Spatial-Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification

arXiv:1905.03577v2 [cs.CV] 11 Apr 2022

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DOI:10.1109/TGRS.2019.2961947.

Wen-Shuai Hu, Heng-Chao Li, *Senior Member, IEEE*, Lei Pan, Wei Li, *Senior Member, IEEE*, Ran Tao, *Senior Member, IEEE*, and Qian Du, *Fellow, IEEE*

Spatial-Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification

Wen-Shuai Hu, Heng-Chao Li, *Senior Member, IEEE*, Lei Pan, Wei Li, *Senior Member, IEEE*, Ran Tao, *Senior Member, IEEE*, and Qian Du, *Fellow, IEEE*

***Abstract*—In recent years, deep learning has presented a great advance in hyperspectral image (HSI) classification. Particularly, Long Short-Term Memory (LSTM), as a special deep learning structure, has shown great ability in modeling long-term dependencies in the time dimension of video or the spectral dimension of HSIs. However, the loss of spatial information makes it quite difficult to obtain the better performance. In order to address this problem, two novel deep models are proposed to extract more discriminative spatial-spectral features by exploiting the Convolutional LSTM (ConvLSTM). By taking the data patch in a local sliding window as the input of each memory cell band by band, the 2-D extended architecture of LSTM is considered for building the spatial-spectral ConvLSTM 2-D Neural Network (SSCL2DNN) to model long-range dependencies in the spectral domain. To better preserve the intrinsic structure information of the hyperspectral data, the spatial-spectral ConvLSTM 3-D Neural Network (SSCL3DNN) is proposed by extending LSTM to 3-D version for further improving the classification performance. The experiments, conducted on three commonly used HSI data sets, demonstrate that the proposed deep models have certain competitive advantages and can provide better classification performance than other state-of-the-art approaches.**

***Index Terms*—Hyperspectral image, Convolutional Long Short-Term Memory, deep learning, feature extraction, classification.**

I. Introduction

THE hyperspectral remote sensing image is a 3-D data cube, which integrates the spectral information with the 2-D spatial information of land covers. With the development of remote sensing technology, together with the continuous improvement of hyperspectral sensors, hyperspectral images (HSIs) can provide more opportunities to manage and analyze the information from the Earth's surface [1]-[3]. Correspondingly, HSIs have been widely used in many fields, such as environmental sciences [2], precision agriculture [4], [5], ecological science [6], and geological exploration [7].

As a basic and important research topic, HSI classification has attracted plenty of attentions. The support vector machine (SVM) is the most widely used classifier and shows great success in HSI classification [8]. Especially, a composite kernel SVM (SVM-CK) was proposed to exploit the spatial and spectral information simultaneously [9]. Inspired by the successful application on face recognition [10], sparse representation (SR) [11] has also been applied in HSI classification, which was further improved to exploit the spatial information by Chen *et al.* in [12], resulting in a joint SR classifier (JSRC). Subsequently, more and more classifiers based on joint sparse model were developed, such as non-local weighted JSRC (NLW-JSRC) [13], nearest regularized JSRC (NRJSRC) [14], and correntropy-based robust JSRC (RJSRC) [15].

In recent years, deep learning has presented a great advance for feature extraction and classification in the field of computer vision, such as object detection [16], object tracking [17], and behavior recognition in crowd scene [18], and also shown the effectiveness in HSI classification task [19]. An increasing number of feature extraction and classification methods deep learning-based have been designed for HSIs. In [20], a classification method based on Stacked AutoEncoder (SAE) was exploited for HSI classification for the first time by extracting spatial-spectral features. Since Hu *et al.* [21] and Chen *et al.* [22] introduced Convolutional Neural Network (CNN) into HSI classification, many new classification models based on CNN have emerged and provided the satisfying performance in HSI classification. Li *et al.* [23] integrated the pixel-pair model into CNN to further obtain discriminative features. Zhao and Du [24] utilized CNN and the balanced local discriminant embedding to

Manuscript received July 23, 2019; revised October 20, 2019; accepted December 10, 2019. Date of publication January 15, 2020; date of current version May 21, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61871335. (*Corresponding author: Heng-Chao Li.*)

Wen-Shuai. Hu, Heng-Chao Li, and Lei Pan are with the Sichuan Provincial Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 610031 China (e-mail: hcli@home.swjtu.edu.cn).

Wei Li and Ran Tao are with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081 China.

Qian Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA.

fuse the spatial and spectral information. For the sake of joint learning of the spatial-spectral features, many classification models [25]-[28] were built by taking the 3-D data cube as the input and deepening the network to obtain more discriminative features. In order to compute the distance between features of the same classes and that between features of different classes more effectively and quickly, Fang *et al.* [29] integrated the hashing learning into CNN to realize the transformation from high-dimensional features to low-dimensional features. In addition, to overcome the problem of the limitation of available labeled samples, two self-taught learning frameworks based on the multiscale Independent Component Analysis (MICA) and the Stacked Convolutional Autoencoder (SCAE) [30], semisupervised CNN [31], and a supervised deep feature extraction method based on Siamese CNN (S-CNN) [32] were proposed. In addition to the single-channel deep network models, there are also some novel works [27], [33] based on multi-channel CNN to improve feature extraction and classification. In particular, Li *et al.* [34] made a systematic review of HSI classification methods, and not only comprehensively classified the existing models based on deep learning, but also presented a review of the strategies that are used to improve the performance when the labeled samples are limited, which is meaningful for the future research.

Obviously, CNN has provided an extremely effective and basic structure for various tasks, and become the core architecture of various models, such as VGG16 [16], ResNet [25], CapsNets [35], DenseNet [36], among which the convolutional layer is the core backbone for feature extraction. To address the gradient vanishing problem caused by deeper and deeper structures [37], effective feature extraction and classification models [25], [35], [36] were proposed by promoting the CNN filters to improve the classification performance for HSIs.

In order to analyze sequential data, Recurrent Neural Network (RNN) [38], as an effective deep learning model, has been widely concerned for modeling long-range dependencies, and also expected to extract features for HSI classification. By using a special activation function and an improved Gated Recurrent Units (GRU), Mou *et al.* [39] built a RNN model for pixel-level spectral classification. Considering that the spatial information has a positive impact on HSI classification performance, Liu *et al.* [40] used multiply RNNs to model long-term dependencies between central and neighborhood pixels. Zhang *et al.* [41] built a local spatial sequential RNN (LSS-RNN) model, in which the low-level features, such as texture and different morphological profile features, were used to construct LSS features, and then input to RNN to extract high-level features. Based on GRU, Hang *et al.* [42] designed an end-to-end RNN model for extracting the useful information from adjacent and non-adjacent spectral bands, and integrated the convolutional layers into RNN to improve the classification accuracy. In addition, Mou *et al.* [43] used RNN to model the temporal dependency of the outputs of convolutional sub-network for change detection in multispectral imagery.

Long Short-Term Memory (LSTM), as a special RNN structure, has demonstrated its stability and power in modeling long-term dependencies in various studies [44]-[45]. The initially proposed structure of LSTM utilizes the special "memory cells" instead of logistic or tanh hidden units [44], and there are three significant gate mechanisms in this kind of structure: input gate, output gate, and forget gate, which are used for implementing information protection, transmission, and control, respectively. Specifically, the input gate is applied to control when the input is allowed to be added to the memory cell, the output gate is designed to decide when the input data has an influence on the output of the memory cell, and the forget gate is mainly used for modeling long-range dependencies. It is the design of this special memory cell with fixed weight and self-connected circular edges that ensures the gradient can pass across many time steps without gradient vanishing or explosion problems [46]. However, there is an inherent drawback in LSTM, in which the spatial structure information will be lost when unfolding the input to the 1-D from. To better model the spatiotemporal relationships, Shi *et al.* [47] extended this data processing method in LSTM to the convolution operation and proposed Convolutional LSTM (ConvLSTM), which consists of convolution structures in both the input-to-state and state-to-state transitions. Due to the characteristics of LSTM, more attentions have been paid to it, and the most common way is to use LSTM in combination with CNN. Wu *et al.* [48] built a hybrid model for HSI classification, in which the convolutional layers followed by the recurrent layers were used to extract spectrally-contextual features. By utilizing the convolutional recurrent layers, Rußwurm *et al.* [49] built an encoder framework for land cover classification. Song *et al.* [50] proposed a recurrent 3D fully convolutional network for change detection in HSIs, in which the ConvLSTM layer is used to perform long-term dependencies modeling on the outputs of fully convolutional network in the temporal field. Moreover, Seydgar *et al.* [51] proposed a two-stage model, in which ConvLSTM, cascading with 3D CNN, was utilized to extract spatial-spectral features.

In addition to the above combination with CNN, there are some works to build deep models using LSTM alone. Zhou *et al.* [52] designed a spatial-spectral LSTMs (SSLSTMs) model based on two independent LSTMs in the way of spatial LSTM (SaLSTM) and spectral LSTM (SeLSTM). However, this way not only insufficiently fuses the spatial and spectral information, but also loses the spatial structure. Rußwurm *et al.* [53] employed

multi-level cascaded LSTMs for land cover classification. Inspired by spatial similarity measurements [54], Ma *et al.* [55] built a LSTM-based model, in which the spatial-spectral features are extracted by these measurement strategies. Moreover, a Bidirectional-ConvLSTM (Bi-CLSTM) model [56] was proposed by cascading all outputs in each layer for extracting spatial-spectral features. However, this simple cascade fails to fully utilize the correlation between different spectral bands.

Inspired by the above works, the main purpose of this paper is to construct two novel deep ConvLSTM neural networks for HSI feature extraction and classification. The main contributions of this paper are listed as follows:

(1) In order to address the problem of underutilization of the correlations between different spectral bands in Bi-CLSTM and the issues of insufficient fusion of the spatial and spectral information and the loss of the spatial structure information in SSLSTMs, an effective feature extraction model, i.e., spatial-spectral ConvLSTM 2-D Neural Network (SSCL2DNN), is proposed by modeling long-term dependencies in the spectral field for joint learning of spatial-spectral features, in which the local window patch is decomposed into a spectral sequence and then input to each memory cell band by band.

