IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 18, 2019, accepted June 10, 2019, date of publication June 19, 2019, date of current version July 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923776

Deep Neural Network Based Hyperspectral Pixel Classification With Factorized Spectral-Spatial Feature Representation

JINGZHOU CHEN[1], SIYU CHEN[1], PEILIN ZHOU[2], AND YUNTAO QIAN[1], (Member, IEEE)
[1]Institute of Artificial Intelligence, College of Computer Science, Zhejiang University, Hangzhou 310027, China
[2]Huawei Company, Shanghai, China

Corresponding author: Yuntao Qian (ytqian@zju.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0505000, and in part by the National Natural Science Foundation of China under Grant 61571393.

ABSTRACT Deep learning has been widely used for hyperspectral pixel classification due to its ability to generate deep feature representation. However, how to construct an efficient and powerful network suitable for hyperspectral data is still under exploration. In this paper, a novel neural network model is designed for taking full advantage of the spectral-spatial structure of hyperspectral data. First, we extract pixel-based intrinsic features from rich yet redundant spectral bands by a subnetwork with the supervised pre-training scheme. Second, in order to utilize the local spatial correlation among pixels, we share the previous subnetwork as a spectral feature extractor for each pixel in a patch of the image, after which the spectral features of all pixels in a patch are combined and fed into the subsequent classification subnetwork. Finally, the whole network is further fine-tuned to improve its classification performance. Especially, the spectral-spatial factorization scheme is applied in our model architecture, making the network size and the number of parameters great less than the existing spectral-spatial deep networks for hyperspectral image classification. Compared with other state-of-the-art deep learning methods, experiments on the hyperspectral data sets show that our method achieves 0.18%–7.6%, 0.1%–3.58%, and 0.21%–3.09% improvement on overall accuracy (OA), average accuracy (AA), and kappa, respectively, while having smaller network size and fewer parameters.

INDEX TERMS Hyperspectral pixel classification, deep neural networks, spectral-spatial feature factorization.

I. INTRODUCTION

Hyperspectral imaging has opened up new opportunities for analyzing a variety of materials in remote sensing as it provides rich information on spectral and spatial distributions of distinct materials. One of its most important applications is pixel classification, which is widely applied in material recognition, target detection, geoindexing, and so on [1]–[4]. However, the classification of hyperspectral image (HSI) still faces some challenges such as, the unbalance between the small number of available training samples and the large number of narrow spectral bands, the high variations of the spectral signature from identical material, high similarities of spectral signatures between some different materials, and the noise impact from the sensors and environment [5]. In order to address the above problems, it is necessary to extract robust and discriminant features. The popular spectral feature extraction algorithms include principal component analysis (PCA) [6], independent component analysis (ICA) [7], linear discriminant analysis (LDA) [8], manifold learning [9], [10], and various band selection methods [11]–[13]. In addition, many studies have demonstrated that it is difficult to well distinguish pixels with spectral information alone, hence the spatial-spectral feature extraction attracts more and more attention. A number of joint spectral-spatial features have been proposed such as extended morphological profiles [14] and 3-D discrete wavelet transform (3D-DWT) [15].

Recently, with the great development of modern neural network technique known as deep learning, it has exhibited more beneficial advantages and obtained enormous success in many fields including image segmentation [16], image classification [17], [18], artistic style transfer [19], object

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

detection [20], face verification/identification [21], [22], speech recognition [23] and translation [24]. Comparing to classic statistical methods that explicitly designate fully specified modeling procedures, deep learning tries to fit an implicit yet potentially powerful function that can both imitate bionic mechanism and extract sophisticated features by a data-driven learning process. A number of state-of-the-art deep learning methods have been applied to HSI processing such as unmixing [25], target detection [26], HSI visualization [27], HSI denoising [28], HSI super-resolution [29] and HSI classification.

Among them, the deep neural network based approaches for HSI classification can also be sketchily categorized into spectral based methods and spectral-spatial based methods. The first type of deep learning methods directly use the spectral signatures of pixels for classification, including stacked autoencoder based method [30], pre-training based method [31], and convolutional neural network (CNN) based method [32]. Even though they bring notable improvement of HSI classification over conventional classification techniques such ask-nearest-neighbor (KNN) and support vector machine (SVM) [33], some pixels are still difficult to be accurately classified using the spectral information alone due to the high inter-class similarity and the high intra-class difference. On the other hand, deep-learning approaches based on spectral-spatial information have been proved to achieve better performance than those spectral information based ones. The examples are spatial stacked autoencoder (SAE) based method [34], CNN based spectral-spatial feature extraction methods [35], [36], spectral-spatial CNN based classifiers [37]–[39]. Particularly, three-dimensional CNN (3D-CNN) [40] is a typical model used in most of these spectral-spatial information based deep-learning approaches. 3D-CNN modifies standard CNN to convolve along both spatial and spectral dimensions for HSI classification. Such a scheme can always employ local spatial information in HSI patches to perform classification learning and inference. However, compared with the one-dimensional (1D) and two-dimensional (2D) CNN, 3D-CNN requires a greater size of model to capture useful features as the number and size of kernels grow rapidly in respect to input size and dimension, especially when the spatial dimension and spectral dimension are distinct in terms of physical mechanism. In general, 3D-CNN tends to have excessive parameters, so as to be prone to over-fit and hard to train. As has been suggested by He *et al.* [41], the oversized network may encounter certain approximation difficulties. Han *et al.* also pointed out in [42] that the weights of CNN may be redundant and most of them do not carry significant information. Li *et al.* [43] showed a similar fact that prevailing deep models suffer from weight redundancy problem. Zhang *et al.* [44] also indicated that the oversized models with the excessive amount of parameters often tend to memorize data sets instead of learning the general task solutions. Furthermore, on account of the limited resources of class-labeled pixels in hyperspectral datasets, this problem will cause severer degradation of classification performance. Therefore, reducing the size of the model becomes a significant problem for spectral-spatial CNN based HSI pixel classification methods.

