Hyperspectral Image Classification With Deep Learning Models

Xiaofei Yang, Yunming Ye, Xutao Li, Raymond Y. K. Lau, *Senior Member, IEEE*, Xiaofeng Zhang, and Xiaohui Huang

***Abstract*— Deep learning has achieved great successes in conventional computer vision tasks. In this paper, we exploit deep learning techniques to address the hyperspectral image classification problem. In contrast to conventional computer vision tasks that only examine the spatial context, our proposed method can exploit both spatial context and spectral correlation to enhance hyperspectral image classification. In particular, we advocate four new deep learning models, namely, 2-D convolutional neural network (2-D-CNN), 3-D-CNN, recurrent 2-D CNN (R-2-D-CNN), and recurrent 3-D-CNN (R-3-D-CNN) for hyperspectral image classification. We conducted rigorous experiments based on six publicly available data sets. Through a comparative evaluation with other state-of-the-art methods, our experimental results confirm the superiority of the proposed deep learning models, especially the R-3-D-CNN and the R-2-D-CNN deep learning models.**

***Index Terms*— Convolutional neural network (CNN), deep learning, hyperspectral image.**

I. Introduction

RECENTLY, the rapid development of optics and photonics has significantly advanced the field of hyperspectral imaging techniques. As a result, hyperspectral sensors are installed in many satellites which can produce images with rich spectral information. The rich information captured in hyperspectral images enables us to distinguish very similar materials and objects using satellites. Accordingly, hyperspectral imaging techniques have been widely used in a variety of fields, such as agriculture, monitoring, astronomy, and mineral exploration. For example, Bioucas-Dias *et al.* [1] analyzed the compact reconnaissance imaging spectrometer for Mars hyperspectral data set and used linear mixing of absorption band techniques to determine the mineralogy of the surface on Mars. Brown *et al.* [2] utilized the visible and near infrared (VNIR) imaging spectrometer instrument that was a hyperspectral scanning pushbroom device sensitive to VNIR wavelengths from 400 to 1000 nm for mineral exploration.

The existing methods for hyperspectral image classification are mostly based on conventional pattern recognition approaches, such as support vector machine (SVM) [3] and K-nearest neighbor classifiers. To address the curse of dimensionality, namely the Hughes phenomenon [4], Krishnapuram *et al.* [5] performed dimensionality reduction against a data set first and then applied multinomial logistic regression to improve image classification performance. Wang *et al.* [6] proposed a novel dimensionality reduction method, namely, the locality adaptive discriminant analysis (LADA) method for hyperspectral image analysis. Another way to cope with the Hughes phenomenon is via the salient band selection method. For example, Wang *et al.* [7] proposed a manifold ranking-based salient band selection method. In addition, Yuan *et al.* [8] proposed a new dual clustering framework, which was applied to tackle the inherent drawbacks of the clustering-based band selection. It has been shown that a composite kernel (CK) approach that requires multiple kernels can enhance the accuracy of classification by fusing the spatial and the spectral information. For example, the generalized CK (GCK) framework is one of the promising methods for hyperspectral image classification. Though kernel-based methods such as GCK can exploit both the spectral and the spatial information, it involves solving a computationally very costly optimization problem.

As a state-of-the-art machine learning technique, deep learning [9], [10] has recently attracted a lot of attention for its application to conventional computer vision tasks. One main reason is that deep learning can automatically discover an effective feature representation for a problem domain, thus avoiding the complicated and hand-crafted feature engineering process. With a specially designed deep learning architecture, convolutional neural networks (CNNs) are widely applied to image recognition and image segmentation which considers the spatial correlation among pixels. Successful examples of CNNs include AlexNet [11], very deep convolutional networks proposed by Visual Geometry Group [12], GoogLeNet [13], and ResNet [14]. However, existing CNNs are applied to conventional image classification tasks rather than hyperspectral

Manuscript received October 21, 2017; revised January 31, 2018; accepted March 4, 2018. Date of publication April 17, 2018; date of current version August 27, 2018. This work was supported by the Shenzhen Science and Technology Program under Grant JCYJ20160330163900579, Grant JCYJ20170413105929681, and Grant JCYJ20170811160212033. R. Y. K. Lau was supported in part by the RGC of the Hong Kong SAR under Project CityU 11502115 and Project CityU 11525716, in part by the National Natural Science Foundation of China (NSFC) Basic Research Program under Project 71671155, in part by the Shenzhen Municipal Science and Technology Innovation Fund under Project JCYJ20160229165300897, and in part by the CityU Shenzhen Research Institute. X. Huang was supported in part by NSFC under Grant 61562027 and in part by the Education Department of Jiangxi Province under Grant GJJ170413. *(Corresponding authors: Yunming Ye; Xutao Li.)*

X. Yang, Y. Ye, X. Li, and X. Zhang are with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, 518055, China, and also with the Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China (e-mail: yeyunming@hit.edu.cn; lixutao@hit.edu.cn).

R. Y. K. Lau is with the City University of Hong Kong, Hong Kong.

X. Huang is with the School of Information Engineering, East China Jiaotong University, Nanchang 330013, China.

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

Digital Object Identifier 10.1109/TGRS.2018.2815613

image classification tasks where both the spatial and spectral correlations need to be effectively exploited.

In this paper, we address the hyperspectral image classification problem using a new deep learning technique. As noted above, both the spectral factor and the spatial factor influence the class label prediction of a pixel. On one hand, the label of a pixel is reflected by its spectral values scanned by using different spectra. On the other hand, as the geographically close pixels tend to belong to the same class, predicting the class label of a pixel should take into account the class labels of the surrounding pixels. Hence, a good hyperspectral image classification method should consider both the spectral factor and the spatial factor. In this paper, we first advocate a 2-D-CNN model and a 3-D-CNN model for classifying hyperspectral images. The intuition is that a 2-D-CNN can exploit the spatial context, whereas a 3-D-CNN can exploit both the spatial and the spectral context. Though the aforementioned models can take into account rich contextual information, the way that they process the spatial information may introduce unwanted noise. Accordingly, we further design the recurrent 2-D-CNN (R-2-D-CNN) and the recurrent 3-D-CNN (R-3-D-CNN) to address the noisy spatial information problem. The main contributions of this paper are summarized as follows.

1) We treat spectral data as the channels of conventional images. To classify each pixel in a hyperspectral image, we extract a small patch centered at the pixel. The patch is treated as an image with multiple channels. Then, we design a 2-D-CNN model with three 2-D convolution layers, followed by a full connection layer, to classify the patch. The label of the patch is considered as the label of its central pixel. Though the pooling layers (such as max-pooling layers and average pooling layers) could reduce the dimensions of feature maps and simplify the computations, they may affect the classification accuracy of the network. To preserve as much contextual information as possible, pooling layers are excluded from our 2-D-CNN model. The convolution layer, pooling layer, and fullyconnection layer of a CNN will be explained in Section II-B.
2) Though the 2-D-CNN model can utilize the spatial context, it fails to consider the spectral correlations. To address such a problem, we further design a 3-D-CNN model which is composed of seven convolution layers and one full connection layer. Different from the 2-D-CNN, the convolution operator of this model is 3-D, whereas the first two dimensions are applied to capture the spatial context and the third dimension captures the spectral context. Though the 3-D-CNN model contains more network parameters than its 2-D counterpart, it should be more effective than its 2-D counterpart because of its ability to evaluate the spectral correlations of a hyperspectral image.
3) The 2-D-CNN model may be noisy, because the classification of a pixel only relies on a small patch centered at the pixel. To effectively utilize the spatial context, we further design an R-2-D-CNN model. The R-2-D-CNN can extract features by gradually shrinking the patch to concentrate on the central pixel. Experimental results show that the R-2-D-CNN model indeed performs better than the 2-D-CNN model.
4) Finally, we design the R-3-D-CNN model to take into account both spatial and spectral contexts while alleviating the problem of a noisy patch. The R-3-D-CNN extends the 3-D-CNN model by shrinking the patch gradually. As a result, the final classification of each pixel mainly depends on the information of the pixel rather than a patch. Experimental results show the superiority of the R-3-D-CNN model. In particular, it converges faster than other methods, and achieves the best classification performance.

The rest of this paper is organized as follows. Section II discusses the related research work. In Section III, we illustrate various CNN-based deep learning models and the corresponding algorithms for hyperspectral image classification. Section VI reports the experimental results of a comparative evaluation of the experimental methods and other baseline methods. Finally, we give conclusion and highlight the directions of the future research work.

II. Related Work

A. Classical Classification Methods

Hyperspectral remote sensing classification has been extensively studied recently. For example, Bandos *et al.* [15] utilize a linear discriminant method to solve the problem. However, when the spectral resolution is low, it is necessary to handle the band mixing problem for better differentiating the pixels or performing feature selection. To this end, Brown [16] develop a robust method to automatically separate overlapping absorption bands, and the advantage of such a method is that it is relatively noise-insensitive. To address the nonlinearity of data, a quadratic discriminant analysis and a logarithmic discriminant analysis are also explored. However, these methods suffer from the Hughes phenomenon, i.e., the classification performance considerably degrades when the dimensionality of the problem space becomes high. Wang *et al.* [6] propose a novel dimensionality reduction method, namely, LADA for hyperspectral image classification. Following the idea of linear discriminant analysis, the LADA learns a projection matrix $\mathbf{G}$ to pull the points of the same class close to each other while pushing one of different classes far away from each other. To further exploit the local data manifold, the LADA adds one adaptive manifold term parameterized by a matrix S into the computation of within-class scatter term and solves the matrix G and S alternatively. In 2016, Wang *et al.* [7] propose a manifold ranking-based salient selection method for hyperspectral image classification. The method first employs an evolution algorithm to group the bands into several subsets and finds some representative bands. Then, it uses the representatives to select salient bands by a manifold ranking strategy. The performance of the method significantly relies on the qualities of chosen representatives and the constructed manifold.

To improve a classification performance, many researchers resort to kernel-based methods. The main idea of kernel-based methods is to project samples into a high-dimensional space in which the samples of different classes become linearly

separable. The trick of kernel-based methods is that one does not need to specify the details of the transformation function. Instead, we only need to define the linear products among samples in the high-dimensional space. For example, Camps-Valls and Bruzzone [17] employ the kernel trick of an SVM in that the separation of classes in a high-dimensional space was achieved via a nonlinear transformation of an SVM.

Apart from employing simple kernel tricks, some researchers employed multiple kernels for hyperspectral image classification. For example, Rakotomamonjy *et al.* [18] advocate the multiple kernel learning (MKL) method which could learn a kernel and a classification predictor at the same time. With the preliminary success of MKL, the same technique is applied to remote sensing in 2010 [19]. In 2012, a representative MKL algorithm is developed which could establish the weights of kernels according to their statistical significance [20].

The aforementioned kernel-based methods do not explicitly exploit a spatial context. To address such a problem, the CK method is proposed [21]. In [22], the CK method is generalized using extended multiattribute profiles (EMAPs). Apart from considering the spatial context, the CK method could exploit the spectral context as well. For example, a GCK is developed to exploit both EMAPs and raw features [23]. The GCK method often achieves a better performance than the conventional methods such as SVM with composite kernels (SVM-CK) [23].

Despite achieving promising classification performance, all the kernel-based methods suffer from two drawbacks: 1) they often involve solving complicated convex problems which are in general difficult for a classifier to learn and 2) kernels must be carefully chosen so as to achieve a good performance.

Recently, some more advanced classification methods are developed for hyperspectral image classification [24], [25]. For example, the logistic regression via variable splitting and augmented Lagrangian (LORSAL) algorithm [26] is developed to tackle larger data sets efficiently. Sun *et al.* [27] propose a hyperspectral image classification model, named sparse multinomial logistic regression and spatially adaptive total variation (SMLR-SpATV), which includes a spectral data fidelity term and a spatially adaptive Markov random field (MRF) prior in the hidden field. Li *et al.* [28] propose a new multiple feature learning (MFL) framework, which pursues the combination of multiple features for the hyperspectral scenes categorization. The method can handle both the linear and the nonlinear classification. In [29], a novel SVM-based classification method is proposed by applying the 3-D discrete wavelet transform (SVM-3-DG).

