Attention-Based Adaptive Spectral–Spatial Kernel ResNet for Hyperspectral Image Classification

Swalpa Kumar Roy, *Student Member, IEEE*, Suvojit Manna, Tiecheng Song, *Member, IEEE*, and Lorenzo Bruzzone, *Fellow, IEEE*

***Abstract*— Hyperspectral images (HSIs) provide rich spectral–spatial information with stacked hundreds of contiguous narrowbands. Due to the existence of noise and band correlation, the selection of informative spectral–spatial kernel features poses a challenge. This is often addressed by using convolutional neural networks (CNNs) with receptive field (RF) having fixed sizes. However, these solutions cannot enable neurons to effectively adjust RF sizes and cross-channel dependencies when forward and backward propagations are used to optimize the network. In this article, we present an attention-based adaptive spectral–spatial kernel improved residual network (A^{2}S^{2}K-ResNet) with spectral attention to capture discriminative spectral–spatial features for HSI classification in an end-to-end training fashion. In particular, the proposed network learns selective 3-D convolutional kernels to jointly extract spectral–spatial features using improved 3-D ResBlocks and adopts an efficient feature recalibration (EFR) mechanism to boost the classification performance. Extensive experiments are performed on three well-known hyperspectral data sets, i.e., IP, KSC, and UP, and the proposed A^{2}S^{2}K-ResNet can provide better classification results in terms of overall accuracy (OA), average accuracy (AA), and Kappa compared with the existing methods investigated. The source code will be made available at https://github.com/suvojit-0 × 55aa/A2S2K-ResNet.**

***Index Terms*— Channel attention, convolutional neural networks (CNNs), hyperspectral images (HSIs), image classification, receptive field (RF), residual network (ResNet).**

I. Introduction

HYPERSPECTRAL images (HSIs) contain rich spectral–spatial information that is encoded in many narrow and contiguous spectral bands. HSI analysis has been applied in several fields related to Earth observation, such as greenery detection, urbanization analysis, and crop area analysis [1]–[4]. Due to the large number of spectral bands, feature redundancy and the curse of dimensionality should be addressed in remote sensing HSI analysis. To tackle the abovementioned challenges, dimentionality reduction techniques, such as band selection [5], [6] and subspace learning [7], have been widely used for HSI classification [8], [9] and denoising [10].

The conventional approaches using handcrafted features [11]–[13] cannot effectively extract discriminative yet robust information for HSI classification. In the last decade, deep learning-based methods have proved very effective [14]–[16]. In this context, convolutional neural network (CNN) acts as an automatic kernelized feature extractor via deploying a series of hierarchical filtering layers. The 2-D CNN stacked autoencoder [17] is the first attempt to extract deep features from its compressed latent space to classify hyperspectral data. Zhao *et al.* [18] exploited a multiscale sparse deep spatial feature representation for HSI classification from the latent space of stacked sparse autoencoder (SSAE). Li *et al.* [19] introduced a deep belief network to jointly extract deep and abstract features for HSI classification with increased computation cost. Makantasis *et al.* [20] designed a 2-D CNN model to extract the spatial information and successfully classify the raw HSI cubes in a supervised manner. The layer-wise feature fusion without attention [21], the multibranch selective kernel (SK) network with attention [22], and the data augmentation strategy [23], [24] were employed to deal with gradient vanishing and overfitting problems.

The use of spectral information alone is difficult to recognize the structure types of surface materials in urban analysis [25]. To achieve better classification results, it is desirable to extract spatial–spectral features jointly from raw HSI. Multiple kernel learning (MKL) [26] has been used to effectively handle the heterogeneous kernels in both spectral–spatial direction for HSI classification. Recently, the encoding of spatial–spectral features through CNN has gained a lot of attention for HSI classification. The 3-D CNN simultaneously extracts spectral–spatial features and thus significantly improves the accuracy [27]. Yang *et al.* [28] proposed a two-branch architecture to extract the joint spectral–spatial features for classification. Hamida *et al.* [29] presented a 3-D deep learning architecture, which takes a 3-D volume as input to extract joint spatial–spectral features. The performance of this model can be further improved by considering

Manuscript received November 7, 2020; accepted December 3, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61702065 and in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2018jcyjAX0033. *(Corresponding author: Tiecheng Song.)*

Swalpa Kumar Roy is with the Computer Science and Engineering Department, Jalpaiguri Government Engineering College, Jalpaiguri 735102, India (e-mail: swalpa@cse.jgec.ac.in).

Suvojit Manna is with CureSkin, Bengaluru 560102, India (e-mail: suvojit@heallo.ai).

Tiecheng Song is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: songtc@cqupt.edu.cn).

Lorenzo Bruzzone is with the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: lorenzo.bruzzone@unitn.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TGRS.2020.3043267.

Digital Object Identifier 10.1109/TGRS.2020.3043267

multiscale information [30]. Lee and Kwon [31] created multiscale spatiospectral relationships using 3-D CNN and fused the features using 2-D CNN. Song *et al.* [21] proposed a deep feature fusion strategy by considering the correlated information among different layers and using residual learning to reduce network overfitting and gradient vanishing. Zhong *et al.* [32] introduced a supervised spectral–spatial ResNet (SSRN) that uses a series of 3-D convolutions in the respective residual blocks to extract discriminative joint representation. Recently, Zhu *et al.* [33] proposed RSSAN by introducing a spectral–spatial attention layer to SSRN. In our prior work [34], we proposed HybridSN using the sequential arrangement of 3-D and 2-D CNNs to extract more robust representation of spectral–spatial information. The attention mechanism has been widely explored in vision community for channel weighting [35]–[38] and has also been recently adopted in the HSI domain. To improve the performance of squeeze-and-excitation network (SENet) [36] for HSI classification, we proposed FuSENet [39] using two bilinear SENets with different squeezing strategies, i.e., global average pooling (GAP) and global max pooling (GMP). We also proposed S3ResBoF [40] to reduce the trainable parameters by introducing a bag-of-features encoding layer before the fully connected layer. Yu *et al.* [41] proposed a CNN model with three convolutions and fused layer-wise features to gain better classification accuracy. Han *et al.* [42] utilized a two-stream CNN to learn the spatial–spectral features at different scales. Zhou *et al.* [43] introduced a compact and discriminative autoencoder to progressively learn a low-dimensional feature mapping and an effective classifier. Kang *et al.* [44] defined a dual-path network by combining ResNet and DenseNet to learn joint features extracted from the bottleneck layer. Fang *et al.* [45] introduced a deep hashing neural network for hyperspectral image (HSI) feature extraction and fast classification. Paoletti *et al.* [46] developed a deep pyramidal residual network (ResNet) (DPyResNet) and a CNN architecture based on Hinton's CapsNets [47] for fast and accurate HSI classification. Ding *et al.* [48] introduced LANet using local attention embedding to improve semantic segmentation of remotely sensed images. Recently, Hong *et al.* [49] introduced graph convolution network (GCN) to model relations between samples for HSI classification. They also investigated different strategies to fuse GCNs and CNNs to make them more suitable for HS image classification.

The receptive field (RF) or kernel size, which is the region that directly affects the convolution operation, plays a key role in deep CNN models [22], [50]–[52]. To the best of our knowledge, most of the existing CNN models for HSI analysis use the fixed size of RFs for feature extraction. This restricts the model to learn weights without addressing the problem of selecting RFs during model training. In general, larger RFs cannot capture fine-grained structures, whereas too small RFs tend to eliminate coarse-grained image structures [52], [53]. In this article, we explore A^{2}S^{2}K-ResNet, an attention-based adaptive spectral–spatial kernel improved ResNet for HSI classification. In A^{2}S^{2}K-ResNet, the RF size of neurons is automatically adjusted and cross-channel relationships among features are strengthened by introducing an efficient feature recalibration (EFR) mechanism into modified ResBlocks, thus boosting the classification accuracy. The contributions of this article are summarized as follows.

1) An attention-based adaptive spectral–spatial kernel module is introduced for the first time to learn selective 3-D convolutional kernels for HSI classification.
2) An improved spectral–spatial ResNet is designed to extract the joint spectral–spatial features. Meanwhile, an EFR mechanism is adopted to better capture nonlinear cross-channel interdependencies of the transformed feature maps.
3) The proposed network architecture improves the feature representation ability and achieves the state-of-the-art classification accuracy on three benchmark data sets (i.e., IP, KSC, and UP) using limited training samples.

The rest of this article is organized as follows. Section II introduces the proposed network architecture of A^{2}S^{2}K-ResNet. Section III reports the classification results. Finally, Section IV draws the conclusion.

TABLE I
DESCRIPTION OF THE SYMBOLS USED THROUGHOUT THIS ARTICLE

Symbols	Description
$X_{orig} \in \mathcal{R}^{H\times W\times B}$	Original hyperspectral dataset
$X \in \mathcal{R}^{S\times S\times B}$	Stacked extracted 3-D patch of size $S \times S \times B$
$\mathcal{F}_{A^2S^2K}(\cdot)$	Attention-based adaptive spectral-spatial kernel module (A^{2}S^{2}K)
$\mathcal{F}_{EFR}(\cdot)$	Efficient feature recalibration (EFR) module
$V \in \mathcal{R}^{S\times S\times B}$	Output of the $\mathcal{F}_{A^2S^2K}(\cdot)$ or selected spectral-spatial feature
$\widetilde{\mathcal{F}}^{(l+1)}_{spectral} : X^l \to \widetilde{U}^{(l+1)}$	Spectral transformation function with input X^l and output $\widetilde{U}^{(l+1)}$ both $\in \mathcal{R}^{S\times S\times B}$
$\widetilde{\mathcal{F}}^{(l+1)}_{spatial} : X^l \to \widetilde{U}^{(l+1)}$	Spatial transformation function with input X^l and output $\widetilde{U}^{(l+1)}$ both $\in \mathcal{R}^{S\times S\times B}$
$W^{(l+1)}_{(1\times1\times7)}$ and $W^{(l+1)}_{(3\times3\times7)}$	Weights for the kernel of size $(1 \times 1 \times 7)$ and $(3 \times 3 \times 7)$ respectively
$\mathcal{F}_{bn}(X^l_j)$	Batch normalization operation on X^l_j
$s^{l+1}_b \in \mathcal{R}^{1\times1\times B}$	Descriptor of b^{th} channel at $(l+1)^{th}$ layer of size $(1 \times 1 \times B)$
z^{l+1}	Compact representation of channel descriptor in $(l+1)^{th}$ layer
$\mu(X^l_j)$	Batch-wise mean of the input feature X^l_j
$\sigma^2(X^l_j)$	Batch-wise variance of the input feature X^l_j
$\mathcal{F}_{fc}(s^{l+1}_b)$	Fully connected (FC) layer where s^{l+1}_b is the input vector
$m^{l+1}_{spectral}$	Spectral kernel soft attention vector
$n^{l+1}_{spatial}$	Spatial kernel soft attention vector
$G(X^l)$	Global average pooling on input X^l
ω	Feature recalibration vector (FRV)
$Conv1D_k(y)$	1-D convolution with kernel size k on input y
$\sigma(\cdot)$	Sigmoid activation function
$\mathcal{F}_{scale}(\cdot)$	Scale function
$\mathcal{F}_{res}(\cdot)$	Feed-forward residual function
$\widetilde{X}^{l+1}$	Output of the $\mathcal{F}_{EFR}(\cdot)$ attention module
$\mathcal{I}(X^l_j) = X^l_j$	Identity function
γ and β	Learnable vector parameters
$\hat{u}_{i,j,c}$ and $\tilde{u}_{i,j,c}$	$\hat{u}_{i,j,c} \in \widehat{U}^{(l+1)}$ and $\tilde{u}_{i,j,c} \in \widetilde{U}^{(l+1)}$
$y^m_{L_c} \log \hat{y}^m_{L_c}$	Cross Entropy

II. PROPOSED METHOD

Let $X_{\text{orig}} \in \mathcal{R}^{H\times W\times B}$ be a spectral–spatial 3-D HSI with height H, width W, and B spectral channels. All the pixels $\mathbf{x}_{i,j} \in X_{\text{orig}}$, where $i = 1, \ldots W$ and $j = 1, \ldots H$, are grouped into L_c land-cover classes represented by $Y = y_1, y_2, \ldots y_{L_c}$. The joint spectral and spatial information is used by considering the regions of size $S \times S$ centered at pixel (i, j) and stacking them into $\mathbf{X}$, which can be defined as a spectral–spatial vector $X_{i,j} = [x_{i,j,1}, \ldots x_{i,j,B}] \in \mathcal{R}^{S\times S\times B}$. The proposed HSI classification framework is composed of three main steps: 1) selection of spectral–spatial kernel attention feature maps; 2) recalibration of the spectral–spatial feature maps in the spectral dimension using spectral attention based improved ResNet; and 3) classification using a $softmax$-based fully connected layer. Table I lists the used symbols throughout this article. Fig. 1 shows the framework of the proposed A^{2}S^{2}K-ResNet network, which is detailed in the following sections.