Besides the routine techniques of model compression for deep learning models such as normalization, regularization, and network pruning, operation factorization is popularly used in CNN and other deep neural networks to decouple a complex computation into many much smaller steps which have far fewer parameters in total [45], [46]. Such as in [47], a convolutional layer with 73728 parameters are factorized into two separate operations: one depth-wise and one point-wise convolution, there are totally $576 + 8192 = 8768$ parameters. In [48], $n \times n$ convolution is factorized into a $1 \times n$ convolution followed by a $n \times 1$ convolution. The decoupled/factorized operations can improve the generalization capability, consume fewer resources, and make training and inference faster. Therefore, operation factorization is also one of the prevailing as well as practical schemes applied in spectral-spatial deep learning models for HSI classification. For example, in [49], CNN with pixel-pair features (CNN-PPF) is based on the similarity of local constituents, which takes a pair of pixels as input, and the output tells if these two pixels are of different classes or which class they all belong to. The spectral-spatial feature in CNN-PPF is factorized into class related latent spectral feature and pixel-pair consistency based spatial feature. The final label of the target pixel is determined via a voting strategy based on the neighboring pixel-pair information. CNN-PPF performs well using its augmented data and L_2 regularization scheme. Similarly, in [50], Siamese CNN (S-CNN) uses paired image patches as training samples, and trains them to classify the central pixels of patches, in which the spectral-spatial feature is factorized into the spectral feature extracted by Siamese network and the spatial information used by an SVM that combines the spectral feature vectors of patches outputted by Siamese network. Unfortunately, the combination schemes of the spectral and spatial operations in both of CNN-PPF and S-CNN are fixed rather than adaptively learned, as the voting strategy is used by CNN-PPF, and a separated SVM is used in S-CNN. Therefore an end-to-end CNN with global optimization cannot be achieved.

In this paper, we propose a deep neural network that utilizes both spectral and spatial information in a novel decoupled/factorized manner, which is designed to be concise, adaptive, end-to-end and easy to train. Our spectral-spatial classification model is factorized into a pixel-based spectral feature extraction subnetwork (SFE-Net) and a patch based spatial classification subnetwork (PSC-Net). The SFE-Net is pre-trained in a supervised learning scheme, as a rough prediction based on spectral information alone, and it can be constructed like any spectrum-based deep neural network. In order to refine our prediction, instead of adopting prior knowledge as majority voting used in CNN-PPF, we prefer a data-driven method. The PSC-Net is then trained to determine the class of the central pixel in a patch by concatenating the prior class probability vectors of surrounding pixels into a

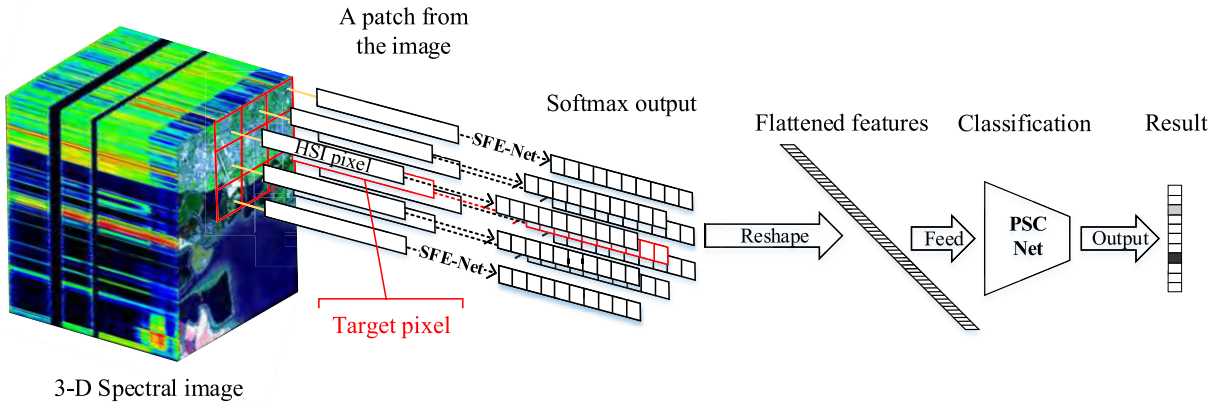


FIGURE 1. Detailed workflow of FSSF-Net for predicting the target pixel.

vector as its input since there are strong spatial relationships among these neighboring pixels. These two subnetworks are combined via a sequential connection. We first share the SFE-Net across each pixel in the patch, then concatenate all the extracted spectral features as the input vector of the PSC-Net. Besides, two factorized subnetworks are trained collaboratively as an entire network by the back-propagation algorithm to improve the overall performance. Because both of SFE-Net and PSC-Net used in the paper are based on deep neural networks, and spectral-spatial features are factorially represented, we name the proposed model *Factorized Spectral-Spatial Feature Network* (FSSF-Net). A demonstration of the proposed framework is shown in Fig. 1.

In summary, a novel deep learning based framework is proposed in the paper that better utilizes the joint spectral-spatial structure of HSI by a flexible and generalized feature factorization scheme. It allows various kinds of deep neural networks to act as subnetworks or hidden layers. Meanwhile, it is an integrated end-to-end model. Compared to some state-of-the-art deep learning based HSI classification methods including CNN based ones, the proposed approach achieves competitive classification performance. Furthermore, the light-weight model resulting from feature factorization brings faster training, lower storage, and better generalization with a small number of class-labeled samples.

The paper is organized as follows. In Section II, we introduce the proposed model, followed by the details of the implementation. In Section III, a number of experiments are made to investigate the effectiveness of the proposed model and the primary techniques used in it, and experimental comparisons with some state-of-the-art approaches are also given. Finally, we conclude our work in Section IV.

II. METHODS

In this section, we will explain some details of our network including the specific network architecture and the training scheme. We first highlight the detail of two subnetworks, SFE-Net and PSC-Net, and how they are connected together. As one of the major concerns, we clarify our training scheme, especially how the error is back propagated through PSC-Net to SFE-Net.

A. MODEL ARCHITECTURE

FSSF-Net takes a 3D patch extracted from an HSI as input. As aforementioned, there are two parts in FSSF-Net: SFE-Net and PSC-Net. Given a 3D patch, we first apply and share the SFE-Net across each pixel to extract spectral features, then concatenate these spectral features into a vector to fuse spatial information embodied in the patch. Utilizing this vector, PSC-Net infers the class label of the central pixel.

