Lightweight Tensor Attention-Driven ConvLSTM Neural Network for Hyperspectral Image Classification

Wen-Shuai Hu, Heng-Chao Li, *Senior Member, IEEE*, Yang-Jun Deng, Xian Sun, *Senior Member, IEEE*, Qian Du, *Fellow, IEEE*, and Antonio Plaza, *Fellow, IEEE*

***Abstract*—Recurrent neural networks, especially the convolutional long short-term memory (ConvLSTM), have attracted plenty of attention and shown promising results due to their ability in modeling long-term dependencies in many research fields. In this paper, a lightweight tensor attention-driven ConvLSTM neural network (TACLNN) is proposed for hyperspectral image (HSI) classification. Firstly, to reduce the trainable parameters and memory requirements of ConvLSTM (specifically, the 2-D version of LSTM, i.e., ConvLSTM2D), a lightweight ConvLSTM2D cell is developed by utilizing tensor-train decomposition, resulting in a TT-ConvLSTM2D cell, with which a spatial-spectral TT-ConvLSTM 2-D neural network (SSTTCL2DNN) is built. However, it is inevitable for SSTTCL2DNN to obtain lower accuracies for HSI classification. To recover the accuracy loss caused by the TT-ConvLSTM2D cell in SSTTCL2DNN, a learnable tensor attention residual block (TARB) module is built to further enhance its geometrical structure. When applied to three widely used HSI benchmarks, the proposed TACLNN model outperforms several state-of-the-art methods for HSI classification. In addition, the proposed TACLNN can effectively reduce the number of parameters and storage requirements achieving higher classification accuracies as compared to other competitive baselines.**

***Index Terms*—Attention mechanism, convolutional long short-term memory, hyperspectral image classification, lightweight cell, performance recovery, tensor representation.**

Manuscript received May 4, 2020; revised December 18, 2020; accepted February 22, 2021. Date of publication March 4, 2021; date of current version March 29, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61871335 and in part by the Fundamental Research Funds for the Central Universities under Grants 2682020XG02, and 2682020ZT35. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Hongyang Chen. *(Corresponding author: Heng-Chao Li.)*

Wen-Shuai Hu and Heng-Chao Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: wshu@my.swjtu.edu.cn; lihengchao_78@163.com).

Yang-Jun Deng is with the College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China (e-mail: dyj2012@yeah.net).

Xian Sun is with the Key Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: sunxian@mail.ie.ac.cn).

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

Antonio Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10060 Cáceres, Spain (e-mail: aplaza@unex.es).

Digital Object Identifier 10.1109/JSTSP.2021.3063805

I. Introduction

HYPERSPECTRAL images (HSIs) contain very detailed spatial and spectral information, which can be exploited to analyze the information of the Earth's surface [1]. These images have been utilized in many applications, i.e., geological exploration [2] and precision agriculture [3].

The classification of HSIs has become an important research topic. According to the (available or not) labeled data, there are three kinds of classification methods, i.e., unsupervised, semisupervised, and supervised. Supervised methods generally provide better accuracy. With the advance of deep learning in the field of computer vision [4], [5], many techniques have been developed for HSI classification. Since Hu *et al.* [6] and Chen *et al.* [7] introduced the convolutional neural network (CNN) into HSI classification, many CNN-based models have been proposed. The pixel-pair model was integrated into CNN to extract the spatial-spectral features [8]. In [9], spatial and spectral information were fused by using CNN and a balanced local discriminant embedding. To jointly learn the spatial-spectral features, some researchers have designed many 3-D models [10], [11], where the 3-D data yielded from the original HSI data serves as their input. Luo *et al.* [12] and Roy *et al.* [13] utilized the hybrid structure by integrating support vector machine (SVM), 2-D CNN with 3-D CNN to solve overfitting problem. Fang *et al.* [14] proposed a deep hashing neural network that improves spatial-spectral features yielded from the classes. Considering the limited availability of labeled data, some deep models (i.e., siamese CNN model [15]) have been designed to overcome this issue. In addition, active learning [16] and transfer learning [17] have been adapted to improve the training of these models under limited samples. Paoletti *et al.* [18] conducted a systematic review of deep models for HSI classification, and compared the commonly-used classifiers, which provides effective guidelines for the future research.

In addition to the above works, recurrent neural networks (RNNs) [19] –especially the long short-term memory (LSTM) [20] and its 2-D version, convolutional LSTM (ConvLSTM) [21], renamed as ConvLSTM2D in [22]–have attracted significant attention due to their unique capacity to model the long-range dependencies, from which many spatial-spectral feature extraction algorithms have been proposed by integrating CNNs and RNNs, such as the cascaded RNN [23], the multi-scale hierarchical recurrent neural networks [24], semisupervised 1-D convolutional RNN (CRNN) [25], 2-D CRNN and its 3-D version [26], and an adaptive spatial-spectral multiscale network [27]. Moreover, by using LSTM and ConvLSTM2D cells as the basic units, the spatial-spectral LSTMs (SSLSTMs) [28], bidirectional-ConvLSTM (Bi-CLSTM) [29], and spatial-spectral ConvLSTM 2-D neural network (SSCL2DNN) [22] were proposed for joint learning of spatial-spectral features.

The attention mechanism was originally derived from computational neurosciences [30] and can allow a model to automatically locate and focus on the most useful information from the input. Since Bahdanau *et al.* [31] utilized the attention mechanism to select reference words from the source sentences, numerous works have shown that deep learning-based models that incorporate attention mechanisms can gain better feature representation ability [32]-[34]. Moreover, it has been applied to the analysis of remote sensing images. Cui *et al.* [35] proposed a dense attention pyramid network for ship detection in synthetic aperture radar (SAR) images, where a convolutional attention module with spatial- and channel-wise attentions is designed for highlighting salient features of specific scales. Chen *et al.* [36] improved the faster region-based CNN model by using multi-scale, spatial- and channel-wise attentions for object detection in remote sensing imagery. For the HSI classification, an attention-based inception model was built to yield a special attention pattern in [37], which can adaptively select different kinds of the spatial information. By utilizing CNN and attention mechanism, a band attention convolutional network was proposed to extract effective spatial-spectral features for HSI classfication [38]. Mei *et al.* [39] integrated CNN, ResNet, and attention mechanism to construct a two-branch spatial-spectral attention network for joint learning of the spatial-spatial information. With the help of densely connected networks (DenseNets), a double-branch multi-attention mechanism network [40] and a 3-D DenseNet with a spectral-wise attention module [41] were proposed for HSI classification. In addition, by combining with the attention mechanism and residual learning, a spatial-spectral attention-driven feature extraction model was designed in [42].

To reduce the memory requirements of the CNN-based models, in [43], the fully connected (FC) layers were compressed by representing their parameters in a tensor-train (TT) format. TT decomposition (TTD) [44] is an useful tensor factorization model with the advantage of being able to scale to an arbitrary number of dimensions. Inspired by [45], Garipov *et al.* [46] applied TTD to the convolution kernels for the design of the lightweight convolutional layer. Furthermore, the TT-format representation of a RNN model was completed in [47]. Yang *et al.* [48] integrated the TTD into the LSTM for video classification. Compared with the convolutional layers in the CNN, more trainable parameters and higher storage requirements are needed by each ConvLSTM2D layer.

In this paper, a new and effective tensor attention-driven ConvLSTM neural network (TACLNN) model is proposed for HSI classification. To reduce the number of the parameters and memory requirements, a lightweight ConvLSTM2D cell is developed by using TTD to improve the efficiency of the calculation, thus builting a spatial-spectral TT-ConvLSTM 2-D neural network (SSTTCL2DNN). However, it is the reduction of the number of parameters in each ConvLSTM2D layer that results in the performance loss of the whole model. Specifically, tensor representation can not only reduce the data dimensionality, but also retain the geometrical structure of the data. To recover the performance loss of SSTTCL2DNN, a learnable tensor attention residual block (TARB) module is built by combining tensor representation of HSI data and attention mechanism to enhance the feature extraction ability. Then, the spatial-spectral features extracted by the SSTTCL2DNN are improved by the TARB module, resulting in a new TACLNN model which can effectively reduce the number of parameters without degrading the classification performance. The main contributions of our work are summarized as follows:

1) To reduce the computational complexity and memory requirements, a lightweight ConvLSTM2D cell is developed by utilizing TTD. By using it as the fundamental unit, an SSTTCL2DNN is further constructed, which can effectively reduce the number of parameters within a very small range of accuracy degradation.
2) For recovering the performance loss caused by the reduction of the parameters in the SSTTCL2DNN, an effective TARB module is proposed to preserve the geometrical structure of HSI, using a lightweight TACLNN model to achieve satisfactory classification accuracy with the addition of two training parameters only.

