Neurocomputing 387 (2020) 150–160

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Spectral-spatial classification for hyperspectral image based on a single GRU

Erting Pan[a], Xiaoguang Mei[a,b], Quande Wang[a,*], Yong Ma[a,b], Jiayi Ma[a,b]

[a] Electronic Information School, Wuhan University, Wuhan 430072, China
[b] Institute of Aerospace Science and Technology, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Article history:
Received 13 August 2019
Revised 25 October 2019
Accepted 8 January 2020
Available online 13 January 2020

Communicated by Dr. B. Hu

Keywords:
Hyperspectral image pixel-level classification
Deep learning
RNN
GRU

ABSTRACT

Deep learning methods have been successfully used to extract deep features of many hyperspectral tasks. Multiple neural networks have been introduced in the classification of hyperspectral images, such as convolutional neural network (CNN) and recurrent neural network (RNN). In this study, we offer a different perspective on addressing the hyperspectral pixel-level classification task. Most existing methods utilize complex models for this task, but the efficiency of these methods is often ignored. Based on this observation, we propose an effective tiny model for spectral-spatial classification on hyperspectral images based on a single gate recurrent unit (GRU). In our approach, the core GRU can learn spectral correlation within a whole spectrum input, and the spatial information can be fused as the initial hidden state of the GRU. By this way, spectral and spatial features are calculated and expanded together in a single GRU. By comparing the different utilization patterns of RNN with a variety of spatial information fusion methods, our approach demonstrates a competitive advantage in both accuracy and efficiency.

1. Introduction

Modern hyperspectral sensors can capture high spectral resolution data up to hundreds of bands, allowing for the distinction of very similar materials and objects. Rich spectral information offers great potential for classification [1–4]. Hence, the analysis of hyperspectral imagery has attracted broad attention in remote sensing. Hyperspectral image (HSI) contains abundant spectral and spatial information, which has been widely applied in many fields such as agriculture, mining, environmental monitoring, land-cover mapping [5–9].

HSI classification aims to identify each pixel vector into a discrete set of specific classes. Many of the traditional approaches have concentrated on processing spectral features. Some of them exclusively employ the advantage of distinguishing the subtle spectral difference to determine its class belonging, such as random forest [10,11], support vector machine (SVM) [12,13], sparse representation models [14–17]. However, these methods depend on manual features and due to their limitations, they cannot extract robust deep feature representations.

Unlike traditional classifiers, deep learning methods exploit high-level features which have the capability to acquire more complex structure representations [18–21]. In particular, convolutional neural network (CNN) and recurrent neural network (RNN) have gained great success in a variety of computer vision tasks [22–24]. Taking CNN's advantage of local connection and weight sharing properties, Wu et al. [25] directly deployed the spectral feature of the original image data as an input vector and utilized 1D-CNN for spectral HSI classification. But due to the kernel size limitations, 1D-CNN can only learn the local spectral dependency. A few works attempt to regard the spectral data as a sequence [26–30], naturally, RNN becomes a candidate model by its advantage of processing sequence data. Mou et al. [26] modeled the spectra of hyperspectral pixel as a 1D sequence vector for classification, and it was shown that GRU in RNN is a better choice for HSI classification, rather than long short-term memory (LSTM) cell. The spectra was input to the RNN which was formed of multiple GRUs, and each band is expanded and then delivered in the corresponding GRU, and the number of GRUs in the entire RNN network equals to the number of bands of hyperspectral data. It acquired competitive performance and showed the huge potential of deep recurrent networks for hyperspectral data analysis. Nevertheless, it only took spectral information and the entire network with hundreds of GRUs cost a heavy computation.

Many researchers also have developed various approaches considering spatial information [31–37]. For instance,

* Corresponding author at Electronic Information School, Wuhan University, Wuhan 430072, China.
E-mail addresses: panerting@whu.edu.cn (E. Pan), meixiaoguang@gmail.com (X. Mei), Qdwang@sohu.com (Q. Wang), mayong@whu.edu.cn (Y. Ma), jyma2010@gmail.com (J. Ma).

https://doi.org/10.1016/j.neucom.2020.01.029
0925-2312/

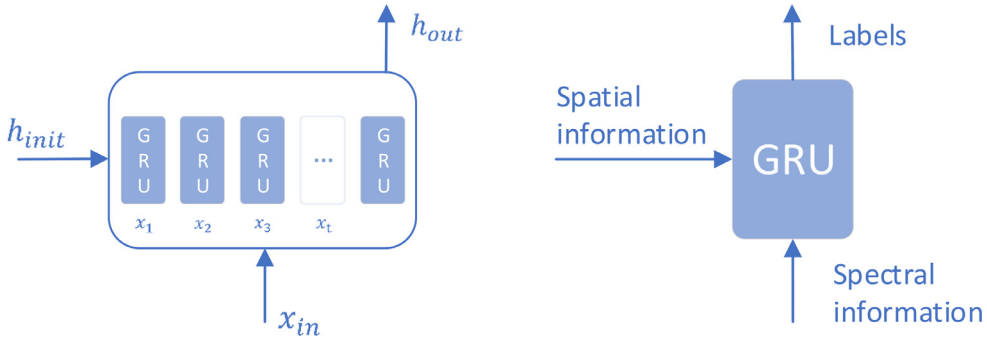

Fig. 1. Illustration of RNN and the proposed single GRU.

Chen et al. [35] proposed a 3D-CNN network which directly learns spectral-spatial features over both spatial and spectral axes, but its computational complexity is dramatically increased. In addition, a combination of CNN and RNN has been developed for hyperspectral image analysis. Xu et al. [36] proposed a unified network with a bands-grouping based LSTM and MSCNN as the spectral and spatial feature extractors. Mei et al [37] proposed to concatenate a spatial attention CNN branch and a spectral attention bi-directional RNN branch to learn joint features.

So far, these deep learning methods mentioned above have yielded good results. Most of them are combinations of complex models, which can lead to heavy computation burdens, and the efficiency of these models can be easily ignored. Accordingly, aiming at the efficiency of computation, we design a simple and most effective method for the hyperspectral pixel-level classification.

Typically, as shown in the RNN unfolding structure(see the left part of Fig. 1), it can be seen that the number of internal units is related to the timestep, and the recurrent connection is between the hidden units corresponding to each timestep. Inspired by the RNN and its various variants [28,36–38], the RNN component (such as a GRU or LSTM unit) corresponding to each timestep can be input to not only a single data but also a subsequence. Therefore, we design multiple comparison experiments by inputting sub-sequences of different lengths for timesteps. For the spectral vector of HSI data, we figure out using a single GRU, which means to input the entire spectral vector directly as one timestep, also can make full use of spectral information.

Unlike an RNN consisting of multiple GRUs, a single GRU does not carry the self-recurrent feature of RNN. From the characteristics of its own internal structure, GRU is a fully connected layer with a gate mechanism. The update gate and reset gate play a role to transform and select inputs. For the lengthy spectral vector in HSI data, this structure is tiny and effective in extracting spectral discriminative features. In addition, we also use the other input of GRU to capture the neighbor spatial information. The overall structure is shown in the right part of Fig. 1. The contribution of this work can be summarized as follows:

- We develop a tiny effective model for HSI spectral-spatial classification, which consists of only a single GRU. Our model acquires competitive performance with fully exploiting spectral and spatial features.
- We propose a tiny structure to extract spectral features. Taking the hyperspectral spectral vector as a 1D sequence, we utilize the RNN to extract spectral features. Instead of inputting each band as one timestep to the RNN which is formed of multiple GRUs, considering the high correlations between the reflected values of the neighboring bands and the integrated spectral profile, the whole spectrum data is input as one timestep into RNN which is formed of a single GRU. It greatly reduces the computation burden of the entire network.

