Hyperspectral Classification Using Deep Belief Networks Based on Conjugate Gradient Update and Pixel-Centric Spectral Block Features

Chen Chen, *Student Member, IEEE*, Yi Ma, and Guangbo Ren

***Abstract*—This article describes the use of deep belief networks (DBNs) based on the conjugate gradient (CG) update algorithm for hyperspectral classification. DBNs perform two processes: unsupervised pretraining and supervised fine-tuning. The parameter update method in the fine-tuning stage plays a key role in optimizing the classification model. The proposed method employs CG-based fine-tuning to avoid the "zig-zagging" problem with the gradient descent algorithm and to accelerate the DBN convergence. First, the spectral features and pixel-centric spectral block features are extracted from hyperspectral images for use as the input vectors. The update variables are then calculated based on a CG algorithm and the 2-norm, and the parameters are updated during the backpropagation step of the proposed CGDBN. Two models with different CG methods are applied to a public hyperspectral image benchmark for classification experiments and analysis, and the results are compared with those from several classification methods that are currently in use. The experimental results show that the proposed classification models have advantages in terms of model convergence and low sensitivity to certain parameters. In addition, application to a hyperspectral image of coastal wetlands in the Yellow River Delta produces a satisfactory classification. The results of this study demonstrate that the proposed CG-update-based DBN provides a new approach for hyperspectral dataset classification.**

***Index Terms*—Backpropagation (BP), conjugate gradient (CG), deep belief network (DBN), hyperspectral image classification, pixel-centric spectral block features, 2-norm.**

I. Introduction

HYPERSPECTRAL images containing rich spectral and spatial information have become the focus of remote sensing image research. Regarding the methods of classifying hyperspectral images, many scholars have explored techniques that make full use of image information and effectively improve the classification effect.

Manuscript received April 16, 2020; revised June 7, 2020; accepted July 3, 2020. Date of publication July 13, 2020; date of current version July 24, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC1405100, in part by the National Natural Science Foundation of China under Grant 41206172, Grant 41706209, and Grant 61601133, and in part by the Background remote sensing monitoring of geographical elements in Shandong Yellow River Delta National Nature Reserve. *(Corresponding author: Yi Ma.)*

Chen Chen is with the College of Geomatics, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: chenchencc@fio.org.cn).

Yi Ma and Guangbo Ren are with the Marine Remote Sensing Division, the First Institute of Oceanography, Ministry of Natural Resources, Qingdao 266061, China (e-mail: mayimail@fio.org.cn; renguangbo@126.com).

Digital Object Identifier 10.1109/JSTARS.2020.3008825

Unlike shallow learning machines, deep learning can fully mine the underlying features of an image and learn the internal mechanism of these features. Therefore, in the context of the rapid development of computers, deep learning has been successfully applied in the field of image classification and significant results have been achieved. Hyperspectral images satisfy the requirements of deep learning in terms of sample information. Deep learning models have received widespread attention in the field of hyperspectral image classification [1]–[5]. Fang *et al.* [6] proposed a novel collaborative learning framework for semi-supervised hyperspectral image classification with joint deep convolutional neural networks (CNNs) and deep clustering, to solve the problem of limited labeled training samples in the hyperspectral image datasets. Considering that transfer learning strategies have the potential for the hyperspectral image classification, He *et al.* [7] proposed a new classification framework that combined transfer learning and deep CNN for hyperspectral image classification, and achieved good results especially when the training samples are limited. To fully take advantage of spatial and spectral information of the hyperspectral image, Han *et al.* [8] proposed a new joint spatial–spectral hyperspectral image classification method based on different-scale two-stream convolutional network and spatial enhancement strategy. Roy *et al.* [9] proposed a hybrid spectral CNN (HybridSN) for hyperspectral image classification, which is a spectral–spatial 3-D CNN followed by spatial 2-D-CNN. HybridSN model combines the complementary information of spatio-spectral and spectral in the form of 3-D and 2-D convolutions, respectively. Deep belief networks (DBNs) combine unsupervised learning and supervised learning. First proposed by Hinton *et al.* in 2006 [10], DBNs first use an unsupervised process to train the network parameters of each layer, and then apply a supervised process to calculate and transfer errors, constantly updating the network parameters through some optimization method. Finally, the network model can effectively describe the characteristics of the data and implement classification. Many studies show that the DBN classification model can be successfully applied to hyperspectral classification [11]–[19]. Tong *et al.* [11] improved the standard training process of DBNs to solve the problem of gradient disappearance when the number of hidden layers increases. They fused principal component analysis (PCA) with kernel PCA to reduce the dimensionality of the input data and achieved an overall accuracy (OA) of 96.72% with their DBN

model using 40% of the Salinas dataset as training samples. Zhong and Gong [12] proposed a novel model that takes advantage of the strength of DBNs in deep learning representations and conditional random fields in contextual (spatial) modeling to improve the classification of hyperspectral images. Liu *et al.* [13] proposed the active learning of deep networks based on the two stages of DBN training and the weighted incremental dictionary learning, enabling the training samples to be actively selected at each iteration. Li *et al.* [15] investigated a novel hyperspectral classification framework based on an optimal DBN algorithm, and a new texture feature enhancement that employs multi-texture features and band grouping for feature extraction and classification. Zhong *et al.* [16] developed a diversified DBN by regularizing the pretraining and fine-tuning procedures to deal with the problem of a limited number of training samples and the performance of "dead" (never responding) or "potentially over-tolerant" (always responding) units in the learned DBNs. Based on the characteristics of hyperspectral images, the combination of spectral information with other features such as the spectral neighborhood and spatial information improves the classification effect of the DBN classification model. Zhou *et al.* [17] developed a deep learning-based method that considers the characteristics of grouped features of the spatial–spectral data. Their approach has the ability to reduce the influence of redundant bands and extracts better features for hyperspectral image classification by incorporating a group-based weight-decay process in the DBN. Mughees *et al.* [18] proposed a deep learning-based spectral-adaptive segmented DBN architecture that analyzes a DBN and solves the relevant problems through spectral and spatial segmentation. This is a two-step classification approach that reduces the complexity of the learning process and extracts local features, making it simpler for the DBN to effectively extract the spectral–spatial features. Chen *et al.* [19] proposed a novel deep architecture that combines the spectral–spatial feature extraction and classification. Their DBN framework is a hybrid of PCA, hierarchical learning-based feature extraction, and logistic regression.

