Hyperspectral Image Classification With Convolutional Neural Network and Active Learning

Xiangyong Cao, *Member, IEEE*, Jing Yao, Zongben Xu, and Deyu Meng, *Member, IEEE*

***Abstract*— Deep neural network has been extensively applied to hyperspectral image (HSI) classification recently. However, its success is greatly attributed to numerous labeled samples, whose acquisition costs a large amount of time and money. In order to improve the classification performance while reducing the labeling cost, this article presents an active deep learning approach for HSI classification, which integrates both active learning and deep learning into a unified framework. First, we train a convolutional neural network (CNN) with a limited number of labeled pixels. Next, we actively select the most informative pixels from the candidate pool for labeling. Then, the CNN is fine-tuned with the new training set constructed by incorporating the newly labeled pixels. This step together with the previous step is iteratively conducted. Finally, Markov random field (MRF) is utilized to enforce class label smoothness to further boost the classification performance. Compared with the other state-of-the-art traditional and deep learning-based HSI classification methods, our proposed approach achieves better performance on three benchmark HSI data sets with significantly fewer labeled samples.**

***Index Terms*— Active learning (AL), convolutional neural network (CNN), deep learning, hyperspectral image (HSI) classification, Markov random field (MRF).**

I. Introduction

HYPERSPECTRAL images (HSIs), which are captured by hyperspectral remote sensors, have been one of the most important data types in the field of remote sensing. Different from traditional natural image, HSIs contain hundreds of continuous narrow bands, of which each represents the imaging at a certain frequency, and thus can provide richer and more detailed information on the ground surface or some targets for various practical cases [1]–[3].

HSI classification, which aims to categorize each hyperspectral pixel vector into a discrete set of meaningful classes according to the image contents, has been extensively studied and has also shown promising prospect in a variety of applications, such as land-use mapping [4]–[6], land-cover [7], forest inventory [8], and urban-area [9]. In the last few decades, researchers have proposed a large number of HSI classification methods, which can be roughly divided into two categories, namely spectral-based approach and spectral–spatial-based approach.

The spectral-based approach makes use of spectral information only in the classification process via some strategies, such as principle component analysis [10] and linear discriminant analysis [11]. However, the methods raised along this line ignore the spatial correlation information and thus cannot obtain excellent performance. Comparatively, spectral–spatial-based approach fully considers both spectral and spatial information in order to tackle this issue. Typical methods in this category are represented by the patch-based methods [12]–[14] which extract features in a local window, the Markov random field (MRF) methods [15]–[19] which utilize the prior that spatially neighboring pixels are very probable to have the same label and some other spectral–spatial methods, such as low-rank representation methods [20]–[22], the wavelet transform-based methods [17], [23]–[25], profile-based method [26], and manifold learning methods [27], [28].

In recent years, deep learning methods have been applied to the task of HSI classification, such as stacked auto-encoder (SAE) [29], deep belief networks (DBNs) [30], and convolutional neural network (CNN) [31]–[37]. Although these methods perform well in a purely data-driven manner, they generally heavily rely on a large number of precollected labeled data. However, it is difficult to acquire the annotated data in practice since labeled data always cost much time and money, and especially requires experts to possess certain domain knowledge in many cases. In this article, we try to alleviate this issue to possibly reduce the labor of annotating data on the guarantee of accuracy when using deep learning techniques in the HSI classification task.

To tackle this issue, active learning (AL) is a promising approach since it can select useful samples for the users to label and also has a theoretical guarantee to significantly

Manuscript received September 5, 2019; revised November 27, 2019 and December 31, 2019; accepted January 2, 2020. This work was supported in part by the China Postdoctoral Science Foundation Funded Project under Grant 2018M643655, in part by the Fundamental Research Funds for the Central Universities, in part by the National Key Research and Development Program of China under Grant 2018YFB1004300, in part by MoE-CMCC "Artifical Intelligence" Project under Contract MCM20190701, and in part by the China NSFC Project under Contract 61906151, Contract 11690011, Contract 61603292, Contract 61721002, Contract U1811461, and Contract 91860125. *(Corresponding author: Deyu Meng.)*

Xiangyong Cao, Jing Yao, and Zongben Xu are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: caoxiangyong@mail.xjtu.edu.cn; jasonyao92@gmail.com; zbxu@mail.xjtu.edu.cn).

Deyu Meng is with the Faculty of Information Technology, Macau University of Science and Technology, Taipa 999078, Macau, and also with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dymeng@mail.xjtu.edu.cn).

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

Digital Object Identifier 10.1109/TGRS.2020.2964627

reduce the number of labeled samples [38]. AL assumes that the training samples are not equally important to a classifier. More specifically, only a few samples define the separating surface and most other ones are redundant to the classifier. Therefore, the redundant labeled samples can be avoided if we can discover useful samples. Additionally, the labeling cost can also be reduced in this way, which motivates us to combine AL with deep learning.

In this article, we propose a method to integrate both deep learning and AL methodology into a unified framework by fully utilizing the benefits of both worlds, namely strong discriminative ability of deep learning and labeling efficiency of AL. Although there have been several works combining AL with deep learning for HSI classification [39]–[41], our proposed method has its specific characteristics. First, the proposed method adopts different deep network architecture, namely CNN, while the previous methods use other network architectures, such as SAE [39], restricted Boltzmann machine (RBM) [40], and Bayesian CNN (BCNN) [41]. Second, the proposed method uses different criteria to actively select informative samples. Third, the proposed method considers contextual information by using MRF. More details about these differences will be discussed in related work. Therefore, this proposed method can be regarded as a further attempt to combine AL with a deep neural network as well as consider more contextual information in order to reduce the labeling cost.

Except for the aforementioned deep learning and AL techniques, our proposed method also employs other novel techniques. More specifically, we utilize data augmentation (DA) strategy [42] including translating, flipping, rotating, and cropping, to generate new samples to help alleviate the issue of insufficient labeled samples. Additionally, since the training sample set is updated by actively selecting some new informative samples after each round, CNN needs to be retrained. In the proposed method, we adopt a fine-tuning (FT) strategy [43] to largely reduce the cost of retraining CNN after each round, and thus reduce the computational complexity. Also, the batch normalization (BN) [44] training strategy is used to help train the CNN. In the experimental section, we will investigate how each of the adopted techniques influences the performance of the proposed HSI classification method.

In summary, motivated by alleviating the issue of insufficient labeled samples for the training of deep learning-based HSI classification methods, we embed the AL strategy, which can select the most informative and uncertain samples by querying for labeling, into the graph-based CNN model, and propose a new supervised HSI classification method. This method can reduce the number of labeled sample needed for the training of the CNN model while still keep or even improve the classification performance. More specifically, the contributions of this proposed method are mainly threefold which are as follows.

1) We propose a new method that integrates the AL strategy with CNN, and thus can fully exploit the merits of both worlds. Specifically, a CNN is used to extract spectral–spatial discriminative features from 3-D patches of HSI, and output the class label predictions of the pixels in the candidate pool. To reduce the labeling cost, an AL strategy is then adopted to select the most useful or informative pixels to require human annotations from the candidate pool based on the output of CNN. Besides, the DA technique is also utilized to alleviate this issue.
2) To further exploit the spatial local correlation information, the MRF, which assumes that adjacent pixels are more likely to belong to the same class, is utilized to model this local correlation.
3) The experimental results on three HSI benchmark data sets illustrate that the proposed HSI classification method outperforms other state-of-the-art traditional and deep learning-based HSI methods with fewer labeled samples.

The rest of this article is organized as follows: In Section II, the related work regarding deep learning and AL-based HSI classification methods are briefly reviewed. In Section III, we present the proposed model in detail. In Section IV, we conduct a series of experiments on three HSI benchmark data sets. The conclusion is provided in Section V.

II. Related Work

A. Deep Learning-Based HSI Classification

More recently, deep learning methods have been extensively studied for HSI classification, such as SAEs) [45], DBNs [30], deep Boltzmann machines (DBMs) [46], and CNNs [32], [34]–[37], [47], [48]. Specifically, SAE [45] is an unsupervised feature extraction method by stacking a series of auto-encoders and can extract spectral–spatial features [45]. Besides, CNN has been extensively applied to HSI classification. For example, a naive CNN model with five layers is proposed [32], but this model considers only spectral information. To tackle this issue, a modified CNN [47] using the 3-D patch as the network's input is proposed, which utilizes both spectral and spatial information. Then, CNN combining with spatial pyramid pooling strategy [33] is presented to fully considering the spatial information. Subsequently, CNN features combining with hand-crafted features and conditional random field (CRF) [49] are proposed. Also, CNN with MRF [34] is proposed to further make use of the label correlations. A dual-channel CNN [50] which applies both 1-D CNN and 2-D CNN to extract feature is presented. To expand the training set for training deep CNN, a novel pixel-pair method [48] is proposed. Additionally, a 3-D convolutional neural network (3D-CNN) [36] approach that enables a joint spectral and spatial information is proposed. Similarly, a 3-D contextual deep CNN (3D-FCN) [37] is proposed to optimally explore local contextual interactions of neighboring individual pixel vectors.

B. AL Based HSI Classification

In recent years, AL has been extensively studied in HSI classification. For example, a semi-supervised multinomial logistic regression model combining entropy (EP)-based active selection strategy [51] is first proposed. After that, the Bayesian classification approach and loopy belief propagation method

combining AL strategies [15], [52] are studied. Later, an MRF model-based AL framework [53] is proposed, and support vector machine (SVM) classifier with six different AL sampling strategies [53] is also designed for HSI classification.

