remote sensing

MDPI

Article

A Spectral-Spatial Cascaded 3D Convolutional Neural Network with a Convolutional Long Short-Term Memory Network for Hyperspectral Image Classification

Wenchao Qi [1,2], Xia Zhang [1,*], Nan Wang [1], Mao Zhang [1,2] and Yi Cen [1]

[1] State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100101, China; qiwc@radi.ac.cn (W.Q.); wangnan@radi.ac.cn (N.W.); zhangmao@radi.ac.cn (M.Z.); cenyi@radi.ac.cn (Y.C.)

[2] University of Chinese Academy of Sciences, Beijing 100101, China

* Correspondence: zhangxia@radi.ac.cn; Tel.: +86-1369-115-9542

Received: 22 August 2019; Accepted: 8 October 2019; Published: 11 October 2019

Abstract: Deep learning methods used for hyperspectral image (HSI) classification often achieve greater accuracy than traditional algorithms but require large numbers of training epochs. To simplify model structures and reduce their training epochs, an end-to-end deep learning framework incorporating a spectral-spatial cascaded 3D convolutional neural network (CNN) with a convolutional long short-term memory (CLSTM) network, called SSCC, is proposed herein for HSI classification. The SSCC framework employs cascaded 3D CNN to learn the spectral-spatial features of HSIs and uses the CLSTM network to extract sequence features. Residual connections are used in SSCC to accelerate model convergence, with the outputs of previous convolutional layers concatenated as inputs for subsequent layers. Moreover, the data augmentation, parametric rectified linear unit, dynamic learning rate, batch normalization, and regularization (including dropout and L2) methods are used to increase classification accuracy and prevent overfitting. These attributes allow the SSCC framework to achieve good performance for HSI classification within 20 epochs. Three well-known datasets including Indiana Pines, University of Pavia, and Pavia Center were employed to evaluate the classification performance of the proposed algorithm. The GF-5 dataset of Anxin County, obtained from China's recently launched spaceborne Advanced Hyperspectral Imager, was also used for classification experiments. The experimental results demonstrate that the proposed SSCC framework achieves state-of-the-art performance with better training efficiency than other deep learning methods.

Keywords: 3D convolutional neural network; convolutional LSTM; cascaded structures; spectral-spatial feature learning; hyperspectral image classification

1. Introduction

Due to the recent development of hyperspectral remote sensing imaging technology, a large number of hyperspectral remote-sensing images with different spatial and spectral resolutions are available. Hyperspectral images (HSIs) contain abundant spatial geometric information as well as spectral information reflecting various characteristics of ground objects. The rich information contained in HSIs has been widely used in numerous applications including land cover classification, target detection, mineral exploration, and precision agriculture [1,2]. HSI classification, the process of labeling each pixel, is a challenging task required for all of these applications.

Due to the high dimension of HSIs, which easily causes the Hughes phenomenon [3,4] with a small training set, previous studies have usually mapped HSIs into a low-dimension space to maintain

most of the useful information contained in HSIs [5]. In addition, the high spatial resolution of HSIs may increase spectral variation for the same ground objects while decreasing spectral variation among different ground objects [6]. The classification results of HSIs based solely on spectral information may be unsatisfactory. Therefore, numerous researchers have attempted to integrate spectral and spatial information into HSI classification, with some success [7,8]. However, traditional HSI classification strategies, which generally consist of feature library construction and classifier training, have two drawbacks [9]. First, feature library construction is time-consuming and requires numerous manual interventions. Due to their weak generalization capacity, these methods cannot be quickly and effectively applied to different applications. Second, the linear and nonlinear transformation methods used for feature extraction have limited capacity to represent features [10]. Without more sophisticated methods, feature selection can only identify shallow features of HSIs, rather than fully exploiting the deep spatial and spectral features of HSIs. Although traditional machine learning methods, which mainly use custom-built features, have been wildly adopted for HSI pixel classification, these methods inadequately extract spectral-spatial features and cannot adapt to different contexts. The generalization capacity of traditional models must be improved to obtain acceptable classification performance [11].

As one of the most vibrant developing topics in computer science, deep learning algorithms that can efficiently learn discriminative and representative features have attracted great attention in the field of image processing, leading to major breakthroughs [12–14]. Deep learning approaches have recently been introduced into HSI classification, and show better performance than traditional machine learning algorithms [15]. Without the complicated process of custom feature extraction and optimization, classification methods based on deep neural networks can extract deep and highly abstract semantic features from HSIs automatically and hierarchically. Using a series of stacked feature extractors and a softmax classifier, deep learning methods carry out the end-to-end classification of HSIs adaptively for different applications. With their powerful feature extraction capacity, deep learning methods can obtain features at different levels from HSIs by constructing various neural network structures with different depths and widths. In particular, shallow features of HSIs such as edges and textures are extracted by the shallow structures of deep neural network models, while complex and abstract features of HSIs are obtained through the deep neural network structures. In addition, one-dimensional convolution neural networks are usually used to extract spectral features of HSIs, while the two-dimensional convolution neural network is used to extract spatial geometric features.

From the perspective of feature extraction, deep learning methods used for HSI classification can be categorized into three strategies: neural networks for spectral, spatial, and spectral-spatial feature classification. Deep learning methods for HSI spectral classification generally assume that each pixel is a type of ground object. Such methods directly use the spectrum of each pixel in the HSI as input to a fully connected neural network [15], one-dimensional convolutional neural network (1D CNN) [16], 1D generative adversarial network (GAN) [17] or recurrent neural network (RNN) [18,19], which is simple in principle and easy to implement. For example, Zhong et al. [20] applied a diversified deep belief network (DBN) model to HSI classification with a limited number of training samples. The diversified DBN regularizes the pretraining and fine-tuning procedures of the DBN by promoting prior over latent factors, improving the performance of HSI classification. Hu et al. [16] developed a 1D CNN to extract spectral features from HSIs. Their method achieved higher classification accuracy than support vector machines. Zhan et al. [21] designed a 1D GAN that automatically extracts the spectral features of HSIs to classify HSIs with few training samples. The 1D GAN obtained promising results with only a small number of labeled pixels. Mou et al. [18] regarded the spectra of HSI pixels as sequential data, and thus used an RNN model with a parametric rectified tanh function for classification of HSIs. However, these methods generally leverage spectral features while ignoring the spatial features of HSIs, and their classification performance requires further improvement.

The literature demonstrates that spatial information about HSIs can enhance the performance of HSI classification [22]. In general, deep learning methods for HSI spatial classification account fully for the influence of adjacent pixels on the labeled pixel (referred to as the central pixel) of an

HSI [16,23]. In addition, dimension reduction approaches such as principal component analysis (PCA) have been used to reduce the dimensionality of raw input data. After dimensionality reduction, image patches around the central pixel are extracted as inputs for two-dimensional convolution neural network models (2D CNN). With the advantages of PCA and CNN, deep learning models based on spatial features reduce the dimensionality of input data and improve computational efficiency, while also extracting invariant features. For example, Yang et al. [24] applied a 2D CNN to exploit the multi-scale convolution features of HSIs and obtained better classification performance than that from traditional machine learning algorithms. Gong et al. [25] fine-tuned pretraining models such as AlexNet and GoogleLeNet to capture the deep spatial features of HSI datasets and achieved high classification accuracy. However, these methods focus on the extraction of HSI spatial information and fail to fully utilize the deep spectral-spatial features of HSIs. Corresponding to a 3D cube of the HSI, a three-dimensional convolutional neural network (3D CNN) is usually used to extract spectral–spatial features of HSIs effectively without any pre- or post-processing [9,11]. In addition, the long short-term memory (LSTM) and convolutional LSTM (CLSTM) methods can also be used to learn HSI spatial and spatial features [26]. In particular, the long short-term memory (LSTM) of CLSTM exploits the spectral features of HSIs by considering spectral bands as an image sequence. To improve extraction of spatial features from HSIs, convolutional operators are used in lieu of fully connected operators in CLSTM [27]. Zhou et al. [28] adopted spectral-spatial LSTM network to learn deep features and a decision fusion method was used to make HSI classification. The spectra of center pixels (1D vectors) were input into SeLSTM to learn spectral feature. The image patches of center pixel (2D vectors), cropped from the first principle component, were transformed into S-length sequences and feed to SaLSTM to capture the deep spatial features of HSI. Then the classification maps of SeLSTM and SaLSTM were fused in a weighted sum manner to obtain a joint spectral-spatial classification results. Li et al. [29] proposed a light 3D CNN model with few parameters for accurate HSI classification. With a lower likelihood of overfitting, the model captured the deep spectral-spatial features simultaneously and outperformed the 2D CNN and DBN models. Zhong et al. [9] designed a spectral–spatial residual network (SSRN) that used residual blocks to efficiently capture spectral and spatial features consecutively. The SSRN has a deep neural network structure, and back-propagation of gradients was facilitated through connection of residual blocks among convolutional layers. Compared to other methods such as support vector machine (SVM) and sparse auto-encoder (SAE), SSRN achieved the best classification performance. Wang et al. [11] proposed a fast dense spectral–spatial convolution (FDSSC) framework using densely connected structures to identify deep features of HSIs with a dynamic learning rate and parametric rectified linear units to avoid overfitting. Classification by the FDSSC reduced the training time and achieved state-of-the-art performance. However, these models have low efficiency and slow convergence speeds, require numerous training epochs, and learn spatial information from HSIs in a manner that may introduce noise [24].

To resolve these problems, we aimed to design an end-to-end spectral-spatial cascaded 3D CNN with a convolutional LSTM (SSCC) framework motivated by the SSRN, FDSSC, and CLSTM. The major contributions of this paper are summarized as follows:

(1) A cascaded CNN model with residual connections was designed for HSI classification. Compared to FDSSC and SSRN, the SSCC model avoids the decreasing-accuracy phenomenon using fewer residual connections. In contrast to SSLSTMs, rather than separately learning features in spectral and spatial dimensions, the SSCC model used 3D CNN and CLSTM to simultaneously capture spectral-spatial features and classified HSI directly, instead of fusing different classification maps in a weighted sum manner.

(2) With two parallel branches for the processes of spatial and spectral feature extraction, more discriminative features are obtained from HSIs separately without dimension-reduction approaches. Moreover, the salt-and-pepper noise present in classification maps can be attenuated.

(3) Combining the advantages of LSTM and CNN, the CLSTM obtains spectral-spatial features of HSIs simultaneously. The spectral features are determined using spectral bands as an image

sequence in LSTM. Meanwhile, convolutional operators are combined into the LSTM to identify deep spatial features of HSIs.

(4) Data augmentation techniques, the dynamic learning rate, and the activation function parametric rectified linear unit (PReLU) are used in the SSCC model to expand training samples and accelerate model convergence. The number of training epochs was reduced to 20 while achieving state-of-the-art performance.

The remainder of this paper is organized as follows. Section 2 describes the proposed SSCC framework in detail. Section 3 introduces the datasets used in the experiments and experimental settings used for HSI classification. Experimental results are described and discussed in Section 4. Section 5 concludes the paper and provides suggestions for future work.

2. Proposed Framework

In this section, we present the proposed SSCC framework for HSI classification, which consists primarily of an input layer, a spectral feature learning process, two CLSTM layers, a spatial feature learning process, and a fully connected layer. In contrast to FDSSC, the SSCC has only four residual connections. In this section, cascaded 3D CNN extraction of spectral-spatial features from HSIs and CLSTM extraction of discriminative features from sequential image data are described in detail.

2.1. Extracting of HSI Spectral and Spatial Features Using Cascaded 3D CNN

The general process of applying 3D CNN to HSIs represented by 3D cubes includes small patch extraction, convolution, pooling and batch normalization steps, followed by feature vector flattening and classification. The small patch generally has a size of $w \times w \times b$, where the target pixel is at the center of an image block of spatial size $w \times w$ and spectral dimension b. After feature learning steps involving convolution layers with different kernel sizes, the data are flattened into 1D vectors and then input to a classifier based on the softmax function to obtain classification maps. In the proposed method, the cascaded 3D CNN is used to capture representative features that include spectral-spatial information from HSIs via two branches operating in parallel. Moreover, to mitigate the gradient disappearance phenomenon, residual connections are used between convolutional layers during the process of learning spectral-spatial features.

As a basic element of the SSCC network, the 3D CNN processes input data from three channels using two convolution operations. For example, the value at position (x, y, z) on the jth feature cube in the ith layer is given by [29]:

$$V_{i,j}^{xyz} = g(b_{i,j} + \sum_{m} \sum_{r=0}^{R_i-1} \sum_{s=0}^{S_i-1} \sum_{t=0}^{T_i-1} W_{i,j,m}^{r,s,t} V_{i-1,m}^{(x+r)(y+s)(z+t)}) \quad (1)$$

where $g(\cdot)$ denotes the activation function, m is the feature cube connected to the current feature cube in the $(i-1)$th layer, $W_{i,j,m}^{r,s,t}$ is the (r, s, t)th value of the kernel connected to the mth feature cube in the preceding layer, and $b_{i,j}$ denotes the bias on the jth feature cube in the ith layer. In addition, the length and width of the convolution kernel in the spatial dimension are denoted by R_i and S_i respectively, and the kernel size of spectral dimension is T_i.

For the $(k+1)$th convolutional layer of the 3D CNN with f^k feature maps of size $w^k \times w^k \times d^k$ and f^{k+1} convolutional filters of size $c^{k+1} \times c^{k+1} \times p^{k+1}$, and subsampling strides of (s_1,s_2, and s_3), the output will contain f^{k+1} feature maps. These f^{k+1} feature maps each have a size of $w^{k+1} \times w^{k+1} \times d^{k+1}$, where the spatial width $w^{k+1} = [1 + (w^k - c^{k+1})/s_1]$ and the spectral depth $d^{k+1} = [1 + (d^k - p^{k+1})/s_3]$. The batch normalization (BN) operation is usually conducted after the convolutional layer to avoid an internal covariance shift [30].

