Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.011988
Article

Tech Science Press

Deep 3D-Multiscale DenseNet for Hyperspectral Image Classification Based on Spatial-Spectral Information

Haifeng Song[1], Weiwei Yang[1,*], Haiyan Yuan[2] and Harold Bufford[3]

[1]School of Electronics and Information Engineering (School of Big Data Science), Taizhou University, Taizhou, 318000, China
[2]College of Science, Heilongjiang Institute of Technology, Harbin, 150050, China
[3]Departments of Interactive Technology, Animax Designs, Nashville, 37207, USA
[*]Corresponding Author: Weiwei Yang. Email: yww_1680@163.com
Received: 09 June 2020; Accepted: 07 July 2020

Abstract: There are two main problems that lead to unsatisfactory classification performance for hyperspectral remote sensing images (HSIs). One issue is that the HSI data used for training in deep learning is insufficient, therefore a deeper network is unfavorable for spatial-spectral feature extraction. The other problem is that as the depth of a deep neural network increases, the network becomes more prone to overfitting. To address these problems, a dual-channel 3D-Multiscale DenseNet (3DMSS) is proposed to boost the discriminative capability for HSI classification. The proposed model has several distinct advantages. First, the model consists of dual channels that can extract both spectral and spatial features, both of which are used in HSI classification. Therefore, the classification accuracy can be improved. Second, the 3D-Multiscale DenseNet is used to extract the spectral and spatial features which make full use of the HSI cube. The discriminant features for image classification are extracted and the spectral and spatial features are fused, which can alleviate the problem of low accuracy caused by limited training samples. Third, the connections between different layers are established using a residual dense block, and the feature maps of each layer are fully utilized to further alleviate the vanishing gradient problem. Qualitative classification experiments are reported that show the effectiveness of the proposed method. Compared with existing HSI classification techniques, the proposed method is highly suitable for HSI classification, especially for datasets with fewer training samples. The best overall accuracy of 99.36%, 99.86%, and 99.99% were obtained for the Indian Pines, KSC, and SA datasets, which showed an effective improvement of the classification accuracy.

Keywords: Deep neural network; residual dense network; spectral-spatial feature extraction; 3D-Multiscale; hyperspectral image classification

1 Introduction

Hyperspectral image classification refers to the process of marking unlabeled pixels. For this, classification algorithms can be divided into two categories: Algorithms based on spectral-spatial features and algorithms based on deep learning.

HSI classification algorithms based on spectral-spatial features refers to the use of both spectral and spatial features. The introduction of spatial features in the classification process is due to the phenomena of "different objects with the same spectrum" and "different spectra of the same objects". In order to alleviate this problem, many scholars begin to consider spatial features. A large number of researchers have shown that combining spatial features can effectively improve the classification accuracy [1–3]. The most representative classification algorithm based on spectral-spatial features is the Composite Kernel (CK) classification algorithm. However, the traditional CK algorithm is prone to misclassification on the boundary of HSI. Menon et al. improved the CK algorithm [2] and proposed a combined kernel HSI classification algorithm based on the nearest neighbor domain. Tabalka et al. proposed an HSI classification algorithm based on Markov random fields (MRF) and SVMs [4]. A probabilistic SVM [5] is used to process the original HSI, and the probability of a pixel belonging to each category is obtained. This algorithm has good accuracy for homogeneous regions, but the pixels in the edge regions and isolated pixels are easily misclassified.

In recent years, many researchers have made great breakthroughs in the field of deep learning. Deep learning is widely used in the field of computer vision. Zhang et al. [6] proposed a lightweight deep network for traffic sign classification. Wang et al. [7] improved the traditional convolutional neural network and proposed a new image classification model. Zhang et al. [8] extracted spatial and semantic convolutional features for robust visual object tracking. Deep learning can also be used in the information safety field, for example, it can be used in image information hiding [9] and packet inspection [10,11]. In the field of intelligent medical treatment [12] and natural language processing [13], deep learning algorithms have also achieved fruitful results.

Among the numerous algorithms based on deep learning, Convolutional Neural Networks (CNNs) [14] are the most representative classification methods. CNNs have been widely used in HSI classification [15,16]. Although CNN models have been used for HSI classification and achieved state-of-the-art results, it is counterintuitive that the classification accuracy decreases with the increase of convolutional layers after four or five stacked layers [17]. Inspired by the latest deep residual learning framework proposed in [18], this issue can be addressed by adding shortcut connections between every other layer and propagating the value of features. Residual Dense Networks can be regarded as an extension of Convolutional Neural Networks with skip connections that facilitate the propagation of gradients and perform robustly with very deep architecture.

In this paper, we proposed a deep 3D-Multiscale DenseNet (3DMSS) for HSI classification based on spectral-spatial information. Our developments mainly consist of three aspects. First, the model consists of dual channels which can extract both the spectral and spatial features, improving the classification accuracy. Second, the discriminant spectral-spatial features for image classification are extracted and the spectral and spatial features are fused, alleviating the problem of low accuracy caused by limited training samples. Third, the connections between different layers are established using a residual dense block, and the feature maps of each layer are fully utilized to further alleviate the vanishing gradient problem.

2 Related Work

2.1 Deep CNN

A CNN is usually composed of several convolutional layers, pooling layers, and fully connected layers which result in a deep network architecture. Therefore, a CNN is able to deal with more complex classification and recognition problems and achieve excellent results.

Specifically, the training sample set is assumed to be $X = \{x^{(1)}, x^{(2)}, \cdots, x^{(m)}, \cdots, x^{(M)}\}$ and its corresponding labeled sample set. For each convolutional layer l, all feature maps are summed by the

convolution operation of the previous layer's feature map with a convolutional kernel. The calculation of the feature map is shown in Eq. (1):

$$x^{(l,s)} = f\left(\sum_{t=1}^{N_{l-1}} x^{(l-1,t)} \bullet k^{(l,s,t)} + b^{(l,s)}\right) \tag{1}$$

where $x^{(l,s)}$ is the *sth* feature map of the *lth* layer, N_{l-1} is the number of the previous layer's feature map, and $k^{(l,s,t)}$ $b^{(l,s)}$ are the convolutional kernel and corresponding bias terms, respectively.

The input feature map is down-sampled by the pooling layer to realize scale-invariance. The number of feature maps is unchanged. The down-sampling operation is shown in Eq. (2):

$$x^{(l,s)} = g(\beta^{(l,s)} \bullet down(x^{(l-1,s)}) + b^{(l,s)}) \tag{2}$$

where $down(\bullet)$ is the pooling function, β is the multiplicative bias, and b is the additive bias. According to this formula, each output feature map of the pooling layer is the down-sampling of its corresponding input feature map.