Many deep learning models use a backpropagation (BP) network [20], [21]. This allows the classification results of the network to successfully approach the true value by calculating the errors and updating the gradients. BP is actually a continuous optimization process with respect to the optimal value, with a gradient descent algorithm typically used during the iterative procedure. This method requires fewer calculations per iteration, takes up less memory, and is not especially sensitive to the initial conditions. Even for poorly chosen initial conditions, the minimum of the objective function can often be attained. In deep learning networks, BP algorithms often use stochastic gradient descent (SGD) [22], [23], mini-batch gradient descent [24], or some other efficient gradient descent-based optimization algorithm to improve the efficiency of model training. However, gradient descent-based approaches face the problem of slow convergence, because the iterative point is approaching a minimum along a tortuous path; effectively, the two search directions are always perpendicular to each other, producing the so-called "zig-zagging" problem [25]. Many studies have reported improved gradient descent-based methods, such as SGD with momentum [26], Nesterov accelerated gradient [27], and automatic adjustment of the learning rate based on SGD, e.g., Adagrad [28], Adadelta [29], RMSprop [30], and Adam [31].

In this article, we describe the construction and application of DBN classification models based on the conjugate gradient (CG) algorithms. As a type of optimization method, the CG algorithms [32], [33] use the conjugate direction as the search direction for each iteration. This effectively avoids the zig-zagging caused by using the gradient direction as the search direction. The CG algorithm has relatively small storage requirements and a fast convergence speed. In the BP process, we obtain the updated term of the parameter matrix by calculating the 2-norm of the gradient in the CG algorithm, thus achieving faster convergence and improved classification accuracy. The pixel-centric spectral block features after PCA dimensionality reduction and image enhancement are combined with the spectral features of the image to form the input of the proposed classification model. The University of Pavia dataset is used to carry out classification experiments, with two classic CG algorithms, namely the Fletcher–Reeves (FR) [34] and Polak–Ribiere–Polyak (PRP) algorithms [35], [36], used to construct CGDBN classification models. Comparison tests are conducted to explore the convergence performance and classification effect of the proposed models.

II. Proposed Method

We propose DBN classification models based on CG optimization with pixel-centric spectral block features as the input data. The framework is shown in Fig. 1. The proposed model includes three main modules: feature extraction from the pixel spectrum and pixel-centric block spectrum, generation of combined samples for model input, and classification in the CG-based DBN. In the proposed model, CG with 2-norm is used instead of gradient descent to fine-tune the parameters in each BP iteration. Two classic CG algorithms are used to obtain CGDBN models. The proposed models are described below.

A. Pixel-Centric Spectral Block Feature Extraction

First, the spectral data in the input image are normalized to [0, 1] in the spectral dimension using (1). Second, PCA dimensionality reduction is performed on the hyperspectral image. After PCA, the image is subjected to neighborhood-average image enhancement processing. Pixel-centric spectral block features are taken from the first 10 principal components. We use a 3-pixel $\times$ 3-pixel window to read the pixel values sequentially according to the principal components and obtain the vectorization feature of the pixel-centric spectral blocks, as shown in Fig. 2. The number of principal components and window size is determined as described in the following analysis. Finally, the vectorization spectral block features are normalized according to

$$y_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x_i is the ith value of the feature vector x and y_i is the ith value of the normalized feature vector y ($i = 1, 2, \ldots, m$, where m denotes the spectral dimension).

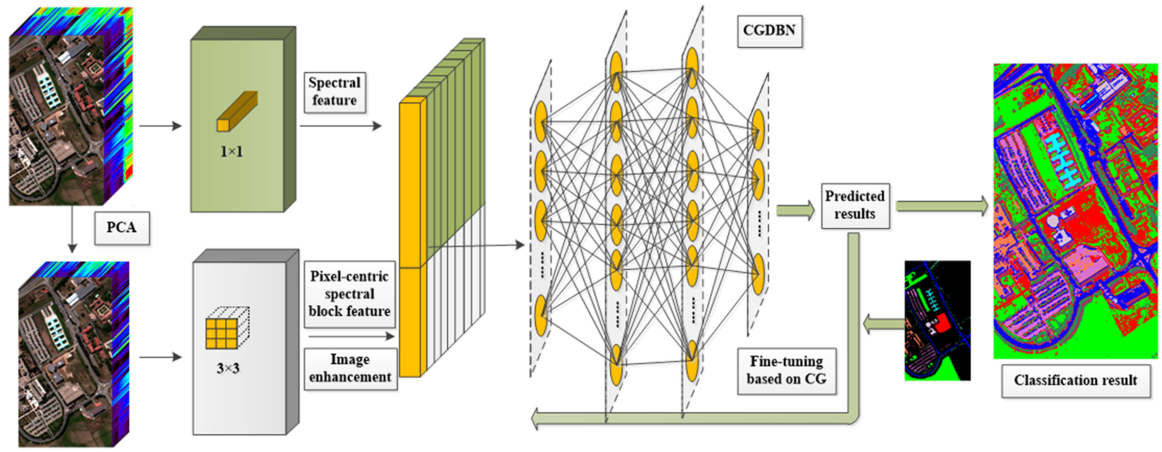

Fig. 1. Framework of proposed CG-based DBN classification model with spectral and pixel-centric spectral block features as input.

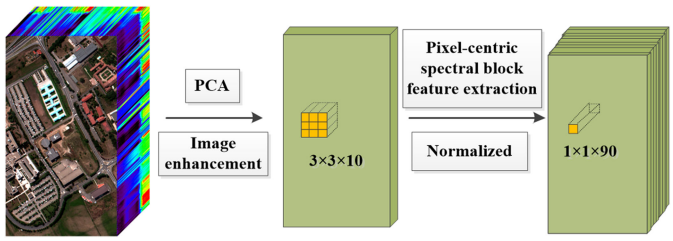

Fig. 2. Pixel-centric spectral block feature extraction.

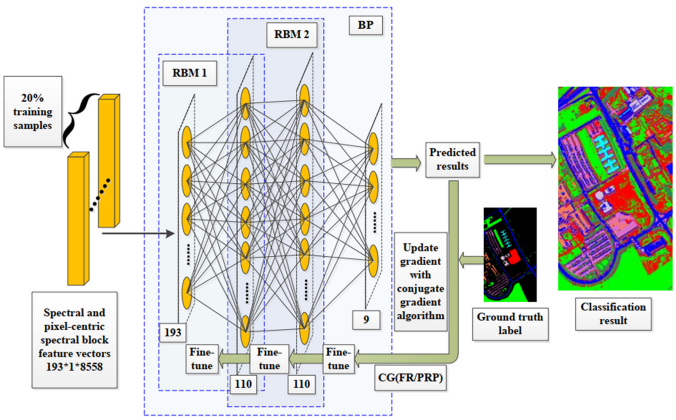

Fig. 3. Proposed CGDBN classification models.