Generally, there is no restriction on the structure and the number of layers in these two subnetworks. SFE-Net and PSC-Net are assumed to be any type of neural networks, i.e., they are not limited to the specific form. The flexibility of our framework allows us to choose and/or develop any network structure in order to maximize the classification performance. In this paper, we use MLP (Multilayer Perceptron) based architectures for both subnetworks. To improve the generalization capability and advert over-fitting, batch-normalization layers (BN), self-normalizing ELU (SELU) layers and Drop-out layers (Drop-out) are added to the architecture as regularization/normalization components. The network architecture for SFE-Net and PSC-Net are illustrated in Fig. 2, where FC stands for fully-connected layers. For both SFE-Net and PSC-Net, the hidden fully-connected layers have the same amount of output units $u = 100$ and all their Drop-out layers share the same probability of retaining $r = 0.5$. Both SFE-Net and PSC-Net output C-dimensional vector activated by softmax as the final output, where C is the number of classes.

B. MODEL TRAINING

An patch based HSI dataset can be represented as $X = \{x_i | x_i \in \mathbb{R}^{W\times W\times D}, i = 1, 2, \ldots, N\}$, where W is the width of the square HSI patch, D is the the number of spectral bands/channels, N is the number of patches in the data set. The corresponding classification label set is $Y = \{y_i | y_i \in \mathbb{Z}^C, i = 1, 2, \ldots, N\}$ where each element y_i is the one-hot

FIGURE 2. Architectures of SFE-Net and PSC-Net. (a) Architecture of SFE-Net. (b) Architecture of PSC-Net.

label of the central pixel in x_i, and C still is the number of classes. The pixel classification of HSI is to find a function $f(x)$ such that the output $Y' = \{y'_i | y'_i = f(x_i), i = 1, 2, \ldots, N\}$ is as close to classification label Y as possible.

In FSSF-Net, the submodule SFE-Net is pre-trained with classification supervision, so as to provide softmax vectors as output spectral features. Then, the submodule PSC-Net takes these spectral features of an HSI patch and outputs the final classification result, which is also achieved by using a classification loss metric. In summary, both SFE-Net and PSC-Net are trained using *cross-entropy* loss metric. Specifically, given an output vector y' and its corresponding one-hot label y, with the j-th element in vector y indexed by $y^{(j)}$, the loss function can be formulated as

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_i^{(j)} \log(y_i'^{(j)}). \tag{1}$$

The training of FSSF-Net includes two stages: pre-training SFE-Net, and fine-tuning SFE-Net, PSC-Net together. In the pre-training stage, we extract the central pixels in each patch x_i and train SFE-Net with corresponding labels. As for the fine-tuning stage, FSSF-Net uses 3D patches x_i as input, whose label is provided by the central pixel. During this stage, both SFE-Net and PCS-Net are set to be trainable. The entire training process completes when the fine-tuning training of FSSF-Net converges. Especially, when back-propagating the error through PSC-Net to SFE-net, the error used for updating SFE-Net is the average of errors calculated on each pixel in a patch. ADAM (Adaptive Moment Estimation [51]) optimizer is used in our experiments with a learning rate set to 0.001 in both stages. The input patch size W is set to 7, which is a tradeoff between the workload of training storage and the amount of spatial information fed into the network. In addition, the patches extracted for training may overlap, which depends on the distribution of the labeled pixels used for training. The number of extracted patches for training in each data set is the same as the number of labeled pixels for training. As all pixels in the patches but the centered pixel only provides their spectral signature features in both training and test procedures, whether overlapping does not affect our method. Overall, there are four hyper-parameters C, W, u, r in our implementation where W, u, r are constant through all our experiments.

III. EXPERIMENTS

In this section, we first introduce five data sets used in our experiments and clarify corresponding experimental settings. Then, three primary ideas applied in our method are examined: the pre-training of SFE-Net, the sharing of such spectrum extractor spatially, and the flexibility of types of hidden layers. Finally, we evaluate our method further by comparing with some state-of-the-art approaches. Tensorflow based on CUDA library is selected as the computational framework, and the unified interface wrapper Keras is applied to simplify implementation. All experiments are running on a workstation equipped with an Intel Xeon E5-2620 v4 with 2.1 GHz and Nvidia GeForce GTX 1080 graphics card. OA, AA, and kappa are used as the criteria of classification accuracy. All experiments are repeated five times.

A. DATA FOR EXPERIMENTS[1]

The first is Indian Pines data set gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1992 from Northwest Indiana, including 16 vegetation classes. There are 220 spectral bands in the 0.4-45 μm range of the visible and infrared spectrum and have 145 $\times$ 145 pixels for each band. The pseudocolor image of the Indian Pines data set is shown in Fig. 3.

The second data set Salinas is also collected by the AVIRIS sensor over Salinas Valley, California, including 512 $\times$ 217 pixels for each band. After 20 water absorption bands are removed, 204 bands remain for the experiments. There are 16 classes contained in the image such as vegetables, bare soils, and vineyard fields. Its pseudocolor image is illustrated in Fig. 4.

The Third is KSC dataset still acquired by the AVIRIS sensor over the Kennedy Space Center (KSC), Florida, in 1996. After removing water absorption and low SNR bands, 176 bands were used. There are 13 land-cover classes and 512$\times$614 pixels with 5211 labeled. Its pseudocolor image is shown in Fig. 4.

The fourth is the University of Pavia data set acquired by Reflective Optics System Imaging Spectrometer (ROSIS) in Northern Italy in 2001. The image scene contains 9 urban land-cover types and 610 $\times$ 340 pixels for each band.

[1] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

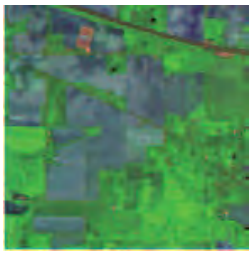


FIGURE 3. Pseudocolor image (bands 5, 50, 200) of indian pines.

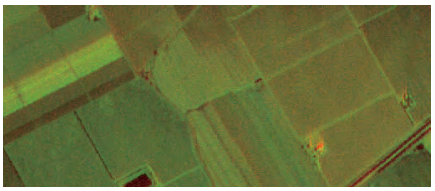


FIGURE 4. Pseudocolor image (bands 1, 102, 204) of salinas.

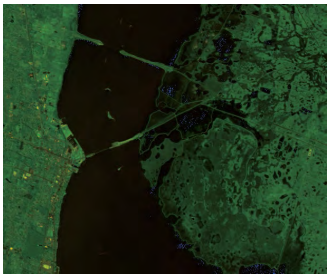


FIGURE 5. Pseudocolor image (bands 5, 50, 150) of KSC.

