Deep Feature Fusion via Two-Stream Convolutional Neural Network for Hyperspectral Image Classification

Xian Li, $^{\odot}$ Student Member, IEEE, Mingli Ding, and Aleksandra Pižurica, Senior Member, IEEE

Abstract—The representation power of convolutional neural network (CNN) models for hyperspectral image (HSI) analysis is in practice limited by the available amount of the labeled samples, which is often insufficient to sustain deep networks with many parameters. We propose a novel approach to boost the network representation power with a two-stream 2-D CNN architecture. The proposed method extracts simultaneously, the spectral features and local spatial and global spatial features, with two 2-D CNN networks and makes use of channel correlations to identify the most informative features. Moreover, we propose a layer-specific regularization and a smooth normalization fusion scheme to adaptively learn the fusion weights for the spectral–spatial features from the two parallel streams. An important asset of our model is the simultaneous training of the feature extraction, fusion, and classification processes with the same cost function. Experimental results on several hyperspectral data sets demonstrate the efficacy of the proposed method compared with the state-of-the-art methods in the field.

Index Terms—Convolutional neural networks (CNNs), feature fusion, hyperspectral image (HSI) classification, squeeze-and-excitation (SE).

I. INTRODUCTION

HYPERSONTICAL remote sensing remains to be one of the key technologies for the Earth observation and also as one of the most demanding and challenging ones for data processing and analysis [1], [2]. Captured with hundreds of contiguous and narrow spectral bands, hyperspectral images (HSIs) enable more accurate discrimination between different materials in the scene than conventional panchromatic and multispectral remote sensing images [3]. Hence, the technology has been widely adopted in a range of applications, including defense and security [4], agriculture [5], geology [6], ocean [7], and environmental monitoring [8].

While different materials can typically be distinguished based on their spectral signatures, scene classification based on spectral information alone is often not accurate enough. Various factors, such as spatial variability of spectral signatures [9] and spectral noise, increase the intraclass variability. If the interclass variability is small, it is difficult to differentiate one class from another [10]. With the improvement of spatial resolution in HSI, it becomes natural to make use of the spatial information as well [11]. For example, knowing that adjacent pixels in homogeneous areas are likely to belong to the same class, we can improve the results in precise mapping. It is generally agreed that combined spectral–spatial classification improves the accuracy significantly compared with spectral classification alone [12].

Feature extraction and feature fusion are the two crucial steps in spectral–spatial classification. Various approaches have been proposed to incorporate spatial context into feature extraction, using, e.g., segmentation [13], [14], morphological filters [15], Markov random field (MRF) models [16], and texture features [17]. State-of-the-art feature extraction approaches include multiple kernel learning [18], sparse representation [19]–[23], and active learning [24]. Recent explosion of deep learning has transformed feature extraction. Instead of hand-crafting features based on domain-specific expert knowledge and a lot of parameter tuning, new deep learning approaches learn automatically a hierarchical feature representation that is optimally suited for complex classification and recognition tasks.

Deep learning models for feature extraction from HSIs can be grouped in four main categories: models employing stacked autoencoders (SAEs) [25], [26], deep belief networks (DBNs) [27], [28], recurrent neural networks (RNNs) [29], [30], and convolutional neural networks (CNNs) [31]–[38]. Compared with the other deep learning models, CNNs facilitate extraction of spatial features, as they can operate directly on image patches, without flattening them to one dimension. Besides, CNNs reduce hugely the number of learning parameters compared to fully connected networks with the same number of hidden units, with their local receptive fields and shared-weights’ architecture, which is the main reason for their dominance in image/video processing.

Spectral–spatial feature extraction and classification based on CNN methods can be generally divided into two categories. The first category extracts jointly spectral–spatial features using 3-D filtering. For instance, Chen et al. [31]...
proposed a 3-D CNN model with a large receptive field in the spectral domain and a small receptive field in the spatial domain to extract the integrated spectral–spatial features. Similarly, in the 3-D CNN framework of Li et al. [39], the spectral–spatial features are extracted simultaneously, taking full advantage of the structural characteristics of the 3-D HSI data. Zhong et al. [40] introduced residual learning to 3-D CNN to consecutively learn discriminative features from abundant spectral signatures and spatial contexts in HSIs. However, 3-D CNN feature extraction and classification methods often exploit shallow networks to avoid overfitting due to an additional filter in the spectral dimension compared to 2-D CNN. This limits their ability in exploiting the available spectral–spatial information, and the resulting classification maps tend to be oversmoothed [41].

The second large category of feature extraction methods extracts the spectral features and the spatial features separately and fuses them subsequently. Most of the spatial feature extraction methods are CNN-based methods inspired by computer vision models, while the spectral feature extraction methods are more diverse, including balanced local discriminant embedding [42], SAE [43], and stacked denoising autoencoder (SdAE) [44]. As opposed to the above-described methods, which apply different architectures in their spatial and spectral stream, several recent works, including [41] and [45], formulated unified approaches to spectral–spatial feature extraction, with an end-to-end training strategy and a uniform objective function. Spectral feature extraction in all these methods is based on 1-D CNN.

Next to the spectral and spatial feature extraction, feature fusion is another key step in the classification task. CNN-based methods typically use one or more fully connected layers with a rectified linear unit (ReLU) nonlinear activation function to fuse the extracted features [31], [34], [44], [46]. For example, Song et al. [46] proposed a deep feature fusion network by introducing residual learning to increase the network depth. The features extracted from multiple (low, middle, and high levels) layers as complementary information were fused by global average pooling (GAP) and the fully connected layers with ReLU. We hypothesize that the fused features using the ReLU in the fully connected layers tend to blow up (the output range is [0, inf]), due to which some detail features may be lost.