The rest of this paper is organized as follows. ConvLSTM2D, TT convolutional layer, and attention mechanism are introduced in Section II. In Section III, the TACLNN model is described in detail. A detailed analysis of parameter settings and a quantitative evaluation on three public HSI data sets is given in Section IV, followed by conclusions in Section V.

II. Related Work

A. ConvLSTM2D

A ConvLSTM cell was developed by extending the input-to-state and state-to-state transitions in LSTM to the 2-D convolution operation in [21]. For convenience, we call it ConvLSTM2D cell, whose calculation is performed as follows:

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

where H_{t-1}, C_{t-1}, and X_t are the output and the state of the last cell and the input of the current cell, respectively. i_t, f_t, and H_t denote three gate structures, i.e., input, forget, and output gates, and the corresponding convolution kernels are $W_{\cdot i}$, $W_{\cdot f}$, and $W_{\cdot o}$, respectively with $\cdot$ representing x, h, and c. In addition, $\circ$, σ, and $*$ are respectively the Hadamard product, nonlinear activation function, and convolution operation.

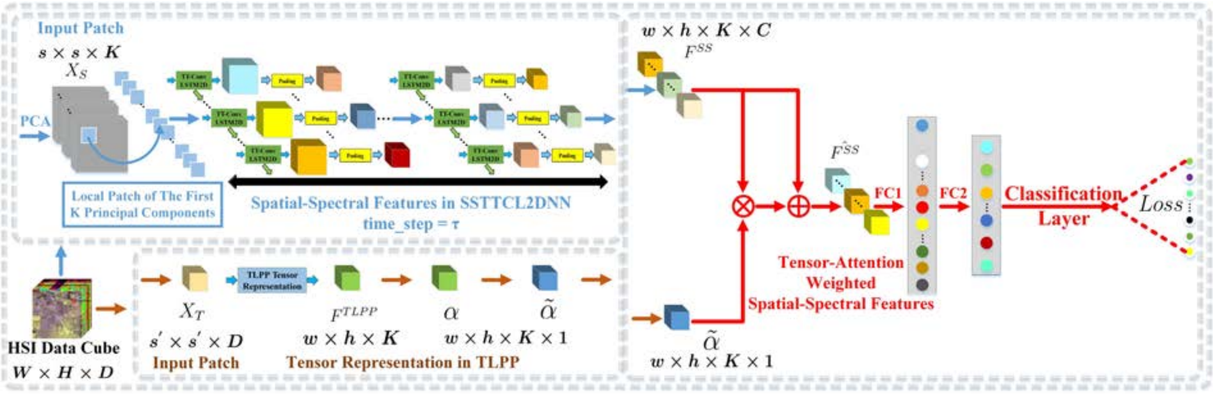

Input Patch
$s \times s \times K$
X_S
PCA
TT-Conv LSTM2D
Pooling
Local Patch of The First K Principal Components
Spatial-Spectral Features in SSTTCL2DNN
time_step = τ
HSI Data Cube
$W \times H \times D$
X_T
$s' \times s' \times D$
Input Patch
TLPP Tensor Representation
F^{TLPP}
$w \times h \times K$
α
$w \times h \times K \times 1$
$\tilde{\alpha}$
Tensor Representation in TLPP
$w \times h \times K \times C$
F^{SS}
$\hat{F^{SS}}$
FC1
FC2
Classification Layer
Loss
Tensor-Attention Weighted Spatial-Spectral Features
$\tilde{\alpha}$
$w \times h \times K \times 1$

Fig. 1. Framework of the proposed TACLNN model.

As shown in (1), there are two kinds of weights (ignoring $W_{c\cdot}$). Specifically, $W_{x\cdot}$ is the weight of the input gate with a size of $k \times k \times C \times S$, and $W_{h\cdot}$ is a $k \times k \times S \times S$ convolution filter of the output gate, where k is the kernel size, and C and S are the number of the input and output channels, respectively. The total number of parameters in each ConvLSTM2D cell is:

$$N_1 = 4\,k^2 S(S + C). \tag{2}$$

Compared with the convolution filters, the number of the bias parameters is insignificant; hence, the bias vectors are not considered in the experiments. From (2), there are 4 $k^2 S(S + C)$ parameters, resulting in large storage requirements than convolutional layer in the CNN. Therefore, it is necessary to develop a lightweight ConvLSTM2D cell to achieve higher computation efficiency and memory savings.

B. Tensor-Train Convolutional Layer

By using TTD, a d-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{l_1 \times l_2 \cdots \times l_d}$ can be decomposed into a set of tensors $\boldsymbol{\mathcal{G}}_p \in \mathbb{R}^{l_p \times r_{p-1} \times r_p}$. In [43], the factorization of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{(u_1 \cdot v_1) \times (u_2 \cdot v_2) \cdots \times (u_d \cdot v_d)}$ can be written as:

$$\begin{aligned} &\boldsymbol{\mathcal{A}}((R_1, T_1), (R_2, T_2), \ldots, (R_d, T_d)) \\ &\quad = \boldsymbol{\mathcal{G}}'_1[R_1, T_1]\boldsymbol{\mathcal{G}}'_2[R_2, T_2] \cdots \boldsymbol{\mathcal{G}}'_d[R_d, T_d], \end{aligned} \tag{3}$$

where $\boldsymbol{\mathcal{G}}'_p[R_p, T_p] \in \mathbb{R}^{r_{p-1} \times r_p}$, $R_p = \lfloor \frac{e_p}{v_p} \rfloor$, and $T_p = e_p - v_p \lfloor \frac{e_p}{v_p} \rfloor$. $l_p = u_p \times v_p, e_p = 1, 2, \ldots, l_p$, and $p = 1, 2, \ldots, d$, in which l_p is the dimension of the mode-p of $\boldsymbol{\mathcal{A}}$.

Taking the convolutional layer as an example, the input tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{w \times h \times C}$ is transformed into the output $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{w \times h \times S}$ by the kernel $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{k \times k \times C \times S}$, where w and h are the width and height, respectively. To reduce the computational complexity, a TT-convolutional (TTC) layer was built in [46] by decomposing the kernel along with the channel dimension with TTD, and the outputs of the TTC layer is expressed as:

$$\begin{aligned} \boldsymbol{\mathcal{Y}}(k_w, k_h, s_1, s_2, \ldots, s_d) &= \sum_{j_w=1}^{k} \sum_{j_h=1}^{k} \sum_{c_1, c_2, \ldots, c_d} \\ &\boldsymbol{\mathcal{X}}(j_w + k_w - 1, j_h + k_h - 1, c_1, c_2, \ldots, c_d) \cdot \\ &\boldsymbol{\mathcal{G}}'_0[j_w, j_h]\boldsymbol{\mathcal{G}}'_1[c_1, s_1]\boldsymbol{\mathcal{G}}'_2[c_2, s_2] \cdots \boldsymbol{\mathcal{G}}'_d[c_d, s_d], \end{aligned} \tag{4}$$

where $C = \prod_{p=1}^{d} C_p$, $S = \prod_{p=1}^{d} S_p$, and $k_w, k_h = 1, 2, \ldots, k$. $\boldsymbol{\mathcal{G}}'_p$ is the $TTC - cores$, which is the parameters that need to be trained in this TTC layer. The set $\{r_p\}_{p=0}^{d+1}$ is the $TTC - ranks$, where r_0 and r_{d+1} are 1.

For convenience, the above calculation can be defined as:

$$Y = TTCL(K, X). \tag{5}$$

In this paper, to reduce the memory consumption of the ConvLSTM2D in (1), a lightweight ConvLSTM2D cell will be constructed by using TTD, which is called TT-ConvLSTM2D cell. A more detailed description will be given in Section III-B.

C. Attention Mechanism

Vaswani *et al.* [49] utilized the following equation to calculate the outputs of attention mechanism:

$$Attention(Q, \widetilde{K}, V) = softmax(f(Q, \widetilde{K}))V, \tag{6}$$

where $f(\cdot)$ means the attention function, and Q, $\widetilde{K}$, and V are the inputs of attention mechanism. $softmax(\cdot)$ is the softmax function for normalization.

By taking the advantages of the tensor representation and attention mechanism, a TARB module is designed to enhance the intrinsic geometrical structure information of deep learning models, which will be introduced in Section III-C. To the best of our knowledge, this is the first attempt of constructing a tensor attention structure for HSI classification purposes.

III. TACLNN

A. Architecture Overview

The framework of the TACLNN model is shown in Fig. 1. To reduce the number of trainable parameters, a lightweight TT-ConvLSTM2D cell is presented in Section III-B. The SSTTCL2DNN model is demonstrated in Section III-C. In Section III-D, a novel TARB module is described in detail. Finally, our TACLNN model can be found in Section III-E.

