Neurocomputing 328 (2019) 39–47

Contents lists available at ScienceDirect

Neurocomputing

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

# Hyperspectral image classification using spectral-spatial LSTMs☆

Feng Zhou, Renlong Hang, Qingshan Liu*, Xiaotong Yuan

*Jiangsu Key Laboratory of Big Data Analysis Technology, School of Information and Control, Nanjing University of Information Science and Technology, Nanjing 210044, China*

ARTICLE INFO

*Article history:*
Received 8 November 2017
Revised 16 January 2018
Accepted 13 February 2018
Available online 20 August 2018

*Keywords:*
Deep learning
Long short term memory
Decision fusion
Hyperspectral image classification

ABSTRACT

In this paper, we propose a hyperspectral image (HSI) classification method using spectral-spatial long short term memory (LSTM) networks. Specifically, for each pixel, we feed its spectral values in different channels into Spectral LSTM one by one to learn the spectral feature. Meanwhile, we firstly use principle component analysis (PCA) to extract the first principle component from a HSI, and then select local image patches centered at each pixel from it. After that, we feed the row vectors of each image patch into Spatial LSTM one by one to learn the spatial feature for the center pixel. In the classification stage, the spectral and spatial features of each pixel are fed into softmax classifiers respectively to derive two different results, and a decision fusion strategy is further used to obtain a joint spectral-spatial results. Experimental results on three widely used HSIs (i.e., Indian Pines, Pavia University, and Kennedy Space Center) show that our method can improve the classification accuracy by at least 2.69%, 1.53% and 1.08% compared to other state-of-the-art methods.



## 1. Introduction

With the development of hyperspectral sensors, it is convenient to acquire images with high spectral and spatial resolutions simultaneously. Hyperspectral data is becoming a valuable tool to monitor the Earth's surface. They have been widely used in agriculture, mineralogy, physics, astronomy, chemical imaging, and environmental sciences [1]. For these applications, an essential step is image classification whose purpose is to identify the label of each pixel [2].

Many methods have been proposed to deal with hyperspectral image (HSI) classification. Traditional methods, such as k-nearest-neighbors and logistic regression, often use the high-dimensional spectral information as features, thus suffering from the issue of "curse of dimensionality" [3]. To address this issue, dimensionality reduction methods are widely used. These methods include principal component analysis (PCA) [4,5] and linear discriminant analysis (LDA) [6–8]. In [9], a promising method called support vector machine (SVM) was successfully applied to HSI classification. It exhibits low sensitivity to the data with high dimensionality and small sample size. In most cases, SVM-based classifiers can obtain superior performance as compared to other methods. However, SVM is still a shallow architecture. As discussed in [10], these shallow architectures have shown effectiveness in solving many simple or well-constrained problems, but their limited modeling and representational power are insufficient in complex scene cases.

In the past few years, with the advances of the computing power of computers and the availability of large-scale datasets, deep learning techniques [11] have gained great success in a variety of machine learning tasks. Among these techniques, CNN [12,13] has been recognized as a state-of-the-art feature extraction method for various computer vision tasks [14–16] owing to its local connections and weight sharing properties. Besides, recurrent neural network (RNN) [17,18] and its variants have been widely used in sequential data modeling such as speech recognition [19,20] and machine translation [21,22].

Recently, deep learning has been introduced into the remote sensing community especially for HSI classification [23–29]. For example, in [1], a stacked autoencoder model was proposed to extract high-level features in an unsupervised manner. Inspired from it, Tao *et al.* proposed an improved autoencoder model by adding a regularization term into the energy function [30]. In [31], deep belief network (DBN) was applied to extract features and classification results were obtained by logistic regression classifier. For these models, inputs are high-dimensional vectors. Therefore, to learn the spatial feature from HSIs, an alternative method is flattening a local image patch into a vector and then feeding it into them. However, this method may destroy the two-dimensional structure of images, leading to the loss of spatial information. To address this issue, a two dimensional CNN model was proposed in [32]. Due to the use of the first principal component of HSIs as input, two

☆ This paper was presented in part at the CCF Chinese Conference on Computer Vision, Tianjin, 2017. This paper was recommended by the program committee.

* Corresponding author.
*E-mail address:* qsliu@nuist.edu.cn (Q. Liu).

https://doi.org/10.1016/j.neucom.2018.02.105
0925-2312/

dimensional CNN may lose the spectral information. To simultaneously learn the spectral and spatial features, three-dimensional CNN considers the local cubes as inputs [33].

Since hyperspectral data are densely sampled from the entire spectrum, they are expected to have dependencies between different spectral bands. First, it is easy to observe that for any material, the adjacent spectral bands tend to have very similar values, which implies that adjacent spectral bands are highly dependent on each other. In addition, some materials also demonstrate long-term dependency between non-adjacent spectral bands [34]. In this paper, we regard each hyperspectral pixel as a data sequence and use long short term memory (LSTM) [35] to model the dependency in the spectral domain. Similar to spectral channels, pixels of the image also depend on each other in the spatial domain. Thus, we can also use LSTM to extract spatial features. The extracted spectral and spatial features for each pixel are then fed into softmax classifiers. The classification results can be combined to derive a joint spectral-spatial result.

The rest of this paper is structured as follows. In the following section, we will introduce the basic knowledge about LSTM. In Section 3, we will present the proposed method in detail. The experiments are reported in Section 4, followed by the conclusion in Section 5.

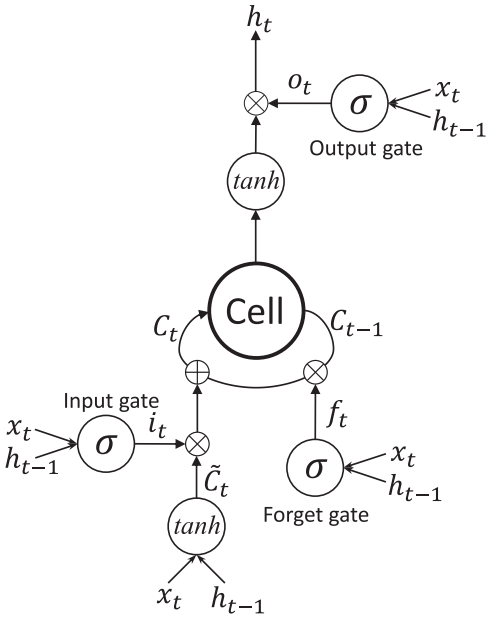


**Fig. 1.** Memory cell of LSTM.

## 2. Long short term memory

RNN has been well acknowledged as a powerful network to address the sequence learning problem by adding recurrent edges to connect the neuron to itself across time. Assume that we have an input sequence $\{x_1, x_2, \ldots, x_T\}$ and a sequence of hidden states $\{h_1, h_2, \ldots, h_T\}$. At a given time $t$, the node with recurrent edge receives the input $x_t$ and its previous output value $h_{t-1}$ at time $t-1$, then outputs the weighted sum of them, which can be formulated as follows:

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b) \tag{1}$$

where $W_{hx}$ is the weight between the input node and the recurrent hidden node, $W_{hh}$ is the weight between the recurrent hidden node and itself from the previous time step, $b$ and $\sigma$ are bias and nonlinear activation function, respectively.