Although the above-described CNN-based methods demonstrated huge success in HSI processing, two important challenges remain. First, a large number of labeled training samples are required to obtain a satisfactory performance. In practice, a limited amount of training data and unbalanced samples constrain the network depth and width, reducing the feature extraction capability. Some strategies, such as data augmentation [34], [47], [48] and transfer learning [45], [49], are adopted to alleviate this problem to a certain extent, but the inherent limitation of the models remains a limiting factor for the network performance. The second challenge is how to exploit the spectral and spatial information more effectively. Although various approaches have been proposed, this question remains relevant, both theoretically and practically. In [34], it was pointed out that a single input architecture has strong limitations in heterogeneous area, and thus, the authors proposed using multiple inputs based on six diverse regions to better extract spectral–spatial features. This led to improved performance compared to most of single input methods [32], [47], [49], but the diverse regions’s construction is time-consuming and each region employs similar shallow networks.

We address the challenges mentioned earlier and propose a novel two-stream spectral and spatial feature extraction, feature fusion, and classification architecture based on 2-D CNN. Specifically, we develop a deep learning framework, which extracts simultaneously local and global spatial–spectral features via two streams that operate in parallel. The first stream is a shallow 2-D CNN that extracts spectral and local spatial correlation features from a relatively small image patch. The second stream is a deep 2-D CNN, which extracts more complex global spatial structure from a relatively large image patch. Hence, the complete network extracts spectral, local spatial, and global spatial features. Inspired by squeeze-and-excitation (SE) networks [50] that were recently introduced in the field of computer vision, we introduce a related SE module to further enhance the feature extraction capability of the two streams. In the fusion stage, our method learns adaptively the fusion weights to form joint features. The output labels are then predicted by a softmax layer.

The main contributions of this article are as follows.

1) We propose a novel two-stream CNN architecture for HSI classification, which extracts spectral, local spatial, and global spatial information in parallel. In this approach, the feature extraction, fusion, and classification are trained in an end-to-end manner under a unified objective function.

2) We introduce an effective approach for improving the spectral–spatial feature extraction capability of the two parallel streams based on interchannel correlations and the so-called SE concept. This is especially important in practice where the actual depth of the feature extraction streams is limited by the available amount of training data. To this end, we derive a formal approach for incorporating the SE concept into HSI spectral–spatial classification.

3) We propose a layer-specific regularization and smooth normalization fusion scheme, which adaptively controls the fusion weights and better fuses the spectral–spatial features.

4) We embed a 2-D CNN into the feature extraction stream. Different from conventional spectral feature extraction streams which were always based on 1-D CNN or other 1-D methods, our proposed feature extraction stream operates on small image patches and extracts simultaneously spectral and local spatial features. Moreover, we combine shallow and deep networks to extract optimally both spectral and spatial information content. This configuration effectively makes use of multiscale spectral–spatial information and fuses features at different depths.

The rest of this article is organized as follows. Section II reviews the basic concepts of CNN and the ideas behind
residual learning and SE approaches. Section III introduces the proposed method. A thorough experimental evaluation and a discussion of the results are given in Section IV, and finally, Section V draws the conclusion of this article.

II. BACKGROUND AND RELATED WORKS

A. CNN

Among all deep learning models including SAEs, DBNs, and RNNs, CNNs have been by far the most extensively employed in computer vision problems, mainly due to their efficiency with local connections, shared weights, and flexibility, admitting different volumes of neurons.

The basic components of CNN models include a convolutional layer, pooling layer, and fully connected layer. The convolutional layer usually contains several convolutional kernels, with which different feature maps are computed and then fed to a nonlinear activation function. Let \( X \in \mathbb{R}^{H \times W \times N} \) be the input of a convolutional layer, where \( H' \times W' \) is the spatial size and \( N' \) is the number of channels. Given \( N \) convolutional kernels \( W_i \) (bias terms are omitted), the output \( F_i(X) \in \mathbb{R}^{H' \times W' \times N} \) is computed as

\[
F_i(X) = \delta(W_i \ast X)
\]  

where \( \delta \) denotes the activation function. ReLU [51] is among the most often used activation functions in CNN models because it offers faster convergence and better performance than the traditional saturated activation functions, such as sigmoid or tanh [31].

The role of the pooling layers is to create more general and abstract features by reducing the size of the feature map. They are usually placed between the convolutional layers. After several convolutional layers and pooling layers, one or more fully connected layers are typically employed to combine all the features from the previous layer into more global features. Batch normalization was proposed to effectively alleviate the problem known as internal covariate shift, i.e., changes in the distribution of layer's inputs during training [52]. It enables using larger learning rates and accelerating this way the training process without the risk of divergence.

B. Residual Learning and SE

Deeper CNN-based models are able to approximate target functions with increased nonlinearity and thus to extract more complex features, leading to improved classification. However, deeper models require abundant training data. In practice, a limited amount of high-dimensional training samples constrains the depth and the width of CNN models due to various phenomena, such as the Hughes phenomenon [53], overfitting, and gradient vanishing [46]. Consequently, the current CNN models for HSI classification are rather small networks [54].

In general, residual learning can increase the network depth [55] and improve hereby HSI classification [40], [46]. Let \( \mathcal{H}(X) \) denote the desired mapping. In the traditional approach, such as in Fig. 1(a), every few stacked layers learn this desired mapping. The idea of residual learning shown in Fig. 1(b) is to learn instead the residual \( \mathcal{R}(X) = \mathcal{H}(X) - X \). It is easier to optimize this residual than the original mapping. In the extreme case where \( \mathcal{H} \) is the identity operator, it is obviously easier to force the function \( \mathcal{R}(X) = 0 \) than \( \mathcal{H}(X) = X \) [55]. With the residual learning modules, the main training task of a deeper network is simplified into training of multiple residual functions, which facilitates the training process and increases the network depth.

The concept of SE networks (SENets) [50] has been recently introduced in the field of computer vision to enhance the feature extraction capability of the network by emphasizing automatically informative features and suppressing the less useful ones. The SE module consists of the squeeze part and the excitation part. The squeeze part squeezes the spatial information from each feature channel into a single number by GAP. In this way, the collection of \( N \) channels \( F_i(X) \) is transformed into a vector with \( N \) elements. The excitation part uses two fully connected layers to learn channel-wise correlations \( e_i \in \mathbb{R}^{N \times 1} \)

\[
e_i = \sigma(W'' \delta(W' A(F_i(X))))
\]

where \( A \) denotes the GAP operation. \( \delta \) and \( \sigma \) are the ReLU and the sigmoid, respectively. \( W' \in \mathbb{R}^{(N/r) \times N} \) and \( W'' \in \mathbb{R}^{N \times (N/r)} \) are the weights of the two fully connected layers, respectively. A reduction ratio \( r \) is to adjust the capability and computational cost. \( F_i(X) \) is then rescaled by \( e_i \), promoting this way more informative feature channels.