B. Deep Learning Models

Recently, CNN models have achieved a breakthrough in the performance of image classification. The CNN model (see Fig. 1) is a multilayer neural network, composed of convolution layer, pooling layer, and full connection layer. The convolution layer contains N filters (C1 in Fig. 1), each of which is a small weighted matrix. By convolving the N filters with an input image and transforming the output with a nonlinear activation function, N feature maps are produced.

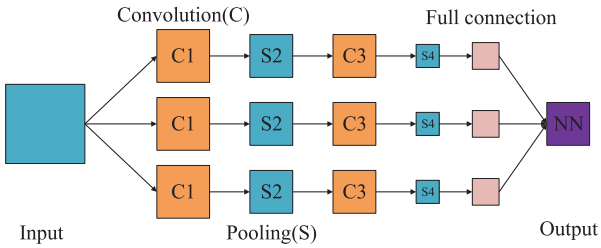


Fig. 1. CNN model consisting of convolution layers, pooling layers, and full connection layer.

The feature maps often contain redundant information. To reduce the redundancy, a pooling layer is appended (S2 in Fig. 1), which summarizes feature maps into small matrices by calculating the average (average pooling) or maximum value (maximum pooling) locally. The convolution layer and pooling layer can be repeated multiple times (C3 and S4 in Fig. 1) until the generated feature maps are of size 1×1. Finally, a fully connected layer will be appended for categorization. The neurons of the fully connected layer take all the 1×1 feature maps as their inputs.

The first CNN model is developed by LeCun *et al.* [30], [31] in 1996. Combined with the backpropagation (BP) model, the CNN model achieves a very good performance in handwritten digit recognition. With the advancement of graphics processing units (GPUs), deep learning has attracted a lot of attention by researchers. On the other hand, the CNN model has been improved by the recent deep learning techniques. For example, Glorot *et al.* [32] introduce the rectified linear units (ReLUs) as the activation function for CNNs in 2011. By doing so, the vanishing gradient problem and the ineffective exploration problem of the BP method can be alleviated. In 2012, Krizhevsky *et al.* [11] designed the AlexNet network which was a deep CNN model with the ReLU activation function. The AlexNet network won the annual ImageNet competition in 2012. To avoid overfitting, Srivastava *et al.* [33] proposed the dropout technique for a deep CNN. In addition, Szegedy *et al.* [13] designed the GoogLeNet model which is a deep CNN model with each layer comprising multiscale CNN. He *et al.* [14] proposed a deep residual CNN model which won the ImageNet competition in 2015. In [34], an end-to-end band-adaptive spectral–spatial feature learning neural network was proposed. Cao *et al.* [35] proposed a hyperspectral image segmentation method by using Markov random fields and a convolutional network. To tackle the street scene labeling problem, Wang *et al.* [36] proposed a hybrid method that utilized priori CNNs at a superpixel level and soft restricted context transfer. The former technique aims to learn prior location information and produces coarse label predication, whereas the latter technique aims to improve the coarse prediction by reducing oversmoothness. However, the algorithm works for conventional images only. It does not take into account the characteristics of rich band information in hyperspectral images.

All the above-mentioned models are 2-D-CNN models within which the convolution operators only deal with 2-D spatial features. In [37], a 3-D convolution network is designed to handle video categorization tasks effectively. Following

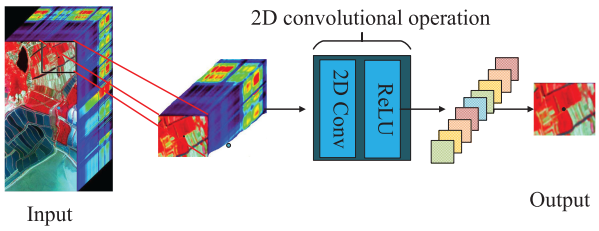


Fig. 2. 2-D-CNN model consisting 2-D convolutional operation with kernel size (k) and number of feature maps (m) at each convolutional layer for hyperspectral image classification.

the framework of 3-D-CNN models, we employ such an architecture for hyperspectral image classification, in which the third dimension refers to the spectral axis.

Apart from CNN models, another important deep learning framework is the recurrent neural network (RNN) which is often applied to process sequence data arising from applications, such as speech recognition [38], machine translation [39], bot chat [40], and so on. Mou *et al.* [41] proposed a novel RNN model for hyperspectral image classification, which could effectively analyzed hyperspectral pixels as sequential data and then determined information categories via network reasoning. The basic intuition of the RNN is that it applies the same neural network block recurrently for sequence prediction. To preserve the information of observed historical sequences, an RNN is fed with the current observation and the hidden layers trained by the previously observed sequences. By doing so, the RNN can take into account both the features of the current sequence and that of the historical observations to improve the current prediction. In contrast to the aforementioned approaches, we apply the RNN model to deal with the spatial contexts recurrently.

III. Proposed Methods

In this section, we illustrate the design of new 2-D-CNN, R-2-D-CNN, 3-D-CNN, and R-3-D-CNN models for hyperspectral image classification. For these methods, we extract a small patch centered at each pixel to build the classification models. Among the proposed models, the 2-D-CNN and R-2-D-CNN models exploit the spatial contexts only, whereas the 3-D-CNN and R-3-D CNN models exploit both the spatial features and the spectral correlations of pixels.

A. 2-D-CNN Model

As illustrated in Fig. 2, our 2-D-CNN model is composed of three main phases: patch extraction, feature extraction, and label identification. Given a hyperspectral image, we first extract a small patch centered at each pixel as the raw feature. Then, a deep learning model is constructed to acquire the feature maps of these patches. Finally, the label of each pixel is classified based on the feature map of the corresponding patch. For all four models, we exclude the pooling layers so as to preserve as much information of a pixel as possible. The three-phase processing of the 2-D-CNN model is illustrated as follows.

Assume that we are given a hyperspectral image of size $N \times M \times D$, where N and M are the width and the height of the image and D denotes the number of spectral bands. We aim at predicting the label of each pixel of the image. As the spatially adjacent pixels often have the same labels, it is desirable for the proposed model to consider the "spatial coherence." To this end, the first processing phase of our model is to extract a $K \times K \times D$ patch for each pixel. In particular, each patch (i.e., the spatial context) is constructed surrounding a pixel, the center point of the patch. For the pixels that reside near the edge of the image, there may not be sufficient information to build a patch of the expected size. Accordingly, we construct the spatial context by performing a mirror padding operation for these pixels.

For the second processing phase, each extracted patch is treated as an image with multiple channels on its own. Thereby, we can apply a deep CNN model with 2-D convolution layers to extract the feature maps for the patch. More specifically, the 2-D-CNN operator at each layer is formulated as follows:

$$v_{ij}^{xy} = F\left(b_{ij} + \sum_{m}\sum_{p=0}^{N_i-1}\sum_{q=0}^{M_i-1} w_{ijm}^{pq} v_{i-1}^{(x+p)(y+q)}\right) \quad (1)$$

where i indicates the particular layer under consideration, j is the number of feature maps of the layer i, v_{ij}^{xy} stands for the output at position (x, y) of the jth feature map at the ith layer, b_{ij} refers to the bias term, $F(\cdot)$ denotes the activation function of the layer, and m indexes over the set of feature maps of the $(i-1)$th layer, which are the inputs to the ith layer. w_{ij}^{mpq} is the value at position (p, q) of the convolution kernel connected to the ith feature map to the jth feature map, and N_i and M_i are the height and width of this kernel. For the proposed model, we adopt the ReLU function as the activation function F, which is defined as follows:

$$F(x) = \max(0, x). \quad (2)$$

In our 2-D-CNN model, three convolutional layers are utilized. To preserve the vital information of each pixel, we exclude the pooling layers from our 2-D-CNN model. Finally, a fully connected layer, which takes the feature maps of the last 2-D convolutional layer as inputs, is constructed to make the prediction. Here, we leverage the softmax function to compute the probability for each class. The softmax function is an extension of the sigmod function, and used for multiple classification. The purpose of the softmax function is to find the parameters in the maximum z value of Y_k. Moreover, the cross entropy function is adopted as the objective function to drive the BP-based training process.

Let $\mathbf{W}$ and $\mathbf{b}$ denote all the parameters of our 2-D-CNN model. We train the 2-D-CNN model by maximizing the likelihood, and transform the scores $f_c(I_{i,j,k}; (\mathbf{W}, \mathbf{b}))$ of each class of interest $c \in \{1, \ldots, N\}$ into the conditional probabilities using the following softmax function [42]:

$$p(c|I_{i,j,k};\ (\mathbf{W}, \mathbf{b})) = \frac{e^{f_c(I_{i,j,k};(\mathbf{W},\mathbf{b}))}}{\sum_{d\in\{1,\ldots,N\}} e^{f_d(I_{i,j,k};(\mathbf{W},\mathbf{b}))}}. \quad (3)$$

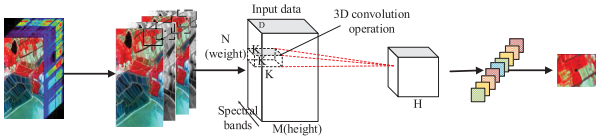


Fig. 3. 3-D-CNN model comprising three 3-D convolution operations with the corresponding kernel size (K) and the number of feature maps (m) for each convolutional layer.

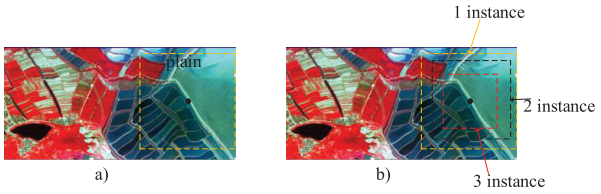


Fig. 4. (a) Context input patch of "plain." (b) Recurrent context input patch. The size of context input patch increases as the number of instances in the recurrent 2-D convolutional network increases.

The parameters (W, b) are learned by minimizing the negative log-likelihood based on the training set

$$L(\mathbf{W}, \mathbf{b}) = -\sum_{I_{i,j,k}} \ln p(l_{i,j,k}|I_{i,j,k}; (\mathbf{W}, \mathbf{b})) \tag{4}$$

where $l_{i,j,k}$ is the correct class label of the pixel at position (i, j) of the image I_k. To optimize the objective function, stochastic gradient descent (SGD) with BP is applied.

At the testing time, the output layer of the proposed model predicts the label of the pixel located at (i, j) of the image I using the argmax function

$$\widehat{l_{i,j}} = \underset{c\in\{1,\ldots,N\}}{\arg\max}\ p(c|I_{i,j}; (\mathbf{W}, \mathbf{b})). \tag{5}$$

B. 3-D-CNN Model

One main difference between a hyperspectral image and a conventional image is that the former is captured by scanning the same region with different spectral bands, while the latter is not. As the image formed by hyperspectral bands may have some correlations, e.g., close hyperspectral bands may result in similar images, it is desirable to take into account hyperspectral correlations. Though the 2-D-CNN model can utilize the spatial context, it ignores the hyperspectral correlations. Hence, we develop a 3-D-CNN model to address this issue.

As shown in Fig. 3, the operational details of our 3-D-CNN model are quite similar to those of the 2-D-CNN model. The main difference is that the 3-D-CNN model has one extra phase of reordering. In this phase, we rearrange the D hyperspectral bands according to an ascending order. By doing so, images of similar spectral bands are sequentially ordered, which can preserve their correlations under a spectral context. The patch extraction phase and the label identification phase of the two models are quite similar. For the feature extraction phase, a 3-D convolution operator instead of 2-D convolution operator is applied to the 3-D-CNN model.