However, there exists an issue when training RNN models. As can be seen from Eq. (1), the contribution of recurrent hidden node $h_m$ at time $m$ to itself $h_n$ at time $n$ may approach infinity or zero as $n-m$ increases whether $|W_{hh}| < 1$ or $|W_{hh}| > 1$. This will lead to the gradient vanishing and exploding problem [36] when back-propagating errors across many time steps. Therefore, it is difficult to learn long range dependencies with RNN. To address this issue, LSTM was proposed to replace the recurrent hidden node by a memory cell shown in Fig. 1 where '$\otimes$' and '$\oplus$' represent dot product and matrix addition, respectively. The memory cell contains a node with a self-connected recurrent edge of a fixed weight one, ensuring that the gradient can pass across many time steps without vanishing or exploding [37]. LSTM unit consists of four important parts: input gate, output gate, forget gate, and candidate cell value. Based on these parts, memory cell and output can be computed by:

$$\begin{aligned}
f_t &= \sigma(W_{hf} \cdot h_{t-1} + W_{xf} \cdot x_t + b_f) \\
i_t &= \sigma(W_{hi} \cdot h_{t-1} + W_{xi} \cdot x_t + b_i) \\
\tilde{C}_t &= \tanh(W_{hC} \cdot h_{t-1} + W_{xC} \cdot x_t + b_C) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\
o_t &= \sigma(W_{ho} \cdot h_{t-1} + W_{xo} \cdot x_t + b_o) \\
h_t &= o_t \circ \tanh(C_t)
\end{aligned} \tag{2}$$

where $\sigma$ is the logistic sigmoid function, '$\cdot$' is a matrix multiplication, '$\circ$' is a dot product, and $b_f$, $b_i$, $b_C$ and $b_o$ are bias terms. The weight matrix subscripts have the obvious meanings. For instance, $W_{hi}$ is the hidden-input gate matrix, $W_{xo}$ is the input-output gate matrix etc.

## 3. Methodology

The flowchart of the proposed spectral-spatial LSTMs (SSLSTMs) is shown in Fig. 2. From this figure, we can observe that SSLSTMs consist of two important components: Spectral LSTM (SeLSTM) and Spatial LSTM (SaLSTM). For each pixel in a given HSI, we feed its spectral values into the SeLSTM to learn the spectral feature and then derive a classification result. Similarly, for the local patch of each pixel, we feed it into a SaLSTM to extract the spatial feature and then obtain a classification result. To fuse the spectral-spatial results, we finally combine these two classification results in a weighted sum manner. In the following subsections, we will introduce these processes in detail.

### 3.1. Spectral LSTM

Hundreds of spectral bands in HSIs provide different spectral characteristics of the object in the same location. Due to the complex situation of lighting, rotations of the sensor, different atmospheric scattering conditions and so on, spectra have complex variations. Therefore, we need to extract robust and invariant features for classification. It is believed that deep architectures can potentially lead to progressively more abstract features at higher layers, and more abstract features are generally invariant to most local changes of the input. In this paper, we consider the spectral values in different channels as an input sequence and use LSTM discussed above to extract spectral features for HSI classification. Fig. 3 shows the flowchart of the proposed classification scheme with spectral features. First, we choose the pixel vector $x_i \in \mathbf{R}^{1\times K}$ where $K$ indicates the number of spectral bands from a given HSI. Second, we transform the vector to a $K$-length sequence $\{x_i^1, \ldots, x_i^k, \ldots, x_i^K\}$ where $x_i^k \in \mathbf{R}^{1\times 1}$ indicates the pixel value of $k$-th spectral band. Then, the sequence is fed into LSTM one by one and the last output is fed to softmax classifier. We set the loss function to cross

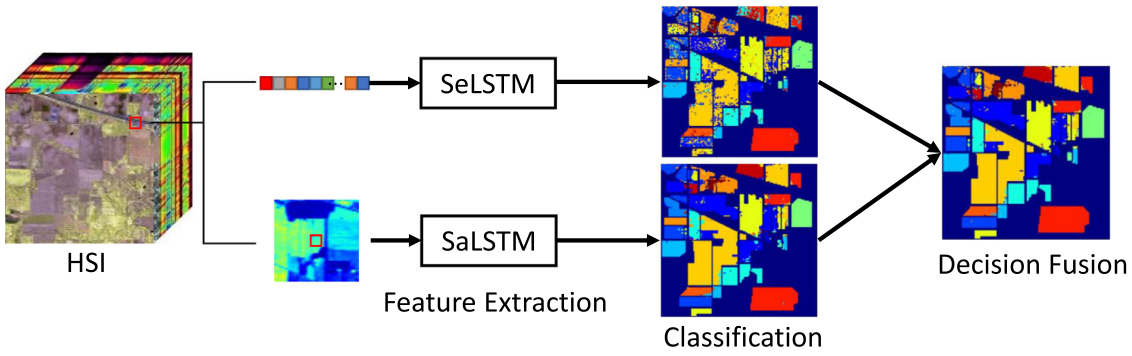


**Fig. 2.** Flowchart of the proposed SSLSTMs.

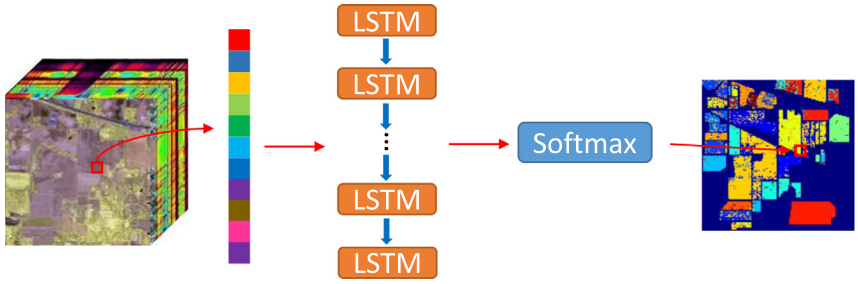


**Fig. 3.** Flowchart of SeLSTM.

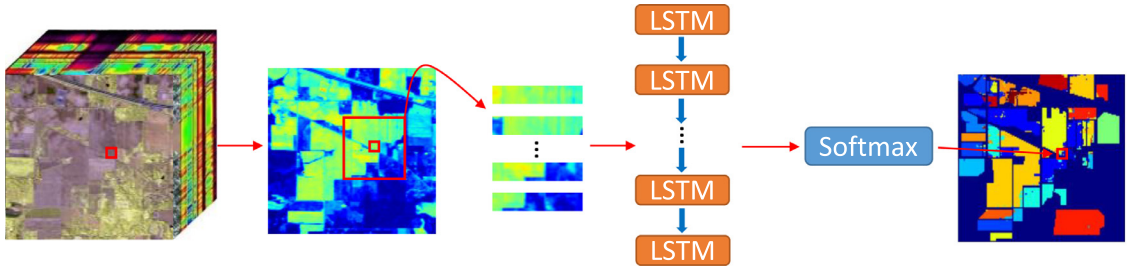


**Fig. 4.** Flowchart of SaLSTM.

entropy $CE = -\sum \mathrm{Y} \log \tilde{Y}$, where Y and $\tilde{Y}$ represent the real label and the predicted label of a pixel, respectively. This loss function can be optimized by Adam algorithm [38]. Finally, we can obtain the probability value $P_{spe}(y = j|x_i), j \in \{1, 2, \ldots, C\}$ where $C$ indicates the number of classes.

