Cascaded Recurrent Neural Networks for Hyperspectral Image Classification

Hang, R.; Liu, Q.; Hong, D.; Ghamisi, P.;

Originally published:

August 2019

IEEE Transactions on Geoscience and Remote Sensing 57(2019)8, 5384-5394

DOI: https://doi.org/10.1109/TGRS.2019.2899129

Perma-Link to Publication Repository of HZDR:

https://www.hzdr.de/publications/Publ-30061

Cascaded Recurrent Neural Networks for Hyperspectral Image Classification

Renlong Hang, *Member, IEEE*, Qingshan Liu, *Senior Member, IEEE*, Danfeng Hong, *Student Member, IEEE*, and Pedram Ghamisi, *Senior Member, IEEE*

***Abstract*— By considering the spectral signature as a sequence, recurrent neural networks (RNNs) have been successfully used to learn discriminative features from hyperspectral images (HSIs) recently. However, most of these models only input the whole spectral bands into RNNs directly, which may not fully explore the specific properties of HSIs. In this paper, we propose a cascaded RNN model using gated recurrent units to explore the redundant and complementary information of HSIs. It mainly consists of two RNN layers. The first RNN layer is used to eliminate redundant information between adjacent spectral bands, while the second RNN layer aims to learn the complementary information from nonadjacent spectral bands. To improve the discriminative ability of the learned features, we design two strategies for the proposed model. Besides, considering the rich spatial information contained in HSIs, we further extend the proposed model to its spectral–spatial counterpart by incorporating some convolutional layers. To test the effectiveness of our proposed models, we conduct experiments on two widely used HSIs. The experimental results show that our proposed models can achieve better results than the compared models.**

***Index Terms*— Gated recurrent unit (GRU), hyperspectral image (HSI) classification, recurrent neural network (RNN), spectral feature, spectral–spatial feature.**

I. Introduction

With the development of imaging technology, current hyperspectral sensors can fully portray the surface of the earth using hundreds of continuous and narrow spectral bands, ranging from the visible spectrum to the short-wave infrared spectrum. The generated hyperspectral image (HSI) is often considered as a 3-D cube. The first two are spatial dimensions, which record the locations of each object. The third one is spectral dimension, which captures the spectral signature (reflective or emissive properties) of each material in different bands along the electromagnetic spectrum [1]. Using such rich information, HSIs have been widely applied to various applications, such as land cover/land use classification, precision agriculture, and change detection. For these applications, one basic but important procedure is HSI classification, whose goal is to assign candidate class labels to each pixel.

In order to acquire accurate classification results, numerous methods have been proposed. For example, one can directly consider the rich spectral signature as features and feed them into advanced classifiers, such as support vector machine (SVM) [2], random forest [3], and extreme learning machine [4]. However, due to the dense spectral sampling of HSIs, there may exist some redundant information among adjacent spectral bands. This easily leads to the so-called curse of dimensionality (the Hughes effect) which causes a sudden drop in classification accuracy when there is no balance between the high number of spectral channels and a limited number of training samples. Therefore, a large number of works were proposed to mine discriminative features from the high-dimensional spectral signature [5]. Popular models include principle component analysis, linear discriminant analysis (LDA) [6]–[8], and graph embedding [9]–[11]. Besides, representation-based models have also been employed to HSI classification in recent years. In [12] and [13], sparse representation was proposed to learn discriminative features from HSIs. Similarly, collaborative representation was also widely explored [14], [15]. In these models, an input spectral signature is usually represented by a linear combination of atoms from a dictionary, and the classification result can be derived from the reconstructed residual without needing to train extra classifiers, which often costs much time.

Although the aforementioned models have demonstrated their effectiveness in the field of HSI classification, there still exist some drawbacks to address. For the traditional feature extraction models, we need to predefine a mining criterion (e.g., maximizing the between-class scatter matrix in LDA), which heavily depends on the domain knowledge and experience of experts. For the representation-based models, their goal is to reconstruct the input signal, leading to suboptimal representation for classification. In addition, all of them can be considered as shallow-layer models, which limit their potentials to learn high-level semantic features. Recently, deep learning [16], [17], a very hot research topic in machine

Manuscript received December 20, 2018; revised January 21, 2019; accepted February 4, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61532009 and Grant 61825601 and in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20180786. *(Corresponding author: Qingshan Liu.)*

R. Hang and Q. Liu are with the Jiangsu Key Laboratory of Big Data Analysis Technology, School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: renlong_hang@163.com; qsliu@nuist.edu.cn).

D. Hong is with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Wessling, Germany, and also with Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: danfeng.hong@dlr.de).

P. Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, D-09599 Freiberg, Germany (e-mail: p.ghamisi@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2899129

learning, has shown its huge superiority in most fields of computer vision [18]–[21] and natural language processing [22], [23]. The goal of deep learning is to learn nonlinear, high-level semantic features from data in a hierarchical manner.

Due to the effects of multipath scattering and the heterogeneity of subpixel constituents, HSI often lies in a nonlinear and complex feature space. Deep learning can be naturally adopted to deal with this issue [24], [25]. In the past few years, many deep learning models were successfully applied to HSI classification. For example, in [26]–[28], the autoencoder model has been used to learn deep features from high-dimensional spectral signature directly. Similar to autoencoder, deep belief network was also explored to extract spectral features [29]–[31]. However, both of them belong to fully connected networks, which contain large numbers of parameters to train. Different from them, convolutional neural networks (CNNs) have local connection and weight sharing properties, thus largely reducing the number of training parameters [32]–[34]. Hu *et al*. [35] proposed to use 1-D CNN to learn and represent the spectral information. This model is comprised of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. The whole model is trained in an end-to-end manner, thus achieving satisfying results for HSI classification.

Besides spectral information, HSIs also have rich spatial information. How to combine them together has been an active research topic in the field of HSI classification [36], [37]. One potential method is to extend the spectral classification model into its spectral–spatial counterpart. For instance, in [38]–[40], a 3-D CNN was employed to spectral–spatial classification of HSIs. However, due to the simultaneous convolution operators in both spectral domain and spatial domain, the computational complexity is dramatically increased. In addition, the number of trainable parameters in 3-D CNNs is also a problem. In order to perform 3-D convolution, the dimensionality of the input and the dimensionality of the kernel (filter) should be equal. This heavily increases the number of parameters. Another candidate method for spectral–spatial classification is the one based on two branch networks. One branch is for spectral classification and the other one for spatial classification. In [41]–[43], 1-D CNN or autoencoder was used to learn spectral features and 2-D CNN was designed to learn spatial features. These two features are then integrated together via feature-level fusion or decision-level fusion. For 2-D CNNs, only a few principal components were extracted and used as inputs, thus reducing the computational consuming compared to 3-D CNNs.

Most of the existing models can be considered as vector-based methodologies. Recently, a few works attempted to regard HSIs as sequential data, so recurrent neural networks (RNNs) were naturally used to learn features. Wu and Prasad *et.al* proposed using RNN to extract spectral features from HSIs. In [45] and [46], a variant of RNN using long short-term memory (LSTM) units was designed to learn spectral–spatial features from HSIs. In [47], another variant of RNN using gated recurrent units (GRUs) was employed. Compared to the widely explored CNN models, RNNs have many superiorities. For example, the key component of CNNs is the convolutional operator. Due to the kernel size limitations of it, 1-D CNNs can only learn the local spectral dependency, while easily ignoring the effects of nonadjacent spectral bands. Different from them, RNNs, especially using GRU or LSTM, often input spectral bands one by one via recurrent operators, thus capturing the relationship from the whole spectral bands. Besides, RNNs often have smaller numbers of parameters to train than CNNs, so they will be more efficient in the training and inferring phases.

