Lightweight Spectral–Spatial Squeeze-and-Excitation Residual Bag-of-Features Learning for Hyperspectral Classification

Swalpa Kumar Roy, *Student Member, IEEE*, Subhrasankar Chatterjee, Siddhartha Bhattacharyya, *Senior Member, IEEE*, Bidyut B. Chaudhuri, *Life Fellow, IEEE*, and Jan Platoš, *Member, IEEE*

***Abstract*— Of late, convolutional neural networks (CNNs) find great attention in hyperspectral image (HSI) classification since deep CNNs exhibit commendable performance for computer vision-related areas. CNNs have already proved to be very effective feature extractors, especially for the classification of large data sets composed of 2-D images. However, due to the existence of noisy or correlated spectral bands in the spectral domain and nonuniform pixels in the spatial neighborhood, HSI classification results are often degraded and unacceptable. However, the elementary CNN models often find intrinsic representation of pattern directly when employed to explore the HSI in the spectral–spatial domain. In this article, we design an end-to-end spectral–spatial squeeze-and-excitation (SE) residual bag-of-feature (`S3EResBoF`) learning framework for HSI classification that takes as input raw 3-D image cubes without engineering and builds a codebook representation of transform feature by motivating the feature maps facilitating classification by suppressing useless feature maps based on patterns present in the feature maps. To boost the classification performance and learn the joint spatial–spectral features, every residual block is connected to every other 3-D convolutional layer through an identity mapping followed by an SE block, thereby facilitating the rich gradients through backpropagation. Additionally, we introduce batch normalization on every convolutional layer (ConvBN) to regularize the convergence of the network and scale invariant BoF quantization for the measure of classification. The experiments conducted using three well-known HSI data sets and compared with the state-of-the-art classification methods reveal that `S3EResBoF` provides competitive performance in terms of both classification and computation time.**

***Index Terms*— Bag-of-feature (BoF), convolutional neural networks (CNNs), hyperspectral image (HSI), residual network (ResNet).**

Manuscript received July 30, 2019; revised October 17, 2019 and November 17, 2019; accepted December 11, 2019. This work was supported by the European Regional Development Fund (ERDF) "A Research Platform focused on Industry 4.0 and Robotics in Ostrava," under Grant CZ.02.1.01/0.0/0.0/17_049/0008425. *(Corresponding author: Siddhartha Bhattacharyya.)*

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

Subhrasankar Chatterjee is with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: sc1933@cse.jgec.ac.in).

Siddhartha Bhattacharyya is with the Department of Computer Science and Engineering, Christ University, Bangalore, Bengaluru 560029, India (e-mail: dr.siddhartha.bhattacharyya@gmail.com).

Bidyut B. Chaudhuri is with the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: bbc@isical.ac.in).

Jan Platoš is with the Department of Electrical Engineering and Computer Science, VŠB-Techincal University of Ostrava, 708 00 Ostrava, Czech Republic (e-mail: jan.platos@vsb.cz).

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

Digital Object Identifier 10.1109/TGRS.2019.2961681

I. Introduction

HYPERSPECTRAL Images (HSIs) are composed in 2-D and contain the hundreds of near-continuous spectral bands of imagery in order to provide rich spectral and spatial information simultaneously. HSI analysis has been predominantly used in several challenging applications related to Earth observations [1] and remote sensing such as greenery detection, urbanization analysis [2], agriculture and crop analysis [3], and surveillance [4]. The widely used supervised and unsupervised classification approaches have been designed to classify HSI data [5]. In the past decade, most of the traditional methods used hand-crafted features for HSI classification [6], [7]. The different kinds of hand engineering features are modeled and fed to a classifier such as support vector machine (SVM) [8] or sparse representation classifier [9]. Mathematical morphology can better characterize the spatial structure in the HSI data. The morphological profile [10] is extracted and combined with the spectrum through early feature concatenation [11] or composite kernel learning [12]. The multiscale spatial structured feature can also be extracted using 3-D wavelets [13] and 3-D Gabor filtering [14]. These hand-engineered features lack the discriminative ability between interclass samples and robustness toward the geometric and photometric changes between intraclass samples. This is due to the fact that in remote sensing, in a majority of the cases classes are imbalanced. Li *et al.* [15] proposed an algorithm based on the orthogonal complement subspace projection (OCSP), which can adequately solve the problem of bound no. of labeled samples via unsupervised learning.

Among all the HSI classification methodologies discussed in recent literature, deep learning based methods have shown very promising performance for many image analysis related tasks, such as HSI classification [16], and deserve special attention to the remote sensing community.

A convolutional neural network (CNN) is a type of automatic feature extractor which directly deals with images and effectively characterize spectral–spatial features [17]. For example, AlexNet was the first CNN model proposed for large-scale image classification over the ImageNet data set [18]. Recently, several deep learning-based approaches like regular stacked autoencoders (SAEs) [19], stacked sparse autoencoders (SSAEs) [20], and deep belief networks (DBNs) [21] have been proposed and adopted for HSI classification [16]. The SAE [19] is the first proposal to stacked deep spectral–spatial feature representation. To extract deep spatial features representation from various scale, the SSAE is exploited by Zhao *et al.* [20] for HSI sparse feature representation. DBN is always combined with principal component analysis (PCA) transformation [21] to explore joint spectral–spatial feature for HSIs classification. Among the initial attempts, Makantasis *et al.* [22] used a supervised 2-D-CNN for HSI classification. Other notable CNN models proposed for HSI classification are deformable CNN [23], super-resolution-aided CNN [24], hierarchical CNN features [25], CNN with transfer learning [26] and Two-CNN [27]. Song *et al.* [28] introduced a deep feature fusion strategy to consider the correlated information among different layers and residual learning to reduce overfitting and gradient vanishing in the deep CNN. Data augmentation is also applied along with CNN for HSI classification to cope with the problem of large-scale database [29], [30]. In reality, to learn the weight parameters layer-wise the CNN requires a huge number of training samples. Unfortunately, a finite number of labeled instances can be accessible for HSIs, and hand-operated labeling of pixels is quite time-consuming. A generative adversarial network (GAN) has also been adapted by a few researchers to solve this problem [31].

The joint spatial–spectral feature encoding approach is the recent trend due to improved performance of fused CNN models for HSI classification tasks. However, most of the abovementioned methods utilize filters in 2-D shape and sacrifice an important spatial information loss which eventually may affect the classification performance under some complex scenarios [32]. The input HSI data are taken in the form of a 3-D cube encoded with two spatial and spectral dimensions. Chen *et al.* [33] adopted a 3-D CNN to calculate deep spectral–spatial representation directly from the input image cubes and have shown promising improvements. Hamida *et al.* [34] proposed a 3-D-CNN which works over 3-D volume input data and learns the joint spatial–spectral features. He *et al.* [35] introduced the multiscale 3-D-CNN by considering filters of varying dimensions. Similarly, a network called 3-D-LWNet [36] proposed to reduce the number of trainable model parameters by balancing the overall classification accuracy. Recently, Roy *et al.* [37] proposed a HybridSN model by sequential fusion of 3-D and 2-D CNNs for HSI classification. The 3-D CNN part of HybridSN extracts the joint spatial–spectral features, whereas the 2-D CNN part extracts the spatial information. Even this model generates good classification maps but the classification accuracy decreases rapidly while increasing the numbers of CNN layers. Zhong *et al.* [38] investigated the spectral–spatial residual network (SSRN) which is also a 3-D-CNN model. Yu *et al.* [39] used a three-layer CNN and fused the features of all three layers. Liang *et al.* [40] extracted the spectral and spatial features separately and fused those using a cooperative sparse autoencoder. Zhou *et al.* [41] also extracted the spatial and spectral features using individual CNNs and then used the compact and discriminative SAE to learn the latent space with reconstruction loss. The encoder is thereafter fine-tuned using supervised learning to facilitate classification. Kang *et al.* [42] proposed a dual-path network consisting of ResNet and DenseNet. The fusion is performed by sharing the bottleneck layer of both networks to learn the joint features. Recently, Paoletti *et al.* [43] introduced a deep pyramidal residual network (DPResNet) for spatial and spectral features at multiple scales to learn complementary spectral–spatial information [44] and CapsNet [45] in order to achieve significant improved performance by reducing the network complexity. However, the above described models could produce good classification accuracy by adding more number of layers thus increasing the numbers of layer-wise trainable parameters and hence the resultant system requires expensive graphical processing unit (GPU) hardware [46].

To solve the above problems and inspired by [46]–[48], we designed a supervised efficient lightweight spectral–spatial squeeze-and-excitation residual bag-of-feature (`S3EResBoF`) learning framework by considering consecutive spectral and spatial squeeze-and-excitation (SE) residual blocks to characterize the HSI channel-wise spectral–spatial features better. Then, we build a scale invariant BoF representation for classification. This article investigates the success of the proposed `S3EResBoF` framework toward spectral–spatial BoF learning and validates robustness in various scenarios.

This article is arranged as follows. The motivation behind the proposed `S3EResBoF` model is described in Section II. The proposed framework is detailed in Section III. The results and analysis are presented in Section IV; and finally the conclusions are drawn in Section V.

II. Motivation

HSIs are captured by scanning the same region in different spectral bands. These spectral bands may introduce some degree of correlations, i.e., two successive bands may output similar visualizations. Thus, it is desirable to consider these hyperspectral correlations. Recently, deep CNNs have shown great performance improvement in HSI classification due to the fact that convolutional filters act as an excellent spectral–spatial feature extractor from raw HSI data. However, training large CNNs model with HSI data is still challenging due to the loss of information effected by the gradient vanishing problem. Moreover, the gradient generated from the activation feature maps of each convolutional layer becomes modest by propagating poor activation information including gradients, which directly affect the cost function of the network. This hampers the network convergence, and the classification accuracy saturates and then degrades rapidly. To handle the above-mentioned challenges and to ensure minimum information loss after each convolutional operation, the SEResNet [47] is designed using a residual block followed by an

SE transformation. The residual block is represented with an identity mapping and adding them using a shortcut connection helps to receive more structural feature representation rather than abstract representation. On the other hand, the SE block slightly improves the quality of feature representation produced through residual block by explicitly modeling the relationship between the channels of its convolutional feature maps. Similarly, for CNN architecture, the dimensionality of the output convolutional feature extraction block varies with the size of the input image and often the size of the extracted feature vector is large, leading to a large fully connected layer which may become less discriminative. It is worth reporting that for the well-known ResNet [46], 90% of trainable weights of network parameters are used on the fully connected layer, whereas the remaining 10% weights surprisingly are used on the previous layer of the network. However, because of available limited labeled training samples and due to the extensive trainable model parameters, deep CNN methods may suffer from overfitting. This can be overcome by a trainable quantization oriented pooling layer referred to as BoF pooling [48] for compact representation. This can deal with a variable number of feature vectors and is capable of preserving the better scale, position invariance and also prevent overfitting in the training stage. This is done because it is essential to encode the spectral information in the HSI classification problem. To exploit the interdependence between the spectral–spatial domains and to extract features which can be represented by less number of variables while maintaining the discriminative power, it is more advantageous to learn the spectral–spatial feature jointly using an end-to-end framework. Hence, this allows us to model an efficient S3ERESBOF learning framework.

