Hyperspectral Image Classification With Mixed Link Networks

Zhe Meng, *Member, IEEE*, Licheng Jiao, *Fellow, IEEE*, Miaomiao Liang, and Feng Zhao, *Member, IEEE*

***Abstract*—Convolutional neural networks (CNNs) have improved the accuracy of hyperspectral image (HSI) classification significantly. However, CNN models usually generate a large number of feature maps, which lead to high redundancy and cannot guarantee to effectively extract discriminative features for well characterizing the complex structures of HSIs. In this article, two novel mixed link networks (MLNets) are proposed to enhance the representational ability of CNNs for HSI classification. Specifically, the proposed mixed link architectures integrate the feature reusage property of the residual network and the capability of effective new feature exploration of the densely convolutional network, extracting more discriminative features from HSIs. Compared with the dual path architecture, the proposed mixed link architectures can further improve the information flow throughout the network. Experimental results on three hyperspectral benchmark datasets demonstrate that our MLNets achieve competitive results compared with other state-of-the-art HSI classification approaches.**

***Index Terms*—Convolutional neural network (CNN), deep learning, hyperspectral image (HSI) classification, mixed link network (MLNet).**

I. Introduction

REMOTE sensing hyperspectral image (HSI) usually encompasses hundreds of spectral bands, which record abundant and unique information of various objects on the surface of the earth. Hence, HSIs have been employed in a wide variety of applications, including disaster monitoring [1], anomaly detection [2], and precision agriculture [3]. Recently, the classification of HSIs has gained remarkable attention in the hyperspectral community, since many hyperspectral applications are, in essence, classification tasks with the purpose of categorizing the pixels of HSIs into meaningful classes [4].

The traditional HSI classification algorithms concentrate on exploiting the spectral characteristic of a hyperspectral pixel to determine its class, such as multinomial logistic regression [5], decision trees [6], and support vector machine (SVM) [7]. However, due to the high intraclass and low interclass spectral variability, using spectral information alone makes the accurate identification of different objects difficult [8]. Considering the strong local space consistency in HSIs, methods that incorporate spatial-contextual information were proposed, which allow the joint exploitation of the spatial and spectral information to differentiate each hyperspectral pixel and further enhance the classification accuracy [9]. For instance, Li *et al.* [10] proposed a multiple features-based HSI classification paradigm, in which local spatial features extracted by the local binary pattern (LBP) operator, global spatial features captured by a Gabor filter, and original spectral features are combined for classification. In addition, multiple kernel learning (MKL) [11], superpixel [12], and sparse representation algorithms [13], [14] have also been explored to integrate spatial-contextual information with spectral signatures to achieve good classification accuracy. In [15], the conventional spectral-spatial feature-based HSI classification methods were systematically reviewed. However, the aforementioned approaches, such as LBP, superpixel, and sparse representation, extract fixed pattern features from raw data, which are highly dependent on prior knowledge and appropriate parameter setting, generally resulting in unsatisfactory performance [16].

Nowadays, deep learning techniques, which allow the automatic extraction of robust and hierarchical features in an end-to-end fashion, have made great breakthrough in many computer vision tasks (e.g., image classification [17] and object detection [18]). In the field of remote sensing, Chen *et al.* [19] first introduced the stacked autoencoders (SAEs) to learn deep spectral features together with deep spatial-dominated features for HSI classification. After that, deep learning models such as deep belief network (DBN) [20], convolutional neural network (CNN) [21], recurrent neural network (RNN) [22], [23], and capsule network (CapsNet) [24], [25] were also successfully applied to deal with HSI classification. Owing to the capability of automatically discovering spatial-contextual features, CNN models have been attracting more attention from researchers for HSI classification [26]–[28]. For instance, work in [26] jointly made use of the balanced local discriminative embedding algorithm and the CNN to conduct spatial-spectral HSI classification. Pan *et al.* [29] proposed a multigrained network (MugNet), which takes full advantage of different grains' spectral and spatial relationship for HSI classification. In [30]–[32], 3-D CNN models were proposed to directly learn spatial-spectral representations from raw hyperspectral data.

Manuscript received October 28, 2020; revised December 29, 2020; accepted January 19, 2021. Date of publication January 25, 2021; date of current version February 22, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61901198 and Grant 62071379, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2019JQ-377, and in part by the New Star Team of Xi'an University of Posts and Telecommunications under Grant xyt2016-01. *(Corresponding author: Zhe Meng.)*

Zhe Meng and Feng Zhao are with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: zhemeng@xupt.edu.cn; fzhao.xupt@gmail.com).

Licheng Jiao is with the School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: lchjiao@mail.xidian.edu.cn).

Miaomiao Liang is with the School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China (e-mail: liangmiaom@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3053567

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Recently, through engineering more powerful CNN architectures, much progress have been achieved in accurate HSI classification [33]–[37]. For instance, in [33], Lee *et al.* introduced residual learning to enhance the learning efficiency of conventional CNN model and employed a multiscale convolutional filter bank to exploit local spatial-spectral relationships of HSIs. In [34], Song *et al.* built very deep residual networks (ResNets) to learn discriminative features and then adopted a feature-fusing mechanism to achieve further performance improvement. Paoletti *et al.* [38] proposed the use of deep pyramid ResNet for HSI classification. In [39], an HSI classification framework based on the densely convolutional network (DenseNet) [40] was proposed, which introduces dense connections in the network to strengthen feature propagation, enhancing both the feature discriminability and the classification performance. In [41] and [42], some improved deep networks based on DenseNet were proposed, which can make full use of the multiscale information of HSIs. Considering that shortcut connections (also known as residual connections) in ResNets contribute to effective feature reusage and dense connections are effective for new feature exploration, Kang *et al.* [43] introduced the dual path network (DPN) that inherits both advantages of ResNet and DenseNet to learn more discriminative features from hyperspectral data [44]. More recently, Wang *et al.* [45] discovered and proved that both the ResNet and DenseNet are derived from the same dense topology intrinsically, in which each layer is connected with all the preceding layers. In addition, they demonstrated that the only difference between the path topologies of these two networks lies in the connection form, i.e., addition in ResNet and concatenation in DenseNet.

In this article, inspired by [45], two novel end-to-end mixed link networks (MLNets) are proposed for HSI classification. In MLNets, the additive links and concatenative links are combined by using mixed link architectures in order to enjoy benefits from both sides. Specifically, concatenative links assembled in the proposed networks could avoid repetitive learning of redundant features but focus on some new and more effective feature exploration, while additive links achieve reasonable feature reuse and avoid unnecessary loss of previous information, all of which help the model to extract discriminative features from HSIs. The blending links improve the information flow throughout the network. Moreover, by introducing shifted additions, the modification of raw features in the proposed mixed link architectures can alleviate feature redundancy to some extent. Experimental results on three hyperspectral benchmark data sets reveal that, compared to several state-of-the-art CNN models, such as ResNet, DenseNet, and DPN, the proposed model can achieve better performance in HSI classification. In particular, the proposed MLNets require fewer parameters than the DPN network whilst achieving better results. Notably, on University of Houston dataset, MLNets surpass DPN while being 3.23 times fewer parameters.

The rest of this article is organized as follows. Section II reviews the DenseNet and ResNet briefly and reveals that both of them are derived from the same dense topology. Section III describes the proposed method. Section IV presents the experimental results conducted on three benchmark HSI data sets. Finally, Section V concludes this article.

II. Dense Topology in Both DenseNet and ResNet

The network architecture plays a crucial role in the classification performance. For accurate HSI classification, much progress have been achieved by engineering network architectures [46]–[48]. In particular, most modern deep neural network-based HSI classification frameworks are built based either on the ResNet or on the DenseNet, both of which have obtained state-of-the-art performance in many computer vision tasks [18], [49].

ResNets can be built by stacking microblocks (also known as residual blocks) sequentially. For each residual block, the input features are element-wisely added to the output ones through identity shortcut connection, which not only helps information propagation but also eases the training of the network [50]. In DenseNet, dense connections enable each layer to receive raw information produced by all preceding layers, drawing representational power through effective new feature exploration. Specifically, the feature-maps learned by previous layers are concatenated and inputted into all subsequent layers, which further strengthens information flow [40].

Consider a network with L layers, each of which implements a nonlinear transformation $H_l(\cdot)$. l refers to the layer index and $H_l(\cdot)$ could be a composite function of several operations including convolution (Conv), linear transformation, batch normalization (BN) [51], activation [52], and pooling [53]. Assume that x_l is the immediate output of $H_l(\cdot)$.

Fig. 1 (left) illustrates the connection pattern in the DenseNet. For the lth layer in the network, it receives c_{l-1} as input, which is the concatenation result of all the previous outputs (i.e., $x_0, x_1, \ldots, x_{l-1}$). Mathematically, the output of the lth layer can be formulated as

$$\begin{aligned} x_l &= H_l(c_{l-1}) \\ &= H_l(x_0 \parallel x_1 \parallel \cdots \parallel x_{l-1}) \end{aligned} \tag{1}$$

where $\parallel$ represents the concatenation operation. Equation (1) indicates that DenseNet belongs to the dense topology clearly, i.e., each layer in the network is connected with all the preceding layers, and the connection function is concatenation, as shown in the right of Fig. 1.