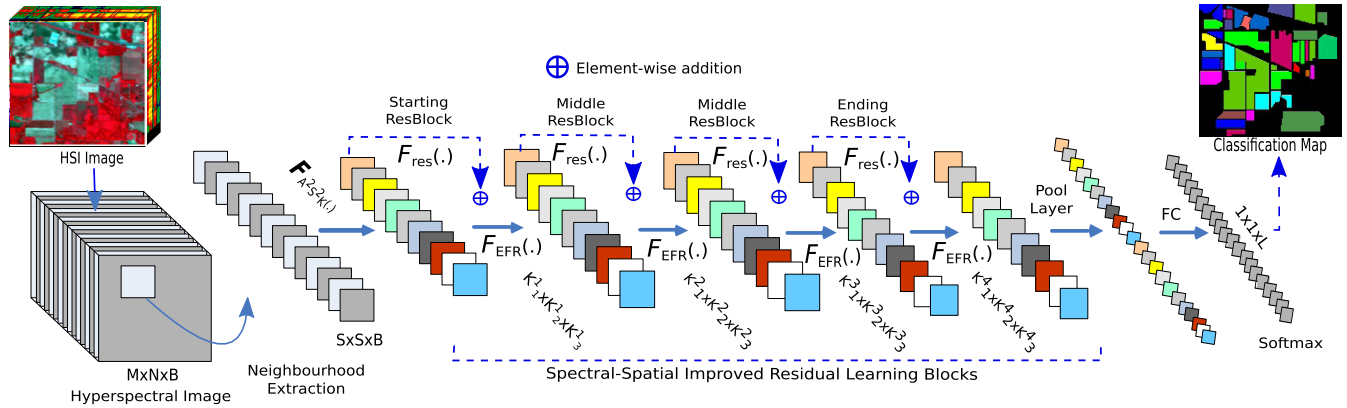


Fig. 1. Framework of the proposed network for HSI classification. Given a raw 3-D input patch of size $S \times S \times B$, where B is the number of spectral bands, the input is first passed through $\mathcal{F}_{A^2S^2K}(\cdot)$ to learn the selection of spectral–spatial kernel among different RFs. Then, discriminative features are extracted through $F_{\text{res}}(\cdot)$ and recalibrated through the spectral–spatial channel attention $\mathcal{F}_{\text{EFR}}(\cdot)$. Finally, the classification is performed using the $softmax$ layer.

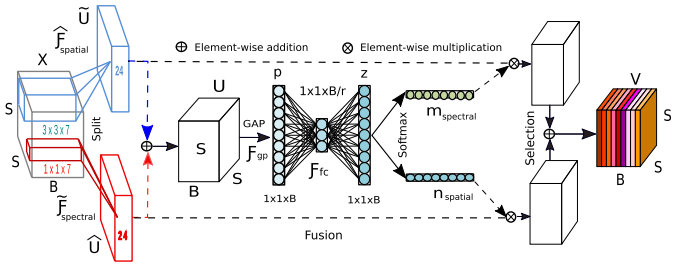


Fig. 2. Adaptive spectral–spatial kernel attention module.

A. Attention-Based Adaptive Spectral–Spatial Kernel Module

To gain robust performance for HSI classification, it is required to address the challenging task of jointly extracting spectral–spatial feature maps while automatically adjusting the RFs of the network. This can be achieved by properly selecting convolutional kernels from different sizes of RFs [52]. Fig. 2 show the attention-based adaptive spectral–spatial kernel (A^2S^2K) module that consists of three main operations, i.e., kernel split, fusion, and selection. The whole module takes HSI cube $X \in \mathcal{R}^{S\times S\times B}$ as input and produces an adaptive spectral–spatial kernel feature map $V \in \mathcal{R}^{S\times S\times B}$ as output

$$V = \mathcal{F}_{\text{A}^2\text{S}^2\text{K}}(X;\theta_a) \tag{1}$$

where θ_a denotes trainable parameters in A^2S^2K. The details of this module are introduced as follows.

Kernel Split: Let $X \in \mathcal{R}^{S\times S\times B}$ be the input HSI cube and $\widehat{\mathcal{F}}^{(l+1)}_{\text{spectral}} : X^l \to \widehat{U}^{(l+1)} \in \mathcal{R}^{S\times S\times B}$ and $\widetilde{\mathcal{F}}^{(l+1)}_{\text{spatial}} : X^l \to \widetilde{U}^{(l+1)} \in \mathcal{R}^{S\times S\times B}$ the $(l+1)$th layer transformations, where X^l is the input to spectral and spatial kernel selection transformation in the $(l+1)$th layer. Zero padding is used for the sake of dimensionality constraint during convolution operations. Two output feature maps $\widehat{U}^{(l+1)}$ and $\widetilde{U}^{(l+1)}$ are defined as

$$\begin{aligned}\widehat{U}^{(l+1)} &= \widehat{\mathcal{F}}^{(l+1)}_{\text{spectral}}(X^l) = X^l * W^{(l+1)}_{(1\times1\times7)} + b^{(l+1)}\\ \widetilde{U}^{(l+1)} &= \widetilde{\mathcal{F}}^{(l+1)}_{\text{spatial}}(X^l) = X^l * W^{(l+1)}_{(3\times3\times7)} + b^{(l+1)}\end{aligned} \tag{2}$$

where $*$ denotes the 3-D convolution operation, W^{l+1} and $b^{(l+1)}$ are weights and biases of the $(l+1)$th convolutional layer, respectively, and two 3-D convolutional kernels having RF sizes $(1\times1\times7)$ and $(3\times3\times7)$ are used to extract spectral and spatial feature maps. The transformation $\widehat{\mathcal{F}}_{\text{spectral}}$ extracts the spectral features, whereas $\widetilde{\mathcal{F}}_{\text{spatial}}$ is associated with the spatial features. The 3-D convolution in the $(l+1)$th layer processes the jth raw of the HSI cube $X^l_j \in \mathcal{R}^{S\times S\times B}$ as input. The layer contains $x^{(l+1)}$ trainable filters of size $k^{l+1}\times k^{l+1}\times d^{l+1}$ with a stride size (s_1, s_1, s_2) in the height, width, and spectral depth dimensions. The size of the output feature map in the $(l+1)$th 3-D convolutional layer is $S^{l+1}\times S^{l+1}\times B^{l+1}$, where height, width, and spectral depth values are determined as $S^{l+1} = \lfloor 1 + (S^l - k^{l+1})/s_1 \rfloor$ and $B^{l+1} = \lfloor 1 + (B^l - d^{l+1})/s_2 \rfloor$, respectively. The output of the αth feature map in the $(l+1)$th layer through 3-D convolution+batch normalization ($ConvBN$) can be mathematically defined as

$$\begin{aligned}X^{l+1}_\alpha &= \text{ReLU}\left(\sum_{j=1}^{x^l} \mathcal{F}_{bn}\left(X^l_j\right) * W^{l+1}_\alpha + b^{l+1}_\alpha\right)\\ \mathcal{F}_{bn}\left(X^l_j\right) &= \frac{X^l_j - \mu\left(X^l_j\right)}{\sqrt{\sigma^2\left(X^l_j\right)+\epsilon}}.\gamma + \beta\end{aligned} \tag{3}$$

where ReLU is an activation function [54] and $\mathcal{F}_{bn}(X^l_j)$ is the BN for the lth layer of the jth feature X^l and $\mu(\cdot)$ and $\sigma^2(\cdot)$ are the batch-wise mean and variance of the input feature maps, respectively. W^{l+1}_α and b^{l+1}_α are kernel parameters and bias of the αth filter bank in the $(l+1)$th layer, while γ and β are learnable vector parameters.

Fusion: The goal of A^2S^2K is to enable neurons to jointly learn spectral–spatial features by automatically adjusting their RF sizes and to amplify the flow of multiscale information in the next layers of neurons. To fulfill the goal, similar to [52], the output feature maps, i.e., the transformations $\widehat{\mathcal{F}}_{\text{spectral}}$ and $\widetilde{\mathcal{F}}_{\text{spatial}}$, are fused via an element-wise addition ($\oplus$) between $\hat{u}_{i,j,c} \in \widehat{U}^{(l+1)}$ and $\tilde{u}_{i,j,c} \in \widetilde{U}^{(l+1)}$ to produce an output $U^{(l+1)} \in \mathcal{R}^{S\times S\times B}$ given by

$$U^{(l+1)} = \widehat{U}^{(l+1)} \oplus \widetilde{U}^{(l+1)} \tag{4}$$

where S is the height and width of the feature maps and B is the total number of channel feature maps in the volume

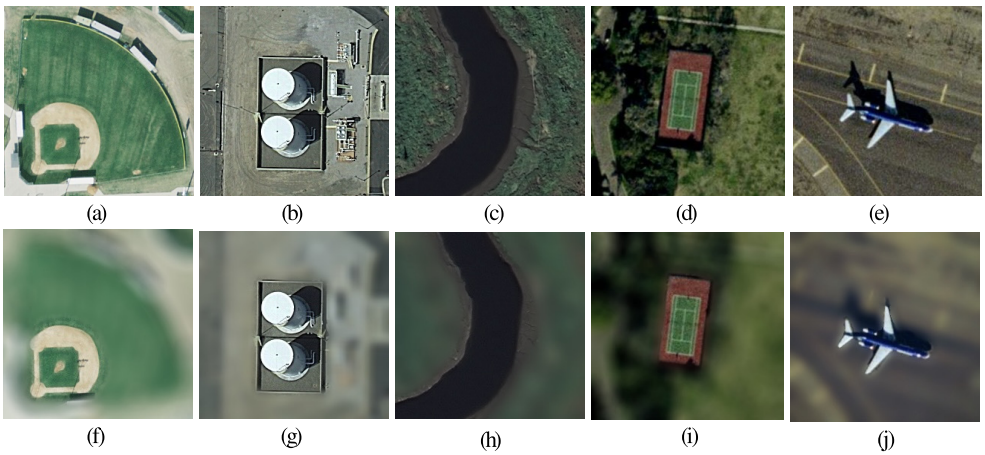


Fig. 3. Illustrating the adaptive spectral–spatial kernel attention as shown in bottom row [(f)–(j)] on some example images given in top row [(a)–(e)] taken from UC Merced Land Use Data set (http://weegee.vision.ucmerced.edu/data sets/landuse).

$\widehat{U}^{(l+1)}$, $\widetilde{U}^{(l+1)} \in \mathcal{R}^{S\times S\times B}$. In order to exploit the feature dependencies obtained from various sizes of RFs, the channel-wise descriptor s_b^{l+1} is computed by simply performing GAP on the feature maps, $U^{(l+1)} \in \mathcal{R}^{S\times S\times B}$ to squeeze the spatial dimension of $U^{(l+1)}$ to $s_b^{l+1} \in \mathcal{R}^{1\times 1\times B}$ along the direction of the bth feature map

$$s_b^{l+1} = \frac{1}{S\times S}\sum_{i=1}^{S}\sum_{j=1}^{S} u_{i,j,b}^{(l+1)}. \tag{5}$$

To learn compact representations and capture nonlinear cross-channel interactions, the channel descriptor s_b^{l+1} is fed into a fully connected (FC) layer with an associated weight matrix $\mathbf{W}^{(l+1)} \in \mathcal{R}^{d\times B}$ followed by an activation function ReLU [54]. The compact representation of the channel descriptor can be computed as

$$z^{(l+1)} = \mathcal{F}_{fc}\left(s_b^{l+1}\right) = \mathrm{ReLU}\left(\mathbf{W}^{(l+1)}\cdot s_b^{l+1}\right) \tag{6}$$

where $z^{(l+1)} \in \mathcal{R}^{d\times 1}$ plays an important role in supervising the adaptive selection of spectral–spatial kernels based on the channel-wise statistics. The key parameter $d = \max(B/r, L)$ is used to achieve model convergence, where r is a reduction ratio of $z^{(l+1)}$ that helps to compress the dimension and L is the minimum value of d experimentally set to 32 [52].

