IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 1, 2019, accepted May 6, 2019, date of publication May 10, 2019, date of current version May 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916095

Adaptive Spatial-Spectral Feature Learning for Hyperspectral Image Classification

SIMIN LI[1], (Student Member, IEEE), XUEYU ZHU[2], YANG LIU[3], AND JIE BAO[1], (Member, IEEE)

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Mathematics, The University of Iowa, Iowa City, IA 52242, USA
[3]QuantaEye (Beijing) Technologies Co., Ltd., Beijing 100086, China

Corresponding author: Jie Bao (bao@tsinghua.edu.cn)

This work was supported in part by the Beijing National Research Center for Information Science and Technology, and in part by the Beijing Innovation Center for Future Chips, Tsinghua University.

ABSTRACT The combination of spectral and spatial information provides an effective way to improve the hyperspectral image (HSI) classification. However, the local spatial contexture changes with different neighborhood regions over the HSI image plane, and methods with fixed weights to integrate spatial information for all neighborhood regions could result in inaccurate spatial features, leading to adverse effects on classification performance. To address this issue, a novel adaptive spatial–spectral feature learning network (ASSFL) has been proposed to reflect spatial contexture changes and learn robust adaptive features in this paper. In the implementation of the proposed method, a convolution neural network (CNN) is first applied to learn weight features for each pixel within a hyperspectral patch and adaptive weights can be obtained based on a softmax normalization. Then, the shallow joint adaptive features can be acquired according to these weights. After that, a stacked auto-encoder (SAE) is proposed to further extract deeper hierarchical features for the final classification. The experimental results on four benchmark HSI data sets demonstrate that the proposed method can achieve competitive classification results compared with other existing classifiers.

INDEX TERMS Adaptive spatial-spectral features, convolutional neural network (CNN), hyperspectral image (HSI) classification, stacked auto-encoder (SAE).

I. INTRODUCTION

With the development of optical sensing technology, hyperspectral sensors can capture rich spectral and spatial information of the observed scene. The wealthy information in hyperspectral images (HSIs) [1] guarantees the superiority in material recognition and object detection. Thus, hyperspectral data has been applied to a wide range of fields, such as agriculture [2], environmental monitoring [3], food safety [4], mineralogy [5], and surveillance [6]. Traditional machine learning methods employed in HSI classification are based on the fact that different materials exhibit different spectral reflective curves and process each pixel independently. At early stage, popular machine learning algorithms widely used in HSI classification field include k-nearest neighbor (KNN) [7], linear discriminant analysis (LDA) [8], decision tree [9], random forest [10], [11], support vector machines (SVM) [12], [13] and so on. Among these methods, SVM is considered a benchmark method since it can handle the "curse of dimensionality" problem [14] and requires a relatively small size of training samples.

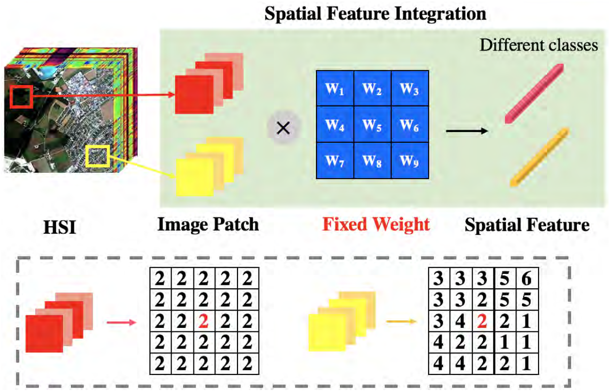
Although the pixel-wised classification methods can make full use of each pixel's spectral information, the obtained classification results can still be noisy. This is mainly because the spatial information has not been utilized. In fact, the spatial feature is equally important as the spectral feature and can be used to improve the classification accuracy. Since spatial adjacent pixels usually share similar spectral characteristics, utilizing spatial information can reduce the uncertainty of each sample and suppress the salt-and-pepper noise of classification maps [15]. Therefore, many spatial-spectral feature joint approaches have been developed. In the early stage, spatial features based on shallow filtering methods [16] are widely applied in HSI classification. In [17], an extended morphological profiles (EMPs) method is proposed for constructing shallow spectral-spatial features with a nonlinear morphological operator, which are adaptive definitions of the neighborhood of pixels. And Zhang *et al.* [18] used a

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

Markov random field (MRF) model to incorporate the spatial information based on the result of SVM, namely SVM-MRF. Though these kinds of methods could improve the classification results to some extent, the handcrafted features are strongly dependent on the prior information assumed by the practical practitioners. Besides, the obtained joint spectral-spatial features are relatively shallow and have poor adaption when the spatial environment changes.

In recent years, deep learning (DL) framework [19] has made attractive achievements in many fields (e.g., computer version [20]–[22], natural language processing (NLP) [23], and artificial intelligence [24]) by learning hierarchical representations from raw data. Motivated by these successes, deep learning models have also been introduced to analyze HSIs in remote sensing field. Compared with traditional machine learning methods, deep models have a good adaptation to different data sets and don't rely on much prior information. Besides, deep features are more robust and invariant in complex imaging conditions and have the capability of representing more abstract information. Chen *et al.* [25] first proposed a novel deep learning framework with stacked autoencoders (SAEs) and compared the classification results based on spectral features, principal component analysis (PCA) based spatial features, and joint spectral-spatial features. He also investigated the autoencoders' behavior in that paper. In a similar manner, a deep belief network (DBN) based method is also developed to extract deep spectral-spatial features using a multilayer restricted Boltzmann machine in [26]. Later, Liu *et al.* [27] combined stacked denoising autoencoders and spatial segmentation constraints to obtain an improved classification result. Although the SAE and DBN based models above are able to incorporate spatial information, they need to transform the spatial patches into 1-D vectors since the networks require 1-D inputs, which results in the loss of spatial information.

To address the loss of spatial information mentioned above, convolutional neural networks (CNNs) are introduced to HSI classification field and have been shown to be successful for hyperspectral data classification. In [28], a CNN containing a convolution layer, a pooling layer and a fully connected layer is employed to extract hierarchical features on spectral dimension and feed the extracted features to an output layer for final classification. Makantasis *et al.* [29] applied CNNs excluding pooling layers to the spatial dimension of a PCA-transformed HSI and obtained joint spectral-spatial features. To better explore the spatial contextual information, in [30] a contextual deep learning (CDL) method has been proposed based on two trainable filters to integrate spatial information. In [31], a regularized 3-D CNN model was proposed and the convolutional operation is applied to both spectral dimension and spatial dimensions simultaneously. Yang *et al.* [32] proposed a two-branch deep CNN (Two-CNN) architecture to extract joint spectral-spatial features for HSI classification and investigated transfer learning to address the limitation of the scarce training samples. Considering the spectral correlation among different wavelengths, recurrent neural networks (RNNs) [33] have also been introduced to perform HSI classification tasks very recently. In [34], the spectrum of each hyperspectral pixel is treated as a sequential data and the author utilized a modified gated recurrent unit (GRU) network to model the spectral dependency and produce the classification accuracy. Combined with CNN, Liu *et al.* [35] presented a bidirectional convolutional LSTM (Bi-CLSTM) to incorporate the spatial feature and capture the spectral-spatial dependency among different channels.



2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2

3	3	3	5	6
3	3	2	5	5
3	4	2	2	1
4	2	2	1	1
4	4	2	2	1

FIGURE 1. **Traditional fixed weights for spatial feature integration.**

Although these DL-based methods have made substantial improvements, none of them consider the inhomogeneous spatial contexture distribution over remote sensing image plane. And applying the same fixed weights to all neighborhood regions with different local spatial contexture would bring in inappropriate spatial information, such as samples from other classes or noise-contaminated samples, which could further result in adverse effects on the central spectrum's prediction. As a motivating example shown in FIGURE 1, the different colors of the two hyperspectral patches represent different local spatial contexture, and the numbers in each patch indicate various class labels which each corresponding spectrum belongs to. It is obvious that the central spectrums of the red image patch and the yellow image patch belong to the same class label 2 but with different neighboring spectra. If applying the same weights to both image patches, we could probably obtain the accurate spatial feature of class 2 for the red patch. But with a large possibility, we would misclassify the yellow patch since the weights are not adaptive to different spatial contexture.