More recently, a few works that combine AL with deep learning have also been studied for HSI classification [39]–[41]. Specifically, Li [39] proposed a method to combine a multiclass-level uncertainty (MCLU) active criterion with SAE neural network. Liu *et al.* [40] presented a method to combine a weighted incremental dictionary learning criterion with RBM. Haut *et al.* [41] proposed a method to combine six AL criteria, such as random acquisition, maximum EP, breaking ties, mutual information, etc., with BCNN.

Although these methods have achieved excellent performance, our proposed method is quite different from them. First, the proposed method adopts a different deep network architecture CNN rather than SAE, RBM, and BCNN. Second, the proposed method uses a new multiclass criterion, namely Best-versus-Second Best (BvSB), to actively select the informative samples. Third, the proposed method employs several novel techniques to accelerate training or reduce computation, such as DA, FT, and BN. Finally, the proposed method further considers contextual information using MRF while the existing methods do not utilize this prior. All the four points are the main differences of the proposed method with the existing related methods.

III. Proposed Model

In this section, we first define the notations used in this article and then introduce the proposed model.

Given a HSI data set, we denote it as $\mathcal{H} \in \mathbb{R}^{H\times W\times D}$, where H and W are the height and width of the spatial image, respectively, and D is the number of spectral bands. We define $N = HW$ as the total number of pixel, and $\mathcal{L} = \{c_1, c_2, \ldots, c_n\}$ as the available labeled pixel set, where c_i is the ith pixel vector, n is the number of labeled pixel and $n \ll N$. Additionally, the training sample set fed into a CNN (shown in Fig. 2) is $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Here x_i is a 3-D patch of pixel c_i and includes c_i as the central pixel vector, and y_i belongs to the label set $\mathcal{K} = \{1, 2, \ldots, K\}$, where K is the number of classes. We denote the unlabeled pixel set as $\mathcal{U} = \{c_{n+1}, c_{n+2}, \ldots, c_N\}$. HSI classification task aims to assign a label y_i to each pixel c_i. For simplicity, we denote all the labels of pixels as $Y = \{y_i\}_{i=1}^N$.

A. Proposed Model

In order to clearly illustrate this proposed method, we provide a framework of this proposed approach in Fig. 1. Overall, this framework combines the AL strategy with deep learning in order to reduce the labeled samples for training. More specifically, the proposed approach contains the following steps. First, we construct an initialized training patch set $\mathcal{D}$ corresponding to a limited number of randomly selected labeled pixels. Next, we augment the training set into a new one, which is used to conduct the first training of the CNN. Then, we actively select the most informative or useful pixels from the candidate pool based on the class probabilities provided by the trained CNN and then add the corresponding patches of the selected pixels into the current training set, which is further regarded as a new training set for the next iterative round. This step together with the previous step is implemented iteratively until the stopping criterion is satisfied. Finally, MRF is utilized to enforce class label smoothness. Next, we introduce each component of this method in detail.

1) DA: In this section, the DA method is adopted to generate an enlarged training set, which is further regarded as the input for CNN. Specifically, transformation strategy is used to augment the training set $\mathcal{D}$, such as flipping horizontally and vertically, respectively, and rotating 90Â°, 180Â°, and 270Â° clockwise, respectively. After conducting DA, the training sample set $\mathcal{D}$ has a fivefold increase and the new training sample set is denoted as $\mathcal{D}_A$.

2) Graph-Based Deep Learning Model: In this section, we present our proposed graph-based deep learning model, which combines the strong discriminative power of CNN and the contextual constraint of the graph model. Specifically, the objective function of this model is defined as

$$L(Y, F_\Theta|\mathcal{H}, \mathcal{D}_A) = L_u(F_\Theta|\mathcal{H}, \mathcal{D}_A) + L_p(Y|\mathcal{H}) \tag{1}$$

where F_Θ is a nonlinear function implemented by CNN with parameter Θ, L_u is the unary term which aims to predict the class probability of each pixel by using CNN, and L_p is the pairwise term which enforces adjacent pixels to have the same label. Here, label Y and parameter Θ are the variables to optimize. Next, we will introduce the two terms, respectively.

a) Unary Term: The unary term L_u is proposed to predict the class probability of each pixel by using the CNN since CNN has exhibited strong discriminative ability in a wide range of computer vision tasks [54]–[56]. In this article, we adopt the CNN structure shown in Fig. 2, which contains one input layer, two pairs of convolution and max pooling layers, two fully connected layers, and a softmax output layer. Therefore, the unary term can be defined as

$$L_u(F_\Theta|\mathcal{H}, \mathcal{D}_A) = -\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y_i = k\} \log P(y_i = k|x_i, F_\Theta) \tag{2}$$

where $1\{\cdot\}$ is the indicator function, and $P(y_i = k|x_i, F_\Theta)$ is the output of the CNN and represents the probability of x_i to have label k.

b) Pairwise Term: The pairwise term L_p utilizes the local correlation of labels, which means that spatially adjacent pixels are probable to have the same class label. Specifically, this term is defined as

$$L_p(Y|\mathcal{H}) = \gamma \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} T_{ij} \tag{3}$$

where γ is a nonnegative constant parameter that controls the level of spatial smoothness, $\mathcal{N}_i$ is the neighboring pixels of pixel i, and T_{ij} is the spatial interaction term defined as

$$T_{ij} = [1 - \delta(y_i, y_j)] \exp\left(-\frac{\|c_i - c_j\|_2^2}{2\sigma}\right) \tag{4}$$

where $\delta(\cdot)$ is the Kronecker function ($\delta(a, b) = 1$ for $a = b$ and $\delta(a, b) = 0$ otherwise), and σ is a scale parameter.

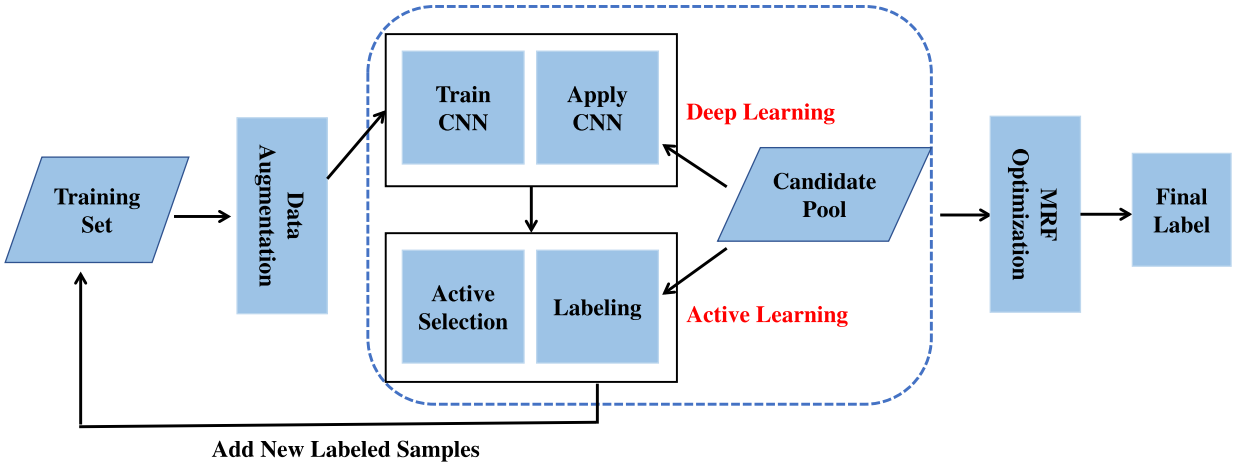


Fig. 1. Proposed active deep learning framework for HSI classification.

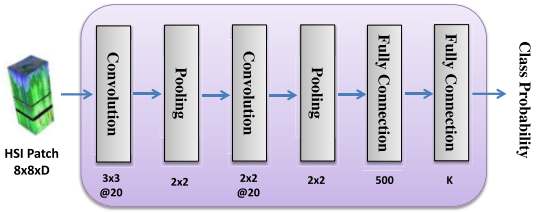


Fig. 2. Network structure of the CNN used for HSI classification. The input HSI patch is with size $8 \times 8 \times D$. The first convolutional layer contains 20 filters with size 3×3, and the second convolutional layer includes 20 filters with size 2×2. After each convolutional layer, there is a max-pooling layer with kernel size 2×2 and a stride of 2 pixels. The first fully connected layer contains 500 units, and the unit number of the second fully connected layer is the class number K.

This pairwise term actually incorporates the information associated with class edges and tends to make adjacent pixels have the same label, which means it encourages the classification boundary to align with strong edges. More specifically, for neighboring pixels i and j within the image flat region, $\exp[-(\|c_i - c_j\|_2^2/2\sigma)]$ is large (close to 1), then y_i and y_j can have the same class label after model optimization. As for neighboring pixels i and j across a strong edge, $\exp[-(\|c_i - c_j\|_2^2/2\sigma)]$ is small, and thus y_i and y_j can take different labels after model optimization.