### 3.2. Spatial LSTM

To extract the spatial feature of a specific pixel, we take a neighborhood region of it into consideration. Due to the hundreds of channels along the spectral dimension, it always has tens of thousands of dimensions. A large neighborhood region will result in too large input dimension for the classifier, containing too large amount of redundancy [1]. Motivated by the works in [1,32], we firstly use PCA to extract the first principle component. Second, for a given pixel $x_i$, we choose a neighborhood $\mathbf{X}_i \in \mathbf{R}^{S \times S}$ centered at it. After that, we transform the rows in this neighborhood to a $S$-length sequence $\{X_i^1, \ldots, X_i^l, \ldots, X_i^S\}$ where $X_i^l$ indicates the $l$th row of $\mathbf{X}_i$. Finally, we feed the sequence into LSTM to extract the spatial feature of $x_i$. Similar to spectral features-based classification, we use the last output of LSTM as an input to the softmax layer and achieve the probability value $P_{spa}(y = j|x_i), j \in \{1, 2, \ldots, C\}$. The configurations of loss function and optimization algorithm in SaLSTM are the same as those of SeLSTM. The overall flowchart of the proposed spatial features-based classification method is demonstrated in Fig. 4.

### 3.3. Joint spectral-spatial classification

The above two subsections introduce the classification methods based on spectral and spatial features respectively. With the development of imaging spectroscopy technologies, current sensors can acquire HSIs with very high spatial resolutions. Therefore, the pixels in a small spatial neighborhood belong to the same class with a high probability. For a large homogeneous region, the pixels may have different spectral responses. If we only use the spectral features, the pixels will be classified into different subregions. On the contrary, for multiple neighboring regions, if we only use the spatial information, these regions will be classified as the same one. Thus, for accurate classifications, it is essential to take into account the spatial and spectral information simultaneously [8]. Based on the posterior probabilities $P_{spe}(y = j|x_i)$ and $P_{spa}(y = j|x_i)$, an intuitive method to combine the spectral and spatial feature is to fuse these two results in a weighted sum manner, which can be formulated as $P(y = j|x_i) = w_{spe}P_{spe}(y = j|x_i) + w_{spa}P_{spa}(y = j|x_i)$, where $w_{spe}$ and $w_{spa}$ are fusion weights that satisfy $w_{spe} + w_{spa} = 1$. For simplicity, we use uniform weights in our implementation, i.e., $w_{spe} = w_{spa} = \frac{1}{2}$.

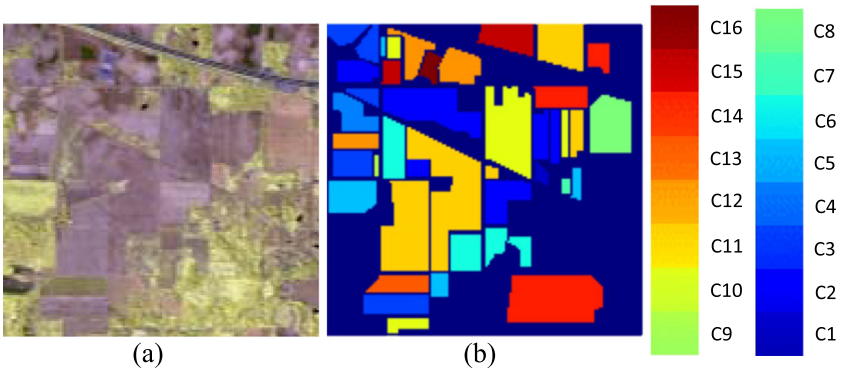


**Fig. 5.** Indian Pines scene dataset. (a) False-color composite image (b) Ground-truth map containing 16 mutually exclusive land cover classes.

**Table 1**
Number of pixels for training/testing and the total number of pixels for each class in IP ground truth map.

| No. | Class | Total | Training | Test |
|---|---|---|---|---|
| C1 | Alfalfa | 46 | 5 | 41 |
| C2 | Corn-notill | 1428 | 143 | 1285 |
| C3 | Corn-mintill | 830 | 83 | 747 |
| C4 | Corn | 237 | 24 | 213 |
| C5 | Grass-pasture | 483 | 48 | 435 |
| C6 | Grass-trees | 730 | 73 | 657 |
| C7 | Grass-pasture-mowed | 28 | 3 | 25 |
| C8 | Hay-windrowed | 478 | 48 | 430 |
| C9 | Oats | 20 | 2 | 18 |
| C10 | Soybean-notill | 972 | 97 | 875 |
| C11 | Soybean-mintill | 2455 | 246 | 2209 |
| C12 | Soybean-clean | 593 | 59 | 534 |
| C13 | Wheat | 205 | 21 | 184 |
| C14 | Woods | 1265 | 127 | 1138 |
| C15 | Buildings-Grass-Trees-Drives | 386 | 39 | 347 |
| C16 | Stone-Steel-Towers | 93 | 9 | 84 |

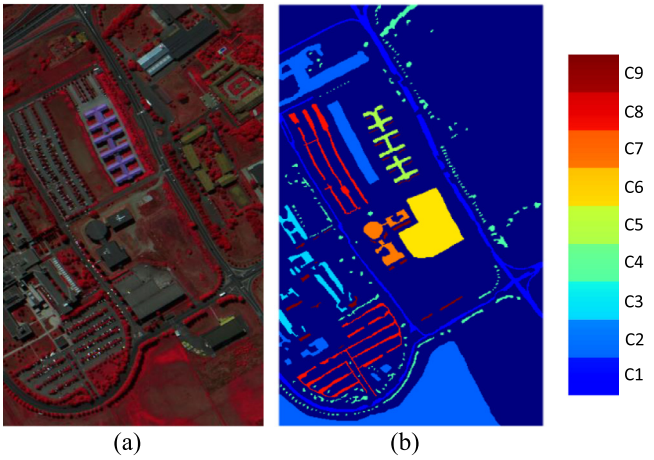


**Fig. 6.** Pavia University scene dataset. (a) False-color composite image (b) Ground-truth map containing 9 mutually exclusive land cover classes.

## 4. Experimental results

### 4.1. Datasets

We test the proposed method on three famous HSI datasets, which are widely used to evaluate classification algorithms.

