BASS Net: Band-Adaptive Spectral-Spatial Feature Learning Neural Network for Hyperspectral Image Classification

Anirban Santara, Kaustubh Mani, Pranoot Hatwar, Ankit Singh, Ankur Garg, Kirti Padia, and Pabitra Mitra

***Abstract*— Deep learning based land cover classification algorithms have recently been proposed in the literature. In hyperspectral images (HSIs), they face the challenges of large dimensionality, spatial variability of spectral signatures, and scarcity of labeled data. In this paper, we propose an end-to-end deep learning architecture that extracts band specific spectral-spatial features and performs land cover classification. The architecture has fewer independent connection weights and thus requires fewer training samples. The method is found to outperform the highest reported accuracies on popular HSI data sets.**

***Index Terms*— Convolutional neural network (CNN), deep learning, feature extraction, hyperspectral imagery, landcover classification, pattern classification.**

I. Introduction

Hyperspectral imaging [1], [2] measures reflected radiation from a surface at a series of narrow, contiguous frequency bands. It differs from multispectral imaging, which senses a few wide, separated frequency bands. Hyperspectral imaging produces 3-D (x, y, λ) data volumes, where x and y represent spatial dimensions and λ represents spectral dimension. Such a detailed spectrum provides richer spectral information about the identity and characteristics of the material at the location of a pixel than is available from a multispectral image.

This paper studies land-cover classification in hyperspectral images (HSIs) [3] where the task is to predict the type of land-cover present in the location of each pixel. There are several challenges associated with the predictive analysis of hyperspectral data, the most critical of which are: 1) curse of dimensionality resulting from a large number of spectral dimensions; 2) scarcity of labeled training examples; and 3) large spatial variability of spectral signature [4]. Challenges 1) and 2) lead to the Hughes phenomenon [5], which means that for a fixed number of training samples, the predictive power reduces with increasing dimensionality of the feature space.

Several approaches have been followed in the literature for HSI classification. The simplest of them are based on the k-nearest neighbors (k-NNs). In these methods, given a test sample, Euclidean distance in the input space or a transformed space is used to find the k nearest training examples and a class is assigned on the basis of them. In [6] and [7], some modified versions of the k-NN algorithm have been proposed for HSI classification. A major drawback of these approaches is poor generalization in the absence of adequate training data.

Support vector machine (SVM) classifier is a maximum margin linear classifier [8]. Melgani and Lorenzo [9] introduced SVM for HSI classification. The SVM-based methods, in general, follow a two-step approach. First, dimensionality reduction in order to address the problems of high spectral dimensionality and scarcity of labeled training examples. Some of the methods followed for dimensionality reduction are subspace projection [10], random feature selection [11], and kernel local Fisher discriminant analysis [12]. Second, classification in the reduced dimensional space using SVM [9], [10], [13]. The dimensionality reduction step in these algorithms is not data-driven as a result of which, the extracted features might be suboptimal for classification. This paper focuses on fully data-driven feature learning for classification.

Li *et al.* [13] propose local Fisher's discriminant analysis for dimensionality reduction and the Gaussian mixture model for classification. Mianji and Zhang [14] propose Gaussian non-linear discriminant analysis for dimensionality reduction and relevance vector machine for classification. Samat *et al.* [15] introduce extreme learning machine (ELM) for HSI classification. ELM [16] is a two layer artificial neural network in which the input to hidden weights is randomly chosen and the hidden to output weights is learned by minimizing a least squares objective function. In [17], local binary pattern (LBP) is used to extract texture-based local descriptors, which are combined with global descriptors such as Gabor and spectral features, and fed into an ELM for classification. However, the minimal

Manuscript received December 1, 2016; revised February 16, 2017 and March 25, 2017; accepted April 21, 2017. Date of publication June 5, 2017; date of current version August 25, 2017. This work was supported by the Space Applications Centre, Indian Space Research Organization through the Project titled Deep Learning for Automated Feature Discovery in Hyperspectral Images (LDH). The work of A. Santara was supported by Google India under the Google India Ph.D. Fellowship Award. *(Anirban Santara and Kaustubh Mani contributed equally to this work.) (Corresponding author: Anirban Santara.)*

A. Santara and P. Mitra are with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: anirban_santara@iitkgp.ac.in; pabitra@cse.iitkgp.ernet.in).

K. Mani is with the Department of Geology and Geophysics, IIT Kharagpur, Kharagpur 721302, India (e-mail: kaustubh3095@gmail.com).

P. Hatwar and A. Singh are with the Department of Electrical Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: pphatwar1995@gmail.com; ankit.vngr@gmail.com).

A. Garg and K. Padia are with the Space Applications Centre, Indian Space Research Organization, Ahmedabad 380015, India (e-mail: agarg@sac.isro.gov.in; kirtipadia@sac.isro.gov.in).