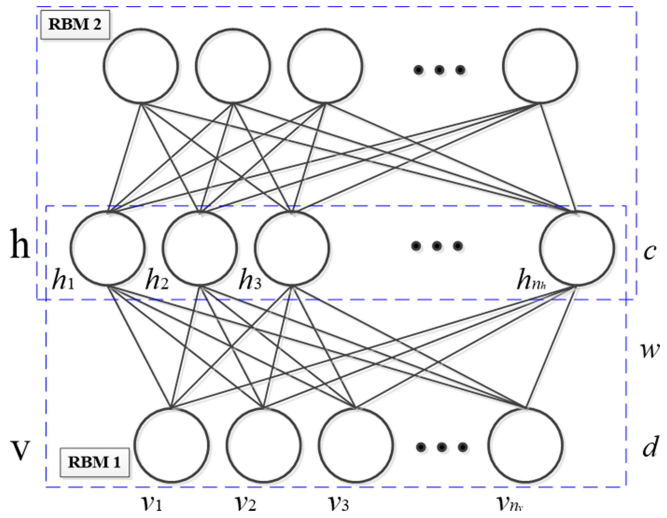

Fig. 4. RBM network structure. n_v represents the number of neurons in the visible layer, n_h represents the number of neurons in the hidden layer, $v = (v_1, v_2, v_3, \ldots, v_{n_v})^T$ represents the state vector of the visible layer, $h = (h_1, h_2, h_3, \ldots, h_{n_h})^T$ represents the state vector of the hidden layer, d is the bias of the visible layer, c is the bias of the hidden layer, and w represents the weight matrix between the hidden layer and the visible layer.

B. DBN Model Based on CG

The proposed model uses multiple restricted Boltzmann machines (RBMs) [37], [38] and BP for the pretraining and fine-tuning stages of model training, respectively. The structure of the CGDBN models is shown in Fig. 3.

The first step is to input sample feature vectors and train each layer of the RBM network separately in an unsupervised manner. RBMs are generative stochastic neural networks that consist of two layers of neurons, namely visible units (corresponding to visible variables) and hidden units (corresponding to hidden variables). The two layers of neurons are fully connected, but there is no connection between the neurons of each layer. The structure of an RBM is shown in Fig. 4. Unsupervised learning of the DBN is realized by stacking multiple layers of RBMs, with the hidden layer of this RBM forming the visible layer of the next RBM. The feature vectors retain as much feature information as possible when mapping to different feature spaces, and ultimately obtain the output feature vectors. The above-mentioned training process uses the contrastive divergence (CD) algorithm [39], which is a fast learning algorithm for RBMs. CD effectively improves the efficiency of fitting the RBM to the training samples. The parameters w, d, and c in the RBM are then updated, and the training process continues for multiple iterations. For the second RBM, the visible-layer neurons are generated based on w and c during the training process.

In the second step, the neural network receives the parameter matrices of the RBMs. The errors between the outputs and labels are calculated and propagated from back to front through each

layer. The weight matrices of each layer are updated through the CG, and the neural network is fine-tuned to produce the classification model. Training the RBMs ensures that the weight parameters are optimal for the feature vector mapping of each RBM, rather than for the feature vector mapping of the DBN. Therefore, the process of training the stacked RBMs can be viewed as the initialization of the deep BP network weight parameters. The BP network then starts to train the network parameters on the basis of the "initialization weight parameters." Compared with forward propagation, RBMs have shorter training time and improved classification efficiency.

Iterative methods are often used to approximate and find optimal solutions to large-scale optimization problems. BP passes the error J_{batch} between the prediction result and the real result to each layer of the network and updates the parameters W^l of each layer through an optimization algorithm. Finally, a trained model is obtained through successive iterations. The error term is given by

$$J_{\text{batch}} = \frac{1}{2\,\text{batch}} \sum_{i=1}^{\text{batch}} \left(Y_i^{\text{predict}} \left(w^l, b^l \right) - Y_i \right)^2 \quad (2)$$

where J_{batch} represents the cost function. The output classifier is the sigmoid function, batch denotes the number of batch training samples, Y_i^{predict} is the result of the output layer, Y_i is the ground truth label, and (w^l, b^l) represents the weights and bias parameters of layer *l*.

The CG algorithm is a classic unconstrained optimization algorithm. Unlike the gradient descent method, the search direction of the CG algorithm is the combination of the negative gradient direction of this iteration and the search direction of the previous iteration. CG requires first-derivative information, which not only overcomes the slow convergence of the gradient descent method but also avoids the computational cost of Newton's method for the Hessian matrix and its inverse. The CG algorithm is one of the most effective algorithms for large-scale nonlinear optimization, requiring relatively little storage space and no external parameters, and offering good convergence and stability. To overcome the slow convergence caused by the gradient descent search directions being perpendicular to each other during the iterative fine-tuning of the parameters, we construct an update factor ΔW_k^l based on the CG to adjust the weight and bias parameters. We construct models using two classic CG algorithms, FR and PRP. The updated weight and bias parameters are calculated as

$$W_{k+1}^l = W_k^l - \Delta W_k^l \quad (3)$$

where W_{k+1}^l is the $(k+1)$th iteration updated parameter matrix (including weights and bias) of the lth layer, W_k^l is the kth iteration parameter matrix of the lth layer, and ΔW_k^l is the update factor of the parameter matrix, given by

$$\Delta W_k^l = \alpha * dW_k^l - dW_{CG}^l \quad (4)$$

where the CG dW_{CG}^l of lth layer is calculated by (5), dW_k^l is the kth iteration gradient of the lth layer, and α is the learning rate. The parameter β_{k-1} in the FR and PRP algorithms is given by (6) and (7), respectively

TABLE I
ALGORITHM DESCRIPTION OF NETWORK UPDATING BASED ON CONJUGATE GRADIENT

Parameter updating based on conjugate gradient algorithm

Input: The parameter matrix of RBM layers W_0^l.
Batch training sample set $S = \{a_k^l\}$.
Learning rate $\alpha = 2$. Iterations *epochs = 500*.
Output: The parameter matrixes W_k^l.
1: ***for*** $t = 1, 2, \ldots,$ epochs **do**
2: ***forall the*** $a_k^l \in S,\ k = 1,2,3, \ldots$ ***do***
3: $Y_k^{perdition} \leftarrow \{a_k^l,\ W_0^l, sigmoid\}$ % sigmoid is activation function
4: $J_k \leftarrow \{ Y_k^{groundtruth}, Y_k^{perdition}, batchsize\}$
% average loss of batch training samples
5: $dW_k^l \leftarrow \{J_k,\ sigmoid\}$ % the calculation of gradient
6: $dW_{CG}^l \leftarrow \{\beta_k,\ \mathrm{d}W_k^l,\ \mathrm{d}W_{k-1}^l\}$
% the calculation of β_k refers to FR or PRP algorithm with 2-norm in (8) or (9).
7: ***if*** $t = 1$ ***then*** $\Delta W_k^l \leftarrow \{\mathrm{d}W_k^l,\ \alpha\}$
% Initial one is the steepest descent gradient
8: ***else*** $\Delta W_k^l \leftarrow \{\mathrm{d}W_k^l,\ dW_{CG}^l,\ \alpha\}$
9: $W_{k+1}^l \leftarrow \{W_k^l,\ \Delta W_k^l\}$
10: ***end if***
11: ***end for***
12: ***end***