III. PROPOSED METHOD

A. Overall Architecture

Here, we propose a novel two-stream CNN architecture for HSI analysis. The two streams that operate in parallel as shown in Fig. 2 extract simultaneously local and global spatial–spectral features. Specifically, the local feature extraction stream is a shallow network that takes as input all spectral bands of an HSI and extracts spectral and local spatial correlation features from small image patches. In parallel, the global feature extraction stream is a deep network that takes as input much less (several to a dozen) principal components of an HSI and extracts more complex global spatial structure features from large image patches. The outputs of the two streams are fused using fully connected layers, and the output labels are predicted by a softmax layer.

The main novelties and differences compared to the earlier related spectral–spatial learning architectures are the following. First, earlier reported spectral feature extraction streams...
were always based on 1-D vectors, and when employing CNN, those were 1-D CNN networks (see [31], [45], [56]). Our feature extraction stream is instead embedded into a 2-D CNN, which operates on small image patches and extracts simultaneously spectral and local spatial features. An important advantage of this approach is that it leads to an elegant mathematical formulation with a unique objective function, as it will be shown in Section III-D.

Second, we effectively improve the spectral–spatial feature extraction capability by incorporating the SE concept into the two parallel streams. This is especially important in practice where the actual depth of the feature extraction streams is limited by the available amount of the training data. To this end, we shall derive a formal approach for incorporating the SE concept into spectral–spatial classification, as detailed next.

Third, we propose a layer-specific regularization and a smooth normalization fusion scheme, which adaptively controls the fusion weights and better fuses the spectral–spatial features. Finally, while in most of the previous methods, including [42] and [44], feature extraction and classifier parts were trained separately and based on different objective functions, in our framework, feature extraction, fusion, and classification processes are trained simultaneously in an end-to-end training manner from scratch. This unified training is one of the important advantages of our two-stream 2-D CNN framework.

B. Local Feature Extraction Stream

The local feature extraction stream in our architecture (see the bottom of Fig. 2) employs a shallow 2-D CNN to extract spectral and local spatial correlation features simultaneously. The input is a small image patch extending over all the spectral bands and containing thus local spatial information as well as abundant spectral information. While current spectral feature extraction models based on 1-D vectors [31], [42]–[45], [56] omit spatial information, our local stream not only extracts spectral and local spatial information but also makes use of spatial information to learn the spectral band correlations and to boost thereby the feature extraction capability. We accomplish this by incorporating SE similar to [50], but instead of processing RGB images as there, we now employ the SE concept to enhance the feature extraction from a rich spectral content.

The main component of our local feature extraction stream is a convolution layer incorporating the SE module that we denote as SE-Conv. Let $E_l \in \mathbb{R}^{H \times W \times N}$ denote the channel-wise correlations of an SE module, and $E_l(:, :, k) = e^l_k \cdot 1_{H \times W}$. Here, $e^l_k$ is the $k$th element of the correlation vector $e_l = [e^1_l, e^2_l, \ldots, e^N_l]^T$ [see (2)], and $1_{H \times W}$ is an $H$-by-$W$ matrix of all ones. Combining with (1), we define SE-Conv as follows:

$$\tilde{F}^l_i(X) = \delta(W^l_{ij} \ast X) \cdot E^l_i$$

(3)

where $W^l_{ij}$ and $E^l_i$ are the kernels and the channel-wise correlations for the $i$th SE-Conv layer of the local stream.

The idea of SE-Conv is to emphasize useful spectral bands and to suppress less useful spectral bands. In this way, SE-Conv enhances the spectral and local spatial feature extraction capability of the local stream. Fig. 3 shows the architectures of SE and SE-Conv. Specifically, we employ $m > 1$ consecutive SE-Conv modules to extract spectral and local spatial features. A max-pooling layer, in the end, reduces the spatial size and yields more general features at a higher level. Let $I_l \in \mathbb{R}^{P \times P \times B}$ be the input of the local stream with a relatively small window size of $P \times P$. $B$ denotes the number of HSI spectral bands. The output feature vector of the local stream is

$$y_l = \mathcal{M}(\tilde{F}^1_m(I_l) \cdots \tilde{F}^1_2(I_l) \tilde{F}^1_1(I_l))$$

(4)

where $\mathcal{M}$ denotes the max-pooling operation. We do not use any max-pooling layers in between SE-Conv in order to preserve the detailed information.
C. Global Feature Extraction Stream

The global feature extraction stream in our model (see the top of Fig. 2) aims to extract global spatial features from relatively large image patches that extend over a relatively small number of principal components (several to a dozen). The current spatial feature extraction models based on 2-D CNN [41], [44], [45], [49], [57] are largely constrained by limited training data. Our proposed spatial feature extraction stream incorporates SE and residual learning concepts to enhance spatial feature extraction capability and network depth. We do not extract the main spectral feature in this stream because it would not only increase computational cost but also result in spectral features redundancy. An ablation study regarding the number of principal components is given in Section IV-F.