Residual connections [31] in the SSCC framework can enable easy training of the deep network and provide the benefit of increased depth. As shown in Figure 1, for a residual block consisting of two

convolutional layers, X represents the input to the first layer, and $F(X)$ denotes the original underlying function obtained after two convolution operations. Using a short connection, the residual block tries to optimize the convolutional layers as an identity map, which can be described by the residual function $G(X) = F(X) - X$. When the residual function $G(X) = 0$, the original underlying function $F(X) = X$. $F(X)$ can also be written in the following form:

$$F(X) = G(X) + X \tag{2}$$

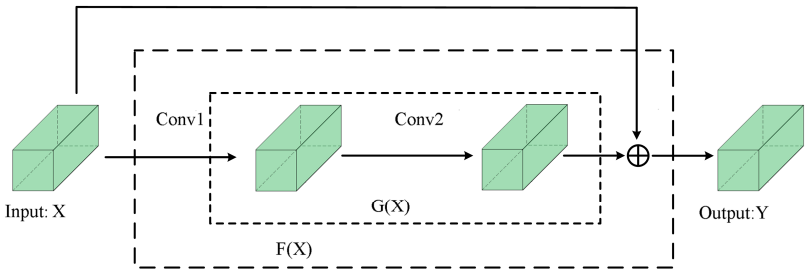


Figure 1. Illustration of a residual block.

The residual function $G(X)$ can be obtained through two convolutional operations on the input X:

$$G() = g(g(*W_1 + b_1) * W_2 + b_2) \tag{3}$$

where $g(\cdot)$ denotes the activation function, W_1 and W_2 are convolutional kernels, b_1 and b_2 indicate biases.

The cascaded 3D CNN is designed with spatial and spectral feature learning stages to sufficiently exploit the deep and abstract information contained in HSIs. By setting different numbers and sizes of convolution kernels, more subtle discriminative features are obtained [32].

2.2. Extracting of HSI Spectral and Spatial Features Using Convolutional LSTM

The structure of the convolutional LSTM (CLSTM) is illustrated in Figure 2. The CLSTM was proposed to address the problem of insufficient spatial information utilization in the traditional LSTM. [27]. Compared to the LSTM, which uses the full input-to-state and state-to-state connections, the CLSTM replaces the full connection layers with convolutional layers. As a modification of the LSTM, the inputs, cell outputs, hidden states and gates of CLSTM are 3D tensors, and the last two dimensions are spatial information arranged in rows and columns of images or features.

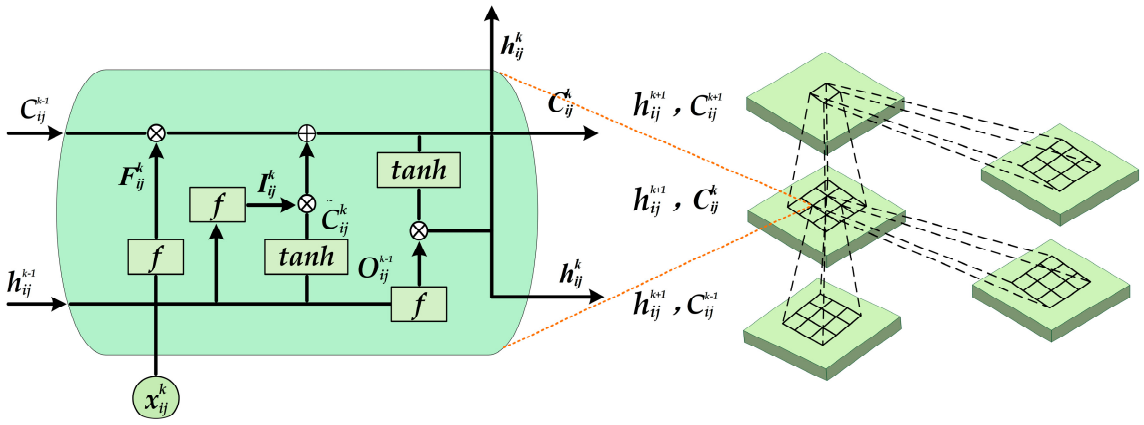


Figure 2. The structure of convolutional LSTM (CLSTM). Left, zoomed view of the inner computational unit called the memory cell. "⊕" and "⊗" represent the matrix addition and the dot product, respectively.

First, for the kth image patch x_{ij}^k in the sequence X_{ij}^k, CLSTM uses the forget gate F_{ij}^k to filter the unwanted information from the previous cell state C_{ij}^{k-1}. The forget gate uses a logistic sigmoid activation function on the image patch x_{ij}^k and hidden state h_{ij}^{k-1}, outputting a value between 0 and 1. A value of 0 denotes unwanted information while 1 represents information to be retained. Secondly, the input gate I_{ij}^k retains the useful information in the same manner as the forget gate. The candidate value $\widetilde{C}_{ij}^{k-1}$ is calculated from h_{ij}^{k-1} and x_{ij}^k in the memory cell. Then the CLSTM updates the cell state C_{ij}^k by adding the product of the cell states C_{ij}^{k-1} and F_{ij}^k to the product of $\widetilde{C}_{ij}^k$ and I_{ij}^k. The final output of information depends on the cell state C_{ij}^k and the output gate O_{ij}^k. The entire CLSTM process can be described with the following equations:

$$\begin{aligned} F_{ij}^k &= f(W_{hf} * h_{ij}^{k-1} + W_{xf} * x_{ij}^k + b_f) I_{ij}^k = f(W_{hi} * h_{ij}^{k-1} + W_{xi} * x_{ij}^k + b_i) \widetilde{C}_{ij}^k \\ &= \tanh(W_{hc} * h_{ij}^{k-1} + W_{xc} * x_{ij}^k + b_c) C_{ij}^k = F_{ij}^k \circ C_{ij}^{k-1} + I_{ij}^k \circ \widetilde{C}_{ij}^k O_{ij}^k \\ &= f(W_{ho} * h_{ij}^{k-1} + W_{xo} * x_{ij}^k + b_o) h_{ij}^k = O_{ij}^k \circ \tanh(C_{ij}^k) \end{aligned} \tag{4}$$

where the f denotes the logistic sigmoid function, and "$\circ$" and "$*$" are dot product and convolutional operator, respectively. The terms b_f, b_i, b_c, b_o denote biases. The subscripts of weight metrics have similar meanings, for example, W_{hf} is the hidden-forget gate matrix and W_{hi} is the hidden-input gate matrix. The padding operation is usually carried out prior to the convolution operation to ensure that the states have the same numbers of rows and columns as the inputs. Boundary points are handled by padding on the states and can be considered to adopt the state of the outside area for computation [27].

2.3. Spectral-Spatial Cascaded 3D CNN with Convolutional LSTM Networks

The proposed SSCC framework uses two feature extraction branches to obtain spatial and spectral information from HSIs for classification at the pixel level. Fewer skip connections between convolution layers can alleviate the decreasing-accuracy phenomenon. In addition, two CLSTM operations model long-term dependencies in the spectral dimension and capture the spectral-spatial features of HSIs. The Indiana Pines dataset is input into the SSCC framework to demonstrate the process of HSI classification.

As shown in Figure 3, input images for the SSCC network are in the form of 3D cubes $9 \times 9 \times 200$ in size. After the first convolution operation with 24 filters of $1 \times 1 \times 7$ size and the subsampling stride of (1, 1, 2), the output contains 24 feature maps of $9 \times 9 \times 97$ size. With a stride of 2 in the spectral dimension, the high dimensionality of input images is reduced and low-level spectral features of the HSIs are obtained.

The spectral feature learning block includes two branches. The first branch includes three convolutional layers and one spectral residual connection. After the first two convolutional layers containing 24 filters of $1 \times 1 \times 7$ size and a subsampling stride of (1, 1, 1), 24 feature maps $9 \times 9 \times 97$ in size are generated. Then the skip connection between the convolutional layer following the input layer and the second convolutional layer in the first branch of the spectral feature learning block generates 48 feature maps of $9 \times 9 \times 97$ size. With 128 filters of $1 \times 1 \times 97$ size and a subsampling stride of (1, 1, 1), the output from the third convolutional layer contains 128 feature maps of $9 \times 9 \times 1$ size. The second branch includes two convolutional layers and one spectral residual connection. After the first convolutional layer, containing 12 filters of $1 \times 1 \times 7$ size and a subsampling stride of (1, 1, 1), the skip connection is designed to generate 36 feature maps of $9 \times 9 \times 97$ size. Then the second convolutional layer with 200 filters of $1 \times 1 \times 97$ size and a subsampling stride of (1, 1, 1) generates 200 feature maps of $9 \times 9 \times 1$ size. Finally, the outputs of the two branches are merged to obtain 328 feature maps that are $9 \times 9 \times 1$ in size.

Next, to reduce the number of trainable parameters and avoid overfitting, a dropout layer is adopted. Then, two CLSTM layers with 24 filters of 3×3 size are used to extract spatial and spectral features simultaneously, resulting in 24 feature maps of $9 \times 9 \times 1$ size. A reshape layer is used to transform the dimension of feature maps for input into the network and spatial feature extraction.

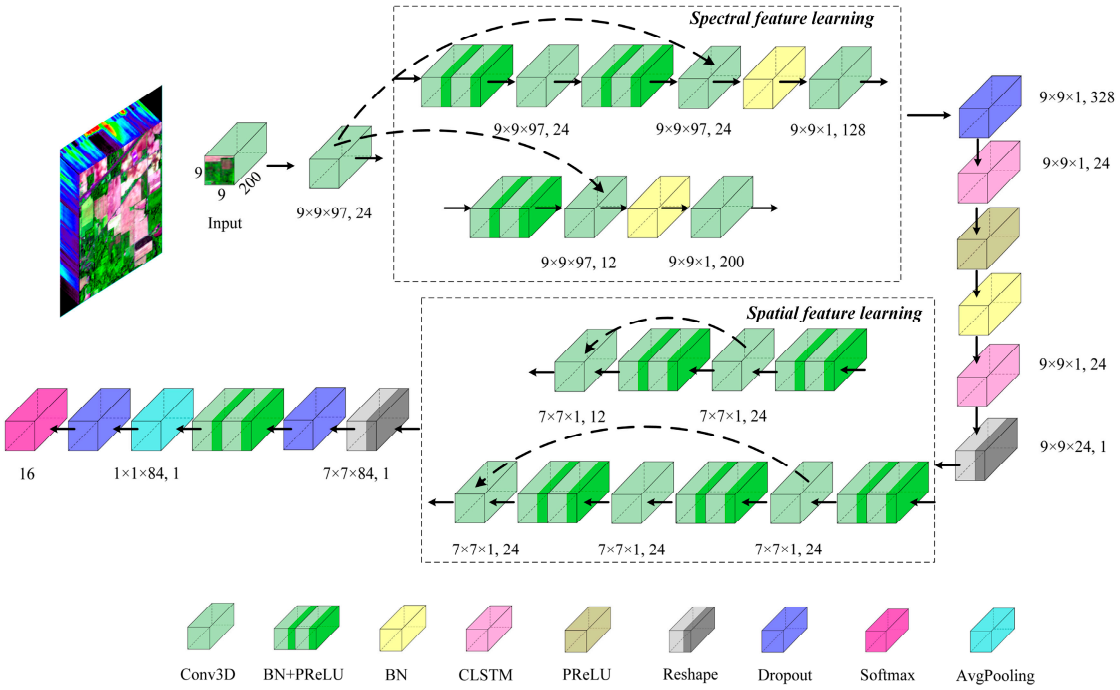


Figure 3. Flowchart of the SSCC network for HSI classification.

Corresponding to the spectral feature learning block, the spatial feature learning block also includes two branches. The first branch includes two convolutional layers and one spatial residual connection. The first convolutional layer, which has24 spatial kernels of $3 \times 3 \times 24$ size and a subsampling stride of (1, 1, 1), convolves the output of the reshape layer to generate 24 feature maps of $7 \times 7 \times 1$ size. With 12 filters of $3 \times 3 \times 1$ size and a subsampling stride of (1, 1, 1), the second convolutional layer outputs 12 feature maps of $7 \times 7 \times 1$ size. Then the outputs of the two convolutional layers are merged into 36 feature maps of $7 \times 7 \times 1$ size. The second branch includes three convolutional layers and one spatial residual connection. Each of the convolutional layers contains 24 spatial kernels with a size of $3 \times 3 \times 24$ for the first convolution operation, and of $3 \times 3 \times 128$ for the second and third convolution operations. Then, a skip connection is employed between the first and third convolutional layers to output 48 feature maps of $7 \times 7 \times 1$ size. Finally, the outputs of the two branches are merged to obtain 84 feature maps of $7 \times 7 \times 1$ size.

After learning the spatial features of HSIs, a reshape layer is used to transform 48 feature maps of $7 \times 7 \times 1$ size into one feature map of $7 \times 7 \times 84$ size. With a pooling size of (7, 7, 1), an average pooling layer transforms the feature map of $7 \times 7 \times 84$ size into a feature vector of $1 \times 1 \times 84$ size. Then, a fully connected layer adapts the SSCC network to different HSIs according to the number of land cover categories present.