3) AL: AL is a popular strategy of selecting the most informative samples by querying for labeling in an iterative way. A variety of heuristic AL strategies have been proposed in the machine learning field, such as uncertainty sampling [57], expected model change [58], variance reduction [59], estimated error reduction [60], and density-weighted methods [59]. Since we can get the class membership probability estimates from the CNN, some effective AL criteria based on the class probability can thus be adopted, such as EP measure and BvSB measure [38]. Both of the two criteria belong to the uncertainty sampling method and can help select the informative candidates to query for annotations. Next, we introduce the two measures separately in detail.

a) EP Measure: For each unlabeled sample, we can consider the class membership to be a random variable $z = (z_1, z_2, \ldots, z_K) \in \mathbb{R}^K$. We have a distribution P for z of estimated class membership probabilities computed in this way, namely $P(z_{ik}) = P(y_i = k|x_i, F_\Theta)$, where z_{ik} represents the kth element of variable z_i. Since EP can measure the uncertainty of class membership, thus the multiclass discrete EP of z_i is

$$H(z_i) = -\sum_{k=1}^{K} P(z_{ik}) \log P(z_{ik}). \tag{5}$$

For this measure, if a sample has a distribution with high EP, the classifier is uncertain about its class membership and thus this sample will be selected since it is expected to contribute more on the performance of CNN.

b) BvSB Measure: BvSB measure is specially designed for the multiclass classification problem and thus is very suitable for the multiclass HSI classification problem. Additionally, BvSB measure can alleviate the issue that the performance is heavily influenced by small class probabilities of unimportant classes, by measuring the class probability difference between the most confused classes, i.e., the first and the second most probable classes. Specifically, this criterion is defined as

$$BvSB(z_i) = P_B(z_i) - P_{SB}(z_i) \tag{6}$$

where $P_B(z_i)$ denotes the best class membership probability of sample z_i, which can be computed as $P_B(z_i) = P(z_{ik})$ where $k = \arg\max_l P(z_{il})$, and $P_{\mathrm{SB}}(z_i)$ is the second best class membership probability of sample z_i, which can be computed in a similar way. For this measure, if a sample has a small BvSB value, the classifier is confused with its class membership and thus this sample will be selected in the consequent training.

Overall, the two AL criteria are implemented in the following iterative way. In each round, we compute the class membership probabilities of all the samples in the active pool $\mathcal{U}$, and samples with the high EP value or low BvSB value are selected to request for labeling. After the labels are obtained, the corresponding samples are incorporated in the training set, and then the CNN is retrained. In the experimental

section, we will compare the two criteria as well as the random selection (RS) strategy. Here, it should be emphasized again that the benefit of using AL strategy is to reduce the labeling cost, and it is necessary to adopt the AL strategy since the HSI classification task suffers the insufficient labeled samples problem. Also, it should be noted that it is infeasible to collect and label portion of samples at one time in practice. This is due to the fact that we generally could only have limited number of supervised labeled samples, not sufficient to facilitate us to obtain a faithful classifier and help obtain enough number of "informative" samples at one time for manual labeling to guide a confident subsequent classifier learning. Besides, the manner we have selected samples for labeling is not "random" at all, but implemented under a rational guidance of elaborately designed selection measures (i.e., RS, EP, and BvSB).

4) Optimization: By combining (1)–(3), the final objective function can be denoted as

$$\begin{aligned} L(Y, F_\Theta|\mathcal{H}, \mathcal{D}_A) &= -\sum_{i,k} 1\{y_i = k\} \log P(y_i = k|x_i, F_\Theta) \\ &\quad + \gamma \sum_{i, j\in\mathcal{N}_i} [1-\delta(y_i, y_j)] \exp\left(-\frac{\|c_i - c_j\|_2^2}{2\sigma}\right) \end{aligned} \tag{7}$$

where class labels Y and parameters Θ of CNN are the variables to be optimized. For simplicity, this optimization problem can be divided into two subproblems. First, we calculate the network parameter Θ given label Y. Then, given the parameter Θ (namely a learned CNN F_Θ), label Y can be updated by solving an MRF optimization problem. In the following, we will present the details of the two steps.

a) Updating Θ: The objective function of this subproblem can be expressed as

$$L(\Theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y_i = k\} \log P(y_i = k|x_i, F_\Theta). \tag{8}$$

The objective function can be optimized by the stochastic gradient descent (SGD) algorithm.

For this subproblem, some points need to be emphasized. First, the optimization is an iterative process with many rounds, of which each contains many iterations (here t is the iteration number of each round). In each round, we have a different training sample set $\mathcal{D}_A$ since AL strategy is adopted to select some informative candidates from the candidate pool and then add them into the training sample set. More specifically, before starting a new round, AL measure is used to select the pixels with minimal BvSB value or maximum EP value from candidate pool $\mathcal{U}$ based on the output of class membership probability of the CNN trained in the previous round. Second, when training the CNN in each round (except the first round), we start with setting the parameter Θ of CNN as the values of the final iteration of the previous round and then fine-tune these parameters based on the current training sample set. In this way, the computational complexity can be reduced since less epochs are needed for the convergence of this model, and performance can be better since this way is more robust than completely retraining.

b) Updating Y: The objective function related to label Y can be defined as

$$\begin{aligned} L(Y) &= -\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y_i = k\} \log P(y_i = k|x_i, F_\Theta) \\ &\quad + \gamma \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} [1-\delta(y_i, y_j)] \exp\left(-\frac{\|c_i - c_j\|_2^2}{2\sigma}\right). \end{aligned} \tag{9}$$

Optimizing (9) is a NP-hard combinatorial problem, which can be reformulated as an MRF model [34]. More specifically, label Y forms an MRF over the graph, and (9) is its energy function, of which the first term represents the cost of a pixel being assigned with different classes, and the second term encourages the labels of neighboring pixels to be the same. Therefore, optimizing Y has been turned into a labeling problem in an MRF, and many approximating algorithms can be utilized, including graph cut [61], [62], belief propagation [63], and message passing [64]. In this article, we adopt the belief propagation algorithm due to its fast convergence.

Algorithm 1 CNN-AL-MRF Algorithm

Input: Training sample set $\mathcal{D}$, unlabeled pixel set $\mathcal{U}$, the number of round R, the number of initialized training pixels a, and the number of actively selected pixel in each round b.
Initialization: $r = 0$
while $r < R$ or stopping criterion is not satisfied **do**
1: Data augmentation: $\mathcal{D} \rightarrow \mathcal{D}_A$
2: CNN training ($r = 1$) or fine-tuning ($r > 1$) based on $\mathcal{D}_A$
3: Compute class probability of each sample in candidate pool $\mathcal{U}$ based on trained CNN
4: Actively select b pixels from $\mathcal{U}$ via BvSB criterion
5: Add corresponding patches of the selected pixels into $\mathcal{D}$
6: Remove selected pixels from $\mathcal{U}$
End while
7: MRF optimization
Output: Labels Y

5) Summary: After presenting how to update parameters Θ of CNN and class label Y, respectively, we can summarize the proposed CNN-AL-MRF method in Algorithm 1. More specifically, the proposed CNN-AL-MRF approach is implemented as follows: First, an initialized training patch set $\mathcal{D}$ corresponding to a limited number of randomly selected a labeled pixels is constructed. Next, the training set $\mathcal{D}$ is augmented into a new training set $\mathcal{D}_A$, which is used for the first training of the CNN. Then, we actively select b most informative or useful pixels from the candidates pool $\mathcal{U}$ based on the class probabilities provided by the trained CNN and then add the corresponding patches of the b selected pixels into the current training set $\mathcal{D}$, which is further regarded as a newly training set for the next iterative round. This step together with the previous step is implemented iteratively until the stopping criterion is satisfied. Finally, MRF is utilized to enforce class label smoothness.

In order to make the following experiments repeatable, we also release the code at https://github.com/xiangyongcao/CNN-AL-MRF.

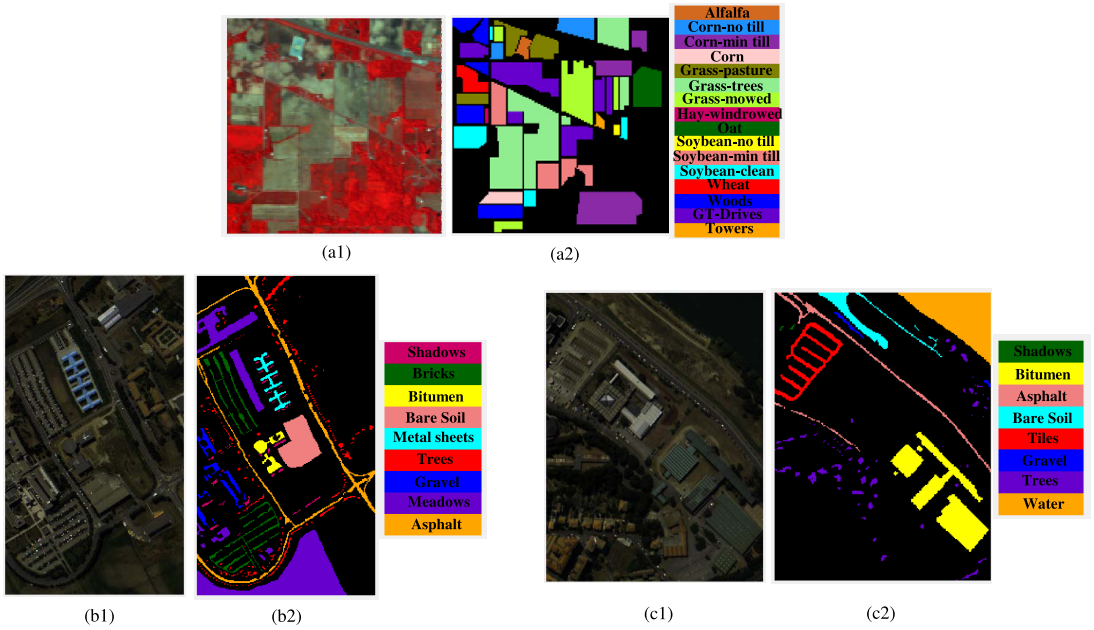


Fig. 3. Experimental data sets. (a1) False color image of Indian Pines data set. (a2) Ground truth categorization map and color codes of Indian Pines data set. (b1) False color image of Pavia University data set. (b2) Ground truth categorization map and color codes of Pavia University data set. (c1) False color image of Pavia Center data set. (c2) Ground truth categorization map and color codes of Pavia Center data set.