More specifically, the 3-D convolution operation is formulated as follows:

$$v_{ij}^{xyz} = F\left(b_{ij} + \sum_{m}\sum_{p=0}^{N_i-1}\sum_{q=0}^{M_i-1}\sum_{r=0}^{D_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \tag{6}$$

where D_i is the size of the 3-D kernel along the spectral dimension and j is the number of kernels of the i layer; w_{ijm}^{pqr} is the value at the (p, q, r)th position of the kernel connected to the mth feature map (a cube) of the preceding layer. Again, the ReLU function is adopted as the activation function F.

The 3-D convolution operation is illustrated in Fig. 2. We can see that the 3-D convolution operation is applied to a 3-D patch step by step, e.g., from top to bottom, from left to right, and from inner to outer. In each step, a convolution scalar is produced and placed at the corresponding position of the feature map (shown as red lines in Fig. 2). This operation produces a smaller 3-D cube as a feature map. Training a 3-D-CNN model is similar to training a 2-D-CNN model in which we utilize the softmax function to compute the probability of each class. Moreover, we formulate the training process as an optimization problem by maximizing the log-likelihood of the training data. In addition, SGD with BP is applied for network training.

C. R-2-D-CNN Model

As noted earlier, though the 2-D-CNN model can exploit the spatial context, it may introduce unwanted noise, because the classification of a pixel relies on the features of a small patch surrounding the pixel rather than the features directly attached to the pixel. To better exploit the spatial context, we design an R-2-D-CNN model. In particular, the R-2-D-CNN model constructs multiple shrunk patches as multilevel instances (see Fig. 4) and leverages a multiscale deep neural network to fuse the multilevel instances for prediction. For clarity, we denote the instances as first level, second level, ..., and the Pth level, corresponding from the bigger patches to the smaller patches, where the Pth level often corresponds to the pixel for classification, i.e., a 1×1 patch. The R-2-D-CNN deep neural network comprises an R-CNN structure, where a basic 2-D-CNN block is reused multiple times. More specifically, it uses the basic 2-D-CNN block to extract the feature maps for the first-level instances at the beginning. These feature maps are then concatenated with the second-level instances, which are fed to the same 2-D-CNN block for extracting the next level feature maps. This procedure is repeated until the Pth level instances are fused. Finally, a softmax layer is then applied to compute the probability of each class. By utilizing the multiple shrunk patches, we can consider the spatial context information and can also focus more on the information closer to the pixel for classification. Hence, the unwanted noises can be reduced.

The main architecture of the R-2-D-CNN model is illustrated in Fig. 5. At the pth level, the network is fed with an input "feature image" F^p of $H + D$ (H represents the number of feature maps produced by the 2-D-CNN block) 2-D images,

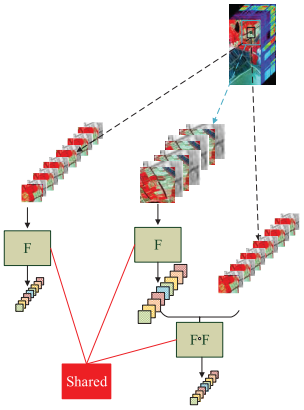


Fig. 5. R-2-D-CNN model comprising two basic 2-D-CNN block with parameters share across levels.

which comprises H feature maps of the $(p-1)$th instances, D hyperspectral images of the pth instances, and $1 \leq p \leq P$. Formally, the procedure is defined as follows:

$$F^p = \left[F\left(F^{p-1}, I^p_{i,j,k}\right)\right], \quad F^1 = [0, I_{i,j,k}] \tag{7}$$

where $I_{i,j,k}$ stands for the input patch surrounding the pixel at location (i, j) of the training image k. At the first level, the network only takes the original image as the input, because there is not instance from a previous layer to produce the feature maps. Though the R-2-D-CNN model is multilevel, the model complexity does not increase with respect to the number of levels. The reason is that the parameters pertaining to different levels are shared (as shown in Fig. 5).

Model training of the R-2-D-CNN model is the same as that of the 2-D-CNN model, where gradients are computed using the BP through time (BPTT) algorithm [43] during the BP process. More specifically, we first unfold the network as shown in Fig. 5, and then train the model with the BPTT algorithm. However, in contrast to the 2-D-CNN model, we have to learn the network parameters $(\mathbf{W}, \mathbf{b})$ by a new loss function due to the multilevel recurrent architecture. The loss function is defined according to (7)

$$L(F) + L(F \circ F) + \cdots + L(F \circ^p F) \tag{8}$$

where $L(F)$ is a shorthand for the log-likelihood defined in (3) of the 2-D-CNN model and $\circ^p$ denotes the composition operation performed p times. Thus, each network instance is trained to produce the correct label at the location (i, j). In this manner, the R-2-D-CNN model is able to learn and corrects its mistakes produced by the earlier iterations. As a by-product, the R-2-D-CNN model can also classify the dependences, that is, predicting the label of an instance based on the label of the previous instance around location (i, j).

It is worth noting that the sizes of multilevel instances in an R-2-D-CNN model must be carefully designed so that the instances can be concatenated with the feature maps of the previous instances. To this end, we first need to establish how the size of a feature map changes when it is applied to a 2-D convolution layer. Let sz_{m-1} denote the size of the feature map of the $(m-1)$th convolution layer. Then, the size of the feature map produced by the mth convolution layer is computed as follows:

$$sz_m = \frac{sz_{m-1} - kW_m}{dW_m} + 1 \tag{9}$$

where kW_m is the size of the convolution kernel of the mth layer and dW_m is the stride size. By (8), we can compute the size of a feature map produced by our 2-D-CNN block. Hence, we can estimate the appropriate sizes of the instances with respect to different levels.

D. R-3-D-CNN Model

To better utilize the spatial and spectral contexts of hyperspectral images, we design the R-3-D-CNN model. As for the R-2-D-CNN model, the R-3-D-CNN model is also underpinned by multilevel RNNs which shrink a patch gradually to form multilevel instances. There are two main differences between the R-3-D-CNN model and the R-2-D-CNN model. The first difference is that the former utilizes 3-D convolution operators, whereas the latter uses its 2-D counterparts. Hence, the R-3-D-CNN model can be regarded as an extension of the 3-D-CNN model in a recurrent manner. The second difference is that the instances of the next level need to be preprocessed and concatenated with the feature maps generated from the current level. The reason is that we adopt 3-D convolution layers which lead to variable length of the spectral bands. Hence, we have to preprocess the instances of the next level by some 3-D convolution operations of the spectral channels to adapt to the changing sizes.

Fig. 6 depicts an example of the proposed R-3-D-CNN model. The model consists of a multilevel RNN with P multilevel instances. As for the R-2-D-CNN model, a "plain" 3-D convolution network is applied to extract the corresponding feature maps, which are then concatenated with the next level instance to form new feature maps at each level. This procedure is repeated until all multilevel instances (patches) are incorporated. To ensure the consistency of the sizes of feature maps of the current level and the sizes of the instances at the next level, a preprocessing step is introduced to the spectral channels. Finally, a softmax layer is applied, and a cross entropy objective function is adopted. The optimization process is again performed using the BPTT algorithm. As for the R-2-D-CNN model, the complexity of the R-3-D-CNN model remains moderate, because the recurrent structure shares the same network parameters across multiple levels. As for the 3-D-CNN model, we need to reorder the hyperspectral images according to the ordering of spectral bands. Also, the size of multilevel instances must be carefully determined as for the R-2-D-CNN model.

IV. Experimental Results

We chose to use six publicly available hyperspectral image data sets for evaluating the performance of the proposed models. For a comparative evaluation, we also adopted SVM-CK, GCK, LORSAL, SMLR-SpATV, MFL, SVM-3-D, SVM-3-DG, and CNN-MRF as the baselines. For the performance metrics, we used the overall accuracy of all classes, denoted

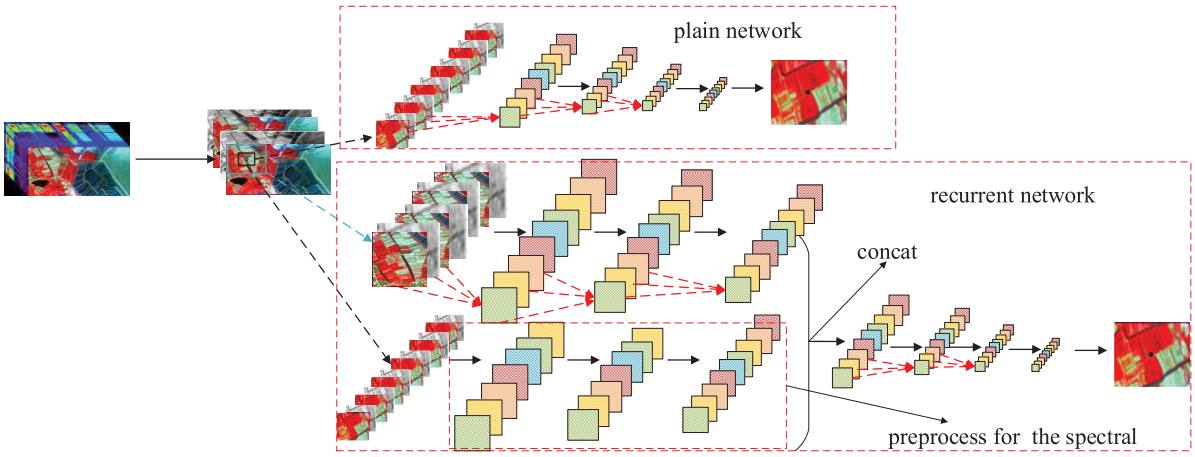


Fig. 6. R-3-D-CNN model with network parameters shared across multiple levels. The plain network is built with a small instance based on the basic 3-D-CNN model, while the recurrent network is built with two instances of the basic 3-D-CNN model; the complexity of the model remains moderate because of the shared parameters across multiple levels.

as OA, and the average accuracy of each class, denoted as AA. We ran all the models on a desktop PC equipped with an Intel Core 7 Duo CPU (at 3.40 GHz) with 12 GB of RAM, and two GTX 1080Ti GPUs (16 GB of ROM) were also used.

A. Data Sets

1) Indian Pines Scene: The data set was collected in 1992 by the AVIRIS sensor which records the remote sensing images of Indian Pines located at north-western India. The hyperspectral image contains 145 × 145 pixels in spatial dimensions and 224 hyperspectral bands. Due to the presence of noisy bands, we only used 200 hyperspectral bands. Specifically, the bands covering the regions of water absorption, i.e., [104–108], [150–163], and 220, were removed. The ground truth available includes 16 classes which are not all mutually exclusive. As shown in Fig. 7(a), we randomly divided the labeled data into the training (70%) and the testing (30%) sets for our experiment.

2) Botswana Scene: Botswana Scene was acquired by the Hyperion sensor on the NASA EO-1 satellite in May 31, 2001. This data set was collected over the Okavango Delta. The hyperspectral image contains 1476 × 1476 pixels taken by 224 bands, from 400 to 2500 nm with an incremental step of 10 nm. As for the Indian Pines Scene data set, we removed the noisy bands to produce an experimental data set containing 145 bands only. The image data set contains 14 categories. As shown in Fig. 7(b), we randomly split the data set to the training (70%) and the testing (30%) sets, respectively.

3) Salinas Scene: The Salinas Scene was a hyperspectral image data set recorded in 1992 by the AVIRIS sensor which captured images about the Salinas Valley, CA, USA. The original images were composed by 224 bands. We discarded 20 noisy bands, for example, bands [108–112], bands [154–167], and band 224 to generate a hyperspectral image data set of 204 bands. For the spatial dimensions, the scene includes 512 × 217 pixels. There are 16 labeled classes in the original data set, as shown in Fig. 7(c).