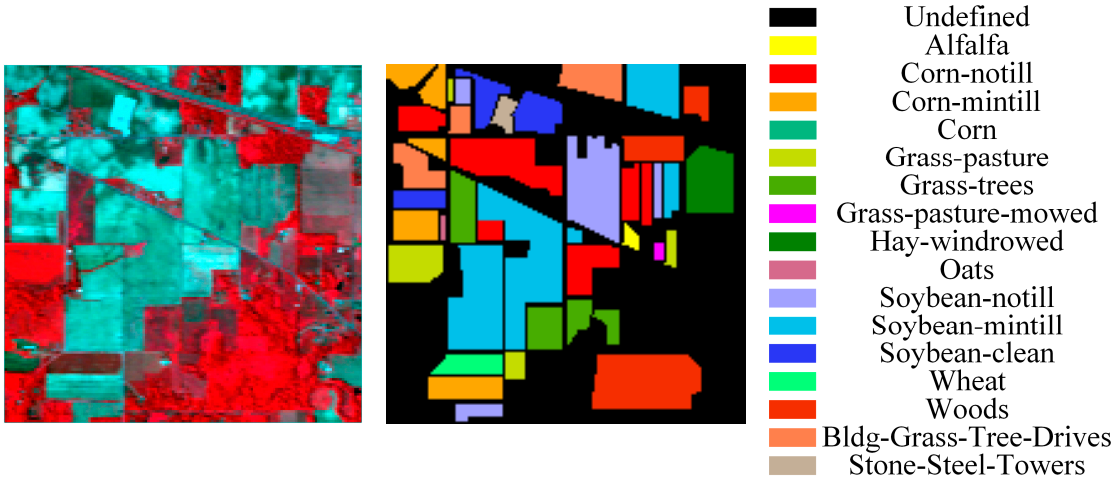
3. Experimental Settings and Datasets

3.1. Datasets

In our experiments, four HSI datasets were used to demonstrate the performance of the proposed SSCC model for HSI classification, including the Indiana Pines (IN), University of Pavia (UP), Pavia Center (PC), and GF-5 datasets.

Indiana Pines (IP): This dataset covers an agricultural area and was gathered using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor [33] from Northwest Indiana in 1992. It contains 145×145 pixels and 16 vegetation classes, with a spatial resolution of 20 m per pixel. After discarding 4 zero bands and 20 bands corrupted by water absorption effects, the remaining 200 spectral bands, which ranged from 400 to 2500 nm at intervals of 10 nm, were used for analyses. False-color composite

and ground truth maps of IP are shown in Figure 4. The numbers of training and test samples in the dataset are listed in Table 1.



(**a**) (**b**)

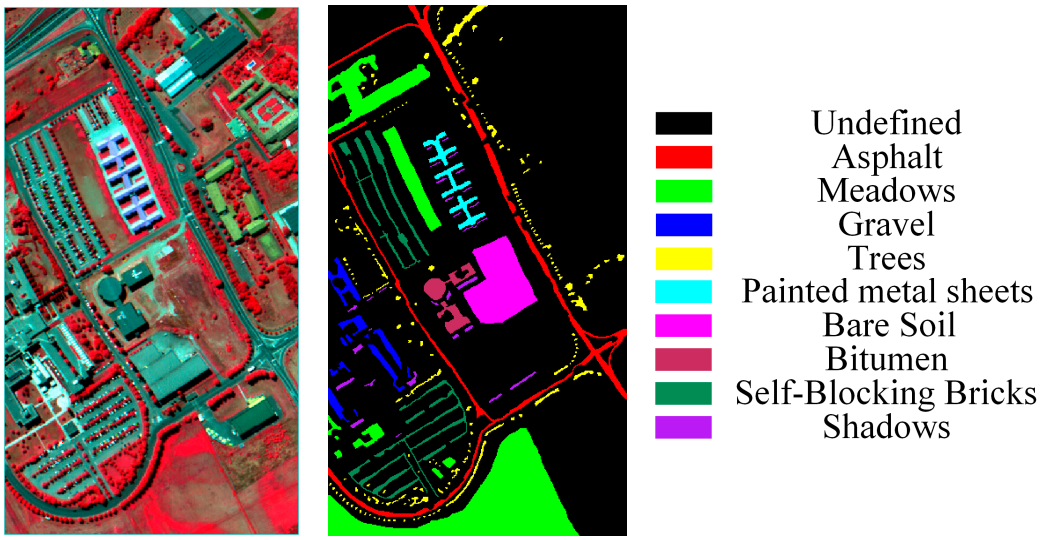
Figure 4. Indiana Pines dataset: (**a**) false-color composite of Indiana Pines, (**b**) ground-truth map.

Table 1. Number of training samples used in the Indiana Pines dataset.

NO.	Indiana Pines (IP)			GF-5 Dataset (GF5AX)		
	Class	**Train**	**Total**	**Class**	**Train**	**Total**
1	Alfalfa	33	46	Corn	1630	5434
2	Corn-notill	200	1428	Water-body	627	2089
3	Corn-mintill	200	830	Rice	713	2377
4	Corn	181	237	Building	1571	5237
5	Grass-pasture	200	483	Reed	707	2356
6	Grass-trees	200	730	Sorghum	53	177
7	Grass-pasture-mowed	20	28	Bare-land	475	1583
8	Hay-windrowed	200	478	Greenhouse	122	408
9	Oats	14	20	Lotus	194	648
10	Soybean-notill	200	972	Corn-notill	1632	5440
11	Soybean-mintill	200	2455	Woods	853	2844
12	Soybean-clean	200	593	Vegetable-field	69	231
13	Wheat	143	205			
14	Woods	200	1265			
15	Bldg-Grass-Tree-Drives	200	386			
16	Stone-Steel-Towers	75	93			
	Total	2466	10,249	Total	8646	28,824
NO.	**University of Pavia (UP)**			**Pavia Center (PC)**		
	Class	**Train.**	**Total**	**Class**	**Train.**	**Total**
1	Alfalfa	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
	Total	1800	42,776	Total	3925	148,152

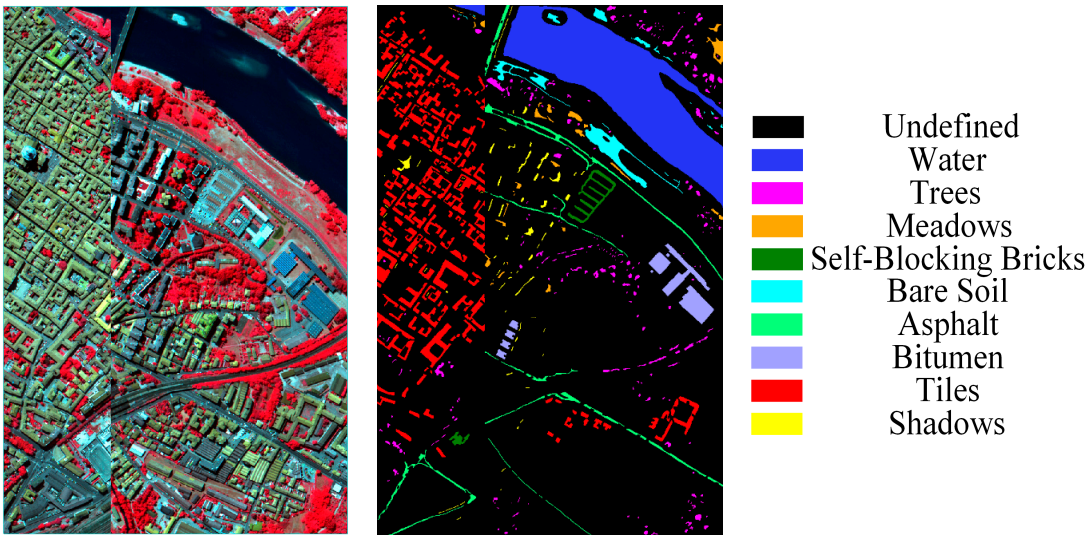
University of Pavia (UP): This urban dataset was gathered using the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [34] over a university area in Northern Italy in 2001. With a spatial resolution of 1.3 m per pixel, it includes 103 spectral bands at intervals of 4 nm from 430 to 860 nm after discarding noisy bands. It contains 610 × 340 pixels and nine urban land-cover types. False-color composite and ground truth maps of UP are shown in Figure 5. The numbers of training and test samples in this dataset are listed in Table 1.

Pavia Center (PC): This urban dataset was gathered with the ROSIS sensor, and includes 102 spectral bands at intervals of 4 nm ranging from 430 to 860 nm after removal of noisy bands. With a spatial resolution of 1.3 m per pixel, it contains 1096 × 715 pixels after removing a strip with no information [35] and includes 9 ground cover types. False-color composite and ground truth maps of PC are shown in Figure 6. The numbers of training and test samples in this dataset are listed in Table 1.



(**a**) (**b**)

Figure 5. University of Pavia dataset: (**a**) false-color composite of the University of Pavia, (**b**) ground-truth map.



(**a**) (**b**)

Figure 6. Pavia Center dataset: (**a**) false-color composite of Pavia Center, (**b**) ground-truth map.

GF-5 dataset of Anxin County (GF5AX): As an important scientific research satellite in the Chinese Key Project of the High-resolution Earth Observation System [36], the GF-5 satellite carries the Advanced Hyper-spectral Imager (AHSI), which currently has the highest spectral resolution for China. The GF5AX dataset was gathered using AHSI from Anxin County, Xiong'an New Area, Hebei Province in 2018. The main payload of the GF-5 satellite, AHSI has a spatial resolution of 30 m per pixel and a spectral resolution of 5 nm for visible and near-infrared (400–1000 nm) and 10 nm for short-wave infrared (1000–2500 nm) wavelengths. The GF5AX dataset contains 1340 × 853 pixels and 12 types of ground cover. After removing bands showing water absorption and noise, 291 spectral bands from 400 to 2500 nm were analyzed. False-color composite and ground truth maps of GF5AX are shown in Figure 7. The numbers of training and test samples in this dataset are listed in Table 1.

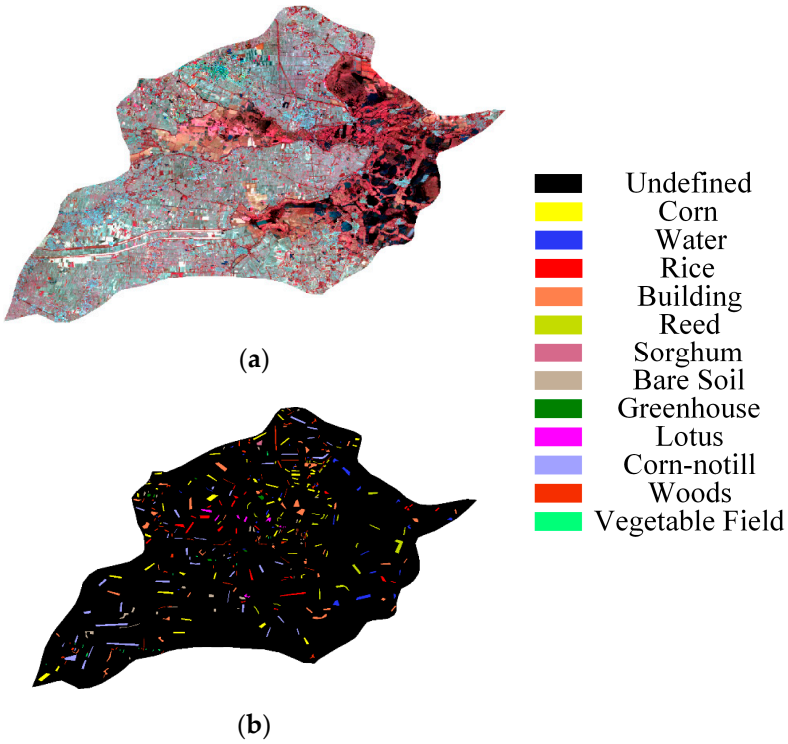


Figure 7. GF-5 dataset of Anxin County, Xiongan New Area: (**a**) false-color composite of GF-5, (**b**) ground-truth map.

3.2. Experimental Settings

The proposed SSCC framework, along with four other algorithms including SSLSTMs [28], SSRN [9], FDSSC [11], and 3D CNN [37], were configured with appropriate parameters to evaluate the efficiency of different approaches. Following the method of sample partitioning used in previous studies [11,24,38,39], the number of training samples selected randomly from all labeled samples for each dataset is provided in Table 1, with proportions of about 25% for IP, 4% for UP, 3% for PC, and 30% for GF5AX. Specifically, for IP and UP datasets, the numbers of training samples are the same with that in [39]. For PC dataset, based on the strategy of sample partitioning in [38], training samples are selected accounting for the approximate proportion of labeled samples with the UP dataset. In addition, the training samples of GF5AX dataset were selected referencing the method in [11]. Based on the optimal parameters of deep learning algorithms [11,28], all input samples for the four datasets have a spatial size of 65 × 65 for SSLSTMs, and 9 × 9 for four other algorithms, while the number of epochs and batch size of all algorithms were set to 20 and 32, respectively, based on the performance of the graphics processing unit (GPU). With a decay rate of 0.00001 for the Adam optimizer [40], the learning rate of the SSRN, FDSSC, and 3D CNN methods was 0.0001. The initial learning rate and

decay rate of the SSLSTMs method were set to 0.0005 and 0.00005, respectively. However, for the SSCC network, a dynamic learning rate was adopted and the learning rate decreased exponentially every five epochs. An initial learning rate of 0.0001 was used for the GF5AX dataset, while 0.0005 was used for the other three datasets.

In addition, with the goal of preventing overfitting, data augmentation was used to generate more samples for model training, resulting in sample sizes of 6400 for IP, 7200 for UP and PC, and 60,000 for GF5AX. The expanded samples in each dataset were divided randomly into three sets of training, validation and test samples at ratios of 70%, 20%, and 10%, respectively. To hasten SSCC network convergence, an activation function PReLU [41] and batch normalization (BN) [42] were used during model training. Using the parameter settings listed above, each classifier could achieve its optimal classification performance.