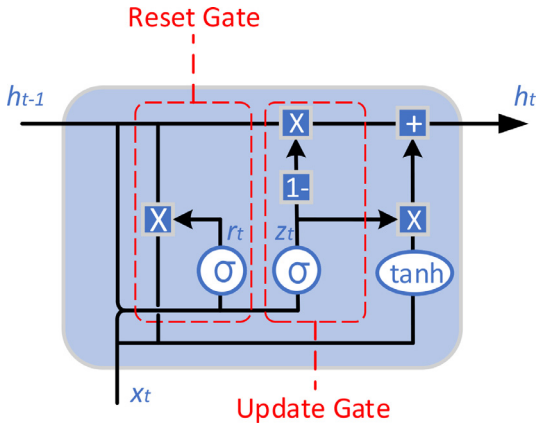

Fig. 2. Illustration of GRU cell.

- We design a novel way to fuse spatial information. As the initial state of RNN can be a trainable variable, in this work, we put spatial neighboring features as the initial state of the GRU to training. By this way, the spectral and spatial information are calculated and expanded in a single GRU. The experimental result shows that the proposed tiny network is indeed effective.

2. Preliminaries

RNN has received extensive concern in modeling sequence data. Unlike feed-forward neural networks, RNN is called recurrent because of its recurrent hidden state, whose activation at each step depends on the previous computations. RNN has a memory function, which can remember the information about what has been calculated so far.

The most commonly used type of RNN is the LSTM and GRU architectures, which are explicitly designed to deal with vanishing gradients and efficiently capture long-term dependencies. These two have no difference of fundamental architecture with RNNs, but they use a different function to compute the hidden state.

LSTM was first proposed in 1997 [39] and is the most widely used model in NLP today. The memory in LSTMs is called cells and and can be regarded as black boxes that take the previous state h_{t-1} and current x_t as input. Internally these cells decide what to keep in (and what to erase from) memory. They use three gates to combine the previous state, the current memory and the input to control what information will be passed through. GRU (see Fig. 2), first proposed in 2014 [40], is a simplified version of LSTM. Compared with LSTM, GRU does not maintain a cell state C and uses two gates instead of three. GRU has fewer parameters and thus may train a bit faster or need less data to generalize.

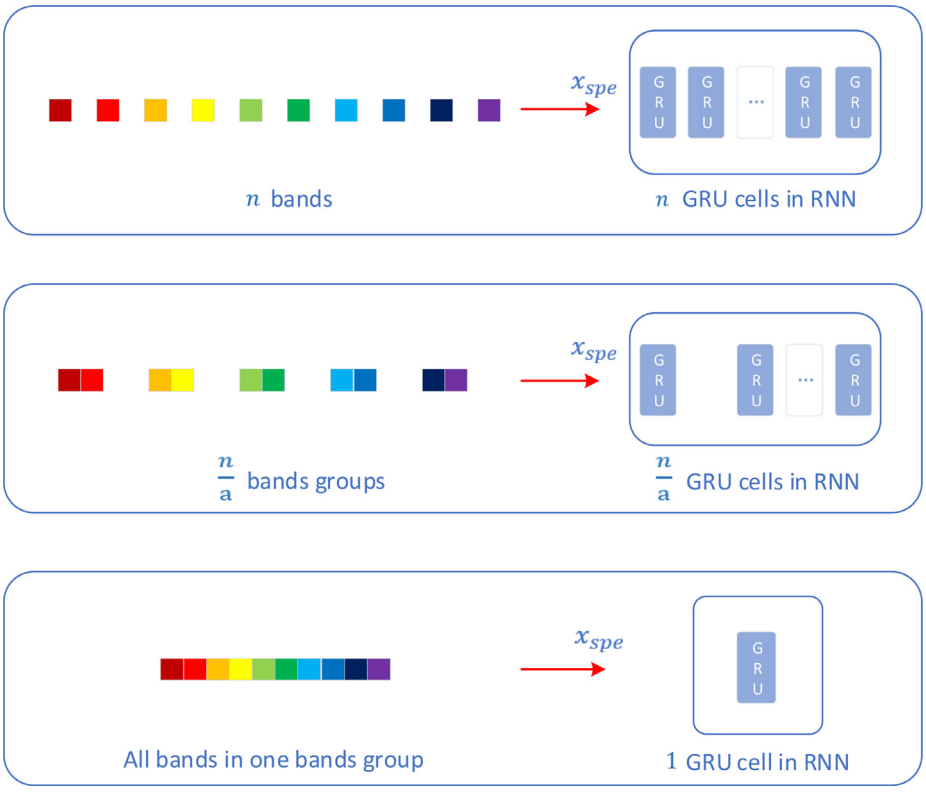

Fig. 3. Illustration of three different strategies for the spectral data as the input of RNN. The top one shows the band by band strategy. The middle one illustrates a bands grouping strategy, and the last one is a special situation to the middle one, named all in one stategy.

A GRU has two gates, *i.e.*, a reset gate r_t and an update gate z_t:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \tag{1}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \tag{2}$$

where $\sigma(\cdot)$ denotes a logistic sigmoid function, W_r and W_z are weight matrices.

Intuitively, the reset gate determines how to combine the new input with the previous memory, and it acts similar to the forget and input gates of an LSTM. It decides what information to throw away and what new information to add. The update gate defines how much of the previous memory to keep around. If we set the reset to all 1 and update gate to all 0, we again arrive at our plain RNN model. The new hidden state is computed as:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t, \tag{3}$$

$$\widetilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]), \tag{4}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, W is the weight matrix. The new hidden state is also the output of GRU.

3. Methodology

3.1. Extracting spectral feature

The data advantage of HSIs is spectral data that contain rich object features, and numerous of hyperspectral imagery studies focus on obtaining information from spectral data. In the hyperspectral data cube, spectral data consist of reflected values from hundreds of narrow and continuous spectral bands, which are one-dimensional ordered sequences.

For a hyperspectral pixel z, the kth spectral band is denoted as z_k, and $x^{(k)}$ represents the input of kth time step in RNN. The kth GRU cell in RNN receives the previous hidden state $x^{(k-1)}$ and the current input $x^{(k)}$ and calculates the current state information. When the total number of bands is n, Eqs. (5), (6), (7) show the input of three strategies to extract spectral features:

Band by band strategy:

$$x^{(1)}, x^{(2)}, \ldots, x^{(n)} = [z_1, z_2, \ldots, z_n]. \tag{5}$$

Bands grouping strategy:

$$\begin{aligned} x^{(1)} &= [z_1, z_2, ..., z_a], \\ x^{(2)} &= [z_{a+1}, z_{a+2}, ..., z_{2a}], \\ &\ldots \\ x^{(\frac{n}{a})} &= [z_{n-a+1}, z_{n-a+2}, ..., z_n]. \end{aligned} \tag{6}$$

All in one strategy:

$$x = [z_1, z_2, \ldots, z_n]. \tag{7}$$

Referring to RNN first introduced in the HSI classification [26], as illustrated in the top figure of Fig. 3. Considering each band as a timestep to input a GRU, the network consists of multiple GRUs which are cascaded to form RNN to learn spectral features. We name it a band-by-band strategy, as in Eq. (5). Each band is non-linearly expressed, stored and selected by the GRU with its gate mechanism, and then transmitted as the hidden state one by one in hundreds of GRUs. Finally, it outputs the hidden state of the last GRU that has conveyed information from all previous GRUs. In this way, the RNN effectively captures the forward correlation in the spectral data. However, for each pixel in the original HSI, the spectral data is a profile, which imply that adjacent spectral bands in the spectrum are correlated. Not only is forward association

