A multi-scale 3D convolution neural network for spectral-spatial classification of hyperspectral imagery

Wenchao Qi\textsuperscript{1,2}, Xia Zhang\textsuperscript{1*}

1. State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100101, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence: zhangxia@radi.ac.cn

Abstract. Along with the remarkable achievements of various neural networks in the field of computer vision, deep learning methods have gradually been applied to hyperspectral image (HSI) classification. Traditional classification methods have several outstanding issues, including the insufficient hand-craft features and time-consuming and laborious feature extraction. We therefore propose a multi-scale, 3D convolutional neural network (CNN) framework (MSCNN), trained in an end-to-end manner for hyperspectral image classification. Using the original hyperspectral image as an input, the MSCNN framework leverages two branches of a 3D residual neural network to extract deep and abstract HSI features. Then, the HSI spectral-spatial features of different scales are fused and fed into the SoftMax layer to achieve an end-to-end hyperspectral image classification. Besides, data augmentation, dynamic learning rate, and regularization methods are used to promote the rapid convergence of MSCNN and avoid model over-fitting. Three well-known hyperspectral datasets (Indian Pines, University of Pavia and Pavia Center) were used to evaluate the classification performance of the proposed MSCNN method. The results indicate that, compared with the existing deep learning methods, the proposed MSCNN method achieves the best classification performance within the 20 training epochs. The overall accuracy (OA), average accuracy (AA), and kappa statistic (K) of MSCNN are 97.16%, 98.69% and 0.9665, respectively, for Indian Pines; 99.17%, 98.97% and 0.9888 for University of Pavia; and 99.87%, 99.70% and 0.9981 for Pavia Center.

Keywords: 3D convolutional neural network, multi-scale, feature extraction, residual connection, hyperspectral image classification

1. Introduction

Hyperspectral images (HSI) are composed of hundreds of contiguous narrow bands, providing abundant spectral and spatial information of various objects. Hyperspectral data analysis has widespread utility in the field of remote sensing applications, such as eco-environmental monitoring [1, 2], land cover classification [3], and spectral unmixing [4]. Hyperspectral image classification, which labels each pixel of HSI, is a very active research field among these applications. Because of the higher dimensions and limited training samples available for HSI, dimension reduction, feature library building, and classifier training are major procedures in traditional HSI classification strategies.
Handcrafted feature extraction used in traditional classification approaches is time-consuming and laborious. In general, shallow but not deep abstract features of HSI are captured by these traditional feature extraction methods. Moreover, with the expert knowledge needed to set parameters, typical traditional classifiers such as support vector machine (SVM) [5] and random forest (RF) [6] are incapable of adapting to complex and varied contexts, showing poor generalization capacities. Therefore, the classification performance of traditional methods can be further improved by exploiting more discriminative features from HSI and enhancing the generalization capacities of classifiers.

Deep learning methods [7-9] have been gradually introduced into the field of hyperspectral image classification, achieving better performance than traditional methods [10]. For instance, the multiple layers used in deep learning methods extract more abstract and representative features automatically in a hierarchical manner, yielding more promising classification results of HSI. Three main deep learning strategies for HSI classification consist of spectral, spatial, and spectral-spatial feature classification. Based on the first principles, deep learning methods with HSI spectral information mainly focus on the spectrum of each pixel [11-13], which reflects the characteristics of a kind of ground object. The spectrum of each pixel is usually fed into a fully connected neural network or recurrent neural network (RNN) [14]. For instance, Chen, et al. [13] proposed a spectral-spatial classification method for hyperspectral data based on a deep belief network (DBN), reducing data dimension and capturing representative features from HSI. However, these one-dimensional (1D) networks mainly exploit the 1D spectral features, instead of utilizing the spatial structures of hyperspectral data for classification effectively [15].