4. Experimental Results and Discussion

The performance of the SSCC model for HSI classification was compared to those of SSLSTMs, SSRN, FDSSC, and 3D CNN for each of the four datasets. The overall accuracy (OA), average accuracy (AA), kappa statistic (K), and normalized confusion matrix were used to evaluate the classification efficacy of each model. The average accuracy of the five experimental results was calculated as a comprehensive performance measure to obtain a more reliable estimate.

4.1. Experimental Results

Along with the corresponding ground-truth maps, classification maps and normalized confusion matrices are visualized in Figures 8–15. Tables 2–5 report the classification accuracy of each class for various datasets and classification methods. Based on comparative analyses of the classification maps, the proposed SSCC method generated smoother results than the other three methods. The classification maps of SSLSTMs, 3D CNN, SSRN, and FDSSC all contain varying degrees of noise, whereas SSCC output noise-free classification maps. In addition, the SSCC had higher classification accuracy and achieved better classification performance than the other methods in all four cases. These experimental results validate the robustness of the SSCC framework to difficult scenarios.

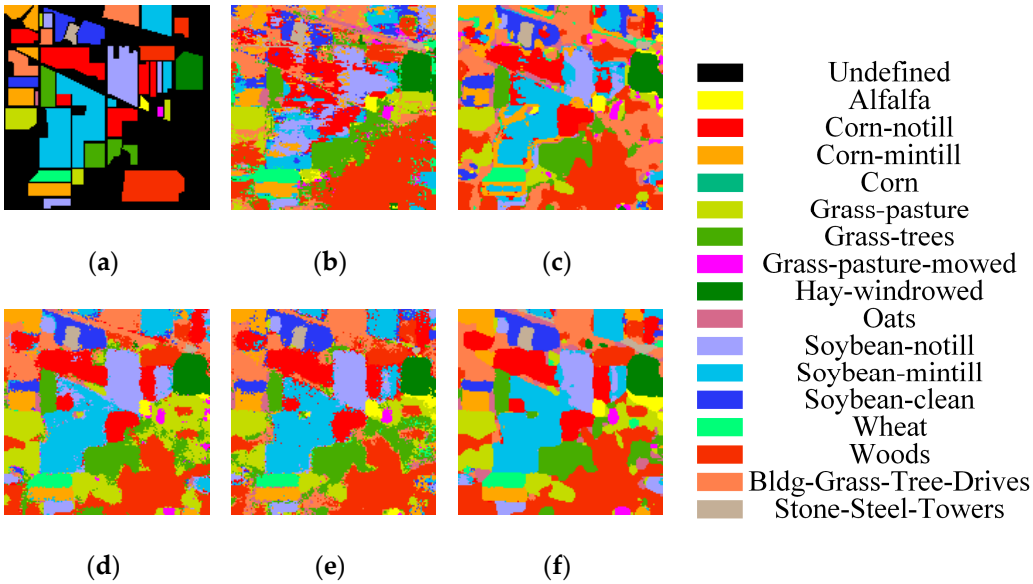


Figure 8. Classification maps for Indiana Pines dataset: (**a**) Ground-truth map, (**b**) SSLSTMs, OA = 67.30%, (**c**) 3D CNN, OA = 79.22%, (**d**) FDSSC, OA = 92.77%, (**e**) SSRN, OA = 95.21%, (**f**) SSCC, OA = 97.70%.

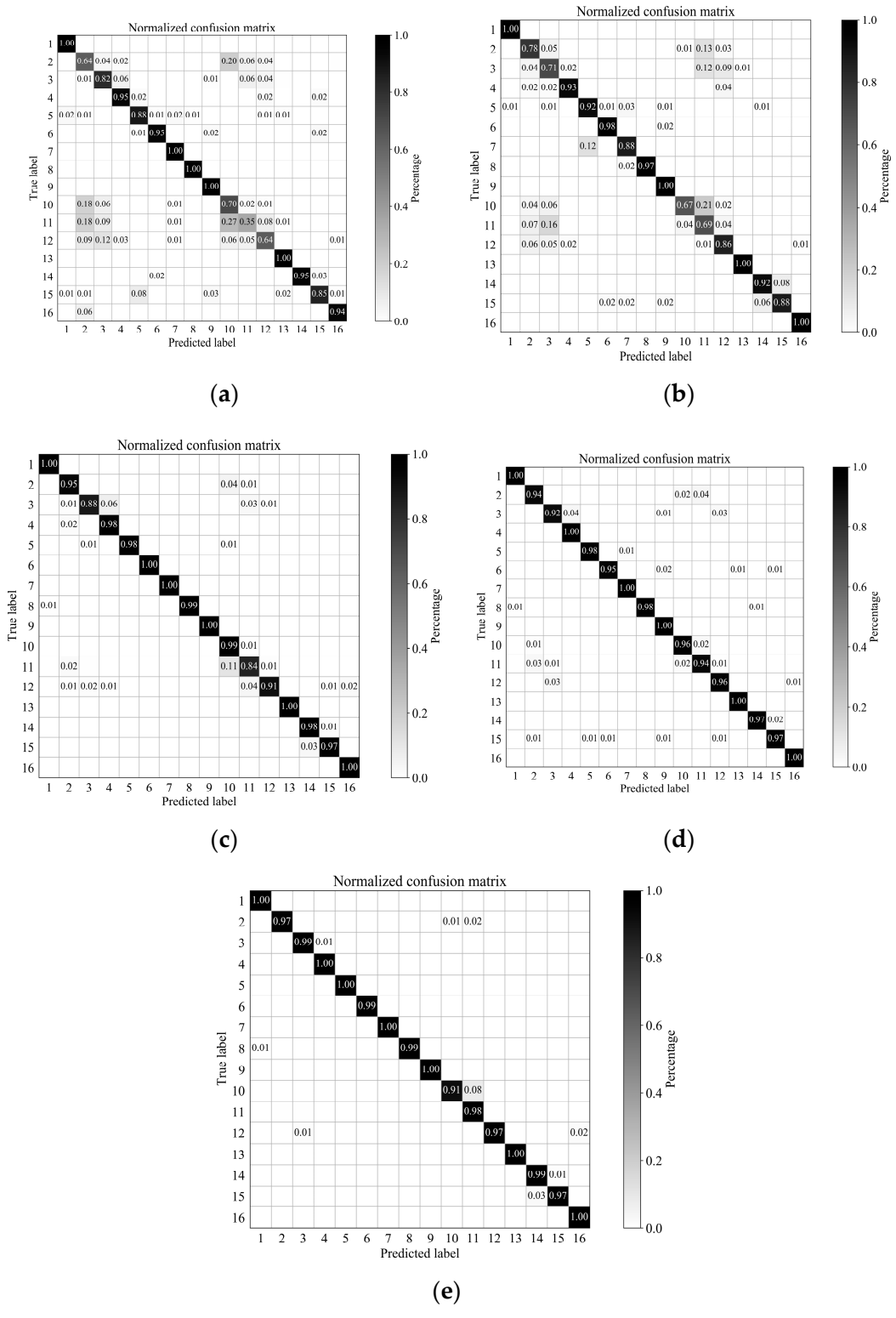


Figure 9. Indiana Pines dataset: Normalized confusion matrix of classification results using different methods (displaying value greater than 0.005): (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

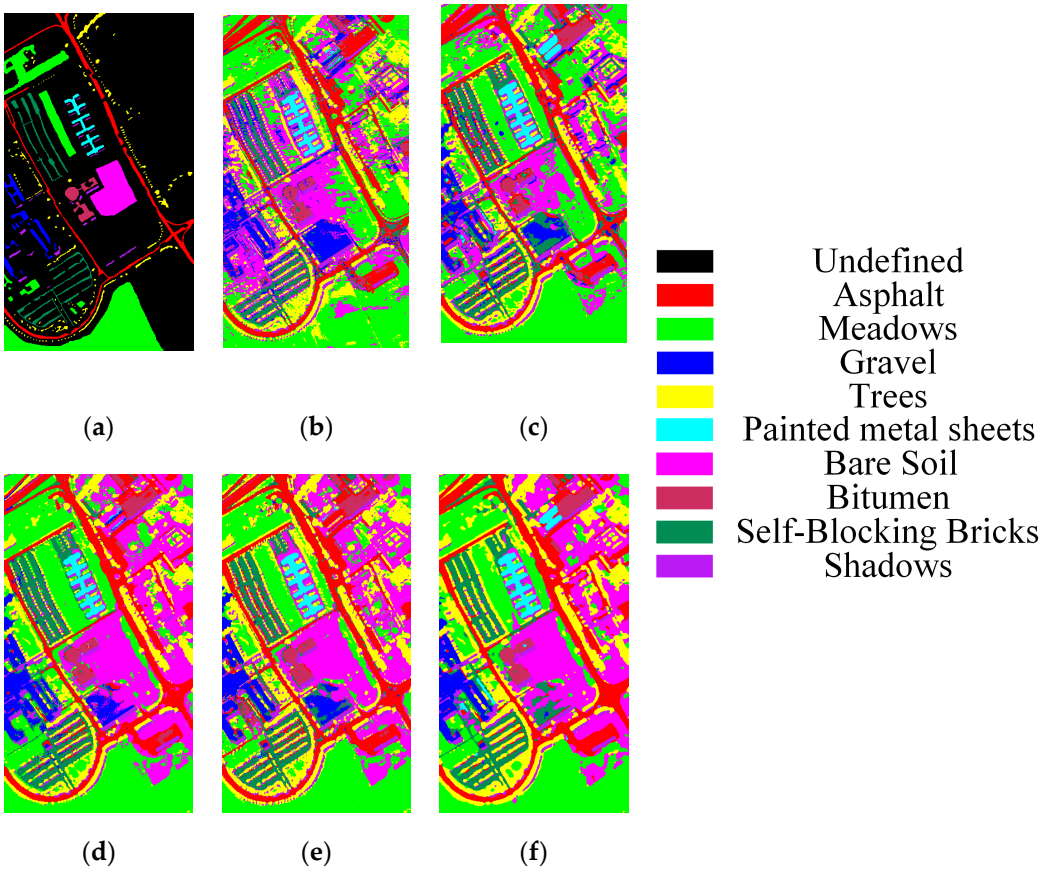


Figure 10. Classification maps for University of Pavia dataset: (**a**) Ground-truth map, (**b**) SSLSTMs, OA = 83.36%, (**c**) 3D CNN, OA = 91.71%, (**d**) FDSSC, OA = 98.39%, (**e**) SSRN, OA = 99.12%, (**f**) SSCC, OA = 99.71%.

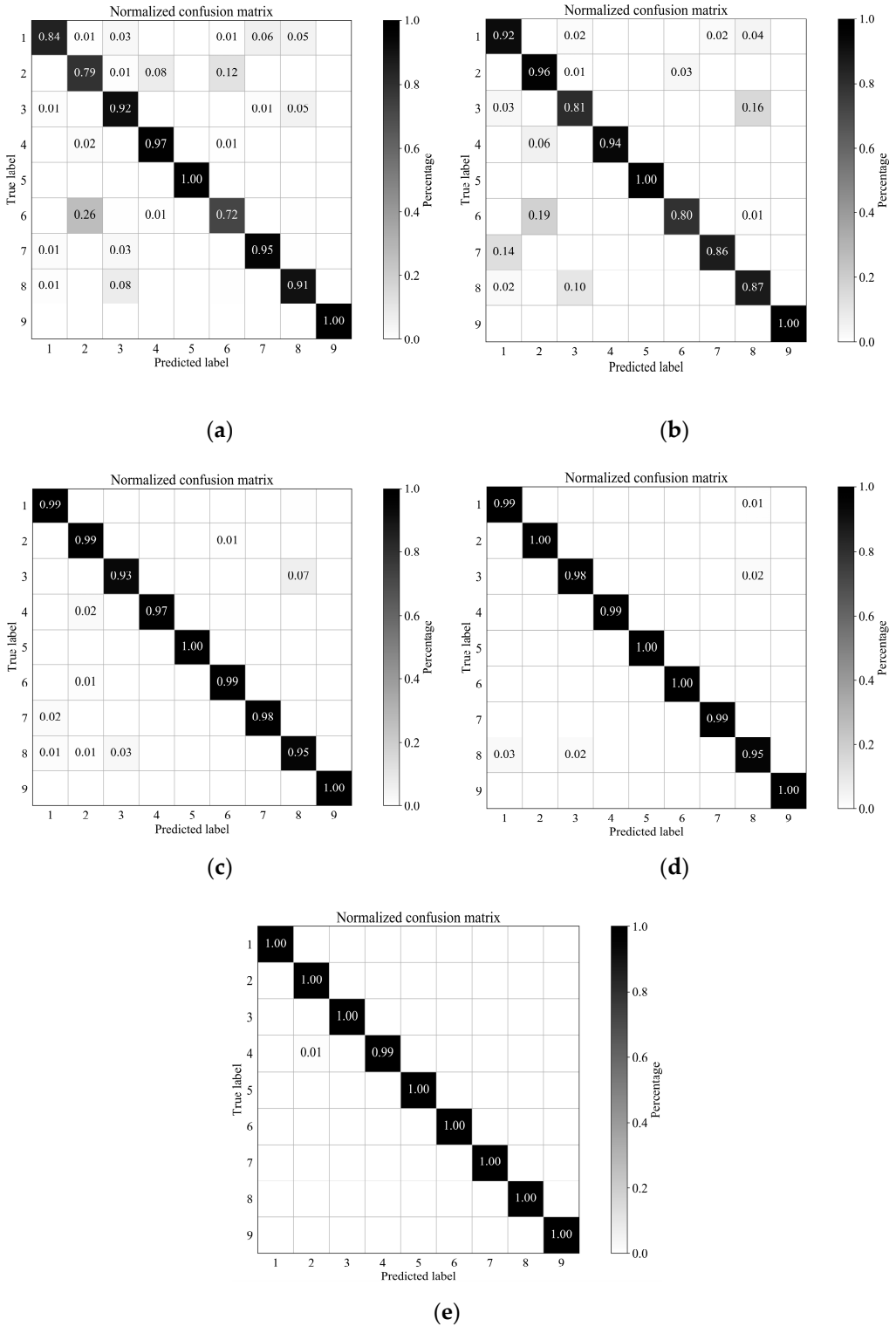


Figure 11. University of Pavia dataset: Normalized confusion matrix of classification results using different methods (displaying value greater than 0.005): (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