Fig. 2 (left) shows the connection pattern in the ResNet, where shortcut connections are introduced to bypass each transformation $H(\cdot)$. Let r denote the addition result after shortcut connection, and r_0 equals x_0. We can formulate the residual learning process as

$$r_l = H_l(r_{l-1}) + r_{l-1}. \tag{2}$$

Note that $H_l(\cdot)$ takes r_{l-1} as input, and its immediate output is x_l, i.e., $x_l = H_l(r_{l-1})$. Considering the recursive property of (2), x_l can be rewritten as

$$\begin{aligned} x_l &= H_l(r_{l-1}) \\ &= H_l(H_{l-1}(r_{l-2}) + r_{l-2}) \\ &= H_l(H_{l-1}(r_{l-2}) + H_{l-2}(r_{l-3}) + r_{l-3}) \\ &= \cdots \end{aligned}$$

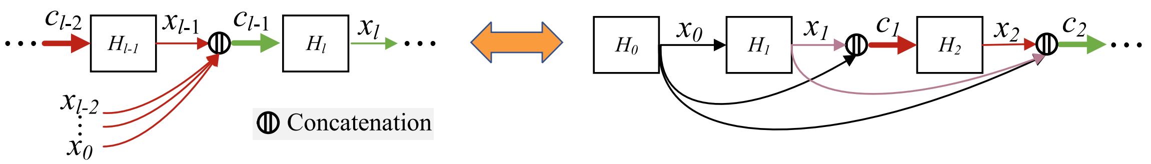


Fig. 1. Dense topology in the DenseNet.

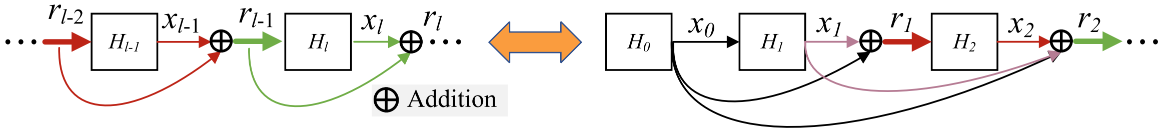


Fig. 2. Dense topology in the ResNet.

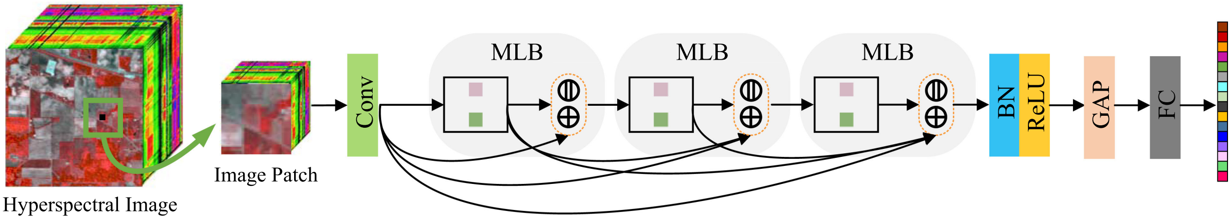


Fig. 3. Flowchart of the proposed MLNet-based HSI classification method.

$$
\begin{aligned}
&= H_l\left(\sum_{i=1}^{l-1} H_i(r_{i-1}) + r_0\right)\\
&= H_l\left(\sum_{i=1}^{l-1} x_i + x_0\right)\\
&= H_l(x_0 + x_1 + \cdots + x_{l-1}). \qquad (3)
\end{aligned}
$$

Equation (3) reveals that r_{l-1} is the element-wise sum of the outputs of the preceding $l-1$ layers, i.e., $r_{l-1} = x_0 + x_1 + \cdots + x_{l-1}$. The graphical view of (3) is illustrated in the right of Fig. 2, and one can see that ResNet also belongs to the dense topology. In addition, by comparing (1) and (3), we can find that the only difference between the topologies of DenseNet and ResNet is in the connection form, i.e., "$\|$" in DenseNet versus "+" in ResNet.

III. Methodology

The extraordinary success of both DenseNet and ResNet prove the effectiveness of dense topology. However, the additive connection in ResNet makes features from different layers aggregated on the same feature space, which may impede the flow of information throughout the network [40]. As for the DenseNet, concatenative connection allows each layer receiving raw features from all preceding layers, which are effective for the exploration of new features, but there may be the same type of raw features from different layers, resulting in feature redundancy [43]. To combine the advantages of the additive and concatenative connections and overcome their weaknesses, two novel dense topology-based MLNets are proposed for HSI classification.

Fig. 3 illustrates the flowchart of the proposed classification approach. Image patches centered at the labeled pixels are extracted and inputted to the MLNet, where a part of patches are used to train the network and the rest of the patches are used to evaluate the classification performance of the trained network. As shown in Fig. 3, the mixed link block (MLB) is the main part of the proposed MLNet. In this article, we propose two different MLBs, i.e., MLB-A and MLB-B, in order to combine the strengths of additive and concatenative connections, which will be detailed as follows.

A. *Mixed Link Blocks*

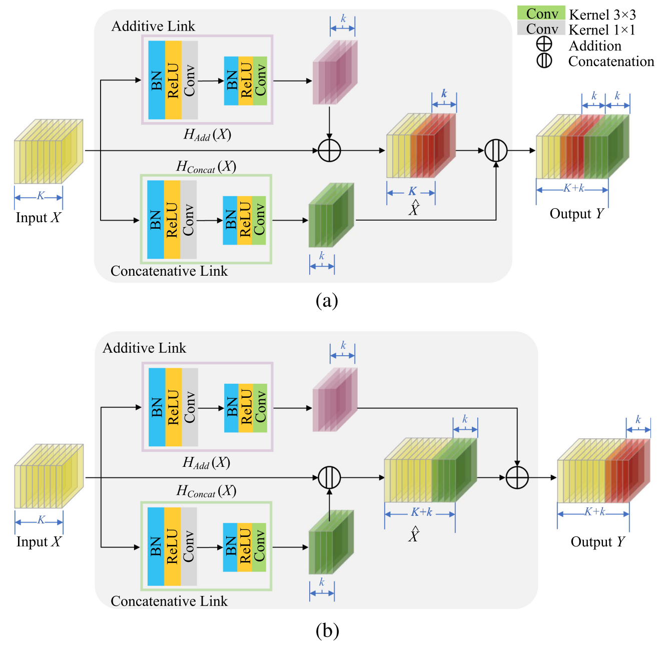
Fig. 4(a) illustrates the architecture of MLB-A. Let us consider X as the input of the MLB-A, which has K channels. The upper additive link first takes X as input and produces k ($k < K$) feature maps, which are added to the last k channels of the input X. The computation process can be formulated as

$$\hat{X} = X + H_{\text{Add}}(X) \qquad (4)$$

where $\hat{X}$ refers to the interim learned features and $H_{\text{Add}}(\cdot)$ denotes the function of generating feature maps for the additive link. As for the concatenative link, it appends k new feature maps outside the interim learned features $\hat{X}$

$$
\begin{aligned}
Y &= \hat{X} \parallel H_{\text{Concat}}(X)\\
&= (X + H_{\text{Add}}(X)) \parallel H_{\text{Concat}}(X) \qquad (5)
\end{aligned}
$$

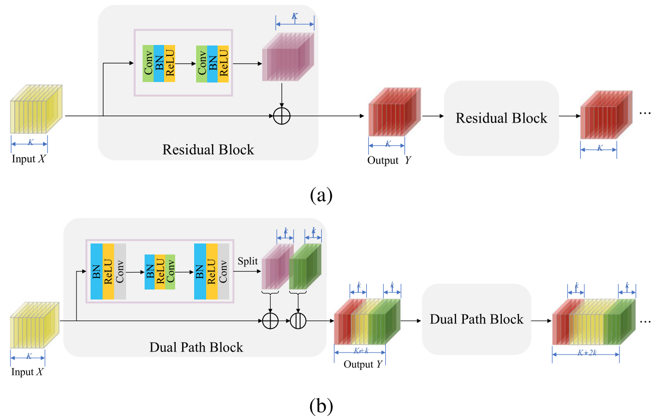
where Y denotes the output of the MLB-A, $\|$ represents the concatenation operation, and $H_{\text{Concat}}(\cdot)$ denotes the function to be learned in the concatenative link. Since the additive link does not change the number of feature maps, both the input X and



(a)

(b)

Fig. 4. Architectures of the proposed two mixed link blocks (MLBs). (a) MLB-A. (b) MLB-B.



(a)

(b)

Fig. 5. Illustration of the fixed additions in the (a) residual architecture [50] and (b) dual path architecture [43].

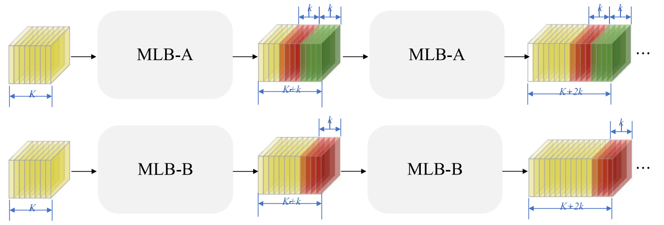


Fig. 6. Illustration of the shifted additions in the proposed two mixed link architectures.

the interim learned features $\hat{X}$ have K channels, and the output Y contains $K + k$ channels.

The mixed link architecture aims to utilize the strengths of both additive and concatenative links. With this motivation, an alternative architecture named MLB-B is proposed, which performs the concatenative link before the additive link, as shown in Fig. 4(b). Specifically, the concatenative link first takes X as input and produces k feature maps, which are appended outside the input X. The computation process can be formulated as

$$\hat{X} = X \parallel H_{\text{Concat}}(X). \tag{6}$$

Then, the feature maps produced by the additive link are added to the last k channels of the interim learned features $\hat{X}$

$$\begin{aligned} Y &= \hat{X} + H_{\text{Add}}(X) \\ &= X \parallel H_{\text{Concat}}(X) + H_{\text{Add}}(X). \end{aligned} \tag{7}$$

Therefore, in MLB-B, the number of channels of the interim features $\hat{X}$ and the output Y is $K + k$.