IV. Experiments

In this section, we conduct a series of experiments on three real-world benchmark data sets to evaluate the performance of our proposed CNN-AL-MRF approach. In Section IV-A, we give a brief introduction to the experimental data sets. In Section IV-B, we discuss the parameter settings involved in the proposed method. In Section IV-C, we conduct experiments to verify the effectiveness of each novel technique used in the proposed method. In Section IV-D, we report the experimental results on three benchmark data sets, where the proposed method is compared with seven state-of-the-art supervised HSI classification methods, including four traditional methods, namely SVM graph-cut method (SVM-GC) [65], SVM based on low-rank feature method (SVM-LR) [66], SVM based on 3-D Gabor wavelet (SVM-3DGW) [24]), and SVM based on the discrete wavelet transform method (SVM-3DWT)[1] [17], and four deep learning-based methods, namely 3-D 3D-CNN[2] [36], 3-D fully convolutional network (3D-FCN)[3] [37], SAE[4] [45] and CNN with Markov random field (CNN-MRF)[5] [34]. The proposed CNN-AL-MRF approach is implemented in the MATLAB environment using the MatConvNet library on a PC with Nvidia GeForce GTX 1080Ti and 32-GB memory.

All the methods are compared numerically via the following criteria: overall accuracy (OA) and average accuracy (AA). OA represents the number of correctly classified samples divided by the total number of test samples, and AA denotes the average of individual class accuracies (CAs), which means the number of correctly classified samples in one class divided by the total number of samples in this class. For all the criteria, larger values indicate better classification performance.

[1]Code is available at: https://github.com/xiangyongcao/3DDWT-SVM-GC
[2]Code is available at: https://github.com/nshaud/DeepHyperX
[3]Code is available at: https://github.com/nshaud/DeepHyperX
[4]Code is available at: https://github.com/hantek/deeplearn_hsi
[5]Code is available at: https://github.com/xiangyongcao/CNN_HSIC_MRF

A. Data Description

In this section, we introduce the three benchmark HSI data sets used in our experiments, and Fig. 3 shows the example images.

The first benchmark data set is Indian Pines, which was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in North-western Indiana. The data consist of 145 $\times$ 145 pixels and 220 spectral reflectance bands in the wavelength range 0.4–2.5 μm. This scene is a subset of a larger one. The Indian Pines scene contains two-thirds of agricultural land and one-third of forest or other natural perennial vegetation. There are two major dual-lane highways, a rail line, as well as some low-density housing, other built structures, and smaller roads. The ground truth available is designated into 16 classes and is not all mutually exclusive. A sample image and the ground-truth classification map of this data set are shown in Fig. 3(a1) and (a2).

The second data set is Pavia University, which was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia, Northern Italy on July 8, 2002. The number of spectral bands is 103 for Pavia University ranging from 0.43 to 0.86 μm. Pavia University is with 610 $\times$ 610 pixels, but some of the samples in both images contain no information and have to be discarded before analysis. Then the spatial dimension of Pavia University becomes 610 $\times$ 340. There are nine land cover classes for this scene. A sample image and the

TABLE I
OA (%) WITH DIFFERENT AL STRATEGIES

#Samples / AL strategy	RS	EP	BvSB
2nd round: 500	94.11(0.31)	93.36(1.17)	**96.03(1.59)**
3rd round: 650	94.70(0.88)	96.21(0.50)	**98.36(0.60)**
4th round: 750	96.31(0.72)	97.37(0.58)	**99.29(0.25)**
5th round: 800	96.75(0.41)	97.78(0.75)	**99.49(0.10)**

ground-truth classification map of this data set are shown in Fig. 3(b1) and (b2).

The third data set is Pavia Center, which was also acquired by the ROSIS sensor. The number of spectral bands is 102 for Pavia Center. The spatial dimension of Pavia Center is 1096 × 715. Since some of the samples in images contain no information and have to be discarded before analysis, we thus crop it into the size of 400 × 300 × 102. There are eight land cover classes for this scene. A sample image and the ground-truth classification map of this data set are shown in Fig. 3(c1) and (c2).

B. Parameter Settings

In this section, we discuss the parameter settings involved in the proposed method. Concretely, we use the Indian Pines data set to analyze the performance sensitivity of the proposed method with different parameters settings. In this experiment, limited by the space of this article, we discuss only two important parameters, namely the type of AL strategy and the smoothness parameter γ. For other parameters, we only provide empirical settings from large off-line experiments.

1) Different Strategies of AL: First, we evaluate the influence of three different AL strategies, namely RS, EP, and BvSB. For this experiment, we conduct five rounds of CNN training. The first round contains 250 training samples and 800 epochs, while the subsequent four rounds contain 400, 400, 300, and 200 epochs, respectively. For each of the subsequent four rounds, we actively select 250, 150, 100, and 50 samples to enlarge the training set, respectively. Additionally, we set the smoothness parameter as ten in this experiments. This experiment is repeated five times and the average OA and standard derivation are reported. The experimental results are shown in Table I. The result of the first round is not reported since no AL strategies are used in the first round and thus all the methods obtain the same result. Also, it can be easily seen from Table I that BvSB strategy achieves the best performance in all the round, and thus we adopt the BvSB AL strategy in the following experiments.

2) Smoothness Parameter: Second, we investigate the performances sensitivity of the proposed CNN-AL-MRF method with different smoothness parameters, namely 0.1, 1, 5, 10, 15, and 20. For this experiment, five rounds of CNN training are also conducted. 800 epochs and 250 training samples are set for the first round, and 400, 400, 300, and 200 epochs are set respectively for the subsequent four rounds. We actively select 250, 150, 100, and 50 samples to enlarge the training set in each of the subsequent four rounds. Additionally, we adopt the BvSB AL strategy in these experiments. This experiment is repeated five times and the average OA and standard derivation are reported. The experimental results are shown in Table II. From Table II, we can easily observe that the proposed method performs almost the best in each round when γ equals 10, and thus we set it as 10 in the following experiments.

Additionally, the number of initialized training samples a has a great influence on the final classification performance since subsequent active sample selection step relies heavily on the performance of initialized trained CNN, and thus it needs to be carefully tuned. Empirically, we set a in the range of [150, 300]. Also, the number of round R and the number of added samples in each round b are both important for the final performance. Empirically, we set b for each round in the range of [50, 200] and R is set in the range of [5, 10]. Moreover, we empirically obtain the parameter settings of the CNN structure, which is shown in Fig. 2. Some other parameters are also empirically set. For example, the batch size is 50, the learning rate α is 0.001, and the scale parameter σ is 1.

C. Effectiveness Verification

In this section, we conduct an experiment to evaluate the effectiveness of each novel technique, including DA, BN, AL, network FT, and MRF. In this experiment, the baseline method is CNN + DA + BN (CNN-DB), which contains two common deep learning techniques, namely DA and BN. Then, we gradually add each of the other technique into the CNN-DB method. Additionally, to better illustrate the effectiveness of DA and BN, we also investigate the pure CNN method (CNN) and the CNN + DA method (CNN-D). In summary, we propose the following methods.

1) *CNN:* Conventional CNN method.
2) *CNN-D:* The method that combines DA strategy with the CNN method.
3) *CNN-DB:* The method that embeds BN trick into the CNN-D method.
4) *CNN + DA + BN + AL (CNN-DBA):* The method that incorporates AL strategy with CNN-DB method. The network parameters Θ are randomly initialized and trained from scratch in each round.
5) *CNN + DA + BN + AL + FT (CNN-DBAF):* The method that combines FT strategy with CNN-DBA method. In each round (except the first round), the network parameters are inherited from the previous round.
6) *CNN + DA + BN + AL + FT + MRF (CNN-AL-MRF):* The method that is using MRF post-processing strategy in the CNN-DBAF method.

More specifically, we use Indian Pines as the data set, and the specific settings are shown as follows. We conduct two rounds of CNN training. For non-AL-based methods including CNN, CNN-D, and CNN-DB, which do not use AL strategy, each round contains 800 epochs, while for AL-based methods including CNN-DBA, CNN-DBAF, and CNN-AL-MRF, the first round contains 800 epochs and the second one contains 400 epochs. Additionally, for all the methods, we randomly select 250 training samples in the first round. Then, in the second round, we randomly add 250 samples into this training set for non-AL methods, and actively select 250 samples for AL methods. Each method is

TABLE II
OA AND STANDARD DERIVATION (%) IN EACH ROUND WITH DIFFERENT SMOOTHNESS PARAMETER γ

#Samples / γ	0.1	1	5	10	15	20
1st round: 250	83.61(4.02)	85.42(2.63)	86.54(2.38)	87.84(1.81)	**88.01(1.77)**	87.87(2.15)
2nd round: 500	94.86(1.89)	95.80(1.51)	96.01(1.63)	**96.03(1.59)**	95.97(1.64)	95.94(1.51)
3rd round: 650	97.55(1.33)	97.92(1.01)	98.35(0.71)	**98.36(0.60)**	98.21(0.88)	98.18(0.79)
4th round: 750	98.62(0.47)	99.05(0.32)	99.24(0.26)	**99.29(0.25)**	98.89(0.51)	98.70(0.40)
5th round: 800	99.12(0.31)	99.38(0.17)	99.40(0.13)	**99.49(0.10)**	99.17(0.21)	99.16(0.19)