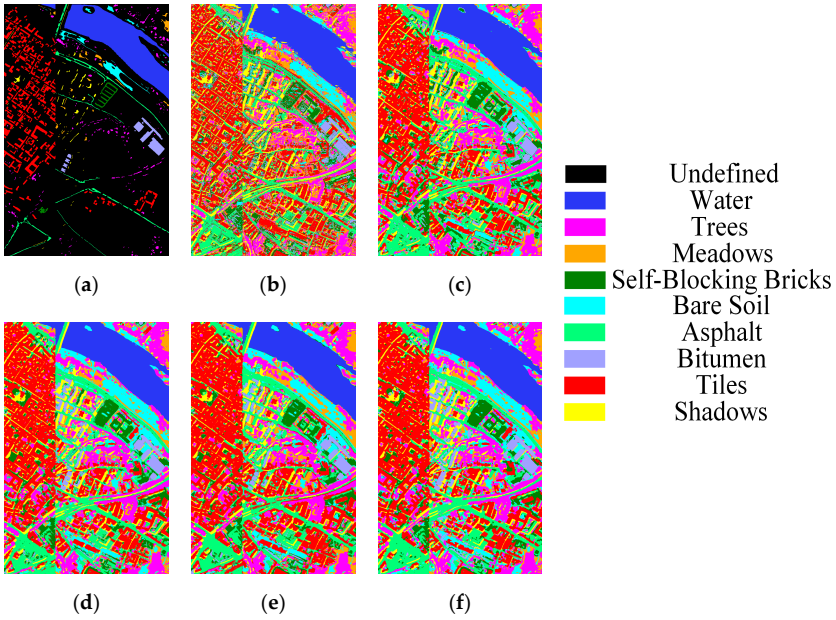


Figure 12. Classification maps for Pavia Center dataset: (**a**) Ground-truth map, (**b**) SSLSTMs, OA = 97.16%, (**c**) 3D CNN, OA = 98.33%, (**d**) FDSSC, OA = 99.83%, (**e**) SSRN, OA = 99.76%, (**f**) SSCC, OA = 99.84%.

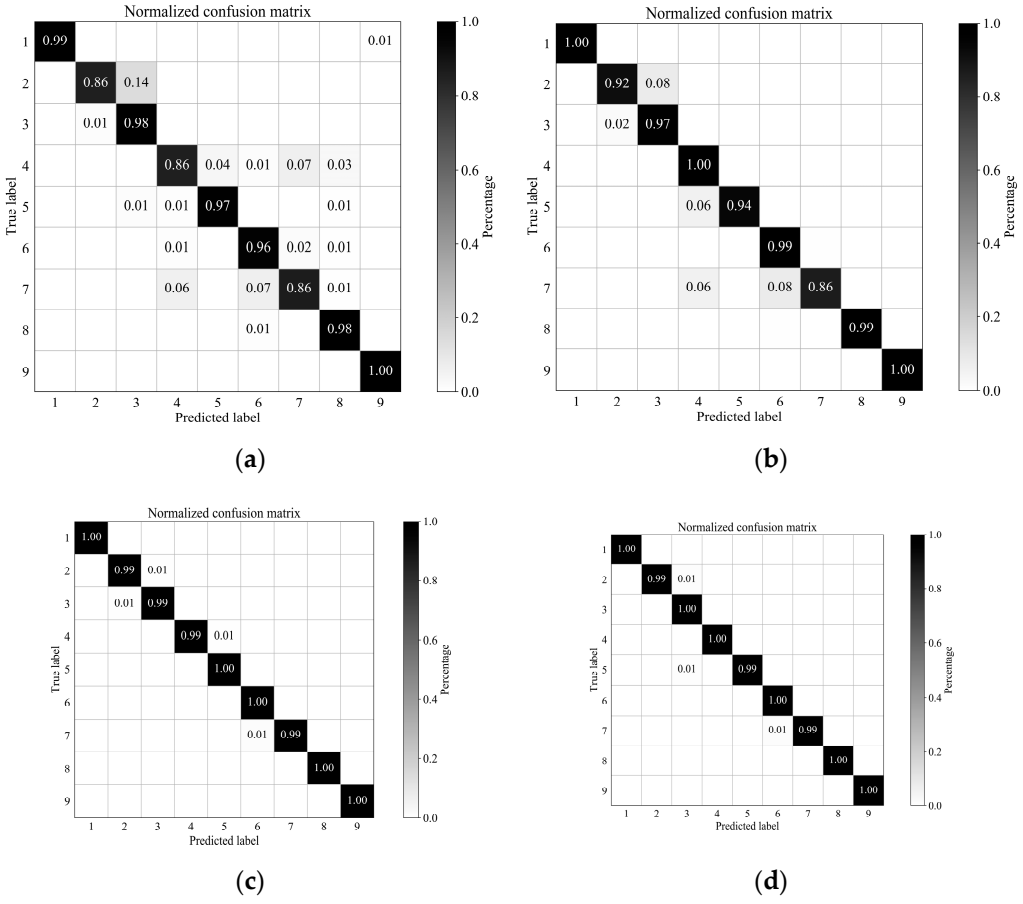
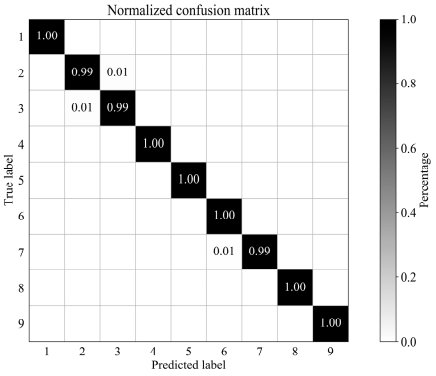


Figure 13. *Cont.*



(**e**)

Figure 13. University of Pavia: Normalized confusion matrix of classification results using different methods (displaying value greater than 0.005): (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

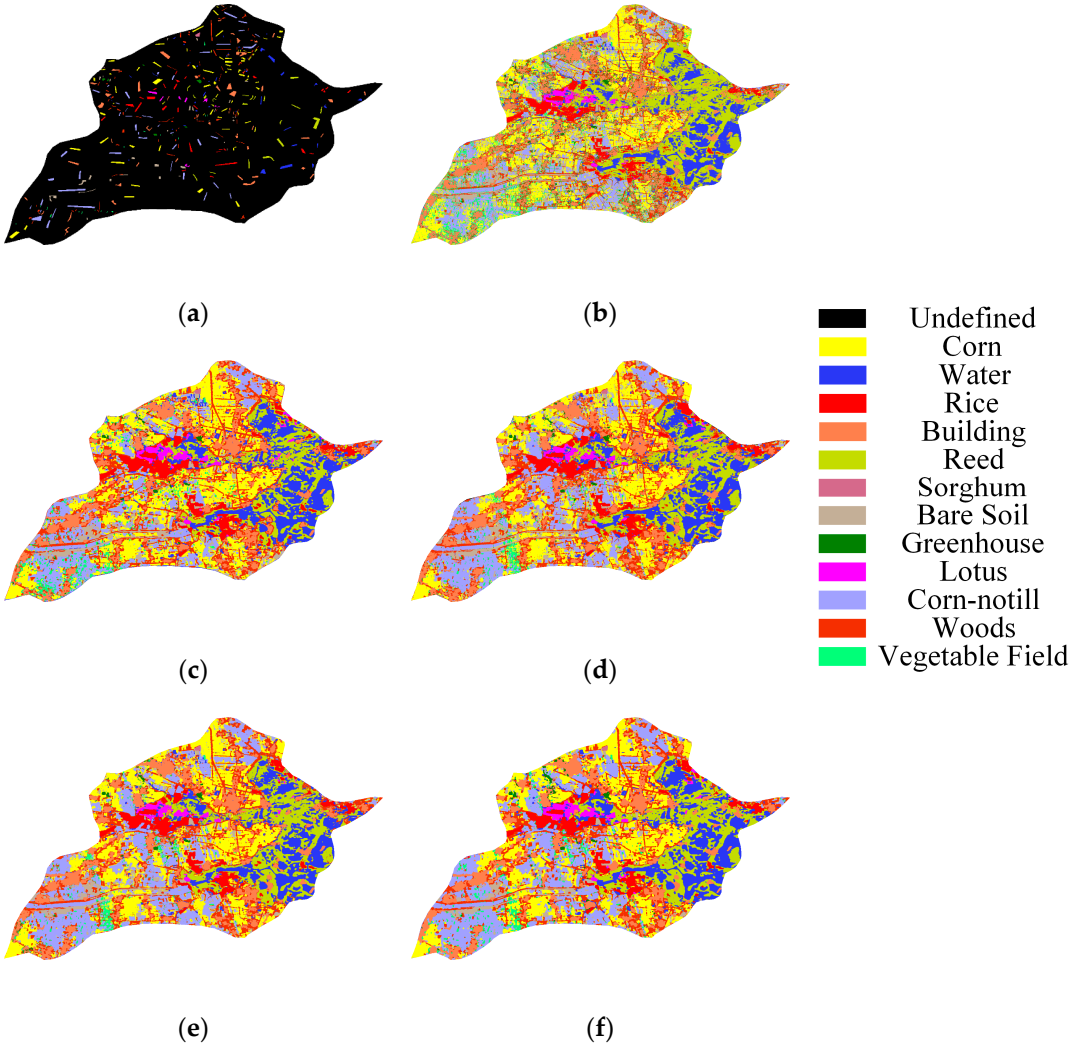


Figure 14. Classification maps for GF5AX dataset: (**a**) Ground-truth map, (**b**) SSLSTMs, OA = 87.47%, (**c**) 3D CNN, OA = 96.75%, (**d**) FDSSC, OA = 98.46%, (**e**) SSRN, OA = 99.32%, (**f**) SSCC, OA = 99.37%.

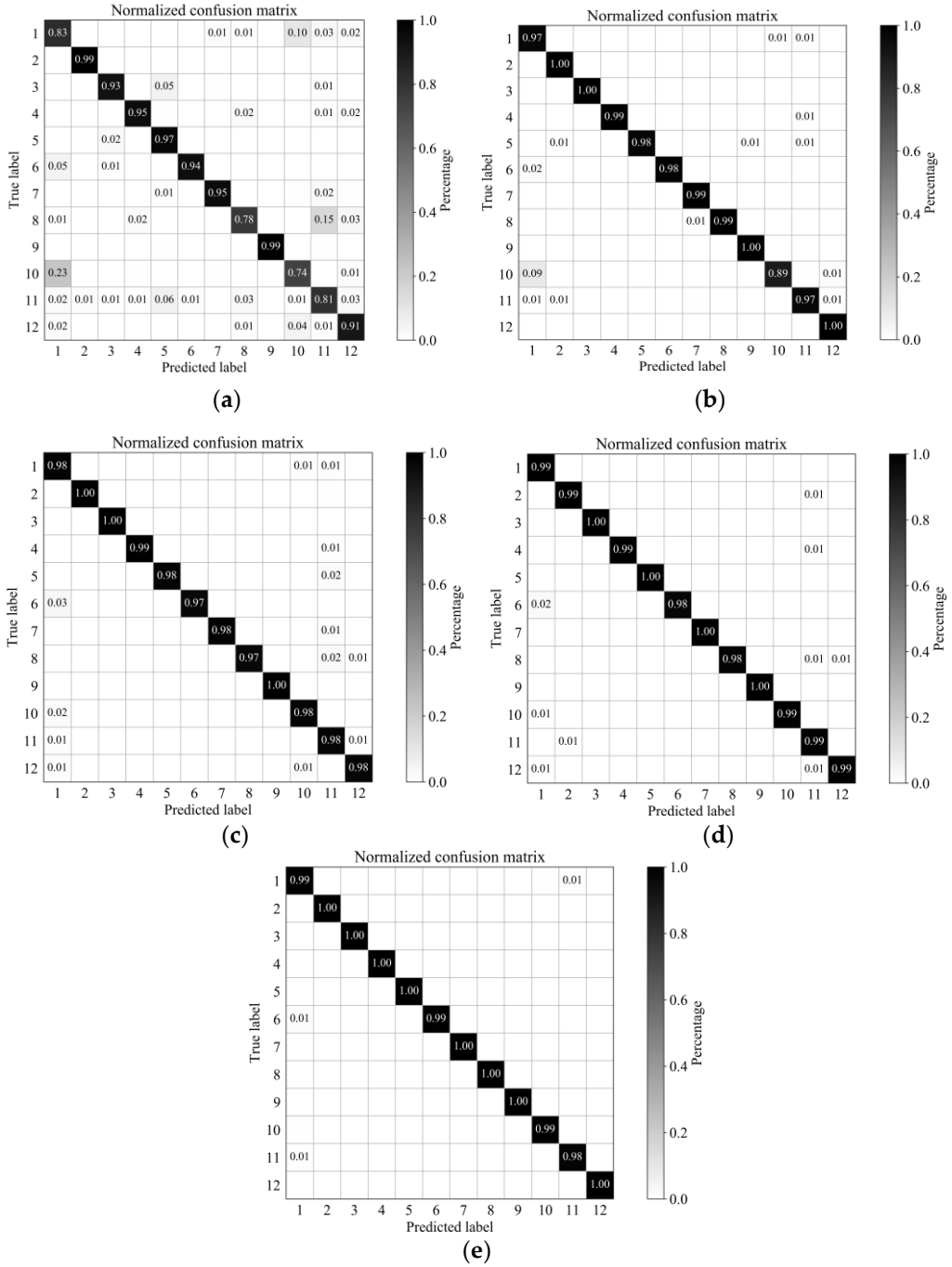


Figure 15. GF5AX dataset: Normalized confusion matrix of classification results using different methods (displaying value greater than 0.005): (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

Table 2. Classification results of different methods for the Indiana pines dataset.