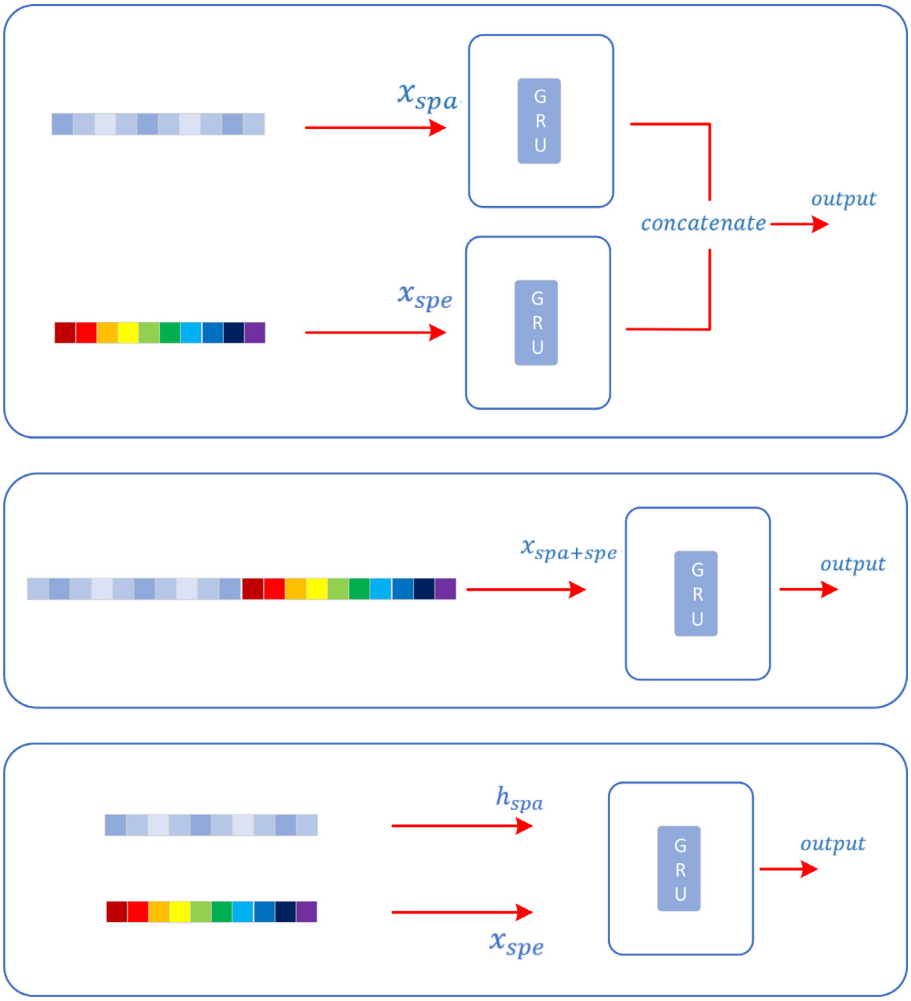

Fig. 4. Illustration of three strategies to fuse the spatial information. The top one sets another branch for spatial data and concatenates the outputs of two branches. The middle one concatenates the 1D spectral vector and the reshaped 1D spatial data as input. The last one fuses the spatial data as the initial state of GRU.

important, but also contextual information about spectral data should be noticed.

A direct solution is a bands-grouping strategy, as shown in the middle plot of Fig. 3. The spectral vector consisting of hundreds of bands can be grouped according to their order. We divide the entire spectrum into $\frac{n}{a}$ bands-groups, as Eq. (6). The relationship between adjacent bands in a group can be fully expressed in the hidden state and then transmitted into the next GRU.

The advantage of this bands-grouping strategy is that each timestep could concentrate on local context features extracted from a bands-group with a small wavelength range. Moreover, another advantage of this strategy is efficiency. The bands-grouping strategy with fewer GRUs has much fewer parameters and greatly reduces the computational burden of the entire network.

There is a special situation when $a = n$, which means that all spectrum bands are only divided into one bands-group, we name it the all-in-one strategy, as in Eq. (7). In other words, we input the entire spectrum into an RNN with only one GRU in one timestep, as shown in the bottom part of Fig. 3. Unlike the bands-grouping strategy focusing on local features in bands-group, the all-in-one strategy takes the whole spectra of the target hyperspectral pixel as the input to a GRU without breaking data continuity. Furthermore, it can also capture features from global context information. The gate mechanism of GRU here is independent of solving the gradient disappearance problem, but relies on the internal filtering function of the reset gate and update gate to play a role in.

With the all-in-one strategy, the entire spectra maintain its internal spectral correlation and are fully expressed by the GRU. Besides, its speed advantage is self-evident.

3.2. Fusing spatial information

The spatial feature is a valuable complement to the spectral signatures. The spatial information of a target pixel in hyperspectral data usually comes from its neighbor region. Principal component analysis (PCA) is commonly executed in the first step to map the data to an acceptable scale with a low information loss. Next, to capture local spatial features, spatial information is collected for each central pixel by dividing a neighboring region of size $k \times k \times m$, where k is the size of the adjacent region and m is the number of principal components. After that, almost all relevant spatial information is collected, and then we reshape the spatial data of this neighboring region into a 1D sequence, which is mapped by a fully connected layer with ReLU activation function and awaits for subsequent processing.

To fuse the spatial information, the first choice is to set another branch for spatial data (same as the spectral branch), and then concatenate the output of two branches in a new fully connected layer, and then add the softmax function to acquire the final label. It is shown in the top plot of Fig. 4. This two-branch strategy utilizes two GRUs.

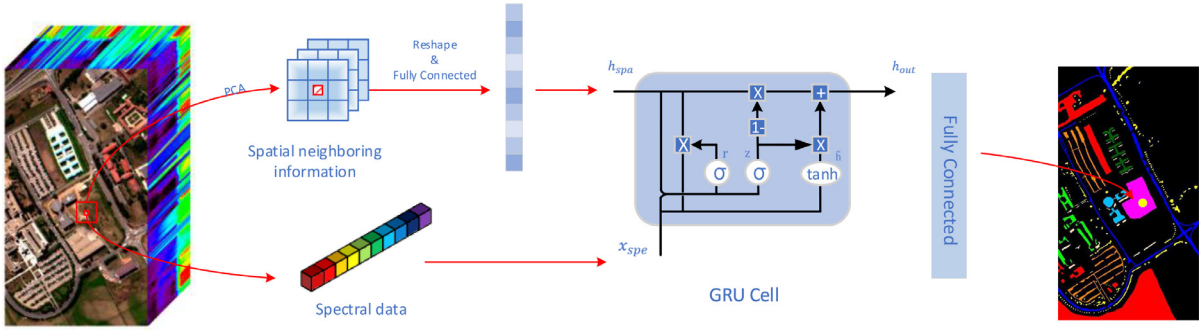

Fig. 5. Illustration of the pipeline of the proposed method. The core member of this model is a single GRU, it receives two parts of input. One is the spectral vector x_{spe} from the origin HSI data. The other is the spatial vector h_{spa}, which is obtained through firstly processing by PCA and then selecting and reshaping a suitable size of the neighboring region into a 1D sequence and finally mapping by a full connection layer. With the inner calculation in GRU, the spectral and spatial features are selected and fused, and then transmitted to the later fully connected layer. We get the final label by softmax function.

Since spatial data have been reshaped into 1D, we have a simple idea that we can directly concatenate the spatial data and the spectral data into a 1D sequence as the input, shown in the middle part of Fig. 4. Subsequent calculations are performed through the RNN with all in one strategy. The spatial and spectral information are jointly developed in the feature space. However, the problem with this strategy is that although the spatial and spectral sequences in the new sequence are related to the target hyperspectral pixels, the internal order of the spectra relates to the wavelength of each band, the spatial data do not have sequential order. There is no ordered correlation between the two subsequences.

In the RNN, each GRU typically receives two parts of data, including the input x^t and the hidden state h^{t-1} passed by the previous GRU. For the first GRU in the RNN, there is a special parameter h^0 called the initial hidden state, usually set to zero. If this parameter is set to other value, it also participates in the internal calculation of the GRU and is passed backward.

Inspired by this, when it comes to processing HSI data within a single GRU, in addition to taking the spectral vector x^{spe} as input, we can also use the initial hidden state parameter to process spatial information as h^{spa}. It is shown in the bottom plot of Fig. 4. Both parts of the data participate in the internal calculation of the GRU. In this way, spectral features and spatial features can be captured simultaneously in only a single GRU.