TABLE III
OA VALUES AND STANDARD DERIVATION (%) OF DIFFERENT METHODS THAT CONTAINS DIFFERENT TECHNIQUES

Settings/Method	CNN	CNN-D	CNN-DB	CNN-DBA	CNN-DBAF	CNN-AL-MRF
1st round, 250 samples	36.43(1.76)	66.18(2.04)	83.82(0.54)	83.82(0.54)	83.82(0.54)	87.83(1.81)
2nd round, 500 samples	45.59(1.26)	77.25(3.00)	90.94(0.92)	93.56(0.61)	94.42(0.52)	96.03(1.58)

TABLE IV
STATISTICS OF THE INDIAN PINES DATA SET, INCLUDING THE NAME, THE NUMBER OF TRAINING, TEST AND TOTAL SAMPLES FOR EACH CLASS

Class		Samples		
No	Name	Train	Test	Total
1	Alfalfa	3	43	46
2	Corn-no till	72	1356	1428
3	Corn-min till	42	788	830
4	Corn	12	225	237
5	Grass-pasture	25	458	483
6	Grass-trees	37	693	730
7	Grass-pasture-mowed	2	26	28
8	Hay-windrowed	24	454	478
9	Oat	1	19	20
10	Soybean-no till	49	923	972
11	Soybean-min till	123	2332	2455
12	Soybean-clean	30	563	593
13	Wheat	11	194	205
14	Woods	64	1201	1265
15	Buildings-Grass-Trees-Drives	20	366	386
16	Stone-Steel-Towers	5	88	93
Total		520	9729	10249

repeated five times, and the average OA value and standard derivation are reported. The experimental results are reported in Table III.

From Table III, it can be easily observed by column that each method performs better as the round goes, which is an obvious result since more training samples are added into the training set after a round. If we observe these results by row, we can see that each adopted technique can help promote the classification performance. More specifically, CNN-D can first achieve about 30% improvement of OA compared with CNN, which means that DA strategy can greatly improve the classification results. Second, comparing CNN-DB with CNN-D, we can see that BN can achieve about 15% improvement. Next, we can observe that AL strategy can improve about 3% classification accuracy by comparing CNN-DBA with CNN-DB. Then, FT strategy can further improve the classification result about 1% compared with CNN-DBAF and CNN-DBA. Finally, in comparison with CNN-DBAF, the proposed CNN-AL-MRF method can obtain about 2% classification improvement, which means that MRF can boost the final performance. Overall, the results in Table III illustrate

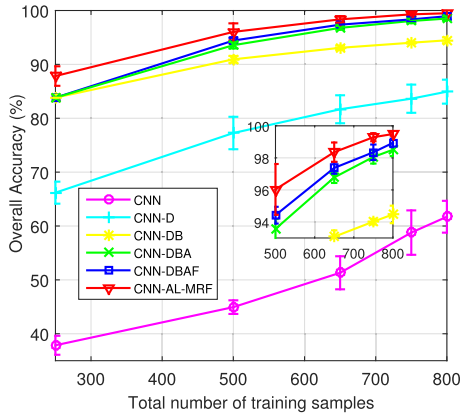


Fig. 4. OA (%) obtained by all competing methods with different number of training samples on Indian Pines data set.

that all the techniques used in the proposed method can help improve the classification accuracy.

Second, to evaluate the performance of the proposed method with training samples increasing, additional experiments are also conducted. Specifically, we conduct five rounds of CNN training. For non-AL methods, each round contains 800 epochs. For AL methods, each round contains 800, 400, 400, 300, and 200 epochs, respectively. Besides, we randomly select 250 training samples in the first round for all the methods. In the subsequent rounds, we then randomly add 250, 150, 100, and 50 samples into this training set for non-AL methods, and actively select 250, 150, 100, and 50 samples to enlarge the training set for AL methods. All the methods are repeated five times, and the average OA value and standard derivation are reported. The experimental results are demonstrated in Fig. 4. First, it can be easily seen from Fig. 4 that the classification accuracy of all the methods increases as the total number of training samples increases. Second, we can also observe that each technique can help improve the performance in all the cases.

D. Comparison With Other State-of-the-Art Methods

In order to further evaluate the performance of the proposed method, we compare the CNN-AL-MRF method with other

TABLE V

MEAN CLASSIFICATION ACCURACIES (%) OBTAINED BY ALL COMPETING METHODS ON INDIAN PINES DATA SET

Class/Method	SVM-GC [65]	SVM-LR [66]	SVM-3DGW [24])	SVM-3DWT [17]	SAE [45])	3D-CNN [36]	3D-FCN [37]	CNN-MRF [34])	CNN-AL-MRF	
Alfalfa	0(0)	7.20(21.62)	13.72(29.16)	48.47(29.30)	0(0)	30.55(12.08)	41.87(6.28)	18.51(9.86)	**92.71(10.31)**	84.17(13.72)
Corn-notill	77.50(6.43)	84.35(6.29)	80.97(7.75)	81.95(7.57)	74.91(9.91)	74.71(9.56)	80.71(3.42)	83.16(2.61)	**92.98(2.34)**	91.00(3.41)
Corn-mintill	53.73(16.24)	70.90(12.30)	68.55(13.10)	**91.58(5.56)**	52.25(13.81)	68.63(4.60)	78.32(5.65)	84.86(1.77)	88.71(2.89)	83.64(8.97)
Corn	49.15(4.55)	64.57(40.60)	67.37(19.27)	89.06(11.86)	47.64(43.75)	88.80(2.33)	72.10(2.62)	86.73(3.35)	**97.70(1.28)**	87.01(5.94)
Grass-pasture	81.17(9.01)	90.89(3.27)	89.60(4.82)	89.30(5.03)	79.25(12.92)	90.54(1.97)	68.66(13.61)	67.19(3.04)	**92.90(3.11)**	91.57(1.50)
Grass-trees	**99.56(0.12)**	99.27(0.60)	99.50(0.76)	98.26(1.42)	98.42(0.71)	97.62(0.81)	92.39(3.25)	90.56(5.56)	98.89(0.40)	95.18(1.90)
Grass-pasture-mowed	0(0)	0(0)	0(0)	0(0)	0(0)	84.02(3.01)	83.24(3.60)	79.39(11.47)	76.74(18.01)	**89.13(15.37)**
Hay-windrowed	**100(0)**	**100(0)**	99.73(0.05)	99.96(0.92)	99.95(0.13)	95.47(1.25)	96.87(0.14)	93.55(2.13)	97.87(1.52)	98.93(0.91)
Oats	0(0)	0(0)	0(0)	0(0)	0(0)	31.13(13.39)	37.12(7.89)	30.53(11.75)	**38.89(21.43)**	16.08(7.03)
Soybean-notill	52.78(7.25)	78.36(6.11)	71.35(9.83)	81.52(4.01)	76.07(2.52)	78.95(5.58)	79.23(3.82)	85.65(3.68)	**92.27(4.33)**	90.68(6.84)
Soybean-mintill	**97.80(2.28)**	95.21(2.88)	96.69(2.41)	96.13(2.86)	96.66(3.27)	81.62(3.64)	86.30(2.61)	87.36(3.34)	95.07(1.18)	94.70(0.63)
Soybean-clean	63.23(12.26)	91.45(9.00)	73.21(25.54)	83.92(8.69)	71.72(14.34)	68.81(7.49)	71.69(6.93)	62.21(6.13)	90.51(7.04)	**91.51(4.68)**
Wheat	99.38(0.38)	**99.63(0.46)**	98.40(1.82)	92.88(8.88)	99.58(0.50)	97.55(0.48)	98.06(0.28)	86.12(8.06)	96.53(2.01)	99.25(1.06)
Woods	98.93(3.95)	**99.32(0.82)**	99.19(0.60)	98.68(1.40)	97.13(3.74)	94.21(1.42)	86.15(5.73)	93.74(1.04)	99.28(0.34)	95.73(2.29)
Buildings-Grass-Trees-Drives	68.68(14.68)	**91.53(9.66)**	89.26(9.56)	91.01(9.33)	84.50(13.97)	86.08(8.71)	68.52(8.48)	71.47(9.88)	88.40(6.26)	83.99(6.89)
Stone-Steel-Towers	55.45(12.44)	36.70(45.13)	30.50(37.18)	37.38(39.40)	27.84(37.89)	98.20(0.22)	95.79(0.85)	83.97(4.47)	**97.12(4.07)**	92.08(3.66)
AA	62.34(3.34)	69.34(3.56)	67.35(2.30)	71.32(3.06)	62.87(3.05)	79.18(2.74)	77.31(3.42)	75.31(1.25)	**89.79(1.82)**	86.54(4.03)
OA	81.26(2.45)	88.61(2.05)	86.43(1.50)	90.22(1.54)	83.12(1.75)	85.05(1.81)	83.31(1.57)	86.05(1.12)	**94.28(0.31)**	92.26(0.87)
Used Label	520(5%)	520(5%)	520(5%)	520(5%)	520(5%)	520(5%)	520(5%)	520(5%)	520(5%)	416(4%)
Average Time (s)	8.17	7.82	8.15	101.98	515.46	6104.86	8413.59	7439.58	8109.34	5913.78