Note that although there are many additive positions can be chosen, learning variable positioning is currently unavailable due to the fact that their arrangement is not derivable directly. Therefore, we make a compromise and in MLB-A, we choose to align the position of additive part with the growing boundary of entire feature embedding, as shown in Fig. 4(a). In MLB-B, the position of additive part is aligned with the newly added channels caused by the concatenative link, as shown in Fig. 4(b). In addition, as shown in Fig. 4, $H_{\text{Add}}(\cdot)$ and $H_{\text{Concat}}(\cdot)$ are implemented with a bottleneck composite layer, i.e., BN-ReLU-Conv(1 × 1)-BN-ReLU-Conv(3 × 3), in order to improve computational efficiency as in [50] and [40]. In our experiments, the number of feature maps produced by the 1×1 and 3×3 Conv layers are 4 k and k, respectively, with $k = 36$.

In MLB-A, only $K - k$ number of channels remain unaltered between the input X and output Y. The rest of the channels in Y will be either modified or new features, as shown in Fig. 4(a). As for MLB-B, the input X with K channels stays unaltered, resulting in higher number of unmodified features being passed to subsequent layers, as shown in Fig. 4(b). Therefore, compared with MLB-A, MLB-B has higher feature redundancy. However, in MLB-B, the features produced by the concatenative link also undergo update because of the upcoming additive link, which helps to learn more complex features. Hence, compared with MLB-A, MLB-B is better in feature exploration. Overall, both of the integration way in our proposed architectures present their advantages. This is further confirmed in our subsequent experiments.

Considering that in ResNet and DPN, too many features are merged together by addition over the same feature space (called fixed additions in this article), which may impede the information flow [40], [43]. As shown in Fig. 5(a), for residual architecture, all the extracted features are merged together by addition. For dual path architecture in the DPN [see Fig. 5(b)], the additive features (denoted by purple color) are merged over the same fixed space. However, for the proposed mixed link architectures, the shifting of additive positions (denoted by red color) in subsequent feature spaces along multiple MLBs can alleviate this problem, as shown in Fig. 6.

TABLE I
NUMBER OF TRAINING AND TEST SAMPLES ON THE INDIAN PINES DATASET

No.	Class Name	Number	Training	Test
1	Alfalfa	54	15	39
2	Corn-notill	1434	50	1384
3	Corn-mintill	834	50	784
4	Corn	234	50	184
5	Grass/pasture	497	50	447
6	Grass/trees	747	50	697
7	Grass/pasture-mowed	26	15	11
8	Hay-windrowed	489	50	439
9	Oats	20	15	5
10	Soybean-notill	968	50	918
11	Soybean-mintill	2468	50	2418
12	Soybean-clean	614	50	564
13	Wheat	212	50	162
14	Woods	1294	50	1244
15	Buildings-Grass-Trees-Drives	380	50	330
16	Stone-Steel Towers	95	50	45
	Total	10366	695	9671

B. MLNets for HSI Classification

Based on the MLBs, this article designs two networks, including MLNet-A and MLNet-B. For instance, the MLNet-A is constructed by stacking several MLB-As. Fig. 3 illustrates the proposed MLNet-based HSI classification framework. Taking the Indian Pines scene as an example, the proposed network aims to classify each hyperspectral pixel into a certain land cover category. As can be seen, it takes image patch centered at each pixel as input. In this way, for each hyperspectral pixel, in addition to its own unique spectral characteristic, the spectral information of adjacent pixels and the spatial contextual information can be considered simultaneously, which reduces the intraclass variability and label uncertainty [38]. The extracted image patch is first fed into a 3×3 Conv layer to learn the initial spectral-spatial features. The number of output feature maps of the initial Conv layer is set as $2\ k$. Then, the obtained features are further processed by three MLBs. Finally, a global average pooling (GAP) [54] layer is utilized to transform the extracted spectral-spatial feature (with $5\ k$ channels) into a 1-D vector for classification. Specifically, we employ a fully connected (FC) layer followed by a softmax function to predict the conditional probability of each category, and the category with the maximum probability is the prediction result.

IV. EXPERIMENTS

A. Hyperspectral Datasets

Three well-known hyperspectral benchmark datasets, Indian Pines, University of Pavia, and University of Houston, were used in the experiments. The standard training and test sets were adopted as recommended in [55]. Tables I–III summarize the number of training samples and test samples of the three datasets.

Indian Pines dataset was captured by the AVIRIS instrument over the northwestern Indiana with spatial resolution of 20 m per pixel (mpp). It includes 145×145 pixels and 200 spectral bands covering the range from 400 to 2450 nm after discarding 20 bands corrupted by water absorption and four null bands. This scene contains 16 classes of interest. The false color composite image, the training map, and the test map of Indian Pines are shown in Fig. 7.

TABLE II
NUMBER OF TRAINING AND TEST SAMPLES ON THE UNIVERSITY OF PAVIA DATASET

No.	Class Name	Number	Training	Test
1	Asphalt	6852	548	6304
2	Meadows	18686	540	18146
3	Gravel	2207	392	1815
4	Trees	3436	524	2912
5	Painted metal sheets	1378	265	1113
6	Bare soil	5104	532	4572
7	Bitumen	1356	375	981
8	Self-Blocking Bricks	3878	514	3364
9	Shadows	1026	231	795
	Total	43923	3921	40002

TABLE III
NUMBER OF TRAINING AND TEST SAMPLES ON THE UNIVERSITY OF HOUSTON DATASET

No.	Class Name	Number	Training	Test
1	Healthy grass	1251	198	1053
2	Stressed grass	1254	190	1064
3	Synthetic grass	697	192	505
4	Trees	1244	188	1056
5	Soil	1242	186	1056
6	Water	325	182	143
7	Residential	1268	196	1072
8	Commercial	1244	191	1053
9	Road	1252	193	1059
10	Highway	1227	191	1036
11	Railway	1235	181	1054
12	Parking lot1	1233	192	1041
13	Parking lot2	469	184	285
14	Tennis court	428	181	247
15	Running track	660	187	473
	Total	15029	2832	12197

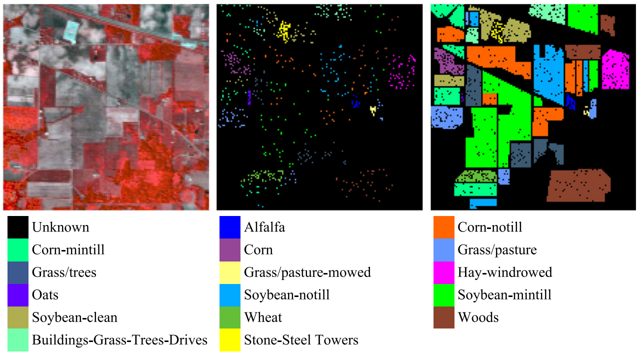


Fig. 7. Indian Pines dataset. Top (left to right): False color composite image, the training map, and the test map. Bottom: The legend.

University of Pavia dataset was gathered by the ROSIS senor over the city of Pavia, Italy, with spatial resolution of 1.3 mpp. It consists of 640×340 pixels and 103 spectral bands covering the range from 430 to 860 nm after discarding 12 noisy bands. This scene has nine reference classes. The false color composite image, the training map, and the test map of University of Pavia are shown in Fig. 8.

University of Houston dataset was collected by the CASI sensor over the campus of the University of Houston and the

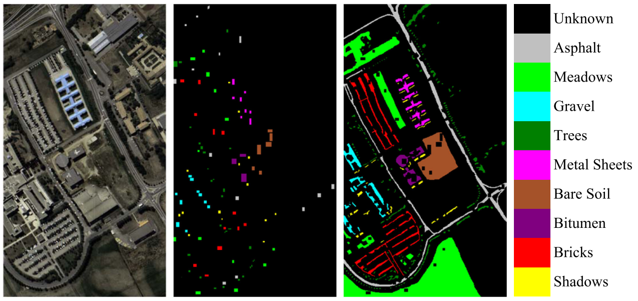


Fig. 8. University of Pavia dataset. From left to right: False color composite image, the training map, the test map, and the legend.

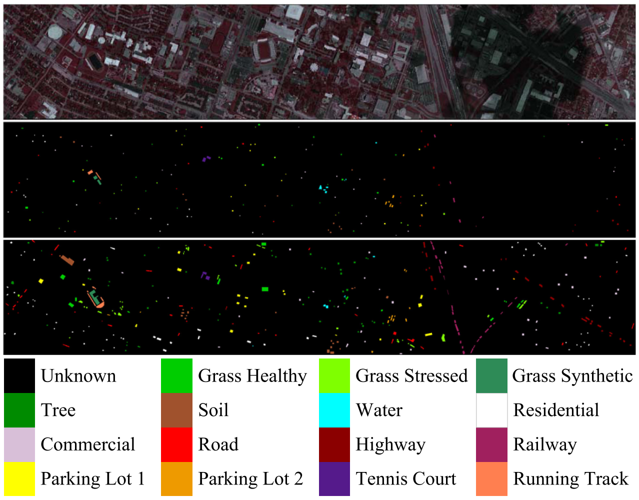


Fig. 9. University of Houston dataset. From top to bottom: False color composite image, the training map, the test map, and the legend.

neighboring area. It comprises 349×1905 pixels and 144 spectral bands covering the range from 380 to 1050 nm. The spatial resolution is 2.5 mpp. This dataset contains 15 classes. The false color composite image, the training map, and the test map of the University of Houston are shown in Fig. 9.