Selection: The automatic selection of discriminative spectral–spatial kernel feature maps is guided by the compact representation of channel descriptor $z^{(l+1)}$ with adaptive RF sizes. Specifically, $z^{(l+1)}$ is applied to a $softmax$ function to calculate the spectral–spatial kernel attention vector (S^2KAV)

$$m_{\text{spectral}}^{l+1} = \frac{e^{\mathbf{M}_b^{(l+1)}z^{(l+1)}}}{e^{\mathbf{M}_b^{(l+1)}z^{(l+1)}} + e^{\mathbf{N}_b^{(l+1)}z^{(l+1)}}} \tag{7a}$$

$$n_{\text{spatial}}^{l+1} = \frac{e^{\mathbf{N}_b^{(l+1)}z^{(l+1)}}}{e^{\mathbf{M}_b^{(l+1)}z^{(l+1)}} + e^{\mathbf{N}_b^{(l+1)}z^{(l+1)}}} \tag{7b}$$

where $m_{\text{spectral}}^{l+1}$ and n_{spatial}^{l+1} denote the soft S^2KAV for $\widehat{U}^{(l+1)}$ and $\widetilde{U}^{(l+1)}$, respectively. Here, $\mathbf{M}_b^{(l+1)} \in \mathcal{R}^{1\times d}$ and $\mathbf{N}_b^{(l+1)} \in \mathcal{R}^{1\times d}$ are the bth row of $\mathbf{M}^{(l+1)} \in \mathcal{R}^{B\times d}$ and $\mathbf{N}^{(l+1)} \in \mathcal{R}^{B\times d}$, respectively. Then, the kernel recalibrated spectral–spatial feature V

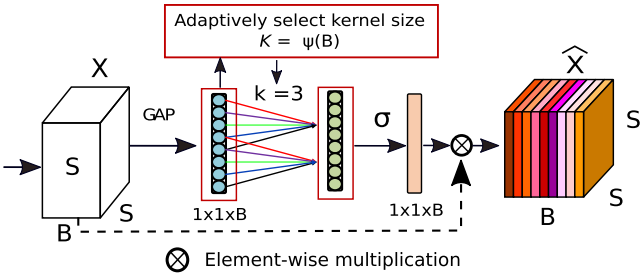


Fig. 4. EFR module.

is calculated by applying soft S^2KAV $m_{\text{spectral}}^{l+1}$ and n_{spatial}^{l+1} along with the spectral dimensions as follows:

$$V = \left(m_{\text{spectral}}^{l+1} \otimes \widehat{U}^{(l+1)}\right) \oplus \left(n_{\text{spatial}}^{l+1} \otimes \widetilde{U}^{(l+1)}\right)$$
$$\text{subject to } \left(m_{\text{spectral}}^{l+1} + n_{\text{spatial}}^{l+1}\right) = 1 \tag{8}$$

where $V = [v_1, v_2, \ldots, v_B]$ and $v_i \in \mathcal{R}^{S\times S}$, $\forall\ i = 1, \ldots, B$. The diagram representing the input shape, the kernel size, and the output feature maps is shown in Fig. 2. In addition, the adaptive spectral–spatial kernel attention is shown on some example images in Fig. 3, demonstrating that RFs can highlight spatial locations of the objects while eliminating irrelevant feature information.

B. Efficient Feature Recalibration Module

To gain improved performance, it is important to capture long range nonlinear cross-channel interdependencies of the transformed feature maps. This can be achieved through feature recalibration or spectral attention mechanism. In this work, we make use of an efficient spectral attention mechanism introduced in [55]. As shown in Fig. 4, the EFR module takes the lth layer transformed feature map $X^l \in \mathcal{R}^{S\times S\times B}$ as input and produces a channel recalibrated feature map $\widehat{X}^{l+1} \in \mathcal{R}^{S\times S\times B}$ as output, that is

$$\widehat{X}^{l+1} = \mathcal{F}_{\text{EFR}}(X^l; \theta_b) \tag{9}$$

where θ_b denotes the trainable parameters in the EFR module.

More specifically, we first calculate a descriptor that characterizes each channel using a spatial squeeze operation via a GAP

$$y = G(X^l) = \frac{1}{S \times S} \sum_{i,j=1}^{S} x_{i,j,c} \tag{10}$$

where $x_{i,j,c} \in X^l$ represents the cth channel of the feature map, $G(X^l)$ denotes the GAP function, and $y \in \mathcal{R}^{1\times 1\times B}$ is the channel-wise descriptor. To find the useful feature maps, the descriptor y is reweighted along the channel dimension. The dimensionality loss has a direct side effect on the appropriate channel prediction and fails to capture proper cross-channel interrelationship across all the channels in feature maps [36]. On the contrary, $\mathcal{F}_{\text{EFR}}(\cdot)$ block can capture interchannel dependencies across all the channels using a 1-D convolution by considering k local neighborhoods. To increase the discriminative power of intermediate feature selection, it is desirable to find the minimum participating local neighbors in the prediction of a channel by adaptively selecting the kernel of size k. As shown in [55], the best weights of $y_i \in y$ $(i = 1, \ldots, B)$ can be obtained by performing a linear interaction between each channel and its k neighbors, where all the channels share the same leaning parameters, that is,

$$\omega_i = \sigma\left(\sum_{j=1}^{k} \beta^j \cdot y_i^j\right), \quad y_i^j \in \Omega_i^k \tag{11}$$

where Ω_i^k denotes a collection of k adjacent channels y_i and β^j represents the shareable weight associated with each y_i^j. Equation (11) can be implemented by a fast 1-D convolution. $\omega \in \mathcal{R}^B$ is a feature recalibration vector (FRV), which emphasizes multiple convolutional channel features by transforming X^l to $\widehat{X}^{l+1}$ with a little computational overhead but brings significant improvement for HSI classification. The recalibrated feature map $\widehat{X}^{l+1} \in \mathcal{R}^{S\times S\times B}$ is produced by performing the channel-wise multiplication as follows:

$$\widehat{X}^{l+1} = \mathcal{F}_{\text{scale}}(x_c \cdot \omega_c) \quad \forall_{c\in[1,\ldots,B]} \tag{12}$$

where $x_c \in X^l$ and $X^l = [x_1, x_2, \ldots, x_B]$.

C. Modified Spectral–Spatial Residual Network

The ResNet and its variants [55]–[61] are powerful CNN, which can well deal with the vanishing gradient problem. To propagate information backward and forward in the network, the original ResNet is formed by depth-wise stacking the residual building blocks (ResBlocks) together. The flows of original and preactivation ResBlocks are shown in Fig. 5(a) and (b).

To extracts more robust and discriminative representation of spectral–spatial features, the spectral–spatial kernel feature map $X \in V$ selected using (8) is passed through the ResBlocks. The ResBlocks, the input is passed through two consecutive 3-D convolution operations parameterized with the feedforward residual function $\mathcal{F}_{\text{res}}(X_j^l; \theta_1, \theta_2)$ using filter banks $W_j = \{W^{l+i} | 1 \le i \le 2\}$ of kernel size $k_1^l \times k_2^l \times k_3^l$ in the $(l+1)$th and $(l+2)$th layers. Then, as shown in Fig. 5, $\mathcal{F}_{\text{res}}(X_j^l; \theta_1, \theta_2)$ goes into an EFR module $\mathcal{F}_{\text{EFR}}(\cdot)$ defined

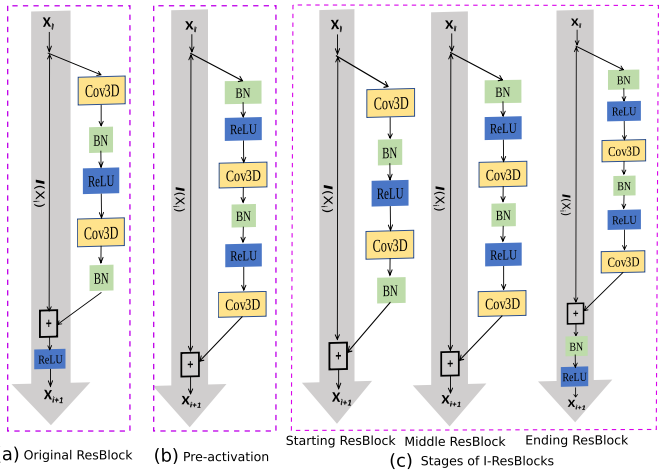


Fig. 5. Stages of various residual blocks. (a) ResNet block. (b) Preactivation block. (c) I-ResNet block.

TABLE II
CONFIGURATION OF THE KERNEL SHAPES IN EACH RESBLOCK OF THE PROPOSED NETWORK

Model	ResBlock	# of Kernels	Kernel shapes $(k_1^i \times k_2^i \times k_3^i)$	Stride shapes $(s_1^i \times s_2^i \times s_3^i)$
I-ResNet	Starting	24	(1, 1, 7)	(0, 0, 3)
	Middle		(1, 1, 7)	(0, 0, 3)
	Middle		(3, 3, 1)	(1, 1, 0)
	Ending		(3, 3, 1)	(1, 1, 0)

in (9) and an identity mapping $\mathcal{I}(X_j^l) = X_j^l$ is directly added as a shortcut connection, that is,

$$2X_j^{l+2} = \mathcal{I}(X_j^l) + \mathcal{F}_{\text{EFR}}\left(\mathcal{F}_{\text{res}}\left(X_j^l; \theta_1, \theta_2\right)\right) \tag{13}$$

$$\mathcal{F}_{\text{res}}\left(X_j^l; \theta_1, \theta_2\right) = \text{ReLU}\left(X_j^{l+1}\right) * W_j^{l+2} + b_j^{l+2} \tag{14}$$

$$X_j^{l+1} = \text{ReLU}\left(X_j^l\right) * W_j^{l+1} + b_j^{l+1} \tag{15}$$

where X^{l+1} and X^{l+2} are output feature maps after the 3-D convolutions in the $(l+1)$th and $(l+1)$th layers, respectively, $\theta_1 = \{W_j^{l+1}, W_j^{l+2}\}$, and $\theta_2 = \{b_j^{l+1}, b_j^{l+2}\}$. $W_j = \{W^{l+i} | 1 \le i \le 2\}$ and $b_j = \{b_j^{l+i} | 1 \le i \le 2\}$ denote the weight matrix and bias vector associated with the jth ResBlock in the $(l+1)$th and $(l+2)$th $ConvBN$ layers, respectively.

To increase the projection power of consecutive residual blocks $\mathcal{F}_{\text{res}}(X_j^l; \theta_1, \theta_2)$ while not hampering the information flow, we combine the original and preactivation ResBlocks to form the modified improved ResNet (I-ResNet). As shown in Fig. 5, I-ResNet contains the starting, middle, and ending residual blocks, and each block contains three operations of Conv3D, BN, and ReLU in different arrangements. Compared with SSRN that only uses the orignal ResBlocks [see Fig. 5(a)], the I-ResNet can improve the projection power according to our experiments.