$$d\,W_{CG}^l = \beta_{k-1}\, dW_{k-1}^l \quad (5)$$

$$\beta_{k-1}^{FR} = \frac{\left\| dW_k^l \right\|^2}{\left\| dW_{k-1}^l \right\|^2} \quad (6)$$

$$\beta_{k-1}^{PRP} = \frac{\left\| dW_k^l \right\|^T \left(\left\| dW_k^l \right\| - \left\| dW_{k-1}^l \right\| \right)}{\left\| dW_{k-1}^l \right\|^2} \quad (7)$$

where k is the number of iterations. Unlike classic CG optimization, we use the 2-norm in calculating β_{k-1}, which produces better DBN classification results. This use of the 2-norm can be expressed as (6) and (7), respectively

$$\beta_{k-1}^{FR\_L2} = \frac{\text{norm}(\left\| dW_k^l \right\|^2)}{\text{norm}\left(\left\| dW_{k-1}^l \right\|^2 \right)}\, \beta_{k-1}^{FR} \quad (8)$$

$$\beta_{k-1}^{\text{PRP}\_L2} = \frac{\text{norm}\left(\left\| dW_k^l \right\|^T \left(\left\| dW_k^l \right\| - \left\| dW_{k-1}^l \right\| \right) \right)}{\text{norm}\left(\left\| dW_{k-1}^l \right\|^2 \right)}\, \beta_{k-1}^{\text{PRP}} \quad (9)$$

where norm(·) returns the 2-norm, which is approximately the maximum singular value of matrix inside. The BP parameter update process is summarized in Table I. We separately apply the two CG methods (FR and PRP) processed using the 2-norm to update the network parameters of the respective CGDBNs.

III. EXPERIMENTS AND ANALYSIS

A. Data

We used the University of Pavia benchmark dataset to conduct experiments examining the performance of the proposed method. The benchmark scene was acquired by the ROSIS

TABLE II
CLASSES AND SAMPLES IN THE UNIVERSITY OF PAVIA DATASET

#	Class	Total	Dataset1 (10%)		Dataset2 (20%)		Dataset3 (30%)	
			Train	Test	Train	Test	Train	Test
1	Asphalt	6631	664	5967	1327	5304	1990	4641
2	Meadows	18649	1865	16784	3730	14919	5595	13054
3	Gravel	2099	210	1889	420	1679	630	1469
4	Trees	3064	307	2757	613	2451	920	2144
5	Painted metal sheets	1345	135	1210	269	1076	404	941
6	Bare Soil	5029	503	4526	1006	4023	1509	3520
7	Bitumen	1330	133	1197	266	1064	399	931
8	Self-Blocking Bricks	3682	369	3313	737	2945	1105	2577
9	Shadows	947	95	852	190	757	285	662
-	Total	42776	4281	38495	8558	34218	12837	29939

TABLE III
CLASSIFICATION RESULTS OF DIFFERENT INPUT FEATURES ON THE UNIVERSITY OF PAVIA DATASET

Training samples	Batch sizes of BP	OA of different input features (%)		
		Spectral	Spectral-spatial	Spectral-spatial IE
10%	34	90.48±0.89	94.99±0.41	95.93±0.23
20%	30	93.84±0.34	96.43±0.09	96.78±0.43
30%	32	93.34±0.68	96.64±0.04	97.64±0.63

sensor during a flight campaign over Pavia, northern Italy, and consists of 103 spectral bands with a geometric resolution of 1.3 m. The University of Pavia dataset covers 610 × 340 pixels and the land cover ground truth extends over nine classes. Of the total of 42 776 labeled samples, 20% were selected as training samples and the rest were used for testing. The OA is employed to evaluate the classification performance of the models. The classification experiments were performed using spectral features, combined features of the spectral and pixel-centric spectral block as input vectors.

B. Experimental Results

We used the classic DBN model and the proposed CGDBN models in the classification experiments. The models had a structure consisting of two hidden layers with 110 neurons each and an output layer with nine neurons (i.e., 110-110-9). According to the different input features, the input layer neurons of the tested models are slightly different, so that the network that takes spectral features as input has an overall structure of 103-110-110-9 whereas the network that takes spectral and pixel-centric spectral block features as input has a structure of 193-110-110-9. The other network parameters were set as follows: 200 unsupervised iterations, 50 RBM batch training samples, 500 supervised iterations, and 30 BP batch training samples. We took 10%, 20%, and 30% of the labeled samples in turn to form three separate datasets for the experiments. The number of samples is listed in Table II.

Table III presents the classic DBN classification accuracy based on different input features. The results show that, compared with the input of spectral features, the classification produced by adding the pixel-centric spectral block features is better than the single-feature input. The pixel-centric spectral block features contain pixel spectral features and their neighborhoods, which enhance the classification effect of the model. This agrees with the results of previous research [19]. After image enhancement, the classification accuracy of the DBN based on inputting the combined features is further enhanced. The results of experiments using different classification models with this combined input are presented in Table IV (where PRP DBN refers to the DBN based on the PRP algorithm and 2-norm, and FR DBN refers to the DBN based on the FR algorithm and 2-norm). The two models proposed in this article are collectively called CGDBNs. The classification accuracy of the CGDBNs is higher than that of the classical DBN. A comparison of the two CGDBNs indicates that FR DBN has slightly higher classification accuracy.

TABLE IV
CLASSIFICATION RESULTS OF DIFFERENT MODELS WITH SPECTRAL-SPECTRAL BLOCK FEATURE INPUT ON THE UNIVERSITY OF PAVIA DATA

Training samples	Batch sizes of BP	OA of different classification models (%)		
		DBN	PRP DBN	FR DBN
10%	34	95.93±0.23	96.08±0.37	96.08±0.42
20%	30	96.78±0.43	97.21±0.11	97.31±0.18
30%	32	97.64±0.63	97.67±0.62	98.00±0.24

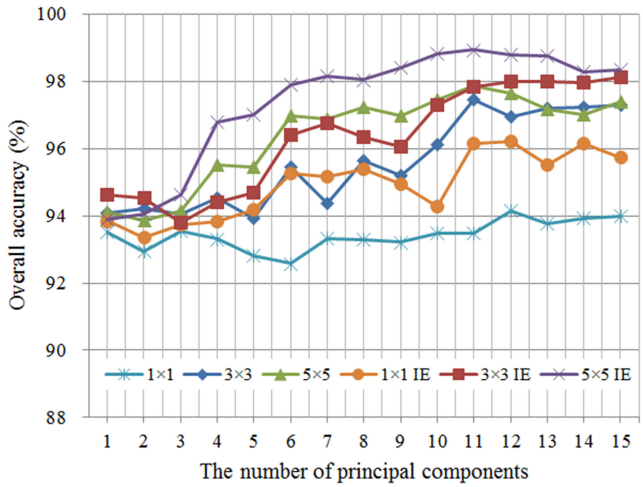

Fig. 5. Influence of the number of principal components, spectral block size, and image enhancement. "IE" denotes image enhancement processing. Spatial average filtering algorithm used for image enhancement.