B. Experimental Setup

We set both the batch size and the training epochs to 100 and chose the Adam algorithm [56] to optimize the proposed network. The initial learning rate and the weight decay penalty were set to 0.001 and 0.0001, respectively. In addition, a cosine shape learning rate schedule was adopted, starting from 0.001 and gradually reducing to 0. The proposed network was designed and implemented using the Pytorch framework. Note that before fed into the network, the input HSI data were standardized to zero mean and unit variance. All the experiments were carried out on a personal computer equipped with AMD Ryzen 7 2700X CPU and a single Graphical Processing Unit (GPU) of NVIDIA GeForce RTX 2080.

To assess the classification performance, the overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ) were used. All the experiments were repeated five times and the averaged classification accuracies were reported.

C. Classification Results

To validate the effectiveness of our MLNet-A and MLNet-B for HSI classification, the proposed models were compared with two different kinds of approaches: 1) three classical methods, including SVM [7], extended morphological profiles (EMP) [9], and 3-D discrete wavelet transform (3DDWT) [57]; and 2) eight deep learning based approaches, including 3-D CNN [31], hybrid spectral convolutional neural network (HybridSN) [58], fully convolutional layer fusion network (FCLFN) [59], DenseNet [39], deep pyramid ResNet (pResNet) [38], MixNet [45], DPN [44], and spatial-spectral squeeze-and-excitation based ResNet (SSSERN) [60].

SVM is a spectral classifier that uses the radial basis function (RBF) as the kernel. For EMP and 3DDWT methods, they are used to extract spatial features from HSIs. The extracted features are concatenated with the original spectral features and fed into the SVM classifier for spectral-spatial classification. 3-D CNN uses 3-D convolution to extract the spectral and spatial information from HSIs simultaneously without relying on any preprocessing. HybridSN is a hybrid 3-D and 2-D model, which reduces the model complexity compared to 3-D CNN alone. FCLFN fuses spectral-spatial features extracted by all Conv layers in the CNN for HSI classification. DenseNet introduces concatenative links between layers, in which each layer is connected with every other layer in a feed-forward fashion. pResNet is an improved ResNet, which introduces additive links in plain CNN and gradually increases the feature map dimension at all Conv layers. MixNet contains three stages each of which is made up of a large number of blocks that have the similar architecture as MLB-A. DPN also combines the advantages of the additive link and concatenative link for HSI classification. However, in DPN the additive features are merged together over the same fixed space, which may impede the flow of information. SSSERN uses spatial-spectral squeeze-and-excitation module to adaptively refine features learned by the residual block, extracting more discriminative features of HSIs.

For all CNN-based compared methods, the network architectures were set according to the corresponding references. Considering that the spatial size of input HSI patch has a great impact on the classification performance, for the sake of fairness, we fixed the input patch size to 11 × 11 when comparing different CNN-based approaches as in [38], [39], and [60].

Table IV and Fig. 10 present the numerical and visual results of different methods on the Indian Pines dataset. The number of training and test samples used for this experiment is summarized in Table I. As shown in Table IV, the values of OA, AA, and κ of the proposed MLNet-A and MLNet-B are higher than that of other compared approaches. Specifically, our MLNet-A is able to reach the best OA (97.27%), AA (98.38%), and κ (0.9687) values. The MLNet-B achieves very similar values, being its OA, AA, and κ only 0.07%, 0.06%, and 0.0008 lower than MLNet-A, respectively. In addition, compared with other methods, the increases of OA scores obtained by the proposed MLNet-A are 22.91% (SVM), 9.15% (EMP), 8.54% (3DDWT), 5.73% (3-D CNN), 6.31% (HybridSN), 3.67% (FCLFN), 3.44% (DenseNet),

TABLE IV
CLASSIFICATION ACCURACIES OF DIFFERENT APPROACHES ON THE INDIAN PINES DATASET

No.	SVM	EMP	3DDWT	3D CNN	HybridSN	FCLFN	DenseNet	pResNet	MixNet	DPN	SSSERN	MLNet-A	MLNet-B
1	82.05	92.31	92.31	99.49	96.41	99.49	100	98.97	97.44	100	100	100	100
2	72.69	83.09	86.63	85.14	80.91	88.86	91.75	91.72	89.57	93.77	90.51	94.81	94.90
3	70.66	93.75	91.96	96.68	96.58	98.37	98.80	99.39	99.23	99.03	99.08	99.29	99.29
4	86.41	94.57	96.74	99.89	97.83	100	100	100	100	100	100	100	100
5	91.72	93.06	97.54	97.49	97.76	95.93	97.14	98.12	97.81	97.27	96.42	95.70	95.35
6	85.80	93.69	89.10	94.35	96.56	97.88	99.31	99.63	99.83	99.77	99.20	99.68	99.66
7	90.91	100	100	100	100	100	100	100	100	100	100	100	100
8	94.08	98.86	100	99.82	99.82	99.73	99.91	99.91	99.68	99.82	99.91	99.86	99.77
9	100	100	100	100	100	100	100	100	100	100	100	100	100
10	75.49	88.13	88.02	91.39	94.58	92.40	90.85	90.02	92.27	92.22	95.97	96.41	96.12
11	59.31	77.79	79.57	86.42	84.57	89.33	87.73	91.63	90.33	95.33	96.62	96.69	96.58
12	78.19	85.28	85.82	88.83	85.78	88.90	93.09	89.68	91.52	91.28	91.31	92.34	92.13
13	96.91	98.77	98.15	100	100	100	100	99.88	100	100	100	100	100
14	80.63	97.83	97.75	95.16	97.17	98.02	97.22	97.23	98.33	99.52	99.31	99.34	99.37
15	70.30	93.33	90.61	98.67	98.73	99.15	99.82	99.70	100	100	100	100	100
16	97.78	100	100	100	100	100	100	100	100	100	100	100	100
OA (%)	74.36	88.05	88.73	91.54	90.96	93.60	93.83	94.63	94.45	96.40	96.52	97.27	97.20
AA (%)	83.31	93.15	93.39	95.83	95.42	96.75	97.23	97.24	97.25	98.00	98.02	98.38	98.32
$\kappa \times 100$	71.05	86.39	87.15	90.34	89.70	92.67	92.95	93.86	93.64	95.87	96.00	96.87	96.79

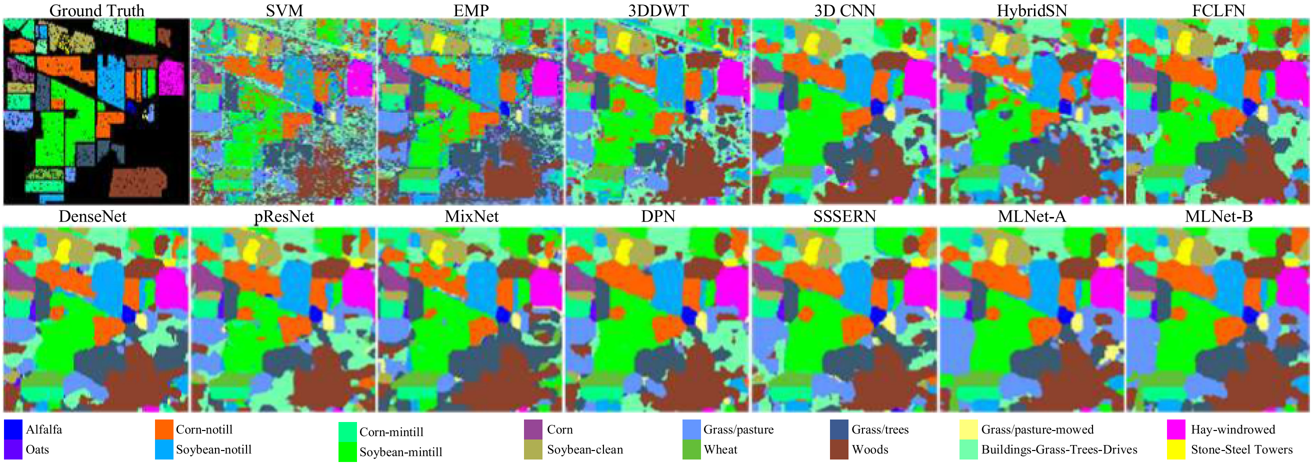


Fig. 10. Classification maps for the Indian Pines dataset.

2.64% (pResNet), 2.82% (MixNet), 0.87% (DPN), and 0.75% (SSSERN). The enhancements of AA scores are 15.07% (SVM), 5.17% (EMP), 4.99% (3DDWT), 2.55% (3D CNN), 2.96% (HybridSN), 1.63% (FCLFN), 1.15% (DenseNet), 1.14% (pResNet), 1.13% (MixNet), 0.38% (DPN), and 0.36% (SSSERN). The improvements of κ values are 0.2582 (SVM), 0.1040 (EMP), 0.0972 (3DDWT), 0.0653 (3-D CNN), 0.0717 (HybridSN), 0.0420 (FCLFN), 0.0392 (DenseNet), 0.0301 (pResNet), 0.0323 (MixNet), 0.0100 (DPN), and 0.0087 (SSSERN). In addition, as can be observed from Fig. 10, the classification maps obtained by our models are close to the ground truth map. These encouraging results demonstrate the effectiveness of the proposed models for the Indian Pines dataset.