The core components of this stream are SE-Conv with max pooling (denoted as MP-SE-Conv) and SE-based residual learning (denoted as SE-Res). Building on SE-Conv from (3), we define MP-SE-Conv as

\[ \tilde{F}_{i}^{g1}(X) = M(\delta(W_{i}^{g1} * X) \cdot E_{i}^{g1}) \]

where \( W_{i}^{g1} \) and \( E_{i}^{g1} \) are the kernels and the channel-wise correlations for the \( i \)th MP-SE-Conv layer. \( M \) is the max-pooling operation. The idea of MP-SE-Conv is to yield more robust features by identifying more or less informative spatial channels and reducing the spatial size using max pooling. We define SE-Res as

\[ \tilde{F}_{i}^{g2}(X) = \delta((W_{i,1}^{g2} * \delta(W_{i,1}^{g2} * X)) \cdot E_{i}^{g2} + X) \]

where \( W_{i,1}^{g2} \) and \( W_{i,2}^{g2} \) are the two kernels for the \( i \)th SE-Res layer of the global stream, respectively. \( E_{i}^{g2} \) is the corresponding channel-wise correlation. The idea of SE-Res is to learn more complex global spatial features by enhancing the feature extraction capability and increasing the network depth. Let \( I_{g} \in \mathbb{R}^{P \times P \times PC} \) be the input of this stream with a relatively large window size of \( P \times P \) and PC be the number of principal components. The output feature vector of the global stream is

\[ y_{g} = \tilde{F}_{i}^{g1}(I_{g}) \tilde{F}_{i}^{g2}(I_{g}) \mathcal{M}[\tilde{F}_{n}^{g2}(I_{g}) \ldots \tilde{F}_{1}^{g2}(I_{g})] \tilde{F}_{i}^{g1}(I_{g}) \]

where \( n \) is the number of SE-Res modules and \( \mathcal{M} \) is the max-pooling operation. An ablation study regarding the number of SE-Conv and SE-Res is given in Section IV-E. The basic structure of the SE-Res module is shown in Fig. 4. The SE module is inserted before and not after the shortcut connection. This is based on the fact that the main training process of the residual learning module is to train the residual function, and thus, the SE module can better boost the representative power of the residual learning module when training.

D. Feature Fusion Scheme and Classification

Having extracted spectral, and local and global spatial features, we need to fuse them adaptively. The current deep learning feature fusion methods (see [34], [44], [46]) employ fully connected layers with ReLU. We propose instead a layer-specific regularization and smooth normalization fusion scheme. We define the fusion scheme as follows:

\[ y = \sigma(W_{y}^{f} \sigma(W_{y}^{l} (y_{y}) + \lambda \|W_{y}^{l}\|^{2}_{F})) \]

where \( \sigma \) denotes the operation of concatenating, \( W_{y}^{f} \) and \( W_{y}^{l} \) are the kernels of the two fully connected layers, respectively, \( \| \cdot \|^{2}_{F} \) is the Frobenius norm, and \( \lambda \) is the regularization parameter, which adjusts all the fusion weights and further decides the degree of features fusion. An ablation study regarding \( \lambda \) is given in Section IV-C. \( \sigma \) is sigmoid activation function. We choose sigmoid (that smoothly normalizes the fused features to \([0, 1]\)) instead of ReLU to avoid the blow up phenomenon (feature values in \([0, \infty]\)). This choice preserves more detailed features and facilitates the following classification. An L2 kernel regularizer term \( \lambda \|W_{y}^{l}\|^{2}_{F} \) is added in the fusion layer to enable adaptive adjustment of the fusion weights alone. With this layer-specific regularization, instead of a common regularizer on all network weights such as in [31], we avoid overfitting.

Finally, the fused features are fed into the last fully connected layer with \( K \) nodes (classes) following a softmax function to generate the predicted probability vector. The cross entropy objective function is computed as

\[ \mathcal{L} = -\frac{1}{T} \sum_{j=1}^{T} \sum_{k=1}^{K} t_{y}^{k} \log \left( \frac{e^{W_{y}y_{y}^{j}+b_{y}}}{\sum_{k=1}^{K} e^{W_{y}y_{y}^{j}+b_{y}}} \right) \]

where \( T \) is the total number of training samples, \( t_{y}^{k} \) is the \( k \)th value (i.e., 0 or 1) of the one-hot encoding ground truth for the \( j \)th training sample, \( W_{y} \) and \( b_{y} \) are the weights and bias for the \( k \)th unit in this layer, respectively, and \( y_{y}^{j} \) is the input of the \( j \)th training sample. We optimize (9) by using the mini-batch Adadelta [58] optimizer. Observe that the proposed two-stream network has a unique objective function and is trained in an end-to-end training manner from scratch. Thus, the local feature extraction stream and the global feature extraction steam interact during the training process through this unique objective function. This is an important asset of the proposed approach compared to most of the earlier reported ones, including [42]–[44].
IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed method is implemented in Keras\(^1\) and TensorFlow\(^2\) deep learning framework with Python language. All the experiments were repeated ten times with different randomly selected training data, and the average results over the ten runs with standard deviations are reported. Three objective performance indexes are used for evaluation: overall accuracy (OA), average accuracy (AA), and Kappa coefficient ($\kappa$).

A. Data Set Description and Parameter Setting

The experiments were conducted on three well-known HSI data sets: Indian Pines, University of Pavia (PaviaU), and Salinas. The Indian Pines data set is captured by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the agricultural Indian Pines site in Northwestern Indiana in 1992. It contains $145 \times 145$ pixels with 220 spectral bands covering the spectral range from 0.4 to 2.5 $\mu m$ with a spatial resolution of 20 m. It contains 16 ground-truth classes, out of which we select 13 large classes with more than 50 training samples. The PaviaU data set is gathered by the ROSIS-03 sensor over an urban area surrounding the University of Pavia, Pavia, Italy. It consists of $610 \times 340$ pixels with 103 spectral bands covering the spectral range from 0.4 to 2.5 $\mu m$ with a spatial resolution of 1.3 m. It contains 9 classes, out of which we select 13 classes with more than 50 training samples. The Salinas data set is collected by the AVIRIS sensor over the area of Salinas Valley, CA, USA. It composes of $512 \times 217$ pixels with 224 spectral bands covering the spectral range from 0.4 to 2.5 $\mu m$ with a spatial resolution of 3.7 m, and 20 water absorption bands were removed. The numbers of training and testing samples for the three HSIs are listed in Table I. We randomly select 50 labeled samples per class for training. Out of these, 10% are randomly selected and regarded as the validation set. We determine the hyperparameters based on the classification performance in the validation set. The remaining labeled samples are used as the test set to evaluate the classification performance. The estimated optimal values of the hyperparameters are as follows. The optimal initial learning rate is 0.3 for PaviaU image and 1 for the other two images. $\lambda = 0.03$ for Salinas image and $\lambda = 0.02$ for other two images. An ablation study regarding $\lambda$ is given in Section IV-C. The optimal number of principal components in the global stream is 3 for the Indian Pines image and 10 for the other two images, and an ablation study regarding the number of principal components is shown in Section IV-F. The number of training epochs and batch size are empirically set to 400 and 50. The reduction ratio $r$ of the SE module is empirically set to 1. The main network architecture of the proposed method is shown in Table II. The same network architecture is used in all the reported experiments, with all the test images.