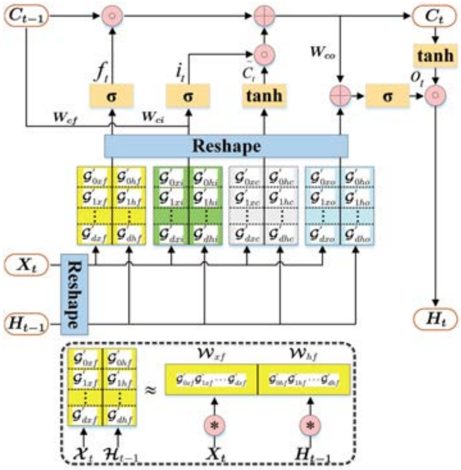


Fig. 2. Illustration of the developed TT-ConvLSTM2D Cell.

B. TT-ConvLSTM2D

From (1), there are two different convolution weights, i.e., $W_{x\cdot}$ and $W_{h\cdot}$. To reduce the number of the parameters and the storage requirements, a lightweight TT-ConvLSTM2D cell is developed by using the TTC layer, as shown in Fig. 2.

Firstly, the input X_t and output H_{t-1} in (1) are reshaped into the $(2+d)$-order tensors $\boldsymbol{\mathcal{X}}_t \in \mathbb{R}^{w\times h\times C_1\times C_2\times\cdots\times C_d}$ and $\boldsymbol{\mathcal{H}}_{t-1} \in \mathbb{R}^{w\times h\times S_1\times S_2\times\cdots\times S_d}$, respectively. And then, inspired by the TTC layer in Section II-B, the decompositions of $W_{x\cdot}$ and $W_{h\cdot}$ corresponding to X_t and H_{t-1} are expressed as:

$$\begin{aligned}\boldsymbol{\mathcal{W}}_{x\cdot} &= \boldsymbol{\mathcal{G}}'_0[j_w, j_h]_{x\cdot}\boldsymbol{\mathcal{G}}'_1[c_1, s_1]_{x\cdot}\cdots\boldsymbol{\mathcal{G}}'_d[c_d, s_d]_{x\cdot}\\ \boldsymbol{\mathcal{W}}_{h\cdot} &= \boldsymbol{\mathcal{G}}'_0[j_w, j_h]_{h\cdot}\boldsymbol{\mathcal{G}}'_1[s_1, s_1]_{h\cdot}\cdots\boldsymbol{\mathcal{G}}'_d[s_d, s_d]_{h\cdot}.\end{aligned} \quad (7)$$

Finally, substituting $\boldsymbol{\mathcal{X}}_t$, $\boldsymbol{\mathcal{H}}_{t-1}$, and (7) into (4), the corresponding outputs $\boldsymbol{\mathcal{Y}}_t^{x\cdot}$ and $\boldsymbol{\mathcal{Y}}_{t-1}^{h\cdot} \in \mathbb{R}^{w\times h\times S_1\times S_2\times\cdots\times S_d}$ are obtained, which are reshaped into the third-order tensors $Y_{t-1}^{x\cdot}$ and $Y_{t-1}^{h\cdot} \in \mathbb{R}^{w\times h\times S}$ as the final output of each gate unit.

Based on the above analysis, the whole calculation formulas of the TT-ConvLSTM2D cell can be written as:

$$\begin{aligned}i_t &= \sigma(TTCL(W_{xi}, X_t) + TTCL(W_{hi}, H_{t-1})\\ &\quad + W_{ci}\circ C_{t-1} + b_i)\\ f_t &= \sigma(TTCL(W_{xf}, X_t) + TTCL(W_{hf}, H_{t-1})\\ &\quad + W_{cf}\circ C_{t-1} + b_f)\\ \tilde{C}_t &= \tanh(TTCL(W_{xc}, X_t) + TTCL(W_{hc}, H_{t-1}) + b_c)\\ C_t &= f_t\circ C_{t-1} + i_t\circ\tilde{C}_t\\ o_t &= \sigma(TTCL(W_{xo}, X_t) + TTCL(W_{ho}, H_{t-1})\\ &\quad + W_{co}\circ C_t + b_o)\\ H_t &= o_t\circ\tanh(C_t).\end{aligned} \quad (8)$$

Obviously, the total number of parameters of all convolution filters in the TT-ConvLSTM2D cell can be obtained as:

$$N_2 = 8\,k^2 r_1 + 4\sum_{p=1}^{d} s_p r_{p+1} r_p (c_p + s_p). \quad (9)$$

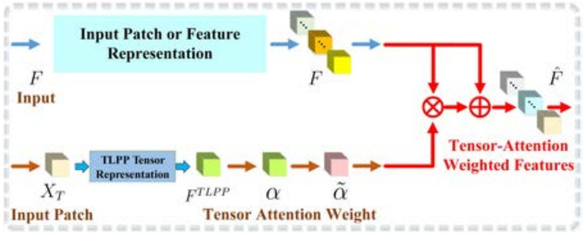

Input Patch or Feature Representation
F
Input
F
$\hat{F}$
Tensor-Attention Weighted Features
X_T
Input Patch
TLPP Tensor Representation
F^{TLPP}
α
$\tilde{\alpha}$
Tensor Attention Weight

Fig. 3. Structure of the proposed TARB module.

Therefore, inspired by (2) and (9), the compression rate of each ConvLSTM2D cell is described as:

$$\frac{N_1}{N_2} = \frac{k^2 S(S+C)}{2\,k^2 r_1 + \sum_{p=1}^{d} s_p r_{p+1} r_p (c_p + s_p)}. \quad (10)$$

Based on the TT-ConvLSTM2D cell in (8) and by utilizing it as a basic unit, a lightweight TT-ConvLSTM2D layer is further built. Similar to an ordinary ConvLSTM2D layer, the TT-ConvLSTM2D layer can also be used alone or with the CNN to build feature extraction models for various applications.

C. SSTTCL2DNN

In [22], by decomposing the local window patch into a spectral sequence as the input of each ConvLSTM2D cell (in band by band fashion), SSCL2DNN was used for spatial-spectral feature extraction by modeling the long-range dependencies in the spectral dimension for HSI classification, whose main backbone contains two ConvLSTM2D and two pooling layers.

Based on the above analysis, the design of gate structures in each ConvLSTM2D cell results in a large number of trainable parameters and high storage requirements. To improve the computational efficiency of SSCL2DNN, a lightweight spatial-spectral feature extraction model (namely SSTTCL2DNN) is built by applying the TT-ConvLSTM2D cell. According to (10), the number of the parameters in SSTTCL2DNN is effectively reduced, which is helpful for its practical application.

Although the trainable parameters and the storage requirements of SSTTCL2DNN will be effectively reduced, its classification accuracy will be inevitably lost. Consequently, to recover the performance loss of SSTTCL2DNN, a trainable and effective TARB module is further constructed, whose detailed structure is given in Section III-D.

D. TARB

To recover the performance loss of SSTTCL2DNN, and inspired by the advantages of the tensor representation of HSI data, a TARB module is developed in this section, whose structure is shown in Fig. 3. In particular, the tensor locality preserving projection (TLPP) [50] model is utilized in the TARB model for the feature extraction and dimensionality reduction, which can effectively preserve the geometrical structure of HSI data. Therefore, it is possible for the SSTTCL2DNN to combine tensor representation to obtain better performance and effectively reduce the number of the parameters.

Suppose that $X \in \mathbb{R}^{W \times H \times D}$ is the original HSI data, where W, H, and D are the width, height, and the number of the spectral bands, respectively. Due to the fact that the spatial information is helpful for HSI classification, the data with size of $s' \times s'$ in a local window is set to capture the spatial information around each pixel x, which results in a 3-D volume expressed by $X_T \in \mathbb{R}^{s' \times s' \times D}$, used as the input of TARB.

For the TLPP in TARB, given q samples $X_{T_1}, \ldots, X_{T_q}$, the heat kernel function is first applied to define a similarity matrix, with which a neighborhood graph is built to describe the local geometric structure of these samples [50]. Then, for the sample X_{T_m}, the transformation matrices $U_p (p = 1, 2, \ldots, d)$ are calculated by minimizing the following objective function:

$$\begin{aligned} &\min J(U_1, \ldots, U_d) \\ &= \min_{U_1,\ldots,U_d} \sum_{m,n} \| X_{T_m} \times_1 U_1 \cdots \times_d U_d \\ &\qquad - X_{T_n} \times_1 U_1 \cdots \times_d U_d \|_F^2 \widetilde{W}_{mn}, \\ &s.t. \ \sum_m \| X_{T_m} \times_1 U_1 \cdots \times_d U_d \|^2 d_{mm} = 1. \end{aligned} \tag{11}$$

where $X_{T_m}, X_{T_n} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_d}$, $m, n = 1, 2, \ldots, q$, and $\widetilde{W} \in \mathbb{R}^{q \times q}$ is the similarity matrix. After that, the optimization problem of (11) is transformed into a generalized eigenvalue problem, thus obtaining transformation matrices $U_p \in \mathbb{R}^{\widetilde{l_p} \times l_p} (\widetilde{l_p} < l_p)$, where $\widetilde{l_p}$ is the dimension of mode-p after dimension reduction. Finally, by executing p-mode product of U_p and X_{T_m} along its all dimensions, the low-dimensional tensor features Y_{T_m} are extracted, where $Y_{T_m} = X_{T_m} \times_1 U_1 \cdots \times_p U_p \cdots \times_d U_d$, and $\times_p$ is the p-mode product.