Table V and Fig. 11 illustrate the numerical and visual results of different methods on the University of Pavia dataset. The number of training and test samples used for this experiment is summarized in Table II. From the observation of Table V, we can easily find that the proposed MLNet-B and MLNet-A achieve the best and the second best results in terms of the OA, AA, and κ scores. Compared with other approaches, the OA values' improvement obtained by the proposed MLNet-B are 17.34% (SVM), 16.25% (EMP), 7.68% (3DDWT), 15.68% (3-D CNN), 10.32% (HybridSN), 4.34% (FCLFN), 4.52% (DenseNet), 4.06% (pResNet), 4.61% (MixNet), 4.99% (DPN), and 1.74% (SSSERN). The AA values' enhancements are 7.48% (SVM), 8.73% (EMP), 4.36% (3DDWT), 8.76% (3-D CNN), 4.89% (HybridSN), 4.98% (FCLFN), 6.13% (DenseNet), 5.74% (pResNet), 6.19% (MixNet), 5.32% (DPN), and 0.64% (SSSERN). The κ values' increases are 0.2149 (SVM), 0.2049 (EMP), 0.0998 (3DDWT), 0.1964 (3D CNN), 0.1312 (HybridSN), 0.0598 (FCLFN), 0.0631 (DenseNet), 0.0562 (pResNet), 0.0643 (MixNet), 0.0693 (DPN), and 0.0227 (SSSERN). In addition, as shown in Fig. 11, our MLNet-A and MLNet-B reduce misclassified pixels and provide cleaner classification maps (particularly, compared with the 3-D CNN and HybridSN). These results demonstrate that the MLNet-A and MLNet-B models are effective for the University of Pavia dataset.

Table VI and Fig. 12 show the numerical and visual results of different approaches on the University of Houston dataset. The number of training and test samples used for this experiment is summarized in Table III. As shown in Table VI, our MLNet-A achieves the best performance from the overall aspect.

TABLE V
CLASSIFICATION ACCURACIES OF DIFFERENT APPROACHES ON THE UNIVERSITY OF PAVIA DATASET

No.	SVM	EMP	3DDWT	3D CNN	HybridSN	FCLFN	DenseNet	pResNet	MixNet	DPN	SSSERN	MLNet-A	MLNet-B
1	84.76	95.78	82.49	82.34	81.78	84.15	88.61	85.49	89.22	86.94	92.63	90.10	91.23
2	65.31	67.79	86.57	70.65	80.02	97.88	98.87	98.91	98.47	97.90	94.15	98.36	97.88
3	73.99	63.25	79.89	68.25	78.83	78.34	71.36	73.29	72.55	82.98	86.72	90.44	90.71
4	96.63	98.97	99.18	85.03	88.39	97.60	96.57	96.74	97.23	95.99	95.77	95.53	95.32
5	99.46	99.46	99.01	99.08	99.60	98.90	99.23	99.14	98.94	98.92	98.72	98.74	98.83
6	94.38	76.77	89.15	99.33	99.43	72.59	63.84	72.10	63.22	61.44	93.99	93.21	95.53
7	90.72	97.04	95.31	89.48	97.55	95.15	95.11	90.87	93.90	96.55	99.84	98.57	98.49
8	90.99	98.78	96.82	94.17	96.99	97.01	97.57	97.30	97.65	97.45	98.85	98.42	98.50
9	99.50	86.67	95.47	95.92	96.53	96.68	96.81	97.58	96.18	97.06	96.68	96.93	96.63
OA (%)	78.78	79.87	88.44	80.44	85.80	91.78	91.60	92.06	91.51	91.13	94.38	95.90	96.12
AA (%)	88.42	87.17	91.54	87.14	91.01	90.92	89.77	90.16	89.71	90.58	95.26	95.59	95.90
$\kappa \times 100$	73.29	74.29	84.80	75.14	81.66	88.80	88.47	89.16	88.35	87.85	92.51	94.47	94.78

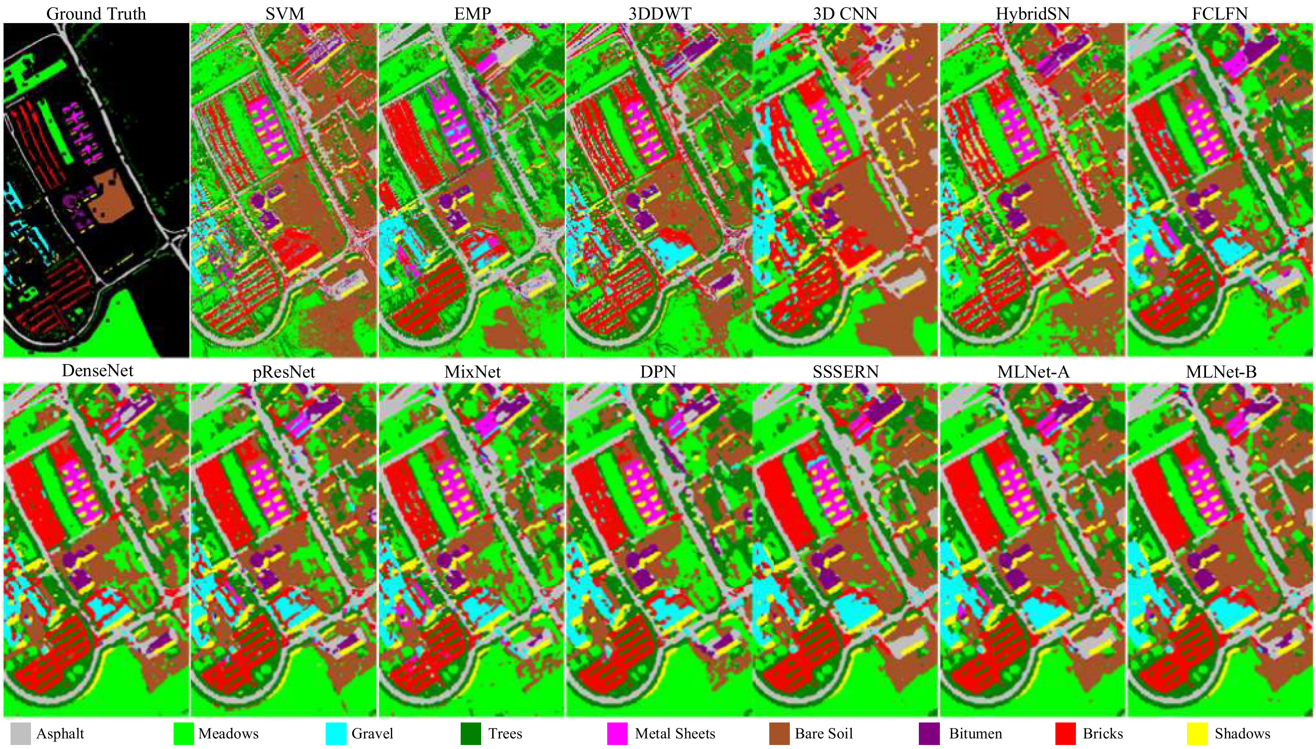


Fig. 11. Classification maps for the University of Pavia dataset.

TABLE VI
CLASSIFICATION ACCURACIES OF DIFFERENT APPROACHES ON THE UNIVERSITY OF HOUSTON DATASET

No.	SVM	EMP	3DDWT	3D CNN	HybridSN	FCLFN	DenseNet	pResNet	MixNet	DPN	SSSERN	MLNet-A	MLNet-B
1	82.34	81.77	91.55	82.89	82.45	82.05	82.34	82.03	82.77	81.56	85.58	81.77	81.79
2	83.36	83.65	96.52	84.34	84.38	84.40	85.15	84.27	85.11	84.38	85.06	85.11	85.00
3	99.80	99.60	99.21	91.13	96.36	96.75	93.35	89.43	88.71	96.04	99.60	99.17	99.13
4	98.96	90.06	96.31	87.16	91.93	91.78	90.64	91.53	91.31	91.95	90.93	90.74	90.06
5	98.96	99.24	99.62	99.41	99.98	99.68	99.68	99.24	99.92	99.32	99.87	99.96	100
6	99.30	99.30	97.90	92.45	97.48	94.97	93.99	96.22	93.85	93.43	96.22	95.80	95.80
7	77.33	87.97	78.92	80.06	85.65	87.48	81.66	84.68	85.43	86.90	83.84	82.74	83.58
8	64.48	65.81	69.42	67.12	77.57	71.00	70.29	76.14	71.66	71.55	74.61	71.43	70.85
9	69.41	80.08	70.25	83.97	82.55	78.43	71.27	78.96	73.88	78.05	79.34	81.95	82.49
10	63.13	63.13	51.06	65.00	63.38	59.73	59.69	61.49	61.12	63.55	60.97	64.69	63.90
11	79.13	75.24	75.62	78.90	81.76	77.50	82.07	74.71	77.53	82.60	84.76	97.95	96.53
12	77.43	76.95	81.27	87.95	98.16	92.49	91.57	90.30	91.70	91.12	91.72	95.08	93.95
13	68.77	68.42	90.18	88.91	87.79	81.75	80.00	78.60	79.16	84.00	82.25	86.95	86.88
14	100	100	100	95.71	99.60	98.46	97.17	93.04	92.87	99.35	100	99.84	99.92
15	97.25	100	97.04	93.87	99.96	86.72	81.01	69.98	79.66	97.72	99.28	96.62	98.22
OA (%)	81.41	82.33	83.27	83.15	86.39	83.70	82.37	82.47	82.55	84.70	85.41	86.65	86.43
AA (%)	83.98	84.75	86.32	85.26	88.60	85.55	83.99	83.37	83.65	86.77	87.60	88.65	88.54
$\kappa \times 100$	79.93	80.91	81.85	81.74	85.24	82.40	80.95	81.04	81.13	83.48	84.25	85.56	85.32