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

Digital Object Identifier 10.1109/TGRS.2017.2705073

architecture of ELM has limited discriminative power. Also the fact that the weights from the input to the hidden layer are not trainable limits the extraction of expressive features. In this paper, we have designed a neural network that has many hidden layers and is fully trainable by gradient descent. Lu *et al.* [18] proposed a set-to-set distance-based method for HSI classification.

Efficient modeling of spectral-spatial context is necessary to address the problem of spatial variability of spectral signatures. This has been emphasized in the context of dimensionality reduction [19] and salient band selection tasks [20]. Recently, deep learning neural networks [21], [22] capable of efficient context modeling have been employed for land cover classification in HSI [23]. These deep learning algorithms fall into two broad categories. The first category [24]–[28] follows a two-step procedure. First, dimensionality reduction and spectral-spatial feature learning using autoencoder. Autoencoder [29] is an artificial neural network architecture that learns to reconstruct the input vector at the output with minimum distortion after passing through a bottleneck. The vector of activations in the bottleneck is a reduced dimensional representation of the input vector that often encodes useful semantic information. Second, classification using multiclass logistic regression. The main drawback of these approaches is the absence of task-specific feature learning. This is mitigated in our proposed method by using end-to-end supervised learning to tune the features to the specific task of landcover classification.

The second category of methods uses convolutional neural networks (CNNs) [22], [30] for feature learning and classification in an end-to-end fashion. CNN uses extensive parameter-sharing to tackle the curse of dimensionality. Hu *et al.* [31] introduced CNN for HSI classification. The proposed architecture is designed to learn abstract spectral signatures in a hierarchical fashion but does not take into account spatial context. In [32], compressed spectral features from a local discriminant embedding method are concatenated with spatial features from a CNN and fed into a multiclass classifier. Yu *et al.* [33] and Chen *et al.* [34] propose end-to-end CNN architectures for spectral-spatial feature learning and classification. In [35], the idea of classifying pixel-pair features (PPFs) using CNN is introduced to compensate for data scarcity. Also, a voting strategy is proposed for test time to provide robustness in heterogeneous regions. However, none of these methods simultaneously address the three principal challenges of HSI classification—the curse of dimensionality, scarcity of labeled examples, and spatial variability of spectral signature. Ours is a novel attempt at designing a single end-to-end deep learning neural network architecture that simultaneously addresses these challenges.

In this paper, we present a deep neural network architecture that learns band-specific spectral-spatial features and gives a state-of-the-art performance without any kind of data set augmentation. The architecture consists of three cascaded blocks. Block 1 takes a $p \times p \times N_c$ input volume ($N_c =$ number of spectral channels) and performs a preliminary feature transformation on the spectral axis. It splits the spectral channels into bands and feeds to Block 2 where parallel neural networks are used to extract low and midlevel spectral-spatial features. The outputs of the parallel networks are fused by concatenation and fed into Block 3, which summarizes them to form a high-level representation of the input. This is eventually classified by logistic regression. Extensive use of convolutional layers and weight sharing among the parallel networks of Block 2 keeps the parameter budget and computational complexity low. Band-specific representation learning and fusion via concatenation in Block 2 make the network discriminative toward spectral locality of low and mid-level features. Experiments on benchmark HSI classification data sets show that the proposed network converges faster and gives superior classification performance than other deep learning-based methods in the literature. Our source code is publicly available on GitHub.[1]

The idea of band specific spectral-spatial feature learning is based on the fact that the data collected by HSI sensors have the unique characteristic of grouped features. Some frequency subbands are more discriminative about certain material characteristics than others. This has motivated prior work on supervised [36], [37] and unsupervised [38] HSI classification. This paper makes a novel attempt at leveraging this characteristic in a deep neural network architecture for improved feature learning at a low parameter budget.

The contributions of this paper can be summarized as follows.

1) A novel end-to-end deep neural network architecture has been proposed that shows state-of-the-art performance on benchmark HSI classification data sets. The design is aimed at efficient band-specific feature learning keeping the number of parameters low.
2) Considerable improvement in training time is observed when compared with other popular deep learning architectures.

Section II gives a detailed description of the proposed architecture along with the design methodology followed. Experimental results are presented in Section III. Comparison with existing methods is also reported. Section IV concludes this paper with a summary of the proposed method and scope of future work.

II. Proposed Framework

The BASS Net architecture, shown in Fig. 1, combines spectral and spatial information processing in a systematic way with a focus on efficient use of parameters. The input to the network is a pixel $\mathbf{X_i}$ from the image with its $p \times p$ neighborhood (for spatial context) in the form of a $p \times p \times N_c$ volume, where N_c is the number of channels in the input image. The output is the predicted class label $\hat{y}_i$ for $\mathbf{X_i}$. The entire network is differentiable end-to-end and can be trained by backpropagation [39].

A. Overview of Architecture

The architecture is organized as three cascaded blocks.