Benefiting from its powerful learning ability from sequential data, current RNN-related models often simply input the whole spectral bands to networks, which may not fully explore the redundant and complementary properties of HSIs. The redundant information between adjacent spectral bands will increase the computational burden of RNNs without improving the classification results. Sometimes such redundancy may reduce the classification accuracy since it increases within-class variances and decreases between-class variances in the feature space. Besides, it may also increase the difficulties in learning complementary information. To address these issues, we propose a cascaded RNN model using GRUs in this paper. This model mainly consists of two RNN layers. The first RNN layer focuses on reducing the redundant information of adjacent spectral bands. These reduced informations are then fed into the second RNN layer to learn their complementary features. Besides, in order to improve the discriminative ability of the learned features, we design two strategies for the proposed model. Finally, we also extend the proposed model to its spectral–spatial version by incorporating some convolutional layers. The major contributions of this paper are summarized as follows.

1) We propose a cascaded RNN model with GRUs for HSI classification. Compared to the existing RNN-related models, our model can sufficiently consider the redundant and complementary information of HSIs via two RNN layers. The first one is to reduce redundancy and the second one is to learn complementarity. These two layers are integrated together to generate an end-to-end trainable model.
2) In order to learn more discriminative features, we design two strategies to construct connections between the first RNN layer and the output layer. The first strategy is the weighted fusion of features from two layers, and the second one is the weighted combination of different loss functions from two layers. Their weights can be adaptively learned from data itself.
3) To capture the spectral and spatial features simultaneously, we further extend the proposed model to its spectral–spatial counterpart. A few convolutional layers are integrated into the proposed model to learn spatial features from each band, and these features are then combined together via recurrent operators.

The rest of this paper is organized as follows. Section II describes the details of the proposed models, including a brief introduction of RNN, and the structure of the proposed model as well as its modifications. The descriptions of data sets and experimental results are given in Section III. Finally, Section IV concludes this paper.

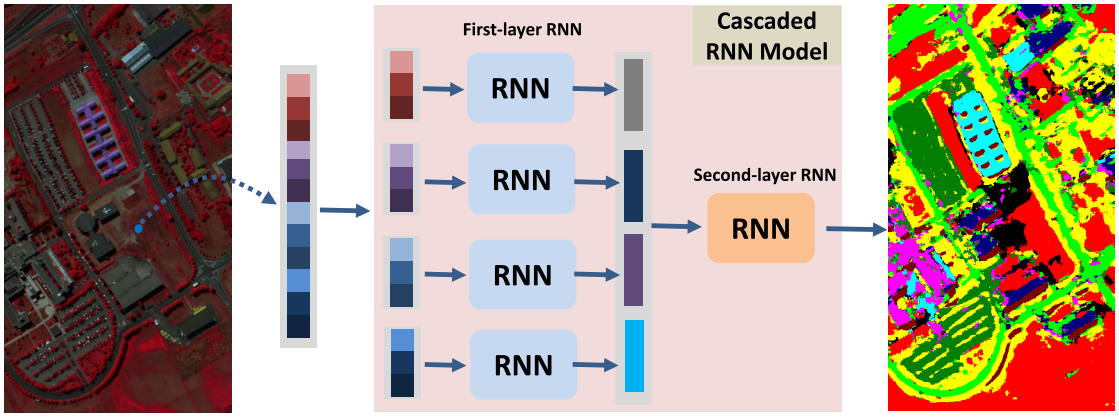


Fig. 1. Flowchart of the proposed model.

II. Methodology

As shown in Fig. 1, the proposed cascaded RNN model mainly consists of four steps. For a given pixel, we first divide it into different spectral groups. Then, for each group, we consider the spectral bands in it as a sequence, which is fed into an RNN layer to learn features. After that the learned features from each group are again regraded as a sequence and fed into another RNN layer to learn their complementary information. Finally, the output of the second RNN layer is connected to a softmax layer to derive the classification result.

A. Review of RNN

RNN has widely been used for sequential data analysis, such as speech recognition and machine translation [23], [48]. Assume that we have a sequence data $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t, t \in \{1, 2, \ldots, T\}$ generally represents the information at the tth time step. When applying RNN to HSI classification, $\mathbf{x}_t$ will correspond to the spectral value at the tth band. For RNN, the output of hidden layer at time t is

$$\mathbf{h}_t = \phi(\mathbf{W}_{hi}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{1}$$

where ϕ is a nonlinear activation function such as logistic sigmoid or hyperbolic tangent functions, $\mathbf{b}_h$ is a bias vector, $\mathbf{h}_{t-1}$ is the output of hidden layer at the previous time, and $\mathbf{W}_{hi}$ and $\mathbf{W}_{hh}$ denote weight matrices from the current input layer to hidden layer and the previous hidden layer to current hidden layer, respectively. From (1), we can observe that via a recurrent connection, the contextual relationships in the time domain can be constructed. Ideally, $\mathbf{h}_T$ can capture most of the time information for the sequence data.

For classification tasks, $\mathbf{h}_T$ is often fed into an output layer, and the probability that the sequence belongs to ith class can be derived by using a softmax function. These processes can be formulated as

$$\begin{aligned} \mathbf{O}_T &= \mathbf{W}_{oh}\mathbf{h}_T + \mathbf{b}_o \\ P(\tilde{y} = i\,|\boldsymbol{\theta}, \mathbf{b}) &= \frac{e^{\boldsymbol{\theta}_i \mathbf{O}_T + b_i}}{\sum_{j=1}^{C} e^{\boldsymbol{\theta}_j \mathbf{O}_T + b_j}} \end{aligned} \tag{2}$$

where $\mathbf{b}_o$ is a bias vector, $\mathbf{W}_{oh}$ is the weight matrix from hidden layer to output layer, $\boldsymbol{\theta}$ and $\mathbf{b}$ are parameters of softmax function, and C is the number of classes to discriminate. All of these weight parameters in (1) and (2) can be trained using the following loss function:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\tilde{y}_i) + (1 - y_i)\log(1 - \tilde{y}_i)] \tag{3}$$

where N is the number of training samples and y_i and $\tilde{y}_i$ are the true label and the predicted label of the ith training sample, respectively. This function can be optimized using a backpropagation through time (BPTT) algorithm.

B. Cascaded RNNs

HSIs can be described as a 3-D matrix $\mathbf{X} \in \Re^{m \times n \times k}$, where m, n, and k represent the width, height, and number of spectral bands, respectively. For a given pixel $\mathbf{x} \in \Re^k$, we can consider it as a sequence whose length is k, so RNN can be naturally employed to learn spectral features. However, HSIs often contain hundreds of bands, making $\mathbf{x}$ a very long sequence. Such long-term sequence increases the training difficulty since the gradients tend to either vanish or explode [49]. To address this issue, one popularly used method is to design a more sophisticated activation function by using gating units such as the LSTM unit and GRU [50]. Compared to LSTM unit, GRU has a fewer number of parameters [49], which may be more suitable for HSI classification because it usually has a limited number of training samples. Therefore, we select GRU as the basic unit of RNN in this paper.