4) Pavia Center Scene: The hyperspectral image data set captured Pavia acquired over northern Italy. It was produced

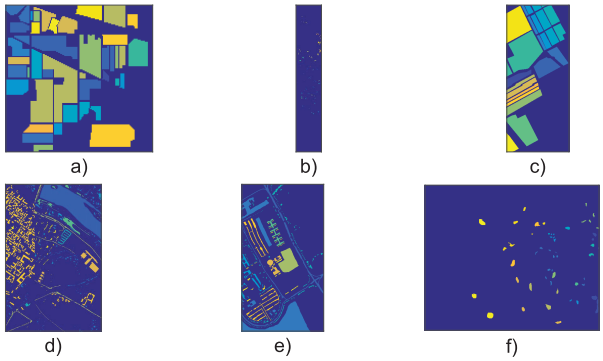


Fig. 7. Labeled images of different data sets. (a) Indian Pines Scene. (b) Botswana Scene. (c) Salinas Scene. (d) Pavia Center Scene. (e) Pavia University Scene. (f) KSC.

in 2001 using the reflective optics system imaging spectrometer (ROSIS) sensor. The Pavia Center Scene comprised 1096 × 1096 pixels with 114 hyperspectral bands. We preprocessed these images by removing 12 noisy bands. There are nine labeled classes in the data set, as shown in Fig. 7(d).

5) Pavia University Scene: This hyperspectral image data set captured the Pavia University in Italy by using the ROSIS sensor. There are 103 hyperspectral bands in the image data set, with 610 × 340 pixels for the spatial dimensions. The image contains nine labeled classes, as shown in Fig. 7(e).

6) Kennedy Space Center: The last data set, namely, Kennedy Space Center (KSC), captured the KSC area in Florida by using the AVIRIS sensor on March 23, 1996. The hyperspectral image consists of 512 × 614 pixels of spatial dimensions, with 224 spectral bands. After removing 48 noisy bands, we obtained 172 spectral bands. There are 13 labeled classes, as shown in Fig. 7(f).

B. Experimental Results

1) Results for the Indian Pines Scene: Before reporting the details of our experimental results, we first elaborate the various settings of the deep learning techniques employed

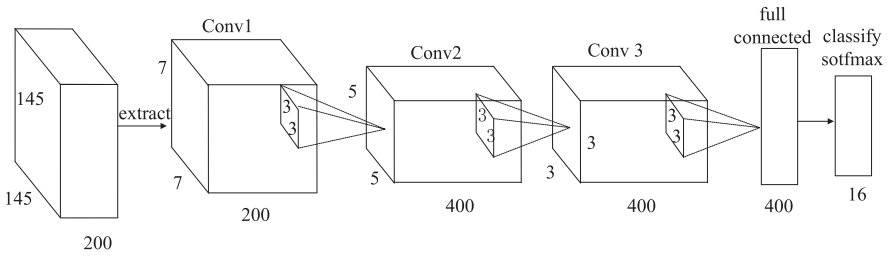


Fig. 8. 2-D-CNN network for hyperspectral remote sensing classification (yhe stride of each layer is 1).

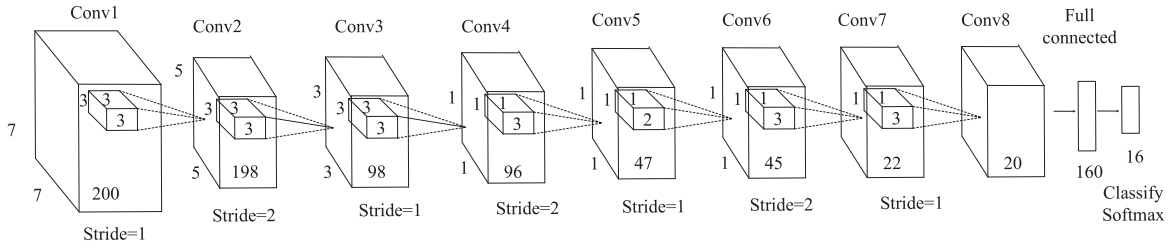


Fig. 9. 3-D-CNN network for remote sensing hyperspectral image classification.

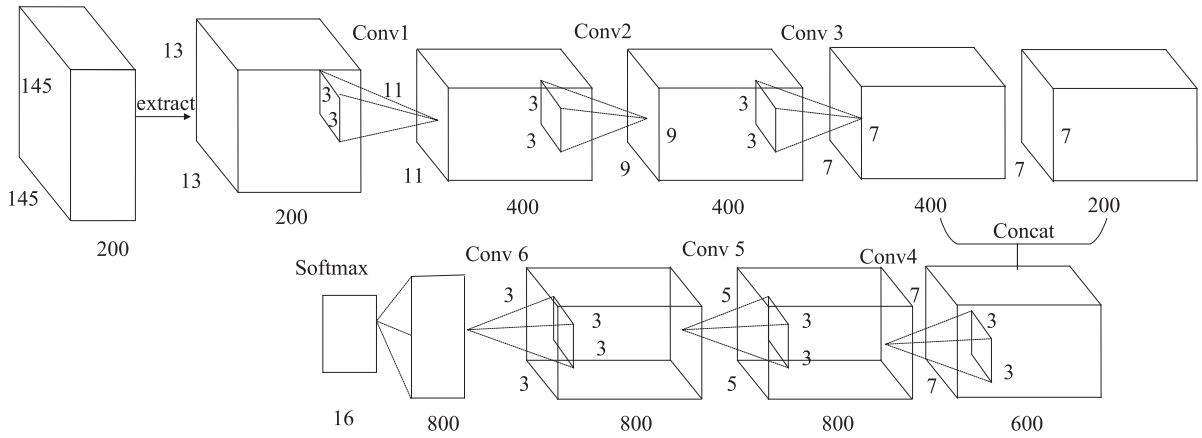


Fig. 10. R-2-D-CNN network for remote sensing hyperspectral image classification (the stride of each layer is 1).

in our experiments. The structure of the 2-D-CNN model is depicted in Fig. 8. For the classification of each pixel, a $7\times7\times200$ patch surrounding it is first constructed. Following this, three 2-D convolution layers of size 3×3 are utilized. Moreover, 200 spectral bands are treated as channels. The number of filters is set to 400 for the respective layers, and the stride is set to 1. As a result, the feature maps produced by the first, second, and last convolution layers are $5 \times 5 \times 400$, $3 \times 3 \times 400$, and $1 \times 1 \times 400$, respectively. Finally, a softmax layer of 16 classes is deployed to classify the images. The proposed network structure does not include the pooling layers so as to keep as much information of each pixel as possible. In addition, we apply SGD for network training and set the mini-batch size to 10.

The structure of the 3-D-CNN model is depicted in Fig. 9. Similar to the 2-D-CNN model, a $7 \times 7 \times 200$ patch is first extracted. Next, we build eight 3-D convolution layers. The size, number, stride, and feature map sizes of the 3-D filters in each layer are shown in Fig. 9. Again, we exclude the pooling layers and adopt a mini-batch size of 10. Before applying the 3-D-CNN model, the hyperspectral bands are first reordered.

Fig. 10 depicts the structure of the R-2-D-CNN model. For this model, the first-level instance is a $13 \times 13 \times 200$ patch. Then, we apply a three-layer 2-D-CNN block to the instance, which results in a $7 \times 7 \times 400$ feature map. After concatenating this feature map with our second-level instance, that is, a $7 \times 7 \times 200$ patch, and reusing the 2-D-CNN block, we obtain an 800-D vector, which is connected to a softmax layer to classify images. Again, we adopt a mini-batch size of 10 and do not utilize any pooling layers.

Similarly, we construct a $13 \times 13 \times 200$ patch and a $7 \times 7 \times 200$ patch at the first two levels of the R-3-D-CNN model. As shown in Fig. 11, we build a seven-layer 3-D-CNN block and apply it to the first-level instance. This produces a $7 \times 7 \times 187$ feature map. Since the spectral band dimension is changed from 200 to 187, we first apply a three-layer 3-D convolution operation to the second-level instance. By doing so, the third dimension is reduced to 187, and it can be concatenated with

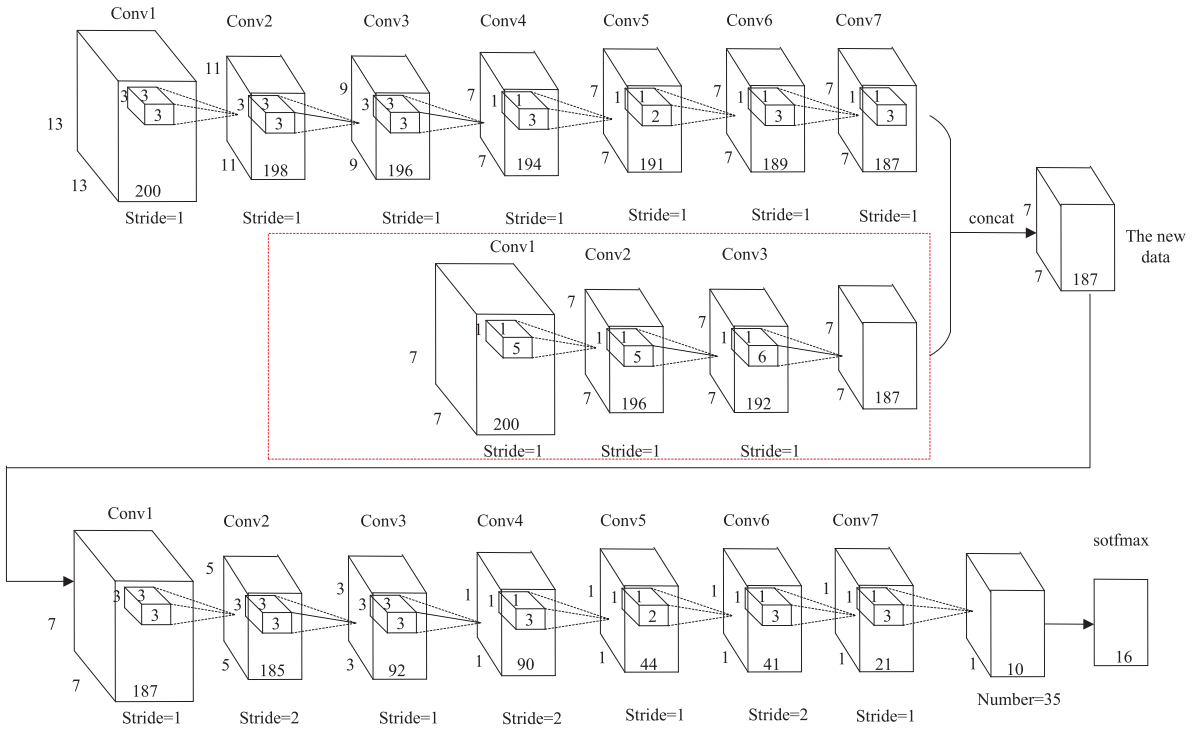


Fig. 11. R-3-D-CNN network for remote sensing hyperspectral image classification.