To overcome the problem mentioned above, we propose an adaptive spectral-spatial feature learning network (ASSFL) for HSI classification tasks. As shown in FIGURE 2, the weights are related to the local spatial contexture of input patches and could adaptively change with different hyperspectral patches. In this way, robust adaptive features of class 2 for both red and yellow image patches could be acquired. Based on this observation, our proposed framework mainly consists of two stages: one is based on the weight learning network consisting of a CNN and a softmax

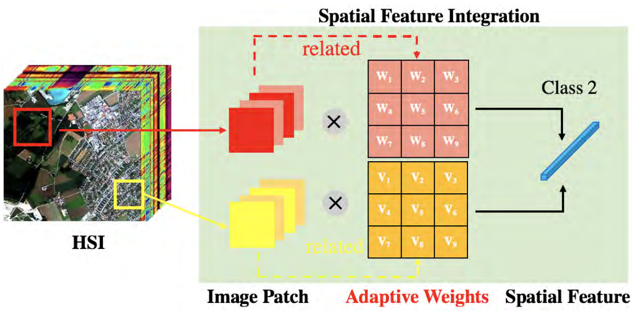


FIGURE 2. Proposed adaptive weights for spatial feature integration.

normalization for adaptive weights generation and outputs shallow joint features given different hyperspectral patches; the other is based on SAE for higher-level joint spatial-spectral feature learning. Together with these two parts within a unified framework, we could obtain discriminative joint spatial-spectral features directly instead of concatenating independent spatial and spectral features. The final classification is performed based on the robust higher-level joint features with a multinomial logistic regression (MLR) layer. The main contributions of this paper are summarized as follows:

1. We proposed a unified framework which combines CNN and SAE to extract joint spatial-spectral features directly from HSI data sets, instead of concatenating independent spatial features and spectral features;

2. Considering different local spatial contexture within different hyperspectral patches, an adaptive feature learning technique is proposed to integrate spatial information adaptively. Unlike conventional methods with the same fixed weights for all input patches, the proposed method could generate adaptive weights for each neighboring spectrum based on its contribution on the central spectrum's prediction. This could reduce the possibility of bringing in inappropriate spatial information and improve the robustness of classification results;

3. We analyze the effectiveness of the adaptive feature learning technique by visualizing the generated adaptive weights given different hyperspectral patches and examining the robustness of our algorithm at different noise levels. The visualization shows that our adaptive weights do change consistently with different spatial contextures of input patches, especially when the patches contain class borders. And the noise resistance experiments demonstrate the excellent robustness of our proposed method owing to the adaptive feature learning even under different noise levels.

The rest of the paper is organized as follows. Section II reviews deep learning models we used in our architecture, namely CNN and SAE. In section III, we present a detailed description of our proposed ASSFL method. Experimental analysis and a comparative evaluation of other baseline methods with four public HSI data sets are reported in section IV. Section V summarizes our work and gives the future instructions.

II. DEEP LEARNING, CNN, AND SAE

Deep learning based method builds a network with typically more than three layers. It tries to learn hierarchical levels [36] of data representation through layer-wised learning, and the high-level features can be learned from the low-level features. The resulted abstract and invariant features are beneficial to a wide variety of tasks such as classification and target detection. In this section, we shall briefly review CNN and SAE which we applied in our proposed model.

A. CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is a class of feed-forward artificial neural network [37] which is biologically inspired. A CNN usually consists of a combination of convolutional layers, pooling layers, and fully connected layers. The convolutional layers are composed of multiple convolution kernels. Each kernel learns a distinct feature map via the convolutional operation with the input and only responds to its receptive field, which can be formulated as:

$$C^i = f(\sum_j (W^i * x^j) + b^i), \tag{1}$$

where C^i is the ith channel's feature map of the present layer, W^i is the ith convolutional kernel matrix, b^i is the bias term of kernel i, x^j is the jth channel of the last layer, the symbol $*$ represents convolution operation, and f is a nonlinear activation function.

Pooling layers reduce the resolution of input feature maps and provide invariance by partitioning the input data into a set of non-overlapped sub-regions and returning the average or maximum values locally. In fact, convolutional layers together with pooling layers mimic the nature of complex and simple cells in mammalian visual cortex [38] and both of them can be repeated multiple times to obtain representative features. Finally, a fully connected layer is followed to further process the extracted features and convert them into category or regression results.

Compared with traditionally neural networks, CNN has much fewer trainable parameters and exhibit invariant characteristics in hierarchical feature extraction. Thus, it has been widely applied in computer vision fields [39] with superior performance.

B. STACKED AUTO-ENCODER

An autoencoder (AE) is a basic component in a SAE network. It consists of one visual layer of d-dimensional inputs $x \in R^d$, one hidden layer of h units $r \in R^h$, and one reconstruction layer $z \in R^d$. During the training process, x is first mapped to r in the hidden layer, which is called encoding; then the encoded feature r is mapped to z for reconstruction, which is also referred to as decoding. These two steps are formulated as follows:

$$r = f(W^1 x + b^1), \tag{2}$$

$$z = f(W^2 r + b^2), \tag{3}$$

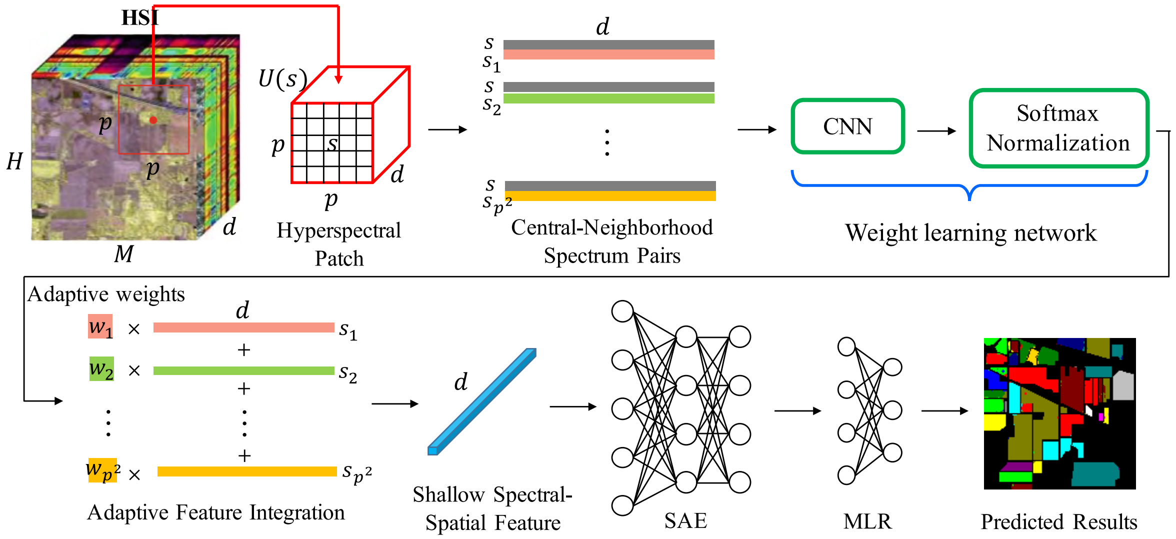


FIGURE 3. Proposed classification architecture for HSI classification based on adaptive spectral-spatial feature integration.

where W^1 and W^2 denote the weight matrixes of encoding and decoding process respectively, b^1 and b^2 are their corresponding biases, and f is an activation function.

By forcing the reconstruction layer z equaling to the input layer x, AE realizes a training process in an unsupervised way. The loss function $J(\theta)$ measures the total loss between the reconstruction z and the original input x of the training set:

$$J(\theta) = \frac{1}{2M}\sum_{i=1}^{M} \|z^i - x^i\|_2^2, \tag{4}$$

where M is the number of training samples, and x^i and z^i are the *ith* training sample and its reconstructed version respectively. By minimizing the loss function $J(\theta)$, we can find the optimized parameters $\theta = (W^1, W^2, b^1, b^2)$ of this AE layer and get the best encoded representation r of input x.

A SAE network consists of multiple layers of AEs, in which the hidden layer (r) of each AE is the input layer of the next AE. The reconstruction layer z is cast away after the training process of each AE layer has been finished. With this greedy layer-wised training, we can get a deep representation of the input data, and the deeper the layer is, the more abstract and representative features we get. The training process does not require any prior knowledge and is called unsupervised pre-training. After the pre-training, the parameters throughout the whole network have been adjusted to a relatively optimal stage as initialization and will be fine-tuned efficiently through the supervised training with back propagation (BP) algorithm [40], given the label of each input sample.

III. PROPOSED FRAMEWORK

Hyperspectral images collected from airplanes or satellites are usually corrupted by different lighting conditions, rotations of sensors and other disturbance. And integrating spatial information can help to reduce the uncertainty of each sample and extract robust features. However, most spatial feature integration methods did not take different local spatial contexture of hyperspectral patches into consideration. In this case, applying the same fixed weights to all input patches is likely to bring in inappropriate spatial information, such as noise-contaminated samples or samples from other classes, which would deteriorate the classification performance.

Considering the problems we mentioned above, we proposed an adaptive spatial-spectral feature learning network (ASSFL) which combines CNN and SAE to extract robust adaptive joint features in this paper. The whole framework is shown in FIGURE 3. Specifically, a weight learning network is first applied to central-neighborhood spectrum pairs within a hyperspectral patch to produce adaptive weights for each pixel. Then the shallow joint adaptive features are obtained based on these weights and fed to a SAE for higher-level spatial-spectral feature extraction. We put an MLR function as an output layer to get final classification results. The whole network is trained in an end-to-end supervised manner and all the parameters are optimized by mini-batch stochastic gradient descent (SGD) algorithm [41].