The core components of GRU are two gating units that control the flow of information inside the unit. Instead of using (1), the activation of the hidden layer for band t is now formulated as

$$\mathbf{h}_t = (1 - u_t)\mathbf{h}_{t-1} + u_t\tilde{\mathbf{h}}_t \tag{4}$$

where u_t is the update gate, which can be derived by

$$u_t = \sigma(w_u x_t + \mathbf{v}_u \mathbf{h}_{t-1}) \tag{5}$$

where σ is a sigmoid function, w_u is a weight value, and $\mathbf{v}_u$ is a weight vector. Similarly, $\tilde{\mathbf{h}}_t$ can be computed by

$$\tilde{\mathbf{h}}_t = tanh(\mathbf{w}x_t + \mathbf{V}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{6}$$

where $\odot$ denotes an element-wise multiplication and $\mathbf{r}_t$ is the reset gate, which can be derived by

$$\mathbf{r}_t = \sigma(\mathbf{w}_r x_t + \mathbf{V}_r \mathbf{h}_{t-1}). \tag{7}$$

Due to the dense spectral sampling of hyperspectral sensors, adjacent bands in HSIs have some redundancy, while nonadjacent bands have some complementarity. In order to take account of such information comprehensively, we propose a cascaded RNN model. Specifically, we divide the spectral sequence $\mathbf{x}$ into l subsequences $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_l)$, each of which consists of adjacent spectral bands. Besides the last subsequence $\mathbf{z}_l$, the length of the other subsequences is $d = \text{floor}(k/l)$, which denotes the nearest integers less than or equal to k/l. Thus, for the ith subsequence $\mathbf{z}_i, i \in \{1, 2, \ldots, l\}$, it is comprised of the following bands:

$$\mathbf{z}_i = \begin{cases} (x_{(i-1)\times d+1}, \ldots, x_{i\times d}), & \text{if } i \neq l, \\ (x_{(i-1)\times d+1}, \ldots, x_k), & \text{otherwise.} \end{cases} \tag{8}$$

Then, we feed all the subsequences into the first-layer RNNs, respectively. These RNNs have the same structure and share parameters, thus reducing the number of parameters to train. In the subsequence $\mathbf{z}_i$, each band has an output from GRU. We use the output of the last band as the final feature representation for $\mathbf{z}_i$, which can be denoted as $\mathbf{F}_i^{(1)} \in \Re^{H_1}$, where H_1 is the size of the hidden layer in the first-layer RNN. After that we can combine $\mathbf{F}_i^{(1)}, i \in \{1, 2, \ldots, l\}$ together to generate another sequence $\mathbf{F} = (\mathbf{F}_1^{(1)}, \mathbf{F}_2^{(1)}, \ldots, \mathbf{F}_l^{(1)})$ whose length is l. This sequence is fed into the second-layer RNN to learn their complementary information. Similar to the first-layer RNNs, we also use the output of GRU at the last time l as the learned feature $\mathbf{F}^{(2)}$. To get a classification result of $\mathbf{x}$, we need to input $\mathbf{F}^{(2)}$ into an output layer whose size equals to the number of candidate classes C. Both of these two-layer RNNs have many weight parameters. We choose (3) as a loss function and use the BPTT algorithm to optimize them simultaneously.

C. Improvement for Cascaded RNNs

As described in Section II-B, the second-layer RNN is directly connected to the output layer, so it may be optimized better than the first-layer RNNs. However, the performance of the first-layer RNNs will have effects on the second-layer RNN. In order to improve the discriminative ability of $\mathbf{F}^{(2)}$, an intuitive method is to construct relations between the first-layer RNNs and the output layer. Here, we propose two strategies to achieve this goal. The first strategy is based on the feature-level connection shown in Fig. 2. Instead of feeding the output of the second-layer RNN into the output layer only, we attempt to feed all the output features from the first- and the second-layer RNNs in a weighted concatenation manner. Specifically, the input of the output layer is computed as follows:

$$\tilde{\mathbf{F}} = \left[w_1^{(1)}\mathbf{F}_1^{(1)}, w_2^{(1)}\mathbf{F}_2^{(1)}, \ldots, w_l^{(1)}\mathbf{F}_l^{(1)}, w^{(2)}\mathbf{F}^{(2)}\right] \tag{9}$$

where $w_i^{(1)} \in \Re^1 and i \in \{1, 2, \ldots, l\}$ are fusion weights for the first-layer RNNs and $w^{(2)} \in \Re^1$ is the fusion weight for

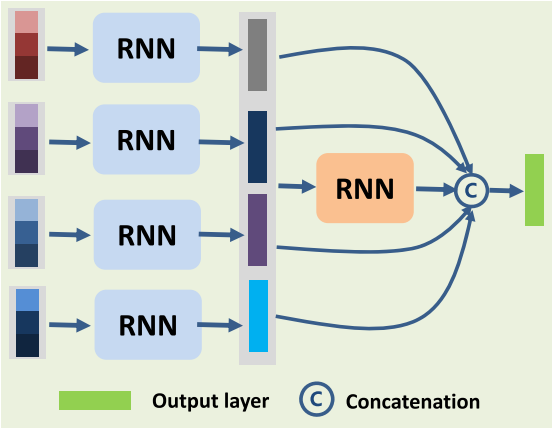


Fig. 2. First improvement strategy.

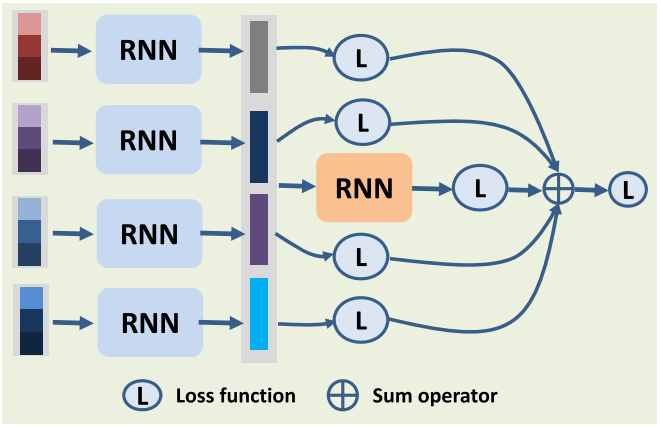


Fig. 3. Second improvement strategy.

the second-layer RNN. These weights can be integrated into the whole network and their optimal values are automatically learned from data. The same as the original two-layer RNN model, we also use (3) to construct the loss function and use the BPTT algorithm to optimize it.

Different from the first improvement strategy, our second strategy is based on the output-level connection. As shown in Fig. 3, we feed the features extracted by the first-layer RNNs into output layers, respectively, so that they can learn more discriminative features. Combining these features together using the second-layer RNN will result in a better $\mathbf{F}^{(2)}$. In particular, for $\mathbf{F}_i^{(1)}, i \in \{1, 2, \ldots, l\}$, we can input it into an output layer and construct a loss function $L_i^{(1)}, i \in \{1, 2, \ldots, l\}$. Meanwhile, we also input $\mathbf{F}^{(2)}$ into an output layer and construct another loss function $L^{(2)}$. After that a weighted summation method can be used to combine them together, which can be formulated as

$$\tilde{L} = \frac{1}{l}\sum_{i=1}^{l} w_i^{(1)} L_i^{(1)} + w^{(2)} L^{(2)} \tag{10}$$

where $w_i^{(1)} \in \Re^1$ and $w^{(2)} \in \Re^1$ are fusion weights and $L_i^{(1)}$ and $L^{(2)}$ are derived from (3). The final loss function $\tilde{L}$ can be optimized by using the BPTT algorithm. In the prediction phase, we can delete the output layers of the first-layer RNNs