Based on spatial information of HSI, the influence of neighboring pixels on the central pixel is fully considered by the 2D convolutional neural network (2D CNN)[16, 17]. In general, after HSI dimensional reduction, the 2D CNN takes the spatial image patch as input, fully capturing the relevant contextual spatial features from HSI. Gong, et al. [18] leveraged off-the-shelf CNN models to extract hierarchical deep spatial features and designed a novel objective function that embeds a metric learning regularization into SVM training, achieving state-of-the-art classification performance. However, deep learning methods with spatial features fail to extract deep abstract spectral features from hyperspectral data. Corresponding to the three-dimensional (3D) structure of HSI, the three-dimensional convolutional neural network (3D CNN) is used to extract the spectral and spatial features simultaneously without pre- or post-processing [19-22]. Zhong, et al. [19] designed a spectral-spatial residual network (SSRN) to consecutively extract representative features from HSI, alleviating the declining-accuracy existed in other deep learning models. Wang, et al. [20] proposed an end-to-end fast dense spectral-spatial convolution network (FDSSC) and applied densely-connected structures to learn deep features of hyperspectral data, improving classification accuracy and preventing model overfitting. However, with numerous training epochs, these models converge slowly and have low efficiency. The manner of learning spatial features from hyperspectral data by these models may introduce noise.

To address these problems, motivated by SSRN and FDSSC, an end-to-end multi-scale, 3D convolution neural network (MSCNN) was proposed for spectral-spatial classification of hyperspectral images. The paper’s major contributions are:

1) A multi-scale 3D convolution neural network (MSCNN) was proposed to extract spectral-spatial features simultaneously for hyperspectral image classification. In comparison with SSRN and FDSSC, fewer residual connections are designed in MSCNN to mitigate the declining-accuracy.

2) Without dimensional reduction, multi-scale image patches are adopted in MSCNN to further utilize spatial features. The MSCNN extracts different high-level features by gradually shrinking the image patch to the center pixel.

3) After expanding training samples through data augmentation methods, the MSCNN utilizes the dynamic learning rate, batch normalization (BN), L2 regularization and parametric rectified linear unit (PReLU) to accelerate model convergence, achieving state-of-the-art classification performance within 20 training epochs.
2. Proposed Framework

2.1. HSI Spectral-Spatial Features Extraction Using 3D CNN

After small patch extraction, the common layers to build a 3D CNN for HSI classification include convolution, pooling, batch normalization (BN), and a vector flattening, followed by a SoftMax layer. Based on a 3D CNN, the proposed MSCNN framework uses two convolution kernels from three channels to generate two feature maps. The value \( V_{i,j}^{xyz} \) at position \((x, y, z)\) on the \(j\)th feature cube in the \(i\)th layer is as follows [23, 24]:

\[
V_{i,j}^{xyz} = g(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} W_{i,j,m}^{u,s,t} V_{i-1,j}^{(x+n)(y+s)(z+l)} + b_{i,j})
\]

where \( g() \) and \( b_{i,j} \) are the activation function and bias, respectively. And \( m \) indicates the feature map connected to the current feature map in the \((i-1)\)th layer. \( U_i \) and \( S_i \) are the length and width of the convolution kernel in the spatial dimension, \( T_i \) is the size of the convolution kernel in the spectral dimension. \( W_{i,j,m}^{u,s,t} \) is the value at position \((u, s, t)\) of the kernel connected to the \(m\)th feature cube in the preceding layer.

Moreover, the MSCNN framework uses residual connections to facilitate model training. Let \( X \) and \( F(X) \) denote the input of the first convolution layer, and original underlying function after two convolution layers, respectively. Then, the residual block optimizes the convolution layers as an identity map between two convolution layers. The residual function \( G(X) = F(X) - X \) is almost zero, the original underlying function \( F(X) = X \). In general, \( F(X) \) can also be formulated as [25]:

\[
F(X) = G(X) + X
\]

After two convolution operations, the residual function \( G(X) \) can be obtained as follows:

\[
G(X) = g(g(X * W_1 + b_1) * W_2 + b_2)
\]

The MSCNN framework leverages 3D residual network to capture hierarchical, abstract and high-level features of HSI in both spatial and spectral feature learning stages.

2.2. Multi-scale 3D CNN for HSI Classification

The proposed multi-scale 3D CNN (MSCNN) framework convolutes multi-scale patches to capture spectral-spatial features in various levels for hyperspectral data classification. We demonstrate the detail classification process of hyperspectral data by inputting the Indiana Pines dataset into the MSCNN framework.