The mean square error is the energy function of the whole network, as shown in Eq. (3):

$$J(W,b) = \frac{1}{2M}\sum_{m=1}^{M} \left\| y^{(m)} - z^{(m)} \right\|_2^2 \tag{3}$$

where $z^{(m)}$ is the actual output.

In practice, the overall accuracy of the convolutional neural network is related to the depth of the network. In general, the accuracy of the model is improved by increasing the network depth, but at a certain point the overall accuracy will decrease if the network depth continues to increase. The main reason is that the deeper the network, the more likely it is to encounter the vanishing gradient problem, and it is easy to fall into a local minimum. Therefore, it is difficult to make full use of the feature extraction ability of the deep network by directly stacking shallow layers into a deep network.

2.2 Deep Residual Networks

To address the gradient degeneration problem, He proposed the Residual Neural Network (ResNet) [19]. The residual block is the basic architecture of ResNet; a residual neural network is composed of several residual blocks, as shown in Fig. 1:

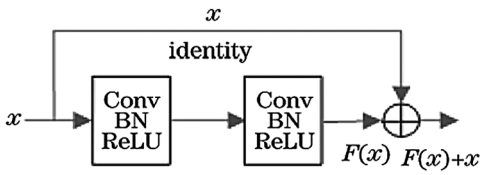

Figure 1: Structure of a residual block

Here, x represents the input data. For a network with no short connections, the output is $F(x)$; for a residual block with short connections, the output is $H(x)$, where $H(x) = F(x) + x$. Experimental results show that it is much easier to optimize the residual mapping $F(x)$ than the original function mapping $H(x)$ in the residual block. $H(x)$ can be understood as the sum of the residual mapping $F(x)$ and the identity mapping x in the network. Identity mapping neither increases the number of parameters nor

affects the complexity of the original network. In the figure, Conv represents the convolution operation, BN represents batch normalization, and ReLU represents the activation function.

ResNet has one more shortcut connection than a traditional neural network. From the perspective of feature flow, it enables features to be transferred directly to the next layer. When the layers of the neural network are very deep, there are still lower features that enhance the higher features, so that the features can be introduced deeper. From the perspective of backpropagation, when the output changes a small amount, the gradient $\frac{\partial H(x)}{\partial x}$ will be very small, which is extremely difficult for directly learning $H(x)$. However, because the difference is calculated in ResNet training $F(x)$, which amplifies the slight change, the gradient becomes $\frac{\partial H(x)}{\partial x} - 1$. The absolute value of the gradient becomes larger, the training process continues, and the degradation problem is solved.

Recently, He et al. built a random depth architecture based on a 1202-layer ResNet [19]. However, they found that randomly discarding the ResNet layer did not change the convergence in training. This phenomenon indicates that ResNet does not make full use of the output feature by each convolutional layer in the residual block, and also ignores the connection between any two convolutional layers. Meanwhile, the mode of adding layers is not conducive to the transmission of features in the network.

2.3 Deep DenseNet

Huang et al. [20] proposed the DenseNet model. DenseNet is able to connect any two convolutional layers in a dense cell, realizing feature reuse and feature transfer. DenseNet is based on a residual dense block, which is composed of several convolutional layers and activation layers, and plays the role of feature extraction. The output of each block will establish a short connection with the output of each convolutional layer of the next block, realizing continuous feature transmission. The structure of a residual network is shown in Fig. 2:

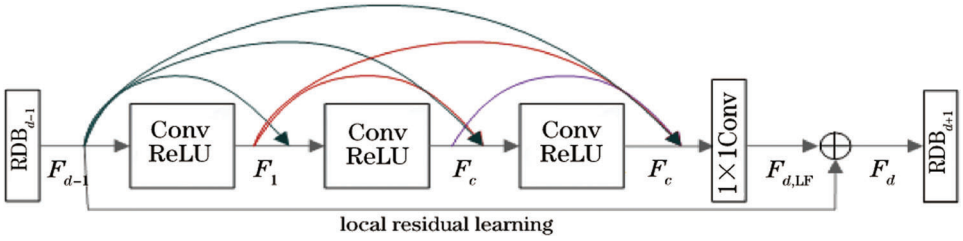

Figure 2: Illustration of a residual dense block

Assume that the input and output of the *dth* block are F_{d-1} and F_d, respectively. The number of input and output feature maps is G_0. The output of the *cth* convolutional layer can be represented as in Eq. (4):

$$F_{d,c} = H\{[F_{d-1}, F_{d,1}, \cdots, F_{d,c-1}]\} \tag{4}$$

where $H(\bullet)$ is a nonlinear operation of the convolutional layer, including convolution and ReLU functions. Let $F_{d,c}$ output G feature maps representing the connection between the feature maps output by the previous block and the feature graph output by the $c-1$ convolutional layer before the block, containing $G_0 + (c-1) \times G$ feature maps in total.

Since full connection is adopted between the input layer of the block and the convolutional layer, it is necessary to compress the feature maps at the end of the block. Therefore, 1×1 convolution is adopted to control the number of feature maps, which can be represented as in Eq. (5):

$$F_{d,LF} = H_{LEF}^{d}\{[F_{d-1}, F_{d,1}, \cdots, F_{d,c}]\} \tag{5}$$

where H_{LEF}^{d} represents the 1×1 convolution operation. The final output of the block is shown in Eq. (6):

$$F_d = F_{d-1} + F_{d,LF} \tag{6}$$

Local residual learning is calculated by adding the output and input of the block, which further preserves a large amount of image detail and improves the feature extraction performance of the residual dense block.

3 Proposed Frameworks

First, we introduce how to apply 3D convolution for HSI. Second, we give a general introduction to the 3DMSS model proposed in this paper: the input of the model is the original HSI data and the output of the model is the classification results of the corresponding pixel. Then, according to the process of 3DMSS, the 3D-multiscale spectral and spatial DenseNet channels, feature fusion, and classification are introduced in detail. Finally, the training and optimization process of the model is introduced.

3.1 3D-Multiscale Convolutional Network

HSI is a 3D cube with rich spectral-spatial features. As a result, the 3D convolution operation [21] is adopted to extract spectral and spatial features. The 3D convolution operation is shown in Fig. 3.