Fig. 1 shows the proposed whole network that comprises four consecutive ResBlocks, each having 24 kernels. The shapes of the applied kernels and strides are shown in Table II. Depending on the kernel shape, ResBlocks are categorized into two groups, i.e., spectral feature learning and spatial feature learning. The first two ResBlocks are used to extract spatially focused spectral features, whereas the last two are used to

extract spectrally focused spatial features. Hence, the joint learning of spectral–spatial features increases the discriminative power of the proposed model. After the ResBlocks, a GAP layer is used to transform 3-D feature maps of size $7 \times 7 \times 24$ into a feature vector of size $1 \times 1 \times 24$. Finally, a fully connected layer with a $softmax$ function is used for classification. The classification objective is the commonly used cross-entropy function given by

$$\text{Loss}^{\text{CE}} = -\frac{1}{M} \sum_{m=1}^{M} \sum_{c=1}^{L_c} y_{L_c}^m \log \hat{y}_{L_c}^m \tag{16}$$

where $y_{L_c}^m$ and $\hat{y}_{L_c}^m$ are actual and predicted class labels, respectively, M is the total number of minibatch samples, and L_c is the total number of land-cover classes.

III. Experimental Results

To show the effectiveness of our method, we conduct classification experiments on three well-known benchmark HSI data sets.[1] The performance is evaluated in terms of overall accuracy (OA), average accuracy (AA), and statistical kappa (κ) coefficient. Among the total test samples, the ratio of correctly classified samples is determined as OA, the average of class-wise accuracy is determined as AA, and κ represents a strong mutual agreement between the generated classification maps of one network model and the given ground truth. We compare classical machine learning and representative deep learning methods available in [63][2]: random forest (RF) [64], multinomial logistic regression (MLR) [65], support vector machine (SVM) with radial basis function [1], gated recurrent unit (GRU) [66], long short term memory (LSTM) [67], ResNet [56], ContextNet [31], MS-3DNet [30], ENL-FCN [62], DPyResNet [46], and SSRN [32]. All experiments are performed in a system with the NVIDIA P100 16 GB GPU and 60 GB of RAM.

A. Description of Data Sets

The Indian Pines (IP) data set is gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [68] sensor over the IP test site in North-western Indiana. This data set contains 224 spectral bands within a wavelength range of 400–2500 nm. The 24 null and corrupted bands have been removed. The image spatial dimension is 145×145, and there are 16 mutually exclusive vegetation classes, some of with highly imbalanced number of samples.

The Kennedy Space Center (KSC) data set was gathered in 1996 by AVIRIS [68] with wavelengths ranging from 400 to 2500 nm. The images have a spatial dimension 512×614 pixels and 176 spectral bands after removal of some low signal-to-noise ratio (SNR) bands. The KSC data set consists of in total 5202 samples of 13 upland and wetland classes.

The University of Pavia (UP) data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [69] during a flight campaign over the university campus at Pavia, Northern Italy. The data set contains images of 9 land-cover classes from urban areas and the spatial dimension is 610×340 pixels, acquired in 103 spectral bands in the wavelength range of 430–860 nm.

[1] http://dase.grss-ieee.org/

[2] https://github.com/mhaut/hyperspectral_deeplearning_review

TABLE III
Summary of the Characteristic of the IP, the KSC, and the UP Data Sets

Description	Datasets		
	IP	KSC	UP
Sensor	AVIRIS	AVIRIS	ROSIS
Spatial Dimension	145 × 145	512 × 614	610 × 340
Spectral Bands	200	176	103
Land-cover	16	13	9
Total sample pixels	10249	5202	42776

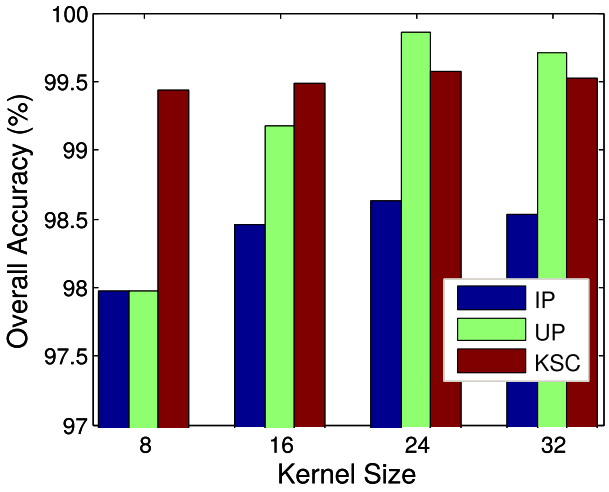


Fig. 6. Impacts of the number of kernels used in ResBlocks on OAs with 10% of training samples randomly taken from IP, KSC and UP, respectively.

Table III summarizes the three data sets. For preprocessing, images in data sets are normalized to zero mean value with unit variance. The 3-D patches of size $9 \times 9 \times 200$, $9 \times 9 \times 176$, and $9 \times 9 \times 103$ are extracted without dimensionality reduction from IP, KSC, and UP data sets, respectively. For our experiments, the whole samples are randomly divided into two disjoint training and test sets. The limited 10% samples are used for training and the remaining 90% for performance evaluation.

B. Experimental Setup and Parameter Evaluation

We learn the weights of 3-D spectral–spatial filters banks using the Adam [70] optimizer, which can smoothly back-propagate the flow of network gradients generated from the loss function. The model is trained using 200 epochs with limited samples randomly taken from the training set. The batch size is set to 32 and the whole process is repeated five times to report the average accuracy and the standard deviation. In every single epoch, the model configuration with the highest accuracy is used to evaluate the test set. In the following paragraphs, we will discuss other factors that affect the network performance.

The learning rate is a crucial part of gradient decent algorithms, which affects gradient update and network convergence. The optimal learning rate in the proposed model

TABLE IV

OVERALL ACCURACY (%) OF RESNET, DPYRESNET, SSRN, AND THE PROPOSED A^{2}S^{2}K-ResNet USING DIFFERENT SPATIAL WINDOW SIZES AND 10% OF TRAINING SAMPLES

Datasets	Spatial Window	Models					
		ResNet [56]	DPyResNet [31]	SSRN [32]	ENL-FCN [62]	MS-3DNet [30]	A^{2}S^{2}K-ResNet
IP	7×7	87.47 ± 0.008	91.31 ± 0.006	98.10 ± 0.005	69.25 ± 0.001	77.33 ± 0.015	**98.77 ± 0.001**
KSC		76.03 ± 0.052	82.12 ± 0.033	99.06 ± 0.004	27.40 ± 0.001	**95.95 ± 0.007**	**99.48 ± 0.003**
UP		96.33 ± 0.001	96.30 ± 0.006	99.54 ± 0.002	90.32 ± 0.001	98.26 ± 0.001	99.72 ± 0.001
IP	9×9	92.44 ± 0.006	91.47 ± 0.029	**98.38 ± 0.004**	**96.15 ± 0.054**	**83.44 ± 0.060**	98.66 ± 0.004
KSC		**95.61 ± 0.019**	**95.61 ± 0.019**	**99.29 ± 0.004**	**99.25 ± 0.020**	95.61 ± 0.019	99.34 ± 0.001
UP		97.38 ± 0.007	97.05 ± 0.010	99.77 ± 0.001	**99.76 ± 0.002**	99.35 ± 0.001	99.85 ± 0.001
IP	11×11	**92.67 ± 0.008**	**92.66 ± 0.003**	98.18 ± 0.003	56.48 ± 0.001	68.90 ± 0.150	98.55 ± 0.002
KSC		93.36 ± 0.023	95.47 ± 0.007	98.83 ± 0.006	53.44 ± 0.001	94.27 ± 0.002	99.47 ± 0.003
UP		**97.83 ± 0.006**	**97.21 ± 0.016**	**99.82 ± 0.001**	93.92 ± 0.001	**99.45 ± 0.001**	**99.90 ± 0.000**

TABLE V

OA, AA, AND κ VALUES ON THE IP DATA SET USING FIXED 10% OF TRAINING SAMPLES

Class	Training	Test	Classical Models					Deep Neural Networks						
			MLR [65]	RF [64]	SVM [1]	GRU [66]	LSTM [67]	ResNet [56]	ContextNet [31]	MS-3DNet [30]	ENL-FCN [62]	DPyResNet [46]	SSRN [32]	A^{2}S^{2}K-ResNet
1	4	42	15.45 ± 0.023	28.46 ± 0.061	51.22 ± 0.190	69.92 ± 0.141	69.11 ± 0.090	**98.66 ± 0.018**	88.78 ± 0.080	66.67 ± 0.471	97.56 ± 0.000	94.59 ± 0.076	57.78 ± 0.423	97.56 ± 0.034
2	142	1286	73.77 ± 0.006	56.63 ± 0.024	81.22 ± 0.037	76.96 ± 0.013	74.22 ± 0.016	87.85 ± 0.029	98.19 ± 0.005	75.94 ± 0.080	93.15 ± 0.000	93.83 ± 0.040	98.37 ± 0.012	**98.62 ± 0.010**
3	83	747	51.14 ± 0.027	48.42 ± 0.013	65.82 ± 0.013	67.20 ± 0.041	71.49 ± 0.030	92.71 ± 0.007	95.37 ± 0.028	81.39 ± 0.007	97.59 ± 0.000	89.30 ± 0.003	97.47 ± 0.010	**98.58 ± 0.006**
4	23	214	43.97 ± 0.051	33.49 ± 0.025	57.75 ± 0.041	61.82 ± 0.039	60.72 ± 0.041	95.43 ± 0.046	97.04 ± 0.021	88.63 ± 0.063	91.55 ± 0.000	93.51 ± 0.055	**99.12 ± 0.0099**	98.29 ± 0.014
5	48	435	83.52 ± 0.034	85.21 ± 0.025	90.04 ± 0.014	85.36 ± 0.022	87.51 ± 0.015	98.23 ± 0.015	97.78 ± 0.015	95.61 ± 0.054	97.47 ± 0.000	**99.26 ± 0.004**	97.79 ± 0.013	99.02 ± 0.003
6	73	657	94.82 ± 0.009	92.64 ± 0.027	96.25 ± 0.006	94.22 ± 0.008	94.77 ± 0.015	97.98 ± 0.011	98.60 ± 0.008	96.78 ± 0.026	99.24 ± 0.000	98.52 ± 0.007	98.50 ± 0.010	**98.71 ± 0.010**
7	2	26	41.33 ± 0.186	2.67 ± 0.038	73.33 ± 0.019	50.67 ± 0.068	85.33 ± 0.094	92.98 ± 0.099	90.35 ± 0.098	**100.00 ± 0.000**	**100.00 ± 0.000**	83.08 ± 0.178	66.67 ± 0.471	93.10 ± 0.097
8	47	431	98.53 ± 0.006	97.67 ± 0.015	97.98 ± 0.006	97.83 ± 0.001	97.83 ± 0.009	95.06 ± 0.014	97.76 ± 0.026	89.51 ± 0.091	97.44 ± 0.000	97.63 ± 0.022	96.45 ± 0.029	**98.83 ± 0.016**
9	2	18	5.56 ± 0.045	9.26 ± 0.094	50.00 ± 0.045	37.04 ± 0.146	53.70 ± 0.139	60.83 ± 0.283	**86.90 ± 0.102**	66.67 ± 0.471	72.22 ± 0.000	66.66 ± 0.471	56.25 ± 0.418	74.26 ± 0.038
10	97	875	65.41 ± 0.041	60.91 ± 0.047	73.87 ± 0.018	76.00 ± 0.035	73.68 ± 0.025	96.05 ± 0.013	96.08 ± 0.018	87.41 ± 0.070	94.74 ± 0.000	93.77 ± 0.029	**98.33 ± 0.009**	98.21 ± 0.016
11	245	2210	80.37 ± 0.015	87.88 ± 0.019	82.90 ± 0.012	80.31 ± 0.027	84.93 ± 0.024	93.32 ± 0.041	97.35 ± 0.004	76.69 ± 0.096	95.61 ± 0.000	89.78 ± 0.040	99.08 ± 0.005	**99.09 ± 0.001**
12	59	534	55.68 ± 0.007	41.26 ± 0.030	74.91 ± 0.043	78.65 ± 0.014	73.35 ± 0.052	86.65 ± 0.077	94.00 ± 0.012	88.65 ± 0.036	97.00 ± 0.000	83.43 ± 0.107	**98.46 ± 0.009**	98.37 ± 0.013
13	20	185	97.66 ± 0.005	90.09 ± 0.040	96.94 ± 0.021	96.94 ± 0.014	98.74 ± 0.005	82.16 ± 0.076	95.01 ± 0.03	99.78 ± 0.003	97.83 ± 0.000	98.19 ± 0.021	**100.00 ± 0.000**	99.80 ± 0.002
14	126	1139	95.61 ± 0.004	95.46 ± 0.014	93.82 ± 0.010	94.50 ± 0.012	96.22 ± 0.004	95.39 ± 0.016	98.49 ± 0.014	90.06 ± 0.087	99.12 ± 0.000	96.00 ± 0.021	98.63 ± 0.010	**99.22 ± 0.007**
15	38	348	56.00 ± 0.045	41.11 ± 0.029	60.42 ± 0.044	65.71 ± 0.019	60.04 ± 0.029	90.96 ± 0.127	94.10 ± 0.031	88.21 ± 0.044	92.80 ± 0.000	91.22 ± 0.040	**99.24 ± 0.005**	97.86 ± 0.013
16	9	84	84.92 ± 0.020	79.37 ± 0.030	91.27 ± 0.054	82.54 ± 0.037	90.87 ± 0.022	94.73 ± 0.038	93.57 ± 0.046	98.53 ± 0.021	**100.00 ± 0.000**	70.90 ± 0.388	95.63 ± 0.062	95.93 ± 0.057
OA	1018	9231	76.23 ± 0.008	72.98 ± 0.006	82.00 ± 0.006	81.24 ± 0.003	82.13 ± 0.004	92.44 ± 0.006	96.98 ± 0.006	83.44 ± 0.060	96.15 ± 0.054	91.47 ± 0.029	98.38 ± 0.004	**98.66 ± 0.004**
AA			65.23 ± 0.019	59.41 ± 0.005	77.36 ± 0.019	75.98 ± 0.008	79.53 ± 0.005	91.19 ± 0.025	94.96 ± 0.003	86.91 ± 0.084	95.21 ± 0.028	94.14 ± 0.006	91.11 ± 0.080	**96.59 ± 0.003**
κ			0.7266 ± 0.010	0.6862 ± 0.007	0.7941 ± 0.007	0.7858 ± 0.004	0.7954 ± 0.004	0.9137 ± 0.006	0.9655 ± 0.007	0.8082 ± 0.070	0.9560 ± 0.030	0.9020 ± 0.034	0.9815 ± 0.005	**0.9848 ± 0.005**