A. ADAPTIVE WEIGHTS LEARNING

The first stage of our ASSFL is to construct a weight learning network consisting of a CNN and a softmax normalization and apply it to the central-neighborhood spectrum pairs within a hyperspectral patch to generate adaptive weights according to the contributions of the neighboring spectrums on the central spectrum's prediction. Since the local spatial contexture changes with different hyperspectral patches, the characteristics of the neighboring spectra are different. Thus, our weight learning network could produce adaptive weights based on different local spatial contexture of input patches. The architecture of the proposed CNN for Salinas and PaviaU data sets is illustrated in FIGURE 4.

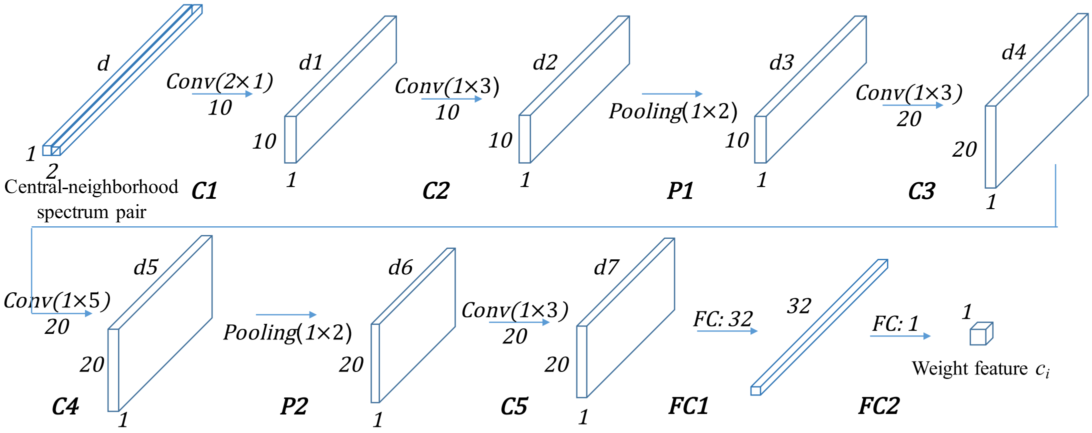


FIGURE 4. **The architecture of the designed CNN. Input data consists of a pair of the central spectrum and one of the neighboring spectrum.**

This designed architecture contains five convolution layers, two max-pooling layers, and two fully connected layers. We use rectified linear unit (*relu*) [42] $f(x) = max(0, x)$ as the nonlinear activation function after each convolution layer. Following are the details of the proposed framework.

Assume that $U(s)$ is a 3D hyperspectral patch from a HSI data set centered at spectrum s with a square window length p, as can be seen in FIGURE 3. In that case, there are p^2 spectra in the $U(s)$. For each one of these p^2 spectra, we generate central-neighborhood spectrum pair $S_i = [s, s_i]$ as the input layer, where s is the central spectrum of $U(s)$, s_i is the *ith* neighboring spectrum in this hyperspectral patch, and $i = 1, 2, \cdots, p^2$. The shape of the input layer is $2 \times d \times 1$ where d denotes the spectral dimension. The first convolution layer (C1) contains ten kernels with shape $2 \times 1 \times 1$, resulting in a $d \times 1 \times 10$ output volume of feature maps (without padding). The feature maps obtained in our first layer mainly measure the similarity between the central spectrum and each neighborhood spectrum. The second layer (C2) combines the feature maps obtained in C1 layer with ten $1 \times 3 \times 10$ kernels, producing a $(d - 3 + 1) \times 1 \times 10$ output volume. Then a max-pooling layer (P1) of 1×2 is followed to reduce the spectral dimension and get a lower resolution feature maps.

The following layers include two convolutional layers (C3 and C4) with twenty $1 \times 3 \times 10$ kernels and twenty $1 \times 5 \times 20$ kernels respectively. Again we use a max-pooling layer (P2) of 1×2 to further reduce the feature size. Another convolutional layer (C5) filters the feature maps from the previous layer P2 with twenty $1 \times 3 \times 20$ kernels. Once the feature dimension is reduced to a desired value, a fully connected layer (FC1) is added to extract more abstract and invariant features. The output layer (FC2) projects each learned feature to a final weight feature c_i with size 1, which represents the contribution of each neighborhood pixel on the central pixel's classification. For the whole neighborhood region, we get p^2 raw weight features. Finally, a *softmax* [43] function is applied to these p^2 weight features for normalization and generate the adaptive weights $W = [w_1, w_2, \ldots, w_{p^2}]$ for all the neighboring spectra $s_i \in U(s)$ where $i = 1, 2, \cdots, p^2$. The *softmax* normalization is applied as follow:

$$w_i = \text{softmax}(c_i) = \frac{e^{c_i}}{\sum_k e^{c_k}}. \tag{5}$$

For each pixel in HSI data sets, we can acquire an adaptive weight matrix W given a neighborhood region and obtain the shallow adaptive joint spatial-spectral feature by multiplying these weights with their corresponding spectra. The parameters of each layer in the designed CNN are only optimized by the classification results. Although these parameters are fixed once the training process is finished, they are able to output adaptive weights together with softmax normalization according to the different characteristics of neighboring spectra within different hyperspectral patches.

B. HIERARCHAL JOINT FEATURE EXTRACTION

Once we obtain the shallow joint spatial-spectral feature z, we feed it to a SAE network to further extract the higher-level representative and discriminative spatial-spectral feature of each pixel for the final classification. Our SAE consists of l hidden layers $r_i, i = 1, 2, \cdots, l$ and each of them can be computed by the following equations:

$$r_i = f(W_i z + b_i), \tag{6}$$

where W_i is the weight matrix connecting the input layer to the next layer i, and b_i is the bias of layer i. We choose *relu* function as an activation function. The pre-training method mentioned in Section II is used to train the designed SAE layer by layer.

C. MULTINOMIAL LOGISTIC REGRESSION

To be used in a classification task, our proposed ASSFL architecture ends up with a MLR layer. Since the last hidden layer r_l from SAE is regarded as the most discriminative feature for each pixel, we feed it to the MLR layer for classification. The output size of the MLR is the same as the total number of classes and we use *softmax* function as the activation function. The pixel's label is determined by the class with the largest probability:

$$y = \Phi(W_{MLR} r_l + b_{MLR}), \tag{7}$$

where r_l is the output feature from our SAE, W_{MLR} and b_{MLR} are the weight matrix and bias of the MLR layer, y is the predicted label and $\Phi(\cdot)$ is the activation function which is defined as:

$$\Phi(x) = \underset{i}{\operatorname{argmax}}\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right). \tag{8}$$

The whole network is trained in an end-to-end supervised manner and all parameters are optimized by minimizing the difference between the predicted outputs and the real labels.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed ASSFL, four publicly available HSI data sets are utilized to perform classification. We also compare with several other HSI classification methods, including SVM, extended morphological profiles with SVM (EMP-SVM) [17], stacked autoencoder (SAE) [25], CNN-based structure on spatial dimension (CNN) [29], contextual deep learning (CDL) [30], LSTM [34] and Two-CNN-transfer [32] as baselines for a comparative evaluation. For the performance metrics of all these methods above, overall accuracy (OA), average accuracy (AA), and κ coefficient [44] are adopted. OA is the overall accuracy for all classes and is defined as follow:

$$OA = \frac{\sum_i x^{(i)}_{\text{test,correct}}}{N}, \tag{9}$$

where $x^{(i)}_{\text{test,correct}}$ is the i-th correctly classified test sample, N is the total number of the test samples. AA represents the averaged accuracy of each class, and is defined as:

$$AA = \frac{1}{M}\sum_{i=1}^{M} \frac{\sum_{j=1} x^{(j)}_{i,\text{correct}}}{N_i}, \tag{10}$$

where M is the class number of a data set, N_i is the total test sample number of i-th class, and $x^{(j)}_{i,\text{correct}}$ is the j-th correctly classified test sample of class i. And kappa coefficient [44] is a statistical measure of agreement degree, referred as κ. The higher of all our measurement metrics the better of the classification performance. We run all experiments on a desktop PC equipped with an Intel Core 5 CPU and four GTX 780Ti GPUs.

A. DATASETS DESCRIPTION

We choose Salinas, Pavia University, Kenned Space Center and Indian Pines as our evaluation data sets. Their corresponding false color images generating from their spectral bands and ground truth maps are shown as FIGURE 5 -FIGURE 8. We randomly split each of these data sets to training sets (10%) and testing sets (90%) for each class respectively. And each spectrum in four data sets is uniformly scaled to the range of 0-1.