(2) To better preserve the intrinsic structure of hyperspectral data, the 3-D structure (namely ConvLSTM3D) is further developed from the basic ConvLSTM cell, with which a novel deep model, i.e., spatial-spectral ConvLSTM 3-D Neural Network (SSCL3DNN), is constructed. Different from the way of band-by-band processing, the local patch directly takes the form of 3D cube as the input of SSCL3DNN, which enables it to further improve classification performance of SSCL2DNN.

The remainder of this paper is organized as follows. Section II presents CNN, LSTM, and ConvLSTM, and discusses the applications on HSI feature extraction and classification. In Sections III and IV, the proposed deep models with the extraction of spatial-spectral features for HSI classification are described in detail, respectively. Comprehensive quantitative analysis and evaluation of the proposed models are implemented in Section V. And conclusively, Section VI summarizes this paper.

II. Related Work

A. Convolutional Neural Network

The fundamental CNN mainly consists of the following parts: convolutional layer, pooling layer, full connection layer, and classification layer. Based on different convolution operations, we can construct various networks to meet a variety of practical requirements. Originally, the calculation formula of each convolutional layer in CNN can be expressed as

$$o_{r+1} = \phi(W_r * o_r + b_r), \tag{1}$$

where o_{r+1} is the output with M feature maps of the rth convolutional layer, W_r denotes the convolution filter, and b_r is the bias of the rth convolutional layer, o_r is the output of the $(r-1)$th convolutional layer, and $\phi(\cdot)$ is the nonlinear activation function. W_r has a size of $k^{(r)} \times k^{(r)}$ for 2-D CNN. With respect to 3-D CNN, W_r is a convolution filter in the rth convolutional layer with the size of $k^{(r)} \times k^{(r)} \times d^{(r)}$, in which $k^{(r)}$ and $d^{(r)}$ denote the size and depth of the convolution filter, respectively.

Since Hu *et al.* [21] and Chen *et al.* [22] proposed HSI feature extraction and classification models based on CNN, more and more improved CNN-based models have been introduced. CNN has become an extremely effective and basic structure for various tasks. It is noteworthy that a sliding window is still used to extract spatial features in CNN, which is a traditional way of exploiting spatial information. Moreover, the data transmission in CNN only exists between adjacent layers, lacking information interaction inside each layer, which may make it difficult to extract more effective features.

B. Long Short-Term Memory

LSTM is proposed to deal with the issues that RNN is not suitable for learning long-term dependencies and prone to bring about gradient vanishing and exploding problems [46]. It is obvious that the data transmission and processing in LSTM are realized by three key gate units: input gate, output gate, and forget gate, which are used for implementing information protection and control [44]. The calculation formulas between these three gate structures in LSTM are written as

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
\tilde{c}_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ \tanh(c_t),
\end{aligned} \tag{2}$$

where x_t, h_{t-1}, and c_{t-1} represent the input of the current cell, the output and state of the last cell in LSTM, respectively. i_t, f_t, and o_t denote the input gate, forget gate, and output gate of LSTM. $W_{\bullet i}$ and b_i are the weight and bias of the input gate, $W_{\bullet f}$ and b_f are the weight and bias of the forget gate, and $W_{\bullet o}$ and b_o are the weight and bias of the output gate, where $\bullet$ indicates x, h, and c. $\circ$ denotes the Hadamard product. σ is the nonlinear activation function.

Zhou *et al.* [52] first attempted to apply LSTM to HSI classification and advanced the spatial-spectral LSTMs (SSLSTMs). The experimental results show that LSTM can also be well used for modeling long-range dependencies in spectral domain. Nevertheless, it is worth noting that the two branches in SSLSTMs are independent of each other. In addition, the unfolding of the original HSI data to one-dimensional vectors as the input of SSLSTMs will actually lose the intrinsic structure of the hyperspectral data, since spatial information is not considered in LSTM.

C. Convolutional LSTM

Considering the shortcomings of LSTM, a modification and extended version of LSTM, i.e., ConvLSTM [47], is developed. Different from LSTM, the input-to-state and state-to-state transitions in ConvLSTM are realized by convolution, and ConvLSTM holds the same structure as LSTM and can be used to model long-term dependencies in the time domain or spectral domain, in which the calculation formulas can be expressed as

$$\begin{aligned} i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\ \tilde{C}_t &= \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\ H_t &= o_t \circ \tanh(C_t), \end{aligned} \tag{3}$$

where X_t denotes the input of the current cell, C_{t-1} and H_{t-1} are state and output of the last cell, respectively. * means the convolution operation. W denotes the 2-D convolution filter with a $k \times k$ kernel, and k is the size of the convolution kernel, respectively. Specially, the definitions of $W_{\bullet i}$, $W_{\bullet f}$, $W_{\bullet o}$, b_i, b_f, and b_o are similar to that in (2), however, the data dimensions and processing methods are different.

Different from the traditional way of extracting spatial features based on sliding window in CNN, the ConvLSTM cell holds three special gate mechanisms to complete data transmission and processing, which makes it possible to utilize spatial information of HSIs more effectively. Simultaneously, the ConvLSTM layer, constructed by the ConvLSTM cell in (3) as the basic unit, can not only implement data transmission and processing of the inter-layer, but also execute those of the intra-layer, which is another great difference from CNN. This special structure enables ConvLSTM layer to extract the more effective feature representation than CNN. Furthermore, compared with LSTM, the implementations of three gate mechanisms are extended from one-dimensional to multi-dimensional convolution operation, and this change can not only capture the spatial context information of the original data similar to the convolutional layer in CNN, but also model the long-range dependencies in the time domain of video or the spectral domain of HSIs.

III. Spatial-Spectral ConvLSTM 2-D Neural Network (SSCL2DNN)

In SSLSTMs [52], the spatial and spectral features are not effectively fused, and the spatial structure information is not well preserved. As for Bi-CLSTM [56], the correlations between different spectral band is not fully utilized. In order to address the above issues, a novel ConvLSTM-based spatial-spectral feature extraction model is proposed. It should be noted that ConvLSTM given in the Section II-C is actually a 2-D extension structure of LSTM, and in what follows, to distinguish the 3-D extension form in Section VI, ConvLSTM involved in this Section is named ConvLSTM2D.

A. SSCL2DNN

On account of the above analysis, a joint spatial-spectral feature extraction and classification model based on ConvLSTM2D is constructed, which is shown in Fig. 1 labeled by ① with blue. The core structure of our proposed SSCL2DNN model consists of ConvLSTM2D layer and pooling layer. Based on the basic ConvLSTM2D cell in (3), a ConvLSTM2D layer will be built, and a 3-D data cube needs to be decomposed into a spectral sequence, which is then input into each memory cell of the ConvLSTM2D layer one by one. However, there is much redundant information in original HSI data cubes, and the more data used, the greater complexity involved. Unlike [22], PCA is selected as the preprocessing method to implement dimension reduction of data.

First of all, the size of the original HSI data cube can be denoted as $W \times H \times D$, where D indicates the number of the spectral bands, and W and H are the width and height of HSI, respectively. In order to reduce computational complexity, the first K components after PCA are selected as the spectral information of each pixel x_i. Furthermore, for each ConvLSTM2D layer in SSCL2DNN, considering that the spatial context information is beneficial to HSI classification, the data in a local spatial window with the size of $s \times s$ is extracted as the spatial information for extracting spatial-spectral features, which is the input of each memory cell in the ConvLSTM2D layer. After data preprocessing, a 3-D input data denoted by $s \times s \times K$ is constructed. In particular, suppose that the special dimension time_step in ConvLSTM2D is expressed as a variable τ, and it needs to be fixed as K to maintain the same dimension as the input data. Concretely, the 3-D input of each pixel x_i is decomposed into K 2-D components and

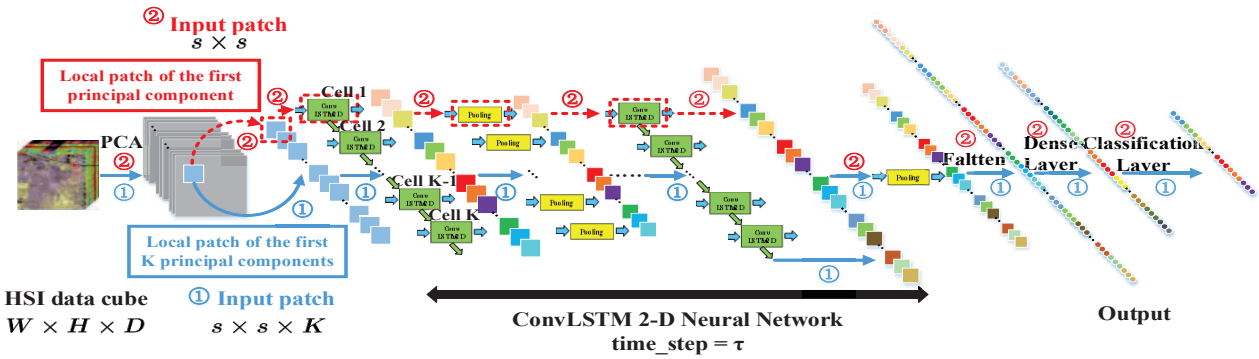


Fig. 1. The proposed spatial-spectral ConvLSTM 2-D Neural Network (SSCL2DNN) is labeled by ① with blue. In particular, SSCL2DNN can be transformed into another deep model when K equals 1 and τ equals 1, which is called spatial ConvLSTM 2-D Neural Network (SaCL2DNN) labeled by ② with red dotted box.

converted into a sequence with the length of K, i.e., $\{X_i^1, \ldots, X_i^k, \ldots, X_i^K\}, k \in \{1, 2, \ldots, K\}$, where X_i^k denotes the kth component of the pixel x_i. Then, this sequence is fed into the ConvLSTM2D layer one by one.