TABLE VI

OA, AA, AND κ VALUES ON THE KSC DATA SET USING FIXED 10% OF TRAINING SAMPLES

Class	Training	Test	Classical Models					Deep Neural Networks						
			MLR [65]	RF [64]	SVM [1]	GRU [66]	LSTM [67]	ResNet [56]	ContextNet [31]	MS-3DNet [30]	ENL-FCN [62]	DPyResNet [46]	SSRN [32]	A^{2}S^{2}K-ResNet
1	76	685	95.57 ± 0.008	94.79 ± 0.012	95.43 ± 0.023	96.98 ± 0.011	94.60 ± 0.004	94.73 ± 0.008	99.78 ± 0.001	96.42 ± 0.009	99.71 ± 0.000	99.06 ± 0.010	**99.95 ± 0.001**	**99.95 ± 0.001**
2	24	219	88.74 ± 0.024	81.58 ± 0.047	83.71 ± 0.012	82.04 ± 0.022	85.69 ± 0.018	66.45 ± 0.310	98.79 ± 0.014	95.88 ± 0.012	**100.00 ± 0.000**	89.72 ± 0.026	**100.00 ± 0.00**	98.68 ± 0.019
3	25	231	90.87 ± 0.050	86.09 ± 0.020	78.41 ± 0.218	89.13 ± 0.023	91.16 ± 0.040	65.08 ± 0.187	82.83 ± 0.047	80.12 ± 0.168	**100.00 ± 0.000**	81.84 ± 0.074	99.66 ± 0.005	98.72 ± 0.012
4	25	227	42.00 ± 0.061	71.22 ± 0.061	27.17 ± 0.173	56.98 ± 0.062	68.14 ± 0.049	73.62 ± 0.185	78.41 ± 0.165	90.06 ± 0.012	**98.67 ± 0.000**	89.83 ± 0.040	91.22 ± 0.047	94.27 ± 0.042
5	16	145	25.06 ± 0.108	47.59 ± 0.060	22.99 ± 0.170	68.74 ± 0.092	49.89 ± 0.107	60.74 ± 0.275	74.22 ± 0.097	85.86 ± 0.034	98.61 ± 0.000	88.34 ± 0.095	**100.00 ± 0.00**	94.46 ± 0.050
6	22	207	50.16 ± 0.026	48.22 ± 0.014	36.89 ± 0.078	63.27 ± 0.100	54.21 ± 0.062	66.58 ± 0.324	92.64 ± 0.050	85.61 ± 0.030	**100.00 ± 0.000**	88.54 ± 0.138	98.45 ± 0.022	99.82 ± 0.003
7	10	95	75.53 ± 0.069	79.43 ± 0.096	87.94 ± 0.027	90.78 ± 0.031	89.01 ± 0.013	66.20 ± 0.468	94.40 ± 0.037	90.75 ± 0.088	**100.00 ± 0.000**	**100.00 ± 0.00**	95.42 ± 0.050	99.61 ± 0.005
8	43	388	77.75 ± 0.020	78.61 ± 0.054	70.19 ± 0.073	90.03 ± 0.031	92.53 ± 0.011	92.53 ± 0.081	97.49 ± 0.009	98.99 ± 0.008	**100.00 ± 0.000**	94.81 ± 0.037	99.80 ± 0.003	**100.00 ± 0.000**
9	52	468	89.32 ± 0.005	89.46 ± 0.011	85.33 ± 0.021	96.01 ± 0.015	95.37 ± 0.028	90.82 ± 0.106	99.92 ± 0.001	97.44 ± 0.028	**100.00 ± 0.000**	99.06 ± 0.002	**100.00 ± 0.000**	**100.00 ± 0.000**
10	40	364	87.60 ± 0.013	88.43 ± 0.034	78.88 ± 0.069	91.09 ± 0.015	94.03 ± 0.007	87.59 ± 0.171	**100.00 ± 0.000**	98.78 ± 0.013	**100.00 ± 0.000**	99.46 ± 0.004	**100.00 ± 0.000**	**100.00 ± 0.000**
11	41	378	92.93 ± 0.024	95.58 ± 0.014	93.81 ± 0.008	96.02 ± 0.026	96.11 ± 0.008	98.96 ± 0.015	99.90 ± 0.001	98.67 ± 0.013	**100.00 ± 0.000**	99.90 ± 0.001	**100.00 ± 0.000**	**100.00 ± 0.000**
12	50	453	93.60 ± 0.013	82.63 ± 0.032	86.98 ± 0.009	89.99 ± 0.035	92.35 ± 0.035	84.26 ± 0.207	99.17 ± 0.006	99.06 ± 0.005	**100.00 ± 0.000**	94.42 ± 0.056	**100.00 ± 0.000**	**100.00 ± 0.000**
13	92	835	**100.00 ± 0.000**	99.60 ± 0.002	**100.00 ± 0.000**	99.92 ± 0.001	99.96 ± 0.001	**100.00 ± 0.000**	99.96 ± 0.001	100.00 ± 0.000	**100.00 ± 0.000**	99.96 ± 0.001	**100.00 ± 0.000**	**100.00 ± 0.000**
OA	516	4695	85.52 ± 0.003	86.17 ± 0.004	81.27 ± 0.008	90.13 ± 0.004	90.18 ± 0.005	79.73 ± 0.207	96.34 ± 0.014	95.61 ± 0.019	99.25 ± 0.020	95.61 ± 0.019	99.29 ± 0.004	**99.34 ± 0.0008**
AA			77.62 ± 0.005	80.25 ± 0.004	72.90 ± 0.021	85.46 ± 0.010	84.85 ± 0.008	80.58 ± 0.169	93.65 ± 0.026	93.66 ± 0.023	98.77 ± 0.021	94.22 ± 0.024	98.80 ± 0.008	**98.88 ± 0.0018**
κ			0.8385 ± 0.004	0.8459 ± 0.004	0.7909 ± 0.009	0.8899 ± 0.004	0.8906 ± 0.006	0.7806 ± 0.222	0.9593 ± 0.016	0.9511 ± 0.021	0.9913 ± 0.018	0.9511 ± 0.021	0.9921 ± 0.004	**0.9927 ± 0.001**

was found using one cycle policy [71]. Once the optimal learning rate was set, a cosine annealing scheduler was used to regulate the learning rate over 200 epochs for each data set. The overfitting is also an important issue associated with the model while dealing with a large number of trainable parameters. To overcome this problem, the regularization technique, namely batch nomalization (BN), was introduced to further optimize the network parameters. Fig. 5 shows the BN used in the main path of ResBlocks before and after the convolutional layer. To achieve stable classification performance, all the results throughout this article are computed using the BN regularizer.

The size and shape of the kernels in ResBlock play an important role in extracting discriminative features. Table II shows the 24 kernels of recorded shapes that are shared in the convolutional layers of respective ResBlock to extract spectral–spatial feature maps. To achieve good OAs, the number of kernels is varied in an interval of 8 from 8 to 32 (more filters increase discriminative power at the cost of computation time [72]). Fig. 6 shows the OAs achieved by the proposed model using a different number of kernels on the IP, KSC, and UP data sets. It is observed that 24 kernels give optimal results and the performance variation is almost negligible for different kernels on the KSC data set. Thus, 24 kernels are used in each convolutional layer for the three data sets.

The spatial size of input 3-D patches also influences the performance of the proposed network. As shown in Table IV, the best OAs of the proposed methods are achieved using windows of size 7×7, 7×7, and 11×11 for IP, KSC, and UP data sets, respectively. For SSRN, the best results are achieved using windows of size 9×9, 9×9, and 11×11. For fair comparisons, the spatial window of the proposed method is fixed as $9 \times 9 \times B$ for all the three data sets.

C. Classification Results

Tables V–VII show the average OAs, AAs, and κs (and their standard deviations based on five runs) on the IP, KSC, and UP data sets, respectively. Overall, the proposed method outperforms all the classical machine learning models in terms