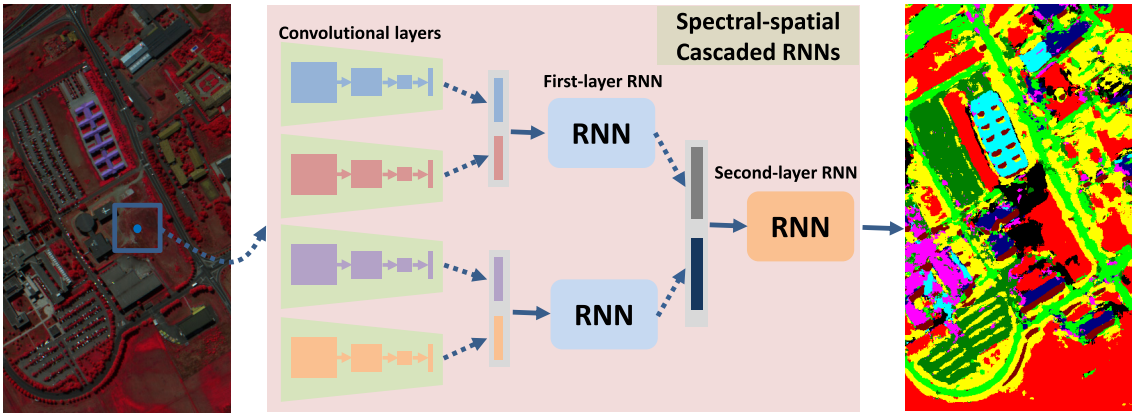


Fig. 4. Flowchart of spectral–spatial cascaded RNN model.

and use the output from the second-layer RNN as the final classification result.

D. Spectral–Spatial Cascaded RNNs

Due to the effects of atmosphere, instrument noises, and natural spectrum variations, materials from the same class may have very different spectral responses, while those from different classes may have similar spectral responses. If we only use the spectral information, the resulting classification maps will have many outliers, which is known as the "salt and pepper" phenomenon. As a 3-D cube, HSIs also have rich spatial information, which can be used as a complement to address this issue. Among numerous deep learning models, CNNs have demonstrated their superiority in spatial feature extraction. In [38], a typical 2-D CNN is designed to extract spatial features from HSIs. The input of this model is the first principle component of HSIs.

Inspired from the 2-D CNN model, we extend the cascaded RNN model to its spectral–spatial version by adding some convolutional layers. Fig. 4 shows the flowchart of the proposed spectral–spatial cascaded RNN model. For a given pixel $\mathbf{x} \in \Re^k$, we select a small cube $\hat{\mathbf{x}} \in \Re^{\omega \times \omega \times k}$ centered at it. Then, we split this cube into k matrices $\hat{\mathbf{x}}_i \in \Re^{\omega \times \omega}, i \in \{1, 2, \ldots, k\}$ across the spectral domain. For each $\hat{\mathbf{x}}_i$, we feed it into several convolutional layers to learn spatial features. The same as [38], we also use three convolutional layers, and the first two layers are followed by pooling layers. The input size $\omega \times \omega$ is 27×27. The sizes of the three convolutional filters are $4 \times 4 \times 32$, $5 \times 5 \times 64$, and $4 \times 4 \times 128$, respectively. After these convolutional operators, each $\hat{\mathbf{x}}_i$ will generate a 128-D spatial feature $\mathbf{s}_i$. Similar to the cascaded RNN model, we can also consider $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k)$ as a sequence whose length is k. This sequence is divided into l subsequences, and they are subsequently fed into the first-layer RNNs, respectively, to reduce redundancy inside each subsequence. The outputs from the first-layer RNNs are combined again to generate another sequence, which are fed into the second-layer RNN to learn complementary information.

Compared to the cascaded RNN model, the spectral–spatial cascaded RNN model is deeper and more difficult to train. Therefore, we propose a transfer learning method to train it.

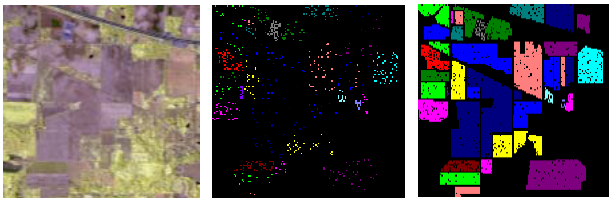


Fig. 5. Visualization of the Indian Pines data. (a) False-color image. (b) Training data map. (c) Test data map.

Specifically, we first pretrain the convolutional layers using all of $\hat{\mathbf{x}}_i, i \in \{1, 2, \ldots, k\}$. We replace two-layer RNNs by an output layer whose size is the number of classes C. Besides, we assume that the label of $\hat{\mathbf{x}}_i$ equals to the label of its corresponding pixel $\mathbf{x}$. Then, we will have $N \times k$ samples. These samples are used to train convolutional layers. After that the weights of these convolutional layers are fixed and the N training samples are used again to train the two-layer RNNs. Finally, the whole network is fine-tuned based on the learned parameters.

III. Experiments

A. Data Description

Our experiments are conducted on two HSIs, which are widely used to evaluate classification algorithms.

1) Indian Pines Data: The first data set was acquired by the AVIRIS sensor over the Indian Pine test site in northwestern Indiana, USA, on June 12, 1992. The original data set contains 224 spectral bands. We utilize 200 of them after removing four bands containing zero values and 20 noisy bands affected by water absorption. The spatial size of the image is 145×145 pixels and the spatial resolution is 20 m. The number of training and test pixels are reported in Table I. Fig. 5 shows the false-color image, as well as training and test maps of this data set.

2) Pavia University Scene Data: The second data set was acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy, on July 8, 2002. The original image was recorded with 115 spectral channels ranging from 0.43 to 0.86 μm. After removing noisy bands, 103 bands are used.

TABLE I
NUMBERS OF TRAINING AND TEST PIXELS USED IN THE INDIAN PINES DATA SET

Class No.	Class Name	Training	Test
1	Corn-notill	50	1384
2	Corn-mintill	50	784
3	Corn	50	184
4	Grass-pasture	50	447
5	Grass-trees	50	697
6	Hay-windrowed	50	439
7	Soybean-notill	50	918
8	Soybean-mintill	50	2418
9	Soybean-clean	50	564
10	Wheat	50	162
11	Woods	50	1244
12	Building-grass-trees	50	330
13	Stone-steel-towers	50	45
14	Alfalfa	15	39
15	Grass-pasture-mowed	15	11
16	Oats	15	5
-	Total	695	9671

TABLE II
NUMBERS OF TRAINING AND TEST PIXELS USED IN THE PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training	Test
1	Asphalt	548	6631
2	Meadows	540	18649
3	Gravel	392	2099
4	Trees	524	3064
5	Metal sheets	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Bricks	514	3682
9	Shadows	231	947
-	Total	3921	42776

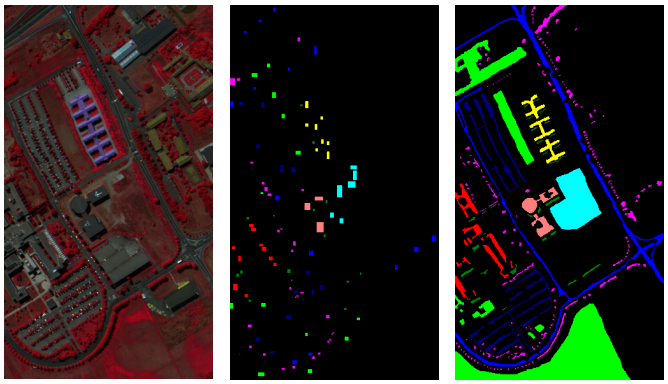


Fig. 6. Visualization of the Pavia University data. (a) False-color image. (b) Training data map. (c) Test data map.