3.3. Pipeline

The pipeline of this methodology is illustrated in Fig. 5. Obviously, the core member of our model is the GRU. For every pixel in HSI data, its spectral data is input as x_{spe} directly. To make the full use of spatial information, the origin HSI data is processed by PCA to reduce the dimension, and then a neighbor region related to the center pixel is acquired, and finally the selected region is deformed as 1D sequence and mapped to the fully connected layer to obtain the spatial feature h_{spa}. Utilizing the trainable initial hidden state parameter in the GRU, the spatial feature h_{spa} can be passed into and participate in the calculation along with the input spectral vector x_{spe}.

The reset gate r and update gate z in GRU are presented as Eqs. (8) and (9):

$$r = \sigma\left(W_r \cdot x_{spe} + U_r \cdot h_{spa} + b_r\right), \tag{8}$$

$$z = \sigma\left(W_z \cdot x_{spe} + U_z \cdot h_{spa} + b_z\right), \tag{9}$$

$$\widetilde{h} = \tanh\left(W_{\widetilde{h}} \cdot x_{spe} + U_{\widetilde{h}} \cdot (r * h_{spa}) + b_{\widetilde{h}}\right), \tag{10}$$

$$h_{out} = (1 - z) * h_{spa} + z * \widetilde{h}, \tag{11}$$

where $\sigma(\cdot)$ denotes a logistic sigmoid function, W_r, W_z, $W_{\widetilde{h}}$, U_r, U_z and $U_{\widetilde{h}}$ are weight matrices, b_r, b_z and $b_{\widetilde{h}}$ are bias vectors. $\tanh(\cdot)$ is the hyperbolic tangent function. x_{spe} represents a $1 \times b$ vector, b equals to the number of spectral bands, the spatial neighbor region is mapped as h_{spa} via a fully connected layer. It is worth to mention that the number of hidden layers of this GRU cell is no more an adjustable parameter, but a fixed number, equaling to the size of h_{spa}.

As Eq. (10), $\widetilde{h}$ consists of two parts, one is to use the reset gate r to store important information related to h_{spa}, and the other is to add important information of x_{spe}. All the memory of the GRU is made up of these two parts. Finally, the network needs to compute the h_{out}. As Eq. (11), the update gate z is used. On the one hand, as the first item, it determines how much spatial information in h_{spa} is retained. On the other hand, the second item indicates the information needs to be forgotten, and updates the corresponding content with $\widetilde{h}$. Then, h_{out} is what to be collected in the memory $\widetilde{h}$ and the initial hidden state h_{spa}. After mapping through a fully connected layer with ReLU activation function, we get the final label by softmax function.

In summary, our model is exquisite and effective. Based on the data characteristics of HSI, its spectral features and spatial features are extracted in a single GRU by fully exploiting the structure of GRU and its unique gating mechanism. Compared with other complex and diverse models, our model has an advantage in efficiency and competitive results in classification performance.

4. Experimental results and analysis

4.1. Data description

We choose three public available HSI classification datasets to evaluate the performance of the proposed model, including Pavia Center dataset, Pavia University dataset and Indian Pines dataset.[1]

The Pavia Center dataset is gathered by reflective optics system imaging spectrometer ROSIS. This data set includes 102 spectral bands after removing 13 noisy channels with 1096×715 pixels, and it presents 9 classes covering the center of Pavia. The false-color composition picture of the Pavia Center image and the corresponding ground truth map are shown in Fig. 6. The 10% samples are set to the training set, and the rest are set to the testing set.

[1] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

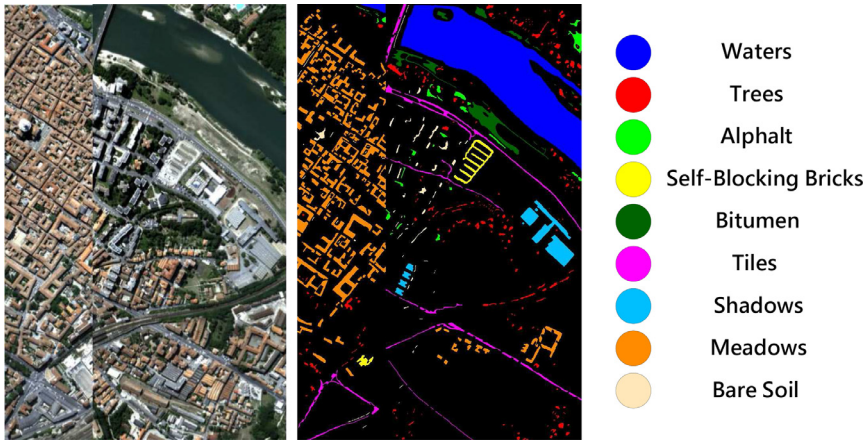

Fig. 6. False color image and ground-truth labels of the Pavia Center dataset.

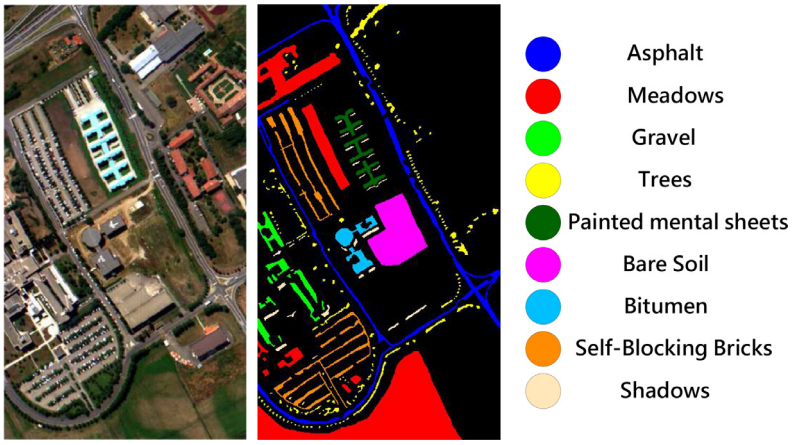

Fig. 7. False color image and ground-truth labels of the Pavia University dataset.

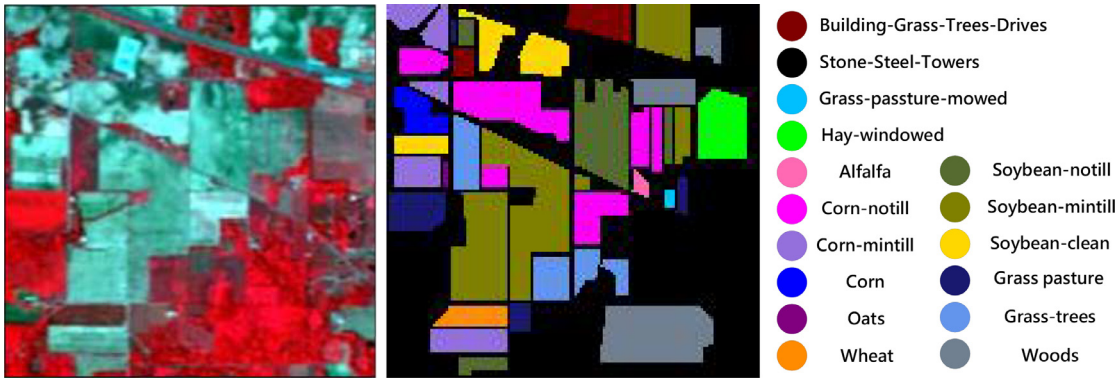

Fig. 8. False color image and ground-truth labels of the Indian Pines dataset.

The Pavia University dataset is another dataset of urban area acquired by ROSIS, the image consists of 103 spectral channels with 610 × 340 pixels covering 9 land cover categories. The false-color composition picture of the Pavia University image and the corresponding ground truth map are shown in Fig. 7. The 10% samples are set to the training set, and the rest are set to the testing set.

The third dataset is Indian Pine dataset [41], which recording 16 crop categories by an airborne visible/infrared imaging spectrometer sensor over the Indian Pines agricultural site. It has 200 spectral bands with 145 × 145 pixels. The false-color composition picture of the Indian Pine data and the corresponding ground truth map are shown in Fig. 8. The 20% samples are set to the training set, and the rest are set to the testing set.