- Indian Pines (IP): The first dataset was acquired by the AVIRIS sensor over the Indian Pine test site in northwestern Indiana, USA, on June 12, 1992 and it contains 224 spectral bands. We utilize 200 bands after removing four bands containing zero values and 20 noisy bands affected by water absorption. The spatial size of the image is $145 \times 145$ pixels, and the spatial resolution is 20 m. The false-colour composite image and the ground-truth map are shown in Fig. 5. The available number of samples is 10249 ranging from 20 to 2455 in each class, which is reported in Table 1.
- Pavia University (PUS): The second dataset 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 m to 0.86 m. After removing noisy bands, 103 bands are used. The image size is $610 \times 340$ pixels with a spatial resolution of 1.3m. A three band false-colour composite image and the ground-truth map are shown in Fig. 6. In the ground-truth map, there are nine classes of land covers with more than 1000 labeled pixels for each class shown in Table 2.
- Kennedy Space Center (KSC): The third dataset was acquired by the AVIRIS sensor over Kennedy Space Center, Florida, on March 23, 1996. It contains 224 spectral bands. We utilize 176 bands of them after removing bands with water absorption and low signal noise ratio. The spatial size of the image is $512 \times 614$ pixels, and the spatial resolution is 18m. Discriminating different land covers in this dataset is difficult due to the similarity of spectral signatures among certain vegetation types. For classification purposes, thirteen classes representing the various land-cover types that occur in this environment are defined. Fig. 7 demonstrates a false-colour composite image and the ground-truth map. The numbers of pixels for training and testing are shown in Table 3.

**Table 2**
Number of pixels for training/testing and the total number of pixels for each class in PUS ground truth map.

| No. | Class | Total | Training | Test |
|---|---|---|---|---|
| C1 | Asphalt | 6631 | 548 | 6083 |
| C2 | Meadows | 18,649 | 540 | 18,109 |
| C3 | Gravel | 2099 | 392 | 1707 |
| C4 | Trees | 3064 | 524 | 2540 |
| C5 | Painted metal sheets | 1345 | 265 | 1080 |
| C6 | Bare Soil | 5029 | 532 | 4497 |
| C7 | Bitumen | 1330 | 375 | 955 |
| C8 | Self-Blocking Bricks | 3682 | 514 | 3168 |
| C9 | Shadows | 947 | 231 | 716 |

### 4.2. Experimental setup

To demonstrate the effectiveness of the proposed LSTM-based classification method, we quantitatively and qualitatively evaluate the performance of SeLSTM, SaLSTM and SSLSTMs. Besides,

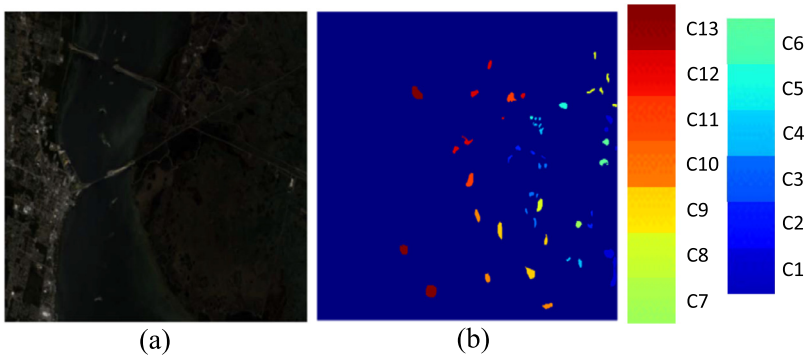


**Fig. 7.** Kennedy Space Center dataset. (a) False-color composite image (b) Ground-truth map containing 13 mutually exclusive land cover classes.

**Table 3**
Number of pixels for training/testing and the total number of pixels for each class in KSC ground truth map.

| No. | Class | Total | Training | Test |
|---|---|---|---|---|
| C1 | Scrub | 761 | 76 | 685 |
| C2 | Willow swamp | 243 | 24 | 219 |
| C3 | Cabbage palm hammock | 256 | 26 | 230 |
| C4 | Cabbage palm/oak hammock | 252 | 25 | 227 |
| C5 | Slash pine | 161 | 16 | 145 |
| C6 | Oak/broadleaf hammock | 229 | 23 | 206 |
| C7 | Hardwood swamp | 105 | 11 | 94 |
| C8 | Graminoid marsh | 431 | 43 | 388 |
| C9 | Spartina marsh | 520 | 52 | 468 |
| C10 | Cattail marsh | 404 | 40 | 364 |
| C11 | Salt marsh | 419 | 42 | 377 |
| C12 | Mud flats | 503 | 50 | 453 |
| C13 | Water | 927 | 93 | 834 |

we compare them with several state-of-the-art methods, including PCA, LDA, non-parametric weighted feature extraction (NWFE) [39], regularized local discriminant embedding (RLDE) [40], matrix-based discriminant analysis (MDA) [41] and CNN [33]. We also directly use the original pixels as a benchmark. For LDA, the within-class scatter matrix $\mathbf{S}_W$ is replaced by $\mathbf{S}_W + \varepsilon\mathbf{I}$, where $\varepsilon = 10^{-3}$, to alleviate the singular problem. The optimal reduced dimensions for PCA, LDA, NWFE and RLDE are chosen from [2,30]. For MDA, the optimal window size is selected from a given set {3, 5, 7, 9, 11}. For CNN, the number of layers and the size of filters are the same as the network in [33]. For LSTM, we only use one hidden layer, and the number of optimal hidden nodes are selected from a given set {16, 32, 64, 128, 256}.

For IP and KSC datesets, we randomly select 10% pixels from each class as the training set, and the remaining pixels as the testing set. For PUS dataset, we randomly choose 3921 pixels as the training set and the rest of pixels as the testing set [41]. The detailed numbers of training and testing samples are listed in Tables 1–3. In order to reduce the effects of random selection, all the algorithms are repeated five times and the average results are reported. The classification performance is evaluated by overall accuracy (OA), average accuracy (AA), per-class accuracy, and Kappa coefficient $\kappa$. OA defines the ratio between the number of correctly classified pixels to the total number of pixels in the testing 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.

### 4.3. Parameter selection

There are two important parameters in the proposed classification framework, including the size of neighborhood regions and the number of hidden nodes. Firstly, we fix the number of hidden nodes and select the optimal region size from a given set $\{8\times 8,\ 16\times 16,\ 32\times 32,\ 64\times 64\}$. Table 4 demonstrates OAs of the SSLSTMs method on three datasets. From this Table, we can observe that as the region size increases, OA will firstly increase and then decrease on PUS dataset. Therefore, the optimal size is chosen as $32\times 32$. For IP and KSC datasets, OA will increase as the size increases. However, larger sizes significantly increase the computation time. Thus, we set the optimal size as $64\times 64$ for IP and KSC datasets.

**Table 4**
OA(%) of the SSLSTMs with different size of neighborhood regions.

| Dataset | Size of neighborhood regions | | | |
|---|---|---|---|---|
| | $8\times 8$ | $16\times 16$ | $32\times 32$ | $64\times 64$ |
| IP | 75.19 | 85.59 | 91.75 | 94.83 |
| PUS | 93.10 | 96.82 | 97.38 | 97.17 |
| KSC | 82.58 | 92.22 | 94.20 | 94.95 |

**Table 5**
OA(%) of SeLSTM and SaLSTM with different numbers of hidden nodes.

| Dataset | Number of hidden nodes | | | |
|---|---|---|---|---|
| | {16, 32} | {32, 64} | {64, 128} | {128, 256} |
| IP | 90.06 | 92.83 | 94.83 | 93.44 |
| PUS | 91.80 | 95.91 | 97.38 | 98.14 |
| KSC | 90.90 | 91.94 | 93.75 | 94.95 |

Secondly, we fix the region size and search for the optimal number of hidden nodes for SeLSTM and SaLSTM from four different combinations {16, 32}, {32, 64}, {64, 128} and {128, 256}. As shown in Table 5, when the number of hidden nodes for SeLSTM and SaLSTM are set to 64 and 128 respectively, the SSLSTMs method achieves the highest OA on IP dataset. Similarly, we can see that SSLSTMs obtains the highest OA on PUS and KSC datasets when the number of hidden nodes for SeLSTM and SaLSTM are set to 128 and 256 respectively.