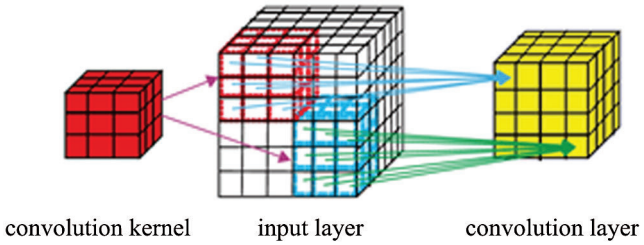

Figure 3: Illustration of the 3D convolutional operation

As we can see from this figure, the input data is a 3D image composed of spectral and spatial dimensions. Therefore, the convolution kernel performs the convolution operation on both spectral and spatial dimensions of the input 3D image. One pixel at a time is obtained in the 3D image by the convolutional operation, and a new 3D feature map is obtained after the processing of the whole image. The calculation is shown in Eq. (7).

$$x_{i,j}^{x,y,z} = \sigma\left(\sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_{i-1}} W_{i,j,m}^{p,q,r} X_{(i-1),m}^{x+p,y+q,z+r} + b_{ij}\right) \tag{7}$$

Here, $x_{i,j}^{x,y,z}$ is the output value of the *jth* feature map at position (x, y, z) of the *ith* layer. m is the set of feature maps connected to the current feature graph at the $(i-1)th$ layer, $W_{i,j,m}^{p,q,r}$ is the weight of the position (p, q, r) of the 3D convolution kernel in the *mth* feature map, and $b_{i,j}$ is the bias. $\sigma(\bullet)$ is the activation function. P_i, Q_i, R_i is the length, width, and height of the convolutional kernel, respectively.

HSI is characterized by large data volumes but with limited data for training. The features learned by the convolution kernel with a fixed scale are not conducive to the training of the model. Therefore, a multi-scale network is used to learn features at different scales, extract more discriminative features, and improve feature extraction for small sample data. HSI classification by the 3D-multiscale network can alleviate the problem of low accuracy caused by limited training samples.

3.2 Overview of 3DMSS

HSI is three-dimensional data, including one-dimensional spectral data and two-dimensional spatial data. Although HSI contains abundant spectral information, there are many bands with high correlation between adjacent bands and data redundancy. Since spectral and spatial information play important roles in HSI classification, spectral and spatial dimensions should be considered in feature extraction. HSI features are extracted by using a 3D convolutional kernel [22,23]. Although these methods improve the classification accuracy, they do not fully extract discriminative spectral-spatial features.

In order to predict the category of ground objects, we propose 3D-Multiscale Spectral-Spatial DenseNet. A convolution kernel of $1 \times 1 \times 5$ and $1 \times 1 \times 7$ is chosen to extract spectral features, and a convolution kernel of $3 \times 3 \times 1$ and $5 \times 5 \times 1$ is chosen to extract spatial features. In the network, spectral and spatial features are extracted continuously, and more discriminative spectral-spatial features are used for classification. The application of multi-scale networks can alleviate the problem of limited training samples. In addition, the feature fusion module is embedded in the multi-scale network. The 3DMSS approach shares feature information of different scales to enhance the information flow of the network, which is conducive to the extraction of spectral-spatial features and improves the classification accuracy. The model of the network is shown in Fig. 4.

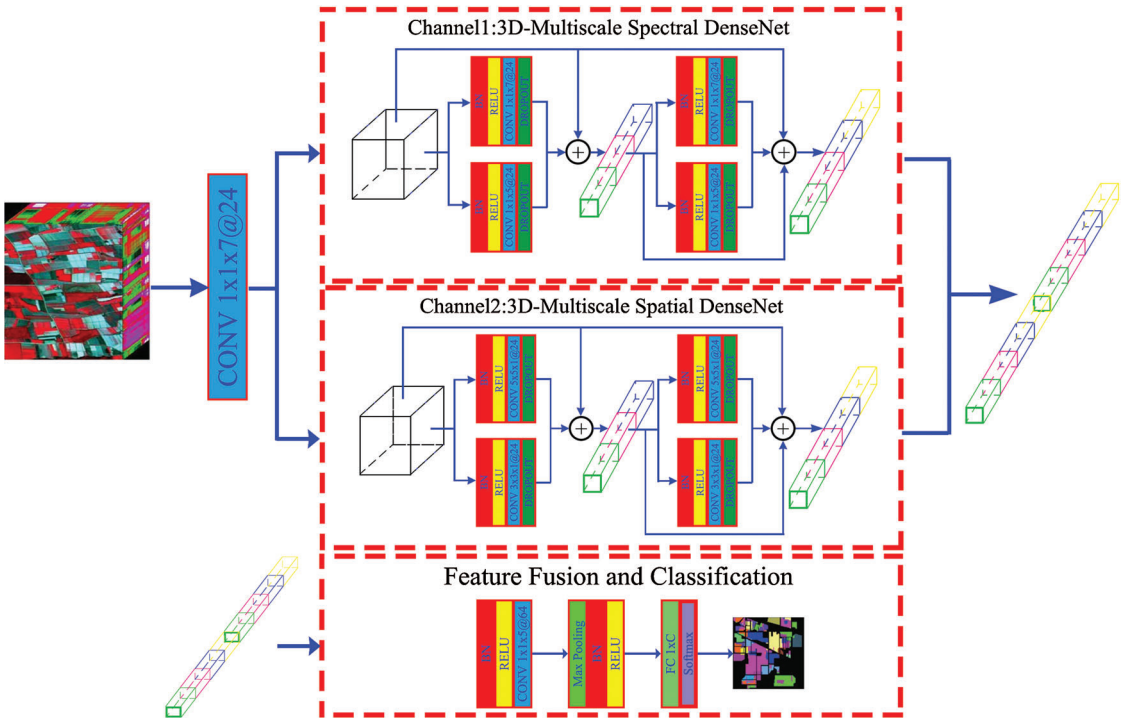

Figure 4: The architecture of 3DMSS

3.3 Channel 1: 3D-Multiscale Spectral DenseNet

3D-Multiscale Spectral DenseNet is shown in Fig. 5. In the training process, for the purpose of dimensionality reduction, the convolution operation is carried out using 24 convolutional kernels with a step size of 2 for the original HSI. The 3D feature map after dimensionality reduction is used as the input to the spectral feature extraction channel.

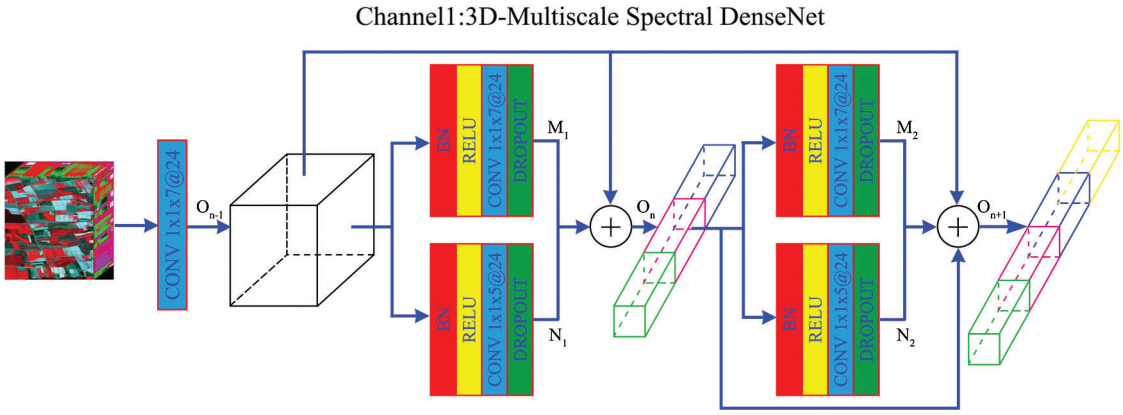

Figure 5: 3D-Multiscale Spectral DenseNet

In 3D-Multiscale Spectral DenseNet, the multi-scale features of the spectral domain are extracted by using the K convolution kernels with of size p and q, respectively. This is shown in Eqs. (8) and (9):