4.2. Sensitivity analysis of proposed method

Learning rate affects the learning steps during training. A too small value may cause a too slow convergence, and a too big one may lead to network oscillation. According to Fig. 9, we choose 0.0005 from {0.01, 0.005, 0.001, 0.0005, 0.0003, 0.0001} as the optimal one. It remains unchanged during the whole training procedure.

Our model utilizes neighbor region to extract spatial information, and its performance badly depends on the size of neighbor regions. We select the patch size from {5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13, 15 × 15} with different number of principle components to find the optimal size of neighbor regions. As shown

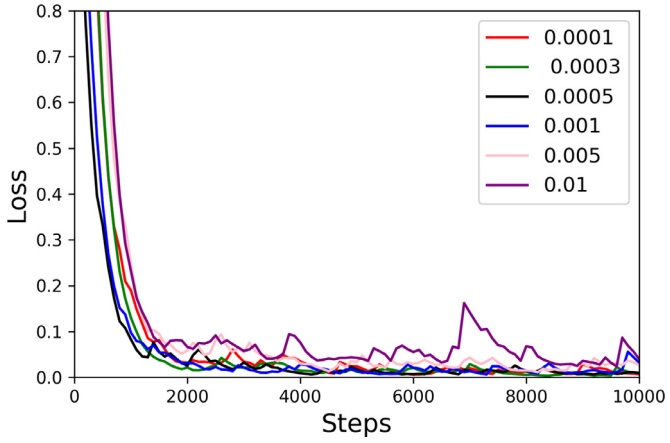

Fig. 9. Illustration of loss with different learning rate.

in Fig. 10, spatial patches with fewer PCs fail to get a higher accuracy due to its poor information, but larger size of path will cause the possibility of heavier computational expense and over-smoothing phenomenon. Thus, the number of PCs is set as 3 and the size of spatial patches is set as 13 to get a better classification accuracy.

The single GRU with spatial initial state we proposed can extract the joint spectral-spatial features, we use t-SNE method to reduce its high dimension for the discriminative ability visualization. As shown in Fig. 11, 2D features are plotted after t-SNE, the separability of different classes is poor, and samples from different class are overlapped with each other. With the increase of iterations in learning spectral-spatial features, different samples from the same classes are gathered into several clusters, so it becomes much easier to separate.

4.3. Classification results

To evaluate effectiveness of the proposed model, we compare it with different classification methods quantitatively and qualitatively. The contrastive methods are summarized as follows: (1) SVM with radial basis function kernel, (2) deep learning method 2DCNN, (3) RNN with band-by-band strategy, (4) RNN with bands-grouping strategy, (5) RNN with all-in-one strategy, (6) spectral-spatial GRU with two branches, (7) spectral-spatial GRU with concatenate input, and (8) spectral-spatial GRU with spatial initial state.

Overall accuracy (OA), average accuracy (AA), and the kappa coefficient are used as the evaluation measurements for the compared methods. The runtime represents the training duration, the batch size is set to 64, and the training steps is set to 10,000. Besides, for a fair comparison, we utilize the same training and testing sets for all methods, and all algorithms are executed ten times, and the mean results are reported to reduce random selection effects. All the experiments are implemented with an NVIDIA RTX 2080Ti GPU, tensorflow-gpu 1.9.0 with python 3.6.

For the Pavia University dataset, as shown in the classification maps with all labeled and unlabeled pixels in Fig. 12 and the results on testing dataset in Table 1, our proposed method effectively surpasses CNN and RNN and obtains better performance. No matter it is a 2D CNN that only focuses on spatial information, or an RNN that only extracts spectral information, the classification results are not good enough because of incomplete information. CNN has a relatively uniform classification map, but its accuracy is worse than that of the network using spectral information. Although RNN has advantages in classification accuracy, due to the lack of spatial distribution information of neighbor regions, the effects of individual categories are poor, such as 56.56% in band-by-band strategy, 62.59% in bands-grouping strategy and 66.15% in all-in-one strategy.

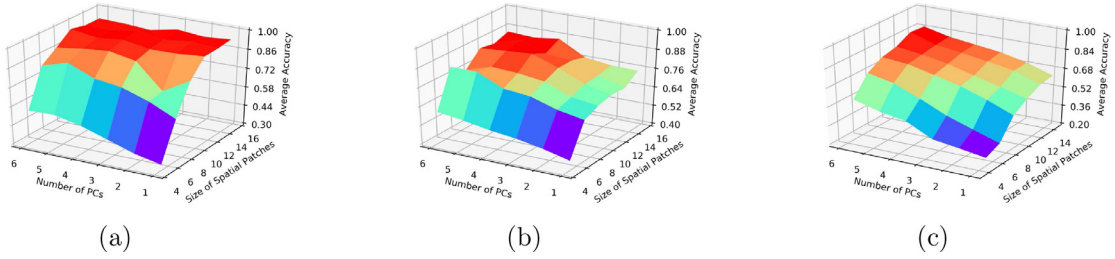

Fig. 10. Illustration of spatial performance with different number of principle components and size of spatial patches. (a) Pavia Center dataset, (b) Pavia University dataset, (c) Indian Pines dataset.

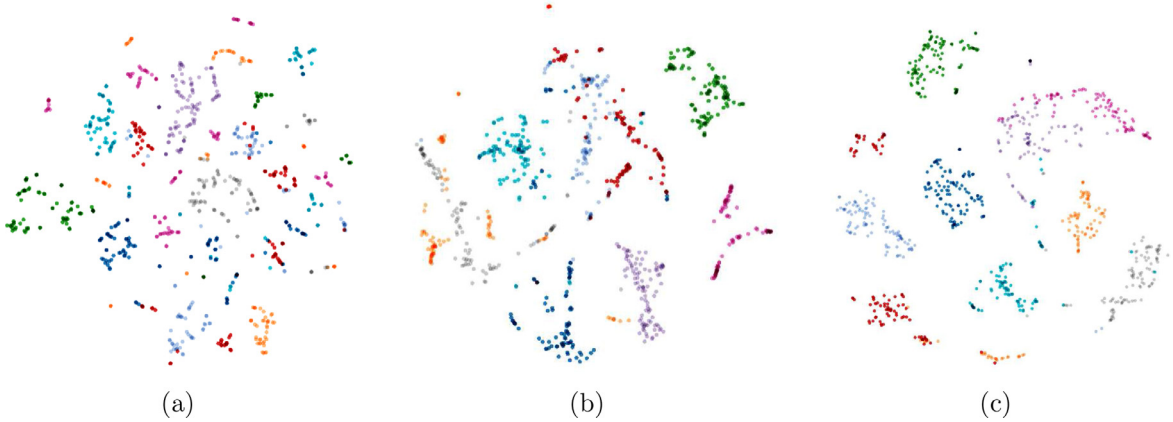

Fig. 11. 2D feature visualization of the joint spectral-spatial features in feature space of the single GRU via t-SNE method. Each points represents one sample, and different colors represent different classes. 100 samples are random chosen from each class to compute the features. (a) raw features, (b) the joint features trained with 500 iterations, (c) the joint features trained with 2000 iterations.

Table 1
Classification performance of different methods for the Pavia University dataset.

Lable	SVM	CNN	RNN(spectral)			GRU(spectral-spatial)		
			(band-by-band)	(bands-grouping)	(all-in-one)	(two branch)	(concat-input)	(spatial initial state)
1	89.91	93.58	87.13	91.57	94.43	91.25	95.09	97.10
2	97.92	95.13	97.88	94.76	98.36	97.75	98.60	99.48
3	69.31	67.67	63.17	48.89	69.89	89.41	86.42	88.41
4	96.23	96.99	89.23	88.46	85.86	96.55	96.88	97.53
5	97.85	99.09	99.34	99.25	99.59	99.42	99.91	99.75
6	58.45	80.00	54.23	66.82	70.88	97.41	92.75	94.94
7	73.26	78.36	57.64	86.13	72.51	93.23	88.22	96.65
8	82.19	86.18	91.58	95.92	90.64	92.93	92.81	97.07
9	94.14	96.60	99.64	99.88	98.01	97.42	98.47	99.76
OA	88.30	88.18	87.06	88.36	89.77	95.70	95.86	98.09
AA	84.36	87.74	82.25	85.74	86.90	95.04	94.35	96.52
kappa	84.23	87.74	82.44	84.41	87.64	94.32	94.52	96.14
Runtime(s)	-	202.77	896.07	218.19	52.51	77.67	65.75	59.96

Table 2
Classification performance of different methods for the Pavia Center dataset.