For the extraction of sample spectral feature blocks, we performed image enhancement processing after PCA, and then spatial average filtering of each band. We varied the number of principal components to be retained from 1 to 15 to examine the effect on the final classification accuracy. In Fig. 5, the classification accuracy increases with the number of principal components and gradually stabilizes. Ten principal components enable 99.8% of the original image information to be retained. Thus, considering the amount of information and the computational complexity, we decided that 10 principal components were

TABLE V
COMPARISON OF DBN CLASSIFICATION BASED ON DIFFERENT OPTIMIZATION ALGORITHMS ON THE UNIVERSITY OF PAVIA DATA

Results	SGD	PRP	FR	PRP 2-norm	FR 2-norm	Nesterov	Adam	Adagrad	Adadetla	RMSprop
Overall accuracy (%)	96.78	96.82	97.15	97.21	97.31	97.21	95.49	95.66	95.67	94.33
Training accuracy (%)	98.34	98.23	98.59	98.66	98.70	98.73	96.96	97.25	97.48	95.34
Training time (s)	124	211	201	544	542	271	167	148	223	165
Training time of BP (s)	83	164	155	496	493	221	118	103	176	117

NAG: gamma $= 0.5$, $\alpha = 2$; Adagrad: eps $= 1e$-8, $\alpha = 0.001$; Adam: beta1 $= 0.9$, beta2 $= 0.999$, eps $= 1e$-8, $\alpha = 0.001$;
Adadetla: gamma $= 0.9$, eps $= 1e$-8, $\alpha = 0.001$; RMSprop: gamma $= 0.9$, $\alpha = 0.001$.

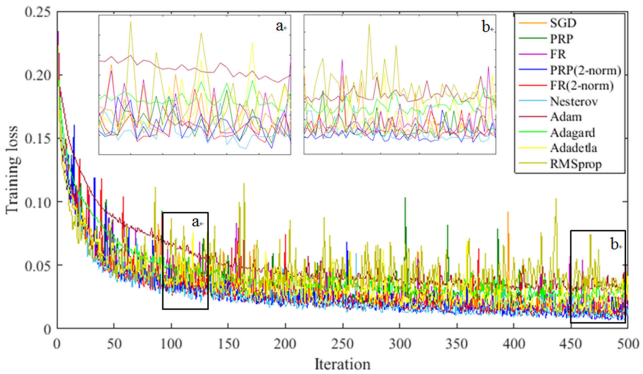

Fig. 6. Training loss of different classification models.

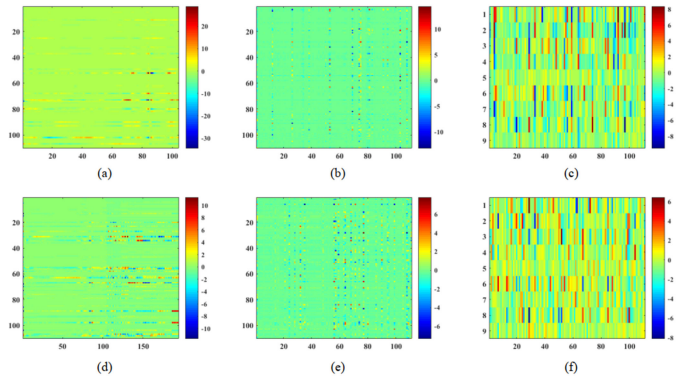

Fig. 7. Color maps of trained network parameters on the University of Pavia dataset. (a)–(c) Color maps of RBM 1, RBM 2, and the output layer of DBN, respectively. The sizes of the parameter matrices are 110×104, 110×111, and 9×111. (d)–(f) Color maps of RBM 1, RBM 2, and the output layer of FR DBN, respectively. The sizes of the parameter matrices are 110×194, 110×111, and 9×111. The final column of the matrices is the bias.

sufficient. We selected the pixel-centric spectral block features in 1 pixel $\times$ 1 pixel, 3 pixel $\times$ 3 pixel, and 5 pixel $\times$ 5 pixel windows to examine the effect of the block size on the classification results. Considering the classification accuracy and computational complexity, a block size of 3 pixel $\times$ 3 pixel is reasonable. We can also conclude from Fig. 5 that the spectral block features after image enhancement produce a better expression of the internal relationship between the center pixel and the neighboring pixel features, ultimately achieving better classification results.

C. Convergence Analysis

Fig. 6 shows the training loss curves of dataset 2 (20% training samples) using DBN models based on different optimization algorithms and the proposed CGDBNs. Note that we used the default parameter values of the Adam, Adagrad, Adadetla, and RMSprop algorithms. For detailed parameter information, see Table V. The input sample data include the spectrum and the spectral block with image enhancement. All the loss curves in Fig. 6 decrease and gradually stabilize. From the perspective of training loss and curve fluctuation, the DBN based on Nesterov and the models proposed in this article (denoted as PRP 2-norm and FR 2-normin the figure) offer better performance. The proposed models have a small training loss and little curve fluctuation, which is comparable to that of Nesterov. Although Adam and Adagard have small curve fluctuations, their training loss is relatively high. RMSprop converges faster in early iterations, but the training loss becomes higher in later stages and the curve fluctuates significantly. The DBN models based on SGD, Adadetla, PRP, and FR are similar, exhibiting low training loss and large fluctuations. The proposed CGDBN models have good convergence. The training loss curves quickly converge with small fluctuations.

Table V presents the classification results and runtimes corresponding to Fig. 6. The two proposed CGDBN models have longer runtimes than the other DBN models. The extra time consumption comes from the BP process. For the proposed CGBDNs, calculating the 2-norm of the conjugate direction matrix is the main reason for the increase in calculation time. The optimization algorithm does not necessarily guarantee that a local minimum value will be reached within a reasonable time, but it can usually find the optimal value of the cost function. Although the algorithm proposed in this article has a longer training stage than the other algorithms, it has advantages in terms of convergence and classification effect for the same number of iterations. The proposed CGDBNs are similar in time consumption. FR DBN is better in classification accuracy and model convergence, while PRP DBN performs better in terms of convergence stability.