III. PROPOSED S3ERESBOF LEARNING FRAMEWORK

Suppose a spectral–spatial HSI, $\mathbf{X}_{\text{org}} \in \mathcal{R}^{M\times N\times D}$, is described by two spatial parameters the width M, height N, and one spectral dimension D, respectively. Let $\mathbf{X}_{\text{org}}$ contain C labeled pixels defined as $Y = (y_1, y_2, \ldots, y_C)$. We define the spectral vector as $\mathbf{x}_{i,j} = [x_{i,j,1}, \ldots, x_{i,j,D}] \in \mathcal{R}^D$, where $\mathbf{x}_{i,j} \in \mathbf{X}_{\text{org}}$ represents the pixel at location (i, j) in any spectral band with $i = 1, \ldots, M$, and $j = 1, \ldots, N$, respectively. However, the hyperspectral data exhibit mixed land-cover classes thereby introducing the high intra-class variability and high inter-class similarity between the classes of $\mathbf{X}_{\text{org}}$. PCA is initially used over the original HSI data, $\mathbf{X}_{\text{org}}$ in order to eliminate the spectral redundancy among the spectral bands. PCA helps to minimize the spectral bands from D to B without affecting the spatial dimensions. We represent the output of PCA as the reduced HSI data cube and denote it by $\mathbf{X}_r \in \mathcal{R}^{M\times N\times B}$, where both M and N remain the same as the width, and the height, whereas B is the number of spectral bands after application of PCA. It is better to extract pixel neighboring region in the preprocessing step to be defined as $\mathbf{x}_{i,j} \in \mathcal{R}^{S\times S}$ centered at pixel (i, j). The mostly used the spectral–spatial HSI classification approaches are based on 3-D CNNs, where a 3-D architecture is used to jointly extract discriminative spectral–spatial information. In order to create the data blocks that easily feed into the input of the network, the pixel neighboring spectral–spatial regions $\mathbf{x}_{i,j} \in \mathcal{R}^{S\times S}$ are extracted from raw HSI. If $\mathbf{x}_{i,j}$ includes the spectral information, it can be further defined as $\mathbf{x}_{i,j} \in \mathcal{R}^{S\times S\times B}$. All the extracted image cubes, $\mathbf{x}_{i,j}$ are stacked and represented by $\mathbf{X}$. To fulfill the desired aim of performing spectral–spatial HSI classification correctly, we present a deep feature learning framework that utilizes both sources of information by taking into account the spectral signature of each pixel $\mathbf{x}_{i,j} \in \mathbf{X}_r$ and its spatial neighborhood around it. However, it is crucial to training a very deep CNN model when the depth increases and the loss of information produced at different levels of activation layer which may suffer from the dying gradient problem [49]. This is due to the zero-hard rectification behavior of its inputs by some existing activation functions, which also fails to utilize the large negative input values.

In this article, we propose an efficient lightweight S3ERESBOF learning framework for HSI classification as illuminated in Fig. 1. To design the proposed S3ERESBOF network a 3-D convolutional layer [38] along with batch normalization (BN) [50] are adopted as basic building blocks which make the deep residual training [shown in Fig. 1(b)] of the learning model more efficient and leads the model to faster convergence. Although CNN models have shown state-of-the-art classification performance but introducing larger number of convolutional layers hampers the convergence of the network and saturates the accuracy initially which drops down rapidly. However, this issue can be alleviated by efficiently adding shortcut connections between every other layer to build residual blocks [46]. Whereas every residual block also followed by an SE block as shown in Fig. 1(c). The SE blocks dynamically "excite" feature maps that help classification and suppress ineffective feature maps that do not help based on the patterns produced by global average pooling of feature maps. The proposed model uses two kinds of a residual block to extract spectral–spatial feature representation. Finally, a light weighted trainable BoF pooling [48] is employed to quantize the feature maps of the last convolutional layer. This basically allows the classification of various sized images by minimizing the used parameters in the network and those parameters of CNN are learned through the backpropagation algorithm [51]. Hence, the proposed end-to-end CNN architecture becomes trainable. The ability of S3ERESBOF framework is to reduce the size of CNN features by retaining the highly discriminating features which are capable of increasing the HSI classification performance and to make the model competitive to other state-of-the-art models. The step-by-step description of the proposed S3ERESBOF framework is given in detail below.

As shown in Fig. 1, the 3-D convolutional $(l+1)$th layer has v^l input feature cubes having size $S^l \times S^l \times B^l$ and the convolutional layer contains v^{l+1} trainable convolutional filters of size $k^{l+1} \times k^{l+1} \times d^{l+1}$. Sub-sampling strides (s_1, s_1, s_2) are used in the convolutional operations. Then the ith layer produces output of size $S^{l+1} \times S^{l+1} \times B^{l+1}$, where the spatial width and spectral depth can be determined by $S^{l+1} = [1+(S^l - k^{l+1})/s_1]$ and $B^{l+1} = [1 + (B^l - d^{l+1})/s_2]$, respectively. The ith output of 3-D convolutional+batch normalization ($ConvBN$) in the

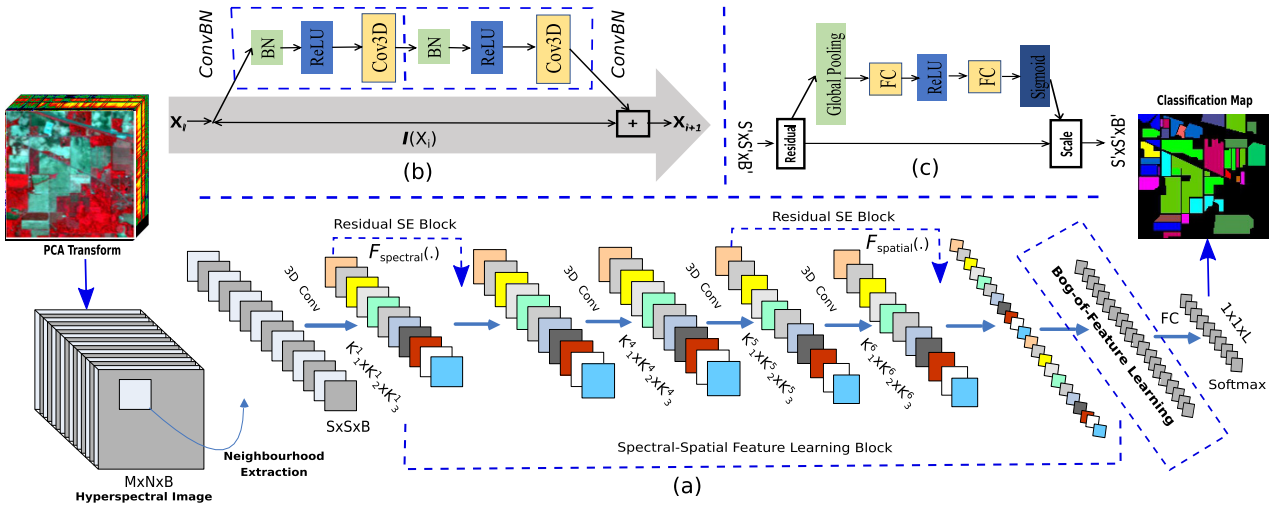


Fig. 1. (a) S3EResBoF learning framework based on ResNet8 for HSI classification. Initially, sample image cubes of sized $S \times S \times B$ are extracted from a neighborhood window centered around the target pixel from raw HSI and then fed through S3EResBoF to extract deep joint spectral–spatial features for accurate estimate of the classification scores. (b) and (c) Basic building block of ResNet and SE block, respectively.

$(l+1)$th layer can be obtained as

$$X^{l+1} = \phi\left(\sum_{j=1}^{v^l} \mathcal{F}_{bn}\left(X_j^l\right) * W_i^{l+1} + b_i^{l+1}\right)$$

$$\mathcal{F}_{bn}(X^l) = \frac{X^l - \mu(X^l)}{\sqrt{\sigma^2(X^l) + \epsilon}}.\gamma + \beta \tag{1}$$

where $X_j^l \in \mathcal{R}^{S\times S\times B}$ is the jth feature map of the $(l+1)$th layer, $\mathcal{F}_{\text{bn}}(.)$ is the BN function of the feature cubes X^l in the lth layer, $\mu(.)$ and $\sigma^2(.)$ represent the mean and variance function of the input feature maps, respectively. W_i^{l+1} and b_i^{l+1} represent the kernel parameters and bias of ith filter bank in $(l+1)$th layer. $\phi(.)$ and * denote ReLU activation function and 3-D convolutional operation while γ and β are the learnable vector parameters.