TABLE I
CLASSIFICATION RESULTS OF INDIAN PINES SCENE

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	85.71±0.4	92.86±0.5	85.71±0.1	92.86±0.2	85.71±0.3	57.14±0.4	64.29±0.4	84.62±0.2	71.72±1.0	85.71±0.4	78.57±0.1	100
2	86.82±0.3	98.12±0.4	89.88±0.3	98.59±0.2	96.24±0.2	79.29±0.3	80.00±0.2	65.65±0.3	95.85±0.6	96.46±0.3	99.29±0.1	100
3	86.12±0.2	94.29±0.3	82.04±0.1	98.37±0.2	92.65±0.3	71.02±0.3	73.47±0.4	96.36±0.4	95.90±0.3	97.13±0.1	98.77±0.3	100
4	88.40±0.5	94.20±0.3	82.61±0.3	100	97.10±0.4	97.10±0.4	97.10±0.5	88.73±0.3	73.91±0.1	98.55±0.3	100	100
5	95.10±0.4	96.50±0.4	91.61±0.3	98.60±0.2	97.20±0.3	95.80±0.1	91.61±0.3	93.06±0.2	97.20±0.2	97.90±0.2	97.90±0.2	100
6	98.61±0.7	99.08±0.1	99.08±0.2	99.54±0.4	99.54±0.2	98.16±0.3	97.70±0.4	99.09±0.4	96.31±0.3	97.68±0.5	99.53±0.3	100
7	75.00±0.1	100	75.00±0.1	100	100	75.00±0.3	62.50±0.3	50.00±0.1	100	100	87.50±0.2	100
8	98.60±0.1	100	97.90±0.2	100	100	99.30±0.2	100	95.10±0.4	100	99.30±0.07	100	100
9	100	83.33±0.3	83.33±0.1	83.33±0.3	100	100	100	100	100	100	100	100
10	87.19±0.7	93.43±0.4	85.81±0.5	98.27±0.3	92.04±0.2	75.09±0.4	75.78±0.4	76.63±0.2	97.20±0.6	98.26±0.3	98.95±0.3	99.65±0.3
11	91.01±0.8	98.09±0.8	88.83±0.1	99.59±0.2	98.50±0.3	88.01±0.2	95.37±0.4	97.55±0.2	99.04±0.4	98.77±0.4	99.45±0.2	99.31±0.2
12	94.84±0.6	94.89±0.4	88.64±0.2	99.43±0.3	96.02±0.3	85.80±0.4	86.36±0.3	76.27±0.3	95.45±0.6	97.15±0.4	99.43±0.2	98.85±0.2
13	100	100	100	100	98.36±0.3	98.36±0.3	98.36±0.2	100	100	96.72±0.1	98.36±0.1	100
14	96.81±0.7	99.20±0.2	96.02±0.2	100	99.47±0.2	97.88±0.4	97.08±0.4	99.47±0.4	98.94±0.3	99.46±0.6	100	99.73±0.2
15	81.57±0.8	95.61±0.6	83.33±0.3	97.37±0.2	97.37±0.3	86.84±0.2	100	95.65±0.2	94.73±0.6	93.80±0.5	98.24±0.2	96.46±0.3
16	100	100	85.71±0.1	100	100	82.14±0.2	782.14±0.3	100	100	100	96.42±0.8	96.42±0.5
AA	91.62±0.3	96.22±0.5	88.47±0.2	97.87±0.2	96.89±0.2	85.23±0.3	87.61±0.1	88.26±0.3	96.37±0.3	97.31±0.2	97.03±0.3	99.42±0.3
OA	91.51±0.2	97.44±0.4	90.10±0.1	99.11±0.2	97.05±0.3	86.55±0.2	89.44±0.2	88.95±0.2	97.08±0.2	98.92±0.4	99.19±0.3	99.50±0.3

the feature map of the first-level instance. Then, we reuse the seven-layer 3-D convolution block which produces a $1 \times 10 \times 35$ feature map. Finally, a softmax layer is applied to the resulting feature map to determine the class label.

Next, we report the experimental results based on various data sets. Table I presents the performance of all the methods. We observe that the R-3-D-CNN model achieves the best performance, of which the OA is 99.50%. Although the OA of the SMLR-SpATV is 99.11%, the R-3-D-CNN model outperforms it by more than 44%, if we consider the reduction of error rates. The main reason is that the R-3-D-CNN model considers both the spectral and the spatial contexts, where the former is inferred via the 3-D convolution operation and the latter is inferred using the multilevel recurrent structure. In terms of AA and OA, the R-2-D-CNN model is ranked as the second best, which is followed by the SMLR-SpATV, the 2-D-CNN model, and the 3-D-CNN model. Though the R-2-D-CNN ignores the spectral correlations, its recurrent structures can effectively capture the spatial context for subsequent image classification. Our experimental results also imply that the spatial context is more important than the spectral correlations for hyperspectral image classification. As shown in Table I, the results of SVM-CK are better than the SVM-3-D, SVM-3-DG, and CNN-MRF. However, its performance is much worse than various deep learning techniques. The first reason is that the SVM-CK classifier ignores the spatial and spectral

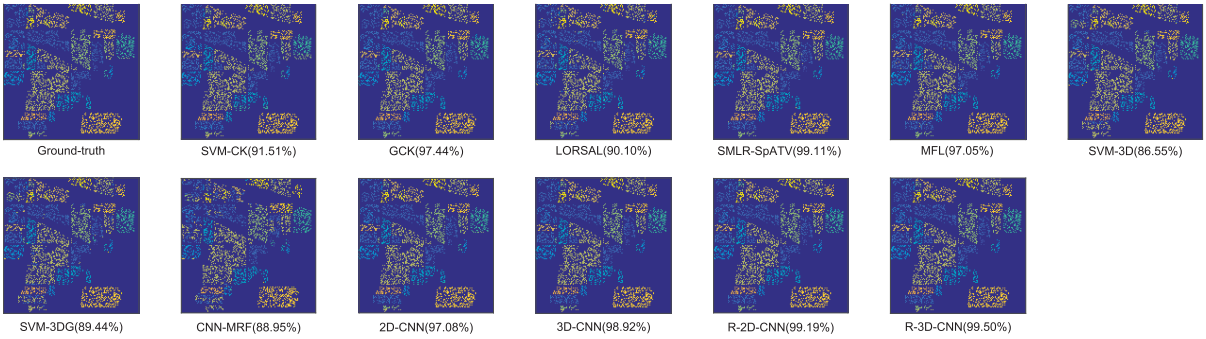


Fig. 12. Classification maps and overall classification accuracies obtained for the AVIRIS Indian Pines data set (OAs are reported in parentheses).

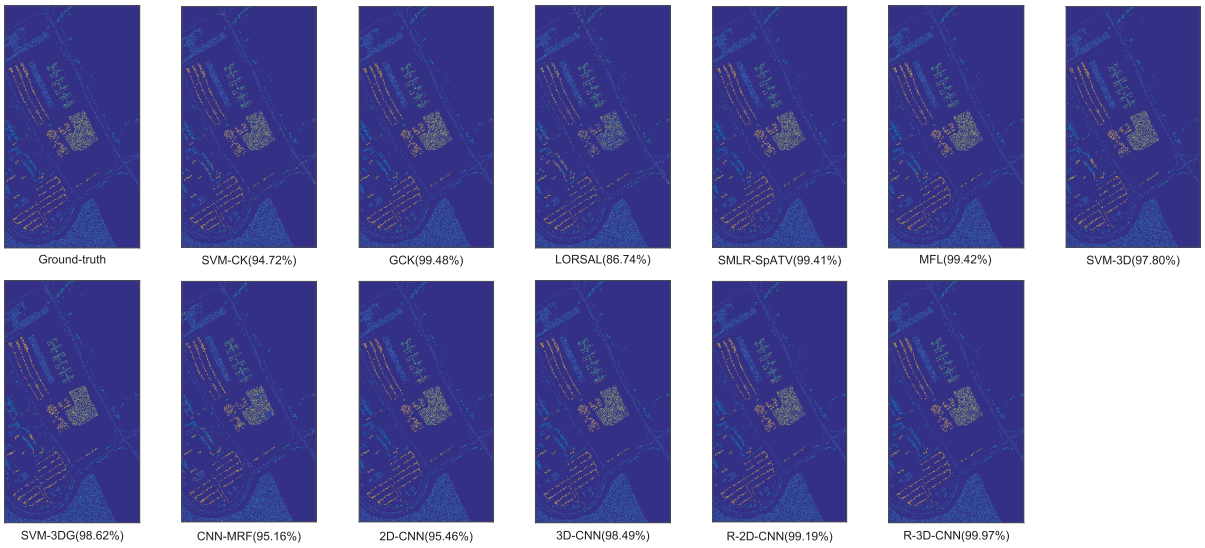


Fig. 13. Classification maps and overall classification accuracies obtained for the Pavia University Scene data set (OAs are reported in parentheses).

contexts. The second reason is that the SVM-CK classifier cannot effectively capture the nonlinear relationships between the features and the class labels of hyperspectral images. As a promising classification method, the GCK achieves a comparable performance as those of the 2-D-CNN and the 3-D-CNN methods, because it can extract EMAP information pertaining to the spectral and spatial contexts. Fig. 12 provides a visual comparison of the performance of all methods.

2) Results for the Pavia University Scene: In this experiment, the structures of the deep learning models were quite like those applied to the original Indian Pines Scene experiment. The only difference was that the numbers of parameters were adjusted to match the 102 hyperspectral bands of our refined data set. Recall that there were 200 bands of the original data set. Table II presents the experimental results of all the methods based on the Pavia University Scene data set. Again, we can see that the proposed R-3-D-CNN performs the best, followed by the R-2-D-CNN model, the SMLR-SpATV method, the MFL method, and the GCK method. The OA of the R-3-D-CNN model is 99.97%, which is 0.39% higher than that of the GCK (99.48%). And the R-CNN-3-D model outperforms the GCK method by more than 94%, when we consider the reduction of error rates. The SVM-3-DG method and the 3-D-CNN and 2-D-CNN models achieve comparable results. The LORSAL classifier produces the worst performance among all the methods. Fig. 13 visualizes the classification results of all the methods.

3) Results for the Botswana Scene: As for the earlier scenes, we only modified the number of parameters for our deep learning models. Table III presents the experimental results of all the methods based on the Botswana Scene data set. We can see that the proposed R-3-D-CNN model and the MFL method achieve the highest performance, followed by the SVM-3-DG, the GCK method, and the R-2-D-CNN model. The OA of the R-3-D-CNN model is 99.38%, which is 0.29% higher than that of the MFL (99.07%). And the R-3-D-CNN model outperforms the MFL method by more than 31% in terms of the reduction of error rates. Again, the other models, such as the 3-D-CNN, SMLR-SpATV, and 2-D-CNN models, perform better than the LORSAL classifier which produces the worst result. Fig. 14 visualizes the classification results of all the methods.

4) Results for the Salinas Scene: Table IV presents the experimental results of all the methods based on the Salinas Scene data set. The R-3-D-CNN model achieves the best performance, followed by the GCK method, the MFL method,

TABLE II
CLASSIFICATION RESULTS OF THE PAVIA UNIVERSITY SCENE

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	99.53±0.3	99.64±0.4	91.20±0.2	99.85±0.2	100	98.14±0.2	99.45±0.2	99.60±0.4	91.65±0.2	98.70±0.3	99.34±0.3	100
2	98.59±0.5	99.85±0.3	96.92±0.1	100	99.93±0.3	99.30±0.2	99.86±0.3	98.09±0.4	99.45±0.1	99.77±0.4	99.96±0.4	100
3	85.21±0.3	97.61±0.5	64.07±0.3	100	93.64±0.2	88.08±0.4	87.12±0.1	76.31±0.2	92.09±0.3	97.94±0.4	98.88±0.2	100
4	96.52±0.2	98.62±0.2	88.57±0.4	93.58±0.4	98.59±0.5	99.02±0.2	99.67±0.3	96.08±0.4	87.26±0.4	94.55±0.2	93.91±0.3	99.89±0.1
5	99.75±0.5	99.34±0.7	99.75±0.2	100	99.50±0.3	100	100	99.75±0.2	91.05±0.2	97.77±0.1	98.27±0.2	100
6	92.24±0.4	99.78±0.5	57.76±0.3	100	99.67±0.3	96.82±0.1	99.07±0.3	88.66±0.3	98.61±0.4	99.60±0.2	99.94±0.4	100
7	91.71±0.1	99.41±0.5	59.05±0.2	100	99.75±0.2	95.48±0.3	95.73±0.2	83.71±0.1	90.72±0.1	98.00±0.2	98.49±0.2	100
8	93.30±0.2	98.52±0.4	80.45±0.3	99.82±0.2	99.10±0.4	95.20±0.4	96.38±0.4	92.75±0.2	93.57±0.4	98.55±0.3	99.81±0.2	100
9	99.65±0.5	99.65±0.3	97.89±0.4	95.77±0.3	100	98.94±0.2	98.94±0.2	98.59±0.4	86.62±0.2	81.34±0.2	94.72±0.3	98.94±0.3
AA	94.52±0.3	99.21±0.3	81.74±0.2	98.78±0.4	98.91±0.2	96.78±0.4	97.36±0.2	92.62±0.2	98.78±0.4	96.25±0.3	98.15±0.2	99.87±0.3
OA	94.72±0.2	99.48±0.2	86.74±0.2	99.41±0.2	99.42±0.2	97.80±0.4	98.62±0.1	95.16±0.3	95.46±0.2	98.49±0.2	99.19±0.2	99.97±0.2