In order to compress the feature maps generated by the ConvLSTM2D layer and reduce the computational complexity, the pooling layer is also used in the proposed deep model. After cascading multiple ConvLSTM2D layers and pooling layers, the final output of the last ConvLSTM2D layer in the proposed SSCL2DNN model is the desired feature representation that is subsequently fed into the classification layer to obtain the final result. Specifically, according to [22], [26], small convolution kernels are efficient for yielding better classification performance. As such, the size of the convolution kernels in our proposed deep model can be set as 4×4 or 3×3, and the kernel size of 2×2 is adopted to implement the operation of the pooling layer.

Due to the exceptional inner structure of LSTM and its extended architecture, it is evident that if we set τ to 1 and K to 1 in each ConvLSTM2D layer, and convert the input from a 3-D data to a 2-D data, the proposed deep model is reduced to spatial ConvLSTM 2-D Neural Network (SaCL2DNN), which is shown in Fig. 1 labeled by ② with the red dotted boxes. Specially, this transform is similar to that from 3D CNN to 2D CNN, however, SaCL2DNN can utilize the spatial structure information of hyperspectral data more effectively to obtain better classification performance than 2-D CNN.

Particularly, the proposed deep models can not only implement the same data transmission between layers as CNN does, but also accomplish effective data transmission and processing within each ConvLSTM2D layer. It is known that adjacent spectral bands are highly correlated, and there may also be some correlations between non-adjacent spectral bands [48]. Therefore, by effectively modeling long-range dependencies in the spectral field, SSCL2DNN can jointly consider the spatial-spectral information, which can address the problems of insufficient feature fusion caused by using two sub-branches independently and the loss of spatial information in SSLSTMs [52] and underutilization of correlation between different spectral bands in Bi-CLSTM [56].

B. Loss Function and Optimization Method

Suppose that there are L ConvLSTM2D layers and pooling layers in SSCL2DNN. In the first ConvLSTM2D layer, each 2-D component X_i^k in the spectral sequence $\{X_i^1, \ldots, X_i^k, \ldots, X_i^K\}$ is the input of the kth memory cell, which is actually X_t in (3), and then there will be two kinds of outputs, i.e., the outputs $\{H_{1i}^1, \ldots, H_{1i}^k, \ldots, H_{1i}^K\}$ of K memory cells and the state and output $\{C_{1i}, H_{1i}\}$ yielded by modeling long-term dependencies, which are actually $\{C_t, H_t\}$ in (3). The outputs of K memory cells are retained as the inputs of the next layer to extract high-level features. After L ConvLSTM2D layers and pooling layers, the output H_{Li} of the last ConvLSTM2D layer generated by modeling long-range dependencies in the spectral field is the desired spatial-spectral features. After reducing dimensions through the last pooling layer, the spatial-spectral features H_{Li} are converted into the 1-D vectors h_{Li} and then fed into a fully connected layer to map the feature space to class label space. Finally, the feature vectors h_{Li} are input to a softmax function to predict the conditional probability distribution $P(y = c|h_{Li}, W, b) = \frac{e^{(W_c h_{Li} + b_c)}}{\sum_{j=1}^{N} e^{(W_j h_{Li} + b_j)}}$ of each class c, where $c \in 1, 2, \ldots, N$, and N is the number of classes in the HSI data sets.

In addition, the cross entropy [52] is used as the loss function to obtain the final classification results, which is optimized by adaptive momentum (ADAM) algorithm.

IV. Spatial-Spectral ConvLSTM 3-D Neural Network (SSCL3DNN)

In order to better preserve the intrinsic structure information of the hyperspectral data, in this Section, the 3-D extended structure (ConvLSTM3D) is further developed,

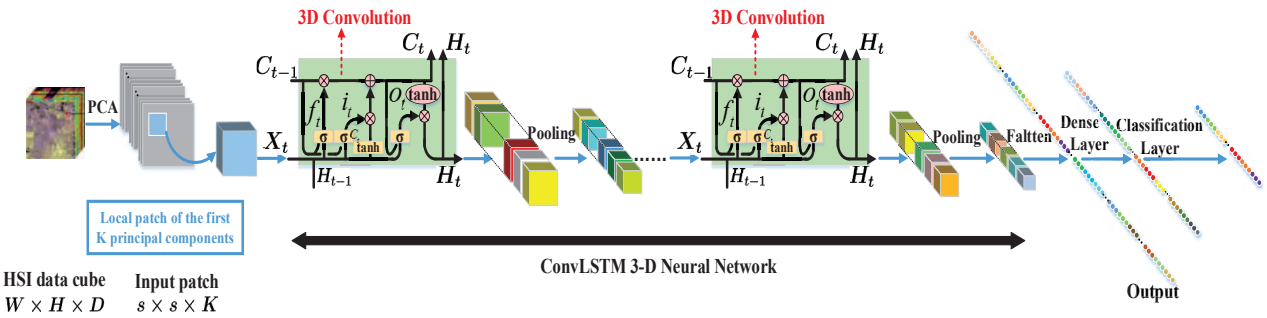


Fig. 2. The proposed spatial-spectral ConvLSTM 3-D Neural Network (SSCL3DNN) structure.

and another novel deep feature extraction and classification model, i.e., SSCL3DNN, is proposed, which can yield more discriminative spatial-spectral features for further improving the classification performance.

A. ConvLSTM3D

On the basis of ConvLSTM2D, the 3-D extended version (ConvLSTM3D) is further developed, in which there are three gate units, and the calculation formulas are similar to that in ConvLSTM2D. However, different from it, the whole 3-D data cube is taken as the input of each memory cell in ConvLSTM3D. In particular, the input $\boldsymbol{\mathcal{X}_t}$, the state $\boldsymbol{\mathcal{C}_{t-1}}$ and $\boldsymbol{\mathcal{C}_t}$, the output $\boldsymbol{\mathcal{H}_{t-1}}$ and $\boldsymbol{\mathcal{H}_t}$, and the gate units i_t, f_t, and o_t of ConvLSTM3D are 4-D tensors, whose last three dimensions are spectral dimension and two spatial dimensions, and the convolution filters $\boldsymbol{\mathcal{W}_{\bullet i}}$, $\boldsymbol{\mathcal{W}_{\bullet f}}$, and $\boldsymbol{\mathcal{W}_{\bullet o}}$ are 3-D tensors. The structure illustration of the ConvLSTM3D layer is shown in Fig. 3. Specifically, the equations of the ConvLSTM3D cell can be written as

$$\begin{aligned}
i_t &= \sigma(\boldsymbol{\mathcal{W}_{xi}} \circledast \boldsymbol{\mathcal{X}_t} + \boldsymbol{\mathcal{W}_{hi}} \circledast \boldsymbol{\mathcal{H}_{t-1}} + \boldsymbol{\mathcal{W}_{ci}} \circ \boldsymbol{\mathcal{C}_{t-1}} + b_i) \\
f_t &= \sigma(\boldsymbol{\mathcal{W}_{xf}} \circledast \boldsymbol{\mathcal{X}_t} + \boldsymbol{\mathcal{W}_{hf}} \circledast \boldsymbol{\mathcal{H}_{t-1}} + \boldsymbol{\mathcal{W}_{cf}} \circ \boldsymbol{\mathcal{C}_{t-1}} + b_f) \\
\boldsymbol{\tilde{\mathcal{C}}_t} &= \tanh(\boldsymbol{\mathcal{W}_{xc}} \circledast \boldsymbol{\mathcal{X}_t} + \boldsymbol{\mathcal{W}_{hc}} \circledast \boldsymbol{\mathcal{H}_{t-1}} + b_c) \\
\boldsymbol{\mathcal{C}_t} &= f_t \circ \boldsymbol{\mathcal{C}_{t-1}} + i_t \circ \boldsymbol{\tilde{\mathcal{C}}_t} \\
o_t &= \sigma(\boldsymbol{\mathcal{W}_{xo}} \circledast \boldsymbol{\mathcal{X}_t} + \boldsymbol{\mathcal{W}_{ho}} \circledast \boldsymbol{\mathcal{H}_{t-1}} + \boldsymbol{\mathcal{W}_{co}} \circ \boldsymbol{\mathcal{C}_t} + b_o) \\
\boldsymbol{\mathcal{H}_t} &= o_t \circ \tanh(\boldsymbol{\mathcal{C}_t}).
\end{aligned} \tag{4}$$

where $\circledast$ is the defined 3-D convolution between 4-D input or output and 3-D convolution filter.