The proposed S3EResBoF architecture comprises two squeeze-and-excitation residual (SERes) blocks to extract the channel-wise spectral and spatial features from original HSI inputs. To describe the end-to-end S3EResBoF architecture first, we need to write about the kind of transformations for the input feature maps $X \in \mathcal{R}^{S\times S\times B}$, that passes through a combination of encoder/decoder block $\mathcal{F}_{\text{tr}}$ to generate the output feature map $\hat{X} \in \mathcal{R}^{S'\times S'\times B'}$, i.e., $\hat{X} = \mathcal{F}_{tr}(X)$ where S/S', S/S', and B/B' are the input–output spatial height, width, and channels, respectively, so that the SE block can efficiently be constructed. The generated $\hat{X}$ combines both the spatial and channel information of X through a sets of convolutional operations and nonlinearities represented by $\mathcal{F}_{tr}(.)$. In the residual block shown in Fig. 1(a) and (b), the kernel of two successive convolutional layers are represented with the filter banks W^{l+1} and W^{l+2} having sizes of $k_1^l \times k_2^l \times k_3^l$ for lth and $(l+1)th$ layers, respectively. However, the spatial size of resultant 3-D feature maps X^{l+1} and X^{l+2} are in two consecutive layers and the spatial dimensions of output feature maps keep unchanged through a boundary padding strategy. Finally, these two conv layers create a feed-forward residual function $\mathcal{F}_{\text{res}}(X^l;\theta)$ and identity mapping of X^l by adding shortcut connections as shown in Fig. 1(b). The transformation of the residual function can be developed as follows:

$$X_j^{l+2} = \mathcal{I}\left(X_j^l\right) + \mathcal{F}_{\text{res}}\left(X_j^l;\theta\right) \tag{2}$$

$$\mathcal{F}_{\text{res}}\left(X_j^l;\theta\right) = \phi\left(X_j^{l+1}\right) * W_j^{l+2} + b_j^{l+2} \tag{3}$$

$$X_j = \phi\left(X_j^l\right) * W_j^{l+1} + b_j^{l+1} \tag{4}$$

where X^{l+1} represents the input 3-D feature maps of the $(l+1)$th layer, $\theta = \{W_j^{l+1}, W_j^{l+2}, b_j^{l+1}, b_j^{l+2}\}$, $W_j = \{W^{l+i}|1 \le i \le 2\}$ and $b_j = \{b_j^{l+i}|1 \le i \le 2\}$ denote the weight matrix and bias of the $(l+1)$th and $(l+1)$th ConvBN layers associated with the jth residual block, and $\mathcal{I}(X_j^l) = X_j^l$ is the identity mapping function. The shape of the convolutional kernel settings determines the kind of features extracted by the residual block from the input cubes. The aim is to learn the residual function $\mathcal{F}_{\text{res}}(X^l;\theta)$ with respect to $\mathcal{I}(X_j^l) = X_j^l$. If the filter bank contains kernel of size $k_1^l \times k_2^l \times k_3^l$ where $k_1^l = k_2^l = 1$, and k_3^l is set to strictly less than the number of channels in input (B), then the learning process is referred to as spectral feature learning $\mathcal{F}_{\text{spectral}}(.)$. On the other hand, when the filter bank contains kernel of size $k_1^l \times k_2^l \times k_3^l$ where $k_1^l = k_2^l > 1$, and $k_3^l = B$, which is equal to the number of channels in input, then the residual block is referred to as spatial feature learning $\mathcal{F}_{\text{spatial}}(.)$. The architecture shown in Fig. 1(a) contains two residual blocks where the initial residual block is used for spectral and the second one for spatial feature learning purpose.

Finally, we add the SE block, $\mathcal{F}_{\text{SE}}(.)$ immediately after the residual transform to perform *feature recalibration* of the transform feature maps on $\hat{X}$. The SE block combines two sequential operations, i.e., squeeze $\mathcal{F}_{\text{sq}}(.)$ and excitation $\mathcal{F}_{\text{ex}}(.)$. The goals of *feature recalibration* is to improve the nature of feature representations produced by the ResNet [46], which surprisingly introduces interdependences between the

channel/bands of its convolutional features. The SE block takes U feature maps as its input and in order to factor out the dependence over the spatial dimension ($S' \times S'$) through an average pooling taken globally to learn a channel-wise descriptor it passed through a squeeze function, $\mathcal{F}_{\text{sq}}$. The goal of the descriptor is to emphasize on useful channels through the feature maps *recalibrate* so as to embed the global distribution of feature maps across different channels. The channel descriptor is obtained using a *squeeze* operation followed by an *excitation* operation. The nomenclature is motivated by the fact that the SE block "squeezes" along the spatial dimensions and "excites" or reweights along the channels which as noted to be effective for classification task among computer vision communities. We describe the SE block in the form of residual transformation. Let us consider the input feature map $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{B'}]$ as a combination of channels $\hat{x}_i \in \mathcal{R}^{S' \times S'}$. The spatial squeeze is performed globally through an average pooling layer to produce the vector $z \in \mathcal{R}^{1\times 1\times B'}$ of its kth element which is given as

$$z_k = \mathcal{F}_{sq}(\hat{x}_c) = \frac{1}{S' \times S'} \sum_{i=1}^{S'} \sum_{j=1}^{S'} \hat{x}_c(i, j). \tag{5}$$

This process embeds the spatial information globally into a vector z. Since it is capable of learning a nonmutually exclusive relation between the convolutional channels thereby ensuring that multiple channels are allowed to be emphasized. This vector z is transformed to s using the excitation operation, $\mathcal{F}_{\text{sq}}$ and is given as

$$s = \mathcal{F}_{ex}(z, W) = \sigma(\phi(z, W)) = \sigma(W_1 \phi(W_2, z)) \tag{6}$$

where $W_2 \in \mathcal{R}^{B'\times B'/r}$, $W_1 \in \mathcal{R}^{B'/r\times B'}$ are the weights of two successive dense layers, and $\phi(.)$ and r denote the *ReLU* activation and reduction ratio, respectively. The *recalibrate* or excite of the resultant vector from $\hat{X}$ to $\bar{X}$ by a rescaling the transformation output as

$$\hat{x_c} = \mathcal{F}_{\text{scale}}(\bar{x_c}, s_c) = s_c . \hat{x}_c \quad \forall_{c\in 1,\ldots,B'} \tag{7}$$

where $\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{B'}]$ and the channel-wise multiplication between the scalar s_c and feature maps $\hat{x}_c \in \mathcal{R}^{S\times S}$ is carried out using $\mathcal{F}_{\text{scale}}(\hat{x}_c, s_c)$.

After the feature extraction for the ith HSI image cubes using a trainable spectral–spatial convolutional feature extractor, the S3EResNet, shown in Fig. 2(a), is represented by $\bar{X}_{ij} \in \mathcal{R}^{B_l'}(i = 1, \ldots N_s; j = 1, \ldots, N_{\text{fe}})$ where N_s, N_{fe}, and B_l' denote the total number of image samples, the length of feature vectors, and the number of channels in the lth convolutional layer, respectively. Instead of simply flattening it into feature vector and fitting that to a *dense* layer to perform the classification task, a variable length BoF codebook is learned where the feature vector $\bar{\mathcal{X}} = \{\bar{x}_{ij} | i = 1, \ldots N_s; j = 1, \ldots, N_{\text{fe}}\}$ is clustered into P_k clusters with the corresponding centroids $v_k \in \mathcal{R}^{B_l'}(k = 1, \ldots, P_k)$ being used to build the codebook $V \in \mathcal{R}^{B_l'\times P_k}$, where the columns of V represent the cluster centroid which are used to quantize the feature vector. To encode the ith feature map, the differentiable normalized similarity function $\Phi(\bar{x}_{ij})$ is calculated [shown in Fig. 2(b)] between the feature vector $\bar{x}_{ij}$ and each centroid of v_k using the kth radial basis function (RBF) as

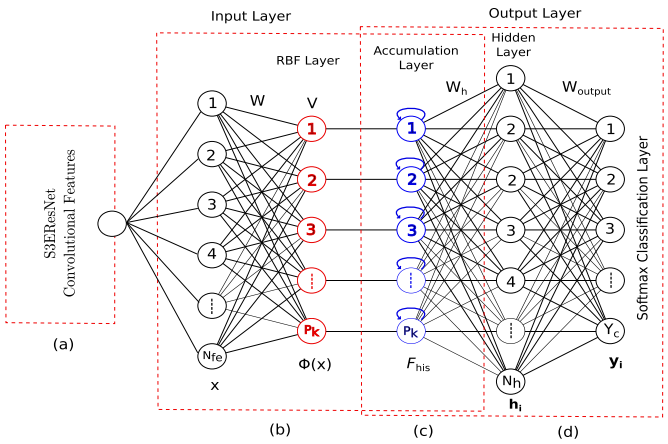


Fig. 2. (a) S3EResNet block for convolutional feature extraction. (b) BoF architecture where the nodes marked in red are RBF neurons. (c) Nodes marked in blue are recurrent accumulation neurons. (d) Next layer is a hidden layer followed by a $softmax$ layer for classification.

$$\Phi(\bar{x}_{ij})_k = \frac{\exp\left(-||\bar{x}_{ij} - v_k||_2/\sigma_k\right)}{\sum_{i=1}^{P_k} \exp\left(-||\bar{x}_{ij} - v_i||_2/\sigma_i\right)} \in \mathcal{R} \tag{8}$$

where σ_k is the scaling hyper-parameter at the Gaussian function of each RBF neuron. The controlled of extracted features dimension depend upon the number of RBF neurons (P_k) used in the BoF layer. It can be noted that the encoding process and training are completely done in an unsupervised manner where no labeled information are required. The final histogram representation, $\mathcal{F}_{\text{his}}$ [shown in Fig. 2(c)], for each sample can be built by assembling the RBF neuron's responses for each feature vector that is fed to the BoF layer as

$$\mathcal{F}_{\text{his}} = \frac{1}{N_{\text{fe}}} \sum_{j=1}^{N_{\text{fe}}} \Phi(\bar{x}_{ij}) \in \mathcal{R}^{P_k}. \tag{9}$$

The built histogram, $\mathcal{F}_{\text{his}}$, has unit $\mathcal{L}_1$ norm which defines the distribution over RBF neurons and describes the visual content of each HSI class and the output of RBF layer represented by $\Phi(\bar{x}) = ([\Phi(\bar{x}_1)], [\Phi(\bar{x}_2)], \ldots, [\Phi(\bar{x}_{P_k})])^T \in \mathcal{R}^{P_k}$. These are fed to a *dense* layer for the classification task. The last layer of S3EResBoF is the $softmax$ layer, as shown in Fig. 2(d), which generates the class scores for Y classes and acts like probability $\mathcal{P}r(.)$. Let $\theta = \{W_j^l, b_j^l\}$, denote the layer-wise weight matrix and bias vectors of the proposed model which are usually trained using supervised backpropagation training algorithm [51]. Here, we adopt a cross-entropy objective function by maximizing the likelihood of each class score $\mathcal{F}_{\mathcal{M}}^c(x_{i,j}; \theta)$ into the conditional probabilities using $softmax$ function as

$$\mathcal{P}r(c|x_{i,j}; \theta) = \frac{\exp^{(\mathcal{F}_{\mathcal{M}}^c(x_{i,j};\theta))}}{\sum_{c\in\{y_1,\ldots,y_L\}} \exp^{(\mathcal{F}_{\mathcal{M}}^c(x_{i,j};\theta))}} \tag{10}$$

where $\mathcal{F}_{\mathcal{M}}^c$ function represents the probability score generated using S3EResBoF for the cth class, y_C is the number of land-cover categories, and $\mathbf{x}_{i,j} \in \mathcal{R}^{S\times S\times B}$ is the input

image cubes. The parameters θ are learned by minimizing the negative log-likelihood defined as

$$\mathcal{L}(\theta) = -\sum_{K_{i,j}} \ln \mathcal{P}r(Y_{i,j,C}|x_{i,j};\theta) \tag{11}$$

where Y_{i,j,y_L} is the true class of the data volume $\mathbf{x}_{i,j}$ corresponding to the (i, j) spatial location. During testing time, the model calculates the label using the function argmax as

$$\hat{Y}_{i,j,y_C} = \arg \max_{c\in\{1,\ldots,C\}} p(c|k_{i,j};(W,b)). \tag{12}$$

IV. Experiments and Discussion

In order to extensively evaluate the performance of S3EResBoF framework, we carry out different experiments on three well-known HSI data sets and compare with several state-of-the-art methods in this section. All experiments are conducted on a system with the NVIDIA Titan V 12-GB GPU and 128 GB of RAM. Regarding the software settings, all the experiments are carried out on 64-bit Ubuntu 16.04LTS operating system, CUDA 9 and Keras deep learning library with Tensorflow backend and Python 3.5.2 programming language are also used. The optimal learning rate [52] is chosen as 0.0003, based on the outcomes of HSI classification. Training is performed for 200 epochs with a batch size of 32 over each HSI data sets.