$$M_1 = \sigma(w^1_{1\times1\times p} \otimes O_{n-1} + b^1_{1\times1\times p}) \tag{8}$$

$$N_1 = \sigma(w_{1\times1\times q} \otimes O_{n-1} + b^1_{1\times1\times q}) \tag{9}$$

where O_{n-1} is the input feature map of the 3D-Multiscale Spectral DenseNet. $\otimes$ is the convolutional operation, w is the weight of the convolution kernel, and b is the bias. The superscripts of w and b are the number of convolutional layers and the subscripts are the size of the convolutional kernel. $\sigma(\bullet)$ is the activation function.

Shallow spectral features at two scales were extracted, and O_n was obtained by fusing the K feature maps (a total of $2 \times K$ feature maps) learned at each scale and the original input. This is shown in Eq. (10).

$$O_n = O_{n-1} + M_1 + N_1 \tag{10}$$

Then, K spectral convolutional kernels of different scales are used to carry out the multi-scale convolution operation on O_n, as shown in Eqs. (11) and (12):

$$M_2 = \sigma(w^2_{1\times1\times p} \otimes O_n + b^2_{1\times1\times p}) \tag{11}$$

$$N_2 = \sigma(w^2_{1\times1\times q} \otimes O_n + b^2_{1\times1\times q}) \tag{12}$$

where the meanings of each variable are the same as in formula (8) and formula (9). The discriminant spectral feature diagram O will be learned after the extraction of spectral features, as shown in Eq. (13).

$$O_{n+1} = O_n + O_{n-1} + M_2 + N_2 \tag{13}$$

3.4 Channel 2: 3D-Multiscale Spatial DenseNet

3D-Multiscale Spatial DenseNet is shown in Fig. 6. In the training process, to achieve the purposes of the dimension reduction, the convolution operation is carried out by using 24 convolutions kernels with a step size of 2 to the original HSI. The 3D feature map after dimension reduction is used as the input data of spatial feature extraction channel.

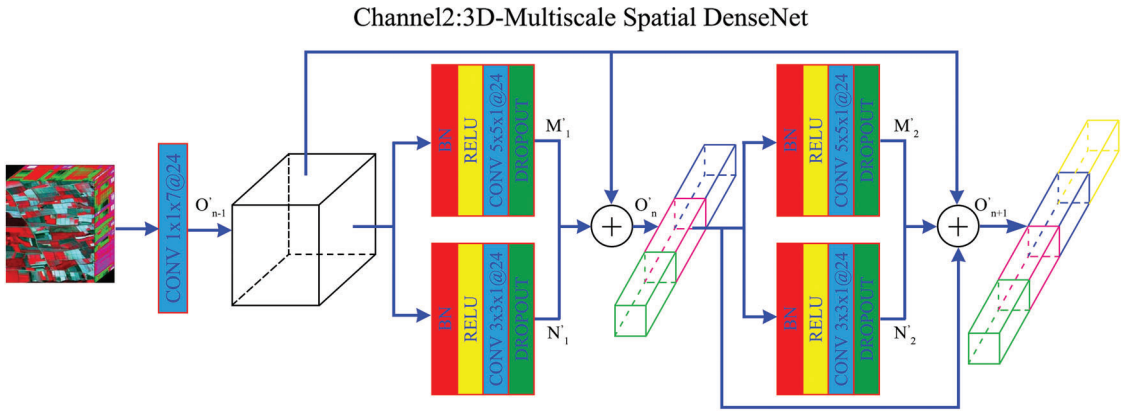

Figure 6: 3D-Multiscale Spatial DenseNet

In 3D-Multiscale Spatial DenseNet, the multi-scale features of spatial domain are extracted by using the $K^{'}$ convolution kernels with sizes of $p^{'}$ and $q^{'}$, respectively. As shown in Eqs. (14) and (15):

$$M_1^{'} = \sigma(w^1_{p^{'}\times p^{'}\times 1} \otimes O^{'}_{n-1} + b^1_{p^{'}\times p^{'}\times 1}) \tag{14}$$

$$N_1^{'} = \sigma(w^1_{q^{'}\times q^{'}\times 1} \otimes O^{'}_{n-1} + b^1_{q^{'}\times q^{'}\times 1}) \tag{15}$$

where, $O^{'}_{n-1}$ is the input feature map of the 3D-Multiscale Spectral DenseNet. "$\otimes$" is the convolutional operation. w is the weight of the convolution kernel. b is the bias. The superscript of w and b is the number of convolutional layers, and the subscript is the size of the convolutional kernel. $\sigma(\bullet)$ is the activation function.

The shallow spatial features at two scales were extracted, and the $O^{'}_n$ is obtained by fusing the $K^{'}$ feature maps (a total of $2 \times K^{'}$ feature maps) learned at each scale and the original input. As shown in Eq. (16):

$$O^{'}_n = O^{'}_{n-1} + M^{'}_1 + N^{'}_1 \tag{16}$$

Then, $K^{'}$ spatial convolution kernels of different scales are used to carry out multi-scale convolution operation on $O^{'}_n$, as shown in Eqs. (17) and (18):

$$M_2^{'} = \sigma(w^2_{5\times 5\times 1} \otimes O^{'}_n + b^2_{5\times 5\times 1}) \tag{17}$$

$$N_2^{'} = \sigma(w^2_{3\times 3\times 1} \otimes O^{'}_n + b^2_{3\times 3\times 1}) \tag{18}$$

The meanings of each variable are the same as formula (14) and formula (15).

The discriminant spectral feature diagram O will be learned after the extraction of spectral features, as shown in Eq. (19):

$$O^{'}_{n+1} = O^{'}_n + O^{'}_{n-1} + M^{'}_2 + N^{'}_2 \tag{19}$$

3.5 Feature Fusion and Classification

As shown in Fig. 7, the results of spectral and spatial learning are concatenated as input followed by a BN, RELU, and convolution layer block, which is the same as the process for Block 2.