As shown in Figure 1, six convolutional layers are built in the spectral feature learning stage. A 3D HSI cube with a size of \(9\times9\times200\) is used as an input to the MSCNN framework. With 24 filters of \(1\times1\times7\) size and subsampling stride of \((1, 1, 1)\), the first convolution operation generates 24 feature maps of \(9\times9\times194\) size. After the first convolutional operation on the input, the data dimension is reduced and shallow spectral features are extracted from hyperspectral data. Then four convolution layers are designed to output 24 feature maps of \(9\times9\times194\) size, keeping the same size during the convolutional operations. The skip connection between the first and third convolutional layers generates 48 feature maps of \(9\times9\times194\) size. The outputs of the first and third convolutional layers are merged to the fifth convolutional layers to generate 72 feature maps of \(9\times9\times194\) size. The sixth convolutional layer, which has 24 filters of \(194\times1\times1\) size, outputs 24 feature maps of \(1\times9\times9\) size. Then a reshape layer used to transform the output of the sixth convolutional layer generates 1 feature map of \(9\times9\times24\) size, preparing for inputting into the spatial feature learning stage.

In the spatial feature learning stage, ten convolutional layers are built. The inputted image patches are cropped to two image patches of \(7\times7\times200\) and \(5\times5\times200\) size. The image patch of \(7\times7\times200\) size is merged into the first convolutional layer to output 224 feature maps of \(7\times7\times1\) size. Then a dropout layer is adopted to reduce the number of trainable parameters and avoid overfitting. The skip connection between the first and third convolutional layers generates 48 feature maps of \(7\times7\times1\) size.
The outputs of the first and third convolutional layers are merged into the fifth convolutional layers to generate 72 feature maps of $7 \times 7 \times 1$ size. Then a reshape layer is used to transform the output of the fifth convolutional layer, generating 1 feature map of $7 \times 7 \times 72$ size. With 24 filters of $3 \times 3 \times 72$ size and subsampling stride of $(1, 1, 1)$, the sixth convolutional layer gives output of 24 feature maps. The image patch of $5 \times 5 \times 200$ size is merged into the sixth convolutional layer to output 224 feature maps of $5 \times 5 \times 1$ size, which are transformed by the reshape layer to 1 feature map of $5 \times 5 \times 224$ size. Then a dropout layer is adopted to reduce the number of trainable parameters and avoid overfitting. The outputs of the sixth convolutional layer are merged into the eighth convolutional layer to generate 48 feature maps of $5 \times 5 \times 1$ size, which are transformed by the reshape layer to 1 feature map of $5 \times 5 \times 48$ size. With 24 filters of $3 \times 3 \times 48$ size and subsampling stride of $(1, 1, 1)$, the ninth convolutional layer outputs 24 feature map of $3 \times 3 \times 1$ size. After the tenth convolutional layer, which also outputs 24 feature maps of $3 \times 3 \times 1$ size, an average pooling layer transforms 24 feature maps of $3 \times 3 \times 1$ size into a feature vector of $1 \times 1 \times 24$ size. Then a dropout layer is adopted to reduce the number of trainable parameters and avoid overfitting. According to the number of land cover categories present, a fully connected layer is designed to adapt the MSCNN framework to different hyperspectral data.

![Figure 1. Flowchart of the MSCNN framework for HSI classification.](image_url)

3. Experimental Datasets
We used three commonly used hyperspectral datasets (Indiana Pines, University of Pavia and Pavia Center) to carry out the experiments and evaluate the classification performances of different models.

The Indiana Pines (IP) dataset was captured by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) sensor [26] over an agricultural area in Northwestern Indiana in 1992. It contains $145 \times 145$ pixels and 200 spectral channels that range from 400 to 2500 nm at intervals of 10 nm. IP has a spatial resolution of 20 m per pixel and includes 16 vegetation classes, which are shown in Figure 2(a). The numbers of training samples and total labeled samples of IP used in the experiments appear in Table 1.

The University of Pavia (UP) dataset was captured by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [27] over a university area in Northern Italy in 2001. It contains $610 \times 340$ pixels and 103 spectral channels, which range from 430 to 860 nm at intervals of 4 nm. With a spatial resolution of 1.3 m per pixel, it includes 9 classes, which are shown in Figure 2(b). Table 2 shows the numbers of training samples and total labeled samples of UP.
The Pavia Center (PC) dataset was captured by the ROSIS sensor. It contains 1096×715 pixels and 102 spectral channels, which range from 430 to 860 nm at intervals of 4 nm. With a spatial resolution of 1.3 m per pixel, it includes 9 classes that are shown in Figure 2(c). The numbers of training samples and total labeled samples of PC used in the experiments are listed in Table 2.