B. Comparisons With the State-of-the-Art Method

We compare the performance of the proposed method with several state-of-the-art CNN-based methods for HSI

---

1. https://keras.io/
2. https://www.tensorflow.org/
TABLE III
COMPARISON OF THE CLASSIFICATION ACCURACIES AMONG THE PROPOSED METHOD AND THE BASELINES USING THE INDIAN PINES IMAGE

<table>
<thead>
<tr>
<th>Methods</th>
<th>Indian Pines</th>
<th>PaviaU</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-MRF</td>
<td>76.54±2.42</td>
<td>83.61±1.28</td>
<td>87.82±3.18</td>
</tr>
<tr>
<td>CNN-PPF</td>
<td>85.48±4.88</td>
<td>95.23±3.18</td>
<td>90.71±4.71</td>
</tr>
<tr>
<td>DR-CNN</td>
<td>94.03±2.30</td>
<td>97.46±1.72</td>
<td>96.61±1.66</td>
</tr>
<tr>
<td>MugNet</td>
<td>99.70±0.99</td>
<td>89.04±1.10</td>
<td>89.86±5.87</td>
</tr>
<tr>
<td>DFFN</td>
<td>93.78±1.81</td>
<td>89.78±3.35</td>
<td>92.10±2.83</td>
</tr>
<tr>
<td>DHCNet</td>
<td>92.10±2.83</td>
<td>91.02±3.13</td>
<td>92.10±3.33</td>
</tr>
<tr>
<td>Proposed</td>
<td>92.67±3.28</td>
<td>92.67±3.28</td>
<td>92.67±3.28</td>
</tr>
</tbody>
</table>

TABLE IV
COMPARISON OF THE CLASSIFICATION ACCURACIES AMONG THE PROPOSED METHOD AND THE BASELINES USING THE PAVIAU IMAGE

<table>
<thead>
<tr>
<th>Methods</th>
<th>Indian Pines</th>
<th>PaviaU</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-MRF</td>
<td>76.54±2.42</td>
<td>83.61±1.28</td>
<td>87.82±3.18</td>
</tr>
<tr>
<td>CNN-PPF</td>
<td>85.48±4.88</td>
<td>95.23±3.18</td>
<td>90.71±4.71</td>
</tr>
<tr>
<td>DR-CNN</td>
<td>94.03±2.30</td>
<td>97.46±1.72</td>
<td>96.61±1.66</td>
</tr>
<tr>
<td>MugNet</td>
<td>99.70±0.99</td>
<td>89.04±1.10</td>
<td>89.86±5.87</td>
</tr>
<tr>
<td>DFFN</td>
<td>93.78±1.81</td>
<td>89.78±3.35</td>
<td>92.10±2.83</td>
</tr>
<tr>
<td>DHCNet</td>
<td>92.10±2.83</td>
<td>91.02±3.13</td>
<td>92.10±3.33</td>
</tr>
<tr>
<td>Proposed</td>
<td>92.67±3.28</td>
<td>92.67±3.28</td>
<td>92.67±3.28</td>
</tr>
</tbody>
</table>

The DFFN method can be regarded as a GSFE method and also as a feature fusion method because it uses a relatively large image patch size and fuses the features extracted from different hierarchical layers. The parameters of the reference methods are set to the default values indicated in their original works. For a fair comparison, we use in all experiments the same number of PCA components and the same patch size for DHCNet [57] and for our global stream. To demonstrate the effectiveness of the local stream and the global stream, we also test the networks that only contain the local stream and the global stream.
TABLE V