TABLE VII

OA, AA, AND κ VALUES ON THE UP DATA SET USING FIXED 10% OF TRAINING SAMPLES

Class	Training	Test	Classical Models					Deep Neural Networks						
			MLR [65]	RF [64]	SVM [1]	GRU [66]	LSTM [67]	ResNet [56]	ContextNet [31]	MS-3DNet [30]	ENL-FCN [62]	DPyResNet [46]	SSRN [32]	A^{2}S^{2}K-ResNet
1	663	5968	92.30 ± 0.004	91.11 ± 0.007	94.30 ± 0.008	93.34 ± 0.003	95.47 ± 0.005	96.82 ± 0.023	99.56 ± 0.002	99.36 ± 0.001	**99.98 ± 0.000**	98.35 ± 0.017	99.85 ± 0.001	99.91 ± 0.000
2	1864	16785	96.18 ± 0.003	98.11 ± 0.003	97.65 ± 0.002	97.54 ± 0.002	96.91 ± 0.002	98.59 ± 0.008	99.85 ± 0.002	99.80 ± 0.000	**100.00 ± 0.000**	98.76 ± 0.008	99.98 ± 0.000	99.99 ± 0.000
3	209	1890	72.75 ± 0.013	67.71 ± 0.014	81.26 ± 0.018	77.08 ± 0.038	78.01 ± 0.011	90.01 ± 0.061	99.19 ± 0.001	98.02 ± 0.017	99.68 ± 0.000	94.22 ± 0.034	99.68 ± 0.003	**99.88 ± 0.001**
4	306	2758	89.28 ± 0.002	88.20 ± 0.006	94.63 ± 0.004	93.22 ± 0.023	94.92 ± 0.007	99.32 ± 0.003	99.80 ± 0.002	99.71 ± 0.001	98.94 ± 0.000	99.20 ± 0.005	99.92 ± 0.001	**99.95 ± 0.001**
5	134	1211	99.42 ± 0.003	98.93 ± 0.002	99.20 ± 0.002	99.42 ± 0.004	99.26 ± 0.003	99.81 ± 0.000	99.91 ± 0.001	99.94 ± 0.000	**100.00 ± 0.000**	99.72 ± 0.003	99.94 ± 0.000	**100.00 ± 0.00**
6	502	4527	77.45 ± 0.005	72.14 ± 0.022	90.58 ± 0.008	87.41 ± 0.016	87.85 ± 0.012	99.41 ± 0.002	99.75 ± 0.003	99.43 ± 0.003	99.87 ± 0.000	98.52 ± 0.006	**99.95 ± 0.001**	99.91 ± 0.001
7	133	1197	55.69 ± 0.043	75.69 ± 0.017	85.71 ± 0.011	85.38 ± 0.039	80.23 ± 0.007	96.90 ± 0.017	98.37 ± 0.022	99.18 ± 0.005	**100.00 ± 0.000**	97.37 ± 0.004	**100.00 ± 0.000**	**100.00 ± 0.000**
8	368	3314	87.04 ± 0.004	89.64 ± 0.013	88.20 ± 0.003	88.56 ± 0.024	88.49 ± 0.008	92.00 ± 0.044	98.48 ± 0.008	97.13 ± 0.005	99.69 ± 0.000	84.51 ± 0.071	98.28 ± 0.015	98.88 ± 0.006
9	94	853	99.77 ± 0.001	99.77 ± 0.002	99.84 ± 0.001	99.84 ± 0.001	**99.88 ± 0.001**	98.88 ± 0.012	99.26 ± 0.005	99.74 ± 0.002	**100.00 ± 0.000**	99.60 ± 0.001	99.39 ± 0.003	99.78 ± 0.003
OA	4273	38503	89.87 ± 0.001	90.41 ± 0.001	94.19 ± 0.002	93.34 ± 0.002	93.45 ± 0.001	97.38 ± 0.007	99.57 ± 0.001	99.35 ± 0.001	99.76 ± 0.002	97.05 ± 0.010	99.77 ± 0.001	**99.85 ± 0.001**
AA			85.54 ± 0.004	86.81 ± 0.002	92.38 ± 0.003	91.31 ± 0.008	91.23 ± 0.001	96.86 ± 0.005	99.35 ± 0.002	99.15 ± 0.002	99.70 ± 0.002	96.69 ± 0.006	99.66 ± 0.001	**99.81 ± 0.001**
κ			0.8646 ± 0.001	0.8710 ± 0.002	0.9229 ± 0.002	0.9115 ± 0.003	0.9130 ± 0.001	0.9652 ± 0.009	0.9943 ± 0.001	0.9913 ± 0.002	0.9972 ± 0.001	0.9608 ± 0.013	0.9969 ± 0.001	**0.9981 ± 0.001**

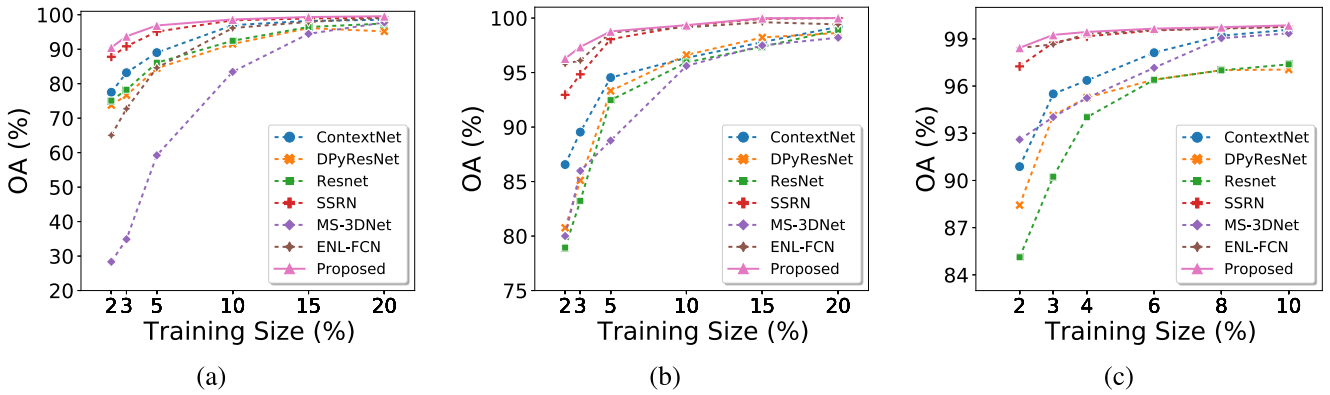


Fig. 7. OA achieved by different methods with varying training sample sizes. (a) IP, (b) KSC, and (c) UP data sets.

of OA, AA, and κ on the three data sets. Specifically, on the IP data set, our method achieves the OA value of 98.66%, and the three classical models, SVM, GRU, and LSTM, achieve 82.00%, 81.24%, and 82.13%, respectively. On the KSC data set, our method achieves the highest OA value of 99.34%. GRU (90.13%) and LSTM (90.18%) perform much better than SVM (81.27%), while RF (86.17%) achieves comparable performance to MLR (85.52%). On the UP data set, the proposed method achieves superior OA performance of 99.85%. By contrast, the best OA performance of the classical models is 94.19% obtained by SVM.

For deep neural networks, when using 10% of training samples, the proposed method performs slightly better than SSRN [32] and significantly outperforms other methods in terms of OA, AA, and κ on three data sets. In particular, the differences of AAs between the proposed method and the second best methods are +1.38, +0.08, and +0.11 on IP, KSC, and UP data sets, respectively. In addition, almost all methods, including some classical models, outperform ResNet by a large margin on the KSC data set. The ENL-FCN and SSRN perform almost equally over the KSC and UP data sets. ENL-FCN achieves worse OA and κ values than ContextNet, SSRN, and the proposed method on IP. The proposed method consistently achieves improved performance compared to SSRN, indicating that our method can better model the spectral–spatial features selected by the adjustment of various RFs. The class-specific samples in the IP data set are highly imbalanced. For example, the class of "Oats" (Class 9) contains 20 samples, whereas "Soybean-mintill" (Class 11) contains 2455 samples. Regarding the class-specific accuracy, ContextNet performs best (86.90%) for the class of "Oats," followed by the proposed method (74.26%). ENL-FCN (72.22%) performs better than MS-3DNet (66.67%), ResNet (60.83%), and SSRN (56.25%).

We further evaluate the performance of different methods with varying training sample sizes, i.e., 2%, 3%, 5%, 10%, 15%, and 20% for IP and KSC data sets and 2%, 3%, 4%, 6%, 8%, and 10% for the UP data set. Figs. 7 and 8 show the OA and *kappa* values of different methods with varying training set sizes, respectively. Overall, the proposed method consistently outperforms other ones on all three data sets. The poor performance is given by ResNet and DPyResNet in terms of OA and *kappa* on all data sets. When a larger portion of training set is used, the proposed method and SSRN perform closely on all three data sets. However, when a sufficiently small training set (e.g., lower than 5%) is used, the robustness of the proposed method can be clearly observed by comparing the large differences among ours and ENL-FCN and SSRN. This is mainly because we use an improved spectral–spatial ResNet with feature recalibration mechanism to extract more discriminative spectral–spatial recalibrated features.

D. Ablation Study

To further validate the effectiveness of different modules used in the proposed framework, we perform ablation experiments while keeping the other experimental settings unchanged (as discussed in Section III-B).

1) The SK and efficient channel attention (ECA) modules are removed from the proposed framework, and feature extraction and classification are performed using IResNet.

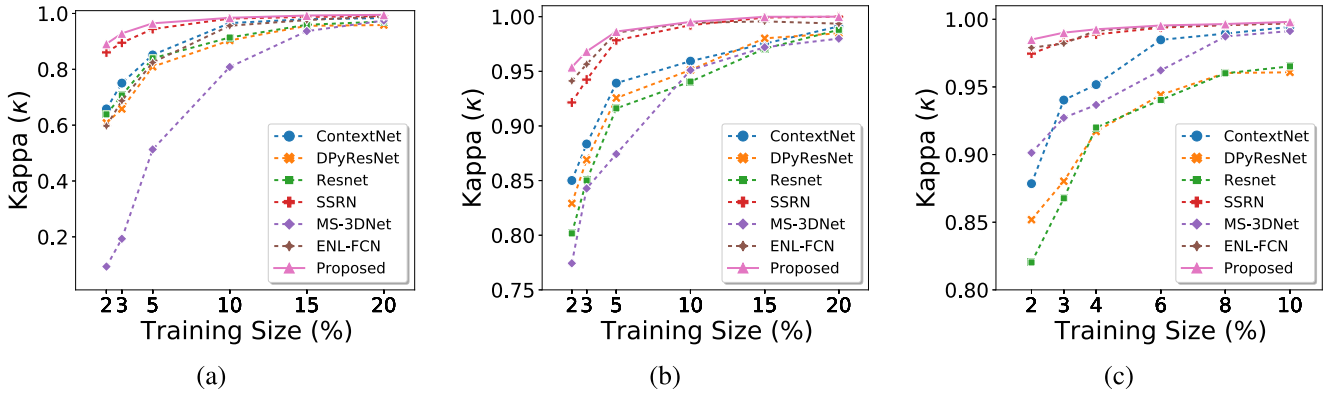


Fig. 8. Kappa coefficient achieved by different methods with varying training sample sizes. (a) IP, (b) KSC, and (c) UP data sets.

TABLE VIII

ACCURACY ANALYSIS IN TERMS OF OA, AA, AND KAPPA FOR DIFFERENT MODULES OF THE PROPOSED FRAMEWORK

Matrices	Datasets	IResNet	SK+SSRN	SK+IResNet	IResNet+ECA	SK+IResNet+ECA
OA	IP	97.83 ± 0.004	96.20 ± 0.006	97.01 ± 0.002	97.29 ± 0.003	98.66 ± 0.004
AA		95.53 ± 0.005	96.22 ± 0.003	95.54 ± 0.021	95.85 ± 0.018	96.59 ± 0.003
Kappa		0.9837 ± 0.001	0.9566 ± 0.006	0.9659 ± 0.003	0.9691 ± 0.004	0.9848 ± 0.005
OA	KSC	98.27 ± 0.009	98.65 ± 0.003	99.21 ± 0.006	99.32 ± 0.002	99.34 ± 0.0008
AA		99.75 ± 0.006	98.12 ± 0.002	98.81 ± 0.007	99.14 ± 0.004	98.88 ± 0.0018
Kappa		0.9870 ± 0.001	0.9849 ± 0.004	0.9913 ± 0.007	0.9926 ± 0.002	0.9927 ± 0.001
OA	UP	99.07 ± 0.002	99.13 ± 0.001	99.60 ± 0.001	99.45 ± 0.001	99.85 ± 0.001
AA		99.21 ± 0.003	98.73 ± 0.001	99.36 ± 0.001	99.23 ± 0.001	99.81 ± 0.001
Kappa		0.9852 ± 0.002	0.9885 ± 0.001	0.9846 ± 0.001	0.9927 ± 0.002	0.9981 ± 0.001