### Table 1. The number of training samples used in the Indiana Pines dataset.

| NO. | Class                      | Indiana Pines (IP) |     |     |
|-----|----------------------------|--------------------|-----|-----|
|     |                            | Train | Total |     |     |
| 1   | Alfalfa                    | 33    | 46    |     |     |
| 2   | Corn-notill                | 200   | 1428  |     |     |
| 3   | Corn-mintill               | 200   | 830   |     |     |
| 4   | Corn                       | 181   | 237   |     |     |
| 5   | Grass-pasture              | 200   | 483   |     |     |
| 6   | Grass-trees                | 200   | 730   |     |     |
| 7   | Grass-pasture-mowed        | 20    | 28    |     |     |
| 8   | Hay-windrowed              | 200   | 478   |     |     |
| 9   | Oats                       | 14    | 20    |     |     |
| 10  | Soybean-notill             | 200   | 972   |     |     |
| 11  | Soybean-mintill            | 200   | 2455  |     |     |
| 12  | Soybean-clean              | 200   | 593   |     |     |
| 13  | Wheat                      | 143   | 205   |     |     |
| 14  | Woods                      | 200   | 1265  |     |     |
| 15  | Bldg-Grass-Tree-Drives     | 200   | 386   |     |     |
| 16  | Stone-Steel-Towers         | 75    | 93    |     |     |

### Table 2. The number of training samples used in the University of Pavia and Pavia Center datasets.

| NO. | Class                | University of Pavia (UP) |     |     |     |
|-----|----------------------|--------------------------|-----|-----|-----|
|     |                       | Train. | Total |     |     |     |
| 1   | Asphalt              | 200   | 6631  | Water | 500  | 65,971 |
| 2   | Meadows              | 200   | 18,649 | Tree | 435  | 7598  |
| 3   | Gravel               | 200   | 2099  | Meadow | 400  | 3090  |
| 4   | Trees                | 200   | 3064  | Brick | 400  | 2685  |
| 5   | Painted-mental-sheets| 200   | 1345  | Bare soil | 400  | 6584  |
| 6   | Bare-soil            | 200   | 5029  | Asphalt | 400  | 9248  |
| 7   | Bitumen              | 200   | 1330  | Bitumen | 400  | 7287  |
| 8   | Self-blocking-bricks  | 200   | 3682  | Tile | 590  | 42,826 |
| 9   | Shadows               | 200   | 947   | Shadow | 400  | 2863  |
4. Experimental Results and Discussion

We compared the effectiveness of our proposed MSCNN framework alongside three other classification algorithms: 3D CNN [23], SSRN [19] and FDSSC [20]. In our experiments, with the epoch number of 20 and batch size of 32, the input image patches of 9×9 size [20] are inputted to the four models. With a decay rate of 0.00001 for the Adam optimizer [28], the learning rate of 3D CNN, FDSSC, and SSRN methods was 0.0001. For the proposed MSCNN method, with the initial learning rate of 0.0005, a dynamic learning rate with learning rate decreasing exponentially for every four epochs was adopted. Moreover, we used data augmentation to avoid over-fitting, generating 6400 samples for IP, 7200 samples for UP and PC. Then, all the expanded samples were divided randomly into training, validation and testing samples at ratios of 70%, 20%, and 10%, respectively. The activation function PReLU [29] and batch normalization [30] were used to accelerate model convergence. The classification performances of different models were evaluated by the overall accuracy (OA), average accuracy (AA), kappa statistic (K), and normalized confusion matrix. To obtain a more reliable estimate, we calculated the average accuracy of the five experimental results as a comprehensive performance measurement.

4.1. Experimental Results

According to the classification maps shown in Figure 3, the proposed MSCNN method generated smoother classified images that have less “speckled” noise patterns than those produced by the three other methods. The results of 3D CNN, SSRN, and FDSSC all contain different degrees of noise. Additionally, compared with other methods, the MSCNN model achieved better classification performances with higher classification accuracy.