<table>
<thead>
<tr>
<th>Classes</th>
<th>CNN-MRF</th>
<th>CNN-PPF</th>
<th>DR-CNN</th>
<th>MugNet</th>
<th>SSRN</th>
<th>DPFN</th>
<th>DHCNet</th>
<th>Local</th>
<th>Global</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.81±0.29</td>
<td>99.96±0.08</td>
<td>99.96±0.04</td>
<td>64.91±3.30</td>
<td>100±0</td>
<td>99.81±0.26</td>
<td>99.96±0.06</td>
<td>99.21±1.08</td>
<td>100±0</td>
<td>100±0</td>
</tr>
<tr>
<td>2</td>
<td>97.51±1.24</td>
<td>99.60±0.31</td>
<td>99.59±0.27</td>
<td>99.90±0.10</td>
<td>99.76±0.65</td>
<td>99.83±0.26</td>
<td>99.97±0.07</td>
<td>99.91±0.17</td>
<td>99.98±0.05</td>
<td>99.95±0.10</td>
</tr>
<tr>
<td>3</td>
<td>98.94±1.17</td>
<td>96.86±2.97</td>
<td>99.63±0.13</td>
<td>99.63±0.18</td>
<td>99.65±0.34</td>
<td>99.95±0.16</td>
<td>99.93±0.22</td>
<td>98.87±0.21</td>
<td>99.95±0.09</td>
<td>99.99±0.02</td>
</tr>
<tr>
<td>4</td>
<td>99.30±0.09</td>
<td>97.04±1.61</td>
<td>99.95±0.06</td>
<td>99.87±0.28</td>
<td>98.85±1.03</td>
<td>98.79±1.41</td>
<td>99.69±0.22</td>
<td>99.90±0.11</td>
<td>99.81±0.49</td>
<td>99.99±0.03</td>
</tr>
<tr>
<td>5</td>
<td>97.87±0.60</td>
<td>98.23±0.55</td>
<td>98.75±0.89</td>
<td>99.34±0.37</td>
<td>99.90±0.18</td>
<td>98.93±1.14</td>
<td>99.72±0.21</td>
<td>99.50±0.51</td>
<td>99.23±0.30</td>
<td>99.50±0.44</td>
</tr>
<tr>
<td>6</td>
<td>99.33±0.30</td>
<td>99.80±0.31</td>
<td>99.99±0.02</td>
<td>98.98±0.18</td>
<td>99.99±0.03</td>
<td>99.75±0.31</td>
<td>99.91±0.17</td>
<td>99.99±0.01</td>
<td>99.77±0.65</td>
<td>99.92±0.27</td>
</tr>
<tr>
<td>7</td>
<td>98.67±0.96</td>
<td>99.64±0.77</td>
<td>99.92±0.07</td>
<td>99.31±0.63</td>
<td>99.99±0.02</td>
<td>99.86±0.14</td>
<td>99.72±0.19</td>
<td>99.98±0.04</td>
<td>99.81±0.23</td>
<td>99.99±0.02</td>
</tr>
<tr>
<td>8</td>
<td>76.69±2.49</td>
<td>84.83±2.65</td>
<td>77.25±10.99</td>
<td>99.40±0.34</td>
<td>89.73±5.65</td>
<td>97.36±1.56</td>
<td>95.09±2.52</td>
<td>67.23±19.03</td>
<td>96.49±2.81</td>
<td>96.61±2.70</td>
</tr>
<tr>
<td>9</td>
<td>98.77±0.42</td>
<td>99.29±0.33</td>
<td>99.64±1.04</td>
<td>98.23±1.12</td>
<td>97.94±1.05</td>
<td>99.78±0.32</td>
<td>99.70±0.37</td>
<td>98.84±0.18</td>
<td>99.89±0.26</td>
<td>99.94±0.19</td>
</tr>
<tr>
<td>10</td>
<td>94.88±1.72</td>
<td>89.00±4.25</td>
<td>96.39±0.59</td>
<td>99.12±0.92</td>
<td>97.84±1.37</td>
<td>99.21±0.55</td>
<td>99.77±0.35</td>
<td>97.09±0.92</td>
<td>99.67±0.28</td>
<td>99.75±0.19</td>
</tr>
<tr>
<td>11</td>
<td>99.08±0.78</td>
<td>89.99±6.91</td>
<td>99.97±0.05</td>
<td>94.72±1.56</td>
<td>98.13±2.59</td>
<td>99.22±0.70</td>
<td>99.62±0.43</td>
<td>99.74±0.39</td>
<td>99.47±0.61</td>
<td>99.78±0.34</td>
</tr>
<tr>
<td>12</td>
<td>99.99±0.02</td>
<td>98.79±0.81</td>
<td>100±0</td>
<td>99.29±0.54</td>
<td>99.65±0.46</td>
<td>99.57±0.62</td>
<td>99.87±0.17</td>
<td>99.99±0.02</td>
<td>99.34±0.98</td>
<td>99.58±1.15</td>
</tr>
<tr>
<td>13</td>
<td>99.75±0.35</td>
<td>98.35±1.59</td>
<td>100±0</td>
<td>99.85±0.22</td>
<td>99.95±0.08</td>
<td>99.38±0.84</td>
<td>99.95±0.14</td>
<td>99.84±0.32</td>
<td>99.76±0.32</td>
<td>99.93±0.18</td>
</tr>
<tr>
<td>14</td>
<td>98.25±0.82</td>
<td>95.33±4.41</td>
<td>100±0</td>
<td>98.43±0.69</td>
<td>98.42±1.42</td>
<td>99.89±0.21</td>
<td>99.70±0.53</td>
<td>99.67±0.40</td>
<td>99.94±0.10</td>
<td>99.91±0.14</td>
</tr>
<tr>
<td>15</td>
<td>82.43±3.55</td>
<td>75.02±7.19</td>
<td>90.38±5.19</td>
<td>97.80±1.14</td>
<td>85.41±4.26</td>
<td>97.50±1.75</td>
<td>98.73±0.97</td>
<td>87.96±5.32</td>
<td>99.01±1.23</td>
<td>99.16±1.05</td>
</tr>
<tr>
<td>16</td>
<td>97.44±1.97</td>
<td>99.86±0.79</td>
<td>99.35±0.19</td>
<td>94.02±2.22</td>
<td>98.85±0.28</td>
<td>99.92±0.26</td>
<td>99.92±0.12</td>
<td>99.39±0.42</td>
<td>99.80±0.29</td>
<td>99.99±0.04</td>
</tr>
</tbody>
</table>

AA(%) 96.18±0.41 | 95.11±0.94 | 97.57±0.51 | 95.86±0.16 | 97.93±0.23 | 99.30±0.19 | 99.45±0.20 | 96.82±1.04 | 99.51±0.19 | 99.63±0.20 |

OA(%) 91.66±0.68 | 91.80±1.48 | 93.55±1.87 | 97.74±1.60 | 95.36±1.94 | 98.86±0.24 | 98.67±0.56 | 91.18±3.63 | 98.98±0.53 | 99.09±0.56 |

κ × 100 90.73±0.75 | 90.87±1.64 | 92.84±2.05 | 74.35±1.75 | 94.82±1.06 | 98.73±0.27 | 98.52±0.63 | 90.23±3.97 | 98.87±0.59 | 98.99±0.62 |

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

classification performance compared to MugNet and SSRN designed for small-scale training data. It is also evident that the proposed method yields better accuracy than any of its two streams alone. This is because the local stream extracts the spectral and the local spatial features that are complementary to the global spatial features extracted in the second stream. Thus, the proposed two-stream method has more robust feature representation power and better generalization ability. In terms of the class-specific accuracy, the proposed method performs best or yields comparable results to the best ones in most of the classes for all the three images. Only in several classes, this is not the case. For instance, in the Salinas image, some “Grapes_untrained” samples were misclassified as “Vinyard_untrained” due to their huge spectral similarity and the large within-class variation in their spectral reflectance.

Apart from quantitative analysis, Figs. 5–7 show the full classification maps. Visually, they are consistent with the results reported in Tables III–V. Obviously, the SLSFE methods (e.g., CNN-MRF, CNN-PPF, and the local stream) exhibit noisier estimations than the GSFE methods (DR-CNN, DHCNet, and the global stream). Furthermore, the proposed method presents more similar results to the reference map exhibiting smoother appearance than other reference methods because of more robust spectral and spatial features. In addition, the feature fusion strategy effectively combines the advantages of both streams, e.g., the regions of Meadows and Bare Soil in Fig. 6.