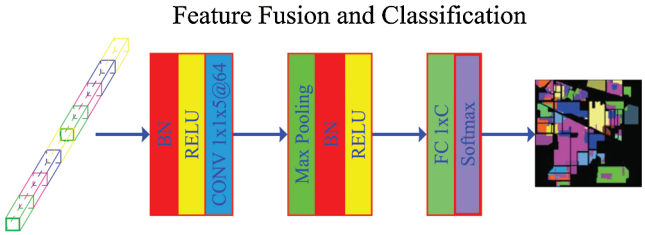

Figure 7: Feature fusion and classification

At the end of the block, global average pooling layers are inserted. It was originally designed to replace the traditional FC layer in CNNs. The global average pooling layer contains a much smaller number of parameters than FC layers and can retain good localization ability for a network. It is important to consider two main problems in HSI classification: the overfitting phenomenon caused by the large model scale with limited training data, and the effective extraction of both spectral and spatial features. After the FC layer, a softmax layer is used to obtain the final classification result.

4 Experimental Results and Discussion

We evaluated the performance of the proposed network on three publicly available HSI datasets. First, the main components of 3DMSS are tested, including the number of kernels, the depth of the spectral kernel, the size of the spatial kernel and the number of training samples. Then, the proposed classification model is compared with mainstream approaches in terms of the overall accuracy (OA), average accuracy (AA), and kappa coefficient (K). These are adopted to qualitatively evaluate the classification results.

4.1 Description of the Experimental Data Sets

Three datasets were used: Indian Pines (IN), Kennedy Space Center (KSC), and Salinas (SA). The Indian Pines dataset contains 220 spectral channels and the spatial resolution is 20 m. Each band contains 145 × 145 pixels. The sample size is shown in Tab. 1. The KSC dataset contains 224 spectral channels and 13 land cover categories; the sample size is shown in Tab. 2. The Salinas dataset contains 224 spectral channels, and the spatial resolution is 3.7 m. The sample size is shown in Tab. 3.

Table 1: Indian Pines data sample distribution

No.	Class	No. of Samples
1	Alfalfa	54
2	Corn-notill	1434
3	Corn-mintill	834
4	Corn	234
5	Grass/pasture	497
6	Grass/tree	747
7	Grass/pasture/mowed	26
8	Hay/Windrowed	489
9	Oats	20
10	Soybean-notill	968

(Continued)

Table 1 (continued).

No.	Class	No. of Samples
11	Soybean-mintill	2468
12	Soybean-clean	614
13	Wheat	212
14	Woods	1294
15	Buildings/grass/trees/drives	95
16	Stone/steel/towers	380
	Total	10366

Table 2: KSC data sample distribution

No.	Class	No. of Samples
1	Scrub	530
2	Willow swamp	165
3	CP hammock	176
4	Slash pine	170
5	Oak/Broadleaf	110
6	Hardwood	161
7	Swap	80
8	Graminoid marsh	299
9	Spartina marsh	377
10	Cattail marsh	283
11	Salt marsh	296
12	Mud flats	341
13	Water	654
	Total	3642

4.2 Experimental Setup for the Classification of Labeled Pixels

To set the parameters of 3DMSS, we determined the optimal parameters through a series of experiments, which included the number of convolution kernels, the convolution kernels' depth of spectral feature channels, the convolution kernels' size of spatial feature channels, and the number of training samples in each batch.

4.2.1 Effect of the Number of Kernels

This experiment analyzes the effect of the number of convolution kernels on the classification results. For the experimentation, the number of convolution kernels of each residual dense block on Channel 1 and Channel 2 was set to 6, 12, 24, 48, and 64, respectively. The classification accuracy for different numbers of kernels was recorded.

Table 3: Indian Pines data sample distribution

No.	Class	No. of Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807
	Total	54129

Fig. 8 shows the experimental results. It can be seen that, under certain conditions, increasing the number of convolution kernels can improve the classification accuracy. However, the classification accuracy does not increase linearly with the increase of convolution kernels. With the increase of the number of kernels, the classification accuracy rises first and then flattens out. The experimental results show that the classification accuracy is highest when the number of kernels is 24. It can also be seen that as the number of convolution kernels increases, the computational complexity of the model increases and the time required for classification increases. Therefore, considering the classification accuracy and time complexity, the number of convolution kernels in the convolutional layer is set to 24.

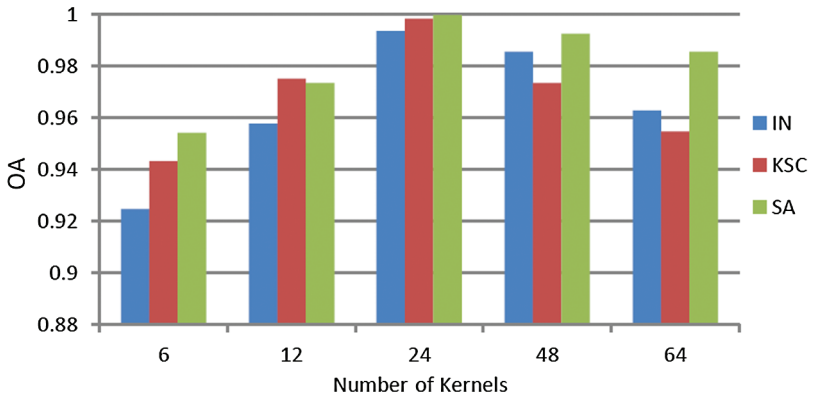

Figure 8: Classification results for each dataset for different kernels

4.2.2 Effect of Different Spectral Kernel Depths

Tab. 4 shows the classification accuracy results for different convolutional kernel depths for 3D-Multiscale Spectral DenseNet. As can be seen from the table, the OA, AA, and Kappa coefficients increased with the increase of convolution kernel depth. As the depth increases $1 \times 1 \times 7$, the accuracy increases slowly or stops increasing. Therefore, the selected convolutional kernel depths for the 3D-Multiscale Spectral DenseNet were $1 \times 1 \times 5$ and $1 \times 1 \times 7$.

Table 4: OA comparison for different spectral kernel depths

Kernels Depth	IN			SS			KSC		
	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
$1 \times 1 \times 3$	96.59	93.37	0.9623	97.29	96.98	0.9549	96.39	95.29	0.9478
$1 \times 1 \times 5$	98.79	95.36	0.9865	99.99	99.99	0.9998	99.86	99.80	0.9984
$1 \times 1 \times 7$	99.36	95.76	0.9927	99.35	99.18	09916	99.45	99.06	0.9883
$1 \times 1 \times 9$	97.28	94.13	0.9694	98.43	98.38	0.9799	97.26	96.72	0.9613

4.2.3 Effect of Different Spatial Kernel Size

Tab. 5 shows the classification accuracy results for different convolutional kernel sizes in 3D-Multiscale Spatial DenseNet. As we can see from the table, the OA, AA, and Kappa coefficients increased with the increase of convolution kernel size. As the size increases 5×5, the accuracy increases slowly or stops increasing. Therefore, convolutional kernel sizes 3×3 and 5×5 were selected based on the main evaluation indexes.