Lable	SVM	CNN	RNN(spectral)			GRU(spectral-spatial)		
			(band-by-band)	(bands-grouping)	(all-in-one)	(two branch)	(concat-input)	(spatial initial state)
1	99.98	99.97	99.98	99.96	99.94	99.99	99.98	99.99
2	95.20	92.78	97.68	90.89	88.55	96.46	98.08	97.28
3	85.15	95.79	56.56	93.85	95.86	95.25	93.56	97.01
4	53.74	89.32	77.36	62.59	66.15	85.10	88.00	99.08
5	94.07	95.27	95.11	95.61	93.65	99.21	99.37	97.36
6	55.08	87.20	96.99	97.81	98.17	97.86	97.29	97.75
7	96.90	93.99	91.84	89.63	91.51	95.76	98.02	94.68
8	97.15	99.12	99.53	99.43	99.35	99.89	99.64	99.55
9	72.14	98.99	99.88	99.76	99.92	99.96	99.88	99.57
OA	94.02	96.76	97.61	97.69	97.68	99.04	99.10	99.14
AA	83.27	94.71	90.55	92.17	92.56	96.61	97.09	98.03
kappa	91.52	96.83	96.62	96.74	96.72	98.64	98.73	98.78
Runtime(s)	-	183.22	827.25	197.27	44.34	79.64	63.94	53.73

Compared RNN within three input strategies, the results indicate that taking all spectra as input is a better choice for this task since it does not break the inter-spectral correlations. Our method GRU with spatial initial state outperforms other methods and gets the OA as 98.09%, AA as 96.52% and Kappa as 96.19%. And it costs much less time than the original band-by-band RNN and other methods.

Fig. 13 and Table 2 present the classification maps with labeled and unlabeled pixels and the results of the Pavia Center dataset. Although methods like CNN or RNN learn higher-level features through deep network, the difference between OA and AA demonstrates that there are unbalanced and insufficiently results. The results of the three RNN strategies indicate that the results of different forms of input and different numbers of GRUs are not much difference in classification accuracy, but the efficiency is greatly improved by the way extracting the spectral features with a single GRU. The single GRU with spatial initial state can adequately make full use of the entire spectral information, and corrects many misclassified pixels caused by lacking information with the trainable spatial priori as the GRU cell state. The proposed model achieves a higher homogeneous result, specifically, OA, AA and kappa is improved to 99.80%, 96.95% and 99.41%, respectively.

Different from the two datasets of urban areas mentioned above, the Indian Pine data set represents a crop area where has more spatially homogeneous categories. Besides, it has fewer annotations and its labeled categories are quite unbalanced. Fig. 14 and Table 3 report the classification results of labeled and unlabeled pixels from the proposed method along with other methods on Indian Pines dataset. SVM and CNN methods yield uniform but inferior results because of lack of spectral information. Only

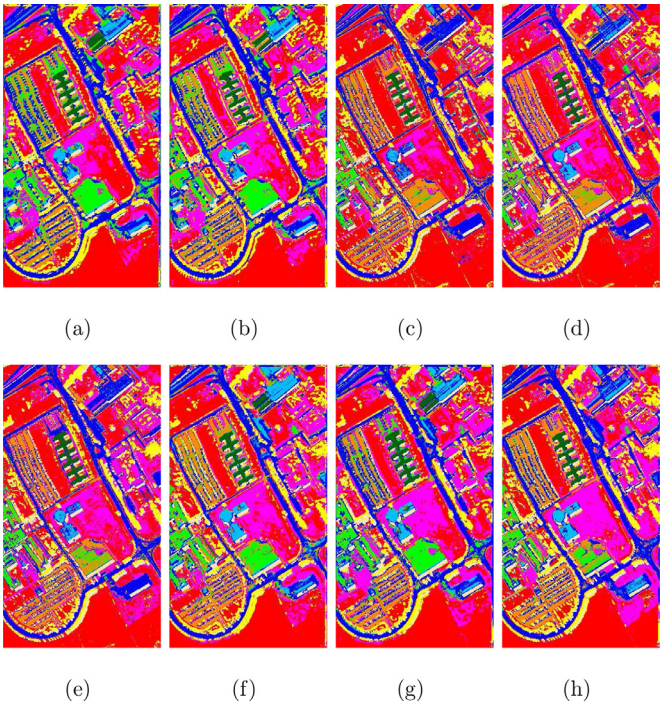

Fig. 12. Visual results on the Pavia University dataset. (a) SVM, (b) 2D-CNN, (c) RNN with band-by-band strategy, (d) RNN with bands-grouping strategy, (e) RNN with all-in-one strategy, (f) spectral-spatial GRU with two branches, (g) spectral-spatial GRU with concatenate input, (h) spectral-spatial GRU with spatial initial state.

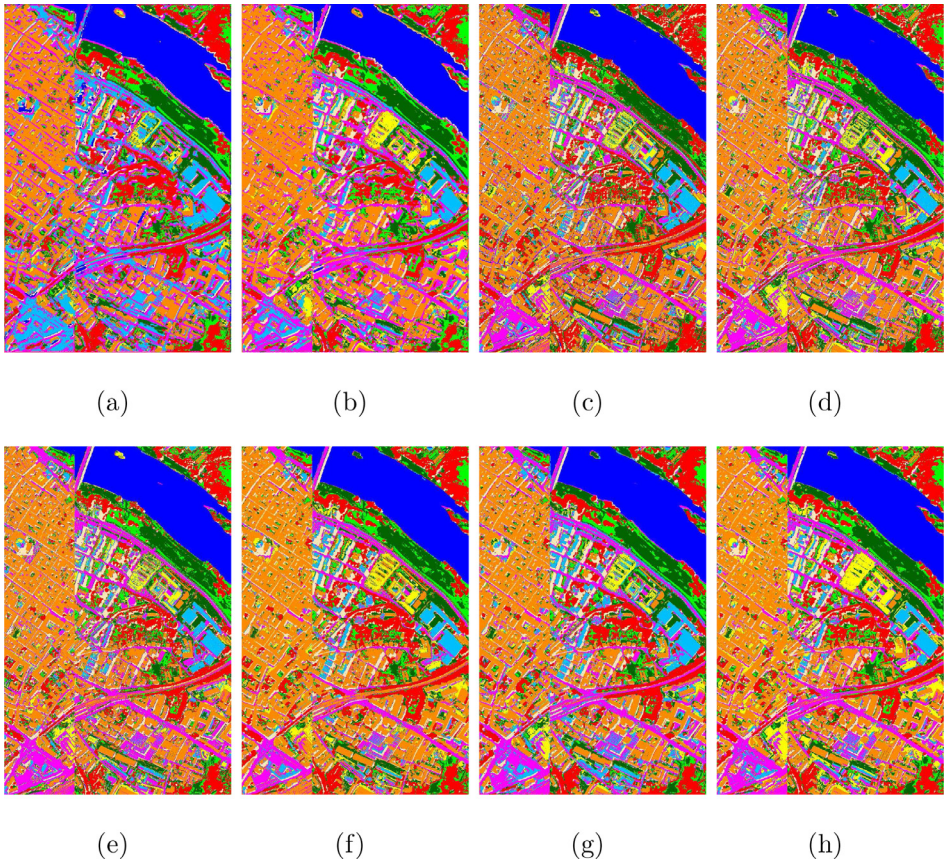

Fig. 13. Visual results on the Pavia Center dataset. (a) SVM, (b) 2D-CNN, (c) RNN with band-by-band strategy, (d) RNN with bands-grouping strategy, (e) RNN with all-in-one strategy, (f) spectral-spatial GRU with two branches, (g) spectral-spatial GRU with concatenate input, (h) spectral-spatial GRU with spatial initial state.