To comprehensively validate the proposed architecture, we also compare the proposed method with several state-of-the-art multistream fusion methods: SdAE-CNN [44],...
Multi-CNN [60], MMFN [61], CSFF [62], a hierarchical architecture MugNet [59], and a very recent graph convolutional method MDGCN [63]. To validate the robustness on more data sets, we include two additional data sets: Pavia Center\(^3\) (PaviaC) and Grss\_dfc\_2013.\(^4\) The optimal values of the hyperparameters of the proposed method for these two data sets are the same as for the PaviaU data set. The results of the reference methods are taken from the original works. To validate the performance with a different sample partitioning method, we also compare with SSRN [40] and DHCNet [57]. The results are given in Table VI, where for the proposed method, we show in brackets the results obtained with the same sample partitioning as in the corresponding reference method. As can be observed in Table VI, the proposed method yields the best OA for all the HSIs compared with all the reference methods. It is worth mentioning that the proposed method exhibits robust classification performance for balanced (e.g., 30 samples per class) and unbalanced (e.g., 1% per class) training samples.

---

\(^3\)Available online: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_scene

\(^4\)Available online: http://www.grss-ieee.org/community/technical-committees/data-fusion/

---

To verify the generalization ability of the proposed method on different numbers of training samples, 50, 100, 150, and 200 samples per class are randomly chosen as training data for three HSIs. For Indian Pines image, following the references
including [32], [34], and [47], we choose eight large classes when the number of training samples is larger than 50. Fig. 8 shows the OA for the proposed method and four kinds of reference methods: 1) SLSFE: CNN-PPF [47]; 2) feature fusion-based: DR-CNN [34]; 3) optimized for small-scale training data: SSRN [40]; and 4) GSFE: DHCNet [57]. Clearly, all the methods yield better classification performance as the number of training samples increases. The proposed method consistently provides superior OA compared with the reference methods for three HSIs. Especially when the number of the labeled training data is limited, the proposed method has an obvious advantage in terms of classification performance over the reference methods.

C. Analysis on Feature Fusion Scheme

To validate the proposed feature fusion scheme, we compare it with ReLU, ReLU with L2, and sigmoid under the same settings, as shown in Tables III–V. The results in Table VII show that both sigmoid and ReLU with L2 regularizer (where $\lambda$ equals 0.02, 0.02, and 0.03 for Indian Pines, PaviaU, and Salinas, respectively) yield better classification performance than without L2 regularizer. Hence, L2 regularizer effectively controls the degree of feature fusion. The scheme with the sigmoid and L2 regularizer performs the best and shows indeed an improvement in OA over ReLU (that was used in earlier reported feature fusion schemes), with gains of 1.11%,
Fig. 8. OA of different methods with different numbers of training samples per class. (a) Indian Pines image. (b) PaviaU image. (c) Salinas image.

Fig. 9. Effect of $\lambda$ on OA for sigmoid and ReLU cases.

Table VIII

<table>
<thead>
<tr>
<th>SE</th>
<th>Indian Pines</th>
<th>PaviaU</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-SE</td>
<td>96.34±0.58</td>
<td>98.07±0.44</td>
<td>98.81±0.86</td>
</tr>
<tr>
<td>SE(=1)</td>
<td>96.75±0.44</td>
<td>98.82±0.40</td>
<td>99.09±0.56</td>
</tr>
<tr>
<td>SE(=4)</td>
<td>96.62±0.63</td>
<td>98.66±0.33</td>
<td>99.01±0.59</td>
</tr>
<tr>
<td>SE(=8)</td>
<td>96.69±0.64</td>
<td>98.62±0.44</td>
<td>98.98±0.70</td>
</tr>
<tr>
<td>SE(=16)</td>
<td>96.53±0.51</td>
<td>98.36±0.57</td>
<td>98.97±0.63</td>
</tr>
<tr>
<td>SE(=128)</td>
<td>96.72±0.52</td>
<td>98.16±0.64</td>
<td>98.89±0.79</td>
</tr>
</tbody>
</table>

D. Analysis of the SE Module

To verify the effectiveness of the SE module, we compare the performance of the proposed method with and without the SE module, as well as with and without the short connection, using different $r$ values. The results are reported in Table VIII. Clearly, the SE module with different values of $r$ consistently performs better than without the SE module for three HSIs. The reason is that the SE module enhances the network feature representation and further improves the classification performance of HSI. The results in Table VIII reveal that the OA does not increase monotonically as $r$ increases for Indian Pines image. A possible reason is that the SE module overfits the feature channel-wise correlations. By contrast, for PaviaU and Salinas images, large $r$ slightly degrades the OA, which means that it underfits the feature channel-wise correlations.

E. Analysis of the Network Depth

We combined a shallow network in the local stream and a deep network in the global stream to extract more robust features (spectral, local spatial, and global spatial features) of HSIs. The network depths for the two streams are, thus, the two key hyperparameters. We fix the other parameters under the same settings, as shown in Tables III–V. As shown in Fig. 10 (L4+G2 denote four SE-Conv modules in the local stream and two SE-Res modules in the global stream, respectively), the results on the PaviaU and the Indian Pines images first improve significantly when the number of SE-Conv modules increases (because they have many small and local regions) and then degrade slightly due to excessive depth and overfitting. By contrast, the result of the Salinas image tends to relatively stable with increasing the network depth because it has many large smooth regions.

Fig. 11 shows the effect of the number of SE-Res modules $n$ in the global stream and the proposed network ($n = 2$) without the short connection (denoted as noRL). Compared with noRL, the proposed provides better classification performance, demonstrating that SE-Res with residual learning mechanism mitigates overfitting problem when the depth of the global
stream increases. Furthermore, the OA indeed increases at first as the number of SE-Res modules increases because deeper network extracts more abstract features, and then, the OA decreases due to overfitting caused by excessive network depth and limited training data. Observe that the OA in the PaviaU image declines dramatically compared with the other two images when the network depth increases. The main reason is that the PaviaU image has more detailed regions.