D. Network Weight Matrices

Fig. 7 shows the network parameter matrices in the University of Pavia dataset. Taking the training result with dataset 2 as an example, the color maps on the second row illustrate the network

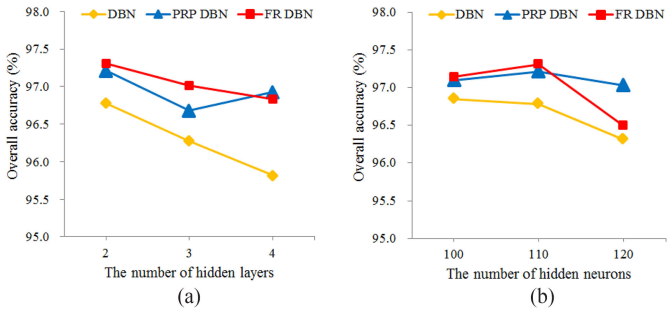

Fig. 8. Experiments with different hidden layer parameters. (a) Varying the number of hidden layers. (b) Varying the number of neurons.

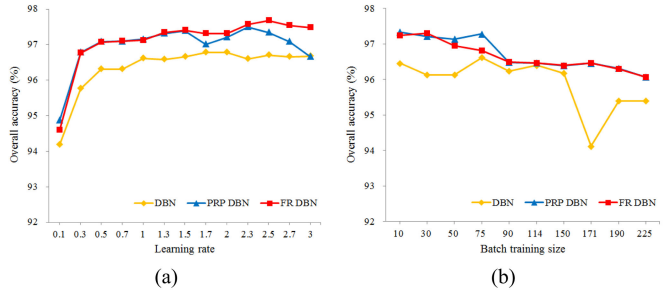

Fig. 9. Experiments with parameter adjustment during BP. (a) Learning rate. The proposed models are more accurate than the DBN for different learning rates. (b) Batch training size. The accuracy of the proposed models is within a range of 1.3%, whereas that of the DBN varies by about 2.5%.

coefficient matrices of the proposed FR DBN. From left to right, they correspond to RBM 1, RBM 2, and the output layer. Each coefficient matrix contains a weight matrix and a bias, and the matrix sizes are 110×194, 110×111, and 9×111, where the last column of the matrix is the bias. The network coefficient matrices of the DBN are shown in the first row. The matrix sizes are 110×104, 110×111, and 9×111. There are some obvious structures in the parameter matrices of both the DBN and FR DBN. Some research has mentioned that the learned weights in the first layer are localized continuous structure filters, whereas the weights in the second layer are local singular filters [16]. It can be seen that, in the same layer, the weight matrix of the proposed FR DBN exhibits more diversity than that of the DBN model. There are more continuously changing rows in the first weight matrix of the proposed model, indicating that the diverse features of the input samples are being learnt. The second weight matrix of the proposed model not only has more diverse rows than traditional models but also has more diverse columns. These are represented as scattered points in the matrix color map. In addition, the hidden layer parameter matrices of the models have different value ranges. Compared with the DBN model, the proposed FR DBN parameter matrices have significantly smaller value ranges.

E. Analysis of Other Parameters

The classification experiments were performed with different hidden layers. With a fixed 110 neurons in each hidden layer, the number of network layers was varied to 2, 3, and 4. The results are shown in Fig. 8(a). The number of hidden layers was then fixed to 2, and the numbers of hidden neurons in each layer were set to 100-100, 110-110, and 120-120. The results are shown in Fig. 8(b). When the number of hidden layer neurons is fixed at 110, the model with two hidden layers achieves the best classification results. When there are two hidden layers, the model with 110 neurons in each layer produces the highest classification accuracy. Thus, we conclude that the optimal structure of the hidden layer network is 110-110.

After determining the number of layers and neurons in the network, we analyzed the learning rate and batch training size involved in the BP process. The learning rate plays an important role in updating the BP gradient. If the update gradient ΔW denotes the adjusted direction to reach convergence, then the learning rate is equivalent to the step size in the direction of convergence. Choosing an appropriate learning rate is of great importance in the optimization of the parameter matrices. Fig. 9(a) shows the classification accuracy of the models with different learning rates. Compared with the DBN model, the proposed models have higher accuracy and lower sensitivity to the learning rate, especially FR DBN. Even for an inappropriate learning rate (such as 0.1), the classification accuracy of the proposed models can reach 94.6%. Fig. 9(b) shows the classification accuracy for different batch training sample sizes. As the batch training size increases, the accuracy decreases in all three models. However, compared with DBN (accuracy reduced by 2.5%), the degradation in the proposed models is much smaller (accuracy reduced by 1.3%). When the batch training size is large, the classification effect of the DBN model is not good.

F. Comparison of Different Methods

The classification results of DBN and the proposed CGDBNs are shown in Fig. 10. The CGDBN results are better than those of DBN. For example, asphalt and bitumen can be more effectively distinguished. Gravel and self-blocking brocks can be effectively distinguished in the CGDBN result. The proposed models performed well in terms of classification accuracy and model convergence, producing a clear advantage over the classic DBN model.

We used several methods to classify the University of Pavia dataset. Experiments were performed with the same input for 2-D-CNN, DBN, DBN with softmax classifier, and the proposed CG-based DBN, and the accuracy of each class and the overall classification are presented in Table VI. The OA of the other methods is greater than 96%, and the single-category classifications given by 2-D-CNN exhibit misclassification in class 9. Classes with few training samples (such as classes 3, 7, and 9) or those with greater similarities in ground features increase the difficulty of classification. After adding the CG and 2-norm to the DBN model based on the softmax classifier, a better classification result is obtained. In contrast, the proposed models perform well in the overall classification and some single-category classification of the CG-based DBN and the CG-based DBN with softmax classifier. There are some advanced hyperspectral classification methods at the bottom

TABLE VI
COMPARISON OF DIFFERENT CLASSIFICATION METHODS ON UNIVERSITY OF PAVIA DATASET

Methods	Classification accuracy of classes (%)									OA(%)	AA(%)	Training samples
	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9			
2D-CNN	98.52	99.41	93.78	96.45	99.77	96.82	95.44	95.52	13.61	96.14	87.70	20%
DBN-softmax	98.01	98.99	94.97	97.05	99.91	96.65	89.62	95.11	99.54	97.64	96.65	20%
PRP-DBN-softmax (proposed)	98.45	98.95	92.97	95.68	99.72	97.39	94.08	96.03	99.74	97.80	97.00	20%
FR-DBN-softmax (proposed)	98.10	99.28	93.30	97.15	99.96	97.07	91.54	95.50	99.67	97.86	96.84	20%
DBN	96.92	98.61	94.14	97.47	99.96	96.32	83.70	91.70	99.67	96.78	95.39	20%
FR DBN (proposed)	97.12	98.68	93.12	97.72	99.91	97.06	88.58	94.69	99.81	97.31	96.30	20%
PRP DBN (proposed)	97.35	99.41	92.53	97.53	99.96	94.82	85.06	94.25	99.81	97.21	95.63	20%
DBN-CRF-S-S [12]	89.43	95.52	89.94	97.17	99.56	94.80	95.75	89.43	100	94.37	95.03	1800
JSSC-DBN [14]	-	-	-	-	-	-	-	-	-	97.67	96.95	60%
WI-DL [13]	-	-	-	-	-	-	-	-	-	92.40	-	30%
R-3D-CNN [1]	100	100	100	99.89	100	100	100	100	98.94	99.97	99.87	70%