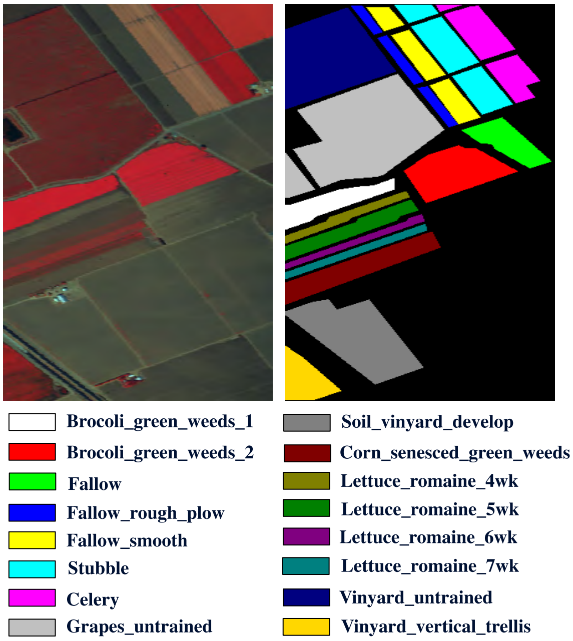


FIGURE 5. **Salinas data set. False-color image (band 52,25,10) and ground truth.**

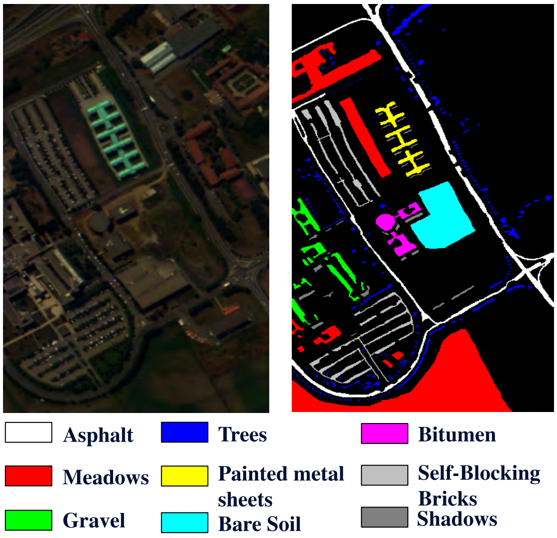


FIGURE 6. **PaviaU data set. False-color image (band 56,28,5) and ground truth.**

(1) Salinas Scene: This data set was collected by AVIRIS sensor at 1992 which recorded the remote sensing images of Salinas Valley, CA, USA. The hyperspectral image cube contains 512 × 217 pixels in spatial dimension and 224 spectral bands. Owing to the noise influence, we discarded 20 noisy bands to generate an experimental data set of only 204 spectral dimension. There are 16 different classes in this image, as shown in FIGURE 5. The number of training samples and test samples are presented in TABLE 1.

(2) Pavia University Scene: The Pavia University Scene (PaviaU) was a hyperspectral image data set captured by the reflective optics system imaging spectrometer (ROSIS)

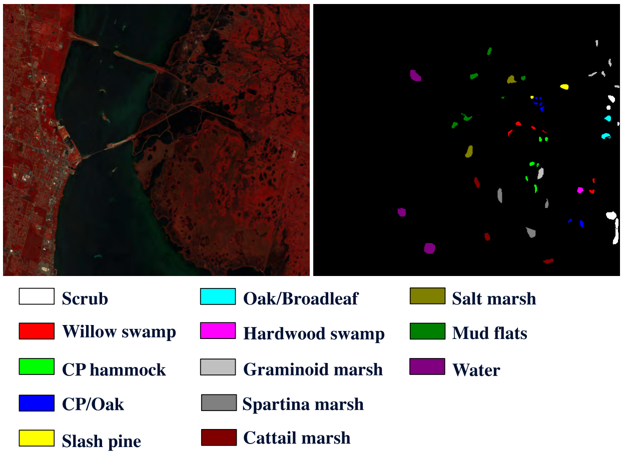


FIGURE 7. KSC data set. False-color image (band 56,28,5) and ground truth.

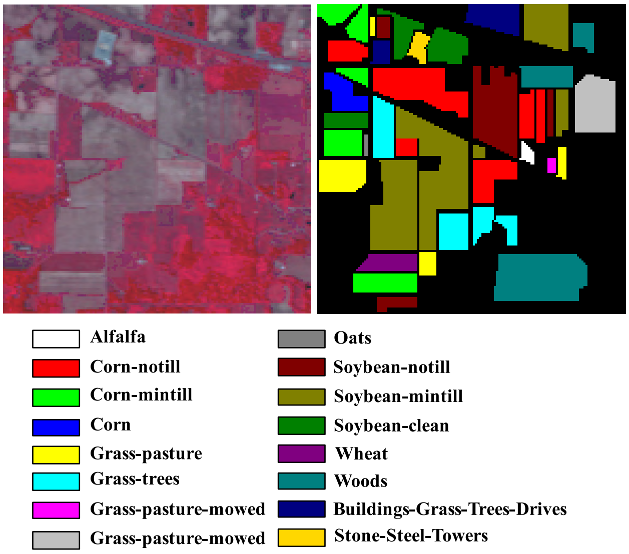


FIGURE 8. Indian Pines data set. False-color image (band 56,28,5) and ground truth.

sensor in year 2001 over northern Italy. The original image consists of 610 × 340 pixels together with 103 spectral bands. Nine labeled material classes are available in this data set, as shown in FIGURE 6.

(3) Kenned Space Center: The third HSI data set Kenned Space Center (KSC), was acquired by the AVIRIS sensor in Florida on March 23, 1996. The spatial dimension of this hyperspectral image is 512 × 614 pixels and the original spectral dimension is 224 spectral bands. Due to the noisy bands, we only use 176 bands for method evaluation. The image set contains 13 labeled categories, which can be seen from the ground truth map in FIGURE 7. The training data and test data are also described in TABLE 3.

(4) Indian Pines: Our last data set was captured by the AVIRIS sensor over the Indian Pines test site in North-western Indiana and consists of 145 × 145 pixels and 200 spectral bands after discarding water absorption bands.

TABLE 1. Numbers of training and test samples used in Salinas data set.

Class number	Class name	Train	Test
1	Brocoli green weeds 1	201	1808
2	Brocoli green weeds 2	373	3353
3	Fallow	198	1778
4	Fallow rough plow	139	1255
5	Fallow smooth	268	2410
6	Stubble	396	3563
7	Celery	358	3221
8	Grapes untrained	1127	10144
9	Soil vinyard develop	620	5583
10	Corn senesced green weeds	328	2950
11	Lettuce romaine 4wk	107	961
12	Lettuce romaine 5wk	193	1734
13	Lettuce romaine 6wk	92	824
14	Lettuce romaine 7wk	107	963
15	Vinyard untrained	727	6541
16	Vinyard vertical trellis	181	1626
total		5415	48733

TABLE 2. Numbers of training and test samples used in PaviaU data set.

Class number	Class name	Train	Test
1	Asphalt	663	5968
2	Meadows	1865	16784
3	Gravel	210	1889
4	Trees	306	2758
5	Painted metal sheets	135	1210
6	Bare Soil	503	4526
7	Bitumen	133	1197
8	Self-Blocking Bricks	368	3314
9	Shadows	95	852
total		4278	37646

TABLE 3. Numbers of training and test samples used in KSC data set.

Class number	Class name	Train	Test
1	Scrub	76	685
2	Willow swamp	24	219
3	Cabbage palm hummock	26	230
4	Cabbage/oak hummock	25	227
5	Slash pine	16	145
6	Oak/broadleaf hummock	23	206
7	Hardwood swamp	11	94
8	Graminoid marsh	43	388
9	Spartina marsh	52	468
10	Cattail marsh	40	364
11	Salt marsh	42	377
12	Mud flats	50	453
13	Water	93	834
total		521	4690

The scene contains two-thirds agriculture, and one-third forest or other natural perennial vegetation. The false color image and the corresponding ground truth map are shown

TABLE 4. Numbers of training and test samples used in Indian Pines data set.

Class number	Class name	Train	Test
1	Alfalfa	5	41
2	Corn-notill	143	1285
3	Corn-mintill	83	747
4	Corn	24	213
5	Grass-pasture	48	435
6	Grass-trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	Soybean-notill	97	875
11	Soybean-mintill	245	2210
12	Soybean-clean	59	534
13	Wheat	20	185
14	Woods	126	1139
15	Buildings-Grass-Trees-Drives	39	347
16	Stone-Steel-Towers	9	84
total		1024	9225

in FIGURE 8. And the training and test set are listed in TABLE 4.

B. BASELINE METHODS

To validate the classification performance of our proposed method, we compared our ASSFL with seven representative HSI classification methods, which includes SVM, EMP-SVM [17], SAE [25], CNN [29], CDL [30], LSTM [34] and Two-CNN-transfer [32]. The details of the parameters used in the baseline methods are listed as follows: (a) The SVM method only utilizes spectral information of each pixel and we use radial basis function (RBF) kernel with the LibSVM [45] in the experiments. (b) For the EMP method, spatial features are exploited by the adoption of five opening and closing operations followed by morphological reconstruction on the first three principal components (PCs) of HSI. We use the disk structure element to perform the morphological operations and the structure sizes range from 1 to 9. (c) For SAE method, first several PCs of each hyperspectral data set are extracted and flattened within a 7 × 7 neighborhood region, then they are concatenated with spectral information and feed to a SAE with default depth in [13] for further feature extraction. (d) The number of PCs in CNN method are determined by reserving at least 99.99% information of each HSI data set, then two convolution layers are followed to extract spectral-spatial features within a 7 × 7 spatial dimension patch. (e) The CDL method employs two trainable filters for spatial information integration and joint feature smoothness, respectively. The window sizes for both filters are set as default in [14] and the depth of SAE is set to two. (f) We adopt the band-by-band LSTM to perform classification. Given the limited size of training samples, the depth of LSTM is set to two for Salinas and PaviaU data sets and one for KSC and Indian Pines data sets. (g) The Two-CNN-transfer method requires source data set from the same sensor to pretrain the base network. For Salinas and PaviaU data sets, we use the default parameter settings as in [32]. As for KSC and Indian Pines, we refer to the parameter settings of Salinas as they come from the same sensor and perform empirical tuning.