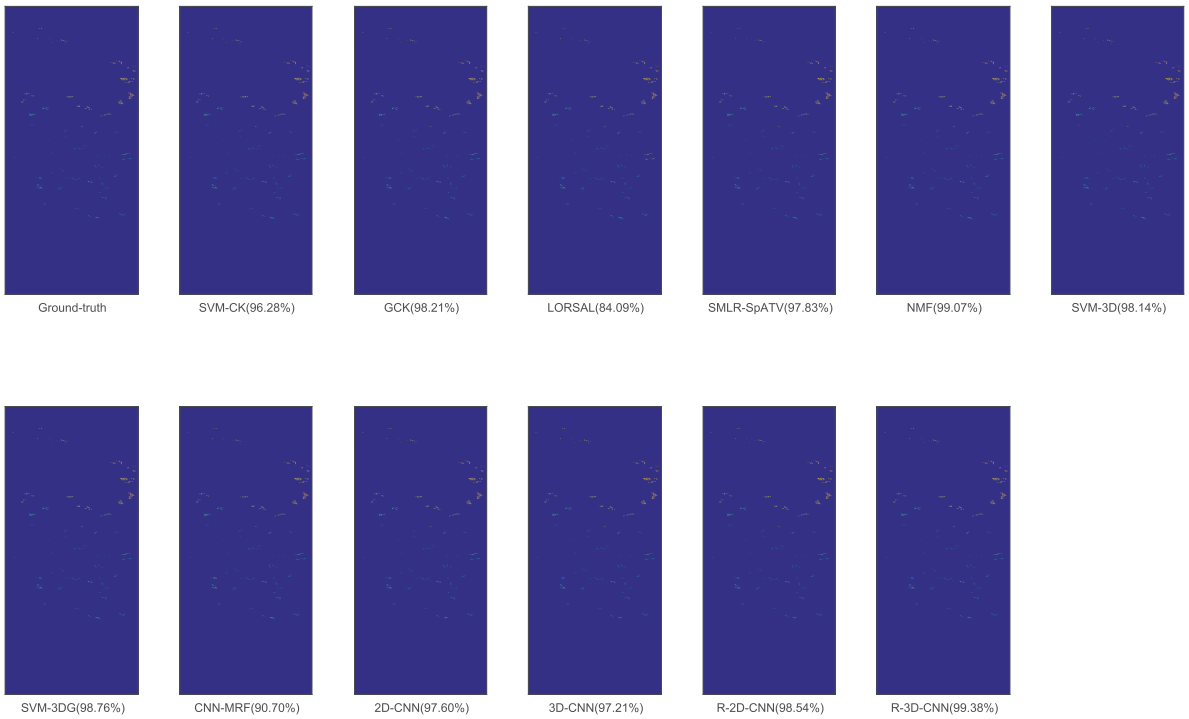


Fig. 14. Classification maps and overall classification accuracies obtained for the Botswana Scene data set (OAs are reported in parentheses).

and the R-2-D-CNN model. The SMLR-SpATV, 3-D-CNN, and 2-D-CNN models also achieve promising results. The OA of the R-3-D-CNN model is 99.80%, which is 0.46% higher than that of the GCK (99.34%). And the R-3-D-CNN model improves the GCK method by more than 70%, when we consider the reduction of error rates. Again, the LORSAL classifier produces the worst result among all the methods. Due to memory limitations of our computer, we cannot perform the CNN-MRF classifier on this data set. Fig. 15 visualizes the classification results of all the methods.

5) Results for the Pavia Center Scene: Table V presents the experimental results of all the methods based on the Pavia Center Scene data set. The results are quite different from those obtained based on the previous four data sets. We observe that the R-2-D-CNN performs the best, followed by the SVM-3-DG and the MFL classifiers. And, the R-2-D-CNN model outperforms the SVM-3-DG method by more than 88% in terms of the reduction of error rates. The R-3-D-CNN model, which achieves the best performance based on the previous data sets, produces unsatisfactory results when compared to those of the R-2-D-CNN model and the SVM-CK classifier. The reason may be that the R-3-D-CNN model fuses the spectral and the spatial information using a 3-D operator. However, the channel of the Pavia Center Scene contains 102 bands only; it is smaller than the other data sets. On the other hand, the 2-D-CNN and 3-D-CNN models perform worst among all the methods, because it is difficult for these models to classify the third class and the ninth class due to the limited number of instances and channels. Since the methods, such as the GCK, the SMLR-SpATV, and CNN-MRF, require more RAM than that equipped with our computer, we cannot obtain their performance on the data set. Fig. 16 shows the classification results of all the methods.

6) Results for the Kennedy Space Center Scene: Table VI presents the experimental results of all the methods based on the KSC Scene data set. Again, we observe that the

TABLE III
CLASSIFICATION RESULTS OF THE BOTSWANA SCENE

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	100	100	100	100	100	100	100	100	98.77±0.4	100	100	100
2	100	100	90.00±0.1	100	100	100	100	100	93.33±0.3	93.33±0.2	100	100
3	98.66±0.8	100	93.33±0.3	100	97.33±0.3	98.67±0.3	100	98.67±0.3	97.33±0.3	94.67±0.3	98.66±0.2	100
4	96.87±0.5	99.48±0.6	89.06±0.3	100	100	98.44±0.2	100	79.69±0.4	98.44±0.5	96.88±0.3	96.87±0.3	100
5	92.59±0.2	93.17±0.2	76.54±0.2	97.53±0.1	97.53±0.1	97.53±0.3	98.77±0.3	90.00±0.1	91.36±0.3	96.30±0.3	96.29±0.2	97.53±0.3
6	90.12±0.4	93.97±0.5	58.02±0.1	98.77±0.3	100	92.59±0.4	88.89±0.4	90.00±0.2	100	98.63±0.1	100	100
7	100	100	98.68±0.4	100	100	100	100	92.21±0.1	100	98.69±0.3	100	100
8	100	100	86.67±0.3	98.33±0.2	100	100	100	90.00±0.3	100	95.00±0.1	96.66±0.2	98.33±0.3
9	96.80±0.6	98.33±0.7	78.72±0.1	100	96.81±0.2	97.87±0.4	100	96.81±0.1	77.66±0.8	95.75±0.2	100	98.94±0.4
10	94.59±0.3	100	74.32±0.1	100	100	97.30±0.4	100	87.84±0.2	96.81±0.4	100	98.64±0.2	100
11	92.30±0.4	97.54±0.4	90.11±0.2	100	98.90±0.3	96.70±0.4	98.90±0.4	100	98.65±0.3	100	100	100
12	94.93±0.5	100	90.74±0.2	98.15±0.3	100	100	100	64.81±0.1	100	100	96.29±0.4	98.14±0.4
13	91.53±0.3	99.59±0.7	93.67±0.3	100	100	100	100	95.00±0.3	98.73±0.5	100	100	100
14	100	96.00±0.1	32.14±0.2	42.86±0.2	96.43±0.2	96.43±0.3	96.43±0.2	53.57±0.3	92.86±0.3	92.86±0.4	85.71±0.4	96.43±0.3
AA	96.79±0.3	98.33±0.6	82.29±0.4	95.40±0.4	99.07±0.3	98.25±0.3	98.78±0.4	88.47±0.3	97.21±0.3	97.30±0.3	98.89±0.3	99.24±0.2
OA	96.28±0.1	98.21±0.2	84.09±0.3	97.83±0.2	99.07±0.4	98.14±0.2	98.76±0.3	90.70±0.4	97.60±0.5	97.21±0.1	98.54±0.2	99.38±0.2

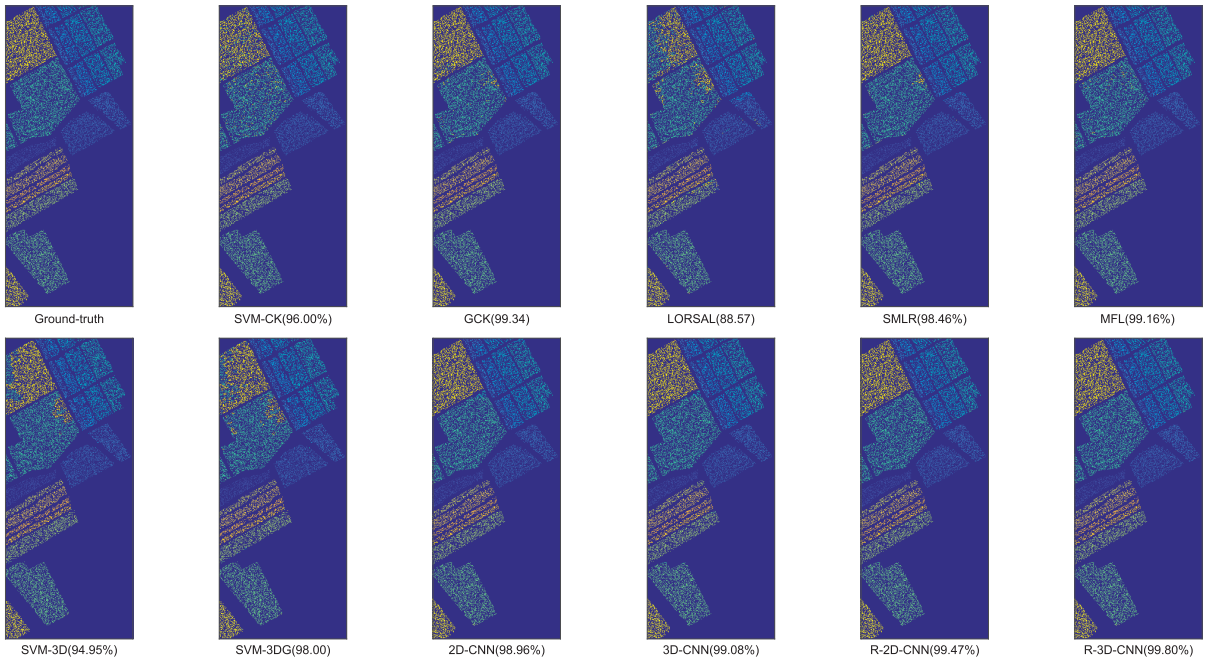


Fig. 15. Classification maps and overall classification accuracies obtained for the Salinas Scene data set (OAs are reported in parentheses).

proposed R-3-D-CNN performs the best, followed by the R-2-D-CNN model and the GCK model. The R-3-D-CNN model outperforms the GCK method by more than 95% in terms of error rate. The 3-D-CNN model, the MFL method, and the SVM-3-DG model achieve comparable results, followed by the CNN-MRF method, the 2-D-CNN model, and the SVM-CK method. The SVM-3-D classifier produces the worst result, and the LORSAL method outperforms the SVM-3-D method by 2% in terms of OA. Fig. 17 shows the classification results of all the methods.

C. Convergent Speed Comparison

Fig. 18 shows the accuracies of different deep learning models against the number of iterations based on the six data sets. We observe that the R-3-D-CNN model can converge with fewer number of iterations when compared to the other models, with the only exception for the Salinas data set. The efficiency improvement brought by the R-3-D-CNN model is attributed to the recurrent structure and the 3-D convolutional operation. Specifically, the feature maps that are extracted by the R-3-D-CNN model contain richer contextual information of the images, which leads to a quicker convergence of model training.