Therefore, based on the above analysis, as for the input data X_T of the pixel x, its low-dimensional tensor feature $F^{TLPP} \in \mathbb{R}^{w \times h \times K}$ after feature extraction and dimension reduction in TARB can be learned as follows:

$$F^{TLPP} = X_T \times_1 U_1 \times_2 U_2 \cdots \times_d U_d, \tag{12}$$

where K denotes the number of the spectral bands, and d in the TARB module is 3 in the experiments.

Based on the above preprocessing stage, the input X_T is transformed into a tensor representation F^{TLPP} with a size of $w \times h \times K$, which is then reshaped into an unnormalized attention map $\alpha \in \mathbb{R}^{w \times h \times K \times 1}$. It is further normalized by a softmax function to obtain a tensor attention weight $\widetilde{\alpha}$, resulting in a normalized one. Inspired by the residual learning, the enhanced feature representation in TARB is written as:

$$\hat{F} = F \odot \widetilde{\alpha} + F, \tag{13}$$

where F means the input data or the features extracted by other models, and $\odot$ is an element-based product operation.

Tensor representation can preserve the geometrical structure information of HSI data, which is beneficial to HSI classification. The TARB module can be utilized to strengthen the attention devoted to the original feature map from the viewpoint of geometrical structure, and is flexible to be inserted into any layer of deep models, resulting in a feature representation that leads to geometrical structure enhancement. It should be noted that, to the best of our knowledge, this is the first attempt to construct a tensor attention structure for HSI classification.

E. TACLNN

Although the efficiency of calculation in SSTTCL2DNN has been improved, its performance is inevitably degraded. To recover the performance loss caused by the parameter reduction, a TACLNN model is proposed by combining SSTTCL2DNN with TARB, whose framework is shown in Fig. 1.

Firstly, for the SSTTCL2DNN, similar with [22], the 3-D data with a size of $X_S \in \mathbb{R}^{s \times s \times K}$ is built from the original HSI data X, in which $s \times s$ is the spatial information extracted from the HSI data around the pixel x, and K is the number of spectral bands after dimension reduction through principal component analysis (PCA). After data preprocessing, X_S is decomposed into τ 2-D components and then converted into a sequence, i.e., $\{X_S^1, \ldots, X_S^t, \ldots, X_S^\tau\}$, $t \in \{1, 2, \ldots, \tau\}$, where τ is the dimension time_step in each TT-ConvLSTM2D layer, and τ is set to K. After the feature extraction by the SSTTCL2DNN, the spatial-spectral features can be obtained by modeling the long-term dependencies in the spectral domain and expressed as $F^{SS} \in \mathbb{R}^{w \times h \times K \times C}$.

To recover the performance loss of SSTTCL2DNN caused by the parameter reduction, a simple but effective TARB module is constructed for enhancing its feature representation ability. As shown in Section III-D, a normalized attention map $\widetilde{\alpha}$ is extracted from input X_T of the pixel x, and inspired by (13), the enhanced spatial-spectral features are given by:

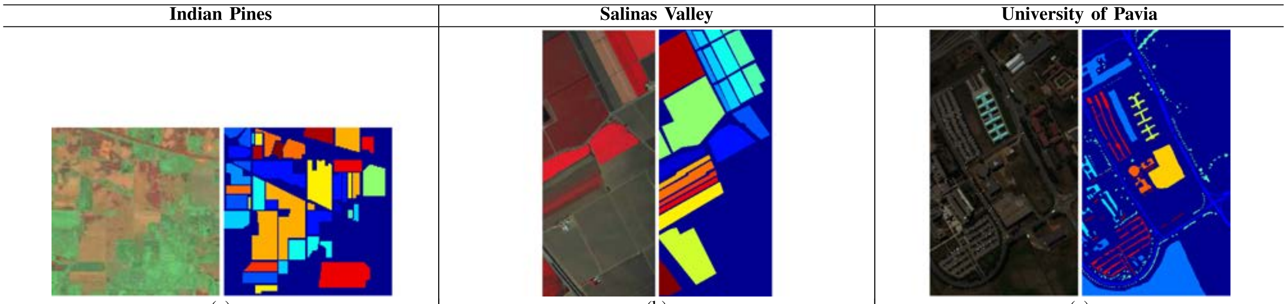
$$F^{\hat{SS}} = F^{SS} \odot \widetilde{\alpha} + F^{SS}, \tag{14}$$

where $F^{\hat{SS}}$ are the enhanced features in our TACLNN model.

Finally, at the top of TACLNN, the spatial-spectral features $F^{\hat{SS}}$ are vectorized as a 1-D vector $f^{\hat{SS}}$, and the FC layers are applied to map the feature space to the class label space, followed by a softmax function to predict the conditional probability distribution $P_{\tilde{c}} = P(y = \tilde{c} | f^{\hat{SS}}, W_{fc}, b_{fc}) = \frac{e^{(W_{fc} f^{\hat{SS}} + b_{fc})}}{\sum_{jc=1}^{N_{\tilde{c}}} e^{(W_{jc} f^{\hat{SS}} + b_{jc})}}$ of each class $\tilde{c}$, where $\tilde{c} \in \{1, 2, \ldots, N_{\tilde{c}}\}$, and $N_{\tilde{c}}$ is the number of classes. Moreover, the cross entropy is utilized as the loss function, which is described as $Loss$.

In our TACLNN model, all the weights and biases need to be learned. To train the whole model, firstly, the 3-D data of different local windows corresponding to pixel x are built for SSTTCL2DNN and TARB, i.e., X_S and X_T. Then, by solving (11), the projection matrices U_p are calculated and, according to (12), the tensor representation F^{TLPP} is obtained, which further results in a tensor attention weight $\widetilde{\alpha}$. Finally, $Loss$ is optimized in N_{steps} epochs to yield final classification result. Algorithm 1 describes the training method in more details.

It should be noted that the adaptive momentum algorithm is adopted to optimize the loss function with the learning rate lr. More detailed parameter settings are given in Section IV.



(a) (b) (c)

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)

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)

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 these 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) Number of the training samples.

```
Algorithm 1: Training TACLNN for HSI Classification.
    Input: HSI data $X$;Ground truth $Y$
    Output: Classification map $\Omega$
    1: Prepare the 3-D data $X_S$ and $X_T$ of the pixel $x$ for
       SSTTCL2DNN and TARB
    2: Parameter setting and weights initialization
    3: Solve (11) to obtain the projection matrices $U_p$
    4: Obtain the tensor representation $F^{TLPP}$ by
       calculating (12) to further obtain the attention weight $\widetilde{\alpha}$
    5: While step $\leq N_{steps}$
    6:    Train the whole classification model by optimizing
          the loss function $Loss$
    7: End While
    8: Return Classification map $\Omega$
```

IV. Experimental Results

To quantitatively and qualitatively verify the validity of our TACLNN, SVM [51], SSLSTMs [28], Bi-CLSTM [29], SSCL2DNN [22], and SSTTCL2DNN [52] are selected as the compared algorithms. Three quantitative metrics are applied to measure their performance, i.e., overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ). To eliminate the bias caused by randomly choosing training samples, the average values of them after 10 Monte Carlo runs are utilized. All experiments are conducted on a desktop with an Intel Core i7-8700 processor and a Nvidia GeForce GTX 1080ti GPU.

A. Experimental Data

To evaluate our TACLNN, three HSI data sets, i.e., Indian Pines, Salinas Valley, and University of Pavia, are utilized in the experiments. In particular, the false-color maps, ground-truth maps, and the training samples are presented in Fig. 4.

1) Indian Pines: It was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in Northwestern Indiana, USA. After removing some noisy spectral bands, 200 bands are used for the final study. Its spatial size is 145×145 pixels, and there are 16 different class labels.

2) Salinas Valley: It was acquired by the AVIRIS sensor over Salinas Valley, California, with a size of 512×217 pixels, and contains 16 classes. After removing the water absorption bands and noise-affected bands, 204 spectral bands are preserved.

3) University of Pavia: It was captured by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over University of Pavia, Northern Italy, and has 610 lines and 340 columns and 9 classes. After removing several noise-corrupted bands, 103 spectral bands are retained for analysis.