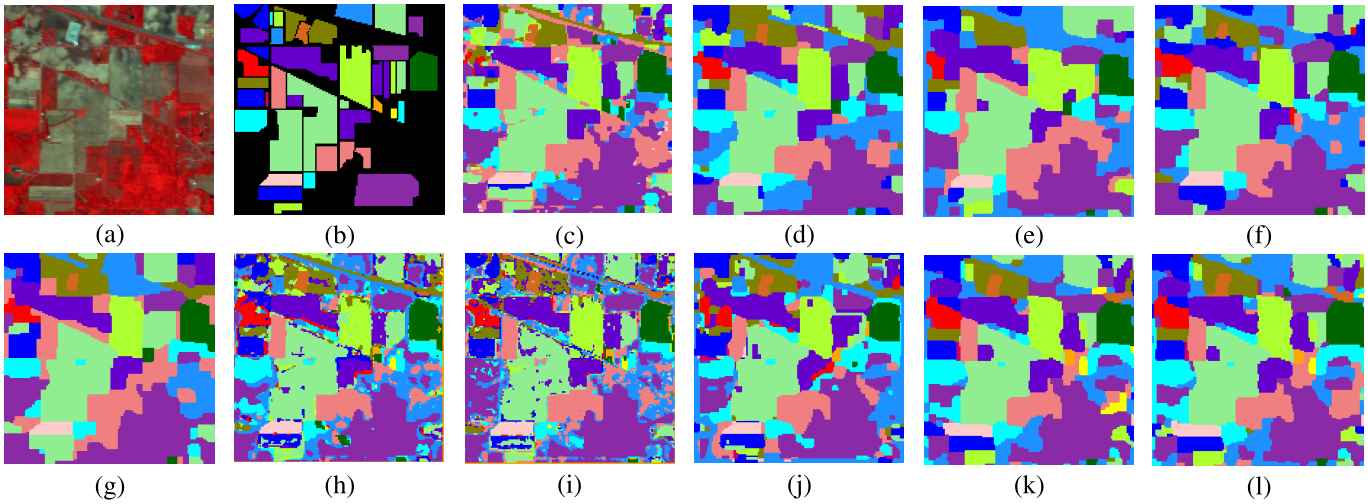


Fig. 5. Classification maps for the Indian Pines data set. (a) False-color image. (b) Ground-truth map. (c) SVM-GC. (d) SVM-LR. (e) SVM-3DGW. (f) SVM-3DWT. (g) SAE. (h) 3D-CNN. (i) 3D-FCN. (j) CNN-MRF. (k) CNN-AL-MRF (5% training samples). (l) CNN-AL-MRF (4% training samples).

seven state-of-the-art HSI classification methods on three benchmark data sets, namely Indian Pines, Pavia University, and Pavia Center.

1) Experiments on Indian Pines Data Set: For the Indian Pines data set, the experimental settings are shown as follows. For other competing methods, we randomly select 5% of the samples from each class as the training set, and thus there are 520 training samples in total. The remaining samples are used to test. Available training and testing sets are summarized in Table IV. For the proposed CNN-AL-MRF method, we conduct this experiment with two different settings. First, we consider the case where the total 520 samples are used. More specifically, we conduct four rounds of CNN training for this setting. The initialized number of training samples is 208, and then each of the subsequent three rounds actively select 104 samples. Second, we consider the case where training samples are fewer. Specifically, we use 416 samples in total (namely, 4% of the total samples). For this setting, there are three rounds of CNN training. We select 208 samples as the initialized training sample set, and then each of the subsequent two rounds actively select 104 samples. This experiment is repeated five times. The average experimental results are shown in Table V, and classification maps of all the methods are illustrated in Fig. 5.

TABLE VI

STATISTICS OF THE PAVIA UNIVERSITY DATA SET, INCLUDING THE NAME, THE NUMBER OF TRAINING, TEST, AND TOTAL SAMPLES FOR EACH CLASS

Class		Samples		
No	Name	Train	Test	Total
1	Shadows	10	937	947
2	Bricks	37	3645	3682
3	Bitumen	14	1316	1330
4	Bare Soil	50	4979	5029
5	Metal sheets	14	1331	1345
6	Trees	30	3034	3064
7	Gravel	20	2079	2099
8	Meadows	186	18463	18649
9	Asphalt	67	6564	6631
Total		428	42348	42776

2) Experiments on Pavia University Data Set: For the Pavia University data set, the experimental settings are shown as follows. For other competing methods, we randomly select

TABLE VII

MEAN CLASSIFICATION ACCURACIES (%) OBTAINED BY ALL COMPETING METHODS ON PAVIA UNIVERSITY DATA SET

Class/Method	SVM-GC [65]	SVM-LR [66]	SVM-3DGW [24])	SVM-3DWT [17]	SAE [45])	3D-CNN [36]	3D-FCN [37]	CNN-MRF [34])	CNN-AL-MRF	
Shadows	97.85(1.48)	98.82(0.33)	97.46(1.67)	98.12(0.92)	96.42(1.92)	93.71(1.16)	92.60(2.02)	93.89(2.86)	**98.90(1.31)**	98.18(0.76)
Bricks	99.25(0.95)	99.71(0.43)	99.47(0.27)	98.80(0.74)	99.45(0.50)	83.69(2.09)	80.22(2.33)	95.58(0.79)	**99.91(0.08)**	99.82(0.10)
Bitumen	24.01(17.59)	70.89(5.71)	63.50(6.98)	57.14(5.59)	34.22(16.24)	84.28(3.47)	57.61(6.41)	61.22(12.82)	**85.56(6.79)**	78.46(11.77)
Bare Soil	78.26(8.51)	89.94(3.37)	81.33(6.81)	**96.80(0.98)**	89.43(2.18)	93.40(1.64)	93.11(1.45)	71.63(11.30)	96.49(1.60)	93.86(4.01)
Metal sheets	98.54(0.43)	99.09(0.80)	97.85(1.31)	**99.90(0.11)**	98.03(0.95)	99.85(0.11)	99.89(0.07)	96.42(4.09)	**99.90(0.06)**	99.05(1.57)
Trees	64.74(12.67)	68.10(6.75)	97.62(2.17)	88.04(4.52)	62.97(5.17)	81.35(4.02)	74.75(5.23)	77.93(7.95)	97.25(3.77)	**98.46(1.35)**
Gravel	28.19(30.04)	91.26(9.44)	19.42(20.50)	68.94(11.42)	12.50(18.40)	82.16(4.34)	71.83(3.39)	53.66(6.09)	**96.04(2.23)**	94.77(2.47)
Meadows	91.28(2.44)	90.98(3.85)	88.96(1.96)	82.05(5.65)	91.39(4.23)	93.45(1.98)	90.23(2.87)	74.76(8.94)	**98.70(1.42)**	97.71(0.50)
Asphalt	**96.54(3.33)**	81.50(8.84)	86.35(10.69)	88.13(6.20)	94.29(7.40)	94.61(0.88)	94.34(0.14)	86.07(2.04)	94.87(2.05)	92.91(5.16)
AA	75.51(5.28)	87.82(1.48)	81.33(2.39)	86.44(1.80)	75.42(2.65)	89.65(1.13)	83.83(1.01)	79.24(1.39)	**96.40(0.28)**	94.80(1.92)
OA	86.80(2.67)	92.31(0.66)	92.14(0.63)	96.27(1.21)	87.22(0.99)	91.27(0.71)	86.42(1.49)	87.60(1.58)	**98.17(0.19)**	97.43(0.87)
Used Label	428(1%)	428(1%)	428(1%)	428(1%)	428(1%)	428(1%)	428(1%)	428(1%)	428(1%)	321(0.75%)
Average Time (s)	24.03	23.61	23.15	196.43	674.25	1867.46	2440.75	1349.76	2001.12	1545.34