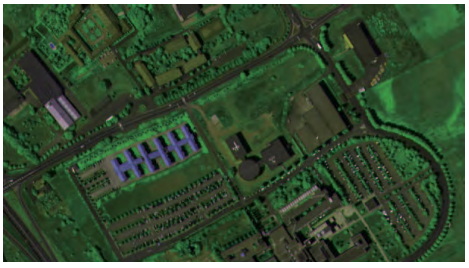


FIGURE 6. Pseudocolor image (bands 10, 50, 80) of pavia university.

After the very noisy bands have been removed, the remaining 103 spectral bands, in the 0.43-0.86 μm range of the visible and infrared spectrum, are employed. Its pseudocolor image is shown in Fig. 6.

The fifth is the Pavia Center data set also acquired by the ROSIS sensor over Pavia, northern Italy. After removing noise bands, there are 102 spectral bands left. The original size of Pavia Center is 1096×1096, but some of the samples in the image contain no information and have to be discarded, thus a subset with a size of 1096×715 is selected, which includes 9 different classes. Its pseudocolor image can be seen in Fig. 7.

B. MODEL SETTING

The proposed architecture is implemented as stated in Section. II. The spectral dimension may differ in the input

FIGURE 7. Pseudocolor image (bands 1, 30, 80) of pavia center.

layer of SFE-Net with regard to different data sets. The number of neurons in the last softmax layer of both SFE-Net and PSC-Net may also vary corresponding to the number of classes (C) in each data set.

Besides, the optimization process is also a very important factor influencing the classification performance. We adopt the ADAM optimizer to harness and speed up the training process. The initial learning rate is set to 0.001 for both pre-training and fine-tuning. The decay of learning rate is set to 0.005 for the pre-training, and 0.01 for the fine-tuning. Considering the limited training data in our task, we are able to set the batch size to be equal to the whole training set for better gradient behavior. The pre-training and fine-tuning take 10000 epochs and 1000 epochs to converge respectively.

Last but not least, it is worth noting that we fix all the hyperparameters mentioned above in all experiments, which means that we do not tune our model to overfit any specific data sets. The augmentation of training samples for pixel classification in hyperspectral images is not easy, and the main end of our paper is to show the effectiveness and power of the proposed model in the case of the small size of the training set, so the data augmentation technique is not used. However, it can be considered as an interesting topic for our future work.

C. EFFECTIVENESS OF PROPOSED MODEL

In this section, we will evaluate the effectiveness of three primary ideas applied in our proposed architecture, which are the supervised pre-training of SFE-Net for extracting spectral feature, the sharing of such feature extractor spatially to include the local spatial correlation into PSC-Net, and the flexibility of network type for hidden layers. In addition, during the fine-tuning stage, the whole FSSF-Net is trained, including the pre-trained network SFE-Net. We examine whether these three schemes in the design of the architecture of our model actually contribute to improving accuracy in the following experiments.

1) SUPERVISED PRE-TRAINING

To verify the necessity of pre-training, we compare the model that the parameters in the SFE-Net are initialized by the pre-training process with the model whose parameters of the SFE-Net are randomly initialized. In other words, the structure of the whole model FSSF-Net remains the same, and only the parameter initializations in the SFE-Net differ.

TABLE 1. Separation of training samples for pavia university data set.

Number	Class	Samples	Training Samples
1	Asphalt	6631	548
2	Meadows	18649	540
3	Gravel	2099	392
4	Trees	3064	524
5	Painted metal sheets	1345	265
6	Bare Soil	5029	532
7	Bitumen	1330	375
8	Self-Blocking Bricks	3682	514
9	Shadows	947	231

TABLE 2. The classification results with and without pre-training (%).

Dataset		Pavia University		Indian Pines	
Pre-training	Metrics	1%	10%	5%	10%
Yes	OA	**73.51**	**97.21**	**95.03**	**98.08**
	AA	**84.07**	**97.69**	**82.64**	**93.86**
	Kappa	**67.54**	**96.32**	**94.33**	**97.81**
No	OA	68.21	96.06	89.92	95.47
	AA	74.21	96.57	75.58	88.24
	Kappa	60.32	94.80	88.47	94.84

The University of Pavia and the Indian Pines data sets are chosen to perform such comparison, and the training sets are selected with the small and medium sizes. For the Indian Pines data set, we randomly selected 5% and 10% of the labeled samples from each class to form the training sets respectively, leaving the rest as the test sets. As for the Pavia University data set, the training samples have been separated from the total labeled ones, see Table 1. We randomly selected 1% and 10% samples from each class in the separated training samples as the training sets, and all the remaining labeled samples are used as test sets. The compared results are displayed in Table 2, from which we can observe that both in the Pavia University and in the Indian Pines, the supervised pre-training can significantly improve the classification accuracy, especially in the case of small training-sample size.

2) SHARING OF PRE-TRAINING NETWORK

Besides the pre-training scheme, parameter sharing is also a popular strategy embodied in a variety of deep learning models. For example, CNN extracts shifting invariant features from images by the convolutional operation, and recurrent neural network (RNN) models the sequential dependency by designing the network with loops in them. Different from these sharing methods in the scenario of CNN and RNN, we assume that the adjacent pixels around a central pixel share similar spectral information, therefore it is natural to use the same pre-trained SFE-Net to extract spectral features from each pixel belonging to a $7 \times 7 \times D$ patch. To verify the effectiveness of this sharing architecture, we set up a contrastive experiment. First, we use the same SFE-Net to extract each pixel's spectral features in a patch. Second, on the contrary, for different pixels in a patch, different networks are used, which have the same structure as the SFE-Net but do not share their parameters. In the latter case, the parameters of those networks are still initialized with the pre-trained parameters, but during the fine-tuning process, those networks for different pixels in a patch may learn different parameters.

TABLE 3. The classification results with and without sharing scheme (%).

Dataset		Pavia University		Indian Pines	
Parameter Sharing	Metrics	1%	10%	5%	10%
Yes	OA	**73.51**	**97.21**	**95.03**	**98.08**
	AA	**84.07**	**97.69**	**82.64**	**93.86**
	Kappa	**67.54**	**96.32**	**94.33**	**97.81**
No	OA	72.57	93.86	80.40	86.51
	AA	74.53	94.39	67.74	77.40
	Kappa	64.54	91.89	77.47	84.53

TABLE 4. The classification results with different types of hidden layers in sfe-net (%).