C. PARAMETERS ANALYSIS

For the proposed ASSFL method, a weight learning network consisting of a CNN and a softmax normalization is first implemented to generate adaptive weights and output shallow joint adaptive features. Then a SAE is followed to extract higher-level and representative spatial-spectral features for the final classification. Except the weights in networks can be automatically learned during the training process, several other important parameters could influence the classification performance, such as window size and layer depth. Therefore, we shall investigate the effects of these parameters in this section.

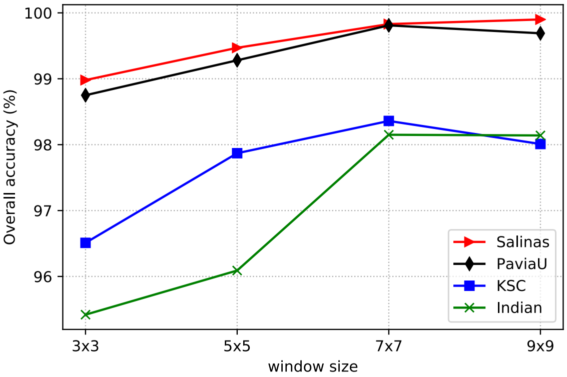


FIGURE 9. Classification accuracies of four hyperspectral data sets with different window sizes.

1) EFFECT OF INPUT WINDOW SIZE

The window size of hyperspectral patches decides how much spatial information could be integrated, thus it could affect the classification results. As we know, large window size will include more spatial information, but it may also introduce samples from other classes and increase computational complexity. On the other hand, a small window size may fail to contain enough spatial information, resulting in relatively poor classification accuracy. In this section, we fix other parameters and perform experiments on various window sizes. FIGURE 9 shows the plot of the overall accuracies (*OAs*) with window size of 3 × 3, 5 × 5, 7 × 7 and 9 × 9 for four HSI data sets. As we can see, the *OAs* first improve as the window size increases, but then become saturated and even drop a little for PaviaU and KSC data sets. To balance with the computational cost, we choose 7 × 7 as the optimal window size for all our four data sets.

2) EFFECT OF LAYER DEPTH AND CONVOLUTIONAL FEATURE SIZE

The depth of our adaptive weight learning network and the number of features can significantly affect the training performance. In the framework of CNN, the depth can affect the training efficiency of adaptive weights for spatial feature integration. Given the training set sizes of four HSI data sets, we choose five convolution layers for Salinas and PaviaU data sets as shown in FIGURE 4, and two convolution layers for KSC and Indian Pines data sets. Furthermore, the number of features determines the dimensionality of extracted weight features in C1, which influences the adaptive weights generation and classification performance. Therefore, we investigate the effect of feature size in the designed CNN. FIGURE 10 presents the *OA* value changes with different feature sizes. Based on the results, 15, 10, 10 and 5 are chosen as the optimal number of features for Salinas, PaviaU, KSC and Indian Pines data sets, respectively.

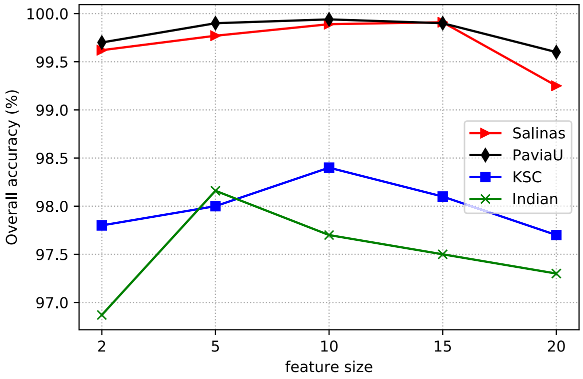


FIGURE 10. Classification accuracies of four hyperspectral data sets with different convolutional feature sizes.

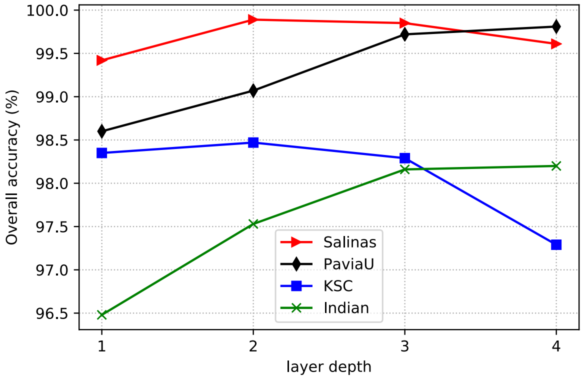


FIGURE 11. Classification accuracies of four hyperspectral data sets with different SAE layer depths.

3) EFFECT OF SAE LAYER DEPTH

The SAE helps to further extract the higher-level and discriminative joint adaptive features for the final classification. Therefore, the depth of SAE can impact the classification performance since it determines the abstraction level and invariance of the final joint adaptive features. In this study, we fix other parameters and analyze the effect of depth with SAE. The layer depth is chosen from 1 to 4 and the hidden units are set to 60 for Salinas, PaviaU and Indian Pines and 20 for KSC for each layer empirically. As can be seen from FIGURE 11, increasing depth can improve the classification accuracies. However, deeper network can also lead to over-fitting due to the limited training samples. According to our experimental results, we choose the SAE with two layers for Salinas and KSC data sets, and four layers for PaviaU and Indian Pines data sets for higher-level joint spatial-spectral feature learning.

TABLE 5. Classification accuracies of different techniques in percentages for Salinas data set.

Class	SVM	EMP	SAE	CNN	CDL	LSTM	Two-CNN	ASSFL
1	99.50	99.67	99.95	**100.00**	**100.00**	90.01	99.87	**100.00**
2	99.85	**100.00**	**100.00**	99.97	99.97	**100.00**	99.55	**100.00**
3	99.38	99.38	99.44	99.90	99.89	89.07	99.59	**100.00**
4	99.60	98.88	99.93	99.06	99.10	99.78	99.91	**99.84**
5	99.25	95.98	99.33	97.27	99.66	98.66	**100.00**	99.78
6	99.47	99.92	**100.00**	**100.00**	**100.00**	99.80	**100.00**	**100.00**
7	99.63	98.76	99.91	99.94	99.71	99.75	99.97	**100.00**
8	85.28	91.51	94.20	95.28	97.71	82.06	95.87	**99.08**
9	99.89	99.59	**100.00**	99.92	**100.00**	99.73	99.89	99.78
10	96.30	97.80	98.15	99.25	99.35	94.78	99.52	**99.65**
11	98.96	99.17	97.59	**100.00**	98.89	96.91	98.50	**100.00**
12	**100.00**	99.88	**100.00**	**100.00**	**100.00**	99.84	**100.00**	**100.00**
13	98.06	97.57	99.66	**100.00**	**100.00**	99.45	99.74	**100.00**
14	99.27	99.58	99.71	99.62	99.23	95.70	99.32	**100.00**
15	77.85	92.13	90.02	90.83	97.79	77.68	91.78	**99.81**
16	98.58	96.43	99.93	99.74	**100.00**	99.17	**100.00**	99.93
OA(%)	93.44	96.45	97.23	97.30	99.09	92.08	97.93	**99.91**
AA(%)	96.93	97.89	98.61	98.18	99.46	99.65	98.97	**99.87**
Kappa	0.9270	0.9606	0.9692	0.9728	0.9887	0.9127	0.9769	**0.9968**

TABLE 6. Classification accuracies of different techniques in percentages for Pavia University data set.