Taking the IP dataset as an example, as shown in Figure 3, the boundaries between different ground objects are ambiguous. The soybean-mintill and grass-trees classes in classification maps obtained by other methods contain more speckled noise. However, the MSCNN learned more discriminative spectral-spatial features of multi-scale image patches, and generated sharp boundaries among different ground objects and homogeneous classification maps. Table 3 presents the quantitative classification results of different methods. Among all the methods, the proposed MSCNN method achieved the highest classification accuracy on the IP dataset, with OA, AA, and Kappa coefficient of 97.16%, 98.69% and 0.9665, respectively. MSCNN represents an improvement over all the other techniques, which have lower OA, AA, and Kappa coefficient values. Moreover, according to Figure 4, the MSCNN model mainly classified Soybean-mintill, Soybean-clean, and Com-mintill to other classes by mistake. Only the Soybean-mintill class was misclassified with a ratio of 0.05 because of the spectral similarities. In contrast to the MSCNN method, the three other methods have more serious misclassifications. Specifically, 3D CNN, FDSSC, and SSRN produced 10, 3, and 3 kinds of ground objects, respectively, whose misclassification ratios are greater than 0.05.

![Figure 3. Classification maps for IP dataset. (a) False-color image. (b) Ground-truth map. (c) 3D CNN. (d) FDSSC. (e) SSRN. (f) MSCNN.](image-url)
Figure 4. Normalized confusion matrix of classification maps by different methods on IP dataset.
(a) 3D CNN. (b) FDSSC. (c) SSRN. (d) MSCNN

Table 3. Classification results of different methods for the IP dataset.

| NO. | Class             | 3D CNN | FDSSC | SSRN | MSCNN |
|-----|-------------------|--------|-------|------|-------|
|     | OA (%)            | 79.22  | 92.77 | 95.21| 97.16 |
|     | AA (%)            | 88.65  | 96.76 | 97.40| 98.69 |
| 1   | Alfalfa           | 100.00 | 100.00| 100.00| 100.00|
| 2   | Corn-no till      | 77.93  | 94.79 | 93.73 | 97.07 |
| 3   | Corn-min till     | 71.11  | 88.25 | 92.38 | 96.19 |
| 4   | Corn              | 92.86  | 98.21 | 100.00| 100.00|
| 5   | Grass-pasture     | 92.23  | 97.88 | 97.88 | 99.65 |
| 6   | Grass-trees       | 97.74  | 99.81 | 95.28 | 99.43 |
| 7   | Grass-pasture-mowed| 87.50  | 100.00| 100.00| 100.00|
| 8   | Hay-windrowed     | 97.48  | 99.28 | 98.20 | 100.00|
| 9   | Oats              | 100.00 | 100.00| 100.00| 100.00|
| 10  | Soybean-no till   | 66.84  | 99.35 | 95.98 | 97.41 |
| 11  | Soybean-min till  | 68.82  | 84.48 | 94.06 | 95.25 |
| 12  | Soybean-clean     | 85.75  | 91.09 | 96.18 | 95.67 |
| 13  | Wheat             | 100.00 | 100.00| 100.00| 100.00|
| 14  | Woods             | 92.49  | 98.22 | 97.46 | 98.87 |
| 15  | Bldg-Grass-Tree-Drives | 87.63  | 96.77 | 97.31 | 99.46 |
| 16  | Stone-Steel-Towers| 100.00 | 100.00| 100.00| 100.00|

As shown in Figure 5, a pattern similar to that for the IP dataset was obtained for the UP dataset. The classification map obtained by 3D CNN contains a more “speckled” noise pattern in the Bare Soil class. Due to the strong feature extraction abilities, the proposed MSCNN outputted classification maps with less speckled noise. The results of the quantitative classification using different methods on the UP dataset (shown in Table 4) indicate that, the proposed MSCNN method also achieved the highest classification accuracy on the UP dataset, with OA, AA, and Kappa coefficient of 99.17%, 98.97% and 0.9888, respectively. Since the classification accuracies of all methods are relatively high, compared with SSRN methods, the OA, AA, and Kappa coefficient of MSCNN presents slight improvements of 0.05%, 0.15%, and 0.06%, with that of 0.81%, 0.89% and 1.08% for FDSSC, and 7.46%, 8.42% and 10.07% for 3D CNN, respectively. Moreover, as shown in Figure 6, the MSCNN misclassified 3 classes and mainly erroneously classified Self-Blocking Bricks to Gravel, with a ratio of 0.03. In contrast with the MSCNN method, 3D CNN, FDSSC, and SSRN misclassified 7, 7, and 5 kinds of ground objects, respectively.
Figure 5. Classification maps for UP dataset. (a) False-color image. (b) Ground-truth map. (c) 3D CNN. (d) FDSSC. (e) SSRN. (f) MSCNN.