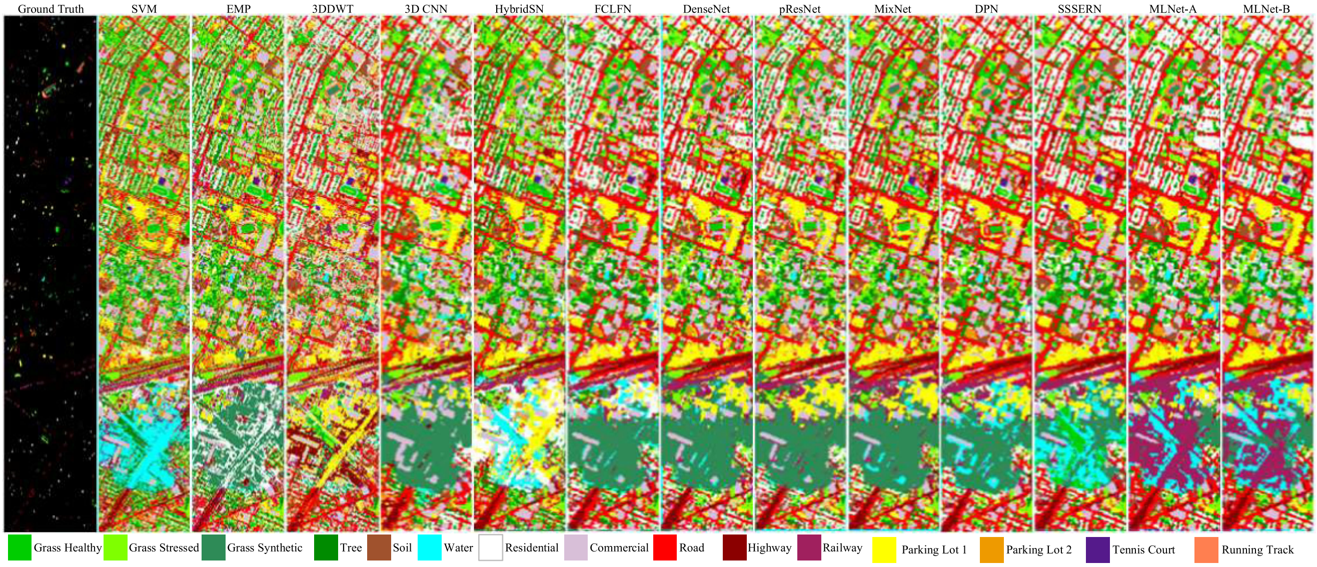


Fig. 12. Classification maps for the University of Houston dataset.

TABLE VII
NUMBER OF PARAMETERS OF DIFFERENT NETWORKS

Data Set	3D CNN	HybridSN	FCLFN	DenseNet	pResNet	MixNet	DPN	SSSERN	MLNet-A	MLNet-B
Indian Pines	100 856	534 656	195 408	1 668 992	1 994 776	1 508 812	685 968	165 404	509 128	509 128
University of Pavia	50 289	533 753	176 813	1 650 705	1 964 537	1 485 459	669 305	152 085	308 673	308 673
University of Houston	72 055	534 527	186 623	1 660 311	1 978 319	1 496 373	677 519	158 107	210 075	210 075

The OA, AA, and κ scores are 86.65%, 88.65%, and 0.8556, respectively. As for MLNet-B, the values of OA, AA, and κ are as high as 86.43%, 88.54%, and 0.8532, respectively, which demonstrates that the architecture of MLB-B can also capture discriminative features. Besides, compared with other methods, the increases of OA scores obtained by the proposed MLNet-A are 5.24% (SVM), 4.32% (EMP), 3.38% (3DDWT), 3.50% (3D CNN), 0.26% (HybridSN), 2.95% (FCLFN), 4.28% (DenseNet), 4.18% (pResNet), 4.10% (MixNet), 1.95% (DPN), and 1.24% (SSSERN). The improvements of AA scores are 4.67% (SVM), 3.90% (EMP), 2.33% (3DDWT), 3.39% (3-D CNN), 0.05% (HybridSN), 3.10% (FCLFN), 4.66% (DenseNet), 5.28% (pResNet), 5.00% (MixNet), 1.88% (DPN), and 1.05% (SSSERN). The enhancements of κ values are 0.0563 (SVM), 0.0465 (EMP), 0.0371 (3DDWT), 0.0382 (3-D CNN), 0.0032 (HybridSN), 0.0316 (FCLFN), 0.0461 (DenseNet), 0.0452 (pResNet), 0.0443 (MixNet), 0.0208 (DPN), and 0.0131 (SSSERN). In addition, as can be seen from Fig. 12, the proposed MLNet-A and MLNet-B can predict most of the categories well. For example, the proposed models show better connectivity for "Railway" category, which is consistent with the numerical results in Table VI. Specifically, for "Railway" category (class 11), the highest performance among all of the comparisons is 84.76% (SSSERN). However, our MLNet-A and MLNet-B can achieve performance as high as 97.95% and 96.53%, respectively. These positive results demonstrate that the proposed MLNets are also effective for the University of Houston dataset.

TABLE VIII
CLASSIFICATION RESULTS OF THE FCLFN, HYBRIDSN, AND THE PROPOSED MLNET-A AND MLNET-B ON THE INDIAN PINES DATASET WHEN USING 10% OF THE LABELED PIXELS AS TRAINING SAMPLES

Method	OA (%)	AA (%)	$\kappa \times 100$
HybridSN*	98.39	98.16	98.01
FCLFN*	98.56	95.94	98.36
MLNet-A	99.03	98.10	98.90
MLNet-B	**99.05**	**98.54**	**98.91**

Note That Accuracies Obtained by the Network Denoted With * Are the Best Results Reported in the Original Paper.

Table VII provides the total number of parameters of different networks. From Table VII, it is easy to observe that the number of parameters in the proposed MLNets is significantly fewer than that in the DenseNet, pResNet, and MixNet. In addition, MLNets outperform DPN while being 1.35 times, 2.17 times, and 3.23 times fewer parameters on Indian Pines, University of Pavia, and University of Houston datasets, respectively. Although our MLNets contain more parameters than SSSERN, they are able to achieve better classification performance.

From Tables IV–VI, one can see that the HybridSN, FCLFN, and DPN methods produce unsatisfactory classification accuracies. The main reason is that these networks need a larger input HSI patch for spatial feature extraction. Next, we further compare the proposed MLNet-A and MLNet-B with these three methods. Note that the best results reported in the corresponding references are used for comparison in this experiment. Table VIII reports the classification accuracies obtained by HybridSN, FCLFN, MLNet-A, and MLNet-B on the Indian Pines dataset. Following HybridSN [58] and FCLFN [59], 10% of the available

TABLE IX
CLASSIFICATION RESULTS OF THE DPN AND THE PROPOSED MLNET-A AND MLNET-B ON THE UNIVERSITY OF HOUSTON DATASET WHEN USING DIFFERENT NUMBER OF TRAINING SAMPLES PER CLASS

	30			40			50		
Method	OA (%)	AA (%)	$\kappa \times 100$	OA (%)	AA (%)	$\kappa \times 100$	OA (%)	AA (%)	$\kappa \times 100$
DPN*	93.93	94.72	93.44	94.99	95.53	94.58	96.40	96.78	96.10
MLNet-A	**94.60**	**95.06**	**94.16**	**95.59**	**96.00**	**95.23**	**96.50**	**96.79**	**96.21**
MLNet-B	94.16	94.71	93.69	95.28	95.68	94.89	96.46	96.77	96.17

Note That Accuracies Obtained by the Network Denoted With * Are the Best Results Reported in the Original Paper.

TABLE X
CLASSIFICATION RESULTS OF NETWORKS WITH DIFFERENT BUILDING BLOCKS ON THE INDIAN PINES DATASET

Block Name	OA (%)	AA (%)	$\kappa \times 100$	Parameters
Resdual Block	96.51	98.07	96.00	532 392
Dense Block	96.99	98.32	96.55	555 280
Dual Path Block	97.01	98.34	96.57	543 156
MLB-A	97.27	98.38	96.87	509 128
MLB-B	97.20	98.32	96.79	509 128

labeled pixels (randomly selected per class) are used as training samples. Table IX illustrates the results obtained by DPN, MLNet-A, and MLNet-B on the University of Houston dataset. Following DPN [44], various numbers of labeled samples (i.e., 30, 40, and 50) are randomly selected from each class as training samples. As we can see in Table VIII, the proposed MLNets achieve better performance than the HybridSN and FCLFN. Specifically, the proposed MLNet-B reaches an OA that is 0.66 percentual points higher than the HybridSN, and 0.49 percentual points higher than the FCLFN. In addition, from Table IX, one can see that our MLNet-A consistently outperforms DPN in terms of the OA, AA, and κ values when using different number of training samples. These comparison results again verify the effectiveness of our MLNets.