A. Hyperspectral Data Sets

In our experiment, three publicly available well-known HSI data sets are used, viz., Indian Pines (IP), University of Pavia (UP), and Salinas Scene (SA), respectively. A more detailed description of each HSI data set is given as follows.

1) The IP data sets was collected by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [53] sensor over the IP test site in North-western Indiana in 1992. IP has images with 145 × 145 spatial dimension of 20 m/pixel and spectral bands of width 224 with wavelength ranging from 400 to 2500 nm, 24 bands have been removed out of which four null spectral bands and the rest are corrupted due to the atmospheric water absorption. 16 mutually exclusive vegetation classes are present in the IP data set. However, about 50% (10 249) pixels from a total of 21 025 contain ground truth information from 16 different classes.
2) The UP data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor at the time of a flight campaign over the university campus at Pavia, Northern Italy in 2001. It consists spatially with 610 × 340 pixels, and spectral information are captured within 103 bands in the range of wavelength 430 to 860 nm with 1.3-mpp spatial resolution. The groundtruth is designed to have nine urban land-cover classes. Moreover, about 20% of the total 207 400 pixels contain ground truth information.
3) The SA data set was gathered by the 224-bands AVIRIS sensor over the Salinas Valley, CA, USA, in 1998 and the images comprise with spatial dimension 512 × 217 and spectral information are encoded in 224 bands having the range of wavelength 360–2500 nm. As in the case of IP data, 20 spectral bands have been discarded due to water absorption. In total, 16 ground truth classes, such as vegetation, bare soils and vineyard fields are already present in this scene.

B. Experimental Configuration

In order to assess the results of the proposed S3EResBoF learning framework with respect to the most widely used supervised methods which are available in the review literature [16], such as SVMs [8], 2-D-CNN [22], 3-D-CNN [34], M3D-CNN [35], SSRN [38], Two-CNN [27], CapsNet [45], and DPResNet [43], respectively. Based on the above classification methods, the following four sets of experiment are performed.

1) The first experiment evaluates the class-specific classification performance in terms of overall accuracy (OAs), average accuracy (AAs), and kappa using the proposed S3EResBoF framework for three HSI data sets, IP, UP, and SA, respectively.
2) The second experiment is carried out and compared with the well-known supervised methods, i.e., 2-D-CNN [22], 3-D-CNN [34], M3D-CNN [35], SSRN [38], Two-CNN [27], CapsNet [45], DPResNet [43], and HybridSN [37] using the training sets of 10% and 20% of available IP, UP, and SA labeled data sets. Moreover, the spatial window size of the 3-D volume data is kept as $15\times15\times30$ for IP and $15\times15\times15$ for both UP and SA, respectively, where 30 and 15 are the selected spectral bands after application of PCA. In addition, a similar experimental setting is used to carry out the popular extreme learning machine (ELM) [6], SVM, and random forest (RF) [54], based spectral classifiers.
3) The third experiment evaluates the performance of the proposed spectral–spatial classifiers using three different spatial sizes, i.e., like 15 × 15, 13 × 13 and 11 × 11, respectively. In training configuration, we consider 3%, 5%, and 10% of the available labeled data randomly taken from each class of HSI data sets.
4) The fourth experiment is performed to compare with the spectral–spatial classifier, SSRN [38] and DPResNet [43] using four varying spatial window sizes, i.e., 5 × 5, 7 × 7, 9 × 9, and 11 × 11, respectively. The classification results of both the methods SSRN and DPResNet are taken by running their implementation. Here, 20% and 10% of training samples are used for IP and UP data sets, respectively.

We have considered three standard evaluation criteria to measure the classification performance: OA, AA, and kappa coefficient (kappa). Here, OA is determined by the number of samples correctly classified out of the whole test samples; AA by the average of class-wise classification accuracies; and Kappa is a metric of statistical measurement which provides mutual information regarding a strong agreement between the ground truth map and generated classification map. In order to perform unbiased comparison, we extract various small

TABLE I

NUMBER OF TRAINING, VALIDATION AND TEST SAMPLES WITH CLASS-WISE ACCURACY INCLUDING OAS, KAPPAS, AND AAS METRICS ON THREE HSI DATA SETS VIZ., IP, SA, AND UP, RESPECTIVELY

Indian Pines (IP)						Salinas Scene (SA)						University of Pavia (UP)					
Class	Name	Training	Validation	Test	Accuracy	Class	Name	Training	Validation	Test	Accuracy	Class	Name	Training	Validation	Test	Accuracy
1	Alfalfa	6	2	38	100	1	Brocoli_green_weeds_1	301	100	1608	100	1	Asphalt	994	331	5306	100
2	Corn-no till	214	71	1143	98.45	2	Brocoli_green_weeds_2	558	186	2982	100	2	Meadows	2797	932	14920	100
3	Corn-min till	124	41	665	99.62	3	Fallow	296	98	1582	100	3	Gravel	314	104	1681	99.84
4	Corn	35	11	191	100	4	Fallow_rough_plow	209	69	1116	100	4	Trees	459	153	2452	100
5	Grass-pasture	72	24	387	100	5	Fallow_smooth	401	133	2144	100	5	Painted metal sheets	201	67	1077	100
6	Grass-trees	109	36	585	100	6	Stubble	593	197	3169	100	6	Bare Soil	754	251	4024	100
7	Grass-pasture-mowed	4	1	23	100	7	Celery	536	178	2865	100	7	Bitumen	199	66	1065	100
8	Hay-windrowed	71	23	384	100	8	Grapes_untrained	1690	563	9018	100	8	Self-Blocking Bricks	552	184	2946	99.80
9	Oats	3	1	16	100	9	Soil_vinyard_develop	930	310	4963	100	9	Shadows	142	47	758	100
10	Soybean-no till	145	48	779	99.50	10	Corn_senesced_green_weeds	491	163	2624	100						
11	Soybean-min till	368	122	1965	100	11	Lettuce_romaine_4wk	160	53	855	100						
12	Soybean-clean	88	29	476	99.21	12	Lettuce_romaine_5wk	289	96	1542	100						
13	Wheat	30	10	165	100	13	Lettuce_romaine_6wk	137	45	734	100						
14	Woods	189	63	1013	100	14	Lettuce_romaine_7wk	160	53	857	100						
15	Buildings-Grass-Trees-Drives	57	19	310	100	15	Vinyard_untrained	1090	363	5815	100						
16	Stone-Steel-Towers	13	4	76	99.15	16	Vinyard_vertical_trellis	271	90	1446	100						
	OA	1528	505	8216	99.82 ± 0.1		OA	8112	2697	43320	100 ± 0.0		OA	6412	2135	34229	99.97 ± 0.0
	Kappa				99.79 ± 0.1		Kappa				100 ± 0.0		Kappa				99.94 ± 0.0
	AA				99.79 ± 0.1		AA				100 ± 0.0		AA				99.96 ± 0.0
	Time				2390.137 ms		Time				2264.211 ms		Time				2317.216 ms

spatial dimensions in 3-D-patches of input volume for each data sets, such as $15 \times 15 \times 30$, $13 \times 13 \times 30$, and $11 \times 11 \times 30$ for IP data set and $15 \times 15 \times 15$, $13 \times 13 \times 15$, and $11 \times 11 \times 15$ for both UP and SA data sets, respectively. All the given data sets are randomly divided into three segments viz., training, validation, and testing. Then, 20% and 10% labeled samples are selected from each class for training. Also 5% unlabeled samples from each class was taken as a validation set and the remaining samples are utilized for testing. Moreover, the evaluated mean ± std value performance of the proposed method is reported to avoid the inclination in sampling and the marginal performance, the experiments were repeated ten times.

To show the convergence of training and validation process progress, the available labeled samples are randomly partitioned into training (20%), validation (10%), and testing (70%) groups, respectively. Additionally, for the sake of fair comparison of the above-mentioned methods, the evaluation was done with the same operational settings using the publicly available code[1]. The source code of S3EResBoF will be made available publicly at https://github.com/swalpa/S3EResBoF.

C. Classification Results

The first described experiments were performed using the proposed S3EResBoF model to evaluate the class-wise classification accuracies. Table I presents the classification results using 20% of training samples for IP, UP, and SA data sets, respectively, where the first to fifth columns of each data set indicate the number of classes, the corresponding name of the classes, the number of training, validation, and testing samples taken from the available labeled data sets, respectively. The sixth column shows the class-wise classification accuracy obtained by the S3EResBoF model. The three rows below contain the result of an evaluation matrix such as OA, AA, and kappa along with time (ms) taken by samples using the proposed model while training. In addition to class-specific performance, the effects with respect to different regularization techniques are also evaluated. To overcome the overfitting of training data, BN and 50% of dropout are used to accelerate the deep training of the model. Table II shows the overall

[1]https://github.com/eecn/Hyperspectral-Classification

TABLE II

OA (%) FOR S3ERESBOF USING DIFFERENT REGULARIZER METHODS WITH EFFECTS OF OCCLUSION WHERE OCCLUSION (LOW/HIGH) REPRESENTS LOWEST AND HIGHEST ACCURACY USING DIFFERENT OCCLUSION RATES, I.E., 25% AND 50%, RESPECTIVELY

S3EResBoF	Indian Pines	University of Pavia	Salinas Screne
None	98.42 ± 0.2	99.88 ± 0.0	99.91 ± 0.0
BN	98.78 ± 0.2	99.90 ± 0.0	99.97 ± 0.0
Dropout	99.10 ± 0.1	99.94 ± 0.0	99.98 ± 0.0
Both	**99.82 ± 0.1**	**99.97 ± 0.0**	**100 ± 0.0**
Occlusion (Low)	98.53 ± 0.2	99.19 ± 0.1	99.29 ± 0.1
Occlusion (High)	99.31 ± 0.1	99.90 ± 0.0	99.94 ± 0.0

classification performance of the model without considering any regularization technique, with BN and dropout, and finally using both the techniques, i.e., BN+dropout under the same experimental settings. In addition to this the last two rows of Table II also reports the lowest and highest OAs among two different occlusion rates, i.e., 25% and 50% using spatial input window of sizes, i.e., 15 × 15, and 23 × 23 for the proposed S3EResBoF model. It can also be observed from Table II, that the S3EResBoF model achieves the best performance when both the regularization techniques are employed for the three HSI data sets.