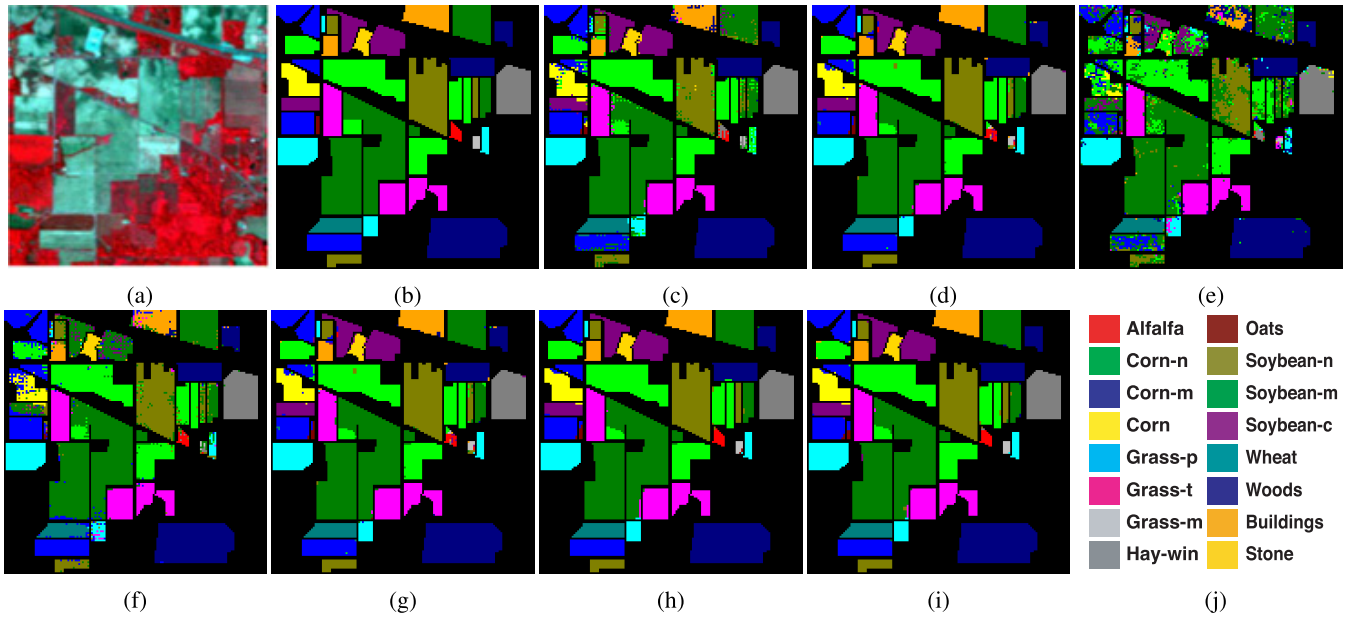


Fig. 9. Classification maps for IP data set. (a) False-color composite image. (b) Ground truth. (c)–(i) Classification maps produced by the ResNet, the ContextNet, the MS-3DNet, the DPyResNet, the ENL-FCN, the SSRN, and the proposed A^{2}S^{2}K-ResNet, respectively. (j) Color labels.

2) The SK module is removed from the proposed framework (the resulting model is denoted as IResNet+ECA).
3) The ECA-based IResNet is replaced by SSRN (the resulting model is denoted as SK+SSRN).
4) The ECA module is removed from the IResNet (the resulting model is denoted as SK+IResNet).

Table VIII shows the results of the ablation study in terms of OA, AA, and Kappa using window size 9×9 and 10% of training samples over IP, KSC, and UP data sets. Compared with SK+SSRN, SK+IResNet provides a significant improvement in OA for all the data sets and achieves comparable performance in terms of Kappa for the UP data set. We also observe that IResNet+ECA works better than SK+IResNet for the IP and KSC data sets. Overall, the proposed network can achieve the state-of-the-art performance. Hence, we can conclude that in the proposed framework, the SK module helps to adaptively learn the importance of each spectral–spatial kernel features. IResNet+ECA extracts and learns the recalibrated feature maps, which emphasizes informative feature maps and suppressing less useful features during training the network.

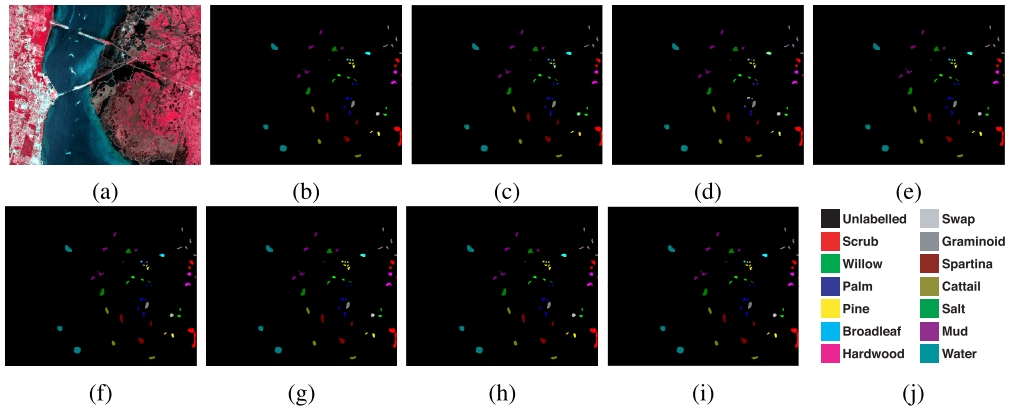


Fig. 10. Classification maps for KSC data set. (a) False color composite image. (b) Ground truth. (c)–(i) Classification maps produced by the ResNet, the ContextNet, the MS-3-DNet, the DPyResNet, the ENL-FCN, the SSRN, and the proposed A^{2}S^{2}K-ResNet, respectively. (j) Color labels.

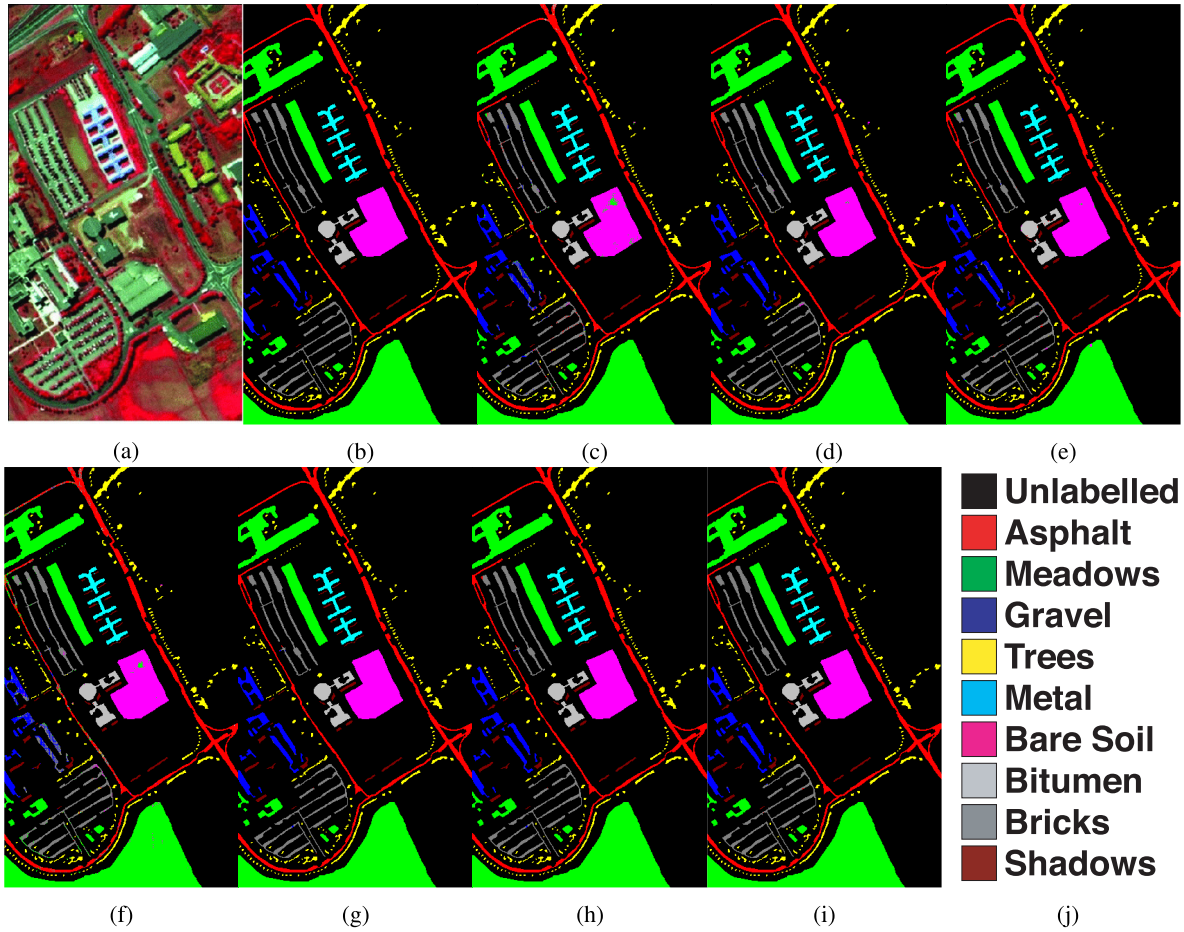
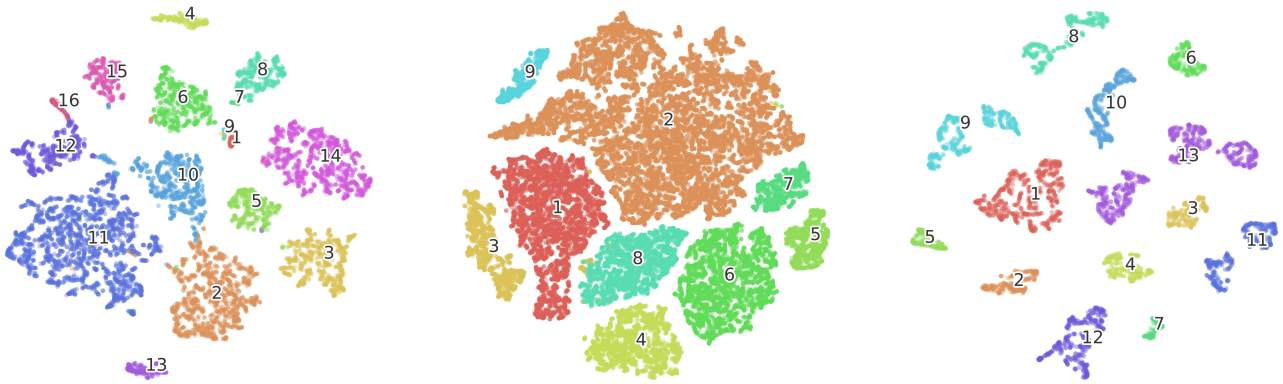


Fig. 11. Classification maps for UP data set. (a) False color composite image. (b) Ground truth. (c)–(i) Classification maps produced by the ResNet, the ContextNet, the MS-3-DNet, the DPyResNet, the ENL-FCN, the SSRN, and the proposed A^{2}S^{2}K-ResNet, respectively. (j) Color labels.

TABLE IX

COMPARISONS OF COMPUTATIONAL COSTS AND MEMORY USES (MB) OVER THREE DATA SETS

	ResNet	ContextNet	MS-3DNet	ENL-FCN	DPyResNet	SSRN	Proposed
Params	21.9M	554.1K	269K	113.9K	22.3M	364.1K	370.7K
GPU Memory (MB)	83.78	2.89	2.94	504.74	85.63	10.27	12.02
Computation Cost ($10^6\times$ FLOPs)	84.1	63.2	89.9	4683.9	85.1	314.8	340.1
Datasets	Computation Time (ms/sample)						
IP	15	14	2	22	17	91	39
KSC	1.6	2.9	0.5	3.7	1.5	2.5	1.4
UP	156	260	33	152	134	259	211



(a) (b) (c)

Fig. 12. Visualization of the proposed 2-D spectral–spatial features for the test samples in different data sets via t-SNE. The points represent the features of test samples and their class labels are shown with different colors. (a) IP, (b) KSC, and (c) UP. (Best viewed in color.)