Compared with the LSTM cell in (2) and the ConvLSTM2D cell in (3), the extended ConvLSTM3D cell has a similar calculational model, however, the main difference is the calculation of the convolution in each gate unit. Taking the input gate of the ConvLSTM3D cell as an example. Suppose that $\boldsymbol{\mathcal{X}_t} \in R^{\tau_t \times w_t \times h_t \times s_t}$ is the input of it, which is the tth component in a sequence decomposed from the input of the ConvLSTM3D layer according to the dimension time_step, and $\boldsymbol{\mathcal{W}_{xi}} \in R^{k_1 \times k_2 \times d}$, where τ_t, w_t, h_t, s_t, k_1, k_2,

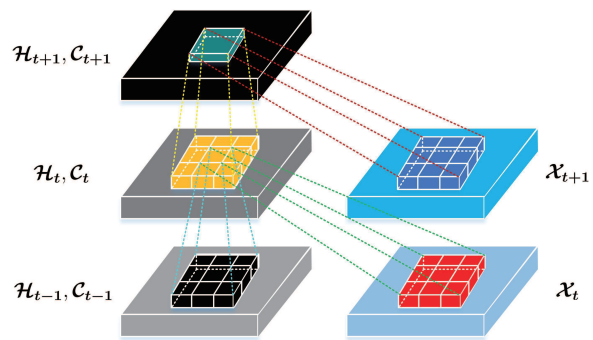


Fig. 3. The illustration of the inner structure of ConvLSTM3D.

and d are the dimension time_step, width, height, the number of the spectral bands, kernel size and depth, respectively. We define the 3-D convolution of $\boldsymbol{\mathcal{X}_t}$ and $\boldsymbol{\mathcal{W}_{xi}}$ as $\boldsymbol{\mathcal{W}_{xi}} \circledast \boldsymbol{\mathcal{X}_t}$, where the output $u^{\tau_t l_x l_y l_z}$ yielded by $\boldsymbol{\mathcal{X}_t}$ at position (l_x, l_y, l_z) in the input gate is defined as

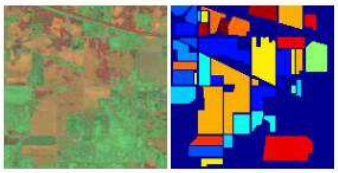
$$u^{\tau_t l_x l_y l_z} = \sum_{j=1}^{k_1} \sum_{p=1}^{k_2} \sum_{q=1}^{d} \boldsymbol{\mathcal{W}_{xi}}^{jpq} \boldsymbol{\mathcal{X}_t}^{\tau_t (l_x+j)(l_y+p)(l_z+q)}. \tag{5}$$

B. SSCL3DNN

According to the extraordinary structure of ConvLSTM3D, we further extend SSCL2DNN to a new deep model, i.e., SSCL3DNN, based on ConvLSTM3D. Different from SSCL2DNN, the whole data cube must be taken as the input of the memory cell in SSCL3DNN. Therefore, the intrinsic structure of HSIs can be captured well, and the spatial-spectral features can be well learned to further improve the HSI classification performance. Fig. 2 clearly shows the structure of the proposed deep model, and the parameter τ is fixed as 1.

In particular, PCA is also used to reduce the redundant information in this Subsection, and similarly, the first K components are chosen. With regard to the selection of the size of the convolution kernel, the small kernel size

Indian Pines

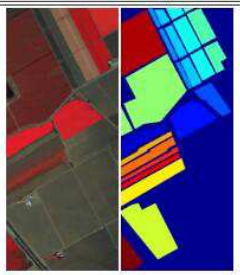


(a)

NO.	Color	Class	Training	Total
1		Alfalfa	5	46
2		Corn-notill	143	1428
3		Corn-mintill	83	830
4		Corn	24	237
5		Grass-pasture	48	483
6		Grass-trees	73	730
7		Grass-pasture-mowed	3	28
8		Hay-windrowed	48	478
9		Oats	2	20
10		Soybean-notill	97	972
11		Soybean-mintill	246	2455
12		Soybean-clean	59	593
13		Wheat	21	205
14		Woods	127	1265
15		Buildings-Grass-Trees-Drives	39	386
16		Stone-Steel-Towers	9	93
Total			1027	10249

(d)

Salinas Valley



(b)

NO.	Color	Class	Training	Total
1		Brocoli-green-weeds-1	20	2009
2		Brocoli-green-weeds-2	37	3726
3		Fallow	20	1976
4		Fallow-rough-plow	14	1394
5		Fallow-smooth	27	2678
6		Stubble	40	3959
7		Celery	36	3579
8		Grapes-untrained	113	11271
9		Soil-vinyard-develop	62	6203
10		Corn-senesced-green-weeds	33	3278
11		Lettuce-romaine-4wk	11	1068
12		Lettuce-romaine-5wk	19	1927
13		Lettuce-romaine-6wk	9	916
14		Lettuce-romaine-7wk	11	1070
15		Vinyard-untrained	73	7268
16		Vinyard-vertical-trellis	18	1807
Total			543	54129

(e)

University of Pavia

(c)

NO.	Color	Class	Training	Total
1		Asphalt	66	6631
2		Meadows	186	18649
3		Gravel	21	2099
4		Trees	31	3064
5		Painted metal sheets	13	1345
6		Bare Soil	50	5029
7		Bitumen	13	1330
8		Self-Blocking Bricks	37	3682
9		Shadows	9	947
Total			426	42776

(f)

Fig. 4. (Left) False color maps and (Right) ground-truth maps of three HSI data sets. (a) Indian Pines (bands 20, 40, and 60). (b) Salinas Valley (bands 46, 27, and 10). (c) University of Pavia (bands 47, 27, and 13). (d)-(f) show the number of training samples.

is also considered, which can be set as $3 \times 3 \times 3$ or $5 \times 5 \times 5$ in the proposed deep model. Besides, the size of the pooling kernel in the pooling layer is set as $2 \times 2 \times 2$. After the cascade of multiple ConvLSTM3D layers and pooling layers, the spatial-spectral features are extracted from the 3-D data cube and subsequently fed into the classification layer to obtain the classification results.

Although the loss function and optimization method in this Section are the same as those of SSCL2DNN in Section III, the spatial-spectral features H_{Li} and its 1-D vector form h_{Li} of each pixel x_i and data processing at each layer in SSCL3DNN are quite different.

V. Experimental Results

To quantitatively and qualitatively analyze the classification performance of the proposed models, SVM [57], 2-D CNN [22], 3-D CNN [22], SaLSTM [52], SSLSTMs [52], and Bi-CLSTM [56] are used as the comparative algorithms. In particular, the special case (SaCL2DNN) of SSCL2DNN is used to compare fairly with SVM, 2-D CNN, and SaLSTM. Three commonly used quantitative metrics are adopted, i.e., overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ). To eliminate the bias introduced by randomly choosing training samples, each experiment is repeated 10 times, and the mean values of each evaluation criterion are presented.

A. Hyperspectral Data Sets

Three common HSI data sets, i.e., Indian Pines, Salinas Valley, and University of Pavia, are considered in our experiments, whose false color maps, groundtruth maps, and the corresponding training size are shown in Fig. 4, respectively.

1) Indian Pines: The Indian Pines data set was captured in 1992 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in Northwestern Indiana, USA, which is mainly composed of multiple agricultural fields. The spatial size of this data set is 145×145 pixels with a spatial resolution of 20 meters per pixel (mpp), and there are 224 spectral bands in the wavelength range from 0.4 to 2.5 μm. Since some of them cannot be reflected by water together with four null bands, there are generally 200 bands remaining for study. After removing the background pixels, 10249 pixels are reserved, which contain useful ground-truth information from 16 different class labels.

2) Salinas Valley: The Salinas Valley data set was collected by the 224-band AVIRIS sensor over Salinas Valley, California. This data set is made up of 512 lines and 217 columns, and contains 16 ground-truth classes. After removing the water absorption bands and noise-affected bands, there are 204 spectral bands preserved.

3) University of Pavia: The University of Pavia data set was acquired by the Reflective Optics System Imag-

ing Spectrometer (ROSIS) sensor over University of Pavia, Northern Italy. There are 103 spectral bands in the spectral range from 0.43 to 0.86 μm by removing several noise-corrupted bands, and it presents a size of 610×340 pixels with a spatial resolution of 1.3 mpp. Different from the Indian Pines and Salinas Valley data sets, this data set contains 9 distinguishable classes.

B. Experimental Settings

In order to reduce the computational complexity, PCA is used as a preprocessing method to reduce data dimensions. For SVM, 2-D CNN, SaLSTM, and SaCL2DNN, the first principal component is preserved as the input. As far as SSCL2DNN and SSCL3DNN are concerned, the top K components are extracted as the spectral features.

TABLE I
SENSITIVITY COMPARISON AND ANALYSIS UNDER DIFFERENT SIZE ($s \times s$) OF LOCAL WINDOWS.

$s \times s$	Indian Pines		Salinas Valley		University of Pavia	
	SSCL2DNN	SSCL3DNN	SSCL2DNN	SSCL3DNN	SSCL2DNN	SSCL3DNN
21×21	96.63	97.90	92.24	98.78	86.97	87.74
23×23	97.09	98.21	92.45	98.95	87.14	91.29
25×25	97.37	98.32	94.03	98.97	88.42	95.58
27×27	**97.72**	**98.79**	**96.30**	99.29	**89.04**	97.10
29×29	97.70	98.47	95.58	99.59	87.95	97.32
31×31	97.68	98.20	95.69	**99.75**	85.87	**97.42**
33×33	97.10	98.07	95.52	99.40	86.92	95.99

TABLE II
SENSITIVITY COMPARISON AND ANALYSIS UNDER DIFFERENT NUMBER (K) OF PRINCIPAL COMPONENTS AFTER PCA.

K	Indian Pines		Salinas Valley		University of Pavia	
	SSCL2DNN	SSCL3DNN	SSCL2DNN	SSCL3DNN	SSCL2DNN	SSCL3DNN
5	**98.03**	98.74	95.64	99.07	**91.47**	93.43
10	97.72	**98.79**	**96.30**	**99.29**	89.04	**97.10**
15	96.90	98.72	95.24	99.18	85.56	96.74
20	96.63	98.51			83.46	96.53

All the parameters of the compared methods are confirmed according to [22], [52], [56], [57] to achieve the quasi-optimal performance. Specifically, for SVM, the radial basis function (RBF) is used in the Libsvm toolbox [57]. There are two key parameters in SVM, i.e., C and γ, which denote the regularization parameter and the kernel function parameter, respectively. According to [57], fivefold cross-validation is adopted to tune C and γ from the range of $\left\{2^{-5}, 2^{-4}, \ldots, 2^{19}\right\}$ and $\left\{2^{-15}, 2^{-14}, \ldots, 2^{4}\right\}$, respectively. For Bi-CLSTM [56], the kernel size and the number of feature maps in each ConvLSTM2D layer are fixed as 3 and 32, respectively. As for SeLSTM, SaLSTM, and SSLSTMs [52], the numbers of output nodes in SeLSTM and SaLSTM are set as 64 and 128, respectively, for the Indian Pines and Salinas Valley data sets, while as 128 and 256, respectively, for the University of Pavia data set.