The image size is 610×340 pixels with a spatial resolution of 1.3 m. There are nine classes of land covers with more than 1000 labeled pixels for each class. The number of pixels for training and test are listed in Table II. Their corresponding distribution maps are demonstrated in Fig. 6.

B. Experimental Setup

In order to highlight the effectiveness of our proposed models, we compare them with SVM, 1-D CNN (1-D-CNN), 2-D CNN (2-D-CNN), and the original RNN using GRU (RNN). For simplicity, the cascaded RNN model using GRUs is abbreviated as CasRNN; the two improvement methods of CasRNN based on feature-level and output-level connections are abbreviated as CasRNN-F and CasRNN-O, respectively, the spectral–spatial CasRNN is abbreviated as SSCasRNN. Some of their explanations are summarized as follows.

1) *SVM:* The input of SVM is the original spectrum signature. We choose Gaussian kernel as its kernel function. The penalty parameter and the spread of the Gaussian kernel are selected from a candidate set $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ using a fivefold cross-validation method.
2) *1-D-CNN:* The structure of 1-D-CNN is the same as that in [35]. It contains an input layer, a convolutional layer with 20 kernels whose size is 11×1, a max-pooling layer whose kernel size is 3×1, a fully connected layer with 100 hidden nodes, and an output layer.
3) *2-D-CNN:* The structure of 2-D-CNN is the same as that in [38], which consists of three convolutional layers and two max-pooling layers. Please refer to [38, Table IX] for the design details of it.
4) *RNN:* GRU is used as the basic unit of RNN. The number of hidden nodes is chosen from a candidate set $\{2^4, 2^5, \ldots, 2^{10}\}$ via a fivefold cross-validation method.

The deep learning models are constructed with a PyTorch framework. To optimize them, we use a mini-batch stochastic gradient descent algorithm. The batch size, the learning rate, and the number of training epochs are set to 64, 0.001, and 300, respectively. For SVM, we use a libsvm package in a MATLAB framework. All of the experiments are implemented on a personal computer with an Intel core i7-4790, 3.60 GHz processor, 32 GB RAM, and a GTX TITAN X graphic card.

The classification performance of each model is evaluated by the overall accuracy (OA), the average accuracy (AA), the per-class accuracy, and the Kappa coefficient. OA defines the ratio between the number of correctly classified pixels to the total number of pixels in the test set, AA refers to the average of accuracies in all classes, and Kappa is the percentage of agreement corrected by the number of agreements that would be expected purely by chance.

C. Parameter Analysis

There exist three important hyperparameters in the proposed models. They are subsequence numbers l, as well as the size of hidden layers in the first-layer RNN and the second-layer RNN. To test the effects of them on the classification performance, we first fix l and select the size of hidden layers from a candidate set $\{16, 32, 64, 128, 256, 384\}$. Then, we fix the size of hidden layers and choose l from another set $\{2, 4, 6, \ldots, 16, 18, 20\}$. Since the same hyperparameter values are used for CasRNN and its two improvements (i.e., CasRNN-F and CasRNN-O), we only demonstrate the performance of CasRNN here, shown in Fig. 7. In this 3-D diagram, the first two axes (named hidden1 and hidden2), respectively, correspond to the number of hidden nodes in the first-layer RNN and the second-layer RNN, while the third axis represents the classification accuracy OA. From this figure, we can observe that when hidden1 ≥ 32 and hidden2 ≥ 128,

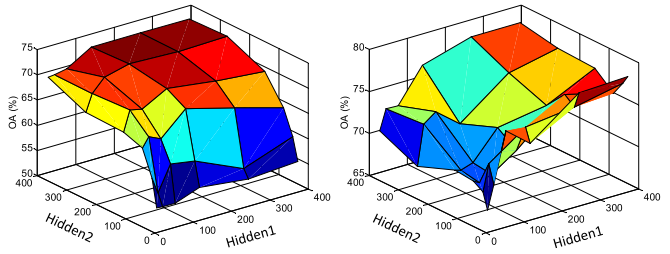


Fig. 7. Performance of the CasRNN model with different sizes of hidden layers on (Left) Indian Pines data and (Right) Pavia University data.

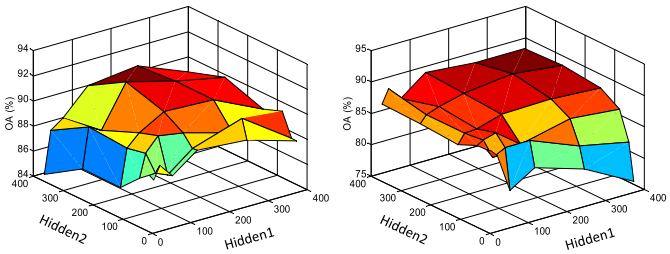


Fig. 8. Performance of the SSCasRNN model with different sizes of hidden layers on (Left) Indian Pines data and (Right) Pavia University data.

CasRNN can achieve better OA than the other values on the Indian Pines data. The best OA appears when hidden1 $= 128$ and hidden2 $= 256$. For the Pavia University data, OA changes a little larger than the Indian Pines data, but we can still find the best value when hidden1 $= 256$ and hidden2 $= 16$. Similarly, Fig. 8 shows OA values achieved by SSCasRNN using different hidden sizes. We can see the optimal parameter values are hidden1 $= 128$, hidden2 $= 256$ for the Indian Pines data, and hidden1 $= 256$, hidden2 $= 256$ for the Pavia University data, respectively.

Figs. 9 and 10 evaluate the effects of l on classifying the Indian Pines and the Pavia University data sets, respectively. In Figs. 9 and 10, different colors represent different models. They are CasRNN, CasRNN-F, CasRNN-O, and SSCasRNN. As l increases, OAs achieved by these models tend to increase first and then decrease. Given the same l, SSCasRNN significantly outperforms the other three models. For the Indian Pines data, the maximal OAs of four models appear at the same l, so their optimal l values are set as 10. Different from the Indian Pines data, four models have different optimal l values on the Pavia University data. As shown in Fig. 10, the optimal l value is 4 for SSCasRNN, and 8 for the other three models.

D. Performance Comparison

In this section, we will report quantitative and qualitative results of our proposed models and their comparisons with the other state-of-the-art models. Table III reports the detailed classification results of different models on the Indian Pines data, including OA, AA, Kappa, and class specific accuracy. The bold fonts in each row denote the best results. Several conclusions can be observed from Table III. First, if we directly input the whole spectral bands into RNN, its OA, AA, and Kappa values are 69.82%, 75.42%, and 65.87%, respectively, which are all lower than those achieved by SVM and 1-D-CNN models. This indicates that RNN cannot fully explore the

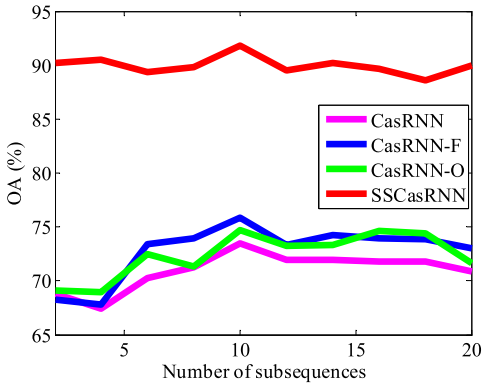


Fig. 9. Performance of different models on the Indian Pines data with different subsequence numbers l.