Based on the earlier analysis, the local stream yields better classification performance on images with many small regions (such as PaviaU) as the depth in the local stream increases. The global stream yields better classification performance on images with many large regions (e.g., the Salinas image) as the depth in the global stream increases due to extracting global spatial features. In addition, we also test a deeper two-stream CNN (dubbed L6 + G4), as shown in Figs. 10 and 11. The results show that the proposed method performs better than this alternative. The proposed network depth settings (L4 + G2) as shown in Table II demonstrate more robustness and better generalizability on the tested data sets.

**F. Analysis of the Patch Size and the Principal Components**

In this section, we discuss the effect of different image patch sizes \( P \) in the two streams on OA. We keep the same settings as in Tables III–V. We adjust the sizes of the max-pooling operations in the two streams for different image patch sizes. Fig. 12 shows the OA versus different values of \( P \) in the local stream varying from \( 3 \times 3 \) to \( 11 \times 11 \) with an interval of 2. The results demonstrate that the OA generally improves at first due to extracting more local spatial features as \( P \) increases and then declines because large \( P \) (e.g., \( 11 \times 11 \)) cannot effectively extract local spatial features. Fig. 13 shows the OA versus different values of \( P \) in the global stream varying from \( 21 \times 21 \) to \( 35 \times 35 \) with an interval of 2. Apparently, large \( P \) gets better or comparative classification accuracy, because it contains more global spatial information, but an overlarge \( P \) increases the computational cost and memory requirements dramatically. \( P \) equals \( 27 \times 27 \) in the global stream as a tradeoff between the classification performance and the running time for three HSIs.

Later, we analyze the effect of the number of principal components in our method under different settings. Fig. 14(a) shows the OA versus the number of principal components \( PC \in \{1, 3, 5, 10, 15, 20\} \) for different HSIs. The OA generally increases and then declines slightly as the number of principal components increases. This is as expected since the first several components contain most of the spatial information. Adding a large number of principal components result in redundant spectral information and require more learning parameters, increasing thereby the computational cost and degrading the classification performance. A sudden drop of OA on PaviaU for \( PC = 3 \) may be attributed to the fact that some classes (e.g., Asphalt and Bitumen) in this image have huge spectral–spatial similarity when \( PC = 3 \), which may result in misclassification.
G. Analysis of the Computational Efficiency

A comparative analysis of the processing time and memory requirements for different representative methods is summarized in Table IX. The training and the testing time are reported together with the memory required (the maximum value during the whole process) for three HSIs. Four kinds of reference methods are used: 1) SLSFE: CNN-PPF [47]; 2) feature fusion-based: DR-CNN [34]; 3) optimized for small-scale training data: SSRN [40]; and 4) GSFE: DHCNet [57].

Table IX

<table>
<thead>
<tr>
<th>Method</th>
<th>Indian Pines</th>
<th>PaviaU</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-PPF</td>
<td>Training (min)</td>
<td>11.4</td>
<td>13.1</td>
</tr>
<tr>
<td></td>
<td>Testing (s)</td>
<td>6.3</td>
<td>26.9</td>
</tr>
<tr>
<td></td>
<td>Memory(GB)</td>
<td>14.3</td>
<td>43.7</td>
</tr>
<tr>
<td>DR-CNN</td>
<td>Training (min)</td>
<td>35.6</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>Testing (s)</td>
<td>110.4</td>
<td>82.2</td>
</tr>
<tr>
<td></td>
<td>Memory(GB)</td>
<td>9.4</td>
<td>37.6</td>
</tr>
<tr>
<td>SSRN</td>
<td>Training (min)</td>
<td>5.7</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>Testing (s)</td>
<td>4.5</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>Memory(GB)</td>
<td>4.6</td>
<td>12.1</td>
</tr>
<tr>
<td>DHCNet</td>
<td>Training (min)</td>
<td>2.6</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>Testing (s)</td>
<td>5.1</td>
<td>22.6</td>
</tr>
<tr>
<td></td>
<td>Memory(GB)</td>
<td>2.8</td>
<td>26.4</td>
</tr>
<tr>
<td>Proposed</td>
<td>Training (min)</td>
<td>2.9</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>Testing (s)</td>
<td>5.6</td>
<td>22.2</td>
</tr>
<tr>
<td></td>
<td>Memory(GB)</td>
<td>5.1</td>
<td>25.6</td>
</tr>
</tbody>
</table>

As an implementation detail, it should be noted that we employ batch normalization layers [52], residual learning mechanism [55], and a clever strategy for terminating the training process and reducing the learning rate, which enables us to use a larger initial learning rate. Fig. 15 shows the evolution of the training and validation losses and the corresponding learning rate for a particular test image (Indian Pines). Similar trends hold for other test images. It can be seen that the training and the validation losses converge quickly (in around 100 epochs) and terminate in advance (in less than 400 epochs). The large initial learning rate (i.e., 1) decreases quickly, converging (in around 100 epochs) to a stable value.

V. Conclusion

In this article, we proposed a novel two-stream spectral and spatial feature extraction and fusion architecture based on 2-D CNN for HSI classification. The proposed method simultaneously extracts spectral, local, and global spatial features via a shallow and a deep 2-D CNN networks. Inspired by SE networks, we developed a formal approach to enhance the spectral–spatial feature extraction capability based on interband correlations. This approach improves significantly the classification performance, especially when the amount of the available training data is limited. In addition, we proposed a layer-specific regularization and smooth normalization fusion scheme to adaptively fuse the spectral–spatial features of the two streams. Experimental results on several HSIs demonstrated the state-of-the-art classification performance.

ACKNOWLEDGMENT

The authors would like to thank Dr. W. Song and Dr. B. Pan for providing the software for the DFFN method [46] and the MugNet method [59], respectively. They would like to thank the associate editor and the anonymous reviewers for their insightful comments and helpful suggestions that have greatly improved this article. They would also like to thank the Hyperspectral Image Analysis Group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the grss_dfc_2013 data set used in this article, and the data fusion technical committee for their preparation and preprocess for this data set.

REFERENCES