D. Impact of the Size of Training Samples

In this experiment, we examined how the performance of the proposed deep learning models is changed against varying sizes of the training samples. To this end, we varied the number of training samples from 10% to 70%, and reported the OA achieved by all methods. Fig. 19 shows the results based on six data sets. From Fig. 19, we can make two important observations. First, for the conventional classifiers,

TABLE IV
CLASSIFICATION RESULTS OF THE SALINAS SCENE

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	99.83±0.3	100	96.85±0.2	99.00±0.3	100	100	100	-	99.50±0.2	97.68±0.3	100	99.83±0.2
2	100	99.82±0.2	98.93±0.2	99.82±0.1	100	99.82±0.2	99.91±0.3	-	99.82±0.5	99.46±0.2	99.46±0.3	99.91±0.4
3	100	100	80.24±0.2	95.61±0.1	100	99.49±0.4	99.49±0.4	-	98.31±0.3	97.80±0.5	99.49±0.1	99.66±0.2
4	99.04±0.4	99.76±0.3	99.28±0.4	99.76±0.3	99.52±0.1	99.04±0.2	99.28±0.4	-	97.61±0.4	97.13±0.3	97.13±0.2	97.37±0.4
5	99.75±0.5	99.50±0.1	98.13±0.2	99.75±0.2	98.63±0.3	99.63±0.3	99.75±0.2	-	98.50±0.5	98.80±0.4	99.13±0.3	99.50±0.1
6	100	100	99.92±0.1	100	99.83±0.3	100	100	-	99.75±0.3	98.91±0.3	99.07±0.3	100
7	99.91±0.1	99.81±0.2	99.16±0.3	99.72±0.1	99.44±0.2	100	100	-	98.42±0.4	96.55±0.4	99.81±0.4	99.53±0.1
8	92.69±0.5	96.54±0.5	88.10±0.4	96.98±0.4	97.93±0.3	92.28±0.3	94.11±0.2	-	99.47±0.4	97.28±0.3	99.91±0.2	99.97±0.4
9	100	100	99.25±0.3	100	100	99.62±0.1	99.62±0.1	-	99.89±0.2	99.84±0.2	99.84±0.2	100
10	99.08±0.4	99.80±0.3	84.33±0.2	96.64±0.2	99.69±0.4	97.55±0.3	98.88±0.4	-	99.70±0.1	98.78±0.2	99.18±0.2	99.90±0.2
11	100	99.69±0.4	85.89±0.4	94.36±0.3	99.69±0.3	98.75±0.3	98.75±0.2	-	99.37±0.5	99.06±0.3	99.06±0.3	100
12	100	100	100	100	100	100	100	-	98.09±0.6	99.14±0.4	98.27±0.5	99.65±0.4
13	99.64±0.4	99.27±0.3	98.91±0.2	99.27±0.4	98.55±0.3	98.55±0.3	98.55±0.3	-	96.00±0.1	90.55±0.5	98.55±0.4	100
14	98.75±0.5	99.07±0.4	94.08±0.4	99.38±0.3	95.64±0.2	96.57±0.2	98.75±0.2	-	96.89±0.3	93.46±0.6	97.20±0.3	98.44±0.2
15	82.61±0.2	96.14±0.4	52.98±0.3	97.84±0.2	99.17±0.4	77.37±0.3	81.46±0.3	-	99.22±0.4	97.47±0.4	99.73±0.4	100
16	99.82±0.3	100	96.67±0.4	98.89±0.4	98.89±0.4	99.26±0.3	99.45±0.3	-	99.41±0.1	99.06±0.3	99.81±0.2	100
AA	96.06±0.5	98.66±0.4	92.05±0.3	98.56±0.3	99.19±0.4	97.37±0.3	98.00±0.4	-	98.90±0.3	98.65±0.2	99.12±0.4	99.61±0.2
OA	96.00±0.3	99.34±0.3	88.57±0.3	98.46±0.3	99.16±0.3	94.95±0.2	96.03±0.3	-	98.96±0.4	99.08±0.3	99.47±0.3	99.80±0.2

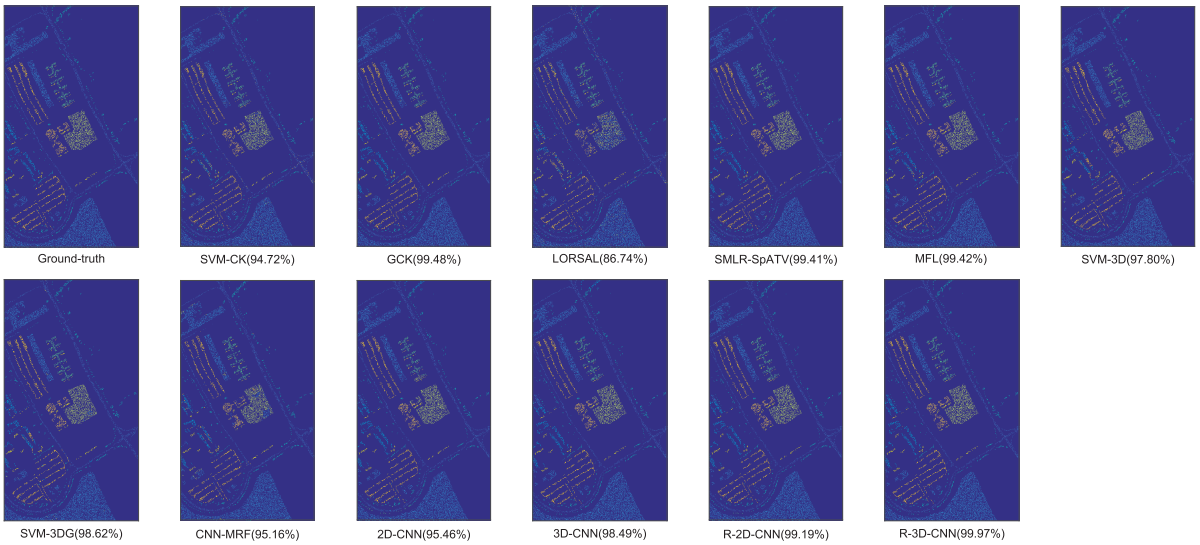


Fig. 16. Classification maps and overall classification accuracies obtained for the Pavia Center Scene data set (OAs are reported in parentheses).

TABLE V
CLASSIFICATION RESULTS OF THE PAVIA CENTER SCENE

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	100	-	-	-	99.99±0.4	100	100	-	99.36±0.2	99.85±0.2	100	99.32±0.2
2	98.24±0.4	-	-	-	97.41±0.2	96.14±0.2	98.03±0.2	-	87.51±0.3	95.73±0.2	99.65±0.4	86.52±0.3
3	96.54±0.5	-	-	-	91.15±0.3	93.96±0.3	92.34±0.2	-	88.35±0.5	95.73±0.1	99.78±0.3	85.11±0.3
4	94.30±0.2	-	-	-	97.89±0.2	90.20±0.2	94.42±0.1	-	88.59±0.4	95.04±0.1	99.88±0.4	91.80±0.1
5	98.18±0.3	-	-	-	99.70±0.4	99.09±0.4	99.44±0.2	-	91.89±0.4	97.52±0.2	99.85±0.4	92.95±0.3
6	99.31±0.5	-	-	-	98.59±0.3	98.34±0.2	99.24±0.2	-	90.48±0.3	97.76±0.3	99.75±0.3	90.87±0.4
7	95.38±0.4	-	-	-	93.87±0.3	96.89±0.5	98.35±0.3	-	96.30±0.4	99.36±0.2	99.95±0.2	96.97±0.2
8	99.80±0.5	-	-	-	99.87±0.2	99.93±0.2	99.95±0.2	-	96.98±0.5	98.84±0.3	99.90±0.3	96.49±0.1
9	100	-	-	-	94.76±0.3	99.65±0.3	99.88±0.4	-	73.69±0.1	91.51±0.4	98.25±0.3	83.80±0.2
AA	97.97±0.4	-	-	-	97.03±0.2	97.13±0.2	97.96±0.3	-	90.32±0.4	96.82±0.3	99.67±0.2	91.54±0.2
OA	97.32±0.3	-	-	-	99.10±0.2	99.17±0.2	99.47±0.3	-	96.02±0.4	98.75±0.1	99.88±0.3	96.79±0.2

i.e., GCK, MFL, SVM-CK, SVM-3-D, SVM-3-DG, SMLR-SpATV, and LORSAL, we find that their classification performances are insensitive to the number of training samples, especially on the Bostwana Scene, Salinas Scene, Pavia Center Scene, Pavia University Scene, and KSC data sets. Promising results are achieved when 10% training samples are utilized

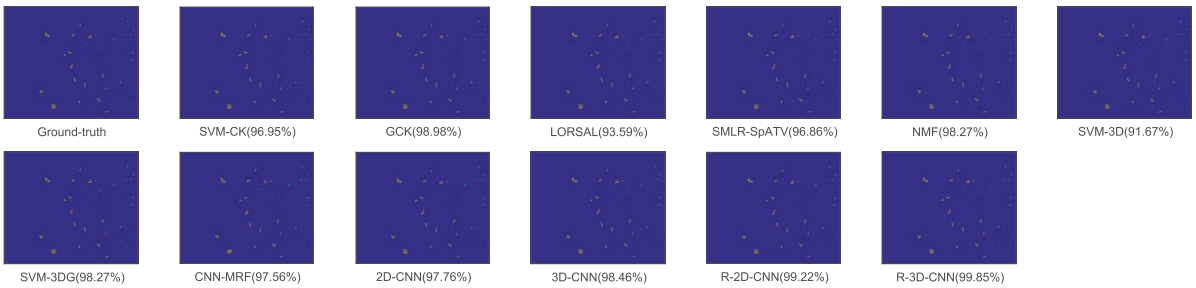


Fig. 17. Classification maps and overall classification accuracies obtained for the KSC data set (OAs are reported in parentheses).

TABLE VI
CLASSIFICATION RESULTS OF THE KSC

Class	Methods											
#	SVM-CK [17]	GCK [23]	LORSAL [26]	SMLR-SpATV [27]	MFL [28]	SVM-3D [29]	SVM-3DG [29]	CNN-MRF [35]	2D-CNN	3D-CNN	R-2D-CNN	R-3D-CNN
1	96.49±0.2	100	94.74±0.2	97.37±0.4	99.56±0.3	98.25±0.2	99.56±0.3	96.50±0.2	98.68±0.4	98.68±0.4	100	100
2	95.90±0.2	100	90.41±0.1	98.63±0.2	100	86.30±0.3	97.26±0.2	97.22±0.2	100	91.78±0.1	100	97.26±0.2
3	97.40±0.3	98.70±0.3	93.51±0.2	98.70±0.4	100	97.40±0.3	97.40±0.2	98.68±0.4	100	98.78±0.1	100	100
4	86.84±0.2	90.79±0.4	75.00±0.4	84.21±0.1	100	77.63±0.2	96.05±0.2	77.33±0.2	94.73±0.6	97.37±0.4	94.74±0.3	98.68±0.4
5	77.08±0.3	97.92±0.3	72.92±0.2	81.25±0.2	100	83.33±0.4	83.33±0.3	83.33±0.3	93.75±0.2	91.67±0.6	97.92±0.3	100
6	78.26±0.1	100	75.82±0.3	90.11±0.2	99.56±0.3	98.55±0.3	98.55±0.4	100	100	97.10±0.1	98.55±0.1	98.55±0.1
7	87.09±0.6	100	96.77±0.3	100	100	100	100	100	100	100	100	100
8	96.90±0.9	99.23±0.4	96.90±0.4	98.45±0.4	98.45±0.3	93.20±0.4	94.57±0.3	99.22±0.2	93.80±0.8	100	96.90±0.2	100
9	100	100	98.72±0.1	100	99.36±0.3	100	100	100	100	99.36±0.6	100	99.35±0.1
10	97.50±0.1	98.33±0.3	97.50±0.4	98.45±0.3	98.33±0.3	68.33±0.2	98.33±0.2	100	98.32±0.8	.94.12±0.6	100	100
11	99.20±0.1	100	100	100	87.20±0.4	99.20±0.2	99.200±0.4	100	100	100	96.80±0.1	99.20±0.1
12	98.67±0.5	97.35±0.1	92.72±0.1	95.36±0.3	96.69±0.4	96.03±0.2	100	100	98.01±0.3	96.69±0.5	99.34±0.3	98.68±0.2
13	100	100	97.84±0.4	100	100	87.77±0.4	100	100	97.48±0.2	97.12±0.2	100	98.56±0.1
AA	95.96±0.2	98.64±0.1	90.95±0.3	95.33±0.2	98.43±0.2	91.22±0.2	97.25±0.3	96.30±0.2	97.81±0.3	98.20±0.1	98.79±0.4	99.23±0.1
OA	96.95±0.6	98.98±0.3	93.59±0.4	96.86±0.3	98.27±0.3	91.67±0.3	98.27±0.3	97.56±0.3	97.76±0.5	98.46±0.3	99.22±0.4	99.85±0.3