TABLE III
THE PARAMETER SETTINGS OF SACL2DNN.

Layer Name	Output Shape	Filter Size	Padding
Input Layer	27×27	N/A	N/A
ConvLSTM2D Layer	$27 \times 27 \times 32$	$3 \times 3 \times 32$	SAME
MaxPooling Layer	$14 \times 14 \times 32$	$2 \times 2 \times 32$	SAME
ConvLSTM2D Layer	$14 \times 14 \times 64$	$5 \times 5 \times 64$	SAME
MaxPooling Layer	$7 \times 7 \times 64$	$2 \times 2 \times 64$	SAME
Droup	$7 \times 7 \times 64$	0.25	N/A
Flatten	3136	N/A	N/A
Dense Layer	128	N/A	N/A
Output	N	N	N/A

TABLE IV
THE PARAMETER SETTINGS OF THE PROPOSED SSCL2DNN.

Layer Name	Output Shape	Filter Size	Padding
Input Layer	$10 \times 27 \times 27$	N/A	N/A
ConvLSTM2D Layer	$10 \times 27 \times 27 \times 32$	$4 \times 4 \times 32$	SAME
MaxPooling Layer	$10 \times 14 \times 14 \times 32$	$2 \times 2 \times 32$	SAME
ConvLSTM2D Layer	$1 \times 14 \times 14 \times 64$	$3 \times 3 \times 64$	SAME
MaxPooling Layer	$1 \times 7 \times 7 \times 64$	$2 \times 2 \times 64$	SAME
Droup	$1 \times 7 \times 7 \times 64$	0.25	N/A
Flatten	3136	N/A	N/A
Dense Layer	128	N/A	N/A
Output	N	N	N/A

TABLE V
THE PARAMETER SETTINGS OF THE PROPOSED SSCL3DNN.

Layer Name	Output Shape	Filter Size	Padding
Input Layer	$10 \times 27 \times 27$	N/A	N/A
ConvLSTM3D Layer	$10 \times 27 \times 27 \times 32$	$4 \times 4 \times 4 \times 32$	SAME
MaxPooling Layer	$5 \times 14 \times 14 \times 32$	$2 \times 2 \times 2 \times 32$	SAME
ConvLSTM3D Layer	$5 \times 14 \times 14 \times 64$	$3 \times 3 \times 3 \times 64$	SAME
MaxPooling Layer	$3 \times 7 \times 7 \times 64$	$2 \times 2 \times 2 \times 64$	SAME
Droup	$3 \times 7 \times 7 \times 64$	0.25	N/A
Flatten	9408	N/A	N/A
Dense Layer	128	N/A	N/A
Droup	128	0.5	N/A
Output	N	N	N/A

For SSCL2DNN and SSCL3DNN, there are four key parameters to be determined, i.e., the size ($s \times s$) of the local window, the number (K) of the principal components after PCA, the size ($k \times k$) of the convolution operation, and the number (M) of the feature maps at each ConvLSTM layer. Firstly, K is fixed as 10. Then, s is generated from $\{21, 23, 25, 27, 29, 31, 33\}$, k is from $\{3, 4, 5\}$, and M is searched from three given combinations $\{16, 32\}$, $\{32, 64\}$, and $\{64, 128\}$. The experimental results under different size of local window are reported in Table I, from which it can be seen that for these three data sets, the optimal s in SSCL2DNN is 27, while for SSCL3DNN, it is 27, 31, and 31, respectively. Although SSCL3DNN can obtain 0.46% and 0.32% gains when s varies from 27 to 31 for the Salinas Valley and University of Pavia data sets, the computational complexity and runtime will obviously increase. Based on the above analysis, the size of local window for the proposed models is fixed as 27×27, which can not only achieve satisfactory classification performance, but also provide convenience for the practical application of the proposed models. In particular, the size of local window in SaCL2DNN is the same as that in SSCL2DNN.

TABLE VI
CLASSIFICATION RESULTS FOR THE INDIAN PINES DATA SET USING 10% TRAINING SAMPLES.

Class	SVM	2-D CNN	SaLSTM	SaCL2DNN	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN
1	70.73	**90.24**	85.98	86.59	99.39	79.88	91.06	**100.00**	98.17
2	89.32	95.47	92.43	**96.73**	97.86	94.24	94.29	98.11	**99.12**
3	91.93	**95.21**	87.25	94.31	**97.05**	90.70	93.13	96.56	96.52
4	86.50	92.84	88.97	**93.54**	96.95	88.85	88.89	96.56	**98.71**
5	90.57	**94.02**	91.49	91.55	96.55	91.09	94.25	96.09	**97.07**
6	97.72	97.45	95.81	**98.36**	99.20	96.58	**99.49**	98.02	99.09
7	58.00	**75.00**	67.00	74.00	94.00	77.00	93.33	84.00	**96.00**
8	98.72	99.07	96.16	**99.71**	**100.00**	97.09	99.46	99.69	**100.00**
9	44.44	61.11	59.72	**63.89**	**76.39**	63.89	38.89	55.56	63.89
10	77.20	94.86	89.37	**95.54**	97.31	91.49	95.73	97.07	**98.74**
11	95.02	98.03	95.30	**99.33**	98.95	95.21	96.60	**99.34**	99.33
12	84.27	91.06	85.53	**95.27**	97.94	86.42	89.37	96.75	**98.78**
13	89.13	93.75	86.68	**97.96**	97.01	91.98	95.65	98.19	**99.73**
14	97.41	98.79	97.72	**99.34**	99.56	98.51	**99.80**	99.65	99.69
15	93.73	97.33	95.24	**98.70**	99.06	94.42	97.12	98.85	**99.42**
16	68.75	**93.15**	70.83	88.39	**94.35**	79.46	89.29	85.32	92.56
OA	91.20	96.21	92.59	**97.07**	98.28	93.69	95.62	98.03	**98.79**
AA	83.34	91.71	86.59	**92.08**	**96.35**	88.56	90.90	93.73	96.05
κ	89.91	95.67	91.53	**96.66**	98.04	92.79	94.99	97.75	**98.62**

TABLE VII
CLASSIFICATION RESULTS FOR THE SALINAS VALLEY DATA SET USING 1% TRAINING SAMPLES.

Class	SVM	2-D CNN	SaLSTM	SaCL2DNN	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN
1	74.26	81.35	89.66	**96.68**	98.59	82.02	93.08	91.64	**99.73**
2	85.23	86.21	76.85	**97.40**	99.96	82.15	99.57	97.80	**100.00**
3	59.56	78.14	51.94	**81.61**	96.22	54.62	98.99	98.77	**100.00**
4	98.19	97.66	95.97	**99.13**	99.81	95.53	98.84	99.40	**100.00**
5	94.33	93.45	93.66	**97.62**	**99.66**	93.05	99.41	98.83	99.14
6	94.40	94.61	94.01	**99.63**	99.98	95.74	99.76	**100.00**	**100.00**
7	78.97	88.90	80.20	**95.48**	98.74	82.30	98.15	96.23	**100.00**
8	85.36	93.66	82.68	**93.52**	89.17	85.36	90.05	93.05	**97.36**
9	68.44	84.14	80.87	**95.91**	99.84	77.75	99.69	99.89	**99.97**
10	71.65	87.81	70.49	**95.78**	99.04	70.77	99.62	98.44	**99.76**
11	93.98	94.54	97.67	**96.53**	97.45	95.90	99.12	98.08	**99.68**
12	88.38	87.47	84.22	**95.25**	99.09	86.60	97.43	99.06	**100.00**
13	80.04	82.21	72.14	**86.99**	97.46	68.72	99.23	97.02	**99.45**
14	84.04	88.10	92.73	**93.08**	99.84	91.34	**99.87**	99.69	99.75
15	90.36	94.59	86.30	**98.11**	**99.49**	88.93	89.46	92.78	99.48
16	64.92	**78.96**	64.82	71.16	99.85	61.13	97.34	93.39	**100.00**
OA	82.39	89.65	82.28	**94.68**	97.17	83.10	95.72	96.30	**99.29**
AA	82.01	88.24	82.14	**93.37**	98.39	81.99	97.48	97.13	**99.65**
κ	80.30	88.47	80.26	**94.06**	96.85	81.15	95.24	95.88	**99.21**

TABLE VIII
CLASSIFICATION RESULTS FOR THE UNIVERSITY OF PAVIA DATA SET USING 1% TRAINING SAMPLES.

Class	SVM	2-D CNN	SaLSTM	SaCL2DNN	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN
1	52.21	79.64	57.22	**80.20**	82.91	70.57	88.24	96.15	**97.62**
2	89.78	93.79	81.84	**98.54**	96.25	89.90	98.18	99.43	**99.95**
3	17.73	44.26	27.74	**45.76**	66.25	32.20	37.91	64.28	**82.96**
4	55.23	77.30	57.98	**81.89**	90.61	69.21	80.39	90.45	**96.52**
5	68.05	77.76	70.80	**79.55**	93.83	77.69	80.56	89.01	**98.51**
6	39.14	70.75	42.07	**85.55**	85.72	50.47	66.77	76.60	**97.75**
7	15.19	**38.22**	27.46	36.05	77.80	35.20	56.09	67.15	**88.61**
8	60.56	84.19	62.86	**87.23**	88.48	68.19	82.22	92.58	**95.41**
9	53.39	79.13	57.23	**90.06**	68.64	85.37	**86.67**	78.04	83.60
OA	65.67	81.89	64.77	**86.69**	89.14	73.90	85.22	91.47	**97.10**
AA	50.14	71.67	53.91	**76.09**	83.39	64.31	75.22	83.74	**93.44**
κ	51.52	75.86	52.90	**81.94**	85.59	64.42	79.85	88.49	**96.14**

After that, the experiments for analyzing the influence of different K on the performance are further completed, and K is searched from $\{5, 10, 15, 20\}$ for the Indian Pines and University of Pavia data sets, while from $\{5, 10, 15\}$ for the Salinas Valley data set because of the memory problems. The results in Table II reveal that for these three data sets, the optimal K in SSCL3DNN is 10, while for SSCL2DNN, it is 5, 10, and 5, respectively.