This experiment validates the proposed model with respect to the most common and well-known state-of-the-art and deep learning based HSI classification methods [16] such as SVM along with kernel function of radial basis [8], ELM [6], 2-D-CNN [22], 3-D-CNN [34], M3D-CNN [35], spectral–spatial ResNet (SSRN) [38], DPResNet [43], and HybridSN [37]. The RF, SVM, NRS, and ELM are the spectral feature classifiers, 2-D-CNN is a popular spatial feature classification technique and 3-D-CNN, M3D-CNN, Two-CNN, SSRN, DPResNet, and HybridSN are recent spectral–spatial classification methods. The experimental results of the proposed S3EResBoF model are compared and the OAs, AAs, and kappa coefficients are reported in Table III. In this experiment, lesser number of labeled data, i.e., 10% and 20% are used for training as per the standard and the spatial dimension of input image cubes is kept as 15×15. It can be seen that the S3EResBoF achieves better or comparable performance on three data sets in terms of all performance measures. In contrast, the variation among

TABLE III

CLASSIFICATION ACCURACIES (IN PERCENTAGES) USING THE PROPOSED S3EResBoF AND OTHER STATE-OF-THE-ART METHODS ON 10% AND 20% AMOUNT OF TRAINING DATA

Training Samples	Methods	Indian Pines Dataset			University of Pavia Dataset			Salinas Scene Dataset		
		OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA
10%	SVM	81.67 ± 0.65	78.76 ± 0.77	79.84 ± 3.37	90.58 ± 0.47	87.21 ± 0.70	92.99 ± 0.36	94.46 ± 0.12	93.13 ± 0.34	93.01 ± 0.60
	2D-CNN	80.27 ± 1.2	78.26 ± 2.1	68.32 ± 4.1	96.63 ± 0.2	95.53 ± 1.0	94.84 ± 1.4	96.34 ± 0.3	95.93 ± 0.9	94.36 ± 0.5
	3D-CNN	82.62 ± 0.1	79.25 ± 0.3	76.51 ± 0.1	96.34 ± 0.2	94.90 ± 1.2	97.03 ± 0.6	85.00 ± 0.1	83.20 ± 0.7	89.63 ± 0.2
	M3D-CNN	81.39 ± 2.6	81.20 ± 2.0	75.22 ± 0.7	95.95 ± 0.6	93.40 ± 0.4	97.52 ± 1.0	94.20 ± 0.8	93.61 ± 0.3	96.66 ± 0.5
	SSRN	98.45 ± 0.2	98.23 ± 0.3	86.19 ± 1.3	99.62 ± 0.0	99.50 ± 0.0	99.49 ± 0.0	99.64 ± 0.0	99.60 ± 0.0	99.76 ± 0.0
	Two-CNN	96.71 ± 0.1	96.10 ± 0.10	96.16 ± 0.12	97.71 ± 0.1	97.62 ± 0.1	97.45 ± 0.2	97.12 ± 0.30	96.98 ± 0.20	97.00 ± 0.20
	DPResNet	99.04 ± 0.09	98.97 ± 0.07	98.93 ± 0.07	99.67 ± 0.1	99.58 ± 0.1	99.27 ± 0.2	99.97± 0.0	99.97 ± 0.0	99.97 ± 0.0
	HybridSN	98.39 ± 0.4	98.16 ± 0.5	98.01 ± 0.5	99.72 ± 0.1	99.64 ± 0.2	99.20 ± 0.2	**99.98 ± 0.0**	**99.98 ± 0.0**	**99.98 ± 0.0**
	Proposed	**99.49 ± 0.2**	**99.42 ± 0.2**	**99.20 ± 0.3**	**99.77 ± 0.1**	**99.69 ± 0.1**	**99.60 ± 0.1**	**99.98 ± 0.0**	**99.98 ± 0.0**	99.96 ± 0.0
20%	SVM	86.24 ± 0.38	84.27 ± 0.45	83.15 ± 1.10	95.20 ± 0.13	93.63 ± 0.17	93.60 ± 0.14	94.15 ± 0.10	93.48 ± 0.11	97.23 ± 0.11
	2D-CNN	86.90 ± 1.3	85.01 ± 1.6	82.70 ± 1.0	96.02 ± 0.4	96.04 ± 0.3	95.10 ± 0.1	96.15 ± 0.6	95.71 ± 0.7	98.27 ± 0.2
	3D-CNN	89.23 ± 0.2	87.70 ± 0.3	87.87 ± 0.1	97.30 ± 0.3	96.22 ± 0.1	97.02 ± 0.1	94.54 ± 0.5	93.81 ± 0.3	96.79 ± 0.6
	M3D-CNN	93.67 ± 0.1	92.70 ± 0.3	93.60 ± 0.6	97.41 ± 0.2	96.05 ± 0.6	98.22 ± 0.1	94.92 ± 0.3	94.40 ± 0.1	97.28 ± 0.2
	SSRN	99.23 ± 0.1	99.12 ± 0.1	92.52 ± 0.1	99.77 ± 0.1	99.69 ± 0.2	99.71 ± 0.1	99.88 ± 0.0	99.87 ± 0.0	99.84 ± 0.0
	Two-CNN	98.73 ± 0.2	98.71 ± 0.2	98.73 ± 0.2	98.72 ± 0.25	98.40 ± 0.17	98.45 ± 0.20	98.13 ± 0.43	98.01 ± 0.20	98.10 ± 0.20
	DPResNet	99.36 ± 0.3	99.44 ± 0.2	99.38 ± 0.3	99.85 ± 0.1	99.90 ± 0.08	99.88 ± 0.1	**100 ± 0.0**	**100 ± 0.0**	**100 ± 0.0**
	HybridSN	99.47 ± 0.1	99.40 ± 0.1	99.38 ± 0.1	99.86 ± 0.1	99.82 ± 0.0	99.71 ± 0.1	**100 ± 0.0**	**100 ± 0.0**	**100 ± 0.0**
	Proposed	**99.87 ± 0.1**	**99.79 ± 0.1**	**99.79 ± 0.1**	**99.97 ± 0.0**	**99.94 ± 0.0**	**99.96 ± 0.0**	**100 ± 0.0**	**100 ± 0.0**	**100 ± 0.0**

"Oats" and "Soybean-mintill" on IP data set differs greatly with 20 and 2455 labeled pixels, respectively. The subsequent processing became more challenging due to high categorical imbalance among different classes. S3EResBoF (99.49%) achieves 1.5% and 1.1% increase of mean OAs compared with SSRN (98.45%) and HybridSN (98.39%) when only 10% of its labeled data is used for training. S3EResBoF provides good OAs compared to SSRN because of the presence of SE blocks within the residual blocks, which helps to learn better spectral–spatial dependences within the spectral feature bands by eliminating spectral redundancies. The deeper network generally shows better performance as compared to 2-D-CNN, 3-D-CNN and M3D-CNN models, respectively. The robustness can be observed at the class-wise performance in IP of those classes having training samples lesser than 10, the S3EResBoF classified the test data by keeping mean accuracy more than 99%. The visualization of land-cover classification maps are compared using the best training model as shown in Fig. 3 and 4, for IP and UP, respectively. Fig. 3(a) and (b) corresponds to false color composite of IP, and UP image, data sets. Fig. 4(a) and (b) the groundtruth map and predicted classification maps using 2-D-CNN, 3-D-CNN, M3D-CNN, SSRN, HybridSN, and S3EResBoF, respectively. It is observed that SSRN, HybridSN, and S3EResBoF produce better quality visual classification as compared to the remaining models.

The capability of generating better feature representation of input always depends on the number of kernels used in the convolutional filter banks which can directly control the computation complexity of the S3EResBoF model. Both the spectral and spatial feature extraction using SERes blocks shown in Fig. 1 have the same number of convolutional kernels. We compare the different number of kernels from 8 to 64 (taken as the power of two) and the classification results are shown in Fig. 5(a). It has been observed from Fig. 5(a) that the proposed model achieves superior classification performance when the number of kernels is 64 and this result is achieved by 200 epochs of training for each data set.

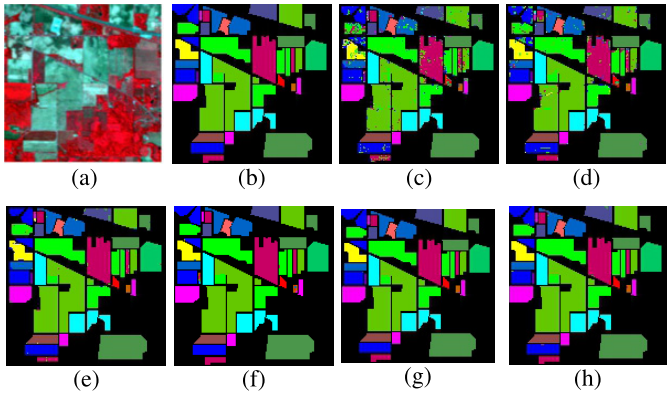


Fig. 3. Predicted class map for IP. (a) False color composite image. (b) Ground truth. (c)–(h) Resultant classification maps predicted using 2-D-CNN, 3-D-CNN, M3D-CNN, SSRN, HybridSN, and S3EResBoF, respectively.