Figure 6. Normalized confusion matrix of classification maps by different methods on UP dataset. (a) 3D CNN. (b) FDSSC. (c) SSRN. (d) MSCNN.

Table 4. Classification results of different methods for the UP dataset.

| NO. | Class               | 3D CNN | FDSSC | SSRN  | MSCNN |
|-----|---------------------|--------|-------|-------|-------|
|     | OA (%)              | 91.71  | 98.36 | 99.12 | 99.17 |
|     | AA (%)              | 90.55  | 98.08 | 98.82 | 98.97 |
|     | κ × 100             | 88.81  | 97.80 | 98.82 | 98.88 |
| 1   | Alfalfa             | 91.90  | 98.24 | 98.99 | 99.50 |
| 2   | Meadows             | 96.06  | 99.32 | 99.90 | 99.71 |
| 3   | Gravel              | 80.67  | 96.68 | 97.79 | 98.10 |
| 4   | Trees               | 93.65  | 98.36 | 99.37 | 98.53 |
| 5   | Painted-mental-sheets | 100.00 | 100.00 | 100.00 | 100.00 |
| 6   | Bare-soil           | 79.66  | 99.05 | 99.61 | 99.77 |
| 7   | Bitumen             | 86.28  | 98.41 | 99.47 | 99.91 |
| 8   | Self-blocking-bricks | 86.73  | 92.62 | 94.54 | 95.23 |
| 9   | Shadows             | 100.00 | 100.00 | 99.73 | 100.00 |

For the PC dataset (shown in Figure 7), the MSCNN method also generated a smoother classification map than the other methods. Table 5 presents the quantitative classification results of different methods on the PC dataset. All the methods outputted relatively high classification accuracy. However, in contrast to other methods, the proposed MSCNN method achieved the highest classification accuracy on the PC dataset, with OA, AA, and Kappa coefficient values of 99.87%, 99.70% and 0.9981, respectively. Compared with SSRN methods, the OA, AA, and Kappa coefficient of MSCNN present slight improvements of 0.11%, 0.25% and 0.15%, with those of 0.04%, 0.1% and 0.06% for FDSSC, and 1.54%, 3.39% and 2.2% for 3D CNN, respectively. Moreover, according to Figure 8, three types of ground objects were misclassified at ratios of 1% among all classified pixels in the class using both the MSCNN and SSRN methods. Meanwhile, almost 6 and 4 kinds of ground
objects were misclassified by 3D CNN and FDSSC methods. In particular, 3D CNN had a maximum percentage of pixels misclassified of 7% in Bare Soil class.

![Classification maps for PC dataset. (a) False-color image. (b) Ground-truth map. (c) 3D CNN. (d) FDSSC. (e) SSRN. (f) MSCNN.](image)

![Normalized confusion matrix of classification maps by different methods on PC dataset. (a) 3D CNN. (b) FDSSC. (c) SSRN. (d) MSCNN.](image)

![Table 5. Classification results of different methods for the PC dataset.](image)

To monitor the training process, with different methods, the accuracy evolution in terms of epochs on training and validation data is shown in Figure 9. The results demonstrate the average level of five experiments. Compared with the existing methods, the proposed MSCNN method performed best on the validation data of all three datasets. Although the SSRN method converged faster than MSCNN on training data, the proposed MSCNN method generated higher classification accuracy than SSRN on validation data in the last few epochs. Moreover, in the last epochs, the proposed MSCNN method presented the smallest standard deviation on all datasets among the four methods.
Figure 9. Accuracy evolution in terms of epochs on training and validation data. The shadow notes the standard deviation of accuracy for five executions. (a) IP training data (b) UP training data (c) PC training data (d) IP validation data (e) UP validation data (f) PC validation data.

4.2. Discussion

When the available samples are limited, deep learning models may mainly learn the specific features of a few samples and be overfitted, resulting in excellent classification performance on small samples, but poor classification performance on other samples. If the model is overfitted, along with the increase of training epochs, the training error is small but the testing error is large. Through the accuracy or loss evolution in terms of epochs on training and validation data, the classification performances on three widely used hyperspectral datasets are further analyzed comparatively.