In order to ensure the convergence of the loss function, the number of training epochs is fixed as 2000, and the learning rate is set as 0.0001 from epochs 1 to 2000 for the proposed deep models. More detailed parameter settings are summarized in Tables III-V.

For hardware system configuration, all the following experiments are completed on a desktop with an 8th Generation Intel Core i7-8700 processor and up to 3.7 GHz, 16 GB of DDR4 RAM with a serial speed of 2400 MHz, a Nvidia GeForce GTX 1080ti GPU with 11 GB memory, an Inter SSD D3-S4510 with 240GB. For software system configuration, we adopt Ubuntu 16.04.4 x64 as our operating system for all experiments. CUDA 8.0 and cuDNN 7.0.5, Tflearn with 0.3.2, Tensorflow-gpu

TABLE IX
CLASSIFICATION RESULTS OF TRAINING UNDER SMALL SAMPLES (10 SAMPLES FOR EACH CLASS).

Class	Indian Pines				Salinas Valley				University of Pavia			
	SVM	2-D CNN	SaLSTM	SaCL2DNN	SVM	2-D CNN	SaLSTM	SaCL2DNN	SVM	2-D CNN	SaLSTM	SaCL2DNN
1	91.67	94.44	88.33	**95.00**	73.31	76.16	70.96	**83.71**	40.07	37.14	32.95	**44.89**
2	**42.99**	37.83	31.20	38.98	58.15	61.93	47.37	**73.62**	42.49	43.51	30.42	**67.08**
3	**56.51**	47.59	41.90	52.34	53.38	46.20	36.31	**67.70**	30.80	**34.96**	21.97	34.67
4	73.83	76.12	65.37	**84.93**	95.94	97.34	93.05	**99.21**	54.21	59.74	31.83	**66.72**
5	56.87	58.01	57.67	**63.00**	87.47	88.78	82.36	**92.50**	**83.97**	75.18	69.46	80.02
6	**60.42**	46.06	43.86	51.44	**87.51**	85.21	81.16	82.60	33.57	23.30	18.84	**45.57**
7	97.78	**100.00**	94.44	**100.00**	63.57	68.06	60.57	**82.64**	**66.39**	35.86	33.23	39.77
8	**96.28**	93.33	70.56	87.78	37.40	**38.37**	40.94	32.38	43.75	63.47	30.15	**79.58**
9	**100.00**	**100.00**	84.00	**100.00**	43.82	34.70	40.19	**45.79**	57.42	86.27	57.45	**95.55**
10	52.77	47.80	42.64	**56.30**	53.64	53.20	47.65	**83.46**				
11	**42.94**	42.45	28.09	42.81	94.67	93.01	89.21	**96.07**				
12	49.67	45.97	38.42	**52.08**	89.17	80.86	69.13	**90.28**				
13	81.13	86.97	75.18	**95.90**	90.53	93.09	85.76	**94.88**				
14	70.66	69.94	73.34	**82.10**	90.60	87.72	88.72	**97.96**				
15	71.33	69.95	67.98	**78.67**	**81.05**	78.31	77.32	72.90				
16	90.84	90.60	77.11	**92.77**	73.32	61.37	51.50	**82.53**				
OA	56.41	53.08	45.77	**57.47**	63.77	62.15	59.15	**67.49**	43.81	44.30	31.02	**60.76**
AA	70.98	69.19	61.26	**73.38**	73.28	71.52	66.39	**79.89**	50.30	51.05	36.26	**61.54**
κ	51.34	47.72	40.40	**52.75**	60.24	58.56	55.22	**64.47**	32.36	32.59	19.04	**50.60**

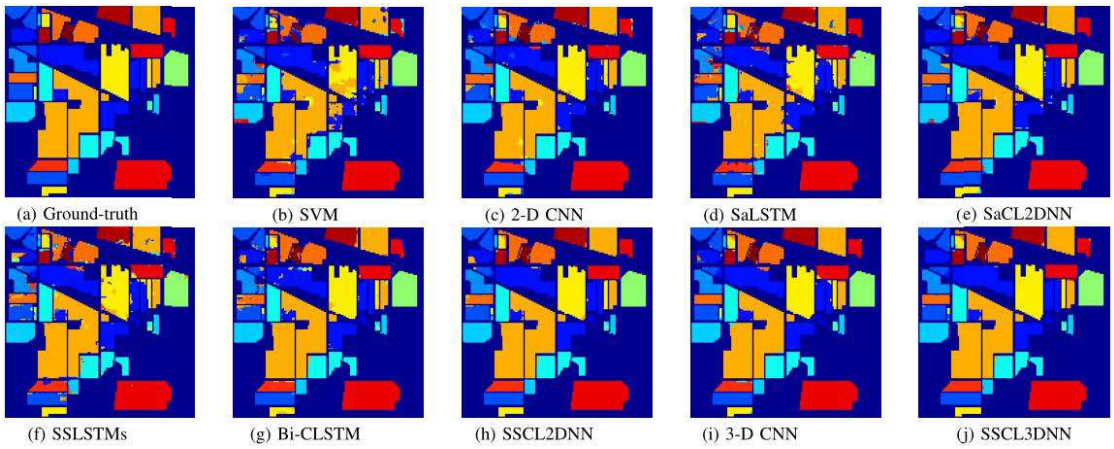


Fig. 5. Classification maps for the Indian Pines data set. (a) Ground-truth map. (b) SVM. (c) 2-D CNN. (d) SaLSTM. (e) SaCL2DNN. (f) SSLSTMs. (g) Bi-CLSTM. (h) SSCL2DNN. (i) 3-D CNN. (j) SSCL3DNN.

with 1.4.0 and python 3.5.4 are the main programming environment. Specially, all methods involved in our experiments are completed in Anaconda 3.4.

It should be noted that Tflearn is a modular and transparent deep learning library built on Tensorflow, and can provide a higher level API than Tensorflow. A combination of Tflearn and Tensorflow constructs the core framework of the proposed deep models.

C. Classification Performance

In order to show the superiority of the proposed deep models, we randomly select 1% of the available labeled samples for training for both the Salinas Valley and University of Pavia data sets while 10% for the Indian Pines data set, respectively. And the remaining samples are used for testing.

According to the experiment settings in Subsection B, the quantitative assessments based on all models are reported in Tables VI-VIII, from which it can be seen that the proposed SSCL2DNN and SSCL3DNN models can provide better classification performance than other considered models. First of all, three special gate mechanisms make it possible for ConvLSTM to take full use of both the spatial and spectral information of HSIs than CNN that is operated by the traditional method based on the sliding window. What's more, the implementations of the gate mechanisms in ConvLSTM are extended from one-dimensional to multi-dimensional convolution operation, which enables ConvLSTM to better preserve and capture spatial context information of HSIs, and fuse the spatial and spectral information more effectively than LSTM. Specifically, on the one hand, compared with 2-D CNN, SaCL2DNN improves OA by 0.86%, 5.03%, and 4.80%, respectively, for the Indian Pines, Salinas Valley, and University of Pavia data sets, which can better capture spatial context information from hyperspectral data. It should be noted that it is the loss of the spatial information that leads to unsatisfactory performance for

TABLE X
CLASSIFICATION RESULTS OF TRAINING UNDER SMALL SAMPLES (10 SAMPLES FOR EACH CLASS).

Class	Indian Pines					Salinas Valley					University of Pavia				
	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN	3-D CNN	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSCL3DNN
1	97.22	86.11	98.15	98.15	**100.00**	**99.84**	74.81	74.95	79.13	99.69	42.58	39.94	51.32	49.43	**69.89**
2	**53.60**	35.35	27.41	38.65	53.30	89.36	50.32	83.88	70.94	**99.90**	**84.89**	29.08	82.59	76.89	82.44
3	**61.78**	38.44	39.02	46.34	50.37	92.56	39.67	98.07	98.80	**99.93**	25.50	25.10	33.89	39.38	**65.10**
4	**96.83**	62.03	81.20	79.88	88.63	99.36	94.86	99.33	98.22	**99.83**	34.01	60.82	52.32	62.48	**90.48**
5	75.90	61.27	66.67	61.95	**76.58**	96.25	83.22	**98.50**	94.80	98.01	91.41	73.71	86.52	84.42	**94.83**
6	78.47	40.39	74.77	65.56	**90.17**	99.22	90.47	95.71	94.02	**99.67**	37.47	29.14	45.37	41.29	**82.37**
7	**100.00**	97.78	**100.00**	**100.00**	**100.00**	95.39	66.74	90.59	87.12	**99.74**	80.61	40.66	70.96	61.16	**89.32**
8	99.83	71.67	97.79	94.59	**100.00**	**87.12**	44.20	64.87	65.67	85.00	10.74	34.70	57.75	66.68	**84.71**
9	**100.00**	92.00	**100.00**	**100.00**	**100.00**	96.33	40.61	99.25	97.44	**99.73**	21.10	85.91	84.95	83.46	**86.91**
10	63.56	44.86	53.40	66.22	**75.36**	95.18	47.55	94.72	91.98	**99.04**					
11	55.09	32.74	**61.83**	50.46	59.66	97.05	89.62	98.20	98.03	**99.41**					
12	**67.03**	42.98	58.43	60.83	64.97	97.27	71.95	97.76	97.00	**99.69**					
13	96.82	82.87	93.16	96.58	**99.79**	98.87	85.52	**99.67**	98.92	99.65					
14	88.11	71.81	83.16	73.86	**88.29**	98.49	87.13	99.09	97.00	**99.75**					
15	84.20	66.60	82.98	84.13	**87.66**	63.49	77.58	64.95	71.52	**86.22**					
16	98.31	86.99	90.36	91.16	**100.00**	95.72	51.60	88.17	79.59	**99.98**					
OA	69.21	47.57	61.91	60.59	**71.28**	89.77	61.63	83.88	82.92	**94.73**	58.50	36.33	66.49	64.59	**81.11**
AA	82.30	63.37	75.52	75.52	**83.42**	93.84	68.49	90.48	88.76	**97.83**	47.59	46.56	62.85	62.80	**82.90**
κ	65.56	42.16	57.33	56.12	**67.83**	88.57	57.92	82.12	81.05	**94.14**	44.68	25.33	55.57	54.45	**75.65**