Dataset	Training	Metrics	MLP	CNN^1	CNN^2
Indian Pines	Spectral only	OA	71.08	56.80	**83.38**
		AA	76.99	60.83	**87.57**
		Kappa	66.53	49.75	**80.70**
	Spectral-spatial	OA	93.50	77.29	**95.19**
		AA	95.80	83.55	**96.87**
		Kappa	92.39	73.68	**94.37**
Salinas	Spectral only	OA	**90.00**	81.25	88.96
		AA	**95.20**	87.48	94.59
		Kappa	**88.87**	79.24	87.74
	Spectral-spatial	OA	94.16	91.13	**96.06**
		AA	97.77	96.30	**98.37**
		Kappa	93.50	90.18	**95.62**
Pavia University	Spectral only	OA	81.37	76.09	**82.72**
		AA	85.40	84.05	**87.69**
		Kappa	75.99	69.82	**77.78**
	Spectral-spatial	OA	**96.97**	96.72	95.60
		AA	97.77	**98.04**	97.50
		Kappa	**96.00**	95.68	94.23
KSC	Spectral only	OA	92.94	81.27	**93.66**
		AA	91.35	75.73	**92.31**
		Kappa	92.15	79.16	**92.95**
	Spectral-spatial	OA	**99.90**	98.90	**99.90**
		AA	99.91	98.47	**99.92**
		Kappa	**99.89**	98.77	**99.89**

We apply the same experimental setting in Section III-C.1 and the classification results are displayed in Table 3. Seen from Table 3, classification accuracy has been apparently boosted due to applying the sharing scheme. The improvement with sharing scheme in the Indian Pines data set is more obvious than that in the Pavia University data set because the local spatial correlation is stronger in the Indian Pines data set than in the Pavia University data set. In addition, since the whole network has less number of parameters with such sharing scheme, it is less likely to overfit.

3) FLEXIBILITY OF NETWORK TYPE IN HIDDEN LAYERS

When applying the proposed framework to build our classification network, we adopt MLP as hidden layers. Specifically, the hidden layers in SFE-Net are FC together with dropout and batch normalization layers, see Fig. 2(a), and the hidden layers in PSC-Net are FC with dropout layer, see Fig. 2(b)). Indeed, network type used to implement this architecture is flexible and the main contribution of this paper is proposing a network architecture suitable to fuse the spatial-spectral

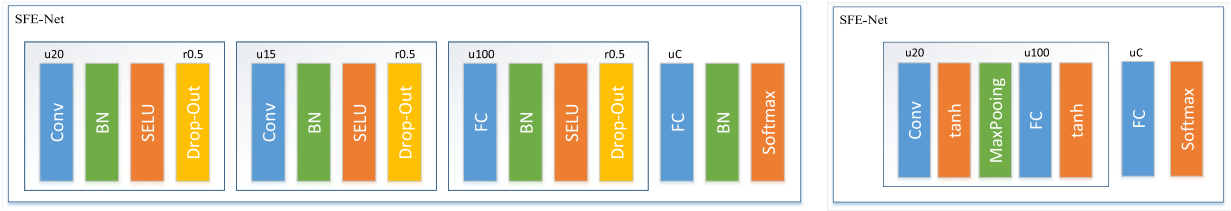


FIGURE 8. Architectures of CNN[1] and CNN[2]. (a) Architecture of CNN[1]. (b) Architecture of CNN[2].

TABLE 5. Experimental setting of indian pines in setting 2.

Number	Class	Training	Test
1	Alfalfa	30	16
2	Corn-notill	150	1198
3	Corn-min	150	232
4	Corn	100	5
5	Grass-pasture	150	139
6	Grass-trees	150	580
7	Grass-pasture-mowed	20	8
8	Hay-windrowed	150	130
9	Oats	15	5
10	Soybean-notill	150	675
11	Soybean-mintill	150	2032
12	Soybean-clean	150	263
13	Wheat	150	55
14	Woods	150	793
15	Buildings-Grass-Trees	50	49
16	Stone-Steel-Towers	50	43
	Total	1765	6223

information in HSI. To explore the flexibility of the proposed architecture, besides the aforementioned MLP architecture (see Fig. 2(a)), one CNN architecture (annotated as CNN[1] in Fig. 8(a)) and another CNN architecture (annotated as CNN[2] in Fig. 8(b)) used in [32] are adopted as SFE-Net respectively. In Fig. 8, convolutional layer is abbreviated as Conv and tanh represents hyperbolic tangent activation function. In fact, FC layer in the SFE-Net can be viewed as a convolutional layer with kernel size $(1, 1, D)$ and stride size $(1, 1, 0)$, and the number of convolutional kernels equals the number of neurons in the FC layer.

Indian Pines, Salinas, Pavia University, and KSC datasets were used in the experiment, 50 samples were randomly selected from each class as training samples and the compared results are displayed in Table 4, in which spectral only model is just SFE-Net used for classification, and the spectral-spatial model is the whole network including SFE-Net and PSC-Net. We can find when taking adjacent spatial information into consideration, the classification performance can be further improved whichever type of hidden layers is used. Moreover, different architectures all achieve competitive results when spatial-spectral information is used, which verifies the flexibility of our high-level architecture. Observed from Table 4, in general, CNN[2] has better performance than CNN[1], which may result from that CNN[2] has simpler architecture and less number of parameters than CNN[1], so CNN[2] is more suitable for the case of small size of training set. This experiment also shows that evaluating MLP and CNN is very difficult. To remain simplicity, we adopt MLP based architecture in the following experiments.

TABLE 6. Experimental setting of ksc in setting 2.

Number	Class	Training	Test
1	Scrub	33	314
2	Willow swamp	23	220
3	CP hammock	24	232
4	Slash pine	24	228
5	Oak/Broadleaf	15	146
6	Hardwood	22	207
7	Swamp	9	96
8	Graminoid marsh	38	352
9	Spartina marsh	51	469
10	Cattail marsh	39	365
11	Salt marsh	41	378
12	Mud flats	49	454
13	Water	91	836
	Total	459	4297

TABLE 7. Experimental setting of pavia university in setting 2.

Number	Class	Training	Test
1	Asphalt	548	5472
2	Meadows	540	13750
3	Gravel	392	1331
4	Trees	542	2573
5	Metal Sheets	256	1122
6	Bare soil	532	4572
7	Bitumen	375	981
8	Bricks	514	3363
9	Shadows	231	776
	Total	3930	33940

D. COMPARISON WITH OTHER METHODS

To further demonstrate the effectiveness of the proposed method, we compare it with some traditional prominent methods such as spectral feature based SVM, 3D-DWT, spectral-spatial feature based SVM, and some state-of-the-art deep learning methods, like CNN [32], 3D-CNN, CNN-PPF, S-CNN. In the experiments, SVM has radial basis function (RBF) kernel, and is implemented by the *libsvm* toolbox. As described earlier, our proposed model share similar ideas with some CNN models, therefore, we compare our method with several CNN based methods. CNN [32] applies the convolutional operation on hyperspectral image classification but it only considers the spectral information. 3D-CNN takes a 3D patch as input and convolves it with 3D convolutional kernels. CNN-PPF and S-CNN adopt data augmentation by pairing similar pixels.