Table 5: OA comparison for different spatial kernel size

Kernels Depth	IN			SS			KSC		
	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
3×3	99.36	95.76	0.9927	99.79	99.68	99.64	99.54	99.28	98.85
5×5	99.11	95.63	0.9887	99.99	99.99	99.98	99.86	99.80	99.84
7×7	98.24	94.17	0.9779	98.35	97.89	97.32	97.54	97.25	96.41
9×9	95.32	92.44	0.9389	96.13	95.78	95.31	95.39	94.32	93.76

4.3 Classification Results and Discussion

In order to verify the classification performance of 3DMSS proposed in this paper, we compared this with five other classical HSI classification methods on the basis of the OA, AA, and Kappa coefficients. These five methods include: A Support Vector Machine method [6], a Gabor-based method (GABOR) [24], the Image Fusion and Recursive Filtering method [25], 3D-CNN [17], and MS3FE [26]. Tabs. 6–8 show the test results of each method for three datasets. Figs. 9–11 are the visual maps of the different methods on the three datasets.

From Tabs. 6–8, it can be seen that the proposed method attains the best classification performance for the three datasets. The OA for the three datasets is 99.36%, 99.86%, and 99.99%, respectively. Figs. 9–11 are the visual maps of the different methods for the three datasets. It can be seen from the figures that the visual

maps for SVM, GABOR, and 3D-CNN have noise and fuzzy classification. The visual maps for RF and MS3FE have clear classification boundaries, but there is still a small amount of noise. The visual maps for 3DMSS are the clearest, and the classification result is the closest to the real object label.

Table 6: Testing of the different methods for the IN dataset

Class	SVM	GABOR	IFRF	3D-CNN	MS3FE	3DMSS
1	34.65	92.71	94.53	60.98	95.47	100
2	65.38	92.76	92.90	78.60	88.84	99.13
3	43.87	88.28	93.05	87.42	93.78	98.26
4	34.64	93.23	90.07	88.32	92.87	99.58
5	81.08	89.72	92.72	80.60	92.31	99.08
6	93.16	91.25	99.34	92.98	98.89	99.44
7	65.19	84.23	98.46	68.00	96.54	100
8	95.20	97.93	99.67	95.57	99.22	100
9	34.17	83.33	88.89	77.78	100	40.00
10	61.16	91.54	92.38	76.91	92.32	99.19
11	78.29	93.92	96.33	84.42	98.72	99.57
12	44.77	91.27	91.93	82.52	92.78	99.00
13	97.40	93.35	99.10	96.20	98.69	100
14	95.74	96.87	98.28	99.30	99.98	99.81
15	42.33	95.30	93.96	89.94	99.46	99.74
16	85.34	87.35	98.30	85.54	93.81	99.37
OA	65.77	91.44	94.99	84.13	95.85	99.36
AA	72.04	93.00	95.22	86.43	95.71	95.76
Kappa	0.6775	0.9203	0.9455	0.8450	0.9510	0.9927

Table 7: Testing of the different methods for the KSC dataset

Class	SVM	GABOR	RF	3D-CNN	MS3FE	3DMSS
1	81.91	81.45	83.93	91.50	94.10	99.92
2	74.59	38.45	67.30	100	93.24	98.71
3	82.99	71.32	99.51	85.59	98.70	99.83
4	40.21	38.86	77.87	60.34	97.62	99.06
5	40.36	75.26	97.15	100	86.49	99.82
6	51.16	43.15	86.67	94.26	99.82	100
7	78.53	86.63	99.32	100	100	100
8	73.97	69.39	95.12	85.89	93.57	100
9	81.14	84.34	89.73	73.80	94.63	100

(Continued)

Table 7 (continued).

Class	SVM	GABOR	RF	3D-CNN	MS3FE	3DMSS
10	79.12	62.61	94.43	99.48	88.85	100
11	91.49	88.25	99.27	93.23	96.32	100
12	86.90	58.50	91.45	83.23	96.98	100
13	99.91	71.28	100	100	100	100
OA	74.02	66.89	90.90	89.79	95.41	99.86
AA	80.58	69.54	91.62	89.90	95.71	99.80
Kappa	0.7840	0.6630	0.8909	0.8876	0.9523	0.9984

Table 8: Testing of the different methods for the SA dataset

Class	SVM	GABOR	IFRF	3D-CNN	MS3FE	3DMSS
1	96.81	94.58	100	78.68	97.97	100
2	91.91	95.08	97.87	75.50	97.40	100
3	88.71	94.14	99.99	97.09	99.63	100
4	99.28	96.43	98.94	99.85	99.45	99.91
5	95.97	88.24	95.93	97.07	97.61	99.98
6	99.74	99.55	99.60	99.85	99.81	100
7	99.51	94.22	99.02	99.72	99.94	100
8	61.09	65.82	87.62	29.83	92.50	99.96
9	97.85	97.39	99.99	95.05	98.31	1
10	76.73	90.64	98.40	92.33	92.21	99.98
11	91.55	92.25	96.61	1000	97.37	100
12	97.52	99.03	97.13	99.42	99.18	100
13	95.27	98.43	97.13	99.96	96.69	100
14	91.30	92.41	96.75	100	94.52	100
15	58.03	76.73	97.00	86.81	96.15	100
16	91.92	89.45	95.77	94.18	99.53	100
OA	89.57	91.53	97.36	90.15	97.39	99.99
AA	82.45	86.32	96.02	79.49	96.57	99.99
Kappa	0.8052	0.8483	0.9558	0.7746	0.9619	99.98

Analyzing the above experimental results, we can draw the following conclusions:

1. The higher the spatial resolution of HSI, the better the classification performance is achieved for the larger convolutional kernel size. The spatial resolution of IN, KSC, and SA is 145 × 145, 512 × 614, and 512 × 217, respectively. Since the resolution of IN is the smallest, IN achieves the best classification accuracy for the convolution kernel size of 3 × 3. The spatial resolution of KSC and

SA is greater than IN, so they achieve the best classification accuracy for the slightly larger convolution kernel size 5 × 5.

2. The more spectral bands, the better the classification results for the deeper convolutional kernel. The number of spectral bands for IN, KSC, and SA is 200, 176, and 184, respectively. IN has the most spectral bands, so IN achieves the highest classification accuracy with a convolution kernel depth of 1 × 1 × 7. KSC and SA have fewer spectral bands, so they achieve the highest classification accuracy with a convolution kernel depth of 1 × 1 × 5.
3. Deep learning methods are superior to statistical methods. Among the six compared methods, SVM, Gabor, and RF are traditional classification methods based on statistics. 3D-CNN, MS3FE, and 3DMSS are deep learning methods, and all three use convolutional neural networks. It can be seen from the experimental results that the classification performance for deep learning methods is better than that for statistical methods.
4. Spectral-spatial features help to improve the classification accuracy. Since 3DMSS and MS3FE take into account the spectral-spatial features of HSI, the OA obtained by these two classification methods is significantly higher than that of other methods.
5. The classification results for the residual dense network are better than for other methods. Compared with other classification methods, 3DMSS achieved the best classification results. This is because 3DMSS can extract HSI spectral-spatial features from different scales so that the features of different channels can be shared, and the information flow can be enhanced. At the same time, in order to improve the classification performance, the residual dense block is introduced into 3DMSS to overcome the vanishing gradient problem.