### 4.4. Performance comparison

Table 6 reports the quantitative results acquired by ten methods on IP dataset. From these results, we can observe that PCA achieves the lowest OA among ten methods, mainly because PCA directly extracts spectral features for classification without considering spatial features. Although LDA and NWFE are still spectral-based methods, they achieve better results than PCA due to the use of label information in training samples. Besides, MDA achieves better performance than the other LDA-related methods which

**Table 6**
OA, AA, per-class accuracy (%), and $\kappa$ performed by ten methods on IP dataset using 10% pixels from each class as the training set.

| Label | Original | PCA | LDA | NWFE | RLDE | MDA | CNN | SeLSTM | SaLSTM | SSLSTMs |
|---|---|---|---|---|---|---|---|---|---|---|
| OA | 77.44 | 72.58 | 76.67 | 78.47 | 80.97 | 92.31 | 90.14 | 72.22 | 91.72 | 95.00 |
| AA | 74.94 | 70.19 | 72.88 | 76.08 | 80.94 | 89.54 | 85.66 | 61.72 | 83.51 | 91.69 |
| $\kappa$ | 74.32 | 68.58 | 73.27 | 75.34 | 78.25 | 91.21 | 88.73 | 68.24 | 90.56 | 94.29 |
| C1 | 56.96 | 59.57 | 63.04 | 62.17 | 64.78 | 73.17 | 71.22 | 25.85 | 85.85 | 88.78 |
| C2 | 79.75 | 68.75 | 72.04 | 76.27 | 78.39 | 93.48 | 90.10 | 66.60 | 89.56 | 93.76 |
| C3 | 66.60 | 53.95 | 57.54 | 59.64 | 68.10 | 84.02 | 91.03 | 54.83 | 91.43 | 92.42 |
| C4 | 59.24 | 55.19 | 46.58 | 59.83 | 70.80 | 83.57 | 85.73 | 43.94 | 90.61 | 86.38 |
| C5 | 90.31 | 83.85 | 91.76 | 88.49 | 92.17 | 96.69 | 83.36 | 83.45 | 88.60 | 89.79 |
| C6 | 95.78 | 91.23 | 94.41 | 96.19 | 94.90 | 99.15 | 91.99 | 87.76 | 90.81 | 97.41 |
| C7 | 80.00 | 82.86 | 72.14 | 82.14 | 85.71 | 93.60 | 85.60 | 23.20 | 51.20 | 84.80 |
| C8 | 97.41 | 93.97 | 98.74 | 99.04 | 99.12 | 99.91 | 97.35 | 95.40 | 99.02 | 99.91 |
| C9 | 35.00 | 34.00 | 26.00 | 44.00 | 73.00 | 63.33 | 54.45 | 30.00 | 38.89 | 74.44 |
| C10 | 66.32 | 64.18 | 60.91 | 69.18 | 69.73 | 82.15 | 75.38 | 71.29 | 88.64 | 95.95 |
| C11 | 70.77 | 74.96 | 76.45 | 77.78 | 79.38 | 92.76 | 94.36 | 75.08 | 94.62 | 96.93 |
| C12 | 64.42 | 41.72 | 67.45 | 64.05 | 72.28 | 91.35 | 78.73 | 54.49 | 86.10 | 89.18 |
| C13 | 95.41 | 93.46 | 96.00 | 97.56 | 97.56 | 99.13 | 95.98 | 91.85 | 90.11 | 98.48 |
| C14 | 92.66 | 89.45 | 93.79 | 93.49 | 92.36 | 98.22 | 96.80 | 90.37 | 98.10 | 98.08 |
| C15 | 60.88 | 47.77 | 65.54 | 58.50 | 67.10 | 87.84 | 96.54 | 30.49 | 88.59 | 92.85 |
| C16 | 87.53 | 88.17 | 83.66 | 89.03 | 89.68 | 94.29 | 81.90 | 62.86 | 64.05 | 87.86 |

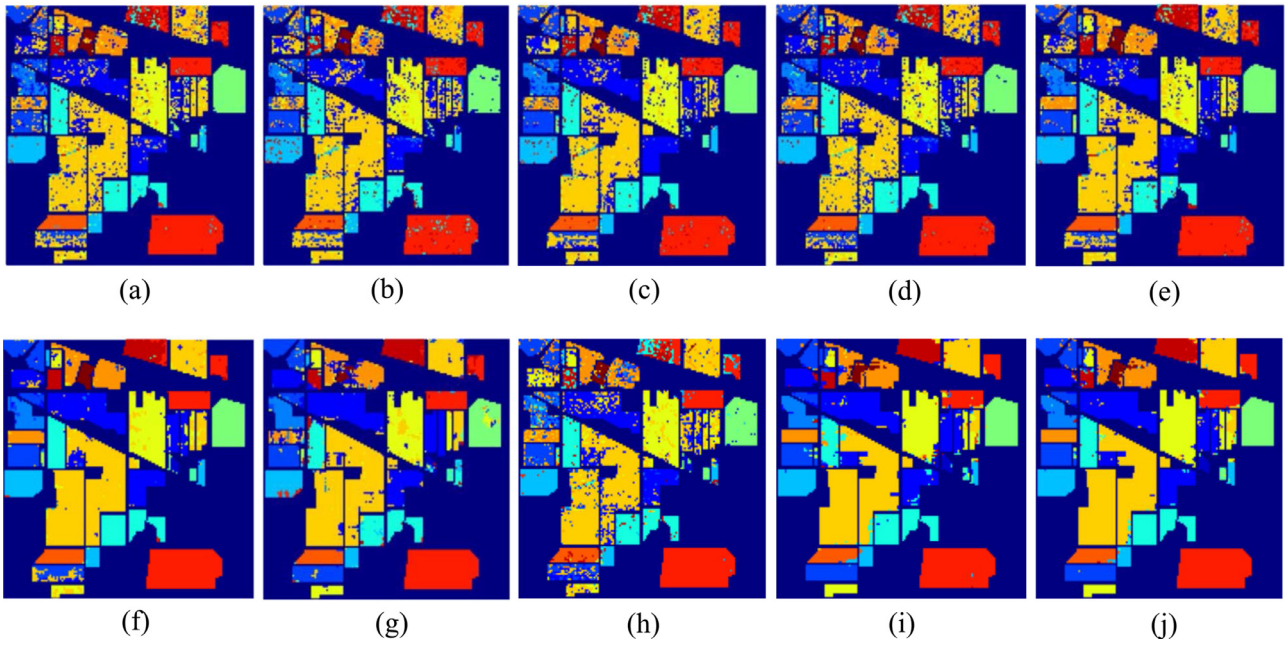


**Fig. 8.** Classification maps on the IP dataset. (a) Original. (b) PCA. (c) LDA. (d) NWFE. (e) RLDE. (f) MDA. (g) CNN. (h) SeLSTM. (i) SaLSTM. (j) SSLSTMs.