TABLE 8. Classification results (%) under deffirent expetimental settings.

Experimental Setting	Data Set	Metrics	CNN	3D-CNN	CNN-PPF	S-CNN	FSSF-Net
Setting 1	Salinas	OA	92.60		94.80		**95.98**
		AA					98.68
		Kappa					95.52
	Indian Pines	OA	90.16		94.34		**97.76**
		AA					98.88
		Kappa					97.32
	Pavia University	OA	92.56		96.48		**99.31**
		AA					99.17
		Kappa					99.07
Setting 2	KSC	OA		96.31			**99.09**
		AA		94.68			**98.26**
		Kappa		95.90			**98.99**
	Indian Pines	OA		97.56			**98.01**
		AA		**99.23**			99.07
		Kappa		97.02			**97.90**
	Pavia University	OA		99.54			**99.72**
		AA		**99.66**			99.62
		Kappa		99.41			**99.62**
Setting 3	Pavia Center	OA				99.68	**99.85**
		AA				99.26	**99.70**
		Kappa				99.55	**99.79**
	Indian Pines	OA				**99.04**	98.81
		AA				99.14	**99.31**
		Kappa				**98.87**	98.60
	Pavia University	OA				99.08	**99.26**
		AA				99.08	**99.18**
		Kappa				98.79	**99.03**

TABLE 9. Network complexities of deeping learning models on deffirent data sets.

Dataset	CNN	3D-CNN	CNN-PPF	S-CNN	FSSF-Net
Salinas	82,216		178,278		123,696
Indian Pines	81,408	45,331,024	65,019	3,331,900	125,296
Pavia University	61,249	5,860,841	33,019	2,207,480	77,854
Kennedy Space Center		5,987,437			105,578
Pavia Center				2,206,940	77,754

It is difficult to make a fair comparison between various deep learning methods because there are many hyperparameters needed to be adjusted properly. Therefore, firstly we compare our method to each one of those deep learning methods (CNN, 3D-CNN, CNN-PPF, S-CNN) under the same setting as they are stated in their original papers, and the classification results of these four methods are directly copied from their papers since the reported results are obtained with the optimal or suboptimal structures and parameters tuned by those authors. According to the papers of CNN and CNN-PPF, Salinas, Indian Pines, and Pavia University are used as experimental data sets, and we denote their experimental setting as ''setting 1'' where 200 samples are randomly selected from each class as the training samples and leave the rest as the test samples. The ''setting 2'' refers to the experimental settings in [50], in which the separations of training and testing samples on Kennedy Space Center, Indian Pines, and Pavia University are illustrated in Tables 5, 6, 7 respectively. The ''setting 3'' is based on the paper of S-CNN, in which 200 samples are randomly selected from each class as training samples in Pavia Center, Indian Pines, and Pavia University data sets, whereas the whole labeled samples in each data set are used as test samples.

For all three ''settings'', the results of the proposed FSSF-Net method are obtained with the aforementioned architecture and parameters in Section III-B. The results of all five deep learning methods in three ''settings'' are gathered in Table 8. In addition to classification accuracy, we also evaluate the complexities of these models by calculating their numbers of parameters for different data sets, which are recorded in Table 9. Overall speaking, observed from Table 8 and Table 9, our method reaches a good tradeoff between the model complexity and the classification accuracy. CNN and CNN-PPF have fewer parameters than ours, but 3D-CNN and S-CNN contain much more parameters than ours. Our method obtains better results than those of CNN and CNN-PPF, and outperforms 3D-CNN and S-CNN in most cases.

The next experiment is to evaluate the HSI classification methods in the situation of highly limited training samples. As we know, the deep learning technique usually needs a significant amount of labeled data for training. However, due to the limit of available labeled data in HSI, it is necessary to evaluate the deep learning methods using limited training samples. Four spectral-spatial deep learning models 3D-CNN, CNN-PPF, S-CNN, and FSSF-Net are evaluated, and we also choose two prominent traditional methods, SVM,

TABLE 10. Classification results (%) OF with 50 training samples of each class in indian pines.

Class	SVM	3D-DWT	3D-CNN	CNN-PPF	S-CNN	FSSF-Net
1	48.72	79.74	77.39	80.19	18.07	92.45
2	45.74	86.72	85.00	91.92	42.82	98.74
3	81.20	97.41	92.98	97.23	86.14	98.80
4	95.18	96.82	96.53	99.56	97.21	99.65
5	96.26	99.72	99.07	99.77	97.66	100
6	58.35	83.95	83.60	88.07	24.62	92.23
7	52.67	77.09	67.05	76.72	53.35	85.61
8	59.04	88.73	90.53	92.82	67.96	95.10
9	87.59	98.57	94.81	99.67	93.58	99.59
OA	64.08	86.40	82.42	87.88	57.50	**93.50**
AA	69.42	89.86	87.44	91.77	64.60	**95.80**
Kappa	58.33	84.10	79.59	85.84	51.09	**92.39**

TABLE 11. Classification results (%) with 50 training samples of each class in salinas.