NO.	Class	SSLSTMs	3D CNN	FDSSC	SSRN	SSCC
$\kappa \times 100$	/	62.87	75.74	91.53	94.35	97.28
AA (%)	/	85.40	88.65	96.76	97.40	98.59
OA (%)	/	67.30	79.22	92.77	95.21	97.70
1	Alfalfa	100.00	100.00	100.00	100.00	100.00
2	Corn-no till	63.52	77.93	94.79	93.73	97.07
3	Corn-min till	81.75	71.11	88.25	92.38	99.05
4	Corn	94.64	92.86	98.21	100.00	100.00
5	Grass-pasture	87.99	92.23	97.88	97.88	99.65
6	Grass-trees	94.53	97.74	99.81	95.28	99.43
7	Grass-pasture-mowed	100.00	87.50	100.00	100.00	100.00
8	Hay-windrowed	100.00	97.48	99.28	98.20	99.28
9	Oats	100.00	100.00	100.00	100.00	100.00
10	Soybean-no till	70.34	66.84	99.35	95.98	91.45
11	Soybean-min till	35.12	68.82	84.48	94.06	98.36
12	Soybean-clean	64.12	85.75	91.09	96.18	96.95
13	Wheat	100.00	100.00	100.00	100.00	100.00
14	Woods	94.93	92.49	98.22	97.46	98.97
15	Bldg-Grass-Tree-Drives	84.95	87.63	96.77	97.31	97.31
16	Stone-Steel-Towers	94.44	100.00	100.00	100.00	100.00

Table 3. Classification results of different methods for the University of Pavia dataset.

NO.	Class	SSLSTMs	3D CNN	FDSSC	SSRN	SSCC
$\kappa \times 100$	/	78.31	88.81	97.83	98.82	99.62
AA (%)	/	90.04	90.55	97.84	98.82	99.71
OA (%)	/	83.36	91.71	98.39	99.12	99.71
1	Alfalfa	84.48	91.90	98.82	98.99	99.69
2	Meadows	79.08	96.06	99.28	99.90	99.79
3	Gravel	91.73	80.67	92.94	97.79	99.53
4	Trees	97.35	93.65	97.17	99.37	98.88
5	Mental-sheets	99.83	100.00	100.00	100.00	100.00
6	Bare-soil	72.17	79.66	99.19	99.61	100.00
7	Bitumen	95.22	86.28	98.23	99.47	100.00
8	Bricks	90.52	86.73	94.89	94.54	99.54
9	Shadows	100.00	100.00	100.00	99.73	100.00

Table 4. Classification results of different methods for the Pavia Center dataset.

NO.	Class	SSLSTMs	3D CNN	FDSSC	SSRN	SSCC
$\kappa \times 100$	/	95.94	97.61	99.75	99.66	99.78
AA (%)	/	94.05	96.31	99.60	99.45	99.62
OA (%)	/	97.16	98.33	99.83	99.76	99.84
1	Water	99.25	99.99	100.00	100.00	99.98
2	Tree	85.58	91.62	99.47	98.56	98.86
3	Meadow	98.29	97.29	99.41	99.89	99.07
4	Brick	85.56	99.78	99.47	99.91	100.00
5	Bare soil	97.07	93.98	99.97	98.93	99.89
6	Asphalt	96.36	99.16	99.90	99.71	99.94
7	Bitumen	86.25	86.13	98.68	98.52	99.24
8	Tile	98.23	99.28	99.82	99.94	99.93
9	Shadow	99.88	99.55	99.72	99.55	99.68

Table 5. Classification results of different methods for the GF-5 dataset of Anxin County.

NO.	Class	SSLSTMs	3D CNN	FDSSC	SSRN	SSCC
$\kappa \times 100$	/	85.56	96.24	98.22	99.21	99.27
AA (%)	/	89.99	98.10	98.27	99.14	99.54
OA (%)	/	87.47	96.75	98.46	99.32	99.37
1	Corn	82.75	97.16	98.13	99.29	98.69
2	Water-body	99.45	99.86	99.52	99.38	100.00
3	Rice	92.85	99.64	99.94	99.94	99.70
4	Building	94.57	99.24	99.18	99.37	99.84
5	Reed	97.21	97.88	97.94	99.76	99.82
6	Sorghum	94.35	98.39	96.77	97.58	99.19
7	Bare-land	95.31	99.46	98.19	99.82	99.55
8	Greenhouse	77.97	99.30	96.50	98.25	100.00
9	Lotus	98.90	100.00	100.00	99.78	100.00
10	Corn-no till	74.29	89.29	97.79	99.16	99.40
11	Woods	80.91	97.04	97.69	98.54	98.24
12	Vegetable-field	91.36	100.00	97.53	98.77	100.00

Using the Indiana Pines dataset as an example, as illustrated in Figure 8, classification maps from the four comparative methods contain obvious speckles, particularly in the soybean-clean class of the classification map generated using SSLSTMs. The classification boundaries of different ground objects are ambiguous. By contrast, the soybean-clean class of the SSCC classification map is homogeneous, with less noise and clearer boundaries than those created using other methods. This comparison holds true for the remaining three datasets, including UP, PC, and GF5AX. The main reason for this difference is that the stronger feature extraction ability of SSCC allowed more discriminative spectral-spatial features of HSIs to be learned consecutively than is the case with other methods.

The quantitative results are consistent with those of qualitative analyses. Based on the classification accuracy presented in Table 2, the proposed SSCC model achieved higher classification accuracy than SSLSTMs, 3D CNN, FDSSC, or SSRN under the same parameter settings. The OA, AA, and Kappa coefficient of the SSCC model are 97.70%, 98.59%, and 0.9728, respectively. In contrast to the results of SSLSTMs, 3D CNN, SSRN, and FDSSC, the OA (AA) of SSCC indicated improvements of 30.40% (13.19%), 18.48% (9.94%), 2.49% (1.83%), and 4.93% (1.19%) for the IP dataset. Since more representative features are extracted, SSCC boosts the performance of state-of-the-art classifiers.

According to the normalized confusion matrix of each method, shown in Figure 9, the SSCC model mistakenly classified most soybean-no till areas as soybean-min till in the Indiana Pines dataset. Among the 16 classes of ground objects, only one land cover type had a misclassification ratio greater than 0.03, which was driven by the spectral and spatial similarities between soybean-no till and soybean-min till. Compared to the SSCC model, more serious misclassification phenomena occurred among land cover types with the other four methods. Specifically, 11, 10, 4, and 6 types of ground objects had misclassification ratios greater than 0.03 for SSLSTMs, 3D CNN, FDSSC, and SSRN, respectively.

As detailed in Table 3, for the University of Pavia dataset, the classification accuracies of all four methods surpassed 90%. The proposed SSCC model achieved the highest classification accuracy among algorithms using the same parameter settings. The OA, AA, and Kappa coefficient of the SSCC model are 99.71%, 99.71%, and 0.9962, respectively. Compared to the results of SSLSTMs, 3D CNN, FDSSC, and SSRN, the OA (AA) of SSCC showed small improvements of 16.35% (9.67%), 8% (9.16%), 1.32% (1.87%), and 0.59% (0.89%). A similar pattern was obtained for Kappa coefficients.

As presented in Figure 11, based on the normalized confusion matrix obtained using SSCC on the University of Pavia dataset, only one type of ground object was misclassified. For almost 1% of pixels, trees were classified as meadows. The proposed method exhibited strong classification performance, with classification accuracy exceeding 0.99 for all ground objects. Using this method, the urban classes were well delineated and distinguished. However, four classes including gravel, trees, bitumen, and

bricks could not be effectively distinguished by the FDSSC method, with misclassification ratios greater than 0.01. Two classes, gravel and bricks, were misclassified by the SSRN method in more than 1% of pixels. Furthermore, six and seven of the nine ground object types were misclassified by SSLSTMs and 3D CNN, with classification accuracies of less than 0.97, respectively. For the five classification methods of SSLSTMs, 3D CNN, FDSSC, SSRN, and SSCC, the lowest classification accuracy among the nine land cover types is 72.17%, 79.66%, 92.94%, 94.54%, and 98.88%, respectively. The SSCC proposed in this paper showed nearly perfect classification performance on the University of Pavia dataset, in contrast to the other four methods.

For the Pavia Center dataset, Table 4 summarizes the results of all five classification methods, indicating high classification accuracy with OA (AA) values of 97.16 (94.05), 98.33 (96.31), 99.83 (99.60), 99.76 (99.45), and 99.84 (99.62) for SSLSTMs, 3D CNN, FDSSC, SSRN, and SSCC, respectively. Moreover, the Kappa coefficients of all five methods are above 0.95. Since the classification accuracies of the five methods used are relatively high, compared to the SSLSTMs, 3D CNN, FDSSC, and SSRN methods, the OA (AA) of SSCC indicated slight improvements of 2.68% (5.57%), 1.51% (3.31%), 0.01% (0.02%), and 0.08% (0.17%). Nonetheless, SSCC achieved the best classification performance among the four methods.

Figure 13 presents the normalized confusion matrix of each method for the Pavia Center dataset. Three types of ground objects were misclassified at ratios of 1% among all classified pixels in the class using both the SSCC and SSRN methods. Meanwhile, four types of ground objects were misclassified with each of the 3D CNN and FDSSC methods. Almost seven types of ground objects were misclassified by the SSCLSTMs method. In particular, serious misclassification phenomena occurred in the SSLSTMs and 3D CNN classification maps, which had a maximum percentage of pixels misclassified of 7% and 8%, respectively, whereas those of the other three methods were only 1%.

After experiments using three well-known hyperspectral datasets, the GF5AX dataset was also used to evaluate the classification performance of SSCC. As shown in Table 5, compared to SSLSTMs, 3D CNN, FDSSC, and SSRN, the proposed SSCC model outperformed all existing methods slightly, with OA, AA, and Kappa coefficient values of 99.37%, 99.54%, and 0.9927, respectively. In addition, four types of ground objects were classified correctly with classification accuracies of 100%.

Figure 15 shows the normalized confusion matrix of each method based on the GF5AX dataset. Three types of ground objects were misclassified by SSCC, and more types of ground objects were misclassified by all of the existing methods. For example, almost 26% of Corn-no till pixels were mistaken as other classes using SSLSTMs. Ten, five, eight, and two types of ground objects have misclassification ratios greater than 0.01 for SSLSTMs, 3D CNN, FDSSC, and SSRN, respectively. By contrast, the SSCC model had only one type of land cover misclassified at a ratio of 0.02. Thus, the SSCC method can be applied to the GF5AX dataset to achieve better classification results.

To monitor the training process, accuracy evolution in terms of epochs of training and validation data are shown in Figures 16–19. For the IP, UP, and PC datasets, the results indicate the average level from five experiments. For the GF5AX dataset, due to the large number of expanded training samples and spectral bands, the results of a single experiment are presented. For these four hyperspectral datasets, the SSCC method obtained the highest classification accuracy for the validation data compared to existing methods. Although the SSRN converged faster than SSCC on the training data, the SSCC method yielded greater accuracy than SSRN for validation data in the last few epochs. In addition, for three well-known hyperspectral datasets, SSCC generated the smallest standard deviation among the five methods, particularly in the last five epochs of model validation.

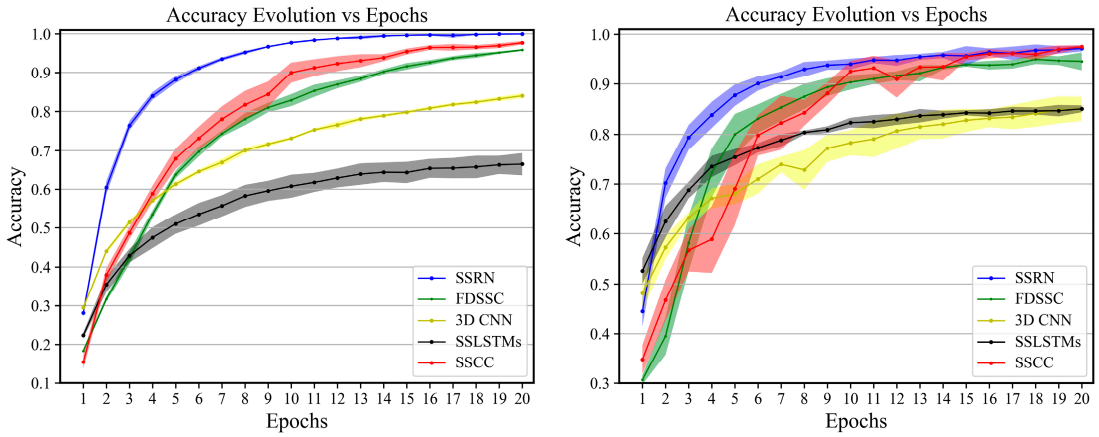


Figure 16. Indiana Pines: Evolution of the accuracy in terms of epochs on (**a**) training data and (**b**) validation data. The shadow shows the standard deviation of the accuracy for five executions.