B. Parameter Settings

Similar to [22], PCA is applied to reduce the computational complexity of the whole model, where the first K principal components are retained to extract the spatial-spectral features.

For all compared methods, the parameter settings of SVM, SSLSTMs, Bi-CLSTM, SSCL2DNN, and SSTTCL2DNN are obtained according to [51], [28], [29], [22], [52] for quasi-optimal results. For our TACLNN, some parameters need to be

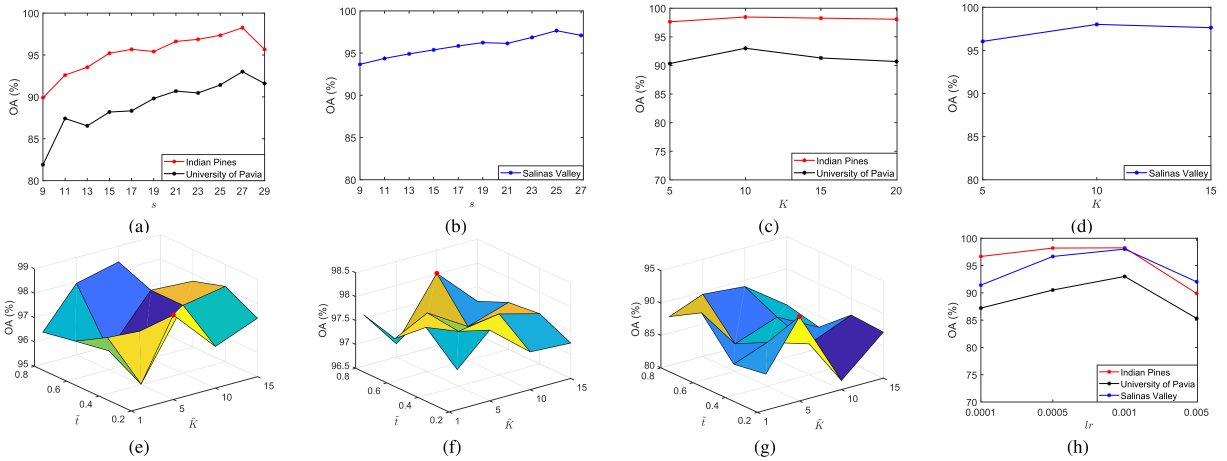


Fig. 5. OA (%) of the proposed TACLNN model with different parameters. (a)-(b) Size $s \times s$ of the local window. (c)-(d) Number K of the principal components. (e)-(g) Value of $[\tilde{K}, \tilde{t}]$ in the developed TARB module. (h) Learning rate lr.

TABLE I
SENSITIVITY ANALYSIS UNDER DIFFERENT VALUE OF M

M	Indian Pines	Salinas Valley	University of Pavia
$\{8, 16\}$	97.39	97.32	87.89
$\{16, 32\}$	97.66	97.45	90.29
$\{32, 64\}$	**98.25**	**98.02**	**93.01**
$\{64, 128\}$	97.36	96.91	85.92

TABLE II
SENSITIVITY ANALYSIS UNDER DIFFERENT VALUE OF $TTC - rank$

$TTC - rank$	Indian Pines	Salinas Valley	University of Pavia
4	97.58	94.78	86.71
6	97.35	97.32	89.85
8	**98.25**	**98.02**	**93.01**
10	97.93	97.85	91.74
12	96.99	97.71	86.64

tuned, i.e., the spatial sizes ($s \times s$ and $s' \times s'$), the number (K) of principal components, the kernel size ($k \times k$), the number (M) of feature maps, and the value of $TTC - rank$ in each TT-ConvLSTM2D layer, the values of $\tilde{K}$ and $\tilde{t}$ in the TARB module, and the learning rate lr. First, K is fixed to 10, and lr is set to 0.001 from epochs 1 to 2000. M is fixed to $\{32, 64\}$. $\tilde{K}$ and $\tilde{t}$ are set to 5 and 0.2, respectively, and $TTC - rank$ is 8. Then, s is yielded from $\{9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29\}$ (the corresponding s' is from $\{3, 3, 5, 5, 5, 5, 7, 7, 7, 7, 9\}$) for the Indian Pines and University of Pavia data sets while from $\{9, 11, 13, 15, 17, 19, 21, 23, 25, 27\}$ for the Salinas Valley data set due to memory problems. k is from $\{3, 4, 5\}$. Experimental results for analyzing the influence of the different values of s on the classification performance can be found in Fig. 5(a)-(b). For these three HSI data sets, the optimal size of the local window is 27×27, 25×25, and 27×27, respectively.

Then, regarding the value of K, the optimal K is obtained 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 due to memory limitation. The value of OA of TACLNN as K varies is given in Fig. 5(c)-(d). The quasi-optimal value of K is set to 10 for these three HSI data sets.

Furthermore, the classification accuracy on different values of M is analyzed in Table I. The optimal number of the feature maps for these three HSI data sets is fixed to $\{32, 64\}$.

The performance of TACLNN on different values of $TTC - rank$ is further analyzed, and $TTC - rank$ is selected from $\{4, 6, 8, 10, 12\}$. From Table II, TACLNN yields optimal performance when $TTC - rank$ is 8 for three HSI data sets.

TABLE III
PARAMETER SETTINGS FOR THE INDIAN PINES DATA SET

Layer Name	Kernel Size	Output Size for SSTTCL2DNN	Output Size for TLPP
Input		$27 \times 27 \times 10 \times 1$	$7 \times 7 \times 200$
TT-ConvLSTM2D Layer	4×4	$27 \times 27 \times 10 \times 32$	
MaxPooling2D Layer	2×2	$14 \times 14 \times 10 \times 32$	
TLPP			$7 \times 7 \times 10$
TT-ConvLSTM2D Layer	3×3	$14 \times 14 \times 10 \times 64$	
MaxPooling2D Layer	2×2	$7 \times 7 \times 10 \times 64$	
TARB		$7 \times 7 \times 10 \times 64$	
Dropout	0.5	$7 \times 7 \times 10 \times 64$	
Flatten		31360	
Dense Layer		512	
Dropout	0.5	512	
Dense Layer		128	
Output		16	

TABLE IV
NUMBER OF THE PARAMETERS AND COMPRESSION RATE IN ALL CONVLSTM2D LAYERS WITH $TTC - rank = 8$

Model	Indian Pines			Salinas Valley			University of Pavia		
	Number	Rate	OA	Number	Rate	OA	Number	Rate	OA
SSCL2DNN	1443840	1	98.03	6524160	1	96.30	3409920	1	91.47
SSTTCL2DNN	253760	5.69	97.33	513280	12.71	95.65	258880	13.17	89.18
TACLNN	507522	2.84	98.25	517762	12.60	98.02	512002	6.66	93.01

For the TARB module, there are two key parameters, i.e., $\tilde{K}$ and $\tilde{t}$, and fivefold cross-validation is utilized to tune them from the range of $\{1, 5, 10, 15\}$ and $\{0.2, 0.4, 0.6, 0.8\}$, respectively. The experiments for studying their effects are given in Fig. 5(e)-(g). The quasi-optimal value of $[\tilde{K}, \tilde{t}]$ for three HSI data sets is respectively [5, 0.2], [10, 0.8], and [5, 0.2].

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT APPROACHES FOR THE INDIAN PINES DATA SET