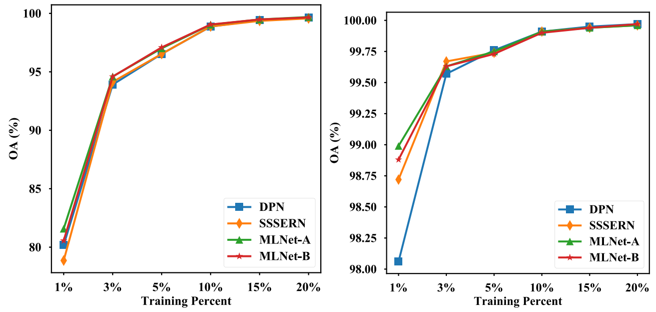
To sum up, MLNet-A and MLNet-B can obtain similar superior performance on the three real HSI datasets, indicating the effectiveness of the proposed two mixed link architectures for HSI classification.

D. *Comparison With Other Popular Building Blocks*

Modern deep neural networks utilize modular design to reduce the complexity of neural architectures. Layers are generally grouped into blocks, e.g., residual block in the ResNet. In this section, the proposed MLB-A and MLB-B are compared with three popular building blocks, including residual block, dense block, and dual path block. This experiment is implemented on the Indian Pines dataset. Specifically, we set the number of blocks to 3 and construct networks on the basis of different building blocks. The number of parameters of different networks is roughly the same for fair comparison. The results are shown in Table X. We find that networks based on the MLB-A and MLB-B can achieve better performance.

E. *Effect of Proportion of Training Data*

Fig. 13 summarizes the OAs of the proposed MLNet-A and MLNet-B with different percent of training data on the Indian Pines and University of Pavia datasets. Specifically, 1%, 3%, 5%, 10%, 15%, and 20% of samples per class are randomly chosen for training. Then, the rest of the samples are used for testing. Here, the DPN and SSSERN models are adopted as the reference, which perform well in the previous experiments.



(a) (b)

Fig. 13. OAs of various approaches with different training percent over the (a) Indian Pines and (b) University of Pavia datasets.

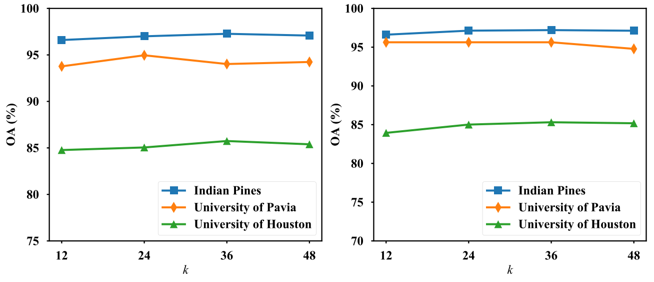
As can be seen, as the proportion of training samples increases, the performance of all approaches improves. Especially, when the training percent changes from 1% to 10%, the OAs of different methods increase dramatically. It is also observed from Fig. 13 that MLNet-A and MLNet-B outperform the other two compared methods in most cases, in particular when the training set is very small. For example, with only 1% of training samples per class, MLNet-A and MLNet-B achieve the best and the second best performance on the two datasets. These results demonstrate the excellent performance of the proposed approach for HSI classification.

F. *Effect of Shifted Additions*

In this section, the effectiveness of shifted additions in the proposed MLNet-A and MLNet-B is analyzed on the Indian Pines, University of Pavia, and University of Houston datasets. Specifically, we construct an MLNet with fixed additive positions (denoted by MLNet-F) for further comparison. For MLNet-F, the features learned by the additive link are always added to the first k channels of the input. Therefore, the additive features are always merged over the same fixed space, which may impede the information flow. To ensure a fair comparison, we make different networks contain the same number of parameters. The only difference among these three networks lies in the additive positions. Table XI shows the experiment results, from which we can easily find that shifted additions have positive contributions to the classification task, demonstrating the effectiveness of the proposed mixed link architectures.

TABLE XI
OA (%) AND THE NUMBER OF PARAMETERS OF THE MLNET WITH AND WITHOUT SHIFTED ADDITIONS

		Indian Pines		University of Pavia		University of Houston	
	Shifted Additions	OA	Parameters	OA	Parameters	OA	Parameters
MLNet-F	✗	96.97	656 224	94.30	591 849	85.30	777 291
MLNet-A	✓	97.14	656 224	94.49	591 849	85.64	777 291
MLNet-B	✓	97.06	656 224	94.58	591 849	85.94	777 291



(a)

(b)

Fig. 14. OAs obtained by the proposed (a) MLNet-A and (b) MLNet-B with different k settings over the three hyperspectral datasets.

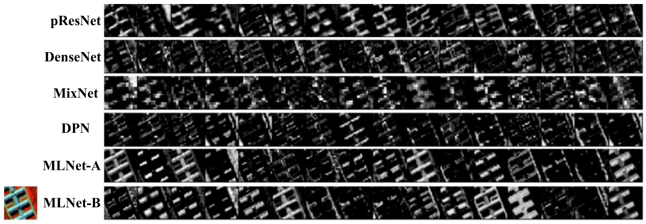


Fig. 15. Feature maps extracted by different networks on a metal sheet sample.

G. Effect of Parameter k

In this section, the parameter k is analyzed by using three datasets. The parameter k controls the number of feature maps generated by each link, which decides the representation capacity of the proposed MLNets. Fig. 14 shows the OAs obtained by MLNet-A and MLNet-B on three datasets under different $k = \{12, 24, 36, 48\}$. It can be observed that MLNet-A with $k = 36$ has achieved the best performance on the Indian Pines and University of Houston datasets. For the University of Pavia dataset, the best choice for parameter k is 24, which is slightly better than that of 36. As for MLNet-B, the curves reach the best OA values when the parameter k is set to 36 for all the three datasets. For the sake of consistency and the generalization of our models, we choose 36 as the default setting of the parameter k.

H. Visualization of Features Extracted by Different Networks

For more direct comparison of the effectiveness for feature learning, we randomly select 16 feature maps from each network, which all take from the final discriminant features before GAP, and visualize their distribution in Fig. 15. In Fig. 15, we can see that feature maps learned by the DenseNet and MixNet are coarse because of the downsampling of input HSI patch, presenting local fine spatial information loss. In addition, compared with the pResNet and DPN, one can see that the feature maps extracted the proposed MLNet-A and MLNet-B present finer local representation and spatial position, which are helpful for distinguishing objects occupying much smaller areas.

I. Effect of the Number of MLBs

From Fig. 3, one can observe that the proposed MLNets are mainly constructed by stacking several MLBs. The number of MLBs determines the network depth, which has an important impact on the representative capacity of the proposed model. Increasing the number of MLBs can generally improve the classification performance, but more MLBs in the network may suffer from overfitting.

Table XII summarizes the OAs of the proposed MLNet-A and MLNet-B with different number of MLBs over the three datasets. This experiment is conducted using the standard training and test sets, and the number of training and test samples are shown in Tables I–III. As can be observed, the best choice of the number of MLBs for the Indian Pines, University of Pavia, and University of Houston datasets are 3, 2, and 1, respectively.

In order to find a suboptimal value for the number of MLBs for all datasets, we further carry out extensive experiments. Specifically, we compare the classification accuracy (in terms of OA) of different number of MLBs with different amount of training data. For the Indian Pines dataset, the percentage of training data varies in the set $\{10\%, 20\%, 30\%\}$. For the University of Pavia and University of Houston datasets, the percentage of training data varies in the set $\{1\%, 2\%, 3\%\}$. The number of MLBs varies in the set $\{1, 2, 3, 4, 5, 6\}$. The corresponding OAs obtained by the proposed MLNet-A and MLNet-B are reported in Tables XIII and XIV. As can be observed, when the number of MLBs is larger than 2, the proposed MLNet-A and MLNet-B are able to achieve relatively stable high accuracy for all datasets. Considering that a larger number of MLBs will cause higher computational cost, it is recommended that the number of MLBs should be set to 3 in more general scenarios.

J. Effect of the Size of Input HSI Patch

As mentioned in Section III, image patches centered at each pixel are selected as the input of the proposed MLNets. In this section, our MLNets with different patch sizes are studied over the three data sets. Table XV reports the OAs of the proposed MLNet-A and MLNet-B when the patch size varies from 5×5 to 13×13. As shown in Table XV, for each dataset, the OA values first climb with the increasing of patch size. It is reasonable because input patches with larger size allow more spatial and spectral information to be utilized for feature extraction.