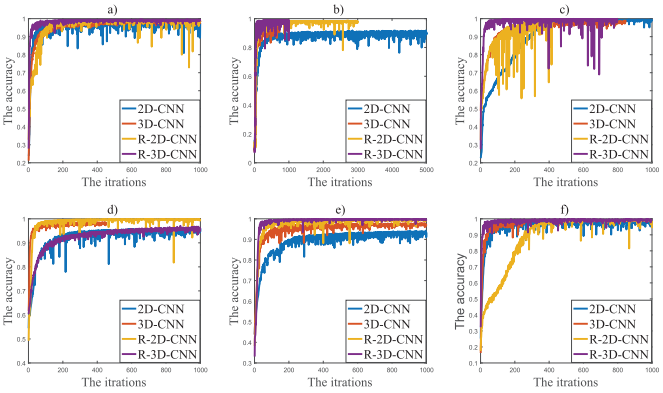


Fig. 18. Accuracy varies against the number of iterations on different data sets. (a) Indian Pines Scene. (b) Bostwana Scene. (c) Salinas Scene. (d) Pavia Center Scene. (e) Pavia University Scene. (f) KSC.

for these methods, and feeding more training samples to these methods only leads to marginal performance improvement. Among the conventional methods, the GCK and MFL methods often perform the best. Second, we can observe an obvious performance improvement when the number of training samples is increased (from 10% to 50%) for the proposed deep learning models, i.e., 2-D-CNN, 3-D-CNN, R-2-D-CNN, and R-3-D-CNN. When more than 60% training samples are employed, the R-2-D-CNN and R-3-D-CNN models often achieve comparable results when compared to the best conventional methods such as GCK and MFL. Moreover, we observe that the deep learning-based model CNN-MRF produces unstable classification performance, that is, results can be better or worse when more training samples are used. As a matter of fact, the proposed R-2-D-CNN and R-3-D-CNN models outperform CNN-MRF when sufficient training samples are provided. All these observations indicate that the proposed deep learning models (R-2-D-CNN and R-3-D-CNN) are more effective than the baselines when sufficient training samples are provided.

E. Discussion

In this section, we briefly discuss the experimental results presented earlier. First, we find that the R-3-D-CNN model often performs better than other models across all the six data sets. There are two possible reasons for such a performance improvement: 1) the R-3-D-CNN model effectively fuses the spatial and spectral correlations and 2) the multilevel recurrent structure can exploit spatial contexts better than a flat nonrecurrent structure. For the same reason, we also observe that the R-2-D-CNN model often outperforms the other models (such as LORSAL, GCK, SMLR-SpATV, CNN-MRF, SVM-CK, SVM-3-D, SVM-3-DG, and MFL). The MFL and GCK models perform better than the 2-D-CNN and 3-D-CNN models because of their well-designed EMAP attributes which can effectively represent the spatial contexts. On the other hand, the 2-D-CNN and 3-D-CNN models perform better than the SVM-CK, the SVM-3-D, and the LORSAL classifiers in general. All our experimental results verify the effectiveness and the advantages of the deep learning-based methods.

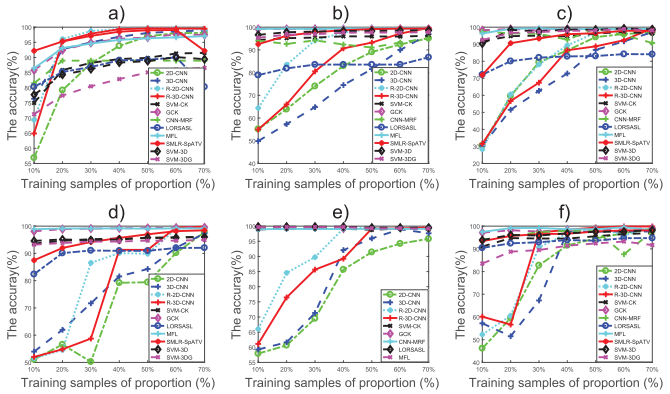


Fig. 19. Influence of training samples proportion. (a) Indian Pines Scene. (b) Botswana Scene. (c) Salinas Scene. (d) Pavia Center Scene. (e) Pavia University Scene. (f) KSC.

Second, the 3-D-CNN model often performs better than the 2-D-CNN model. The main reason is that the 3-D convolution operation can exploit both spatial features and spectral correlations, while the 2-D convolution operation can only exploit spatial features. On the other hand, the R-2-D-CNN model often performs better than the 3-D-CNN and 2-D-CNN models, because its recurrent structure can more effectively exploit the spatial contexts than the latter two models. Among all the four models, the R-3-D-CNN model not only performs the best for most data sets but it also converges faster.

Finally, we find that the proposed deep learning models (e.g., R-3-D-CNN and R-2-D-CNN) may be slightly inferior to conventional machine learning techniques if the training samples are limited. However, when a reasonable number of training samples are available, their performance is considerably better than that of the conventional machine learning techniques, such as LORSAL, GCK, MFL, SMLR-SpATV, SVM-3-D, SVM-3-DG, and SVM-CK. The main reason is that deep learning models usually contain more model parameters, and hence, more training samples are required to estimate the values of these parameters.

V. Conclusion

In this paper, we have explored deep learning techniques for solving the hyperspectral image classification problem. In particular, four deep learning models, such as 2-D-CNN, 3-D-CNN, R-2-D-CNN, and R-3-D-CNN, have been designed and developed. Rigorous experiments were conducted based on six publicly available hyperspectral image data sets, and our experimental results confirm the superiority of these deep learning methods when compared to traditional machine learning methods, such as LORSAL, MFL, GCK, SVM-3-D, and SVM-CK. In addition, the proposed R-3-D-CNN and R-2-D-CNN models outperform the CNN-MRF, SVM-3-DG, and SMLR-SpATV. As a whole, the proposed R-3-D-CNN model often outperforms other models for most of the data sets, and it can also converge faster because of its 3-D convolutional operators and the recurrent network structure which can effectively exploit both the spectral and the spatial contexts. If we measure a classification performance in terms of error rate, the proposed methods (R-2-D-CNN and R-3-D-CNN) outperform the baselines by more than 30%. Despite the superiority of the proposed models, we find that our deep learning models often require more training samples than the traditional machine learning methods. Accordingly, it will be a very interesting future research topic of incorporating prior domain knowledge into the proposed deep learning models. Alternatively, we will explore applying transfer learning approaches to alleviate the shortcomings of our current deep learning models.

References

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[2] A. J. Brown, B. Sutter, and S. Dunagan, "The MARTE VNIR imaging spectrometer experiment: Design and analysis," *Astrobiology*, vol. 8, no. 5, pp. 1001–1011, 2008.

[3] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[4] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.

[5] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, Jun. 2005.

[6] Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2077–2081, Nov. 2017.

[7] 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.

[8] Y. Yuan, J. Lin, and Q. Wang, "Dual-clustering-based hyperspectral band selection by contextual analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1431–1445, Mar. 2016.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, 2017, pp. 4278–4284.

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

[12] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

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

[16] A. J. Brown, "Spectral curve fitting for automatic hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1601–1608, Jun. 2006.

[17] 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. 2004.

[18] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[19] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3780–3791, Oct. 2010.

[20] Y. Gu, C. Wang, D. You, Y. Zhang, S. Wang, and Y. Zhang, "Representative multiple kernel learning for classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 7, pp. 2852–2865, Jul. 2012.

[21] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.

[22] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *Int. J. Remote Sens.*, vol. 31, no. 22, pp. 5975–5991, Dec. 2010.

[23] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[24] Q. Huang, C. K. Jia, X. Zhang, and Y. Ye, "Learning discriminative subspace models for weakly supervised face detection," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2956–2964, Dec. 2017.

[25] X. Ma, Q. Liu, Z. He, X. Zhang, and W.-S. Chen, "Visual tracking via exemplar regression model," *Knowl.-Based Syst.*, vol. 106, pp. 26–37, Aug. 2016.

[26] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.

[27] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, "Supervised spectral–spatial hyperspectral image classification with weighted Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1490–1503, Mar. 2015.

[28] J. Li *et al.*, "Multiple feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, Mar. 2015.

[29] 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.

[30] Y. Le Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[31] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Aistats*, vol. 15. 2011, pp. 315–323.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] 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.

[35] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley. (2017). "Hyperspectral image classification with Markov random fields and a convolutional neural network." [Online]. Available: https://arxiv.org/abs/1705.00727

[36] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[37] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[38] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 6645–6649.

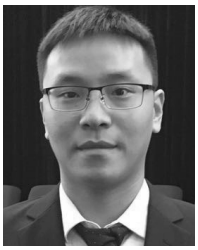
[39] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[40] K. Cho *et al.* (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

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

[42] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. 31st Int. Conf. Mach. Learn.*, Jun. 2014, pp. 82–90. [Online]. Available: http://proceedings.mlr.press/v32/pinheiro14.html

[43] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.



Xiaofei Yang received the B.Sc. degrees from Suihua University, Suihua, China, in 2007 and 2011, respectively, and the M.Sc. degrees from the Harbin Institute of Technology, Harbin, China, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree with the Shenzhen Graduate School.

His research interests include semisupervised learning, deep learning, remote sensing, transfer learning, and graph mining.

Yunming Ye received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China.

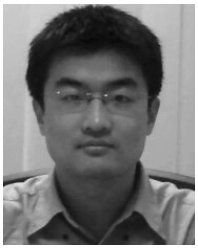
He is currently a Professor with the Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China. His research interests include data mining, text mining, and ensemble learning algorithms.

Xutao Li received the bachelor's degree from the Lanzhou University of Technology, Lanzhou, China, in 2007, and the master's and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2009 and 2013, respectively.

He is currently an Associate Professor with the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, machine learning, graph mining, and social network analysis, especially tensor-based learning and mining algorithms.

Raymond Y. K. Lau (SM'08) is currently an Associate Professor with the Department of Information Systems, City University of Hong Kong, Hong Kong. He has authored 200 refereed international journals and conference papers. His research work has been published in renowned journals, such as *Management Information System Quarterly (MIS)*, the *INFORMS Journal on Computing*, the *ACM Transactions on Information Systems*, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE INTERNET COMPUTING, the *Journal of MIS*, and *Decision Support Systems*. His research interests include big data analytics, social media analytics, FinTech, and artificial intelligence for business.

Dr. Lau is a Senior Member of the ACM.



Xiaofeng Zhang received the M.Sc. degree from the Harbin Institute of Technology, Harbin, China, in 1999, and the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2008.

He was with the Research and Development Center, Peking University Founder Group, Beijing, China, and also with the E-Business Technology Institute, The University of Hong Kong, Hong Kong. He is currently an Associate Professor with the Department of Computer Science, Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, machine learning, and graph mining.

Xiaohui Huang received the B.Eng. and master's degrees from Jiangxi Normal University, Nanchang, China, in 2005 and 2008, respectively, and the Ph.D. degree from the Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China, in 2014.

Since 2015, he has been with the School of Information Engineering Department, East China Jiaotong University, Nanchang, where he is currently a Lecturer of computer science. His research interests include clustering, social media analysis, and deep learning.