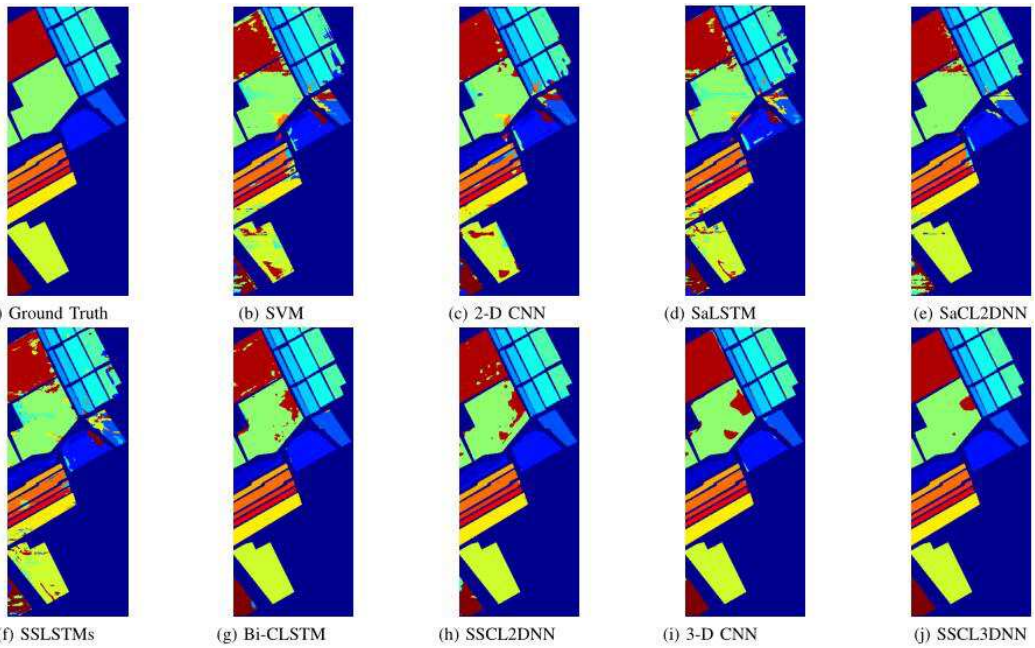


Fig. 6. Classification maps for the Salinas Valley data set. (a) Ground-truth map. (b) SVM. (c) 2-D CNN. (d) SaLSTM. (e) SaCL2DNN. (f) SSLSTMs. (g) Bi-CLSTM. (h) SSCL2DNN. (i) 3-D CNN. (j) SSCL3DNN.

SVM and SaLSTM. In addition, SaCL2DNN is a special case of SSCL2DNN, and compared with it, SSCL2DNN can obtain 0.96%, 1.62%, and 4.78% gains in OA for these three data sets, respectively. For Bi-CLSTM, although ConvLSTM2D is also the core backbone for the feature extraction, this way of simply cascading all the outputs not only does not fully exploit the correlations between different spectral bands, but also easily leads to overfitting problem. Different from it, SSCL2DNN is built by alternating cascades of multi-layer ConvLSTM2D layers and pooling layers, in which the outputs yielded by modeling long-range dependencies are the final feature representations. To some extent, this approach can not only make great use of the characteristics of ConvLSTM2D, but also reduce the number of features and the complexity of the model, and the gains in OA yielded by SSCL2DNN are 2.41%, 0.58%, and 6.25% for these three HSI data sets, respectively, which verifies the effectiveness of SSCL2DNN, and demonstrates that joint learning of spatial-spectral features by modeling long-term dependencies in the spectral field can provide higher classification performance.

On the other hand, SSCL3DNN fuses the spatial and spectral information more effectively by the special 3-D operation, which can preserve the intrinsic structure of hyperspectral data to further improve the classification

performance of SSCL2DNN, and obtains 0.76%, 2.99%, and 5.63% gains in OA for these three HSI data sets, respectively. Compared with 3-D CNN, the improvements in OA provided by SSCL3DNN are 0.51%, 2.12%, and 7.96% for three data sets, respectively. In particular, SSCL3DNN produces the remarkable gains for the Salinas Valley and University of Pavia data sets. From Tables VI-VIII, it can be seen that the 3-D extended architecture of LSTM makes it possible for SSCL3DNN to generate better classification performance by preserving the intrinsic structure of hyperspectral data, and the special gate structures enable SSCL3DNN to extract more discriminative spatial-spectral features. In addition, the classification performance of some classes with high correlation, such as class 10, class 11, and class 12 in the Indian Pines data set, class 8 and class 15 in the Salinas Valley data set, and class 1, class 6, and class 8 in the University of Pavia data set, is improved, which demonstrates the superiority of the proposed models.

Corresponding to Tables VI-VIII, similar conclusions can be drawn from the classification maps presented in Figs. 5-7, from which it is obvious that the maps provided by the proposed models are closest to the ground-truth maps for these three data sets. In addition, there are only fewer misclassifications, and the boundaries of each class are better recognized, especially for class 2, class 10, and class 11 in Fig. 5, class 8 and class 15 in Fig. 6, and class 1, class 6, and class 8 in Fig. 7, which further verifies the effectiveness of the proposed models.

As we all know, it is greatly expensive and difficult to obtain samples with labels. Therefore, it is necessary to investigate the performance under small training size.

D. Sensitivity Comparison and Analysis under Small Samples

In order to further demonstrate the performance of the proposed deep models, we randomly select 10 samples from each labeled class to construct smaller training sets.

The experimental results are reported in Tables IX-X, from which it can be observed that even in the case of small size of training samples, the proposed deep models can also show better classification performance. Compared with 2-D CNN, the improvements in OA yielded by SaCL2DNN are 4.39%, 5.34%, and 16.46%, respectively, and SSCL2DNN improves the performance of SaCL2DNN and obtains 3.12%, 15.43%, and 3.83% gains in OA for these three HSI data sets, respectively, which further verifies the validity of joint learning of the spatial-spectral features. However, Bi-CLSTM obtains better performance than SSCL2DNN, and yields 1.32%, 0.96%, and 1.90% gains in OA, respectively. The main reason is the way of cascading all the outputs of each ConvLSTM2D layer determines that Bi-CLSTM can provide more available features than SSCL2DNN. In particular, it is the defects of Bi-CLSTM and SSCL2DNN that lead to the destruction of the intrinsic structure of the hyperspectral data when putting each component of the local patch as the input of each memory cell. SSCL3DNN overcomes this shortcomings, and yields the best classification performance, which obtains 2.07%, 4.96%, and 22.61% gains in OA for these three data sets, respectively, when compared with 3-D CNN. More detailed results are reported in Tables IX-X, which further demonstrate the advantages of the proposed models.

To further show the effectiveness of the proposed deep models, 20, 30, and 40 samples from each class are randomly extracted for training in these three data sets. In particular, the number of training samples for class 7 and class 9 in the Indian Pines data set is fixed as 10. When taking different numbers of training samples into account, the OA curves of all models are provided in Fig. 8, from which it can be seen that for these three HSI data sets, SSCL3DNN provides the highest classification accuracy. It is worth noting that SSCL3DNN obtains significant performance improvements in the Salinas Valley and University of Pavia data sets. In particular, compared with Bi-CLSTM, SSCL2DNN achieves higher accuracy when the number of training samples is greater than or equal to 20, which means the way of using ConvLSTM2D to model long-range dependencies between different spectral bands may be more effective than that of simply cascading all the outputs. The results in Tables IX-X and Fig. 8 further demonstrate the effectiveness of the proposed deep models, and the design of the special gate structure makes it possible for them to better capture spatial information and preserve the intrinsic structure information of the original data, which is extremely important for improving the classification performance.

VI. Conclusion

In this paper, two novel deep ConvLSTM Neural networks, i.e., SSCL2DNN and SSCL3DNN, have been proposed to extract more effective and discriminative spatial-spectral features for HSI classification. In SSCL2DNN, by taking the local patch as a spectral sequence and feeding into each memory cell band by band, the outputs by modeling long-range dependencies in the spectral domain are the spatial-spectral features, which can reduce the number of features to solve the overfitting problem to a certain extent while achieving satisfactory classification performance. By further developing the 3-D extended structure of LSTM, SSCL3DNN can better preserve the intrinsic structure of hyperspectral data, and the special gate mechanisms enable it to extract more discriminative spatial-spectral features, which can further improve classification performance. The experimental results conducted on three widely used HSI data sets show that the proposed deep models offer competitive advantages over state-of-the-art approaches, especially in the case of small training size.