Class	SVM	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSTTCL2DNN	TACLNN
1	70.73 ± 18.36	79.88 ± 11.09	91.06 ± 1.41	**100.00 ± 0.00**	98.37 ± 1.41	98.78 ± 1.41
2	89.32 ± 2.44	94.24 ± 0.68	94.29 ± 1.68	**98.11 ± 0.82**	97.02 ± 0.78	97.74 ± 1.04
3	91.93 ± 1.66	90.70 ± 1.05	93.13 ± 3.18	96.56 ± 1.71	96.74 ± 1.15	**97.79 ± 1.08**
4	86.50 ± 8.41	88.85 ± 3.73	88.89 ± 12.39	96.56 ± 3.52	95.62 ± 4.44	**97.65 ± 2.99**
5	90.57 ± 1.45	91.09 ± 1.38	94.25 ± 0.80	96.09 ± 2.22	96.32 ± 3.04	**96.90 ± 2.26**
6	97.72 ± 1.22	96.58 ± 1.43	**99.49 ± 0.23**	98.02 ± 0.93	98.83 ± 0.35	98.90 ± 0.71
7	58.00 ± 19.18	77.00 ± 19.87	**93.33 ± 11.55**	84.00 ± 8.00	92.00 ± 8.00	93.00 ± 11.49
8	98.72 ± 0.30	97.09 ± 1.30	99.46 ± 0.27	99.69 ± 0.36	98.84 ± 0.40	**99.94 ± 0.12**
9	44.44 ± 7.86	63.89 ± 19.44	38.89 ± 5.56	55.56 ± 9.62	**81.48 ± 11.56**	79.17 ± 13.13
10	77.20 ± 6.58	91.49 ± 1.90	95.73 ± 2.53	97.07 ± 2.07	95.62 ± 2.52	**97.34 ± 1.02**
11	95.02 ± 1.89	95.21 ± 2.10	96.60 ± 0.25	**99.34 ± 0.47**	98.43 ± 0.38	99.06 ± 0.56
12	84.27 ± 6.53	86.42 ± 3.16	87.39 ± 0.92	96.75 ± 1.68	**96.82 ± 2.76**	96.40 ± 1.36
13	89.13 ± 3.17	91.98 ± 3.96	95.65 ± 1.96	**98.19 ± 2.26**	91.67 ± 4.69	97.83 ± 3.07
14	97.41 ± 0.83	98.51 ± 1.13	**99.80 ± 0.18**	99.65 ± 0.26	98.18 ± 1.02	99.19 ± 0.58
15	93.73 ± 6.93	94.52 ± 3.42	97.12 ± 2.29	98.85 ± 0.50	97.98 ± 0.29	**99.28 ± 0.76**
16	68.75 ± 8.55	79.46 ± 10.86	89.29 ± 8.58	85.32 ± 5.99	92.46 ± 2.75	**94.64 ± 1.54**
OA	91.20 ± 2.01	93.69 ± 0.72	95.62 ± 0.26	98.03 ± 0.29	97.33 ± 0.46	**98.25 ± 0.43**
AA	83.34 ± 2.23	88.56 ± 2.88	90.90 ± 1.38	93.73 ± 0.37	95.40 ± 0.46	**96.48 ± 1.56**
κ	89.91 ± 2.32	92.79 ± 0.81	94.99 ± 0.30	97.75 ± 0.33	96.96 ± 0.52	**98.01 ± 0.49**

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT APPROACHES FOR THE SALINAS VALLEY DATA SET

Class	SVM	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSTTCL2DNN	TACLNN
1	74.26 ± 6.51	82.02 ± 7.09	93.08 ± 6.60	91.64 ± 10.83	94.27 ± 4.58	**98.12 ± 2.12**
2	85.23 ± 9.56	82.15 ± 4.89	**99.57 ± 0.16**	97.80 ± 2.21	96.82 ± 0.89	98.76 ± 0.65
3	59.56 ± 2.99	54.62 ± 4.25	98.99 ± 1.11	98.77 ± 1.55	99.10 ± 0.79	**98.93 ± 0.80**
4	98.19 ± 1.54	95.53 ± 2.00	98.84 ± 2.01	**99.40 ± 0.98**	98.21 ± 1.88	98.36 ± 1.88
5	94.33 ± 0.56	93.05 ± 0.86	99.41 ± 0.53	98.83 ± 0.21	**98.98 ± 0.26**	98.79 ± 1.64
6	94.40 ± 2.31	95.74 ± 2.05	99.76 ± 0.41	**100.00 ± 0.00**	99.98 ± 0.03	99.46 ± 0.91
7	78.97 ± 1.82	82.30 ± 5.41	98.15 ± 2.99	96.23 ± 1.34	97.97 ± 1.61	**99.76 ± 0.21**
8	85.36 ± 2.22	85.36 ± 1.63	90.05 ± 4.23	93.05 ± 4.98	89.15 ± 0.17	**95.52 ± 0.92**
9	68.44 ± 1.62	77.75 ± 3.22	99.69 ± 0.45	**99.89 ± 0.19**	99.64 ± 0.57	99.73 ± 0.29
10	71.65 ± 5.32	70.77 ± 8.53	**99.62 ± 0.34**	98.44 ± 0.92	97.90 ± 1.95	98.98 ± 1.10
11	93.98 ± 0.11	95.90 ± 2.15	**99.12 ± 0.27**	98.08 ± 1.31	97.82 ± 0.62	94.26 ± 2.08
12	88.38 ± 3.49	86.60 ± 10.28	97.43 ± 3.26	99.06 ± 0.82	**99.55 ± 0.42**	99.42 ± 0.87
13	80.04 ± 10.46	68.72 ± 5.55	**99.23 ± 1.34**	97.02 ± 2.61	97.61 ± 1.79	97.87 ± 2.00
14	84.04 ± 3.78	91.34 ± 4.51	**99.87 ± 0.14**	99.69 ± 0.11	99.24 ± 0.59	99.02 ± 1.69
15	90.36 ± 4.14	88.93 ± 2.18	89.46 ± 5.31	92.78 ± 7.57	94.14 ± 1.08	**97.10 ± 2.30**
16	64.92 ± 1.97	61.13 ± 5.73	97.34 ± 0.98	93.39 ± 5.41	90.20 ± 5.79	**99.12 ± 1.47**
OA	82.39 ± 1.21	83.10 ± 0.40	95.72 ± 0.70	96.30 ± 1.97	95.65 ± 0.32	**98.02 ± 0.09**
AA	82.01 ± 0.83	81.99 ± 0.78	97.48 ± 0.22	97.13 ± 1.51	96.91 ± 0.56	**98.33 ± 0.27**
κ	80.30 ± 1.32	81.15 ± 0.46	95.24 ± 0.78	95.88 ± 2.19	95.16 ± 0.36	**97.79 ± 0.10**

Finally, the optimal learning rate lr is searched from $\{0.0001, 0.0005, 0.001, 0.005\}$. From Fig. 5(h), the optimal value of lr for these three HSI data sets is 0.001.

The detailed parameter settings of TACLNN for the Indian Pines data set are reported in Table III. Particularly, for the Salinas Valley data set, the kernel sizes of two TT-ConvLSTM2D layers are respectively set to 4×4 and 5×5, while 4×4 and 4×4 for the University of Pavia data set, and the window sizes of the input data for these two data sets are obtained from Fig. 5(a)-(b). Other parameter settings for these two HSI data sets are the same as those in Table III.

C. Classification Performance

To verify the effectiveness of our TACLNN model, we randomly select 10% of samples to build the training set for the Indian Pines data set, while 1% for the other two HSI data sets. The rest of the samples are used for testing.

According to Tables IV–VII, TACLNN can recover the performance loss caused by the reduction of the number of the parameters in each ConvLSTM2D layer to obtain better classification performance. This also illustrates that TACLNN presents better feature representation ability for more effective spatial-spectral features. On the one hand, similar to the ConvLSTM2D layer, the special design of the gate structure makes it possible for the TT-ConvLSTM2D layer to better learn the spatial information and fuse the spatial-spectral information than the LSTM-based models. On the other hand, the way of modeling the long-range dependencies in the spectral field in the ConvLSTM2D-based models is more effective than simply cascading all the outputs in Bi-CLSTM, as verified in [22].

To make a fair comparison of SSCL2DNN, SSTTCL2DNN, and TACLNN, additional experiments for analyzing the number of parameters and the compression rate in all ConvLSTM2D layers are conducted, and the results are reported in Table IV. Compared with SSCL2DNN, SSTTCL2DNN leads to a small range of accuracy degradation after reducing the number of the parameters, achieving up to 5.69 ×, 12.71 ×, and 13.17 × compression with 0.70%, 0.65%, and 2.29% accuracy degradations for three HSI data sets, respectively, while to 2.84 ×, 12.60 ×, and 6.66 × compression with 0.22%, 1.72%, and 1.54% accuracy improvements for TACLNN. It is evident that, with the help of TARB, TACLNN can recover the performance loss

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT APPROACHES FOR THE UNIVERSITY OF PAVIA DATA SET

Class	SVM	SSLSTMs	Bi-CLSTM	SSCL2DNN	SSTTCL2DNN	TACLNN
1	52.21 ± 4.46	70.57 ± 3.33	88.24 ± 3.56	**96.15 ± 1.14**	88.67 ± 3.68	93.95 ± 0.82
2	89.78 ± 3.48	89.90 ± 0.72	98.18 ± 0.36	**99.43 ± 0.25**	97.40 ± 0.53	98.41 ± 0.49
3	17.73 ± 4.99	32.20 ± 4.30	37.91 ± 8.91	64.28 ± 7.56	**71.88 ± 8.56**	64.39 ± 6.55
4	55.23 ± 8.18	69.21 ± 4.91	80.39 ± 3.36	90.45 ± 2.12	83.16 ± 2.26	**92.19 ± 2.45**
5	68.05 ± 6.23	77.69 ± 6.33	80.56 ± 6.76	89.01 ± 5.01	88.84 ± 8.12	**96.07 ± 3.26**
6	39.14 ± 9.73	50.47 ± 3.16	66.77 ± 4.95	76.60 ± 5.21	80.10 ± 9.41	**93.87 ± 4.29**
7	15.19 ± 6.76	35.20 ± 5.04	56.09 ± 4.64	67.15 ± 7.68	72.67 ± 5.18	**73.45 ± 5.82**
8	60.56 ± 3.20	68.19 ± 3.12	82.22 ± 5.32	**92.58 ± 2.17**	85.27 ± 2.38	88.95 ± 2.01
9	53.39 ± 9.02	85.37 ± 9.22	**86.67 ± 8.64**	78.04 ± 5.48	76.05 ± 13.78	80.45 ± 3.02
OA	65.67 ± 1.83	73.90 ± 0.61	85.22 ± 0.75	91.47 ± 1.02	89.18 ± 0.66	**93.01 ± 0.70**
AA	50.14 ± 2.98	64.31 ± 0.91	75.22 ± 2.12	83.74 ± 2.04	82.67 ± 1.67	**86.86 ± 0.72**
κ	51.52 ± 3.24	64.42 ± 0.91	79.85 ± 1.09	88.49 ± 1.42	85.55 ± 0.96	**90.72 ± 0.95**