As shown in Figure 10, the training and validation accuracy is nearly identical after 20 training epochs, and the classification accuracy of the 3D CNN is the lowest among four methods. These results indicate that 3D CNN has insufficient ability to extract high-level spatial and spectral features from hyperspectral images. At the seventh training epoch, the training and validation accuracy of SSRN are almost identical, but both of them are relatively low. Further model training causes training accuracy to be larger than validation accuracy, but training error is smaller than validation error, which may lead to model overfitting. For the FDSSC method, the training accuracy is almost equal to validation accuracy at nineteenth training epoch, but with a larger standard deviation than MSCNN. In terms of the proposed MSCNN method, the training accuracy curve almost overlaps the validation accuracy curve at the last few training epochs. Although MSCNN converges slowly with a large standard deviation in the early training epochs, it converges rapidly with a small standard deviation of the training and validation accuracy, achieving the highest classification accuracy among all methods after 20 training epochs without overfitting. Therefore, in contrast to other methods, MSCNN can be considered to be an effective approach that achieves reliable classification performance.
Figure 10. Accuracy or loss evolution in terms of epochs on training and validation data. The shadow notes the standard deviation of accuracy for five executions. The results of 3D CNN, FDSSC, SSRN and MSCNN are shown in (a)-(d) for IP, (e)-(h) for UP, and (i)-(l) for PC.

5. Conclusion and Future Work
In this paper, we have proposed an end-to-end multi-scale 3D convolution neural network, called MSCNN, for hyperspectral image classification. With multi-scale hyperspectral image patches of different sizes as input, the proposed MSCNN framework leverages 3D CNN to efficiently extract features in both the spectral and spatial feature learning stages. Compared with FDSSC, the MSCNN adopts fewer residual connections to alleviate the declining-accuracy. Besides, with more labeled samples after data augmentation, the MSCNN utilizes dynamic learning rate and activation function PReLU, and L2 regularization to accelerate model convergence. The dropout layers are also used to reduce model parameters. Based on three well-known hyperspectral datasets, the experimental results show that the MSCNN method can achieve better classification performances than other state-of-the-art methods within 20 training epochs.

In future work, we will verify the robustness and generalization of the MSCNN method in more hyperspectral scenarios. Besides, the transfer learning methods also need to be developed for HSI classification to improve the efficiency of model training.

References
[1] W. Sun and Z. Xia, 2017, *International Journal of Applied Earth Observation & Geoinformation*, "Estimating soil zinc concentrations using reflectance spectroscopy," 58, pp 126-133.
[2] S. Zhang, J. Li, Z. Wu, and A. Plaza, 2018, *IEEE Transactions on Geoscience & Remote Sensing*, "Spatial Discontinuity-Weighted Sparse Unmixing of Hyperspectral Images," 56, pp 5767-5779.
[3] G. Le, J. Xia, P. Du, and J. Chanussot, 2017, *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, "Class-oriented Weighted Kernel Sparse Representation with Region-level Kernel for Hyperspectral Imagery Classification," PP, pp 1-13.

[4] J. Li, J. M. Bioucas-Dias, A. Plaza, and L. Lin, 2016, *IEEE Transactions on Geoscience & Remote Sensing*, "Robust Collaborative Nonnegative Matrix Factorization for Hyperspectral Unmixing," 54, pp 6076-6090.

[5] L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, 2008, *IEEE Geoscience & Remote Sensing Letters*, "Semisupervised Image Classification With Laplacian Support Vector Machines," 5, pp 336-340.

[6] B. Waske, J. A. Benediktsson, K. Árnason, and J. R. Sveinsson, 2009, *Canadian Journal of Remote Sensing*, "Mapping of hyperspectral AVIRIS data using machine-learning algorithms," 35, pp S106-S116.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 2012, in *International Conference on Neural Information Processing Systems*, "ImageNet classification with deep convolutional neural networks."

[8] Y. LeCun, Y. Bengio, and G. Hinton, 2015, *Nature*, "Deep learning," 521, p 436.

[9] B. Yoshua, C. Aaron, and V. Pascal, 2013, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, "Representation learning: a review and new perspectives," 35, pp 1798-1828.

[10] Y. Chen, Z. Lin, Z. Xing, W. Gang, and Y. Gu, 2017, *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, "Deep Learning-Based Classification of Hyperspectral Data," 7, pp 2094-2107.