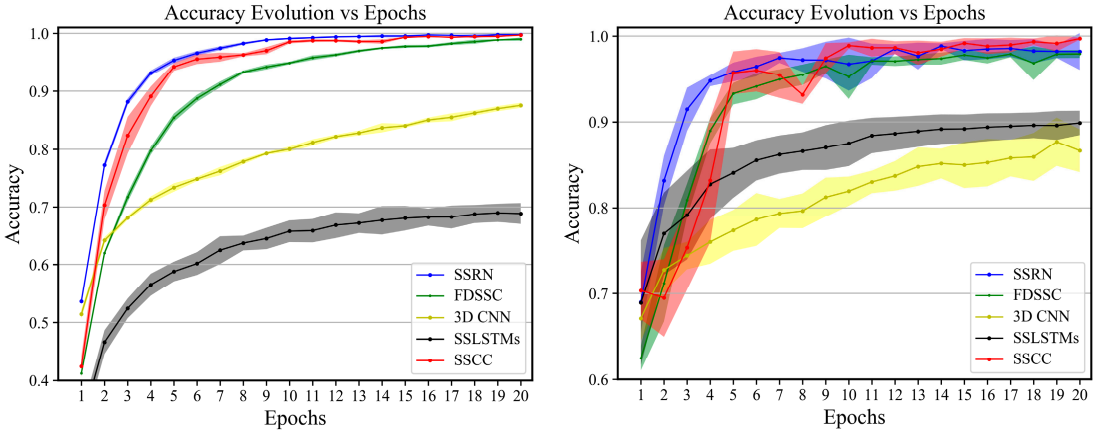


Figure 17. University of Pavia: Evolution of the accuracy in terms of epochs on (**a**) training data and (**b**) validation data. The shadow shows the standard deviation of the accuracy for five executions.

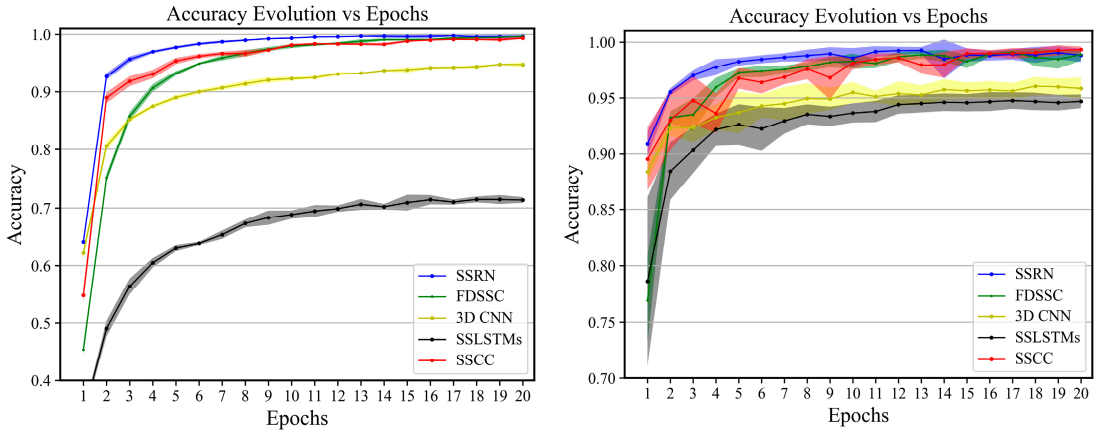
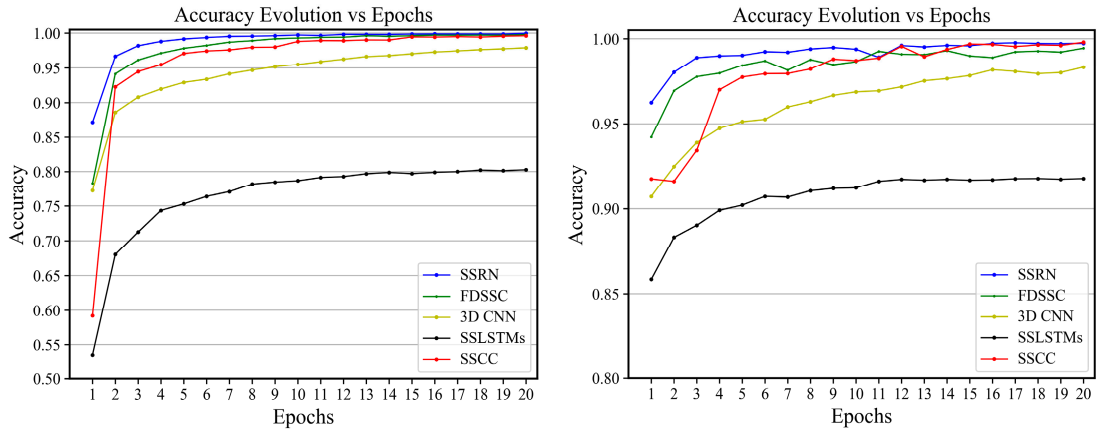


Figure 18. Pavia Center: Evolution of the accuracy in terms of epochs on (**a**) training data and (**b**) validation data. The shadow shows the standard deviation of the accuracy for five executions.



(**a**) (**b**)

Figure 19. GF-5 datasets of Anxin County: Evolution of the accuracy in terms of epochs on (**a**) training data and (**b**) validation data.

4.2. Discussion

Based on the experimental results from four hyperspectral datasets, the effectiveness of the proposed SSCC framework was validated for multiple scenarios. Notably, model hyperparameters that are difficult to determine are essential to training deep learning models. Moreover, the shortage of labeled pixels is the greatest challenge to HSI classification, particularly for classification methods that use neural networks [43]. In this paper, to reduce model training time and avoid conducting multiple experiments with each dataset to identify the best parameters for various models, we fine-tuned parameter settings used in previous studies to determine the optimal parameters for our five models through only a few experiments.

Due to the limited availability of samples, data augmentation, which is useful for training deep learning models, was employed to generate additional virtual samples and enhance the generalization capacity of models in an intuitive manner. The training samples were randomly selected from each class of the IP, UP, and PC datasets as described previously [39], while the training samples from GF5AX were selected following a different strategy [24] due to the lower spatial resolution and greater number of spectral channels. Through image blur, noise addition, and 90°, 180°, and 270° rotations, the number of training samples for each class listed in Table 1 was expanded by around two to four times for various datasets [44] to improve the robustness of the model. Each class contained the same number of samples after data augmentation, and the sample imbalance among different ground objects was mitigated through improved classification accuracy of ground objects. Meanwhile, the dropout and L2 regularization methods were used to avoid overfitting, and the dynamic learning rate strategy was used to carry out adaptive learning. Moreover, the PReLU activation function was used in SSCC rather than a rectified linear unit (ReLU) to address problems of neuronal death and the offset phenomenon [11]. When the output is close to zero, PReLU converges faster than ReLU. By setting the above parameters and employing data augmentation technologies, the SSCC model with few residual connections only needs 20 epochs to achieve its highest classification accuracy, while the SSRN and FDSSC described previously [11] require 200 and 80 epochs, respectively. In addition, due to the very high computation times on the GF5AX dataset caused by the large number of training samples and spectral bands, the average training time per epoch and average testing time of five experiments on three other HSI datasets were calculated to evaluate the computational efficiency of all the methods. Table 6 shows that through the comparative analysis of different models, the average training time of the proposed SSCC is relatively less. After 20 training epochs, the SSCC model can converge in a shorter time and achieve a high classification accuracy. The average testing time of

the proposed SSCC is relatively long. The main reason is that the CLSTM operation in the SSCC contains more parameters and requires a larger amount of computational power in the testing stage. A similar pattern was obtained for SSLSTMs method. Since the spatial and spectral features were both extracted by LSTM, the SSLSTMs is more computationally expensive in contrast to the SSCC method. However, in the future, multithread parallel computing is an effective way to improve the speed of model testing and alleviate the extra computational costs. Therefore, based on the limited training samples, the overall performance of the SSCC model is good, and it can be used as an effective method for hyperspectral classification.

Table 6. The average testing times and training time for different methods on three HSI datasets.

Method\Datasets	**IP**		**UP**		**PC**	
\	Train (s)	Test (m)	Train (s)	Test (m)	Train (s)	Test (m)
SSLSTMs	189.19	49.12	142.74	469.31	142.25	1542.33
3D CNN	48.62	1.72	44.12	15.73	47.14	57.53
FDSSC	46.14	3.41	44.91	32.41	45.28	123.77
SSRN	49.50	3.79	45.17	36.01	43.87	133.02
SSCC	46.09	6.50	44.58	61.94	43.76	234.38

For further comparative analyses of the classification performance of various classifiers, the accuracy or loss evolution in terms of epochs on training and validation data was determined, and the results are provided in Figures 20 and 21. When labeled training data are limited, deep learning models may learn the specific patterns of a few training samples perfectly, leading to poor performance on test data. If the model is overfitted, training error is insignificant but test error is large [45,46]. Using the Indiana Pines dataset as an example, Figure 20 shows the accuracy or loss evolution in terms of epochs on training and validation data. The training and validation accuracy of SSLSTMs is the lowest among the five classification methods after 20 training epochs, demonstrating that the SSLSTMs model has limited ability to extract spatial and spectral features. At the end of model training, the training and validation accuracy of 3D CNN are almost identical, but both of them are relatively low. The training accuracy of SSRN is equal to its validation accuracy at the 7th epoch, but this method has low accuracy. With follow-up training of the SSRN model, the training accuracy exceeds the validation accuracy and a lower training error is obtained than validation error, indicating the risk for model overfitting. The accuracy of FDSSC increases with the number of training epochs, and the training accuracy is equal to the validation accuracy at the 19th epoch, but with a larger standard deviation than SSCC. Although the SSCC model converges slowly with a large standard deviation in the early training epochs, the training accuracy and validation accuracy are highest among the four methods after training. The training accuracy and validation accuracy are almost equal during the last few epochs, as evidenced by the two precision curves in Figure 20 almost overlapping. With fewer residual connections compared to FDSSC, the SSCC framework converges rapidly with minimal standard deviation in the last training epochs, and achieves the best classification performance after 20 training epochs without overfitting, supporting the effectiveness of the proposed algorithm. Using the GF5AX dataset, most classifiers reached high classification accuracy after 20 training epochs. This difference is mainly because the GF5 dataset has sufficient training samples generated through data augmentation, and the model can extract deep abstract discriminative features from HSIs. The robustness of all five deep learning models became stronger with more training samples. Compared to the other four methods, the SSCC framework achieves the highest classification accuracy, and its validation loss is almost always less than the training loss.

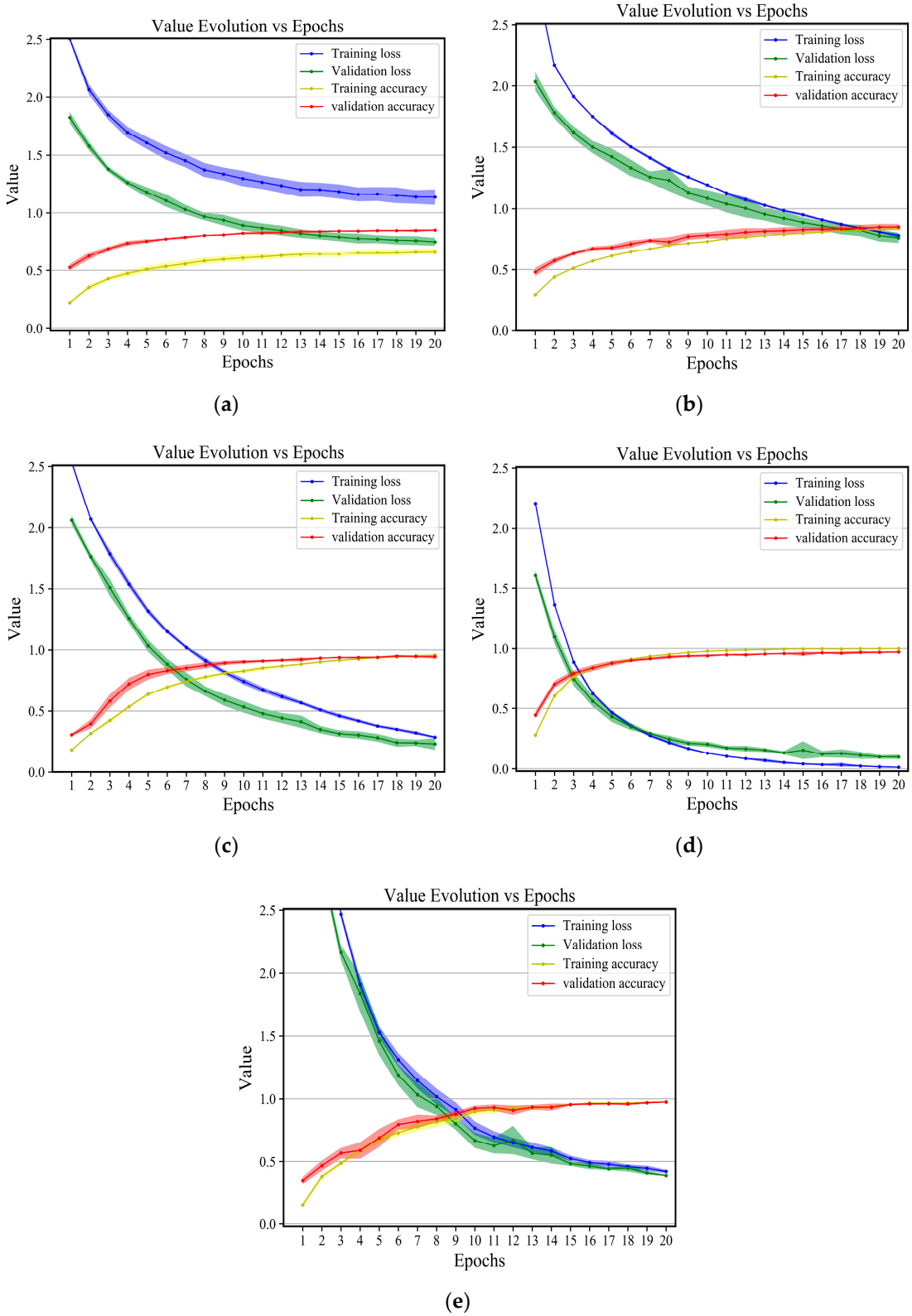


Figure 20. Indiana Pines dataset: Evolution of the accuracy or loss in terms of epochs on training and validation data. The shadow shows the standard deviation of the values for five executions: (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