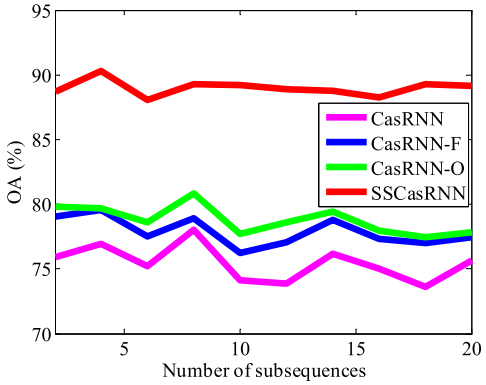


Fig. 10. Performance of different models on the Pavia University data with different subsequence numbers l.

long-term spectral sequence of HSIs. On the contrary, considering the redundant and complementary properties of spectral signature, our proposed model CasRNN can improve the performance of RNN by 4 percents, thus outperforming SVM and 1-D-CNN. Second, compared to CasRNN, CasRNN-F, and CasRNN-O can obtain better results, which validates the effectiveness of the two improvement strategies. In terms of each class accuracy, CasRNN-F almost increases all of them in comparison with CasRNN, so it might be more powerful than CasRNN-O on the Indian Pines data. Third, compared to spectral classification models, 2-D-CNN significantly improves the classification results by about 10 percents. It means that the consideration of spatial information is very important on the Indian Pines data, because there are many large and homogeneous objects shown in Fig. 5(c). By incorporating the spatial information into CasRNN model, our proposed model SSCasRNN can further increase the performance to above 90 percents. Besides, it can obtain highest accuracies in 15 different classes, which sufficiently certifies the effectiveness of SSCasRNN.

In addition to the quantitative results, we also visualize classification results of different models shown in Fig. 11. Different colors in Fig. 11 correspond to different classes. Compared to the groundtruth map in Fig. 5(c), spectral classification models (i.e., SVM, 1-D-CNN, RNN, CasRNN, CasRNN-F, and CasRNN-O) have many outliers in the classification map due to the spectral variability of materials.

TABLE III
CLASSIFICATION RESULTS (%) OF DIFFERENT MODELS ON THE INDIAN PINES DATA

Class No.	SVM	1D-CNN	RNN	CasRNN	CasRNN-F	CasRNN-O	2D-CNN	SSCasRNN
1	64.31	61.34	64.74	68.35	68.93	68.21	82.51	**86.99**
2	70.92	60.33	61.35	64.8	67.6	67.35	88.14	**98.72**
3	84.78	80.43	74.46	77.17	83.7	85.87	**100**	**100**
4	91.05	89.04	83.45	91.50	90.60	89.93	**94.85**	94.41
5	85.94	90.53	77.04	79.34	80.49	80.92	85.80	**97.42**
6	93.62	96.13	87.70	92.03	92.94	92.94	99.77	**100**
7	69.17	72.11	76.03	74.84	78.54	79.30	82.35	**87.15**
8	52.90	54.47	60.79	67.41	67.49	66.91	73.86	**85.98**
9	76.60	75.71	61.17	65.60	67.02	65.43	86.00	**87.23**
10	97.53	99.83	93.21	95.06	96.91	98.15	**100**	**100**
11	77.49	80.87	81.67	83.28	90.03	86.09	94.53	**97.51**
12	73.33	78.48	55.45	54.85	67.88	54.55	97.27	**99.70**
13	**100**	91.11	86.67	93.33	95.56	93.33	**100**	**100**
14	87.18	94.87	69.23	76.92	84.61	76.92	97.44	**100**
15	90.91	90.91	90.91	90.91	90.91	90.91	**100**	**100**
16	**100**	**100**	80.00	**100**	**100**	80	**100**	**100**
OA	70.55	70.79	69.82	73.49	75.85	74.60	85.43	**91.79**
AA	82.23	82.23	75.24	79.71	82.70	79.80	92.66	**95.94**
Kappa	66.90	67.07	65.87	69.91	72.57	71.19	83.49	**90.62**

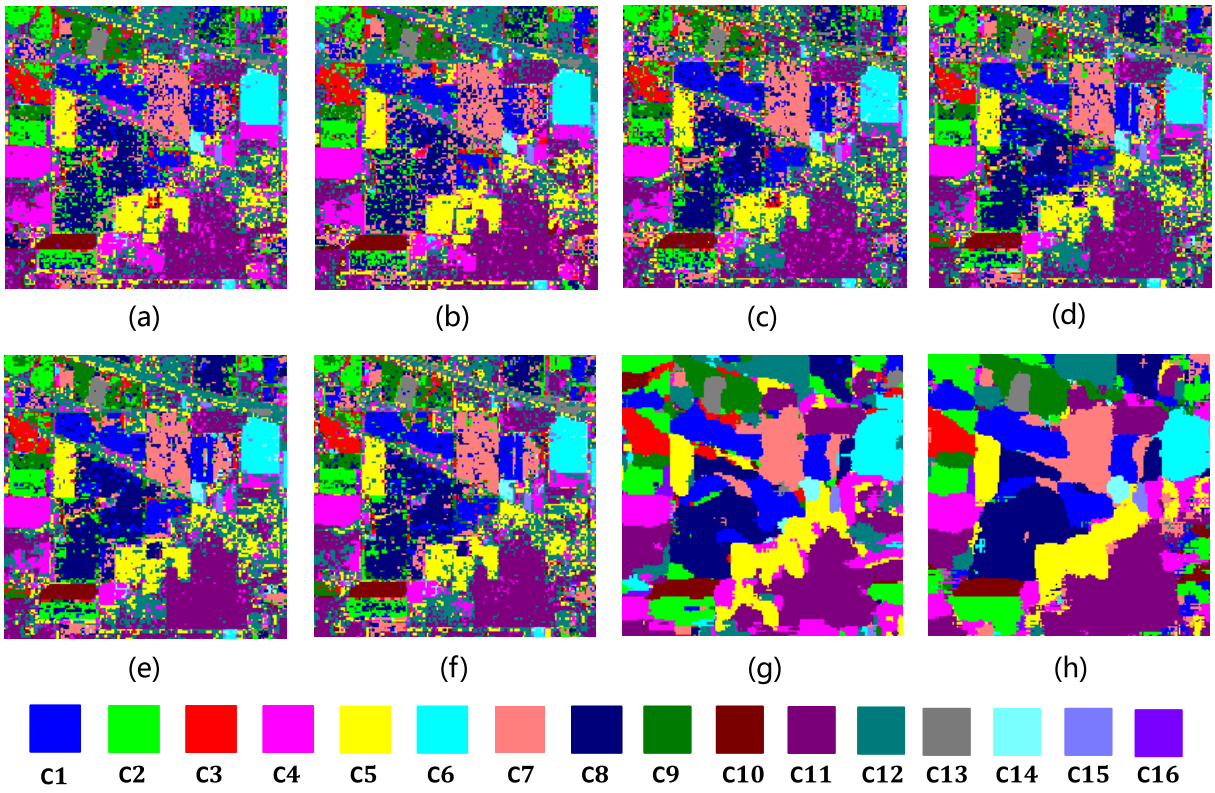


Fig. 11. Classification maps of the Indian Pines data using different models. (a) SVM. (b) 1-D-CNN. (c) RNN. (d) CasRNN. (e) CasRNN-F. (f) CasRNN-O. (g) 2-D-CNN. (h) SSCasRNN.