E. Analysis of Computational Cost and Memory Usage

Table IX shows the model complexity of different networks in terms of number of trainable weight parameters updated during backpropagation, GPU memory requirements, and computational cost. It is observed that SSRN [32] and the proposed model use approximately the same number of parameters. ENL-FCN [62] uses the least number of parameters, whereas ResNet and DPyResNet require the largest number of parameters. The memory usage of each model is also given in Table IX. It is observed that the proposed network uses approximately 12-MB GPU memory, while the memory requirement is comparatively less for MS-3DNet and ContextNet. Due to the LSTM-based cell architecture, ENL-FCN requires 504.74-MB GPU memory. The computational cost is calculated by floating-point operations ($10^6\times$ FLOPs). It is observed from the results that SSRN and the proposed method have close FLOPs (i.e., 314.8 and 340.1). ENL-FCN and ContextNet take the highest and lowest FLOPs, i.e., 4683.9 and 63.2, respectively. In addition, Table IX shows the per sample training time for different methods calculated over IP, KSC, and UP data sets. The proposed model takes 39, 1.4, and 211 ms per sample for training over the IP, KSC, and UP data sets, respectively. For the IP data set, SSRN and MS-3DNet take the highest and lowest training time, i.e., 91 and 2 ms per sample, respectively. For the KSC data set, MS-3DNet and ENL-FCN take 0.5 and 3.7 ms per sample, respectively. In the case of UP, ContextNet takes the highest training time, whereas MS-3DNet takes the lowest.

F. Visualization

The visualization of classification maps generated by different methods is shown in Figs. 9–11, for IP, KSC, and UP data sets. Figures (a) and (b) shows the false-color composites and ground-truth images, respectively, while (c)–(i) are generated by different methods using 10% of training samples. One can see that the quality of the proposed model and SSRN looks better compared to others due to the less artifacts present in the generated maps.

To show the representation ability of our trained model, we visualize the extracted features in 2-D space via the t-SNE algorithm [73]. As shown in Fig. 12, samples from the same class are clustered into one group, whereas samples from different classes become easily separable from each other. Thus, the proposed model can well learn the abstract representation of spectral–spatial features.

IV. Conclusion

In this article, we have presented an attention-based adaptive spectral–spatial kernel ResNet (A^{2}S^{2}K-ResNet) for HSI classification. The proposed network selects spectral–spatial 3-D convolutional kernels with automatically adjusted RF sizes in an end-to-end training fashion. The spectral and spatial features are jointly extracted from the selected 3-D kernel feature maps using improved ResBlocks. Every ResBlock is followed by an EFR mechanism to augment the discriminative power of convolutional feature maps. The classification performance has been evaluated on three well-known HSI data sets, and the proposed A^{2}S^{2}K-ResNet outperformed other state-of-the-art methods, especially when using a limited number of training samples.

References

[1] 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.

[2] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, Mar. 2006.

[3] T. Qiao *et al.*, "Joint bilateral filtering and spectral similarity-based sparse representation: A generic framework for effective feature extraction and data classification in hyperspectral imaging," *Pattern Recognit.*, vol. 77, pp. 316–328, May 2018.

[4] Z. Wang, T. Yang, and H. Zhang, "Land contained sea area ship detection using spaceborne image," *Pattern Recognit. Lett.*, vol. 130, pp. 125–131, Feb. 2020.

[5] Y. Cai, X. Liu, and Z. Cai, "BS-nets: An end-to-end framework for band selection of hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1969–1984, Mar. 2020.

[6] S. K. Roy, S. Das, T. Song, and B. Chanda, "DARecNet-BS: Unsupervised dual-attention reconstruction network for hyperspectral band selection," *IEEE Geosci. Remote Sens. Lett.*, early access, Aug. 11, 2020, doi: 10.1109/LGRS.2020.3013235.

[7] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Joint and progressive subspace analysis (JPSA) with spatial-spectral manifold alignment for semisupervised hyperspectral dimensionality reduction," *IEEE Trans. Cybern.*, early access, Nov. 11, 2020, doi: 10.1109/TCYB.2020.3028931.

[8] D. Hong, N. Yokoya, J. Xu, and X. Zhu, "Joint & progressive learning from high-dimensional data for multi-label classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 469–484.

[9] D. Hong *et al.*, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 24, 2020, doi: 10.1109/TGRS.2020.3016820.

[10] L. Gao, D. Yao, Q. Li, L. Zhuang, B. Zhang, and J. Bioucas-Dias, "A new low-rank representation based hyperspectral image denoising method for mineral mapping," *Remote Sens.*, vol. 9, no. 11, p. 1145, Nov. 2017.

[11] Y. Li, Q. Li, Y. Liu, and W. Xie, "A spatial-spectral SIFT for hyperspectral image matching and classification," *Pattern Recognit. Lett.*, vol. 127, pp. 18–26, Nov. 2019.

[12] Y. Shao, N. Sang, C. Gao, and L. Ma, "Spatial and class structure regularized sparse representation graph for semi-supervised hyperspectral image classification," *Pattern Recognit.*, vol. 81, pp. 81–94, Sep. 2018.

[13] B. Tu, N. Li, L. Fang, X. Yang, and J. Wu, "Hyperspectral image classification with a class-dependent spatial–spectral mixed metric," *Pattern Recognit. Lett.*, vol. 123, pp. 16–22, May 2019.

[14] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. Atli Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

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

[16] B. Rasti *et al.*, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep," 2020, *arXiv:2003.02822*. [Online]. Available: https://arxiv.org/abs/2003.02822

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

[18] C. Zhao, X. Wan, G. Zhao, B. Cui, W. Liu, and B. Qi, "Spectral-spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest," *Eur. J. Remote Sens.*, vol. 50, no. 1, pp. 47–63, Jan. 2017.

[19] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5132–5136.

[20] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4959–4962.

[21] 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.

[22] T. Alipour-Fard, M. E. Paoletti, J. M. Haut, H. Arefi, J. Plaza, and A. Plaza, "Multibranch selective kernel networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, early access, May 11, 2020, doi: 10.1109/LGRS.2020.2990971.

[23] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 593–597, Apr. 2019.

[24] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and L. Plaza, "Hyperspectral image classification using random occlusion data augmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1751–1755, Nov. 2019.

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

[26] 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.

[27] 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.

[28] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[29] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[30] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.

[31] 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.

[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] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral-spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, May 28, 2020, doi: 10.1109/TGRS.2020.2994057.

[34] 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, Feb. 2020.

[35] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5659–5667.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[37] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[38] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[39] S. K. Roy, S. R. Dubey, S. Chatterjee, and B. Baran Chaudhuri, "FuSENet: Fused squeeze-and-excitation network for spectral-spatial hyperspectral image classification," *IET Image Process.*, vol. 14, no. 8, pp. 1653–1661, Jun. 2020.

[40] 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.

[41] Y. Yu, Z. Gong, C. Wang, and P. Zhong, "An unsupervised convolutional feature fusion network for deep representation of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 1, pp. 23–27, Jan. 2018.

[42] M. Han, R. Cong, X. Li, H. Fu, and J. Lei, "Joint spatial-spectral hyperspectral image classification based on convolutional neural network," *Pattern Recognit. Lett.*, vol. 130, pp. 38–45, Feb. 2020.

[43] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[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. 2019.

[45] L. Fang, Z. Liu, and W. Song, "Deep hashing neural networks for hyperspectral image feature extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1412–1416, Sep. 2019.

[46] 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.

[47] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[48] L. Ding, H. Tang, and L. Bruzzone, "LANet: Local attention embedding to improve the semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, early access, May 27, 2020, doi: 10.1109/TGRS.2020.2994150.

[49] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 18, 2020, doi: 10.1109/TGRS.2020.3015157.

[50] A. Coates and A. Y. Ng, "Selecting receptive fields in deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2528–2536.

[51] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4898–4906.

[52] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[53] Y. Lee, H. Jung, D. Han, K. Kim, and J. Kim, "Learning receptive field size by learning filter size," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1203–1212.

[54] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[55] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: Efficient channel attention for deep convolutional neural networks," 2019, *arXiv:1910.03151*. [Online]. Available: http://arxiv.org/abs/1910.03151

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*

(CVPR), Jun. 2016, pp. 770–778.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, Sep. 2016, pp. 630–645.
[58] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146
[59] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.
[60] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5927–5935.
[61] I. Cosmin Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," 2020, *arXiv:2004.04989*. [Online]. Available: http://arxiv.org/abs/2004.04989
[62] Y. Shen *et al.*, "Efficient deep learning of nonlocal features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 17, 2020, doi: 10.1109/TGRS.2020.3014286.
[63] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271619302187
[64] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
[65] J. Haut, M. Paoletti, A. Paz-Gallardo, J. Plaza, A. Plaza, and J. Vigo-Aguiar, "Cloud implementation of logistic regression for hyperspectral image classification," in *Proc. 17th Int. Conf. Comput. Math. Methods Sci. Eng. (CMMSE)*, vol. 3. Cádiz, Spain: Costa Ballena (Rota), 2017, pp. 1063–2321.
[66] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259
[67] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *J. Supercomput.*, vol. 76, no. 11, pp. 8866–8882, Feb. 2020, doi: 10.1007/s11227-020-03187-0.
[68] R. O. Green *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.
[69] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. Van der Piepen, and M. Schroder, "ROSIS (Reflective Optics System Imaging Spectrometer)-A candidate instrument for polar platform missions," *Proc. SPIE*, vol. 0868, pp. 134–141, Apr. 1988.
[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[71] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*. [Online]. Available: http://arxiv.org/abs/1803.09820
[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[73] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Swalpa Kumar Roy (Student Member, IEEE) received the bachelor's degree in computer science and engineering from the West Bengal University of Technology, Kolkata, India, in 2012, and the master's degree in computer science and engineering from the Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India, in 2015. He is pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of Calcutta, Kolkata, India.

He was a Project Linked Person with the Optical Character Recognition (OCR) Laboratory, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, from July 2015 to March 2016. He is an Assistant Professor with the Department of Computer Science and Engineering, Jalpaiguri Government Engineering College, Jalpaiguri, India. His research interests include computer vision, deep learning, remote sensing, and texture feature description.

Suvojit Manna received the bachelor's degree in computer science and engineering from the Jalpaiguri Government Engineering College, Jalpaiguri, India, in 2018.

He is a Data Scientist with CureSkin, Bengaluru, India. His research interest includes computer vision and deep learning applications.

Tiecheng Song (Member, IEEE) received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China (UESTC), in 2015.

From October 2015 to April 2016, he joined the Multimedia Lab, Nanyang Technological University, Singapore, as a Visiting Student. He is an Associate Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China. His research interests include image processing and computer vision.

Lorenzo Bruzzone (Fellow, IEEE) received the Laurea (M.S.) degree *(summa cum laude)* in electronic engineering and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

He is the Founder and the Director of the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento. He is a Full Professor of telecommunications with the University of Trento, Trento, Italy, where he teaches remote sensing, radar, and digital communications. He is the Principal Investigator of many research projects. He is the Principal Investigator of the Radar for icy Moon exploration (RIME) instrument in the framework of the JUpiter ICy moons Explorer (JUICE) mission of the European Space Agency (ESA) and the Science Lead for the High Resolution Land Cover project in the framework of the Climate Change Initiative of ESA. He is the author (or coauthor) of 294 scientific publications in referred international journals (221 in IEEE journals), more than 340 articles in conference proceedings, and 22 book chapters. He is editor/coeditor of 18 books/conference proceedings and 1 scientific book. His articles are highly cited, as proven from the total number of citations (more than 35 700) and the value of the h-index (89) (source: Google Scholar). He was invited as Keynote Speaker in more than 40 international conferences and workshops. His research interests are in the areas of remote sensing, radar and SAR, signal processing, machine learning and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects.

Dr. Bruzzone has been a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS), since 2009, where since 2019, he has been a Vice-President for Professional Activities. He ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Seattle, July 1998. Since that he was a recipient of many international and national honors and awards, including the recent IEEE GRSS 2015 Outstanding Service Award, the 2017 and 2018 IEEE IGARSS Symposium Prize Paper Awards, and the 2019 WHISPER Outstanding Paper Award. He was a Guest Coeditor of many Special Issues of international journals. He is the Co-Founder of the IEEE International Workshop on the Analysis of Multi-Temporal Remote-Sensing Images (MultiTemp) series and is a member of the Permanent Steering Committee of this series of workshops. Since 2003, he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He has been the Founder of the *IEEE Geoscience and Remote Sensing Magazine* for which he has been an Editor-in-Chief from 2013 to 2017. He is an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has been a Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society from 2012 to 2016.