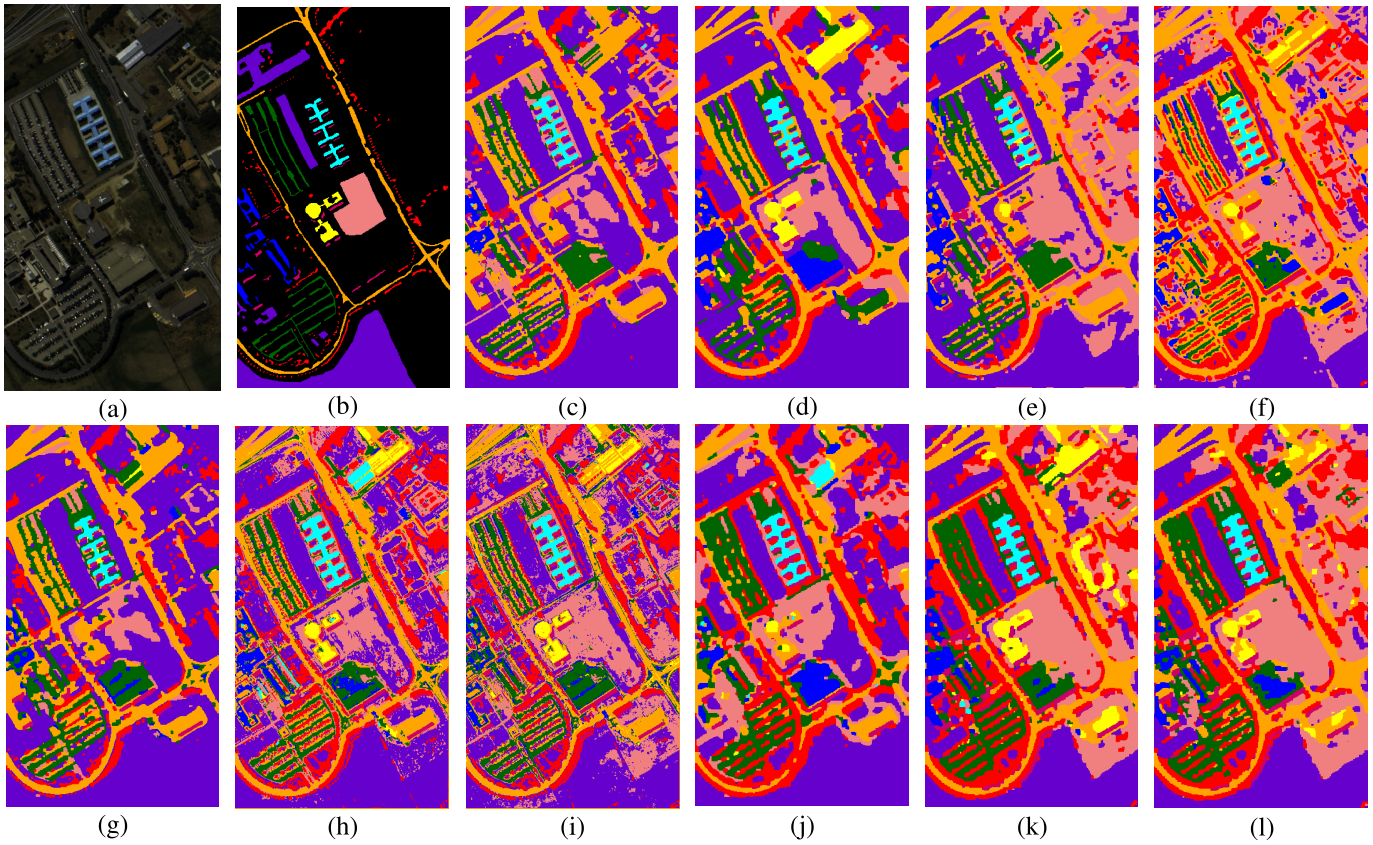


Fig. 6. Classification maps for the Pavia University data set. (a) False-color image. (b) Ground-truth map. (c) SVM-GC. (d) SVM-LR. (e) SVM-3DGW. (f) SVM-3DWT. (g) SAE. (h) 3D-CNN. (i) 3D-FCN. (j) CNN-MRF. (k) CNN-AL-MRF (1% training samples). (l) CNN-AL-MRF (0.75% training samples).

TABLE VIII

CLASSIFICATION ACCURACY (%) AND STANDARD DEVIATION (IN BRACKET) OBTAINED BY ALL COMPETING METHODS ON THE TEST SET OF THE PAVIA CENTER DATA SET

Class/Method	SVM-GC [65]	SVM-LR [66]	SVM-3DGW [24])	SVM-3DWT [17]	SAE [45])	3D-CNN [36]	3D-FCN [37]	CNN-MRF [34])	CNN-AL-MRF	
Shadows	0(0)	22.26(30.84)	13.75(27.51)	0(0)	29.04(28.31)	87.97(6.11)	46.66(10.25)	**85.39(19.72)**	82.88(7.92)	71.73(5.42)
Bitumen	**100(0)**	98.79(1.58)	99.46(1.07)	**100(0)**	**100(0)**	98.08(2.01)	97.45(1.84)	97.70(3.25)	**100(0)**	**100(0)**
Asphalt	**99.99(0.01)**	98.81(1.04)	98.80(1.43)	99.44(0.53)	98.77(0.81)	98.94(0.89)	98.69(0.39)	90.79(12.10)	98.32(0.17)	98.24(0.19)
Bare Soil	99.25(1.49)	99.93(0.11)	**99.77(0.30)**	98.71(1.66)	99.88(0.14)	97.61(1.37)	95.57(1.17)	85.88(18.89)	99.76(0.14)	98.83(1.66)
Tiles	95.93(7.99)	94.30(6.08)	99.85(0.23)	96.24(6.03)	99.91(0.16)	94.91(2.56)	94.80(2.21)	93.93(8.59)	**99.85(0.21)**	98.36(2.32)
Gravel	0(0)	46.19(9.82)	0(0)	0(0)	0(0)	0(0)	5.74(13.97)	0(0)	**100(0)**	**100(0)**
Trees	97.42(2.07)	96.63(2.24)	96.79(2.10)	95.94(2.60)	96.91(3.27)	98.43(1.08)	95.62(1.51)	97.73(3.21)	98.83(1.37)	99.30(0.71)
Water	**100(0)**	**100(0)**	**100(0)**	**100(0)**	**100(0)**	98.91(1.25)	99.08(0.43)	98.92(1.53)	**100.00(0)**	**100(0)**
AA	74.08(1.05)	82.12(6.63)	76.86(3.25)	73.79(0.49)	78.07(3.81)	84.36(3.12)	79.19(2.62)	81.29(0.53)	**97.45(0.81)**	95.81(0.66)
OA	98.04(0.74)	97.79(0.54)	98.11(0.37)	97.73(0.34)	98.35(0.44)	97.78(0.37)	97.11(0.41)	97.06(0.34)	**99.15(0.07)**	98.81(0.13)
Used Label	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	104(0.5%)	84(0.4%)
Average Time (s)	5.79	5.04	5.58	40.61	480.57	1231.06	1497.25	1043.76	1378.57	859.46

1% of the samples from each class, and thus there are 428 samples in total. The available training and testing sets are summarized in Table VI. For the proposed CNN-AL-MRF method, we conduct experiments with two different settings. First, we consider the case where 428 samples are used in total. More specifically, there are four rounds of CNN training for

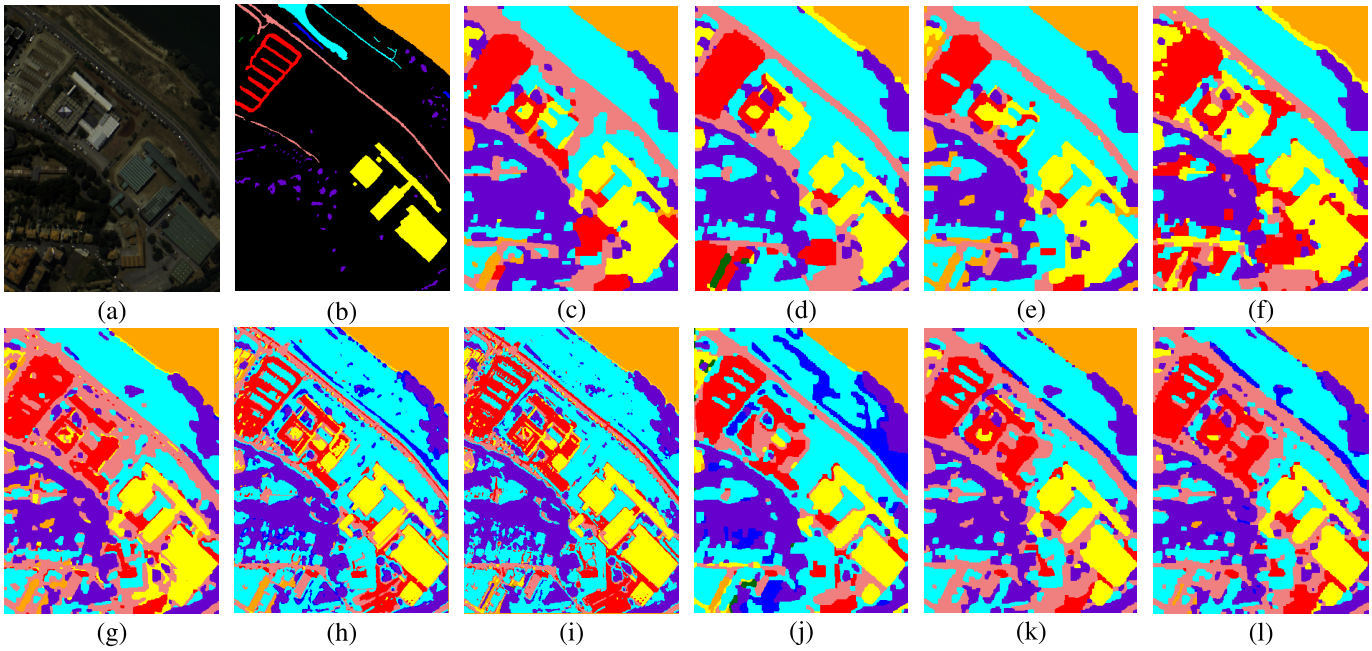


Fig. 7. Classification maps for the Pavia Center data set. (a) False-color image. (b) Ground-truth map. (c) SVM-GC. (d) SVM-LR. (e) SVM-3DGW. (f) SVM-3DWT. (g) SAE. (h) 3D-CNN. (i) 3D-FCN. (j) CNN-MRF. (k) CNN-AL-MRF (0.5% training samples). (l) CNN-AL-MRF (0.4% training samples).

this setting. We select 107 samples as the initialized training set, and actively select 107 samples in each of the subsequent three rounds. Second, we consider the case where training samples are fewer. More specifically, 321 samples are used in total (0.75% of the total samples). For this setting, we conduct three rounds of CNN training. We use 107 samples as training set in the first round and actively select 107 samples in each of the subsequent two rounds. This experiment is repeated five times. Experimental results are shown in Table VII, and classification maps of all the methods are illustrated in Fig. 6.

From Table VII, we can easily observe that the proposed CNN-AL-MRF method achieves the best performance in terms of AA and OA measures compared with all the other methods. Also, the proposed method has the best classification accuracy in most of the classes by comparing the CAs of all the methods.

Additionally, the proposed method even outperforms all the other methods in the case where the number of used labels is less (only 0.75% of the total samples) compared with other methods (1% of the total samples), which also implies that the proposed CNN-AL-MRF method can help reduce the labeling cost while still obtains the best performance. Moreover, it can be seen from Fig. 6 that the proposed method obtains more accurate and smooth classification maps compared with other competing methods.