Class	SVM	3D-DWT	3D-CNN	CNN-PPF	S-CNN	FSSF-Net
1	98.82	98.39	98.42	99.80	99.95	100
2	99.31	97.67	92.86	99.54	53.56	100
3	97.90	97.93	97.59	99.79	59.71	100
4	99.38	98.78	99.87	99.85	100	99.97
5	97.05	98.93	99.29	95.51	99.39	99.39
6	99.52	99.37	99.22	99.69	99.87	100
7	99.42	98.16	94.66	99.77	97.00	99.99
8	64.11	79.29	69.95	89.15	74.32	81.23
9	98.69	97.71	95.37	98.68	96.86	100
10	89.23	92.43	97.68	93.25	79.18	97.22
11	97.31	99.61	98.45	99.31	96.66	99.69
12	99.40	99.98	98.87	100	100	100
13	97.64	97.74	99.31	99.31	99.88	99.95
14	94.59	96.08	98.86	96.67	96.08	99.76
15	71.17	81.17	84.13	68.36	73.75	87.71
16	98.43	98.45	96.98	98.98	97.61	99.44
OA	87.08	91.65	89.57	92.33	84.30	**94.16**
AA	93.87	95.73	95.09	96.10	88.99	**97.77**
Kappa	85.65	90.72	88.45	91.57	82.56	**93.50**

and 3D-DWT, so as to contrast with these deep learning methods. In our experiments, we randomly select 50 samples from each class to form our training set, leaving the rest as the test set. Salinas, Indian Pines, and Pavia University are used as our experimental data sets. Especially, as for Indian Pines, because some classes have fewer than 50 labeled samples, we only choose 9 classes that contain more than 400 samples for classification. The experimental setting of our method remains the same as stated in Section III-B, and for the other methods their settings follow their original papers.

The compared results are listed in Tables 10, 11, 12. Compared with the classification results in Table 8, all deep learning methods suffer performance degradation in different degrees. However, our method consistently outperforms other ones, indicating that our method can better deal with limited training sets by taking full advantage of the spectral and spatial properties of HSI with well-designed network architecture. As the size of the training set decreases, there is more impact on 3D-CNN and S-CNN than CNN-PPF, because CNN-PPF has fewer parameters and is less likely to over-fit. Surprisingly, 3D-DWT achieves relatively competitive results when compared with deep learning methods.

We further drill down to the details of these deep learning methods and compare with our model. To deal with limited samples in HSI, CNN-PPF and S-CNN use data augmentation

TABLE 12. Classification results (%) with 50 training samples of each class in pavia university.

Class	SVM	3D-DWT	3D-CNN	CNN-PPF	S-CNN	FSSF-Net
1	75.50	91.72	84.00	95.40	86.22	95.22
2	79.88	93.74	94.23	87.08	95.37	97.14
3	77.99	84.56	84.19	90.24	76.28	90.39
4	92.49	95.57	97.31	92.80	97.84	95.98
5	99.55	99.60	99.74	99.92	100	100
6	80.37	90.83	88.93	92.70	82.95	93.83
7	91.84	96.59	92.92	94.38	88.59	100
8	79.53	91.87	91.21	87.94	77.73	95.79
9	99.60	99.89	97.61	99.22	99.55	99.84
OA	81.42	93.01	91.69	90.54	90.25	**96.16**
AA	86.31	93.82	92.24	93.30	89.39	**96.47**
Kappa	76.10	90.79	89.06	87.67	87.06	**94.91**

TABLE 13. Execution time (second) of training and testing procedures.

Methods	Procedure	Univeristy of Pavia	Indian Pines	Salinas
3D-CNN	training	1498	5387	9130
	testing	106	82	568
CNN-PPF	training	726	952	2431
	testing	14	4	18
S-CNN	training	120	558	795
	testing	2	1	2
FSSF-Net	training	357	370	401
	testing	65	17	84

technique, in which each pixel is combined with other pixels belonging to the same or different classes to form the paired training samples. In our model, we merge spectral and spatial properties of HSI into the network structure, so it also plays a role of data augmentation but by a new way in which the central pixel in a patch is enhanced by the neighboring pixels with or without labels. However, both CNN-PPF and S-CNN are feature extractors and adopt separated classifiers to classify the extracted features. On the contrary, our model is end-to-end, which is not only easy to implement but also able to improve the performance of feature extraction and classification as well. 3D-CNN also takes a patch in HSI, but it does not have the sharing scheme, so it contains more parameters, making it perform inferior to our model in the case of small training sample size.

As a supplement, we further record the training and inference time in Table 13, from which it can be found that our method converges faster than 3D-CNN, CNN-PPF, and S-CNN, but costs more time than CNN-PPF and S-CNN to inference. The reason may be that in our method, the pre-trained SFE-Net needs to process each pixel in a patch individually whereas it can be implemented using 1D-CNN for better parallelization. Moreover, for better visualization of classification performance, classification maps of our method on three data sets are illustrated in Figs. 9, 10, 11 respectively. Comparing with the corresponding ground truth, our results are quite close to the real ones except failing at some verges of certain patches in the classification maps.

In order to evaluate the performance of our method in the case of a larger proportion of the labeled samples as the training set, we split the labeled samples as 60% for training and 40% for testing. As shown in Table 14, our

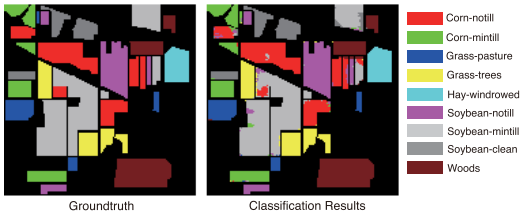


FIGURE 9. Classification map of indian pines.

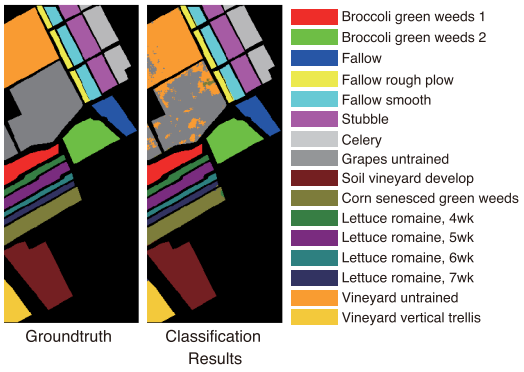


FIGURE 10. Classification map of salinas.

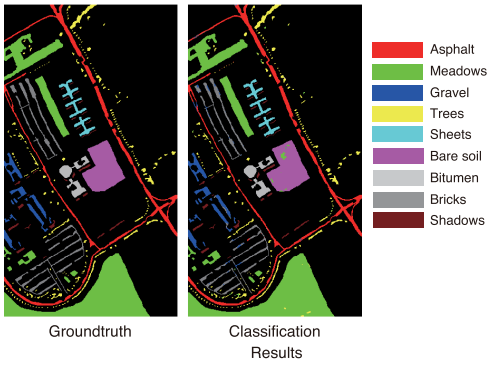


FIGURE 11. Classification map of pavia university.

TABLE 14. The classification results on different data sets with 60% labeled samples as training set.