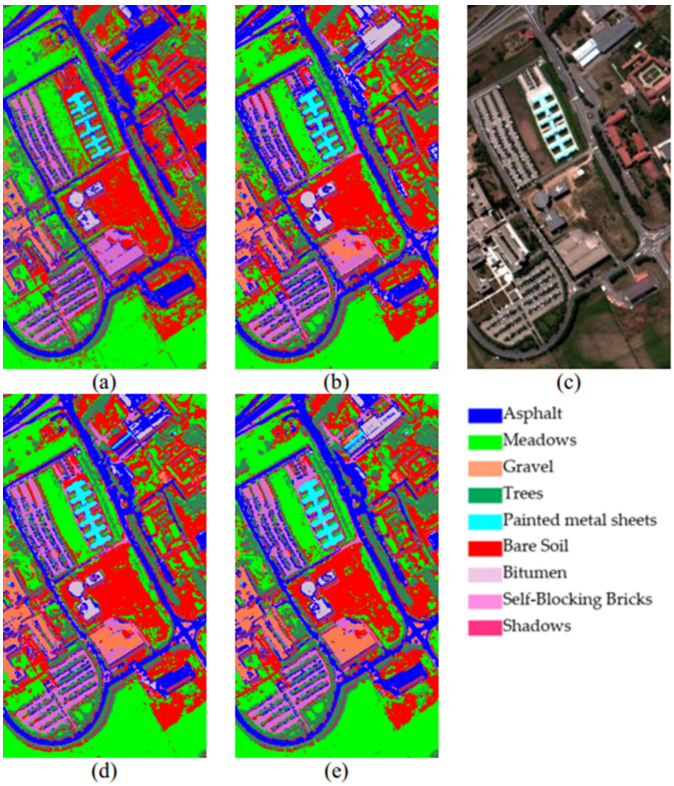

Fig. 10. Classification results on the University of Pavia dataset. (a) DBN classification result with spectral–spatial feature input. (b) DBN classification result with combined feature input. (d) FR DBN classification result with combined feature input. (e) PRP DBN classification result with combined feature input. (c) False color image composite (bands 50, 27, and 17). Combined feature refers to the combination of spectral feature and pixel-centric spectral block with image enhancement feature.

of the table. In [1], the R-3-DCNN outperforms our proposed methods only when sufficient training samples (at least 40% sampling rate) are provided. The proposed methods with the 20% sampling rate show higher OA than the methods (such as WI-DL and JSSC-DBN) with more training samples.

G. Application to a Real Scene

We applied the proposed classification model to real scenarios of the Yellow River Delta Coastal Wetland. The scene was acquired by the visible shortwave infrared hyperspectral camera onboard the Gaofen-5 (GF-5) satellite. The image covers part of the Bohai Sea in northeastern Dongying, China. We took 150 spectral bands in the spectral range of 390–1030 nm as the spectral data for experiments. The spatial resolution was

TABLE VII
SAMPLE DESCRIPTION AND CLASSIFICATION RESULTS

#	Class	Sample		Classification accuracy of methods (%)				
		Train	Test	NN	2D-CNN	DBN	DBN-softmax	FR DBN
1	Reed swamp	535	537	99.07	98.88	94.41	96.83	99.63
2	Spartina alterniflora	563	562	98.40	100	97.51	98.93	100
3	Water	681	680	82.06	100	97.21	98.97	100
4	Tamarisk shrub	281	280	93.93	92.14	95.36	95.71	99.64
5	Sparse tideland reeds	336	335	98.51	98.21	94.33	97.91	100
6	Suaeda salsa	507	506	97.43	98.42	91.90	97.04	98.42
7	Tide beach	530	529	96.98	91.87	94.14	96.98	93.19
8	Bare land	155	154	97.40	0	96.10	98.05	98.70
-	Total	3588	3583	-	-	-	-	-
	OA (%)			94.67	93.33	95.17	97.68	98.63
	AA (%)			95.47	84.94	95.12	97.55	98.70

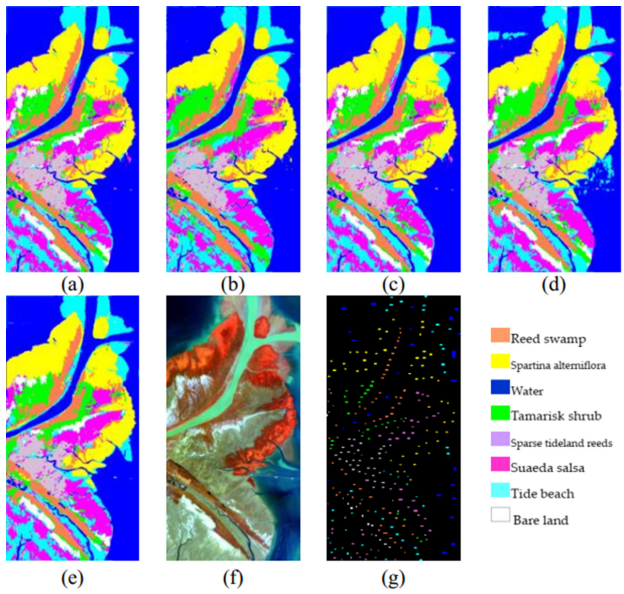

Fig. 11. Classification results on Yellow River Delta Coastal Wetland dataset. (a) NN. (b) 2-D CNN. (c) DBN. (d) DBN based on softmax classifier. (e) FR DBN. (f) False color image composite (bands 111, 69, and 13). (g) Sample distribution.

30 m. The Yellow River Delta Coastal Wetland dataset covers 677×339 pixels and the land cover ground truth has eight classes. We labeled 7174 samples for model training and testing and used the proposed FR DBN model (with a structure of 240-110-110-8) as an example CGDBN in the experiments. The other network parameters were set as follows: 200 unsupervised iterations, 52 RBM batch training samples, 200 supervised iterations, and 26 BP batch training samples. Experiments were performed using the same training samples as input for all methods. The samples and accuracy results of several methods are presented in Table VII. The classification results are shown in Fig. 11.