In order to evaluate the effects of having the BoF layer in the proposed network, we consider different sizes of the codebook for three data sets IP, UP, and SA, respectively. The network is trained for 200 epochs and the classification accuracies using a different number of codewords over the test set are shown in Fig. 5(b). It can be observed that the presence of the BoF layer the S3EResBoF network is capable of achieving remarkable classification performance even if a smaller number of codewords are used. We have fixed the number of codewords or RBF neurons to 64 in the BoF layer to generate all the reported results. The authors thus claim that the proposed S3EResBoF architecture is lightweighted based on the observation shown in Fig. 5(c), which shows that the number of used trainable parameters for S3EResBoF is less compared to the other state-of-the-art methods while preserving the higher classification performance. It is seen that while S3EResBoF uses 246 800 weight parameters, 1 869 904 and 763 008 are used by the DPResNet and 3-D-LWNet [36] architecture, respectively. To demonstrate the robustness, we also depict the convergence

TABLE IV

IMPACT OF THE TRAINING DATA PERCENTAGE (I.E., 10%, 5%, AND 3%) AND THE SPATIAL WINDOW SIZE ($S \times S$) (I.E., 11× 11, 13× 13, AND 15 × 15) OVER THE PERFORMANCE OF THE PROPOSED MODEL ON IP, UP, AND SA DATA SETS, RESPECTIVELY

Training(%)	Window Size	Indian Pines Dataset			University of Pavia Dataset			Salinas Scene Dataset		
		OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
10%	15 x 15	**99.49 ± 0.2**	**99.42 ± 0.2**	**99.20 ± 0.3**	**99.77 ± 0.1**	**99.69 ± 0.1**	99.60 ± 0.1	**99.98 ± 0.0**	**99.98 ± 0.0**	**99.98 ± 0.0**
5%		98.65 ± 0.3	98.44 ± 0.3	98.72 ± 0.2	99.74 ± 0.2	99.50 ± 0.3	**99.64 ± 0.2**	99.86 ± 0.1	99.84 ± 0.1	99.83 ± 0.1
3%		95.90 ± 1.3	95.17 ± 1.0	95.91 ± 1.1	99.22 ± 0.2	98.71 ± 0.2	98.96 ± 0.2	99.80 ± 0.1	99.81 ± 0.1	99.78 ± 0.1
10%	13 x 13	**98.96 ± 0.4**	**98.34 ± 0.4**	**98.77 ± 0.4**	**99.84 ± 0.1**	**99.74 ± 0.1**	**99.79 ± 0.1**	**99.98 ± 0.0**	**99.96 ± 0.0**	**99.97 ± 0.0**
5%		97.78 ± 0.4	97.21 ± 0.4	97.19 ± 0.5	99.43 ± 0.1	99.12 ± 0.1	99.24 ± 0.1	99.94 ± 0.0	99.87 ± 0.0	99.90 ± 0.0
3%		94.23 ± ‘1.3	94.09 ± 1.4	93.86 ± 1.6	98.78 ± 0.2	98.26 ± 0.4	98.39 ± 0.4	99.76 ± 0.1	99.78 ± 0.1	99.81 ± 0.1
10%	11 x 11	**97.76 ± 0.4**	**97.45 ± 0.4**	**96.28 ± 0.4**	**99.77 ± 0.2**	**99.67 ± 0.2**	**99.69 ± 0.2**	**99.97 ± 0.0**	**99.97 ± 0.0**	**99.96 ± 0.0**
5%		96.14 ± 1.3	96.17 ± 1.2	95.45 ± 1.6	99.67 ± 0.2	99.62 ± 0.2	99.56 ± 0.2	99.94 ± 0.0	99.92 ± 0.0	99.93 ± 0.0
3%		93.48 ± 2.2	93.11 ± 2.1	91.27 ± 2.3	99.58 ± 0.1	99.36 ± 0.1	99.44 ± 0.1	99.14 ± 0.2	99.56 ± 0.1	99.04 ± 0.2

TABLE V

OAS ACHIEVED BY THE SSRN, DPRESNET, CAPSNET, AND THE PROPOSED TWO APPROACHES S3ERESNET, AND S3ERESBOF, RESPECTIVELY, FOR DIFFERENT INPUT SPATIAL WINDOW SIZES

Spatial Size	Indian Pines (IP)					University of Pavia (UP)				
	SSRN [44]	DPResNet [49]	CapsNet [51]	S3EResNet	S3EResBoF	SSRN [44]	DPResNet [49]	CapsNet [51]	S3EResNet	S3EResBoF
5×5	92.83 ± 0.66	97.17 ± 0.31	97.79 ± 0.40	98.31 ± 0.15	98.85 ± 0.13	98.72 ± 0.17	99.52 ± 0.10	99.13 ± 0.08	99.68 ± 0.11	99.83 ± 0.10
7×7	97.81 ± 0.34	97.86 ± 0.12	99.30 ± 0.11	98.44 ± 0.11	99.11 ± 0.11	99.54 ± 0.11	99.72 ± 0.10	99.75 ± 0.03	99.75 ± 0.06	99.88 ± 0.05
9×9	98.68 ± 0.29	98.71 ± 0.08	99.67 ± 0.06	98.93 ± 0.06	99.71 ± 0.08	99.73 ± 0.15	99.79 ± 0.03	99.89 ± 0.02	99.81 ± 0.05	99.90 ± 0.03
11×11	**98.70 ± 0.21**	**99.03 ± 0.10**	**99.74 ± 0.09**	**99.22 ± 0.08**	**99.87 ± 0.10**	**99.79 ± 0.08**	**99.84 ± 0.04**	**99.93 ± 0.02**	**99.90 ± 0.02**	**99.93 ± 0.03**

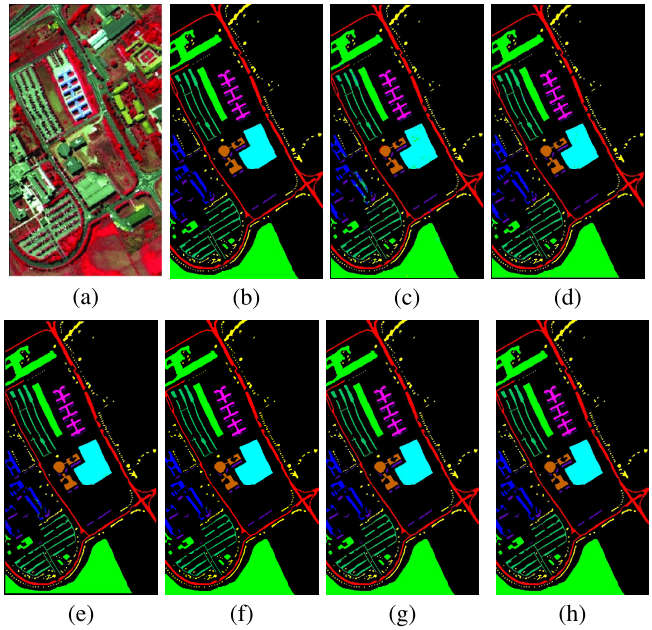


Fig. 4. Predicted map for Pavia University. (a) False color composite image. (b) Ground truth. (c)–(h) Output classification maps predicted using 2-D-CNN, 3-D-CNN, M3D-CNN, SSRN, and HybridSN, and S3ERESBOF, respectively.

of training and validation accuracy using the S3EResNet and S3ERESBOF as illustrated in Fig. 6(a)–(d) over the IP, UP and SA data sets with 200 and 50 iterations, respectively. Here S3EResNet denotes the same spectral–spatial squeeze-and-excitation residual network but without the BoF learning layer. In this experiment, the IP data set is divided into three parts, i.e., 15% training, 5% validation, and 80% unlabeled data are used for testing. It is clear that the S3ERESBOF network is able to converge within 50 epochs, whereas S3EResNet network converges slowly around in 200 epochs of training.

To evaluate feature generalization ability and the influence of spatial window size of S3ERESBOF, the conducted experiments are repeated using varied training samples of 3%, 5%, and 10%, respectively, based on three different spatial window sizes, i.e., 15 × 15, 13 × 13, and 11 × 11, respectively. The classification results over IP, UP, and SA data sets are summarized in Table IV. It is noted that the performance reduces significantly over the IP data set when 3% training samples are chosen as compared to 5% and 10%. Whereas, the results over UP and SA data sets remain reasonable even with lesser training samples such as 3%. Moreover, the improvements for a large percentages of training samples are not clear since S3ERESBOF is capable of producing the best classification OA results higher than 99% for all three HSI data sets using different input spatial sizes.

The last series of experiments are conducted to compare the proposed S3ERESBOF and two state-of-the-art deep spectral–spatial network for the HSI classification methods SSRN [38], DPResNet [43], CapsNet [45], and S3EResNet, respectively. Table V shows the mean classification accuracy with varying spatial window sizes where the training configuration is the same as given in SSRN [38]. The first column shows the considered spatial size and the next two columns arrange for the data and show the mean classification of OAs for IP and UP data sets, respectively. In addition, it is also observed that the standard deviation of S3ERESBOF is considerably lower compared to that in SSRN, DPResNet, and S3EResNet, respectively. The proposed architecture achieves improvements over DPResNet in terms of average OAs by +1.68, +1.25, +1.00, and +0.84 for IP and average OAs by +0.31, +0.16, +0.11, and +0.09 for UP using the spatial window of sizes, 5 × 5, 7 × 7, 9 × 9, and 11 × 11, respectively. This is because the spectral–spatial SE residual block together with BoF learning can enhance the generalizability of the network and are able to efficiently reduce the uncertainty when applied to HSI data. The proposed approach is able to achieve a significant performance improvement, i.e., 98.86% when the spatial window size is 9 × 9 for IP. The S3ERESBOF

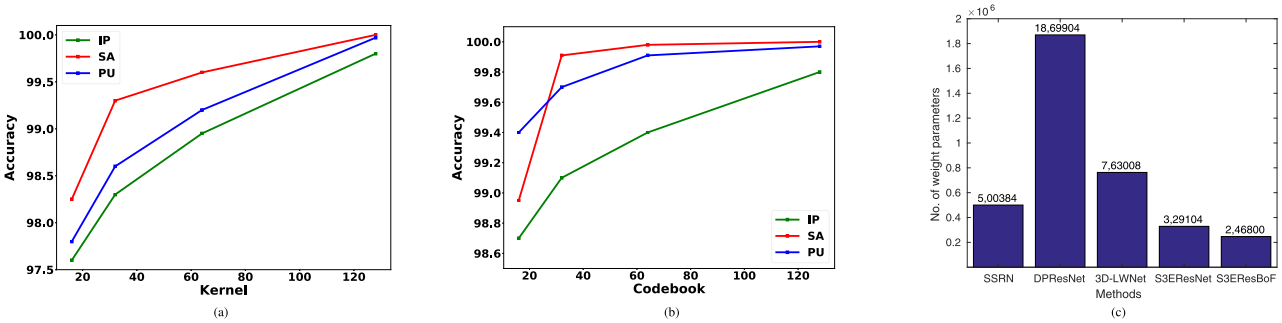


Fig. 5. (a) OAs(%) of S3EResBoF using different numbers of kernel for IP, UP and SA, respectively. (b) Visualization of test accuracy of BoF for different number of codewords. (c) Number of trainable weight parameters for the state-of-the-art methods.