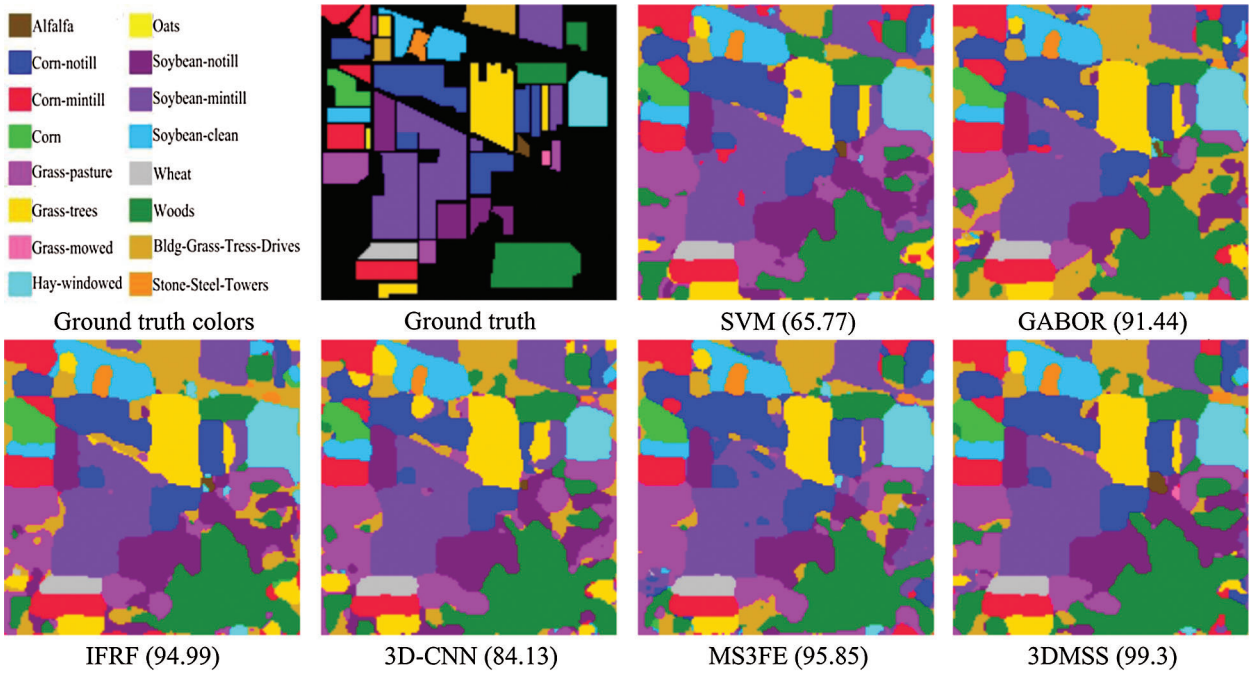

Figure 9: Experimental results for the different methods for the IN dataset

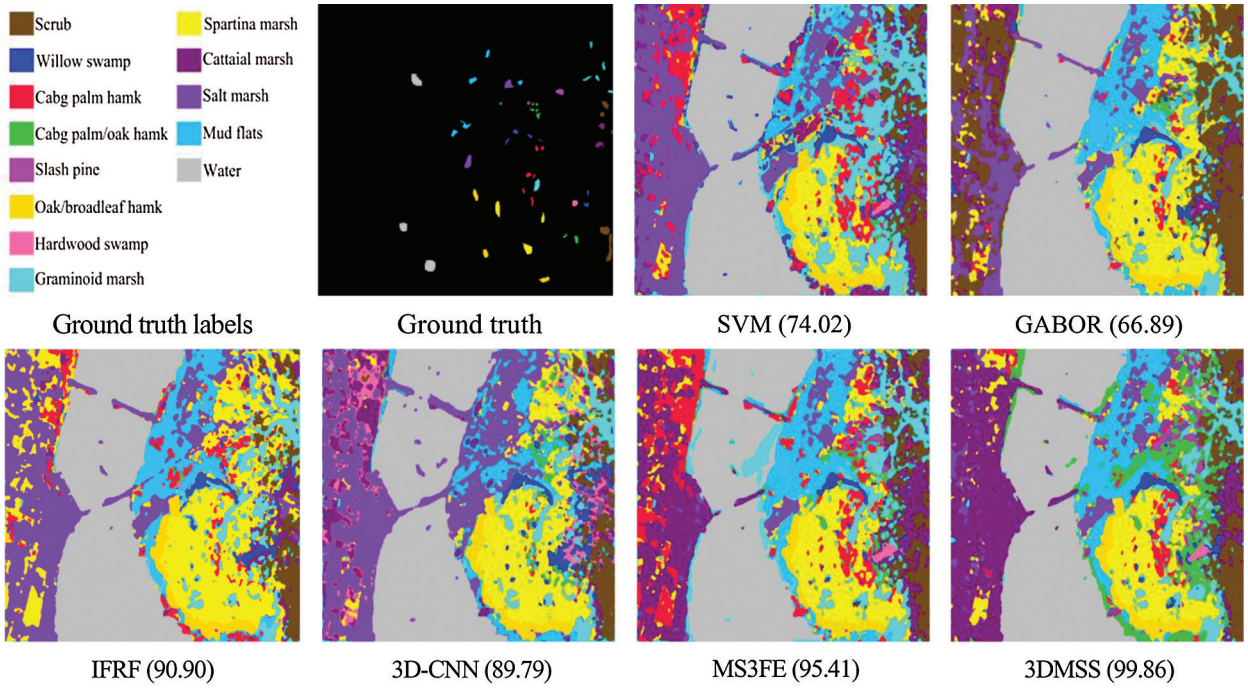

Figure 10: Experimental results for the different methods for the SA dataset

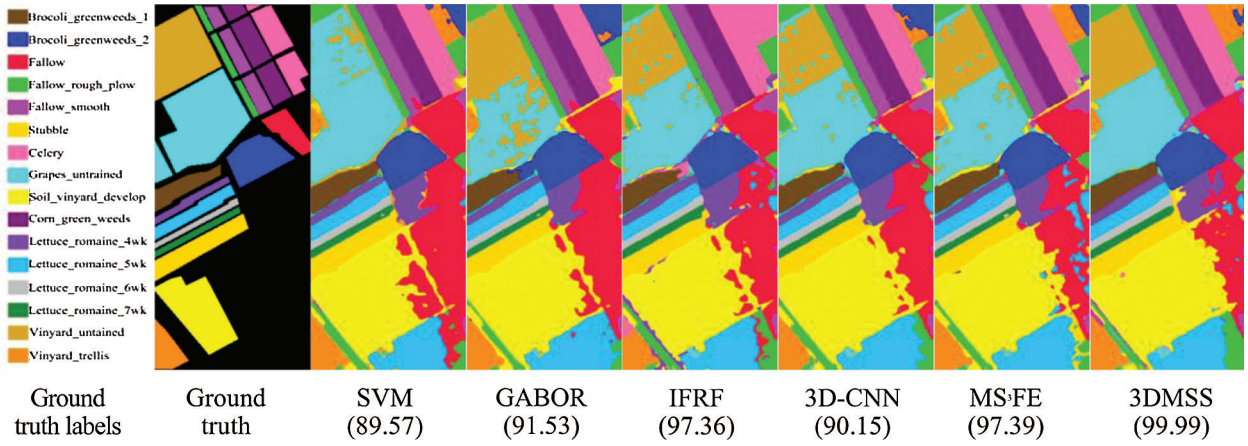

Figure 11: Experimental results for the different methods for the SA dataset

5 Conclusion

In order to improve the classification performance for HSI, an end-to-end deep 3D-Multiscale Spatial-Spectral DenseNet was proposed in this paper. The work was proposed to handle the problems associated with HSI data such as multiple bands, data redundancy, and limited training samples. The discriminative spectral-spatial features were extracted using 3D-multiscale methods; the features of different blocks can be shared, and the information flow can be enhanced, which solves the problem of the lack of training samples. At the same time, in order to improve the classification performance, residual dense blocks are introduced into 3DMSS to address the vanishing gradient problem. Comparing the classification accuracy with available HSI classification methods for three public HSI datasets, the proposed method shows very promising results, and is highly effective. There is still plenty of scope to develop the proposed method,

such as more successful strategies in multi-scale feature fusion and robust classification accuracy for the boundary region. Also, a parallel and distributed fusion strategy, such as in [27,28], will be very helpful in improving the computational efficiency in practice.

Acknowledgement: We thank the anonymous reviewers for their feedback which helped in the improvement of this article.

Funding Statement: The work described in this paper is supported by the National Natural Science Foundation of China (Project No. 11901173), the Heilongjiang Province Natural Science Found (LH2019A030), and the Cultivating Science Foundation of Taizhou University (2019PY014, 2019PY015), the Agricultural Science and Technology Project of Taizhou (20ny13).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao *et al.,* "Spectral-spatial constraint hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1811–1824, 2014.

[2] V. Menon, S. Prasad and J. E. Fowler, "Hyperspectral classification using a composite kernel driven by nearest-neighbor spatial features," in *2015 IEEE International Conference on Image Processing*, Milan, Italy, pp. 2100–2104, 2015.

[3] Y. Tarabalka and A. Rana, "Graph-cut-based model for spectral-spatial classification of hyperspectral images," in *International Geoscience and Remote Sensing Symposium*, Quebec, Canada, pp. 3418–3421, 2014.

[4] Y. Tarabalka, M. Fauvel, J. Chanussot and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 736–740, 2010.

[5] Y. Wang, W. Yu and Z. Fang, "Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information," *Remote Sensing*, vol. 12, no. 1, pp. 120, 2020.

[6] M. Zhang, W. Wang, C. Q. Lu, J. Wang and A. K. Sangaiah, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications*, vol. 75, pp. 369–379, 2019.

[7] W. Wang, Y. Li, T. Zou, X. Wang, J. You *et al.,* "A novel image classification approach via Dense-MobileNet models," *Mobile Information Systems*, vol. 2020, pp. 1–8, 2020.