**Table 7**
OA, AA, per-class accuracy (%), and $\kappa$ performed by ten methods on PUS dataset using 3921 pixels as the training set.

| Label | Original | PCA | LDA | NWFE | RLDE | MDA | CNN | SeLSTM | SaLSTM | SSLSTMs |
|---|---|---|---|---|---|---|---|---|---|---|
| OA | 89.12 | 88.63 | 84.08 | 88.73 | 88.82 | 96.95 | 96.55 | 93.20 | 94.98 | 98.48 |
| AA | 90.50 | 90.18 | 87.23 | 90.38 | 90.45 | 96.86 | 97.19 | 93.13 | 94.86 | 98.51 |
| $\kappa$ | 85.81 | 85.18 | 79.59 | 85.31 | 85.43 | 95.93 | 95.30 | 90.43 | 92.84 | 97.56 |
| C1 | 87.25 | 87.07 | 82.91 | 86.86 | 87.20 | 96.69 | 96.72 | 91.33 | 92.20 | 96.83 |
| C2 | 89.10 | 88.38 | 80.68 | 88.50 | 88.40 | 97.76 | 96.31 | 94.58 | 95.86 | 98.74 |
| C3 | 81.99 | 81.96 | 69.21 | 82.20 | 81.69 | 90.69 | 97.15 | 83.93 | 92.42 | 96.57 |
| C4 | 95.65 | 95.14 | 95.99 | 95.27 | 95.79 | 98.44 | 96.16 | 97.78 | 91.59 | 98.43 |
| C5 | 99.76 | 99.76 | 99.90 | 99.81 | 99.87 | 100.00 | 99.81 | 99.46 | 98.70 | 99.94 |
| C6 | 88.78 | 88.06 | 89.53 | 88.16 | 88.67 | 96.26 | 94.87 | 91.73 | 96.91 | 99.43 |
| C7 | 85.92 | 85.32 | 81.11 | 86.57 | 86.06 | 97.95 | 97.44 | 90.76 | 98.74 | 99.31 |
| C8 | 86.14 | 86.06 | 85.81 | 86.13 | 86.42 | 93.98 | 98.23 | 88.78 | 94.79 | 97.98 |
| C9 | 99.92 | 99.92 | 99.92 | 99.89 | 99.94 | 100.00 | 98.04 | 99.83 | 92.54 | 99.39 |

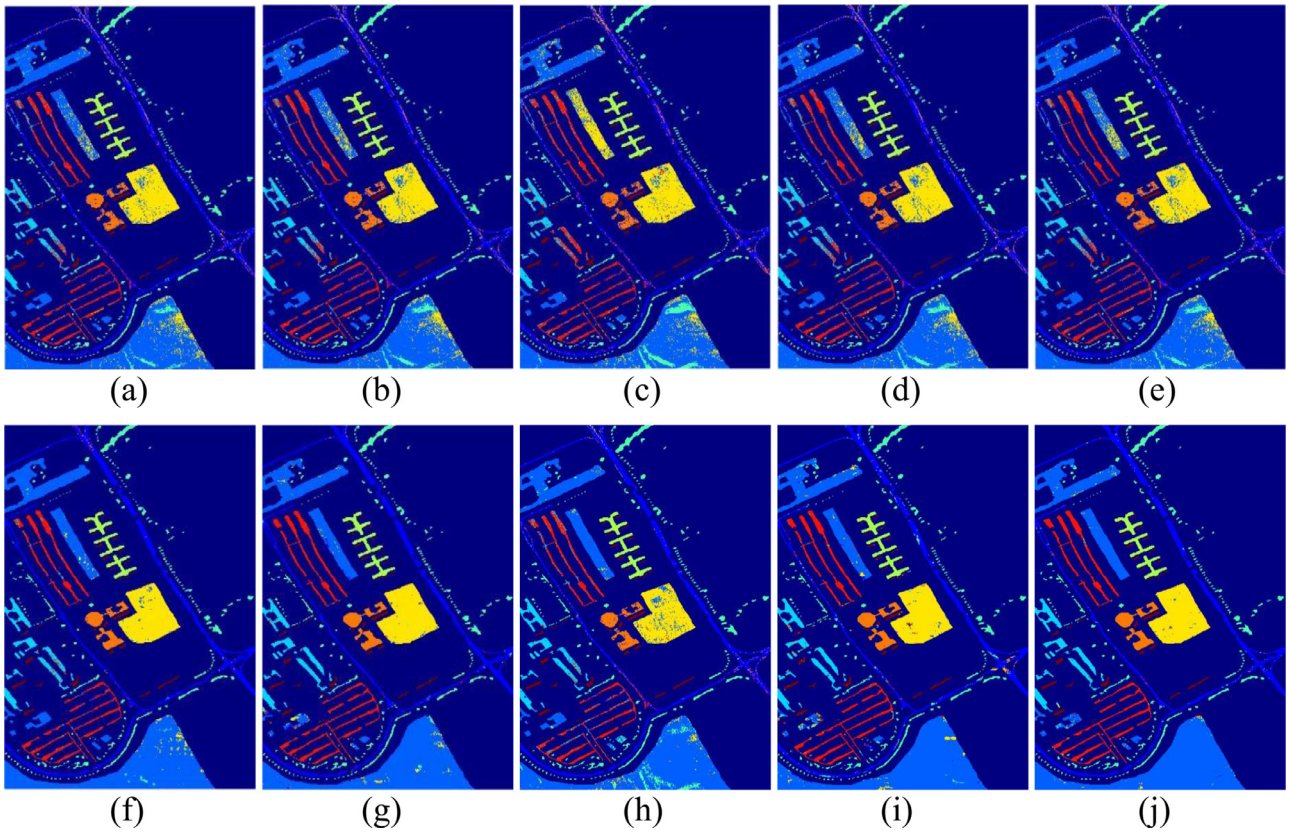


**Fig. 9.** Classification maps on the PUS dataset. (a) Original. (b) PCA. (c) LDA. (d) NWFE. (e) RLDE. (f) MDA. (g) CNN. (h) SeLSTM. (i) SaLSTM. (j) SSLSTMs.