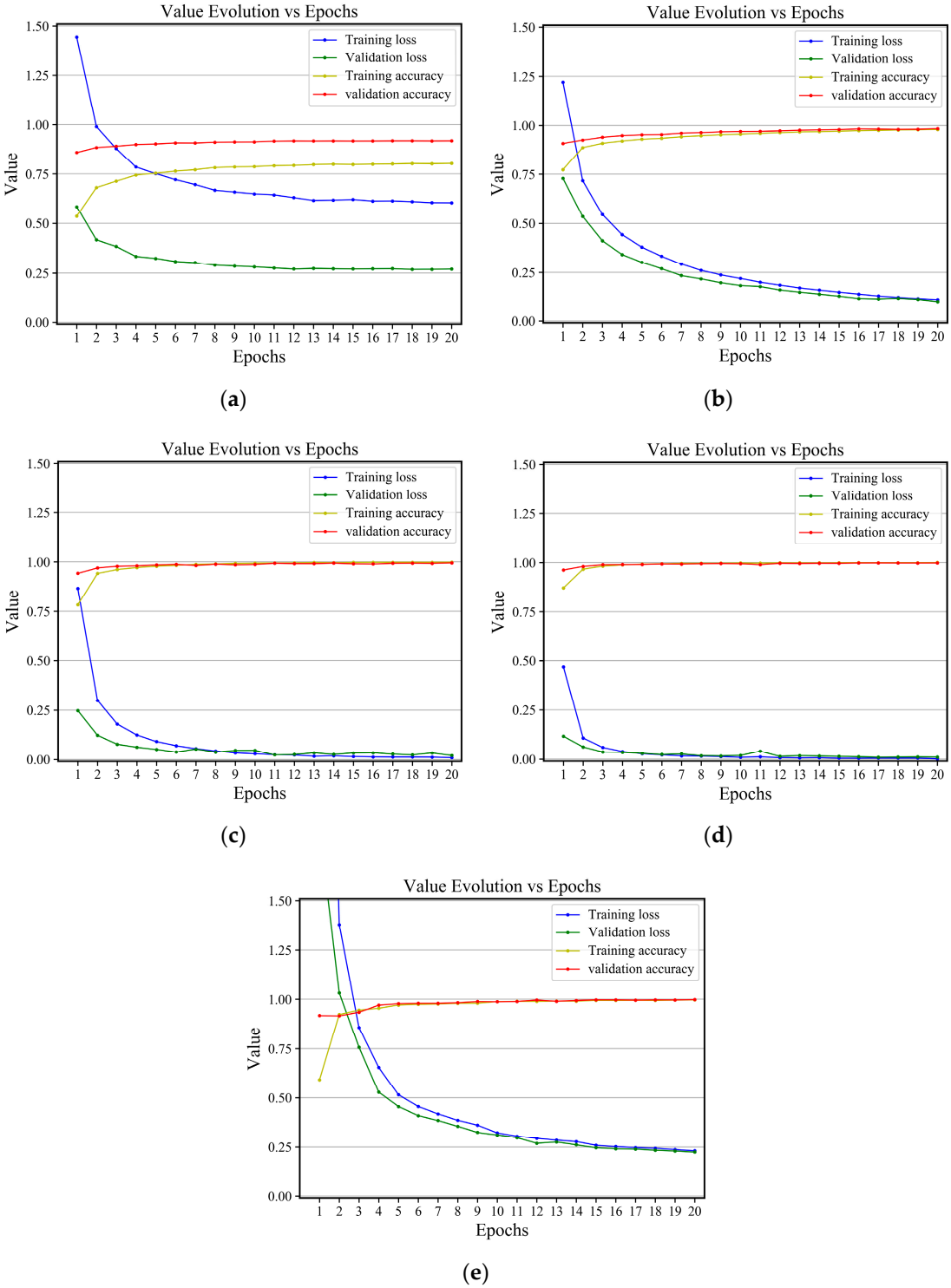


Figure 21. GF5AX dataset: Evolution of the accuracy or loss in terms of epochs on training and validation data. (**a**) SSLSTMs, (**b**) 3D CNN, (**c**) FDSSC, (**d**) SSRN, (**e**) SSCC.

Based on the training epoch and classification accuracy results for the models, the SSCC model is capable of achieving state-of-the-art performance for HSI classification.

5. Conclusions and Future Work

The extraction and utilization of spatial and spectral features are crucial for HSI classification with deep learning algorithms. In this paper, we proposed an end-to-end deep learning framework called

the spectral-spatial cascaded 3D CNN with a convolutional LSTM network (SSCC) for HSI classification. The proposed SSCC framework uses two parallel branches to efficiently capture features in both the spectral and spatial feature learning processes. Convolutional LSTM layers were used to provide more discriminative and deeper spectral-spatial features for HSI classification. Due to its simple structure, SSCC has fewer residual connections than FDSSC. Moreover, data augmentation was used to expand the limited set of training samples. The dynamic learning rate, BN, PReLU activation function, dropout layers, and L2 regularization were introduced into the 3D convolutional neural network to accelerate convergence of the model. In various classification scenarios including agricultural, urban and rural-urban areas, experimental results reveal that the SSCC framework requires only 20 epochs to achieve better classification accuracy than other state-of-the-art approaches without overfitting.

In future works, additional hyperspectral datasets will be used to further verify the robustness of the proposed SSCC algorithm. In addition, transfer learning will be introduced into HSI classification to provide a priori information and accelerate the model.

Author Contributions: W.Q. performed the experiments, analyzed the results and wrote the paper; X.Z. designed the experiments, supervised the study and revised the paper; N.W. and Y.C. gave valuable suggestions for the paper; M.Z. prepared the datasets.

Funding: This research 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).

Acknowledgments: The authors are grateful to the editor and reviewers for their constructive comments, which have significantly improved this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liang, L.; Di, L.; Zhang, L.; Deng, M.; Qin, Z.; Zhao, S.; Hui, L. Estimation of crop LAI using hyperspectral vegetation indices and a hybrid inversion method. *Remote Sens. Environ.* **2015**, *165*, 123–134. [CrossRef]
2. Zhang, S.; Li, j.; Wu, Z.; Placa, A. Spatial Discontinuity-Weighted Sparse Unmixing of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5767–5779. [CrossRef]
3. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.M.; Chanussot, J. Hyperspectral Remote Sensing Data Analysis and Future Challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
4. Hughes, G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory* **1968**, *14*, 55–63. [CrossRef]
5. Qi, W.; Meng, Z.; Li, X. Locality Adaptive Discriminant Analysis for Spectral–Spatial Classification of Hyperspectral Images. *IEEE Geosci. Remote Sens.Lett.* **2017**, *14*, 2077–2081.
6. Rajan, S.; Ghosh, J.; Crawford, M.M. An Active Learning Approach to Hyperspectral Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1231–1242. [CrossRef]
7. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
8. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-Spatial Classification of Hyperspectral Data Using Loopy Belief Propagation and Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 844–856. [CrossRef]
9. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [CrossRef]
10. Jia, X.; Kuo, B.C.; Crawford, M.M. Feature Mining for Hyperspectral Image Classification. *Proc. IEEE.* **2013**, *101*, 676–697. [CrossRef]
11. Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A Fast Dense Spectral–Spatial Convolution Network Framework for Hyperspectral Images Classification. *Remote Sens.* **2018**, *10*, 1068. [CrossRef]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2012, Volume 25, pp. 1097–1105. Available online: https:

//papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks (accessed on 11 October 2019).
13. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. 2014. Available online: https://www.cs.unc.edu/~{}wliu/papers/GoogLeNet.pdf (accessed on 10 October 2019).
14. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
15. Chen, Y.; Lin, Z.; Xing, Z.; Gang, W.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *7*, 2094–2107. [CrossRef]
16. Hu, W.; Huang, Y.; Li, W.; Fan, Z.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
17. Lin, Z.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063.
18. Mou, L.; Ghamisi, P.; Xiao, X.Z. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
19. Hang, R.; Liu, Q.; Hong, D.; Ghamisi, P. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1–11. [CrossRef]
20. Zhong, P.; Gong, Z.; Li, S.; Schonlieb, C.B. Learning to Diversify Deep Belief Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3516–3530. [CrossRef]
21. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 212–216. [CrossRef]
22. Ghamisi, P.; Maggiori, E.; Li, S.; Souza, R.; Tarablaka, Y.; Moser, G.; De Giorgi, A.; Fang, L.; Chen, Y.; Chi, M. New Frontiers in Spectral-Spatial Hyperspectral Image Classification: The Latest Advances Based on Mathematical Morphology, Markov Random Fields, Segmentation, Sparse Representation, and Deep Learning. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 10–43. [CrossRef]
23. Vetrivel, A.; Gerke, M.; Kerle, N.; Nex, F.; Vosselman, G. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *Isprs J. Photogramm. Remote Sens.* **2018**, *140*, 45–49. [CrossRef]
24. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.K.; Zhang, X.; Huang, X. Hyperspectral Image Classification With Deep Learning Models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [CrossRef]
25. Gong, C.; Li, Z.; Han, J.; Yao, X.; Lei, G. Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6712–6722.
26. Liu, Q.; Feng, Z.; Hang, R.; Yuan, X. Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1330.
27. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Advances in Neural Information Processing Systems*. 2015, Volume 28, pp. 1049–5258. Available online: https://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting (accessed on 10 October 2019).
28. Zhou, F.; Hang, R.; Liu, Q.; Yuan, X. Hyperspectral Image Classification Using Spectral-Spatial LSTMs. *Neurocomputing* **2019**, *328*, 39–47. [CrossRef]
29. Li, Y.; Zhang, H.; Shen, Q. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
30. Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaza, A. Deep&Dense Convolutional Neural Network for Hyperspectral Image Classification. *Remote Sens.* **2018**, *10*, 1454.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*. 2016, Volume 90, pp. 770–778. Available online: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf (accessed on 10 October 2019).
32. Cho, J.; Park, K.S.; Karki, M.; Lee, E.; Ko, S.; Kim, J.K.; Lee, D.; Choe, J.; Son, J.; Kim, M. Improving Sensitivity on Identification and Delineation of Intracranial Hemorrhage Lesion Using Cascaded Deep Learning Models. *J. Digit. Imaging.* **2019**, *32*, 450–461. [CrossRef] [PubMed]
33. Green, R.O.; Eastwood, M.L.; Sarture, C.M.; Chrien, T.G.; Aronsson, M.; Chippendale, B.J.; Faust, J.A.; Pavri, B.E.; Chovit, C.J. Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). *Remote Sens. Environ.* **1998**, *65*, 227–248. [CrossRef]

34. Kunkel, B.; Blechinger, F.; Lutz, R.; Doerffer, R.; Van der Piepen, H.; Schroder, M. ROSIS (Reflective Optics System Imaging Spectrometer)—A Candidate Instrument For Polar Platform Missions. *Optoelectronic Technologies for Remote Sensing from Space*. 1988. Available online: https://spie.org/Publications/Proceedings/Paper/10.1117/12.943611?SSO=1 (accessed on 10 October 2019).
35. Guo, A.J.; Fei, Z. Spectral-Spatial Feature Extraction and Classification by ANN Supervised with Center Loss in Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1755–1767. [CrossRef]
36. Sun, Y.; Jiang, G.; Li, Y.; Yang, Y.; Dai, H.; He, J.; Ye, Q.; Cao, Q.; Dong, C.; Zhao, S. GF-5 Satellite: Overview and Application Prospects. *Spacecr. Recovery Remote Sens.* **2018**, *39*, 1–13.
37. Liu, B.; Yu, X.; Zhang, P.; Tan, X.; Wang, R.; Zhi, L. Spectral–spatial classification of hyperspectral image using three-dimensional convolution network. *J. Appl. Remote Sens.* **2018**, *12*, 1–18.
38. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
39. Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaze, A. A new deep convolutional neural network for fast hyperspectral image classification. *Isprs J. Photogramm. Remote Sens.* **2018**, *145*, 120–147.
40. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization: Computer Science. 2014. Available online: http://arxiv.org/abs/1412.6980v8 (accessed on 10 October 2019).
41. He, K.; Zhang, X.; Ren, S.; Jian, S. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; Volume 123, pp. 1026–1034.
42. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on International Conference on Machine Learning JMLR.org, Lille, France, 6–11 July 2015.
43. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral Image Classification Using Deep Pixel-Pair Features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
44. Lee, H.; Kwon, H. Going Deeper with Contextual CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef] [PubMed]
45. Gupta, S.; Gupta, R.; Ojha, M.; Singh, K.P. A Comparative Analysis of Various Regularization Techniques to Solve Overfitting Problem in Artificial Neural Network. In *International Conference on Recent Developments in Science, Engineering and Technology*; Springer: Singapore, 2018; Volume 799, pp. 363–371.
46. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.