[8] M. Zhang, X. K. Jin, J. Sun, J. Wang and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, vol. 79, no. 21–22, pp. 15095–15115, 2018.

[9] J. Luo, J. H. Qin, X. Y. Xiang, Y. Tan, Q. Liu *et al.,* "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[10] X. Sun, L. F. Shi, C. Y. Yin and J. Wang, "An improved method in deep packet inspection based on regular expression," *Journal of Supercomputing*, vol. 75, no. 6, pp. 3317–3333, 2019.

[11] Y. Yin, H. Y. Wang, X. Yin, R. X. Sun and J. Wang, "Improved deep packet inspection in data stream detection," *Journal of Supercomputing*, vol. 75, no. 8, pp. 4295–4308, 2019.

[12] R. Zhou and B. Tan, "Electrocardiogram soft computing using hybrid deep learning CNN-ELM," *Applied Soft Computing*, vol. 86, pp. 105778, 2020.

[13] P. He, Z. L. Deng, C. Z. Gao, X. N. Wang and J. Li, "Model approach to grammatical evolution: Deep-structured analyzing of model and representation," *Soft Computing*, vol. 21, no. 18, pp. 5413–5423, 2017.

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[15] J. Leng, T. Li, G. Bai, Q. Dong and H. Dong, "Cube-CNN-SVM: A novel hyperspectral image classification method," in *ICTAI*, San Jose, CA, USA, pp. 1027–1034, 2016.

[16] Y. Li, H. Zhang and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sensing*, vol. 9, no. 1, pp. 67–78, 2017.

[17] 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 Transactions of Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[18] S. Wu, S. Zhong and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2017.

[19] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Nevada, USA, pp. 770–778, 2016.

[20] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, USA, pp. 2261–2269, 2017.

[21] Z. Xiong, Y. Yuan and Q. Wang, "AI-Net: Attention inception neural networks for hyperspectral image classification," in *International Geoscience and Remote Sensing Symposium*, Valencia, Spain, pp. 2647–2650, 2018.

[22] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang *et al.,* "Hyperspectral image classification with deep learning models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, 2018.

[23] Z. Zhong, J. Li, Z. Luo and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[24] L. Z. Huo and P. Tang, "Spectral and spatial classification of hyperspectral data using SVMs and Gabor textures," in *International Geoscience and Remote Sensing Symposium*, Vancouver, BC, Canada, pp. 1708–1711, 2011.

[25] X. Kang, S. Li and J. A. Benediktsson, "Feature extraction of hyperspectral images with image fusion and recursive filtering," *IEEE Transactions of Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3742–3752, 2014.

[26] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu *et al.,* "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 10, pp. 5585–5599, 2017.

[27] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao *et al.,* "Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2270–2278, 2016.

[28] W. Jing, S. Huo, Q. Miao and X. Chen, "A model of parallel mosaicking for massive remote sensing images based on spark," *IEEE Access*, vol. 5, pp. 18229–18237, 2017.