This phenomenon can be alleviated by 2-D-CNN, because it makes use of the spatial contextual information instead of the spectral information. For homogeneous regions, especially large objects, 2-D-CNN performs very well. However, it will easily result in an over-smoothing problem, especially for small objects, as demonstrated in Fig. 11(g). Different from 2-D-CNN and spectral models, SSCasRNN takes advantage of spectral and spatial information simultaneously. As shown in Fig. 11 (h), it has significantly fewer outliers than spectral models, and retains more boundary details of objects than 2-D-CNN.

Table IV and Fig. 12 are the classification results of different models on the Pavia University data. Similar conclusions can be observed from them. For spectral models, CasRNN is better

TABLE IV
CLASSIFICATION RESULTS (%) OF DIFFERENT MODELS ON THE PAVIA UNIVERSITY DATA

Class No.	SVM	1D-CNN	RNN	CasRNN	CasRNN-F	CasRNN-O	2D-CNN	SSCasRNN
1	84.74	80.94	81.51	82.34	83.56	83.52	77.39	**89.82**
2	64.50	70.37	62.58	67.13	70.65	71.37	**98.89**	96.06
3	72.56	77.32	64.65	60.51	68.75	64.51	56.74	**78.89**
4	97.13	85.93	**98.89**	98.63	98.11	98.43	92.75	95.89
5	99.55	99.70	99.26	99.41	99.55	99.33	99.78	**100**
6	**93.30**	93.26	88.90	84.97	88.29	89.08	47.27	57.67
7	91.28	**95.41**	92.63	90.60	76.54	91.13	80.08	80.53
8	91.99	84.47	91.04	92.23	86.04	93.54	**96.69**	96.80
9	95.56	92.08	95.35	94.40	95.35	94.72	**96.30**	95.99
OA	78.75	79.55	76.58	78.03	79.56	80.86	86.18	**90.30**
AA	87.85	86.61	86.09	85.58	85.21	87.29	82.88	**87.97**
Kappa	73.62	74.28	71.02	72.55	74.31	75.93	81.22	**86.26**

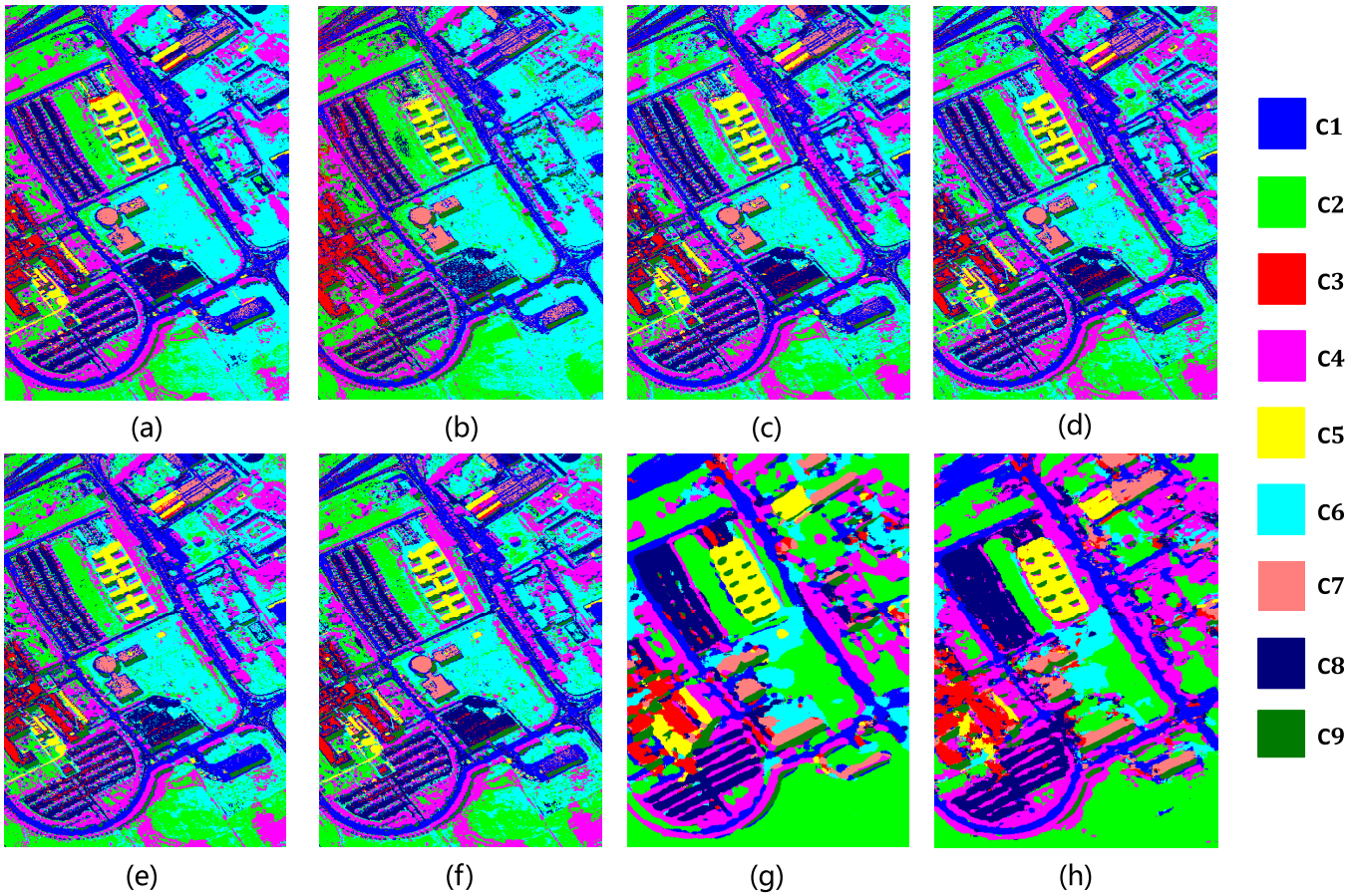


Fig. 12. Classification maps of the Pavia University data using different models. (a) SVM. (b) 1-D-CNN. (c) RNN. (d) CasRNN. (e) CasRNN-F. (f) CasRNN-O. (g) 2-D-CNN. (h) SSCasRNN.

than RNN, while CasRNN-F and CasRNN-O are superior to CasRNN. All of these models have the "salt and pepper" phenomenon in their classification maps. Compared to the best spectral model, 2-D-CNN can improve OA and Kappa by more than 5 percents. In addition, it generates fewer outliers and leads to a more homogeneous classification map. Nevertheless, without using the spectral information, its performance is not very high, and the classification map is easily to be over-smoothed. Combining the spectral and spatial information together, our proposed model SSCasRNN can alleviate these issues. It improves OA from 86.18% to 90.30%, and generates more details in the classification map. However, in comparison with the Indian Pines data, the classification results achieved by SSCasRNN are still not very high. One possible reason is that there exist many small objects in the Pavia University data, which increases the difficulty in exploring spatial features.