IV. Conclusion

In this article, we have proposed CGDBN classification models that are updated using the CG method and the 2-norm, based on the combined spectral and pixel-centric spectral block features. Models based on the FR algorithm or PRP algorithm improves the parameter update procedure in the BP network. We used the 2-norm to calculate the update factors and realize the updating of the hidden layer network parameters (weight and bias). The combination of spectral and pixel-centric spectral block features can mine the hyperspectral image features to improve the classification accuracy. The image enhancement processing of the principal components before spectral block extraction strengthens the spatial connection between the central pixel and the surrounding pixels. Classification experiments were performed on a hyperspectral image benchmark dataset, and the classification accuracy of the proposed models was found to be higher than that of classic DBN. We observed that the proposed models have advantages in terms of model convergence and model stability during the parameter adjustment process. Although the proposed models spend more time on training, the model convergence is comparable to that of advanced optimization algorithms. Comparing the two CGDBNs, FR DBN is superior to PRP DBN in terms of parameter sensitivity and computational complexity. Moreover, FR DBN obtained satisfactory results in an application to coastal wetlands imagery. The DBN classification model based on CG and the 2-norm provides a novel approach for hyperspectral classification.

References

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

[2] A. Santara *et al.*, "BASS Net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.

[3] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.

[4] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[5] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral-spatial classification of hyperspectral image using autoencoders," in *Proc. 9th IEEE Int. Conf. Inf., Commun. Signal Process.*, Dec. 2013, pp. 1–5.

[6] B. Fang, Y. Li, H. Zhang, and J. Chan, "Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 164–178, Jan. 2020.

[7] X. He, Y. Chen, and P. Ghamisi, "Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3246–3263, May 2019.

[8] M. Han, R. Cong, X. Li, H. Fu, and J. Lei, "Joint spatial-spectral hyperspectral image classification based on convolutional neural network," *Pattern Recognit. Lett.*, vol. 130, pp. 38–45, Feb. 2020.

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

[10] 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, Jul. 2006.

[11] G. Tong, Y. Li, L. Cao, and C. Jin, "A DBN for hyperspectral remote sensing image classification," in *Proc. 12th IEEE Conf. Ind. Electron. Appl.*, Jun. 2017, pp. 1757–1762.

[12] P. Zhong and Z. Gong, "A hybrid DBN and CRF model for spectral-spatial classification of hyperspectral images," *Stat., Optim. Inf. Comput.*, vol. 5, no. 2, pp. 75–98, Jun. 2017.

[13] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.

[14] C. Li, Y. Wang, X. Zhang, and H. Gao, "Deep belief network for spectral–spatial classification of hyperspectral remote sensor data," *Sensors*, vol. 19, no. 1, Jan. 2019, Art. no. 204.

[15] J. Li, B. Xi, Y. Li, Q. Du, and K. Wang, "Hyperspectral classification based on texture feature enhancement and deep belief networks," *Remote Sens.*, vol. 10, no. 3, p. 369, 2018.

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

[17] X. Zhou, S. Li, F. Tang, K. Qin, S. Hu, and S. Liu, "Deep learning with grouped features for spatial spectral classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 97–101, Jan. 2016.

[18] A. Mughees and L. Tao, "Multiple deep-belief-network-based spectral-spatial classification of hyperspectral images," *Tsinghua Sci. Technol.*, vol. 24, no. 2, pp. 183–194, Apr. 2019.

[19] 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.

[20] D. J. C. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[21] Y. L. Cun *et al.*, "Handwritten digit recognition with a back-propagation network," *Adv. Neural Inf. Process. Syst.*, vol. 2, no. 2, pp. 396–404, 1997.

[22] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, Toronto, ON, Canada: Springer, 2012, pp. 421–436.

[23] D. Das *et al.*, "Distributed deep learning using synchronous stochastic gradient descent," 2016, *arXiv:1602.06709*.

[24] J. Konečný, J. Liu, P. Richtárik, and M. Takáč, "Mini-batch semi-stochastic gradient descent in the proximal setting," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 2, pp. 242–255, Mar. 2016.

[25] G. P. Mccormick, "Anti-zig-zagging by bending," *Manage. Sci.*, vol. 15, no. 5, pp. 315–320, 1969.

[26] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[27] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence," *Sov. Math. Docl.*, vol. 27, no. 2, pp. 543–547, 1983.

[28] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, Jul. 2011.

[29] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[30] T. Tieleman and G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw. for Mach. Learn.*, 2012.

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, Dec. 2014, pp. 1–15.

[32] M. R. Hestenes and E. Stiefel, "Method of conjugate gradient for solving linear equations," *Res. Nat. Bur. Stand.*, vol. 49, no. 6, pp. 409–436, 1952.

[33] G. Yuan, X. Lu, and Z. Wei, "A conjugate gradient method with descent direction for unconstrained optimization," *J. Comput. Appl. Math.*, vol. 233, no. 2, pp. 519–530, Nov. 2009.
[34] R Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, no. 2, pp. 149–154, Jan. 1964.
[35] E. Polak and G. Ribière, "Note on the convergence of methods of conjugate directions," *ESAIM: Math. Modelling Numer. Anal. (ESAIM: M2AN)*, vol. 3, no. 16, pp. 35–43, 1969.
[36] B. T. Polyak, "The conjugate gradient method in extreme problems," *USSR Comp. Math Math. Phys.*, vol. 9, no. 4, pp. 94–112, Dec. 1969.
[37] N. L. Roux, Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural Comput.*, vol. 20, no. 6, pp. 1631–1649, Jun. 2008.
[38] Y. Bengio, *Learning Deep Architectures for AI*. Delft, The Netherlands: Now Foundations and Trends, 2009.
[39] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Proc. Neural Netw., Tricks Trade*, 2012, pp. 599–619.

Chen Chen (Student Member, IEEE) received the bachelor's degree in mathematics and applied mathematics from the College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao, China, in 2015, and the master's degree in physical oceanography from the First Institute of Oceanography, Ministry of Natural Resources, Qingdao, China, in 2018. She is currently working toward the Ph.D. degree at the College of Geomatics, Shandong University of Science and Technology.

Her research interests include hyperspectral image processing, deep learning, and remote sensing applications.

Yi Ma received the Ph.D. degree with Institute of Oceanology, Chinese Academy of Sciences, Qingdao, China, in 2005.

He is currently a Professor with the Marine Remote Sensing Division, the First Institute of Oceanography, Ministry of Natural Resources, Qingdao, China. He has authored or coauthored more than 50 peer-reviewed papers, and he is the Inventor or Coinventor of three patents. His research interests include marine hyperspectral remote sensing, deep learning, high-resolution remote sensing of island and coastal zone, and active and passive remote sensing detection in shallow seawater depth.

Guangbo Ren received the Ph.D. degree in ocean information detection and processing from Ocean University of China, Qingdao, China, in 2010.

He is currently an Associate Professor with the Marine Remote Sensing Division, First Institute of Oceanography, Ministry of Natural Resources, Qingdao, China. He has authored or coauthored more than 30 peer-reviewed papers. His research interest includes high-resolution remote sensing of coastal wetlands.