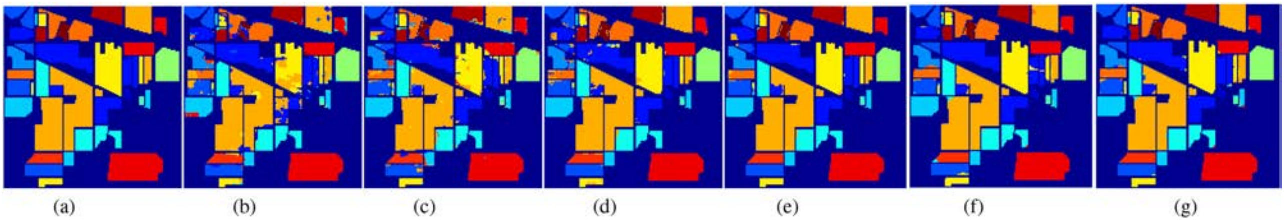


Fig. 6. Classification maps for the Indian Pines data set. (a) Ground-truth map. (b) SVM (91.20% ± 2.01%). (c) SSLSTMs (93.69% ± 0.72%). (d) Bi-CLSTM (95.62% ± 0.26%). (e) SSCL2DNN (98.03% ± 0.29%). (f) SSTTCL2DNN (97.33% ± 0.46%). (g) TACLNN (98.25% ± 0.43%).

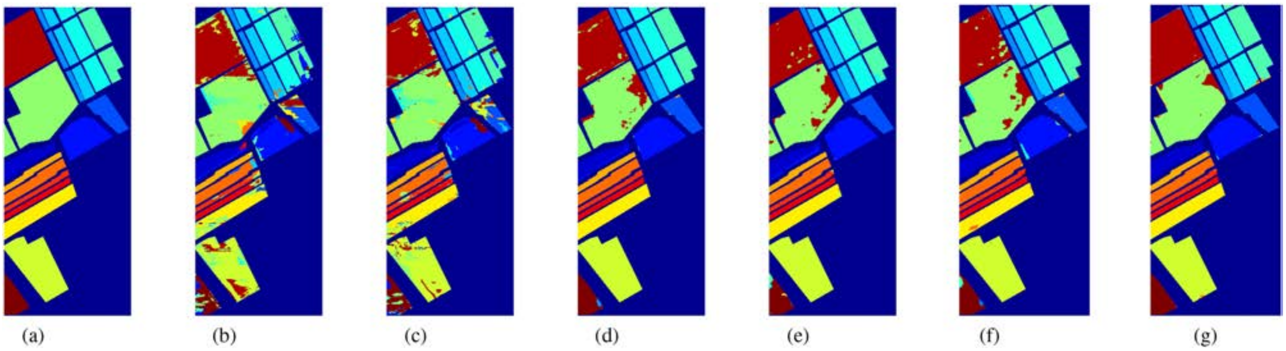


Fig. 7. Classification maps for the Salinas Valley data set. (a) Ground-truth map. (b) SVM (82.39% ± 1.21%). (c) SSLSTMs (83.10% ± 0.40%). (d) Bi-CLSTM (95.72% ± 0.70%). (e) SSCL2DNN (96.30% ± 1.97%). (f) SSTTCL2DNN (95.65% ± 0.32%). (g) TACLNN (98.02% ± 0.09%).

caused by the reduction of parameters in SSTTCL2DNN on the premise of adding only two parameters to be trained, and yield better classification performance than SSCL2DNN, which verifies the advantages of our TACLNN model. More detailed experimental results are reported in Tables V–VII.

From the classification maps in Figs. 6 –8, similar conclusions can also be drawn. It can be seen that the maps generated by TACLNN are improved for these three HSI data sets. Specifically, the maps contain less mislabeled pixels and the boundaries of different classes are better delineated, especially for class 3, class 15, and class 16 in Fig. 5, class 8, class 15, and class 16 in Fig. 6, and class 5 and class 6 in Fig. 7, which also demonstrates the superiority of our TACLNN model.

As is known to all, the cost of generating labeled samples is greatly high for HSI classification. To further investigate the performance of TACLNN with limited training samples, additional experiments are conducted. Specifically, 10, 20, 30, and 40 samples of each class are randomly selected from these three HSI data sets, and for class 7 and class 9 in the Indian Pines data set, the number of the samples is fixed to 10. Based on the above settings, the OA curves of all models are shown in Fig. 9, where even in the case of small training samples, TACLNN can also achieve better classification performance than other algorithms. Furthermore, compared with other three ConvLSTM2D-based methods, the classification accuracy of TACLNN can be effectively improved by the TARB module. TARB can effectively enhance its feature extraction ability and the intrinsic structure of the extracted spatial-spectral features, which is the main reason why the performance loss of SSTTCL2DNN can be

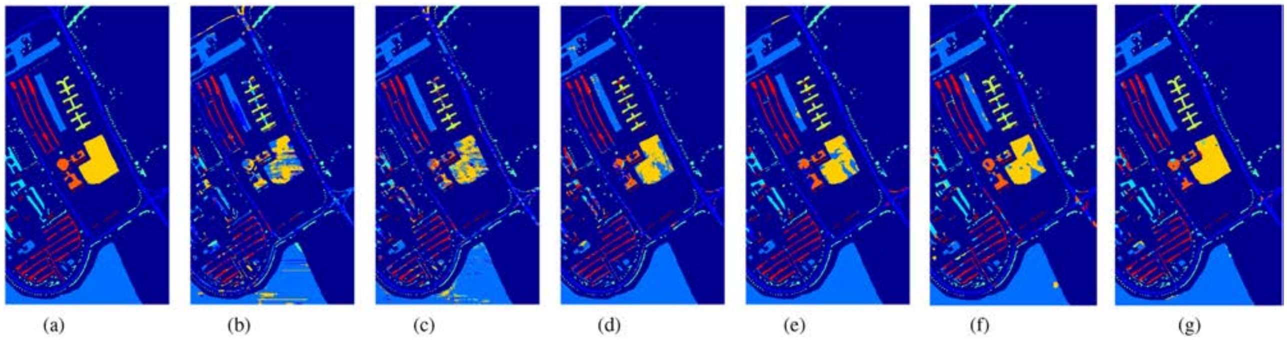


Fig. 8. Classification maps for the University of Pavia data set. (a) Ground-truth map. (b) SVM (65.67% ± 1.83%). (c) SSLSTMs (73.90% ± 0.61%). (d) Bi-CLSTM (85.22% ± 0.75%). (e) SSCL2DNN (91.47% ± 1.02%). (f) SSTTCL2DNN (89.18% ± 0.66%). (g) TACLNN (93.01% ± 0.70%).

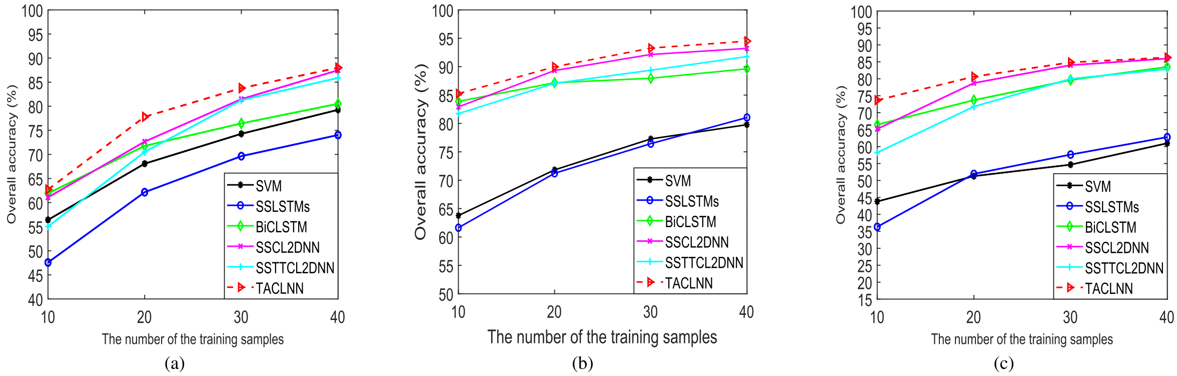


Fig. 9. OA (%) of all classification models under different number of the training samples. (a) Indian Pines, (b) Salinas Valley, (c) University of Pavia.

effectively recovered. This further confirms the effectiveness of the proposed TACLNN model.