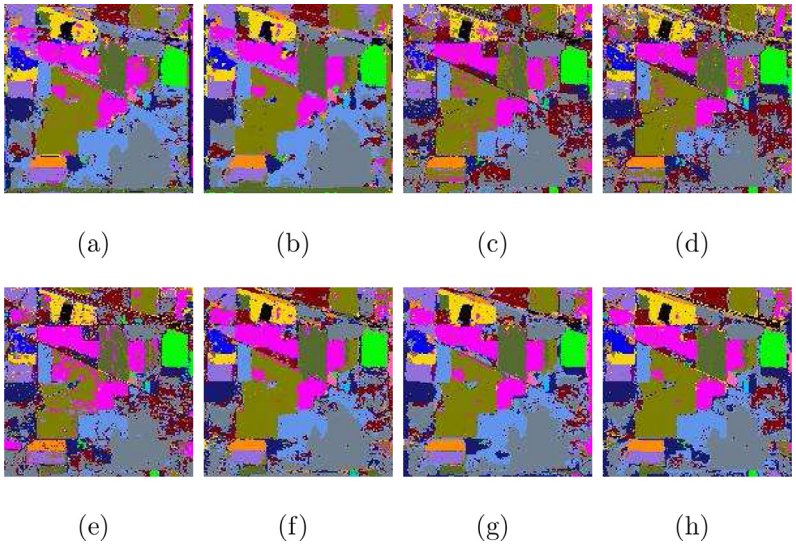

Fig. 14. Visual results on the Pavia Center dataset. (a) SVM, (b) 2D-CNN, (c) RNN with band-by-band strategy, (d) RNN with bands-grouping strategy, (e) RNN with all-in-one strategy, (f) spectral-spatial GRU with two branches, (g) spectral-spatial GRU with concatenate input, (h) spectral-spatial GRU with spatial initial state.

spectral or spatial feature is insufficient for HSI classification. RNN performs unfavorable results in a similar way, the classification maps show uneven areas with many misclassification points, as shown in the classification maps. It can be seen that our proposed spectral-spatial GRU with spatial initial state achieves better performance compared with other methods in terms of OA, AA and Kappa and yields a cleaner classification map.

The classification results of these three datasets show that our proposed method, the single GRU with spatial initial state, exhibits the best performance among all compared methods in all scenarios. The comparisons of the three input strategies of RNN show that the spectral characteristics can be fully expressed in the GRU's feature space by inputting the entire spectrum in a single GRU. All-in-one strategy performs better and costs much less time when it

Table 3
Classification performance of different methods for the Indian Pines dataset.

Lable	SVM	CNN	RNN(spectral)			GRU(spectral-spatial)		
			(band-by-band)	(bands-grouping)	(all-in-one)	(two branch)	(concat-input)	(spatial initial state)
1	78.79	90.91	72.72	36.36	66.67	90.91	96.97	99.98
2	85.91	89.30	76.90	76.91	89.60	92.90	92.50	93.50
3	85.88	90.02	74.21	74.73	80.72	88.98	96.55	95.94
4	74.09	76.51	72.29	84.34	74.46	83.73	83.13	92.77
5	92.03	93.51	92.03	93.51	90.56	93.51	94.39	96.75
6	95.69	98.04	97.45	90.99	96.09	98.82	99.41	99.41
7	45.02	70.02	61.02	60.00	75.02	80.01	95.00	95.50
8	96.12	97.31	99.40	99.70	98.80	99.10	99.40	99.40
9	82.85	99.52	74.28	81.42	85.71	78.57	99.87	99.89
10	75.76	96.62	77.53	69.31	74.89	92.36	96.33	92.07
11	88.62	94.41	82.72	89.59	73.12	95.40	95.05	96.39
12	84.85	85.34	73.07	75.48	76.20	83.89	88.94	93.51
13	86.53	99.31	99.30	98.61	92.36	99.30	99.30	99.31
14	96.84	98.87	96.84	95.59	96.16	98.64	98.75	99.09
15	77.49	83.39	66.46	67.56	69.78	87.45	88.19	93.72
16	90.91	86.36	95.45	96.97	95.45	96.97	99.98	98.48
OA	82.95	93.05	86.57	87.32	88.76	93.73	95.14	96.33
AA	83.96	90.62	80.17	80.19	81.22	91.29	95.25	96.21
kappa	80.54	92.07	81.89	81.69	82.40	92.84	94.45	95.67
Runtime(s)	-	127.33	1525.61	124.48	43.80	75.08	67.48	60.96

comes to inter spectral correlations. The two-branch GRU learns the spectral features and the spatial features separately, and it performs worse than the other two learning the joint spectral-spatial features in the same feature space. In all spectral-spatial strategies, the one which regards spatial information as the initial state has advantage both in performance evaluation and training efficiency.

5. Conclusion

In this study, a tiny effective model is proposed to extract spectral-spatial features for hyperspectral image classification based on a single GRU. We utilize the superior of GRU that the initial state could be a trainable factor. Based on the similarity of neighbor pixels in the spatial domain, we can learn spatial contextual features in spatial dimensions by adding spatial neighbor information as the trainable initial state. Numerous inner spectral correlations in the continuous spectrum domain are extracted with an entire spectra data input. Take both spectral input and spatial prior, the GRU cell can learn a joint feature for pixel-wise classification and perform robustly.

We design contrast experiments on different input modes in GRU of spectral information and multiple ways of fusing spatial information. Experimental results above on three public datasets show that our method not only outperforms other traditional and deep learning methods but also extracts more homogeneous discriminative feature representations. We will generalize our method for other remote sensing applications, such as change detection, in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by the National Natural Science Foundation of China under Grant Nos. 61805181, 61705170, 61903279 and 61605146.

References

[1] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, J. Ma, Hyperspectral image denoising with superpixel segmentation and low-rank representation, Inf. Sci. 397 (2017) 48–68.
[2] J.M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, J. Chanussot, Hyperspectral remote sensing data analysis and future challenges, IEEE Geosci. Remote Sens. Mag. 1 (2) (2013) 6–36.
[3] L. Ma, M.M. Crawford, L. Zhu, Y. Liu, Centroid and covariance alignment-based domain adaptation for unsupervised classification of remote sensing images, IEEE Trans. Geosci. Remote Sens. 57 (4) (2019) 2305–2323.
[4] X. Mei, Y. Ma, C. Li, F. Fan, J. Huang, J. Ma, Robust GBM hyperspectral image unmixing with superpixel segmentation based low rank and sparse representation, Neurocomputing 275 (2018) 2783–2797.
[5] L. Zhang, L. Zhang, B. Du, Deep learning for remote sensing data: a technical tutorial on the state of the art, IEEE Geosci. Remote Sens. Mag. 4 (2) (2016) 22–40.
[6] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J.V. Borre, C. Mücher, Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers, Remote Sens. Environ. 126 (2012) 222–231.
[7] L.G. Olmanson, P.L. Brezonik, M.E. Bauer, Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: the mississippi river and its tributaries in minnesota, Remote Sens. Environ. 130 (2013) 254–265.
[8] H. Lyu, H. Lu, L. Mou, Learning a transferable change rule from a recurrent neural network for land cover change detection, Remote Sens. 8 (6) (2016) 506.
[9] M. Xie, Z. Ji, G. Zhang, T. Wang, Q. Sun, Mutually exclusive-KSVD: Learning a discriminative dictionary for hyperspectral image classification, Neurocomputing 315 (2018) 177–189.
[10] J. Ham, Y. Chen, M.M. Crawford, J. Ghosh, Investigation of the random forest framework for classification of hyperspectral data, IEEE Trans. Geosci. Remote Sens. 43 (3) (2005) 492–501.
[11] B. Ayerdi, M.G. Romay, Hyperspectral image analysis by spectral–spatial processing and anticipative hybrid extreme rotation forest classification, IEEE Trans. Geosci. Remote Sens. 54 (5) (2015) 2627–2639.
[12] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Trans. Geosci. Remote Sens. 42 (8) (2004) 1778–1790.
[13] G. Camps-Valls, L. Bruzzone, Kernel-based methods for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 43 (6) (2005) 1351–1362.
[14] Y. Tarabalka, J.A. Benediktsson, J. Chanussot, J.C. Tilton, Multiple spectral–spatial classification approach for hyperspectral data, IEEE Trans. Geosci. Remote Sens. 48 (11) (2010) 4122–4132.
[15] L. Fang, S. Li, X. Kang, J.A. Benediktsson, Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation, IEEE Trans. Geosci. Remote Sens. 52 (12) (2014) 7738–7749.
[16] C. Li, Y. Ma, X. Mei, C. Liu, J. Ma, Hyperspectral image classification with robust sparse representation, IEEE Geosci. Remote Sens. Lett. 13 (5) (2016) 641–645.
[17] J. Jiang, J. Ma, Z. Wang, C. Chen, X. Liu, Hyperspectral image classification in the presence of noisy labels, IEEE Trans. Geosci. Remote Sens. 57 (2) (2019) 851–865.

