in 1990 and 1998, respectively.

He is currently a Professor of Computer Science with Shenyang Aerospace University, Shenyang, China. His research interests include computer vision, virtual reality, and intelligent systems.