Class	SVM	EMP-SVM	SAE	CNN	CDL	LSTM	Two-CNN	ASSFL
1	92.71	98.42	95.85	99.22	98.51	92.11	99.17	**100.00**
2	97.14	98.14	98.95	99.51	99.84	96.87	**99.94**	99.12
3	78.19	94.71	91.47	77.69	98.31	73.32	90.33	**99.19**
4	93.04	99.24	98.11	98.75	98.61	92.33	99.46	**99.73**
5	99.34	99.59	**100.00**	99.93	**100.00**	99.93	99.83	**100.00**
6	90.21	91.12	89.56	98.33	99.95	90.79	99.73	**100.00**
7	86.80	96.07	88.04	94.89	99.00	88.95	95.24	**100.00**
8	86.08	98.70	93.81	98.28	98.55	92.48	98.10	**99.72**
9	99.76	99.76	99.79	99.58	99.65	99.58	**100.00**	98.10
OA(%)	93.27	97.33	96.14	97.88	99.23	93.21	98.96	**99.88**
AA(%)	91.28	97.31	95.07	96.24	99.16	91.82	97.98	**99.54**
Kappa	0.9107	0.9674	0.9493	0.9732	0.9914	0.9127	0.9867	**0.9971**

D. CLASSIFICATION RESULTS

In this section, we report the classification results of the proposed ASSFL and other baseline methods. The parameters are chosen as discussed above. We use RMSprop [46] as the optimization algorithm. Table 5-Table 8 show the quantitative assessments of Salinas, PaviaU, KSC and Indian Pines datasets with different methods and FIGURE 12-FIGURE 15 show their classification maps

TABLE 7. Classification accuracies of different techniques in percentages for KSC data set.

Class	SVM	EMP-SVM	SAE	CNN	CDL	LSTM	Two-CNN	ASSFL
1	95.03	98.83	99.07	98.54	98.88	91.85	94.24	**100.00**
2	88.99	89.45	81.48	97.53	96.71	75.31	91.93	**99.76**
3	69.13	43.48	96.48	91.41	98.04	86.33	91.34	**99.25**
4	62.83	78.32	66.26	58.73	71.03	63.89	59.47	**88.97**
5	51.39	**100.00**	53.42	76.40	85.71	57.14	73.66	93.11
6	46.60	77.66	60.70	81.66	89.95	48.91	76.43	**91.27**
7	79.79	**100.00**	99.05	95.24	97.14	30.48	92.21	**100.00**
8	84.75	80.88	94.56	98.35	**100.00**	72.85	92.66	98.10
9	90.81	**100.00**	**100.00**	**100.00**	**100.00**	92.12	97.37	**100.00**
10	95.04	**100.00**	97.52	96.04	93.81	79.21	95.21	99.75
11	94.16	99.20	98.09	**100.00**	**100.00**	98.57	**100.00**	**100.00**
12	83.19	98.67	97.42	92.64	97.81	81.91	92.49	**99.60**
13	99.64	99.64	99.46	99.89	99.68	99.68	99.88	**100.00**
OA(%)	86.48	93.61	92.07	93.82	96.14	82.98	93.04	**98.47**
AA(%)	80.10	91.30	87.96	91.26	94.54	75.25	88.91	**97.51**
Kappa	0.8493	0.9287	0.9204	0.9380	0.9613	0.8186	0.9202	**0.9765**

TABLE 8. Classification accuracies of different techniques in percentages for Indian Pines data set.

Class	SVM	EMP-SVM	SAE	CNN	CDL	LSTM	Two-CNN	ASSFL
1	70.73	**97.56**	36.96	47.83	76.19	73.91	78.57	90.48
2	74.63	91.91	79.62	93.21	97.65	73.46	95.33	**98.60**
3	74.43	94.38	78.62	94.82	91.67	53.86	92.29	**96.90**
4	56.34	95.77	80.59	91.14	96.88	80.17	83.21	**99.53**
5	90.32	95.16	90.95	94.26	94.84	90.48	92.48	**96.81**
6	93.61	99.09	98.63	98.49	99.24	97.26	96.96	**99.70**
7	80.00	**100.00**	75.00	67.86	61.54	57.14	88.46	**100.00**
8	94.42	99.77	98.95	**100.00**	99.51	97.91	**100.00**	**100.00**
9	38.89	**100.00**	95.00	85.00	55.56	45.00	43.33	**100.00**
10	72.43	91.53	88.77	93.76	89.69	83.33	91.70	**98.50**
11	83.61	93.12	88.71	96.82	96.14	85.58	95.74	**98.61**
12	65.29	83.68	69.32	**96.10**	95.56	92.07	90.83	92.09
13	97.83	97.28	**100.00**	99.51	**100.00**	98.05	**100.00**	**100.00**
14	94.55	99.56	96.52	99.45	99.30	98.34	**100.00**	99.56
15	61.96	97.12	81.85	**99.36**	87.96	44.30	92.19	93.99
16	90.36	96.39	97.85	**100.00**	86.90	76.34	86.90	**100.00**
OA(%)	81.12	94.36	87.31	95.56	95.69	82.02	95.40	**98.18**
AA(%)	77.46	95.77	84.83	91.10	89.29	77.95	89.25	**97.80**
Kappa	0.7842	0.9357	0.8566	0.9544	0.9506	0.8052	0.9451	**0.9789**

obtained by different approaches. As can be seen, the SVM and LSTM methods obtain relatively poor performances and exhibit noisy estimations in the classification maps, since they fail to consider spatial information. In contrast, the classification results of EMP-SVM, SAE, CDL, CNN and Two-CNN methods show much improvement and deliver smoother appearance in visualization results by combining spectral and spatial features. Compared with the baseline methods, our proposed ASSFL can generate adaptive spatial-spectral features for better classification performance given different spatial regions. This avoids introducing inappropriate spatial information from other classes and suppresses the noise's disturbance. Thus, it achieves the best classification results on four public data sets, with *OA* = 99.91% for Salinas, *OA* = 99.88% for Pavia University, *OA* = 98.47% for Kenned Space Center, and *OA* = 98.18% for Indian Pines, and yields the cleanest visualization results much more similar to the reference maps than others, which demonstrates the superiority of our proposed method.

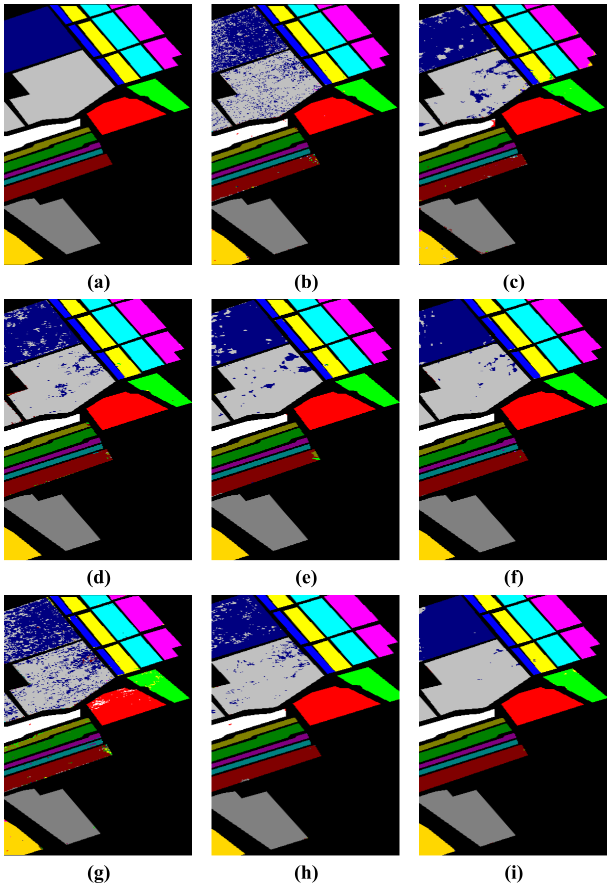


FIGURE 12. Classification maps of Salinas data set with 10% randomly selected training samples. (a) Ground truth map, (b) SVM, (c) EMP-SVM, (d) SAE, (e) CNN, (f) CDL, (g) LSTM, (h) Two-CNN, and (i) ASSFL.

E. ADAPTIVE WEIGHTS VISUALIZATION

As we mentioned above, our weight learning network could generate adaptive weights to neighboring spectra within a hyperspectral patch according to their contribution on the central spectrum's prediction. In this way, it could reduce the possibility of bringing in inappropriate spatial information such as samples from other classes or noise-contaminated samples, and obtain robust spectral-spatial features. To better understand how the adaptive weights change given different spatial contexture, we extract two kinds of 7 × 7 hyperspectral patches from each HSI data set with red boxes as shown in FIGURE 16, and analyze the visualizations of the generated adaptive weights corresponding to different input contexture.

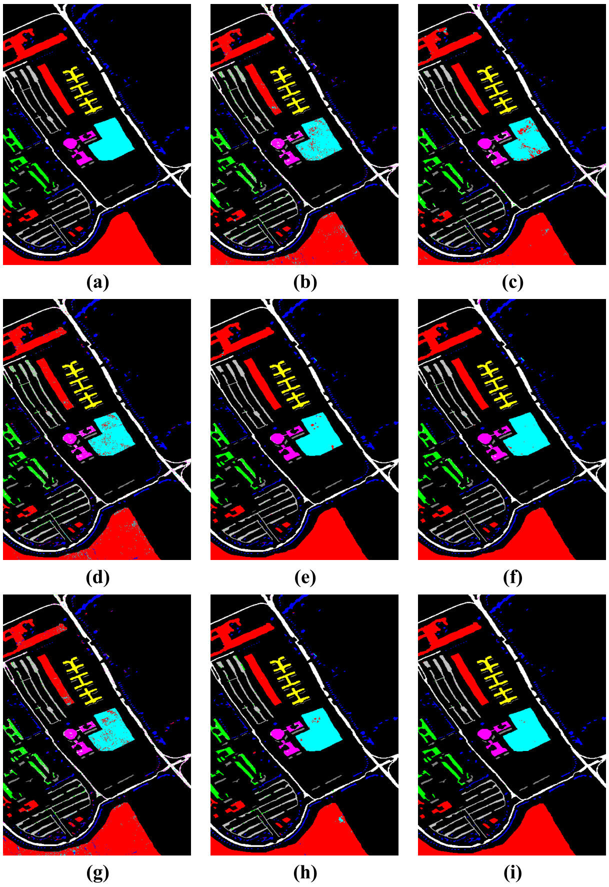


FIGURE 13. Classification maps of PaviaU data set with 10% randomly selected training samples. (a) Ground truth map, (b) SVM, (c) EMP-SVM, (d) SAE, (e) CNN, (f) CDL, (g) LSTM, (h) Two-CNN, and (i) ASSFL.