TABLE XII
OA (%) Obtained by the Proposed MLNet-A and MLNet-B With Different Number of MLBs

Number	Indian Pines		University of Pavia		University of Houston	
of MLBs	MLNet-A	MLNet-B	MLNet-A	MLNet-B	MLNet-A	MLNet-B
1	97.03	96.78	93.84	93.77	**86.65**	**86.43**
2	96.95	96.93	**95.90**	**96.12**	86.03	85.49
3	**97.27**	**97.20**	94.02	95.63	85.74	85.31
4	97.14	97.06	94.49	94.58	85.26	85.34
5	97.03	97.16	94.24	94.58	85.64	85.94
6	96.86	97.03	93.32	93.75	85.42	84.97

TABLE XIII
OA (%) Obtained by the Proposed MLNet-A With Different Number of MLBs When Using Different Percentages of Training Samples

		Number of MLBs					
Data Set	Training Percentage	1	2	3	4	5	6
Indian Pines	10%	98.86	98.97	99.03	99.06	99.11	99.08
	20%	99.63	99.66	99.69	99.67	99.66	99.68
	30%	99.82	99.84	99.85	99.87	99.85	99.86
University of Pavia	1%	98.60	98.99	98.90	98.89	99.00	98.87
	2%	99.30	99.40	99.48	99.49	99.49	99.45
	3%	99.47	99.63	99.70	99.66	99.70	99.68
University of Houston	1%	87.44	89.89	90.67	90.50	90.30	90.25
	2%	87.95	89.42	90.55	91.30	90.91	91.19
	3%	94.41	94.75	94.77	94.98	94.92	94.77

TABLE XIV
OA (%) Obtained by the Proposed MLNet-B With Different Number of MLBs When Using Different Percentages of Training Samples

		Number of MLBs					
Data Set	Training Percentage	1	2	3	4	5	6
Indian Pines	10%	98.66	98.87	99.05	99.05	99.04	99.09
	20%	99.63	99.66	99.65	99.66	99.65	99.67
	30%	99.80	99.85	99.83	99.86	99.86	99.85
University of Pavia	1%	98.22	98.88	98.85	98.83	98.97	98.88
	2%	99.23	99.35	99.49	99.44	99.49	99.44
	3%	99.45	99.63	99.68	99.64	99.68	99.67
University of Houston	1%	84.13	89.35	90.58	90.44	90.14	90.43
	2%	86.04	89.01	90.24	90.22	90.92	90.99
	3%	94.01	94.53	94.63	94.64	94.84	94.82

TABLE XV
OA (%) Obtained by the Proposed MLNet-A and MLNet-B With Different Patch Sizes

	Indian Pines		University of Pavia		University of Houston	
Patch Size	MLNet-A	MLNet-B	MLNet-A	MLNet-B	MLNet-A	MLNet-B
5×5	94.84	95.26	93.93	93.96	83.03	83.55
7×7	97.59	97.67	94.42	94.64	83.49	84.67
9×9	**97.94**	**97.82**	94.84	95.19	85.36	86.11
11×11	97.27	97.20	**95.90**	96.12	86.65	**86.43**
13×13	96.48	96.50	95.40	**96.37**	**87.04**	86.13

However, too large patch size also results in the degradation of performance, especially for the Indian Pines dataset. The reason behind this is that, multiple materials from different categories might be included in an HSI patch with large size, which will harm the classification tasks.

V. Conclusion

In this article, by embracing both additive links and concatenative links, the proposed MLNets enable effective feature reusage and new feature exploration, which not only reduce the relearning of redundant features but also help extract informative spatial-spectral features. Furthermore, through shifted additions, the proposed blending connections further enhance the flow of information between layers in the network. To verify the performance of the proposed MLNets, experiments based on three hyperspectral benchmark datasets are conducted. Experimental results demonstrate the superiority of the proposed MLNets over several state-of-the-art methods, such as DenseNet, ResNet, and DPN.

In the future works, we will carry out further research and try to figure out the importance of each link by integrating attention mechanism, which may be helpful for extracting more discriminative features from HSIs.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments and constructive suggestions for this article.

References

[1] S. Veraverbeke *et al.*, “Hyperspectral remote sensing of fire: State-of-the-art and future perspectives,” *Remote Sens. Environ.*, vol. 216, pp. 105–121, Oct. 2018.

[2] J. Lei, W. Xie, J. Yang, Y. Li, and C.-I. Chang, “Spectral-spatial feature extraction for hyperspectral anomaly detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8131–8143, Oct. 2019.

[3] S. W. Shivers, D. A. Roberts, and J. P. McFadden, “Using paired thermal and hyperspectral aerial imagery to quantify land surface temperature variability and assess crop stress within California orchards,” *Remote Sens. Environ.*, vol. 222, pp. 215–231, Mar. 2019.

[4] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, “Advanced spectral classifiers for hyperspectral images: A review,” *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[5] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 318–322, Mar. 2012.

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

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

[8] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. 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.

[9] 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.

[10] W. Li, C. Chen, H. Su, and Q. Du, “Local binary patterns and extreme learning machine for hyperspectral imagery classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[11] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, “Multiple kernel learning for hyperspectral image classification: A review,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.

[12] T. Lu, S. Li, L. Fang, X. Jia, and J. A. Benediktsson, “From subpixel to superpixel: A novel fusion framework for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4398–4411, Aug. 2017.

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

[14] Q. Gao, S. Lim, and X. Jia, “Spectral-spatial hyperspectral image classification using a multiscale conservative smoothing scheme and adaptive sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7718–7730, Oct. 2019.

[15] L. He, J. Li, C. Liu, and S. Li, “Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

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

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.

[18] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

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

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

[21] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, “Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5585–5599, Oct. 2017.

[22] 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.

[23] W. Qi, X. Zhang, N. Wang, M. Zhang, and Y. Cen, “A spectral-spatial cascaded 3D convolutional neural network with a convolutional long short-term memory network for hyperspectral image classification,” *Remote Sens.*, vol. 11, no. 20, Oct. 2019, Art. no. 2363.

[24] M. E. Paoletti *et al.*, “Capsule networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2018.

[25] K. Zhu, Y. Chen, P. Ghamisi, X. Jia, and J. A. Benediktsson, “Deep convolutional capsule network for hyperspectral image spectral and spectral-spatial classification,” *Remote Sens.*, vol. 11, no. 3, Jan. 2019, Art. no. 223.

[26] W. Zhao and S. Du, “Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[27] M. Zhang, W. Li, and Q. Du, “Diverse region-based CNN for hyperspectral image classification,” *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[28] Q. Gao and S. Lim, “Classification of hyperspectral images with convolutional neural networks and probabilistic relaxation,” *Comput. Vis. Image Underst.*, vol. 188, Nov. 2019, Art. no. 102801.

[29] B. Pan, Z. Shi, and X. Xu, “MugNet: Deep learning for hyperspectral image classification using limited samples,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, Nov. 2018.

[30] 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.

[31] Y. Li, H. Zhang, and Q. Shen, “Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network,*” Remote Sens.*, vol. 9, no. 1, Jan. 2017, Art. no. 67.

[32] 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.

[33] H. Lee and H. Kwon, “Going deeper with contextual CNN for hyperspectral image classification,” *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[34] 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.

[35] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, “Hyperspectral image classification based on deep deconvolution network with skip architecture,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018.

[36] N. Audebert, B. Le Saux, and S. Lefévre, “Deep learning for classification of hyperspectral data: A comparative review,” *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.

[37] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, “A CNN with multiscale convolution and diversified metric for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.

[38] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, “Deep pyramidal residual networks for spectral-spatial hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.

[39] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, “Deep&dense convolutional neural network for hyperspectral image classification,” *Remote Sens.*, vol. 10, no. 9, Sep. 2018, Art. no. 1454.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[41] Z. Meng, L. Li, L. Jiao, Z. Feng, X. Tang, and M. Liang, "Fully dense multiscale fusion network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 22, Nov. 2019, Art. no. 2718.

[42] C. Zhang, G. Li, and S. Du, "Multi-scale dense networks for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9201–9222, Nov. 2019.

[43] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 4467–4475.

[44] X. Kang, B. Zhuo, and P. Duan, "Dual-path network-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 447–451, Mar. 2018.

[45] W. Wang, X. Li, J. Yang, and T. Lu, "Mixed link networks," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2819–2825.

[46] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral-spatial convolution network framework for hyperspectral images classification," *Remote Sens.*, vol. 10, no. 7, Jul. 2018, Art. no. 1068.

[47] Z. Meng, L. Li, X. Tang, Z. Feng, L. Jiao, and M. Liang, "Multipath residual network for spectral-spatial hyperspectral image classification," *Remote Sens.*, vol. 11, no. 16, Aug. 2019, Art. no. 1896.

[48] S. K. Roy, S. Chatterjee, S. Bhattacharyya, B. B. Chaudhuri, and J. Platoš, "Lightweight spectral-spatial squeeze-and-excitation residual bag-of-features learning for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5277–5290, Aug. 2020.

[49] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[50] 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.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. IEEE Int. Conf. Mach. Learn.*, Jul 2015, pp. 448–456.

[52] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf.* Artif. Intell. Statist., Apr. 2011, pp. 315–323.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[54] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. IEEE Int. Conf. Learn. Represent.*, Apr. 2014, pp. 1–8.

[55] P. Ghamisi*et al.*, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

[56] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980.*

[57] 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.

[58] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Jun. 2019.

[59] G. Zhao, G. Liu, L. Fang, B. Tu, and P. Ghamisi, "Multiple convolutional layers fusion framework for hyperspectral image classification," *Neurocomputing*, vol. 339, no. 28, pp. 149–160, Apr. 2019.

[60] L. Wang, J. Peng, and W. Sun, "Spatial-spectral squeeze-and-excitation residual network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 7, Apr. 2019, Art. no. 884.

Zhe Meng (Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2014 and 2020, respectively.

He is currently a Lecturer with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications. His research interests include deep learning and hyperspectral image classification.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Since 1992, he has been a Professor with Xidian University, Xi'an, China, where he is currently the Director of Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, the Fellow of IET/CAAI/CIE/CCF/CAA, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.

Miaomiao Liang received the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2018.

She is currently a Lecturer with the School of Information Engineering, Jiangxi University of Science and Technology. Her research interests include computer vision, machine learning, and remote sensing images processing.

Feng Zhao (Member, IEEE) received the B.S. degree in computer science and technology from Heilongjiang University, Heilongjiang, China, in 2004, the M.S. degree in signal and information processing from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2007, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2010.

She has been a Professor with the School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, since 2015. She has authored or coauthored more than 30 articles and two books. Her research interests include fuzzy information processing, pattern recognition, and image processing.

Dr. Zhao was a recipient of New-Star of Young Science and Technology Award supported by Shaanxi, in 2014, and the IET International Conference on Ubi-media Computing Best Paper Award, in 2012.