D. Visualization of Tensor Attention Module

To compensate for the performance loss caused by the reduction of parameters in SSTTCL2DNN, a TARB module is designed by combining tensor representation of HSI data and attention mechanism. However, considering that the extracted tensor attention weight in TARB presents a fourth-order tensor, it is difficult to give its comprehensible visualization directly.

Therefore, we first extract a channel map from the outputs of our TARB module, leading to a third-order tensor attention map. Then, we show some attended spatial attention maps along with different spectral dimensions to see whether they highlight clear semantic areas. Fig. 10 visualizes some examples of the tensor attention maps from TARB for three HSI data sets, where the spectral dimension is 10. Specifically, Fig. 10(a), (c) and (e) show some examples of the input X_S in SSTTCL2DNN, while Fig. 10(b), (d), and (f) illustrate the corresponding tensor attention maps in TARB. From the perspective of each spectral dimension, the spatial attention can capture relatively clear semantic similarity and the boundary information can be described clearly, while the response of the specific semantics after applying the spectral attention is noticeable in the spatial dimension. This means that TARB can highlight the response of specific semantic area and capture its boundary information to enhance the ability of modeling the spatial-spectral features of TACLNN for HSI classification.

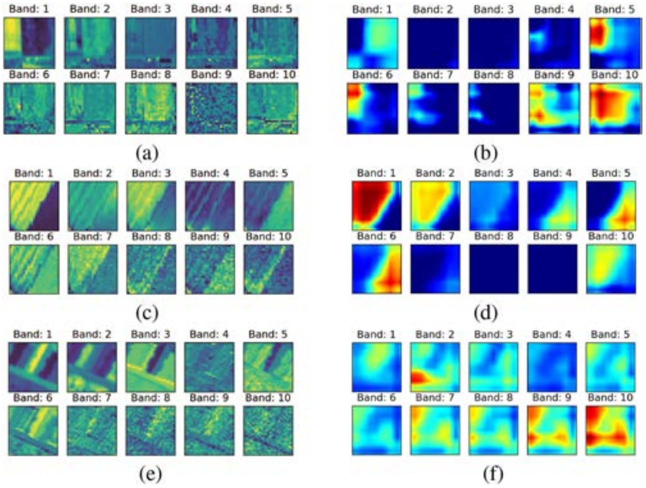


Fig. 10. Visualization results of the TARB module on these three HSI data sets. (a)-(b) Indian Pines, (c)-(d) Salinas Valley, (e)-(f) University of Pavia.

V. Conclusion

In this paper, a new TACLNN model has been developed for the feature extraction and classification of HSIs. Particularly, a lightweight TT-ConvLSTM2D cell is constructed by applying TTD, with which an SSTTCL2DNN model is further developed.

To recover the performance loss caused by parameter reduction of each TT-ConvLSTM2D layer in the SSTTCL2DNN, a trainable TARB module is designed by combining the tensor representation of HSI data and attention mechanism, which can enhance the geometry structure information. Extensive experiments have been conducted on three commonly used HSI data sets, indicating that our TACLNN model can not only effectively reduce the amount of parameters to be trained, but also improve the feature extraction ability to yield more satisfactory classification performance.

References

[1] 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.

[2] 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.

[3] 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. Top. Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4117–4128, Sep. 2016.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] H. Li, W. Wang, L. Pan, W. Li, Q. Du, and R. Tao, "Robust capsule network based on maximum correntropy criterion for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 738–751, 2020.

[12] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "HSI-CNN: A novel convolution neural network for hyperspectral image," in *Proc. Int. Conf. Audio Lang. Image Process.*, 2018, pp. 464–469.

[13] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[14] 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.

[15] 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.

[16] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.

[17] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral-spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.

[18] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS-J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.

[19] A. Graves, M. Liwicki, S. Fernández, 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.

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

[21] 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. Conf. Neural Inf. Process. Syst.*, Montréal, 2015, pp. 802–810.

[22] W. Hu, H. Li, L. Pan, W. Li, R. Tao, and Q. Du, "Spatial-spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4237–4250, Jun. 2020.

[23] 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.

[24] C. Shi and C. -M. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, pp. 82–93, Jun. 2018.

[25] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[26] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[27] D. Wang, B. Du, L. Zhang, and Y. Xu, "Adaptive spectral-spatial multiscale contextual feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2461–2477, Mar. 2021.

[28] 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.

[29] 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, Dec. 2017, Art. no. 1330.

[30] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 996–1010, Apr. 2013.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[32] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6298–6306.

[33] H. Guo, K. Zhu, M. Tang, and J. Wang, "Two-level attention network with multi-grain ranking loss for vehicle re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4328–4338, Sep. 2019.

[34] J. Zhang, Y. Xie, Y. Xia, and C. Shen, "Attention residual learning for skin lesion classification," *IEEE Trans. Med. Imag.*, vol. 38, no. 9, pp. 2092–2103, Sep. 2019.

[35] Z. Cui, Q. Li, Z. Cao, and N. Liu, "Dense attention pyramid networks for multi-scale ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8983–8997, Nov. 2019.

[36] J. Chen, L. Wan, J. Zhu, G. Xu, and M. Deng, "Multi-scale spatial and channel-wise attention for improving object detection in remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 681–685, Apr. 2020.

[37] Z. Xiong, Y. Yuan, and Q. Wang, "AI-NET: Attention inception neural network for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 2647–2650.

[38] H. Dong, L. Zhang, and B. Zou, "Band attention convolutional networks for hyperspectral image classification," 2019, *arXiv:1906.04379*.

[39] X. Mei *et al.*, "Spectral-spatial attention networks for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 8, pp. 963-1-18, Apr. 2019.

[40] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, Jun. 2019, Art. no. 1307.

[41] B. Fang, Y. Li, H. Zhang, and J. C. -W. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, no. 2, pp. 159-1-18, Jan. 2019.

[42] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.

[43] A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, "Tensorizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 442–450.

[44] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[45] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for matlab," in *Proc. 23 rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[46] T. Garipov, D. Podoprikhin, A. Novikov, and D. Vetrov, "Ultimate tensorization: Compressing convolutional and FC layers alike," 2016. *arXiv:1611.03214.*.

[47] A. Tjandra, S. Sakti, and S. Nakamura, "Compressing recurrent neural network with tensor train," in *Proc. Int. Joint Conf. Neural Netw.*, Anchorage, AK, USA, 2017, pp. 4451–4458.

[48] Y. Yang, D. Krompass, and V. Tresp, "Tensor-train recurrent neural networks for video classification," in *Proc. Int. Conf. Mach. Learn.*, Sydney, Australia, 2017, pp. 3891–3900.

[49] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Conf. Neural Inf. Process. Syst.*, Long Beach, 2017, pp. 6000–6010.

[50] Y. Deng, H. Li, L. Pan, and W. J. Emery, "Tensor locality preserving projection for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 771–774.

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

[52] W. Hu, H. Li, T. Ma, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral image classification based on tensor-train convolutional long-short memory," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 858–861.

Wen-Shuai Hu received the B.Sc. degree in electronic information science and technology from Shanxi Agricultural University, Jinzhong, China, in 2016. He is currently working toward the Ph.D. degree in information and communication engineering with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China.

His research interests include remote sensing image processing and neural networks.

Heng-Chao Li (Senior Member, IEEE) received the B.Sc. degree in information and communication engineering and the M.Sc. degree in information and communication engineering from Southwest Jiaotong University, Chengdu, China, in 2001 and 2004, respectively, and the Ph.D. degree in information and communication engineering from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently a Full Professor with the School of Information Science and Technology, Southwest Jiaotong University. His research interests include statistical analysis of SAR images, remote sensing image processing, and pattern recognition.

Yang-Jun Deng received the Ph.D. degree in signal and information processing from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2020.

He is currently an Assistant Professor with the College of Information and Intelligence, Hunan Agricultural University, Changsha, China. His research interests include pattern recognition and remote sensing image processing.

Xian Sun (Senior Member, IEEE) received the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2006 and 2009, respectively.

His research interests include computer vision, geospatial data mining, and remote sensing image understanding.

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

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

Antonio Plaza (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cȁceres, Spain, in 1999 and 2002, respectively.

He is currently the Head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. His research interests include hyperspectral data processing and parallel computing of remote sensing data.