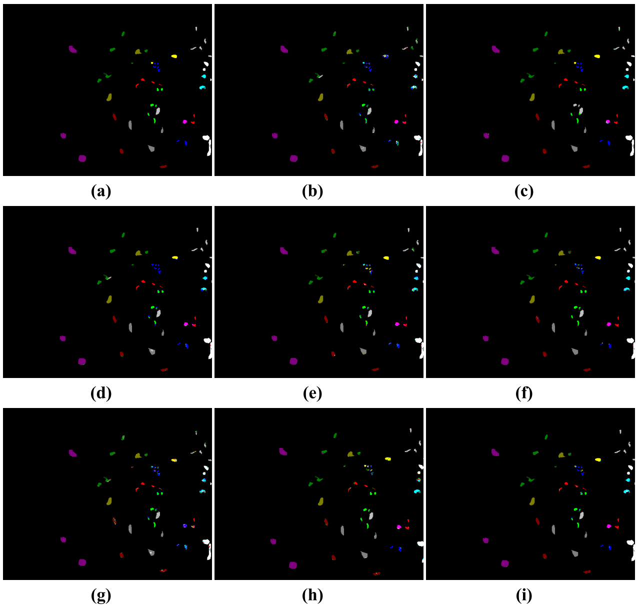


FIGURE 14. Classification maps of KSC data set with 10% randomly selected training samples. (a) Ground truth map, (b) SVM, (c) EMP-SVM, (d) SAE, (e) CNN, (f) CDL, (g) LSTM, (h) Two-CNN, and (i) ASSFL.

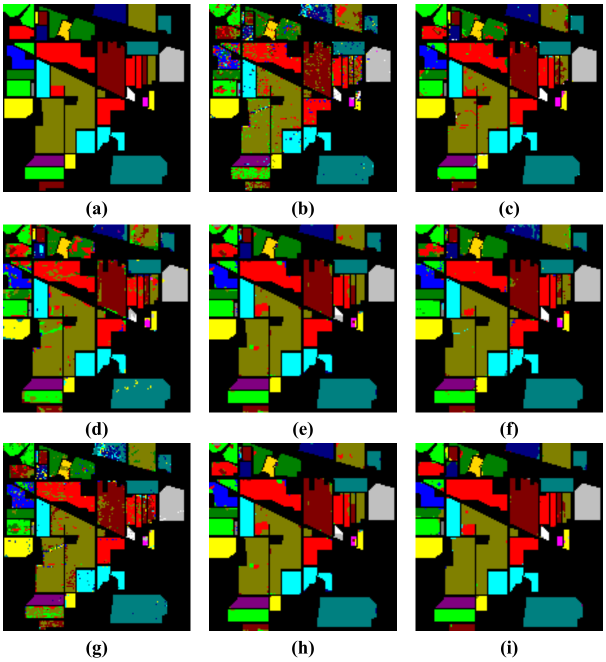
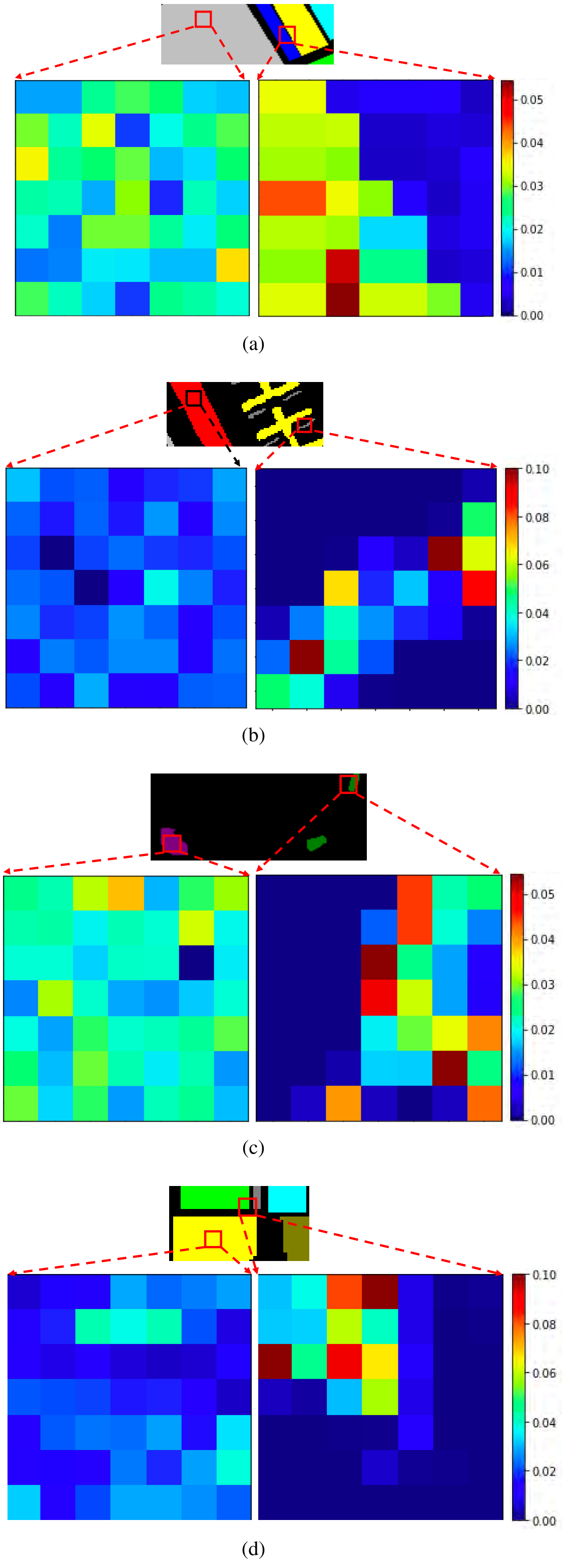


FIGURE 15. Classification maps of Indian Pines data set with 10% randomly selected training samples. (a) Ground truth map, (b) SVM, (c) EMP-SVM, (d) SAE, (e) CNN, (f) CDL, (g) LSTM, (h) Two-CNN, and (i) ASSFL.

As it can be seen from their ground truth maps in FIGURE 16, the left hyperspectral patches are extracted from homogeneous regions where spectra belong to the same class, while the right ones contain objects from different classes with clear class boundary lines. We feed these two kinds of patches to our weight learning network to generate adaptive weights according to different spatial contexture. The weight visualizations are displayed as 7 × 7 enlarged filters corresponding to their 7 × 7 input patches, as indicated by the red dotted lines. The colors in the visualized 7 × 7 adaptive filters represent the relative values of the generated adaptive weights for their corresponding spectra at the same positions in the input patch. The sum of adaptive weights within each filter equals to 1.

We take Salinas data set for example and first analyze the visualization of adaptive weights for the homogeneous patch on the left. The color of the weight visualization almost ranges from light blue to light yellow with the corresponding values from 0.02 to 0.035, and the colors are distributed evenly within the filter. This means that our weight learning network generates weights with similar magnitudes given the relatively homogeneous patch. A few dark blue pixels in the visualization imply their corresponding spectra are not so helpful in the central spectrum's prediction, probably due to noise contamination. Thus, the weight learning network assigns them smaller weights to weaken their influence in the final classification. In contrast, the visualization of adaptive weights for the right hyperspectral patch extracted at the class border exhibits large color variation, ranging from dark blue to dark red with corresponding values from nearly 0 to 0.055, which suggests the inhomogeneous contexture of the input patch. Moreover, there exists a clear boundary line of color change in the visualized filter, which is consistent with the



(a)

(b)

(c)

(d)

FIGURE 16. Visualization of adaptive weights for different hyperspectral patches from four HSIs. (a) Salinas data set. (b) PaviaU data set. (c) KSC data set. (d) Indian Pines data set.

class boarder of the input patch. As we can see from the ground truth map, there are objects from another different class above the class boarder. Thus, the weight learning network produces small adaptive weights for all these spectra at the corresponding positions. As for the spectra of the same class below the boundary line, our weight learning network assigns relative large weights with brighter colors in the weights visualization. This adaptiveness to different contexture of hyperspectral patches also exists in the weights visualizations of Pavia University, KSC and Indian Pines data sets, which can be seen in FIGURE 16(b) - FIGURE 16(d).