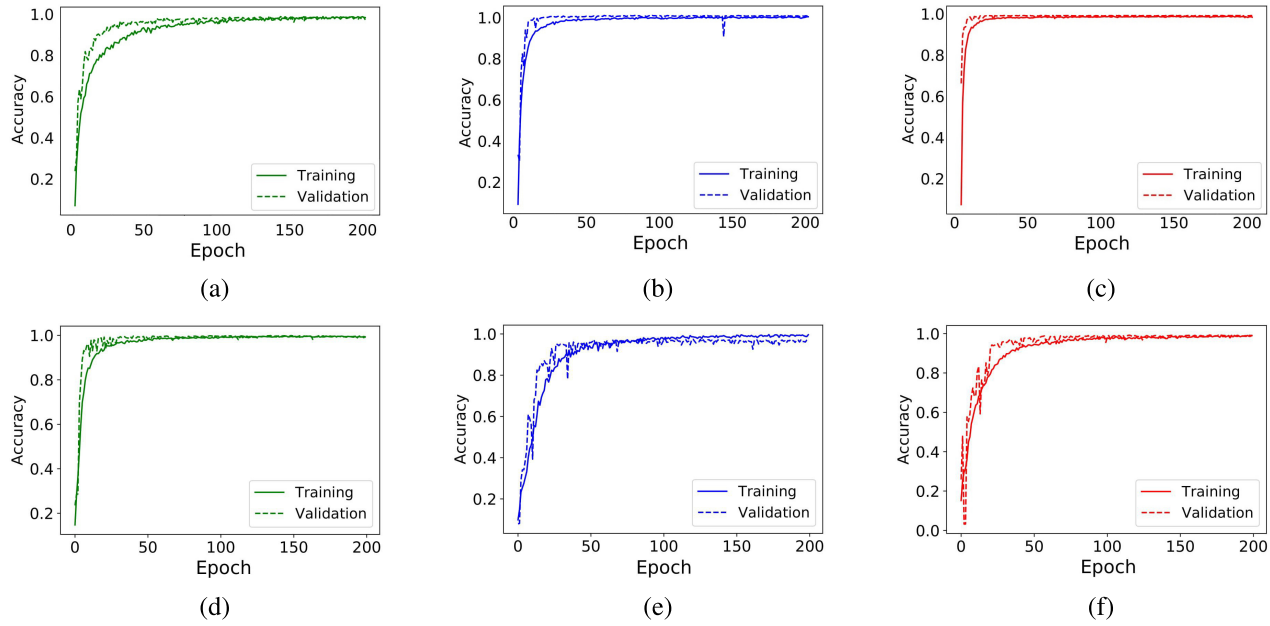


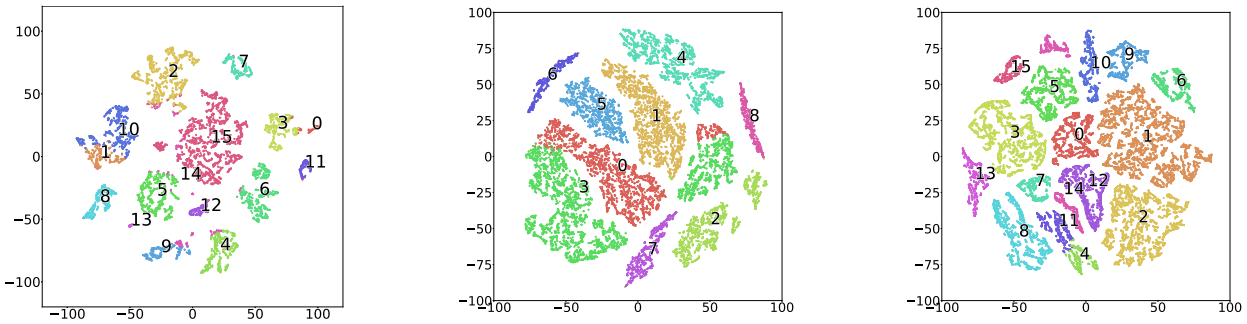
Fig. 6. Convergence of accuracy versus epochs using the S3EResNet and S3EResBoF shown in (a)–(c) and (d)–(f) over IP, UP, and SA data sets, respectively.

TABLE VI

OAs, AA, and Kappa Achieved by the Proposed S3EResBoF and ROhsi [30] Model for Different Occlusion Effects and Spatial Input Window of Size (i.e., 15 × 15 and 23 × 23) Over the IP, UP, and SA Data Sets, Respectively

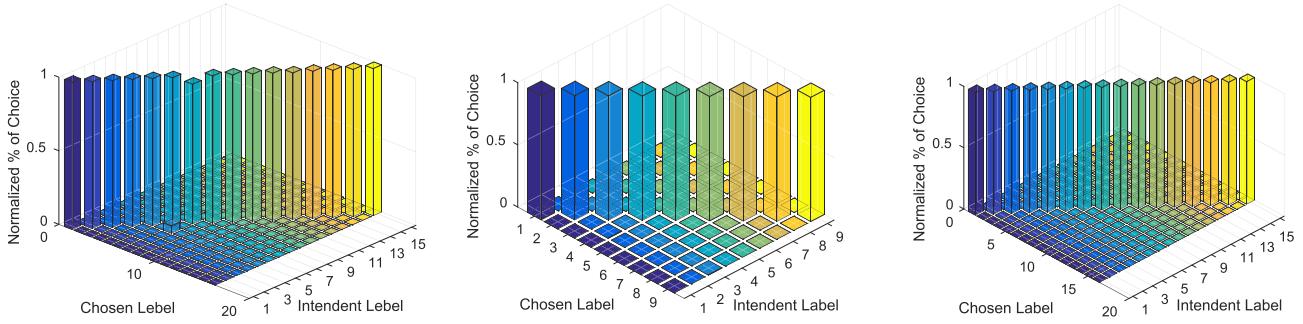
Methods	Occlusion(%)	Window Size	Indian Pines Dataset			University of Pavia Dataset			Salinas Scene Dataset		
			OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
ROhsi [35]	25%	15 × 15	97.53 ± 0.2	97.61 ± 0.2	97.55 ± 0.3	98.71 ± 0.1	98.73 ± 0.1	98.74 ± 0.1	99.13 ± 0.1	99.22 ± 0.1	99.17 ± 0.1
	50%		97.73 ± 0.5	97.81 ± 0.4	97.79 ± 0.5	98.98 ± 0.3	98.96 ± 0.3	99.02 ± 0.3	99.37 ± 0.2	99.41 ± 0.1	99.47 ± 0.1
	25%	23 × 23	98.15 ± 0.3	98.12 ± 0.4	98.21 ± 0.3	99.73 ± 0.1	99.69 ± 0.1	99.63 ± 0.1	99.84 ± 0.0	99.83 ± 0.0	99.88 ± 0.0
	50%		**98.47 ± 0.2**	**98.42 ± 0.2**	**98.39 ± 0.3**	**99.86 ± 0.0**	**99.86 ± 0.0**	**99.87 ± 0.0**	**99.91 ± 0.0**	**99.92 ± 0.0**	**99.91 ± 0.0**
S3EResBoF	25%	15 × 15	98.53 ± 0.2	98.54 ± 0.2	98.58 ± 0.2	99.19 ± 0.1	99.16 ± 0.1	99.24 ± 0.1	99.29 ± 0.1	99.37 ± 0.1	99.31 ± 0.1
	50%		99.06 ± 0.2	98.72 ± 0.2	98.88 ± 0.2	99.60 ± 0.1	99.54 ± 0.1	99.58 ± 0.1	99.61 ± 0.2	99.54 ± 0.1	99.59 ± 0.1
	25%	23 × 23	99.18 ± 0.1	99.24 ± 0.1	99.17 ± 0.1	99.81 ± 0.1	99.79 ± 0.1	99.85 ± 0.1	99.90 ± 0.0	99.91 ± 0.0	99.90 ± 0.0
	50%		**99.31 ± 0.1**	**99.37 ± 0.1**	**99.42 ± 0.1**	**99.90 ± 0.0**	**99.89 ± 0.0**	**99.91 ± 0.0**	**99.94 ± 0.0**	**99.94 ± 0.0**	**99.94 ± 0.0**

architecture can extract high-dimensional abstract spectral–spatial features. In order to increase feature generalization ability of the proposed network and to mitigate the overfitting problems, the experiments are conducted by erasing areas of the input HSI cubes randomly while training. Table VI shows the classification accuracies in terms of OAs, AAs, and Kappa compared with the ROhsi method proposed by Haut *et al.* [30]. The proposed network achieves comparable performance as compared to the ROhsi and the feature generalization ability of the proposed S3EResBoF is higher than ROhsi, since it utilizes the benefits of the spectral–spatial block in the form of ResNet. In the experiments, the chosen window sizes are 15 × 15 and 23 × 23 with varying occlusion rates set to 25%, and 50%, respectively, over IP, UP, and SA, respectively. In order to visualize the discriminative power, the learned high-dimensional features are reduced to 2-D via t-SNE [55]. The plotted 2-D features are shown in Fig. 7(a)–(c) for IP, UP, and SA, respectively.



(a) (b) (c)

Fig. 7. 2-D spectral–spatial feature visualization of the S3EResBoF via t-SNE where samples are represented through points and classes are shown in different colors for (a) IP (b) UP, and (c) SA, respectively. (Best can be viewed on a color monitor.)



(a) (b) (c)

Fig. 8. Obtained confusion matrix using S3EResBoF over (a) IP, (b) UP, and (c) SA data sets, respectively.