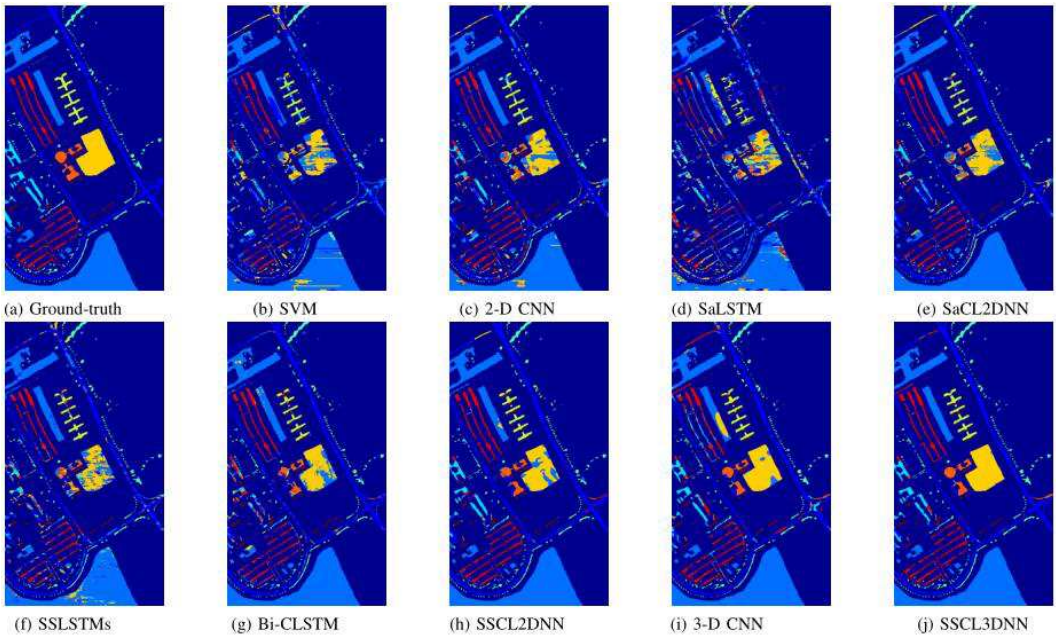


Fig. 7. Classification maps for the University of Pavia data set. (a) Ground-truth map. (b) SVM. (c) 2-D CNN. (d) SaLSTM. (e) SaCL2DNN. (f) SSLSTMs. (g) Bi-CLSTM. (h) SSCL2DNN. (i) 3-D CNN. (j) SSCL3DNN.

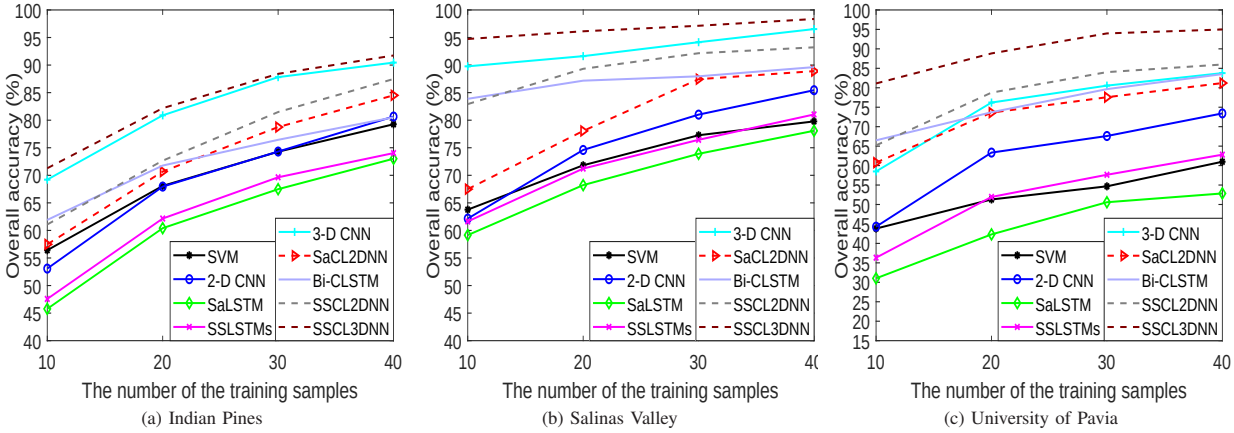


Fig. 8. Overall accuracy (%) of considered methods with different number of training samples for three HSI data sets: (a) Indian Pines, (b) Salinas Valley, (c) University of Pavia.

Acknowledgment

The authors would like to thank the Associate Editor and the Anonymous Reviewers for their valuable comments and suggestions, which are greatly helpful to improve the quality and presentation of this paper.

References

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17-28, Jan. 2002.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6-36, Jun. 2013.

[3] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45-54, Jan. 2014.

[4] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4117-4128, Sep. 2016.

[5] F. M. Lacar, M. M. Lewis, and I. T. Grierson, "Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia," in *IEEE Int. Geosci. Remote Sens. Symp.*, Sydney, NSW, Australia, Jul. 2001, pp. 2875-2877.

[6] A. Ghiyamat and H. Z. Shafri, "A review on hyperspectral remote sensing for homogeneous and heterogeneous forest biodiversity assessment," *Int. J. Remote Sens.*, vol. 31, no. 7, pp. 1837-1856, 2010.

[7] F. van der Meer, "Analysis of spectral absorption features in hyperspectral imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 5,

no. 1, pp. 55-68, Feb. 2004.

[8] L. Pan, H. Li, W. Li, X. Chen, G. Wu, and Q. Du, "Discriminant analysis of hyperspectral imagery using fast kernel sparse and low-rank graph," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6085-6098, Nov. 2017.

[9] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93-97, Jan. 2006.

[10] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210-227, Feb. 2009.

[11] M. Cui and S. Prasad, "Class-dependent sparse representation classifier for robust hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2683-2695, May 2015.

[12] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *Remote Sens.*, vol. 49, no. 10, pp. 3973-3985, Oct. 2011.

[13] H. Zhang, J. Li, Y. Huang, and L. Zhang, "A nonlocal weighted joint sparse representation classification method for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2056-2065, Jun. 2014.

[14] C. Chen, N. Chen, and J. Peng, "Nearest regularized joint sparse representation for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 424-428, Mar. 2016.

[15] J. Peng and Q. Du, "Robust joint sparse representation based on maximum correntropy criterion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7152-7164, Dec. 2017.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[17] G. Zhu, F. Porikli, and H. Li, "Robust visual tracking with deep convolutional neural network based object proposals on pets," in *IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Las Vegas, NV, 2016, pp. 1265-1272.

[18] A. N. Shuaibu, A. S. Malik, and I. Faye, "Adaptive feature learning CNN for behavior recognition in crowd scene," in *IEEE Int. Conf. Signal Image Process. Appl.*, Kuching, 2017, pp. 357-361.

[19] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271-2282, May 2010.

[20] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094-2107, Jun. 2014.

[21] W. Hu, Y. Huang, W. Li, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, no. 2, pp. 1-12, Jul. 2015.

[22] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232-6251, Oct. 2016.

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

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

[25] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847-858, Feb. 2018.

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

[27] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120-147, Nov. 2018.

[28] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173-3184, Jun. 2018.

[29] L. Fang, Z. Liu, and W. Song, "Deep hashing neural networks for hyperspectral image feature extraction," *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1412-1416, Sep. 2019, doi: 10.1109/LGRS.2019.2899823.

[30] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2693-2705, May 2017.

[31] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839-848, Sep. 2017.

[32] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909-1921, Apr. 2018.

[33] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438-447, 2017.

[34] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690-6709, Sep. 2019, doi: 10.1109/TGRS.2019.2907932.

[35] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 3859-3869, 2017.

[36] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep & dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 9, pp. 1454, Sep. 2018.

[37] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2377-2385.

[38] A. Graves, M. Liwicki, S. Fernàndez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855-868, May 2009.

[39] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639-3655, Jul. 2017.

[40] B. Liu, X. Yu, A. Yu, P. Zhang, and G. Wan, "Spectral-spatial classification of hyperspectral imagery based on recurrent neural networks," *Remote Sens. Lett.*, vol. 9, no. 12, pp. 1118-1127, 2018.

[41] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, "Spatial sequential recurrent neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4141-4155, Nov. 2018.

[42] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384-5394, Aug. 2019, doi: 10.1109/TGRS.2019.2899129.

[43] L. Mou, Q. Liu, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924-935, Feb. 2019.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.

[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, Canada, 2014, pp. 3104-3112.

[46] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural network for sequence learning," *Computer Science*, 2015.

[47] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.

[48] H. Wu and S. Prasad, "Convolutional recurrent neural networks for hyperspectral data classification," *Remote Sens.*, vol. 9, no. 3, pp. 298, Mar. 2017.

[49] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, pp. 129, Mar. 2018.

[50] A. Song, J. Choi, Y. Han, and Y. Kim, "Change detection in hyperspectral images using recurrent 3D fully convolutional networks," *Remote Sens.*, vol. 10, no. 11, pp. 1827, Nov. 2018.

[51] M. Seydgar, A. A. Naeini, M. Zhang, W. Li, and M. Satari, "3-D convolution-recurrent networks for spectral-spatial classification of hyperspectral images," *Remote Sens.*, vol. 11, no. 7, pp. 883, Apr. 2019.

[52] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, no. 7, pp. 39-47, Feb. 2017.

[53] M. Rußwurm and M. Körner, "Temporal vegetation modelling using long short-term nemory networks for crop identification from medium-resolution multi-spectral satellite images," in *Conf. Comput. Vis. Pattern Recognit. Workshops*, Honolulu, HI, 2017, pp. 1496-1504.

[54] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *Proc. European Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 312-329.

[55] A. Ma, A. M. Filippi, Z. Wang, and Z. Yin, "Hyperspectral image classification using similarity measurements-based deep recurrent neural networks," *Remote Sens.*, vol. 11, no. 2, pp. 194, Jan. 2019.

[56] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, pp. 17-28, Dec. 2017.

[57] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1-27, Mar. 2011.