Overall, the visualizations of adaptive weights demonstrate that our weight learning network does generate adaptive weights based on different local spatial contexture of input patches and has the ability to recognize useful neighboring spectra and reduce inappropriate spatial information by assigning adaptive weights. Therefore, our proposed method could adaptively integrate spatial information given various hyperspectral patches and generate robust and discriminative joint spectral-spatial features for HSI classification.

F. ROBUSTNESS TO DIFFERENT NOISE LEVELS

To further investigate the adaptiveness of our proposed method, we conduct noise robustness experiments with a noise-contaminated HSI and compare the classification performances of different methods under various noise levels. For simplicity, we take the Salinas data set for example and add a set of Gaussian noises with 0 mean and different variances (σ) to conduct classification experiments. The σ is chosen from [0.05, 0.15, 0.2, , 0.25, 0.3, 0.35, 0.4, 0.45, 0.5] and the architectures of all the methods remain the same.

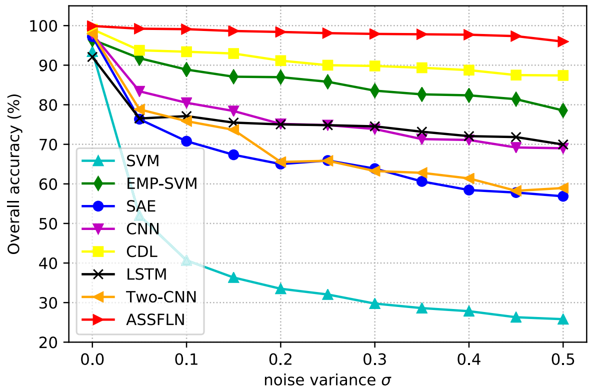


FIGURE 17. OAs on Salinas data set with different methods under various noise levels.

FIGURE 17 plots the *OA* values of different methods under various noise levels. From this figure, it is clear to see an obvious performance degradation for almost every compared method. As the noise variance σ increases, the OA first decreases and then becomes saturated later on. Compared the *OA* curves of different methods, it can be observed that the SVM is most sensitive to the noise and has a sharp performance degradation in classification accuracy. This is mainly because it fails to integrate spatial information and only extracts shallow features. Meanwhile, the performance of EMP-SVM method is beyond our expectation though it only extracts shallow features. Loosely speaking, the performance of CDL under different noise levels is the best among all the baseline methods, which is mainly due to the two trainable filters' smoothing effect on reducing the added noise.

Moreover, our proposed ASSFL presents superior classification performance under different noise levels, and has the least performance degradation from $OA = 99.91\%$ to $OA = 95.96\%$, which largely outperforms other baseline methods. This demonstrates the robustness of our proposed method under different noise levels owing to the adaptive feature learning. The experiments also demonstrate our ASSFL could achieve excellent performance even under a very noisy condition.

V. CONCLUSION

In this paper, we proposed a novel HSI classification framework, namely adaptive spectral-spatial feature learning network (ASSFL), to extract both spatial and spectral information adaptively within a unified structure. Compared with the traditional classification methods, our proposed model considers different local spatial contexture of input hyperspectral patches during spatial feature integration and introduces a weight learning network which consists of a CNN and a softmax normalization to generate adaptive weights given different hyperspectral patches. This could reduce the possibility of bringing in inappropriate spatial information, such as noise contaminated samples or samples from other classes. The shallow joint adaptive features are then obtained based on these generated weights and fed to a SAE for higher-level joint spatial-spectral feature learning. Benchmark results on four public HSI data sets demonstrate that our ASSFL has the superior performance with the cleanest classification maps and the highest *OA* values out of other baseline methods. The visualizations of adaptive weights and noise robustness experiments demonstrate the effectiveness of the adaptive feature learning in integrating robust and discriminative joint spatial-spectral features from HSI data sets. Since the adaptive joint feature learning technique has been proven to be effective in the HSI classification, we shall further explore the characteristics of weight learning network with some supervised constraints in our future work.

ACKNOWLEDGMENT

The authors would like to thank the editors and the reviewers for their kind work and constructive suggestions. They would also like to thank the Computational Intelligence Group, Basque University, for providing the hyperspectral images and corresponding reference information.

REFERENCES

[1] J. R. Jensen and K. Lulla, "Introductory digital image processing: A remote sensing perspective," *Geocarto Int.*, vol. 2, no. 1, p. 65, 2008.

[2] M. L. Whiting, S. L. Ustin, P. Zarco-Tejada, A. Palacios-Orueta, and V. C. Vanderbilt, "Hyperspectral mapping of crop and soils for precision agriculture," *Proc. SPIE*, vol. 6298, Sep. 2006, Art. no. 62980B.

[3] M. Moroni, E. Lupo, E. Marra, and A. Cenedese, "Hyperspectral image analysis in environmental monitoring: Setup of a new tunable filter platform," *Procedia Environ. Sci.*, vol. 19, pp. 885–894, Jan. 2013.

[4] Y.-Z. Feng and D.-W. Sun, "Application of hyperspectral imaging in food safety inspection and control: A review," *Critical Rev. Food Sci. Nutrition*, vol. 52, no. 11, pp. 1039–1058, 2012.

[5] F. A. Kruse, "Identification and mapping of minerals in drill core using hyperspectral image analysis of infrared reflectance spectra," *Int. J. Remote Sens.*, vol. 17, no. 9, pp. 1623–1632, 1996.

[6] P. W. Yuen and M. Richardson, "An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition," *Imag. Sci. J.*, vol. 58, no. 5, pp. 241–253, Oct. 2010.

[7] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[8] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[9] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre, and C. Mücher, "Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers," *Remote Sens. Environ.*, vol. 126, pp. 222–231, Nov. 2012.

[10] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[11] J. C.-W. Chan and D. Paelinckx, "Evaluation of random forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery," *Remote Sens. Environ.*, vol. 112, no. 6, pp. 2999–3011, Jun. 2008.

[12] J. A. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 2, Jul. 2000, pp. 813–815.

[13] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[14] D. L. Donoho *et al.*, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lect.*, vol. 1, p. 32, Aug. 2000.

[15] P. Ghamisi, J. A. Benediktsson, and J. R. Sveinsson, "Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5771–5782, Sep. 2014.

[16] L. Wei and D. Qian, "An efficient spatial-spectral classification method for hyperspectral imagery," *Proc. SPIE*, vol. 9124, May 2014, Art. no. 912410.

[17] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[18] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive Markov random field approach for classification of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 973–977, Sep. 2011.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

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

[21] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4346–4354.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[23] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167.

[24] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—A new frontier in artificial intelligence research [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010.

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

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

[27] Y. Liu, G. Cao, Q. Shen, and M. Siegel, "Hyperspectral classification via deep networks and superpixel segmentation," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3459–3482, Jul. 2015.

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

[29] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4959–4962.

[30] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, p. 20, Dec. 2015.

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

[32] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[33] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. New York, NY, USA: Wiley, 2001.

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

[35] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1330, 2017.

[36] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[37] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[38] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.

[39] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[40] S.-I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 801–806, Sep. 1992.

[41] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.* Paris, France: Springer, 2010, pp. 177–186.

[42] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, Jun. 2013, p. 3.

[43] S. Gold and A. Rangarajan, "Softmax to softassign: Neural network algorithms for combinatorial optimization," *J. Artif. Neural*, vol. 2, no. 4, pp. 381–399, 1996.

[44] W. D. Thompson and S. D. Walter, "A reappraisal of the kappa coefficient," *J. Clin. Epidemiol.*, vol. 41, no. 10, pp. 949–958, Jan. 1988.

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

[46] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

SIMIN LI (S'18) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, in 2014. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her research interests include machine learning, deep learning-based hyperspectral image classification, and computer vision.

XUEYU ZHU received the Ph. D. degree in applied mathematics from Brown University, Providence, RI, USA, in 2013. He is also with an interdisciplinary Ph.D. program in applied mathematical and computational sciences. He is currently an Assistant Professor with the Department of Mathematics, The University of Iowa. His research interests include computational mathematics, scientific computing, uncertainty quantification, model reduction, and high-performance computing.

YANG LIU received the Ph.D. degree in theoretical physics from the Institute of Physics, Chinese Academy of Science, in 2015. He is currently a Data Scientist with QuantaEye (Beijing) Technologies Co., Ltd., Beijing, China. His research interests include hyperspectral image classification, object detection, and image processing.

JIE BAO received the B.S. degree in chemistry from Tsinghua University, in 2006, and the Ph.D. degree in chemistry from Brown University, Providence, RI, USA, in 2010. He held a postdoctoral position at the Department of Chemistry, Massachusetts Institute of Technology. He joined the Department of Electronic Engineering, Tsinghua University, as an Assistant Professor, in 2013. He is currently an Associate Professor with Tsinghua University. His current research interests include semiconductor nanomaterial-based optoelectronic devices, algorithm development, and data processing and their applications.