3) Experiments on Pavia Center Data Set: For the Pavia Center data set, the experimental settings are shown as follows. For other competing methods, we randomly select 0.5% of the samples from each class, and thus there are 104 samples in total. For the proposed CNN-AL-MRF method, we conduct experiments with two different settings. First, we consider the case where 104 samples are used in total. Specifically, there are three rounds of CNN training for this setting. We select 52 samples as the initialized training set, and actively select 32 and 22 samples in each of the subsequent two rounds. Second, we consider the case where training samples are fewer. Specifically, we only use 84 samples in total (0.4% of the total samples). For this setting, we conduct two rounds of CNN training. We use 52 samples as a training set in the first round and actively select 32 samples in the second round. This experiment is repeated five times. The average experimental results are shown in Table VIII, the available training and testing sets are summarized in Table IX, and classification maps of all the methods are illustrated in Fig. 7.

TABLE IX
STATISTICS OF THE PAVIA CENTER DATA SET, INCLUDING THE NAME, THE NUMBER OF TRAINING, TEST AND TOTAL SAMPLES FOR EACH CLASS

Class		Samples		
No	Name	Train	Test	Total
1	Shadows	1	42	43
2	Bitumen	32	6283	6315
3	Asphalt	9	1677	1686
4	Bare Soil	11	2088	2099
5	Tiles	9	1671	1680
6	Gravel	2	221	223
7	Trees	8	1499	1507
8	Water	32	6282	6314
Total		104	19763	19867

From Table VIII, we can obtain the following observations. First, almost all the competing methods fail to classify the Meadows class since only 1 sample is selected for training, and thus the classifier is not trained well to distinguish this class while the proposed CNN-AL-MRF method is successful of classifying this class. Second, the proposed CNN-AL-MRF method achieves the best performance in terms of AA and OA measures compared with other methods. Additionally, the proposed method even outperforms all the other methods in the case where the number of used labels is less (only 0.4%

of the total samples) compared with other methods (0.5% of the total samples), which further indicates that the proposed CNN-AL-MRF method can help reduce the labeling cost while still performs best. Finally, it can be easily seen from Fig. 7 that the proposed method obtains more accurate and smooth classification maps compared with other competing methods.

V. Conclusion and Future Work

In this article, a novel approach combining AL with deep learning is presented for HSI classification. More specifically, the proposed method has the following advantages. First, this method inherits the benefits of both worlds, namely strong discriminative ability of deep learning and labeling efficiency of AL. Second, by actively selecting the most useful samples as well as using DA strategy, this method can significantly reduce the labeling cost. Finally, the spatial local correlation of labels is further exploited to improve the classification performance by using the MRF. Experiments on three benchmark data sets demonstrate that the proposed method can outperform other state-of-the-art traditional and deep learning-based HSI classification methods with less labeled training samples. The future work of this research mainly focuses on extending the supervised framework to a semi-supervised and unsupervised framework. Also, we attempt to design more useful AL measures to select informative samples.

References

[1] A. Plaza *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. S110–S122, Sep. 2009.

[2] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[3] J. Yao, D. Meng, Q. Zhao, W. Cao, and Z. Xu, "Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2991–3006, Jun. 2019.

[4] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[5] G. P. Petropoulos, C. Kalaitzidis, and K. Prasad Vadrevu, "Support vector machines and object-based classification for obtaining land-use/cover cartography from Hyperion hyperspectral imagery," *Comput. Geosci.*, vol. 41, pp. 99–107, Apr. 2012.

[6] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 35–49, Dec. 2019.

[7] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193–205, Jan. 2019.

[8] T. Matsuki, N. Yokoya, and A. Iwasaki, "Hyperspectral tree species classification of Japanese complex mixed forest with the aid of LiDAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2177–2187, May 2015.

[9] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "CoSpace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.

[10] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.

[11] T. 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.

[12] A. Soltani-Farani, H. R. Rabiee, and S. A. Hosseini, "Spatial-aware dictionary learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 527–541, Jan. 2015.

[13] X. Sun, Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Structured priors for sparse-representation-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1235–1239, Jul. 2014.

[14] X. Cao, Z. Xu, and D. Meng, "Spectral–spatial hyperspectral image classification via robust low-rank feature extraction and Markov random field," *Remote Sens.*, vol. 11, no. 13, p. 1565, Jul. 2019.

[15] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Aug. 2012.

[16] Y. Tarabalka and A. Rana, "Graph-cut-based model for spectral–spatial classification of hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2014, pp. 3418–3421.

[17] X. Cao, L. Xu, D. Meng, Q. Zhao, and Z. Xu, "Integration of 3-dimensional discrete wavelet transform and Markov random field for hyperspectral image classification," *Neurocomputing*, vol. 226, pp. 90–100, Feb. 2017.

[18] H. Bi, J. Sun, and Z. Xu, "A graph-based semisupervised deep learning model for PolSAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2116–2132, Apr. 2019.

[19] H. Bi, F. Xu, Z. Wei, Y. Xue, and Z. Xu, "An active deep learning approach for minimally supervised PolSAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9378–9395, Nov. 2019.

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

[21] D. Hong and X. X. Zhu, "SULoRA: Subspace unmixing with low-rank attribute embedding for hyperspectral data analysis," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1351–1363, Dec. 2018.

[22] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 911–923, Feb. 2019.

[23] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[24] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, Dec. 2011.

[25] H. Bi, L. Xu, X. Cao, and Z. Xu, "Polsar image classification based on three-dimensional wavelet texture features and Markov random field," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3921–3928.

[26] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," 2019, *arXiv:1912.08847*. [Online]. Available: https://arxiv.org/abs/1912.08847

[27] D. Hong, N. Yokoya, and X. X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2960–2975, Jun. 2017.

[28] Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, Jun. 2016.

[29] 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.

[30] 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.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 346–361.

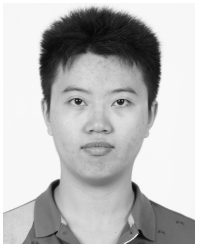
[32] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.

[33] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sens. Lett.*, vol. 7, no. 9, pp. 875–884, Sep. 2016.

[34] 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.

[35] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.

[36] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.
[37] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
[38] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2372–2379.
[39] J. Li, "Active learning for hyperspectral image classification with a stacked autoencoders based neural network," in *Proc. 7th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2015, pp. 1–4.
[40] 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.
[41] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.
[42] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*. [Online]. Available: https://arxiv.org/abs/1712.04621
[43] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7340–7351.
[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[45] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral–spatial classification of hyperspectral image using autoencoders," in *Proc. IEEE 9th Int. Conf. Inf., Commun. Signal Process. (ICICS)*, Dec. 2013, pp. 1–5.
[46] M. Midhun, S. R. Nair, V. Prabhakar, and S. S. Kumar, "Deep model for classification of hyperspectral image using restricted Boltzmann machine," in *Proc. ACM Int. Conf. Interdiscipl. Adv. Appl. Comput.*, 2014, Art. no. 35.
[47] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, Jun. 2015.
[48] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
[49] S. Paisitkriangkrai *et al.*, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 36–43.
[50] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, May 2017.
[51] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.
[52] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.
[53] S. Sun, P. Zhong, H. Xiao, and R. Wang, "An MRF model-based active learning framework for the spectral–spatial classification of hyperspectral imagery," *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 6, pp. 1074–1088, Sep. 2015.
[54] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
[55] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[56] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.
[57] J. Zhu, H. Wang, T. Yao, and B. K. Tsou, "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification," in *Proc. 22nd Int. Conf. Comput. Linguistics*, vol. 1, 2008, pp. 1137–1144.
[58] W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 51–60.
[59] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2009.
[60] N. Roy and A. McCallum, "Toward optimal active learning through Monte Carlo estimation of error reduction," in *Proc. ICML*, Williamstown, VIC, Australia, 2001, pp. 441–448.
[61] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[62] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
[63] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.
[64] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.
[65] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.
[66] Y. Xu, Z. Wu, and Z. Wei, "Spectral–spatial classification of hyperspectral image based on low-rank decomposition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2370–2380, Jun. 2015.



Xiangyong Cao (Member, IEEE) received the B.Sc. and Ph.D. degrees from the Xi'an Jiaotong University, Xi'an, China, in 2012 and 2018, respectively.

From 2016 to 2017, he was a Visiting Scholar with Columbia University, New York, NY, USA. He is currently an Assistant Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include low-rank modeling, statistical modeling, and hyperspectral image analysis.

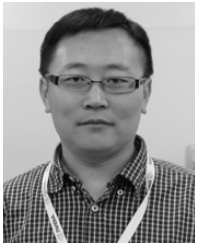
Jing Yao received the B.Sc. degree from Northwest University, Xi'an, China, in 2014. He is currently pursuing the Ph.D. degree with Xi'an Jiaotong University, Xi'an.

His research interests include low-rank matrix factorization and hyperspectral image analysis.

Zongben Xu received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1987.

He is currently the Director of the Institute for Information and System Sciences, Xi'an Jiaotong University. His research interests include intelligent information processing and applied mathematics.

Dr. Xu was elected as a member of the Chinese Academy of Science in 2011. He was a recipient of the National Natural Science Award of China in 2007 and the Winner of the CSIAM Su Buchin Applied Mathematics Prize in 2008. He delivered a speech at the International Congress of Mathematicians 2010. He serves as the Chief Scientist for the National Basic Research Program of China (973 Project).



Deyu Meng (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively.

From 2012 to 2014, he took his two-year sabbatical leave in Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University, and an Adjunct Professor with the Faculty of Information Technology, Macau University of Science and Technology, Taipa, Macau. His research interests include self-paced learning, noise modeling, and tensor sparsity.