[11] W. Hu, Y. Huang, W. Li, Z. Fan, and H. Li, 2015, *Journal of Sensors*, "Deep convolutional neural networks for hyperspectral image classification," 2015, pp 1-12.

[12] P. Zhong, Z. Gong, S. Li, and C. B. Schonlieb, 2017, *IEEE Transactions on Geoscience & Remote Sensing*, "Learning to Diversify Deep Belief Networks for Hyperspectral Image Classification," 55, pp 3516-3530.

[13] Y. S. Chen, X. Zhao, and X. P. Jia, Jun 2015, *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, "Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network," 8, pp 2381-2392.

[14] L. Mou, P. Ghamisi, and X. Z. Xiao, 2017, *IEEE Transactions on Geoscience & Remote Sensing*, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," 55, pp 3639-3655.

[15] Y. Zhan, D. Hu, Y. Wang, and X. Yu, 2017, *IEEE Geoscience & Remote Sensing Letters*, "Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks," 15, pp 212-216.

[16] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, 2016, *IEEE Transactions on Geoscience & Remote Sensing*, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," 54, pp 6232-6251.

[17] J. Yue, W. Zhao, S. Mao, and H. Liu, 2015, *Remote Sensing Letters*, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," 6, pp 468-477.

[18] C. Gong, Z. Li, J. Han, X. Yao, and G. Lei, 2018, *IEEE Transactions on Geoscience & Remote Sensing*, "Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification," 56, pp 6712-6722.

[19] Z. Zhong, J. Li, Z. Luo, and M. Chapman, 2018, *IEEE Transactions on Geoscience & Remote Sensing*, "Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework," 56, pp 847-858.

[20] W. Wang, S. Dou, Z. Jiang, and L. Sun, 2018, *Remote Sensing*, "A Fast Dense Spectral–Spatial Convolution Network Framework for Hyperspectral Images Classification," 10, pp 1068-1087.

[21] W. Song, S. Li, L. Fang, and T. Lu, 2018, *IEEE Transactions on Geoscience & Remote Sensing*, "Hyperspectral Image Classification With Deep Feature Fusion Network," PP, pp 1-12.
[22] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, 2018, *IEEE Transactions on Geoscience & Remote Sensing*, "Hyperspectral Image Classification With Deep Learning Models," 56, pp 5408-5423.

[23] B. Liu, X. Yu, P. Zhang, X. Tan, R. Wang, and L. Zhi, 2018, *Journal of Applied Remote Sensing*, "Spectral–spatial classification of hyperspectral image using three-dimensional convolution network," 12, p 18.

[24] Y. Li, H. Zhang, and Q. Shen, 2017, *Remote Sensing*, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," 9, pp 67-88.

[25] K. He, X. Zhang, S. Ren, and S. Jian, 2016, in *IEEE Conference on Computer Vision & Pattern Recognition*, "Deep Residual Learning for Image Recognition," vol. 90, pp. 770-778.

[26] R. O. Green et al., 1998, *Remote Sensing of Environment*, "Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)," 65, pp 227-248.

[27] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, and H. V. D. Piepen, 1988, in *Optoelectronic Technologies for Remote Sensing from Space*, "ROSIS (Reflective Optics System Imaging Spectrometer) - A Candidate Instrument For Polar Platform Missions," vol. 0868.

[28] G. Yang, Y. Jing, S. Li, and J. Hu, 2018, *Journal of Huazhong University of Science & Technology*, "Modified CNN algorithm based on Dropout and ADAM optimizer," 46, pp 122-127.

[29] S. Mei, J. Ji, J. Hou, L. Xu, and D. Qian, 2017, *IEEE Transactions on Geoscience & Remote Sensing*, "Learning Sensor-Specific Spatial-Spectral Features of Hyperspectral Images via Convolutional Neural Networks," 55, pp 4520-4533.

[30] M. Liu, W. Wei, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, 2018, *Neurocomputing*, "Deep Learning Based on Batch Normalization for P300 Signal Detection," 275, pp 288-297.

**Acknowledgments**

This work was funded by the National Key R&D Program on Monitoring, Early Warning and Prevention of Major National Disaster (No. 2017YFC1502802) and the Central Public Welfare Project (No. 2018SYIAEZD1).