IV. CONCLUSION

In this paper, we proposed a cascaded RNN model for HSI classification. Compared to the original RNN model, our

proposed model can fully explore the redundant and complementary information of the high-dimensional spectral signature. Based on it, we designed two improvement strategies by constructing connections between the first-layer RNN and the output layer, thus generating more discriminative spectral features. In addition, considering the importance of spatial information, we further extended the proposed model into its spectral–spatial version to learn spectral and spatial features simultaneously. To test the effectiveness of the proposed models, we compared them with several state-of-the-art models on two widely used HSIs. The experimental results demonstrate that the cascaded RNN model can obtain higher performance than RNN, and its modifications can further improve the performance. Besides, we also thoroughly evaluated the effects of different hyperparameters on the classification performance of the proposed models, including the hidden sizes and the number of subsequences. In the future, more experiments will be conducted to validate the effectiveness of our proposed models. In addition, more powerful spectral–spatial models will be explored. Since the sizes and shapes of different objects vary, using the patches or cubes with same sizes as inputs easily leads to the loss of spatial information.

REFERENCES

[1] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[2] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.

[3] M. Belgiu and L. Drăgut, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016.

[4] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[5] X. Jia, B.-K. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.

[6] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.

[7] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 783–794, Feb. 2016.

[8] R. Hang *et al.*, "Robust matrix discriminative analysis for feature extraction from hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 2002–2011, May 2017.

[9] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 55–66, Jan. 2014.

[10] R. Hang and Q. Liu, "Dimensionality reduction of hyperspectral image using spatial regularized local graph discriminant embedding," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 9, pp. 3262–3271, Sep. 2018.

[11] W. Zhao, W. J. Emery, Y. Bo, and J. Chen, "Land cover mapping with higher order graph-based co-occurrence model," *Remote Sens.*, vol. 10, no. 11, p. 1713, 2018.

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

[13] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.

[14] J. Li, H. Zhang, Y. Huang, and L. Zhang, "Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3707–3719, Jun. 2014.

[15] W. Li and Q. Du, "Joint within-class collaborative representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2200–2208, Jun. 2014.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[17] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[22] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

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

[24] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the Art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[25] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[26] 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.

[27] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

[28] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[29] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[30] X. Zhou, S. Li, F. Tang, K. Qin, S. Hu, and S. Liu, "Deep learning with grouped features for spatial spectral classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 97–101, Jan. 2017.

[31] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[32] Y. Li, W. Xie, and H. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," *Pattern Recognit.*, vol. 63, pp. 371–383, Mar. 2017.

[33] W. Zhao, S. Du, and W. J. Emery, "Object-based convolutional neural network for high-resolution imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 7, pp. 3386–3396, Jul. 2017.

[34] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[35] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.

[36] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2017.

[37] P. Ghamisi *et al.*, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.
[38] 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.
[39] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, 2017.
[40] C. Shi and C.-M. Pun, "Superpixel-based 3D deep neural networks for hyperspectral image classification," *Pattern Recognit.*, vol. 74, pp. 600–616, Feb. 2018.
[41] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.
[42] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 937–949, Feb. 2018.
[43] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.
[44] H. Wu and S. Prasad, "Convolutional recurrent neural networks for hyperspectral data classification," *Remote Sens.*, vol. 9, no. 3, p. 298, 2017.
[45] 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, p. 1330, 2017.
[46] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, Feb. 2018.
[47] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Integrating convolutional neural network and gated recurrent unit for hyperspectral image spectral-spatial classification," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis. (PRCV)*. Guangzhou, China: Springer, 2018, pp. 409–420.
[48] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555
[50] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2014, pp. 1724–1734.

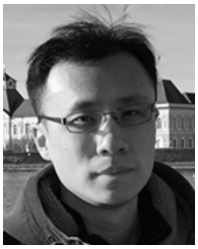
Renlong Hang (M'17) received the M.S. and Ph.D. degrees from the Nanjing University of Information Science and Technology, Nanjing, China, in 2014 and 2017, respectively.

Since 2017, he has been a Lecturer with the School of Automation, Nanjing University of Information Science and Technology. He is currently a Post-Doctoral Researcher with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, Kansas City, MO, USA. He has authored or co-authored over 20 peer-reviewed articles in international journals, such as the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and *Remote Sensing*. His research interests include machine learning, pattern recognition, and their applications to remote sensing image processing.

Qingshan Liu (M'05–SM'07) received the M.S. degree from Southeast University, Nanjing, China, in 2000, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2003.

He was an Associate Professor with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. From 2010 to 2011, he was an Assistant Research Professor with the Department of Computer Science, Computational Biomedicine Imaging and Modeling Center, Rutgers, NJ, USA, and The State University of New Jersey, Piscataway, NJ, USA. He was an Associate Professor with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. From 2004 to 2005, he was an Associate Researcher with the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong. He is currently a Professor with the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing. His research interests include image and vision analysis.

Dr. Liu was a recipient of the President Scholarship of the Chinese Academy of Sciences in 2003.



Danfeng Hong (S'16) received the B.Sc. degree in computer science and technology from the Neusoft College of Information, Northeastern University, Shenyang, China, in 2012, the M.Sc. degree in computer vision from Qingdao University, Qingdao, China, in 2015. He is currently pursuing the Ph.D. degree in signal processing in earth observation with the Technical University of Munich, Munich, Germany.

Since 2015, he has been a Research Associate with the Remote Sensing Technology Institute, German Aerospace Center, Oberpfaffenhofen, Germany. In 2018, he joined the GIPSA-lab, Grenoble INP, CNRS, Univ. Grenoble Alpes, Grenoble, France, as a Visiting Student, under the supervision of Prof. Jocelyn Chanussot.

His research interests include signal/image processing and analysis, pattern recognition, machine/deep learning, and their applications in earth Vision.

Pedram Ghamisi (S'12-M'15-SM'18) received the B.Sc. degree in civil (survey) engineering from the Tehran South Campus of Azad University, Tehran, Iran, the M.Sc. degree (First Class Hons.) in remote sensing from the K. N. Toosi University of Technology, Tehran, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2015.

From 2013 to 2014, he was with the School of Geography, Planning and Environmental Management, The University of Queensland, Brisbane, QLD, Australia. He was a Post-Doctoral Research Fellow at the University of Iceland. He has been a Post-Doctoral Research Fellow at the Technical University of Munich, Munich, Germany, and Heidelberg University, Heidelberg, Germany, since 2015. He was a Research Scientist at the German Aerospace Center, Remote Sensing Technology Institute, Oberpfaffenhofen, Germany, from 2015 to 2018.He was the Head of the Machine Learning Group at Helmholtz-Zentrum Dresden-Rossendorf, Dresden, Germany. His research interests include interdisciplinary research on remote sensing and machine (deep) learning, image, and signal processing, and multisensory data fusion.

Dr. Ghamisi was a recipient of the prestigious "Alexander von Humboldt Fellowship" in 2015, the prestigious "High Potential Program" in 2018, the Best Researcher Award for M.Sc. students in K. N. Toosi University of Technology, from 2010-2011, the 2013 IEEE International Geoscience and Remote Sensing Symposium, Melbourne, in 2013, the IEEE Mikio Takagi Prize for winning the Student Paper Competition, competing with almost 70 submissions, the Data Fusion Contest 2017 organized by the Image Analysis and Data Fusion Technical Committee of the Geoscience and Remote Sensing Society. He was also the winner of the 2017 Best Reviewer Prize of IEEE Geoscience and Remote Sensing Letters. In 2016, he was selected as the Talented International Researcher by Iran's National Elites Foundation. He serves as an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and the *Remote Sensing*.