TABLE VII

OAs, AA, and Kappa Achieved by the Proposed Model for the Disjoint Training and Testing Samples With Input Spatial Window of Size (7 × 7) Over IP, UP, and SA Data Sets, Respectively

Datasets	Training(%)	Window Size	Evaluation Matrix		
			OA	AA	Kappa
IP	20%	7 x 7	91.48 ± 0.2	91.73 ± 0.1	92.01 ± 0.2
	30%		**92.81 ± 0.2**	**92.93 ± 0.2**	**93.13 ± 0.2**
UP	20%		93.55 ± 0.3	94.11 ± 0.2	93.46 ± 0.3
	30%		**94.85 ± 0.1**	**95.71 ± 0.2**	**95.56 ± 0.1**
SA	20%		92.99 ± 0.2	93.07 ± 0.2	93.16 ± 0.2
	30%		**94.79 ± 0.1**	**94.63 ± 0.1**	**94.61 ± 0.1**

It can be clearly observed that the learned abstract spectral–spatial features of different pixels from the same classes are assembled into clusters but with more number of epochs in training makes it becomes much easier to separate each cluster. S3EResBoF improves class separation by assigning different regions of a class into different clusters through high intercluster overlap distance. Moreover, the confusion matrices obtained using the S3EResBoF method over IP, UP, and SA data sets are shown in Fig. 8, respectively. It can be seen that mostly diagonal values along are present in the confusion matrices.

D. Performance on Disjoint Training-Testing Samples

In order to show the effectiveness of spectral–spatial features and the spatial influences over the local neighborhood, we perform different experiments by creating nonoverlapped samples for both the training-testing [56]. Since IP data sets are highly imbalanced and some of its associate classes contain inadequate samples, i.e., for “Oats" it is 20 if created using overlapping fashion otherwise even less for disjoint scenario. So, we discard that the specific class from IP while performing the experiments with existing network settings for the remaining classes. The rest of the data sets, i.e., UP and SA, respectively, are kept as it is without any modification. Table VII shows the classification performance measured in terms of OAs, AAs, and Kappa with a fixed window size of 7 × 7 and varying training samples 20% and 30%, over IP, UP, and SA, respectively, when both the training-testing samples are completely nonoverlapped. It can be observed from Table VII, that the accuracy for the proposed method degrades to some extent as compared to the experiments shown in Table III, since the neighborhood information is preserved while creating training-testing samples. However, the classification ability of the proposed method, as well as other existing methods are not able to achieve sound performance when the spatial input patch size of training-testing samples is increased and hence the experiment became challenging. This can be attributed to the fact that either the methods are spatially less influenced within the local neighborhood of input patch over the training-testing samples or even do not have adequate number of available training samples in an individual class.

V. Conclusion

This article presents a supervised and efficient S3EResBoF learning end-to-end trainable framework for HSI classification.

S3EResBoF contains spectral and spatial residual learning blocks to accelerate the classification accuracy. Apart from this, in order to improve the nature of feature representation, every residual block is followed by an SE operation to perform the feature weight recalibration which is produced by ResNet. The above process assigns weight to the feature maps closer to $sigmoid$ one if it strongly participates in the classification task or else suppresses the ineffective feature maps by assigning their weights closer to $sigmoid$ zero. This helps us to eliminate the number of feature maps while the BoF learning layer reduces the number of trainable weight parameters of the fully connected layer to some extent where most of the network parameters are usually used and are able to generate feature invariant to the small distribution shifts. Hence, the proposed S3EResBoF framework is a light-weighted deep network architecture having lesser number of parameters and requires lower computation resources, resulting in superior OAs performance even if the number of available labeled training samples is less. It is investigated that the proposed method is also effective when the training and testing sample patches are formed in completely nonoverlapping fashion, achieving sound performance. However, the designed S3EResBoF framework has its own potential to deal and consistently generate formidable classification performance with a lesser number of available training samples. The impact of different window sizes and the t-SNE visualization also prove the feature generalization ability of the proposed S3EResBoF framework. The proposed method gains considerable classification performance under the random erasing data augmentation scenario with varying occlusion rates. Moreover, the BoF learning layer can be combined with various existing networks and can experience the HSI classification accuracy. In future, the layer-wise feature representation can be extracted by combining the BoF learning with multiple layers and it would also be exotic to explore an optimized CNN structure through some neural search algorithms.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the three anonymous reviewers for their outstanding comments and suggestions, which greatly helped them to improve the technical quality and presentation of this article.

REFERENCES

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.

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

[3] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observat., Remote Sens.*, vol. 9, no. 9, pp. 4117–4128, Sep. 2016.

[4] V. Singhal and A. Majumdar, "Row-sparse discriminative deep dictionary learning for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5019–5028, Dec. 2018.

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

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

[7] X. Kang, C. Li, S. Li, and H. Lin, "Classification of hyperspectral images by Gabor filtering based deep network," *IEEE J. Sel. Top. Appl. Earth Observat., Remote Sens.*, vol. 11, no. 4, pp. 1166–1178, Apr. 2018.

[8] 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.

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

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

[11] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

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

[13] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral Image classification based on three-dimensional scattering wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2467–2480, May 2015.

[14] S. Jia, L. Shen, and Q. Li, "Gabor feature-based collaborative representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1118–1129, Feb. 2015.

[15] J. Li, Q. Du, Y. Li, and W. Li, "Hyperspectral image classification with imbalanced data based on orthogonal complement subspace projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3838–3851, Jul. 2018.

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

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

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

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

[20] 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.

[21] 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.

[22] 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.

[23] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.

[24] S. Hao, W. Wang, Y. Ye, E. Li, and L. Bruzzone, "A deep network architecture for super-resolution-aided hyperspectral image classification with classwise loss," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4650–4663, Aug. 2018.

[25] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[26] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observat., Remote Sens.*, vol. 11, no. 11, pp. 4048–4062, Nov. 2018.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[32] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[33] 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.

[34] A. B. Hamida, A. Benoit, P. Lambert, and C. B. 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.

[35] 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.

[36] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[37] 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.*, to be published, doi: 10.1109/LGRS.2019.2918719.

[38] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[39] 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.

[40] M. Liang, L. Jiao, S. Yang, F. Liu, B. Hou, and H. Chen, "Deep multiscale spectral–spatial feature fusion for hyperspectral images classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2911–2924, Aug. 2018.

[41] 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.

[42] 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.

[43] 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.

[44] J. Feng *et al.*, "CNN-based multilayer spatial–spectral feature fusion and sample augmentation with local and nonlocal constraints for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observat., Remote Sens.*, vol. 12, no. 4, pp. 1299–1313, Apr. 2019.

[45] 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.

[46] 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.

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

[48] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1705–1715, Jun. 2019.

[49] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.

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

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

[52] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh, and B. B. Chaudhuri, "diffGrad: An optimization method for convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2955777.

[53] 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.

[54] Y. Zhang, G. Cao, X. Li, and B. Wang, "Cascaded random forest for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observat., Remote Sens.*, vol. 11, no. 4, pp. 1082–1094, Apr. 2018.

[55] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[56] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao, "On the sampling strategy for evaluation of spectral–spatial methods in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 862–880, Feb. 2017.

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 currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of Calcutta, Kolkata.

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

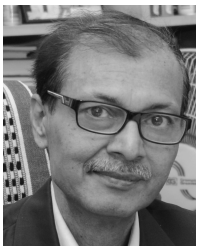
Subhrasankar Chatterjee received the bachelor's degree in computer science and engineering from the Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal, India, in 2019.

He is currently a Research Assistant with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur, India. His research interests include deep learning, remote sensing, and image classification.

Siddhartha Bhattacharyya (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Jadavpur University, Kolkata, India, in 2008.

He is currently serving as a Professor with the Department of Computer Science and Engineering, Christ University, Bengaluru, India. Prior to this, he was the Principal of the RCC Institute of Information Technology, Kolkata. He also served as a Senior Research Scientist with the Faculty of Electrical Engineering and Computer Science, VŠB Technical University of Ostrava, Ostrava, Czech Republic, from 2018 to 2019. He has published more than 250 research articles in international journals and conference proceedings. He holds three patents. His research interests include hybrid intelligence, pattern recognition, multimedia data processing, quantum computing, and social networks.

Dr. Bhattacharyya is a Life Fellow of the Optical Society of India (OSI), India, and a fellow of the Institute of Electronics and Telecommunication Engineers (IETE), India, and the Institution of Engineers (IEI), India. He is an Associate Editor of the IEEE ACCESS, *Evolutionary Intelligence*, *Applied Soft Computing*, and the *IET Quantum Communication*.



Bidyut Baran Chaudhuri (Life Fellow, IEEE) received the Ph.D. degree from the IIT Kanpur, Kanpur, India, in 1980.

He was a Leverhulme Post-Doctoral Fellow with the Queen's University, Belfast, U.K., from 1981 to 1982. He joined the Indian Statistical Institute, Kolkata, India, in 1978, where he was an Indian National Academy of Engineering (INAE) Distinguished Professor and a J. C. Bose Fellow with the Computer Vision and Pattern Recognition Unit. He is currently with the Techno India University, Kolkata, as a Pro-Vice Chancellor (Academic). He pioneered the first workable optical character recognition (OCR) system for printed Indian scripts Bangla, Assamese, and Devnagari. He also developed computerized *Bharati Braille system* with speech synthesizer and has done statistical analysis of Indian language. He has published about 425 research articles in international journals and conference proceedings. His research interests include pattern recognition, image processing, computer vision, natural language processing (NLP), signal processing, digital document processing, and deep learning. Also, he has authored/edited seven books in these fields.

Dr. Chaudhuri is a fellow of INSA, NASI, INAE, IAPR, and The World Academy of Sciences (TWAS). He received the Leverhulme Fellowship Award, the Sir J. C. Bose Memorial Award, the M. N. Saha Memorial Award, the Homi Bhabha Fellowship, the Dr. Vikram Sarabhai Research Award, the C. Achuta Menon Award, the Homi Bhabha Award: Applied Sciences, the Ram Lal Wadhwa Gold Medal, the Jawaharlal Nehru Fellowship, the J. C. Bose Fellowship, and the Om Prakash Bhasin Award. He has been the general chair and technical co-chair at various international conferences. He is an associate editor of three international journals.

Jan Platoš (Member, IEEE) received the Ph.D. degree in computer science from the VŠB-Technical University of Ostrava, Ostrava, Czech Republic, in 2006.

He became an Associate Professor in computer science in 2014. Since 2017, he has been the Head of the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB-Techincal University of Ostrava. He has coauthored more than 180 scientific articles published in proceedings and journals, and his citation report consists of 302 citations and H-index of 9 on the Web of Science, 685 citations and H-index of 13 on Scopus, and 965 citations and H-index of 15 on Google Scholar. His primary fields of interest are text processing, data compression, bioinspired algorithms, information retrieval, data mining, data structures, and data prediction.