[18] H. Lin, J. Li, C. Liu, S. Li, Recent advances on spectral-spatial hyperspectral image classification: an overview and new guidelines, IEEE Trans. Geosci. Remote Sens. 56 (3) (2018) 1579–1597.
[19] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, J.A. Benediktsson, Deep learning for hyperspectral image classification: an overview, IEEE Trans. Geosci. Remote Sens. (2019), doi:10.1109/TGRS.2019.2907932.
[20] J. Ma, W. Yu, P. Liang, C. Li, J. Jiang, Fusiongan: a generative adversarial network for infrared and visible image fusion, Inf. Fusion 48 (2019) 11–26.
[21] G. Zhao, G. Liu, L. Fang, B. Tu, P. Ghamisi, Multiple convolutional layers fusion framework for hyperspectral image classification, Neurocomputing 339 (2019) 149–160.
[22] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neurocomputing 219 (2017) 88–98.
[23] Y. Li, H. Zhang, Q. Shen, Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network, Remote Sens. 9 (1) (2017) 67.
[24] K. Makantasis, K. Karantzalos, A. Doulamis, N. Doulamis, Deep supervised learning for hyperspectral data classification through convolutional neural networks, in: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2015, pp. 4959–4962.
[25] H. Wu, S. Prasad, Convolutional recurrent neural networks for hyperspectral data classification, Remote Sens. 9 (3) (2017) 298.
[26] L. Mou, P. Ghamisi, X.X. Zhu, Deep recurrent neural networks for hyperspectral image classification, IEEE Trans. Geosci Remote Sens. 55 (7) (2017) 3639–3655.
[27] Q. Liu, F. Zhou, R. Hang, X. Yuan, Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification, arXiv:1703.07910 (2017).
[28] R. Hang, Q. Liu, D. Hong, P. Ghamisi, Cascaded recurrent neural networks for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. (2019) 1–11.
[29] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, H. Zhou, Spatial sequential recurrent neural network for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 11 (11) (2018) 4141–4155.
[30] F. Zhou, R. Hang, Q. Liu, X. Yuan, Hyperspectral image classification using spectral-spatial LSTMS, Neurocomputing 328 (2019) 39–47.
[31] H. Lee, H. Kwon, Going deeper with contextual CNN for hyperspectral image classification, IEEE Trans. Image Process. 26 (10) (2017) 4843–4855.
[32] M. Han, R. Cong, X. Li, H. Fu, J. Lei, Joint spatial-spectral hyperspectral image classification based on convolutional neural network, Pattern Recognit. Lett. (2018) in press.
[33] B. Ayerdi, I. Marqués, M. Graña, Spatially regularized semisupervised ensembles of extreme learning machines for hyperspectral image segmentation, Neurocomputing 149 (2015) 373–386.
[34] H. Xu, H. Zhang, W. He, L. Zhang, Superpixel-based spatial-spectral dimension reduction for hyperspectral imagery classification, Neurocomputing (2019).
[35] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 54 (10) (2016) 6232–6251.
[36] Y. Xu, L. Zhang, B. Du, F. Zhang, Spectral-spatial unified networks for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 56 (10) (2018) 5893–5909.
[37] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, J. Ma, Spectral-spatial attention networks for hyperspectral image classification, Remote Sens. 11 (8) (2019) 963.
[38] Z. Yu, G. Liu, Sliced recurrent neural networks, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2953–2964.
[39] S. Hochreiter, S. Jrgen, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[40] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078 (2014).
[41] M.F. Baumgardner, L.L. Biehl, D.A. Landgrebe, 220 band AVIRIS hyperspectral image data set: June 12, 1992 indian pine test site 3, 2015. https://purr.purdue.edu/publications/1947/1.

Erting Pan received the B.S. degree in electrical engineering and its automation from the Northeast Normal University, Changchun, China, in 2018. She is currently pursuing the M.S. degree with the Electronic Information School, Wuhan University, Wuhan. Her current research interests include hyperspectral imagery and deep learning.

Xiaoguang Mei received the B.S. degree in communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2007, the M.S. degree in communications and information systems from Huazhong Normal University, Wuhan, in 2011, and the Ph.D. degree in circuits and systems from the HUST, in 2016. From 2010 to 2012, he was a Software Engineer with the 722 Research Institute, China Shipbuilding Industry Corporation, Wuhan. From 2016 to 2019, He was a Post- Doctoral Fellow with the Electronic Information School, Wuhan University, Wuhan. He is currently an assistant professor with the Electronic Information School, Wuhan University. His research interests include hyperspectral imagery, machine learning, and pattern recognition.

Quande Wang received the B.S. degree in physics and M.S. degree in circuits and systems from the Central China Normal University (CCNU), Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in control science and engineering from the Wuhan University, in 2004. From 2005 to 2007, he was a PostDoctoral Fellow with the Electronic Information School, Wuhan University, Wuhan. Between 2014 and 2015, he was a Visiting Scholar at the University of Texas at Austin, Austin, USA. He is currently an associate professor with the Electronic Information School, Wuhan University. His research interests include computer vision, machine learning, and pattern recognition.

Yong Ma graduated from the Department of Automatic Control, Beijing Institute of Technology, Beijing, China, in 1997. He received the Ph.D. degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003. His general field of research is in signal and systems. His current research projects include remote sensing of the Lidar and infrared, as well as Infrared image processing, pattern recognition, interface circuits to sensors and actuators. Between 2004 and 2006, he was a Lecturer at the University of the West of England, Bristol, U.K. Between 2006 and 2014, he was with the Wuhan National Laboratory for Optoelectronics, HUST, Wuhan, where he was a Professor of electronics. He is now a Professor with the Electronic Information School, Wuhan University.

Jiayi Ma received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He was a Post-Doctoral with the Electronic Information School, Wuhan University from August 2014 to November 2015, and received an accelerated promotion to Associate Professor and Full Professor in December 2015 and December 2018, respectively. He has authored or coauthored more than 120 refereed journal and conference papers, including IEEE TPAMI/TIP/TSP/TNNLS/TIE/TGRS/TCYB/TMM/TCSVT, IJCV, CVPR, ICCV, IJCAI, AAAI, ICRA, IROS, ACM MM, etc. His research interests include computer vision, machine learning, and pattern recognition. Dr. Ma has been identified in the 2019 Highly Cited Researchers list from the Web of Science Group. He was a recipient of the Natural Science Award of Hubei Province (first class), the CAAI (Chinese Association for Artificial Intelligence) Excellent Doctoral Dissertation Award (a total of eight winners in China), and the CAA (Chinese Association of Automation) Excellent Doctoral Dissertation Award (a total of ten winners in China). He is an Editorial Board Member of Information Fusion and Neurocomputing, and a Guest Editor of Remote Sensing.