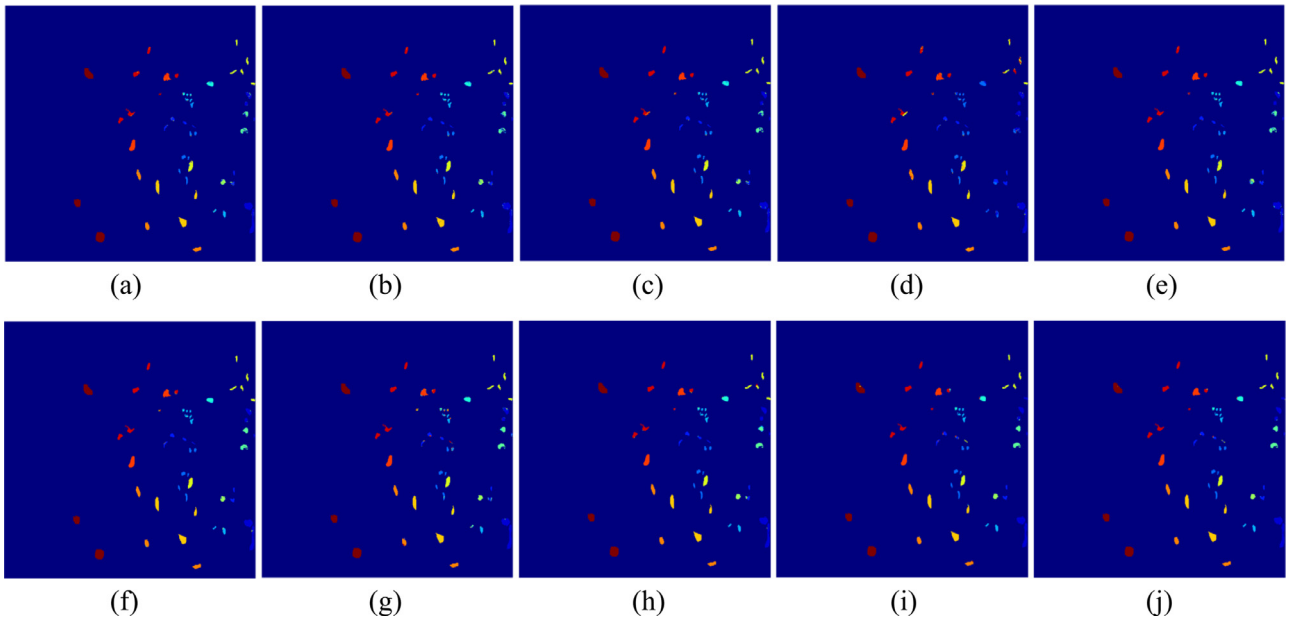


**Fig. 10.** Classification maps on the KSC dataset. (a) Original. (b) PCA. (c) LDA. (d) NWFE. (e) RLDE. (f) MDA. (g) CNN. (h) SeLSTM. (i) SaLSTM. (j) SSLSTMs.

consider spectral information only, because it can extract spatial and spectral features simultaneously. This indicates the importance of spatial features for HSI classification. So, as spatial based methods, CNN and SaLSTM perform better than other spectral-based methods. However, they only use the first principal component of all spectral bands, leading to the loss of spectral information. Therefore, the performance obtained by CNN or SaLSTM is inferior to that by MDA. Nevertheless, if we combine the spectral information and spatial information together, SSLSTMs can significantly improve the performance as compared to SeLSTM and SaLSTM. Additionally, as a kind of neural network, SSLSTMs is able to capture the non-linear distribution of hyperspectral data, while the linear method MDA may fail. Therefore, SSLSTMs obtains better results than MDA. Fig. 8 demonstrates classification maps achieved by different methods on the IP dataset. It can be observed that SSLSTMs obtains a more homogeneous map than other methods.

**Table 8**
OA, AA, per-class accuracy (%), and $\kappa$ performed by ten methods on KSC dataset using 10% pixels from each class as the training set.

| Label | Original | PCA | LDA | NWFE | RLDE | MDA | CNN | SeLSTM | SaLSTM | SSLSTMs |
|---|---|---|---|---|---|---|---|---|---|---|
| OA | 93.16 | 92.60 | 92.05 | 75.70 | 93.50 | 96.81 | 92.55 | 96.55 | 96.08 | 97.89 |
| AA | 89.15 | 88.45 | 87.02 | 59.47 | 90.09 | 95.30 | 89.20 | 94.31 | 95.38 | 97.28 |
| $\kappa$ | 92.38 | 91.76 | 91.14 | 72.65 | 92.77 | 96.45 | 91.69 | 96.15 | 95.63 | 97.65 |
| C1 | 95.43 | 95.14 | 95.40 | 97.14 | 95.30 | 96.93 | 94.86 | 99.71 | 98.54 | 99.56 |
| C2 | 91.44 | 91.36 | 92.51 | 91.19 | 92.26 | 97.26 | 77.53 | 88.13 | 82.65 | 90.41 |
| C3 | 90.86 | 90.55 | 82.89 | 77.19 | 88.44 | 98.92 | 84.52 | 100.00 | 99.57 | 100.00 |
| C4 | 79.52 | 77.94 | 71.98 | 0.08 | 76.90 | 90.31 | 77.71 | 85.02 | 100.00 | 99.56 |
| C5 | 68.20 | 65.34 | 62.36 | 0.00 | 77.64 | 80.00 | 80.97 | 78.62 | 92.41 | 93.79 |
| C6 | 67.34 | 64.54 | 74.93 | 4.37 | 77.82 | 92.47 | 72.62 | 82.52 | 94.66 | 95.15 |
| C7 | 84.19 | 85.52 | 72.95 | 0.00 | 82.67 | 94.68 | 93.19 | 100.00 | 100.00 | 100.00 |
| C8 | 95.17 | 94.66 | 89.88 | 36.33 | 91.97 | 96.26 | 93.87 | 95.36 | 83.51 | 88.40 |
| C9 | 95.92 | 94.15 | 95.12 | 94.92 | 98.08 | 99.89 | 95.85 | 99.15 | 98.93 | 99.57 |
| C10 | 96.78 | 96.68 | 99.21 | 90.10 | 96.78 | 98.35 | 96.81 | 100.00 | 97.53 | 100.00 |
| C11 | 98.14 | 98.14 | 97.85 | 94.18 | 98.23 | 99.33 | 94.27 | 100.00 | 97.61 | 99.47 |
| C12 | 95.90 | 95.83 | 96.22 | 87.95 | 95.39 | 94.59 | 97.35 | 97.57 | 94.92 | 98.90 |
| C13 | 100.00 | 100.00 | 100.00 | 99.98 | 99.68 | 99.94 | 100.00 | 100.00 | 99.64 | 99.88 |

Similar conclusions can be observed from the PUS dataset in Table 7 and Fig. 9. Again, MDA, CNN, and LSTM-based methods achieve better performance than other methods. Specifically, OA, AA and $\kappa$ obtained by CNN are almost the same as MDA, and SSLSTMs obtains better performance than CNN and MDA. It is worth noting that the improvement of OA, AA and $\kappa$ from MDA or CNN to SSLSTMs is not remarkable as those on IP dataset, because CNN and MDA have already obtained a high performance and a further improvement is very difficult. Table 8 and Fig. 10 show the classification results of different methods on the KSC dataset. Similar to the other two datasets, SSLSTMs achieves the highest OA, AA and $\kappa$ than other methods.

## 5. Conclusion

In this paper, we have proposed a HSI classification method based on a LSTM network. Both the spectral feature extraction and the spatial feature extraction issues were considered as sequence learning problems, and LSTM was naturally applied to address them. Specifically, for a given pixel in HSIs, its spectral values in different channels were fed into LSTM one by one to learn spectral features. For the spatial feature extraction, a local image patch centered at the pixel was firstly selected from the first principal component of HSIs, and then the rows of the patch were fed into LSTM one by one. By conducting experiments on three HSIs collected by different instruments (AVIRIS and ROSIS), we compared the proposed method with state-of-the-art methods including CNN. The experimental results indicate that using spectral and spatial information simultaneously improves the classification performance and results in more homogeneous regions in classification maps compared to only using spectral information.

## Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant Numbers: 61532009, 61522308, 61672292 and, in part, by the Foundation of Jiangsu Province of China, under Grant 18KJB520032.

## References

[1] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, IEEE J. Select. Top. Appl. Earth Obser. Remote Sens. 7 (6) (2014) 2094–2107.
[2] R. Hang, Q. Liu, H. Song, Y. Sun, F. Zhu, H. Pei, Graph regularized nonlinear ridge regression for remote sensing data analysis, IEEE J. Select. Top. Appl. Earth Obser. Remote Sens. 10 (1) (2017) 277–285.
[3] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (1968) 55–63.
[4] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2002.
[5] F. Palsson, J.R. Sveinsson, M.O. Ulfarsson, J.A. Benediktsson, Model-based fusion of multi- and hyperspectral images using pca and wavelets, IEEE Trans. Geosci. Remote Sens. 53 (5) (2015) 2652–2663.
[6] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (405) (1989) 165–175.
[7] T.V. Bandos, L. Bruzzone, G. Camps-Valls, Classification of hyperspectral images with regularized linear discriminant analysis, IEEE Trans. Geosci. Remote Sens. 47 (3) (2009) 862–873.
[8] R. Hang, Q. Liu, Y. Sun, X. Yuan, H. Pei, J. Plaza, A. Plaza, Robust matrix discriminative analysis for feature extraction from hyperspectral images, IEEE Trans. Geosci. Remote Sens. 10 (5) (2017) 2002–2011.
[9] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Trans. Geosci. Remote Sens. 42 (8) (2004) 1778–1790.
[10] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Trans. Signal Inf. Process. 3 (2014) e2.
[11] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[12] Y.L. Cun, B. Boser, J.S. Denker, R.E. Howard, W. Habbard, L.D. Jackel, D. Henderson, Handwritten digit recognition with a back-propagation network, in: Advances in Neural Information Processing Systems, 1990, pp. 396–404.
[13] Y. Lecun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[14] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Comput. Sci. (2014).
[17] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Comput. 1 (2) (1989) 270–280.
[18] P. Rodriguez, J. Wiles, J.L. Elman, A recurrent neural network that learns to count, Connec. Sci. 11 (1) (1999) 5–40.
[19] A. Graves, N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 1764–1772.
[20] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.
[21] K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, Comput. Sci. (2014).
[22] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Adv. Neural Inf. Process. Syst. 4 (2014) 3104–3112.
[23] M. Xu, H. Fang, P. Lv, L. Cui, S. Zhang, B. Zhou, D-stc: deep learning with spatio-temporal constraints for train drivers detection from videos, Pattern Recognit. Lett. (2017).
[24] M. Xu, Y. Wu, P. Lv, H. Jiang, M. Luo, Y. Ye, misfm: on combination of mutual information and social force model towards simulating crowd evacuation, Neurocomputing 168 (2015) 529–537.
[25] W. Li, G. Wu, F. Zhang, Q. Du, Hyperspectral image classification using deep pixel-pair features, IEEE Trans. Geosci. Remote Sens. 55 (2) (2017a) 844–853.
[26] W. Li, G. Wu, Q. Du, Transferred deep learning for anomaly detection in hyperspectral imagery, IEEE Geosci. Remote Sens. Lett. 14 (5) (2017b) 597–601.
[27] Q. Liu, F. Zhou, R. Hang, X. Yuan, Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification, Remote Sens. 9 (12) (2017).

[28] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, B. Zhang, Multisource remote sensing data classification based on convolutional neural network, IEEE Trans. Geosci. Remote Sens. PP (99) (2017) 1–13.
[29] Q. Liu, R. Hang, H. Song, Z. Li, Learning multiscale deep features for high-resolution satellite image scene classification, IEEE Trans. Geosci. Remote Sens. 56 (1) (2018) 117–126.
[30] C. Tao, H. Pan, Y. Li, Z. Zou, Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification, IEEE Geosci. Remote Sens. Lett. 12 (12) (2015) 2438–2442.
[31] Y. Chen, X. Zhao, X. Jia, Spectral-spatial classification of hyperspectral data based on deep belief network, IEEE J. Select. Top. Appl. Earth Obser. Remote Sens. 8 (6) (2015) 2381–2392.
[32] W. Zhao, S. Du, Spectral-spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach, IEEE Trans. Geosci. Remote Sens. 54 (8) (2016) 4544–4554.
[33] Y. Chen, H. Jiang, C. Li, X. Jia, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 54 (10) (2016) 1–20.
[34] H. Wu, S. Prasad, Convolutional recurrent neural networks for hyperspectral data classification, Remote Sens. 9 (3) (2017) 298.
[35] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Springer Berlin Heidelberg, 1997.
[36] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, A Field Guide to Dynamical Recurrent Neural Networks, in: Kremer, Kolen (Eds.), Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, IEEE Press, 2001.
[37] Z.C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, Comput. Sci. (2015).
[38] D. Kingma, J. Ba, Adam: a method for stochastic optimization, Comput. Sci. (2015).
[39] B.-C. Kuo, D.A. Landgrebe, Nonparametric weighted feature extraction for classification, IEEE Trans. Geosci. Remote Sens. 42 (5) (2004) 1096–1105.
[40] Y. Zhou, J. Peng, C.L.P. Chen, Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 53 (2) (2015) 1082–1095.
[41] R. Hang, Q. Liu, H. Song, Y. Sun, Matrix-based discriminant subspace ensemble for hyperspectral image spatial-spectral feature fusion, IEEE Trans. Geosci. Remote Sens. 54 (2) (2016) 783–794.

**Feng Zhou** is currently working toward the Master degree in the School of Information and Control, Nanjing University of Information Science and Technology. His research interests include deep learning and pattern recognition.

**Renlong Hang** (M'17) received the Ph.D. degree in meteorological information technology from Nanjing University of Information Science and Technology, Nanjing, China, in 2017. He is currently a lecturer in the School of Information and Control, Nanjing University of Information Science and Technology. His research interests include machine learning and pattern recognition.

**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. From 2010 to 2011, he was an Assistant Research Professor with the Department of Computer Science, Computational Biomedicine Imaging and Modeling Center, Rutgers, The State University of New Jersey, Piscataway, NJ, USA. Before he joined Rutgers University, he was an Associate Professor with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. During June 2004 and April 2005, he was an Associate Researcher with the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong. He is currently a Professor in the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing. His research interests include image and vision analysis. Dr. Liu received the President Scholarship of the Chinese Academy of Sciences in 2003.

**Xiaotong Yuan** received the B.A. degree in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2002, the M.E. degree in electrical engineering from Shanghai Jiao-Tong University, Minhang Qu, China, in 2005, and the Ph.D. degree in pattern recognition from the Chinese Academy of Sciences, Beijing, China, in 2009. After graduation, he held various appointments as a Postdoctoral Research Associate working in the Department of Electrical and Computer Engineering at the National University of Singapore, the Department of Statistics and Biostatistics at Rutgers University, and the Department of Statistical Science at Cornell University. In 2013, he joined the Nanjing University of Information Science and Technology, Nanjing, where he is currently a Professor of computer science. His main research interests include machine learning, data mining, and computer vision.