Metrics	Indian Pines	Salinas	Pavia University	KSC
OA	99.81	99.55	99.61	99.996
AA	99.86	99.85	99.55	99.994
Kappa	99.79	99.50	99.49	99.996

method exceeds 99% accuracy on four data sets, which proves that more training samples can learn all the varieties of the existing features and have better generalization.

IV. CONCLUSION

In this paper, a novel end-to-end factorized architecture is proposed, motivated by the spectral-spatial structure of hyperspectral images. The proposed method factorizes spectral-spatial feature learning by two sequential sub-networks: SFE-Net and PSC-Net. SFE-Net is shared across pixels in a patch to extract spectral features from rich yet redundant spectral signatures, and PSC-Net is trained to learn local spatial information in a patch. Some popular and effective techniques for DNN are involved in our method. Pre-training is used to boost the training efficiency of SFE-Net, which enables the model to learn the inherent spectral structure from the individual labeled pixel. The end-to-end fine-tuning is introduced to coordinate SFE-Net and PSC-Net modules, which learns the joint spectral-spatial structure for classification. Furthermore, the factorization scheme can downsize the model to avoid over-fitting and small-size-training-sample problem. The experimental results on real hyperspectral data sets demonstrate that the proposed method outperforms state-of-the-art methods in most cases.

REFERENCES

[1] 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.

[2] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[3] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6844–6857, Nov. 2014.

[4] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, 2015.

[5] 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.

[6] P. Bajorski, "Statistical inference in PCA for hyperspectral images," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 438–445, Jun. 2011.

[7] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

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

[9] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Improved manifold coordinate representations of large-scale hyperspectral scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2786–2803, Oct. 2006.

[10] 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.

[11] M. J. Mendenhall and E. Merenyi, "Relevance-based feature extraction for hyperspectral images," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 658–672, Apr. 2008.

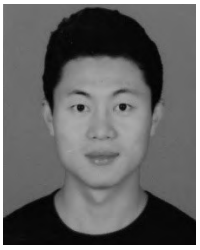
[12] A. MartÍnez-UsÓMartinez-Uso, F. Pla, J. M. Sotoca, and P. GarcÍa-Seville, "Clustering-based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.

[13] J. Feng, L. C. Jiao, X. Zhang, and T. Sun, "Hyperspectral band selection based on trivariate mutual information and clonal selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4092–4105, Jul. 2014.

[14] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

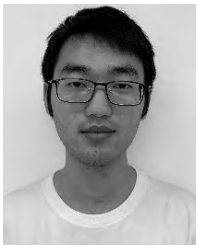
[15] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[16] S. R. Buló, G. Neuhold, and P. Kontschieder, "Loss max-pooling for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7082–7091.
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1106–1114.
[18] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3288–3291.
[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, Jan. 2016, pp. 1–16.
[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580–587.
[21] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, Jun. 2014, pp. 1891–1898.
[23] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2013, pp. 8614–8618.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[25] S. Ozkan and G. B. Akar, "Deep spectral convolution network for hyperspectral unmixing," in *Proc. IEEE Int. Conf. Inf. Process.*, Oct. 2018, pp. 3313–3317.
[26] L. Pan, Q. Zhang, W. Zhang, Y. Sun, P. Hu, and K. Tu, "Detection of cold injury in peaches by hyperspectral reflectance imaging and artificial neural network," *Food Chem.*, vol. 192, pp. 134–141, Feb. 2016.
[27] S. Chen, D. Liao, and Y. Qian, "Spectral image visualization using generative adversarial networks," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2018, pp. 388–401.
[28] W. Xie and Y. Li, "Hyperspectral imagery denoising by deep learning with trainable nonlinearity function," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 1963–1967, Nov. 2017.
[29] Y. Li, J. Hu, X. Zhao, W. Xie, and J. Li, "Hyperspectral image super-resolution using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 29–41, Nov. 2017.
[30] 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.
[31] L. Windrim, A. Melkumyan, R. J. Murphy, A. Chlingaryan, and R. Ramakrishnan, "Pretraining for hyperspectral convolutional neural network classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2798–2810, May 2018.
[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] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.
[34] X. Ma, H. Wang, and J. Geng, "Spectral-spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.
[35] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.
[36] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.
[37] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.
[38] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
[39] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.
[40] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[42] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *Fiber*, vol. 56, no. 4, pp. 3–7, 2016.
[43] G. Li, J. Ye, H. Yang, D. Chen, S. Yan, and Z. Xu, "BT-Nets: Simplifying deep neural networks via block term decomposition," 2017, *arXiv:1712.05689*. [Online]. Available: https://arxiv.org/abs/1712.05689
[44] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2017, pp. 1–15.
[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.
[46] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," 2017, *arXiv:1709.01507*. [Online]. Available: https://arxiv.org/abs/1709.01507
[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861
[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
[49] 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.
[50] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.
[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2015, pp. 1–15.



JINGZHOU CHEN received the B.E. degree in computer science and technology from Sichuan University, Sichuan, China, in 2016. He is currently pursuing the Ph.D. degree with the College of Computer Science, Zhejiang University, Hangzhou, China. His research interests include machine learning, pattern recognition, and hyperspectral image processing.

SIYU CHEN received the bachelor's degree from the Harbin Institute of Technology, in 2016. He is currently pursuing the master's degree with the College of Computer Science, Zhejiang University, Hangzhou, China. His research interests include statistical machine learning, pattern recognition, and computer vision.



PEILIN ZHOU received the B.E. degree in electrical engineering and automation from Wuhan University, Wuhan, China, in 2015, and the M.E. degree from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2018. He is currently with Huawei Company, Shanghai, China. His research interests include machine learning, pattern recognition, and hyperspectral image processing.

YUNTAO QIAN (M'04) received the B.E. and M.E. degrees in automatic control from Xi'an Jiaotong University, Xi'an, China, in 1989 and 1992, respectively, and the Ph.D. degree in signal processing from Xidian University, Xi'an, in 1996. From 1996 to 1998, he was a Postdoctoral Fellow with Northwestern Polytechnical University, Xi'an. Since 1998, he has been with Zhejiang University, Hangzhou, China, where he is currently a Professor in computer science. He was a Visiting Professor with Concordia University, Montreal, QC, Canada, the Hong Kong Baptist University, Hong Kong, Carnegie Mellon University, Pittsburgh, PA, USA, and the Canberra Research Laboratory, NICTA, Canberra, Australia, from 1999 to 2001, and in 2006 and 2010, respectively. His research interests include machine learning, signal and image processing, and pattern recognition.