1) Block 1 (Spectral Feature Selection and Band Partitioning): Block 1 takes the input $p \times p \times N_c$ volume $\mathbf{X_i}$ and

[1] https://github.com/kaustubh0mani/BASS-Net

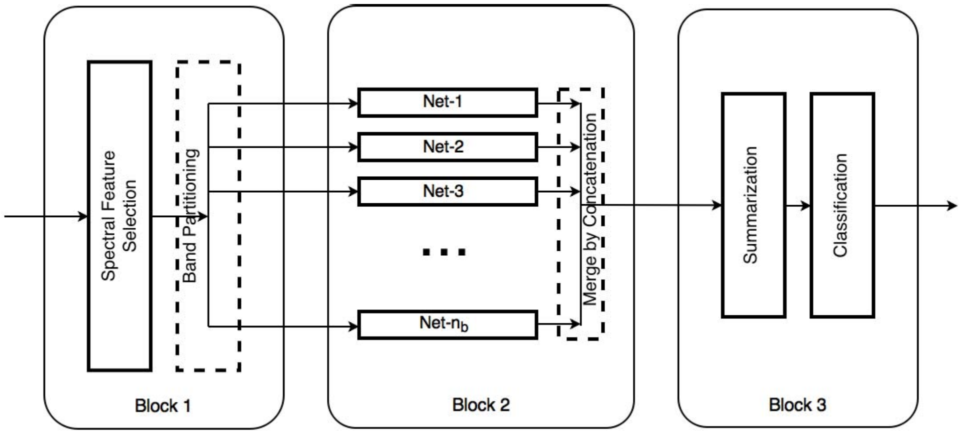


Fig. 1. Block diagram of the BASS Net architecture.

TABLE I

COMPARISON OF DIFFERENT ARCHITECTURAL DESIGN CHOICES IN TERMS OF ACCURACY ON THE VALIDATION SPLIT OF THE INDIAN PINES DATA SET

<table>
<tr><td></td><td colspan="2">Configuration 1</td><td colspan="2">Configuration 2</td><td colspan="2">Configuration 3</td><td colspan="2">Configuration 4</td></tr>
<tr><td colspan="9">Input volume: $3 \times 3 \times 220$</td></tr>
<tr><td>Block 1</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">$*conv_{xy} - 1, 220$</td><td colspan="2">$conv_{xy} - 1, 220$</td></tr>
<tr><td colspan="9">Split into n_b bands along the λ-axis</td></tr>
<tr><td rowspan="4">Block 2</td><td>$fc - 150$</td><td rowspan="4">$n_b = 10$</td><td>$*conv_{\lambda} - 3, 20$</td><td rowspan="4">$n_b = 10$</td><td>$conv_{\lambda} - 3, 20$</td><td rowspan="4">$n_b = 10$</td><td>$conv_{\lambda} - 3, 20$</td><td rowspan="4">$n_b = 10$</td></tr>
<tr><td>$fc - 100$</td><td>$*conv_{\lambda} - 3, 20$</td><td>$conv_{\lambda} - 3, 20$</td><td>$*conv_{\lambda} - 3, 20$</td></tr>
<tr><td rowspan="2"></td><td>$fc - 100$</td><td>$fc - 100$</td><td>$*conv_{\lambda} - 3, 10$</td></tr>
<tr><td></td><td></td><td>$conv_{\lambda} - 5, 5$</td></tr>
<tr><td colspan="9">Concatenate the outputs of the parallel networks</td></tr>
<tr><td rowspan="3">Block 3</td><td colspan="2">$fc - 500$</td><td colspan="2">$fc - 500$</td><td colspan="2">$fc - 500$</td><td colspan="2">$fc - 100$</td></tr>
<tr><td colspan="2">$fc - 100$</td><td colspan="2">$fc - 100$</td><td colspan="2">$fc - 100$</td><td colspan="2">$fc - 9$</td></tr>
<tr><td colspan="2">$fc - 9$</td><td colspan="2">$fc - 9$</td><td colspan="2">$fc - 9$</td><td colspan="2"></td></tr>
<tr><td colspan="9">9-way softmax layer for classification</td></tr>
<tr><td>Validation Accuracy
$PS = OFF$</td><td colspan="2">93%</td><td colspan="2">95.5%</td><td colspan="2">97.5%</td><td colspan="2">98%</td></tr>
<tr><td>Validation Accuracy
$PS = ON$</td><td colspan="2">94%</td><td colspan="2">97.5%</td><td colspan="2">98.5%</td><td colspan="2">99.5%</td></tr>
</table>

performs the operation described by

$$\{B_1, B_2, \ldots, B_{n_b}\} = \Psi(\Phi(\mathbf{X_i}), n_b). \tag{1}$$

$\Psi(\cdot, \cdot)$ is a function that takes as input an HSI volume $\mathbf{X}$ with N spectral channels and an integer n_b. It splits $\mathbf{X}$ into n_b nonoverlapping adjacent bands $\{B_i\}_{i=1}^{n_b}$ of equal bandwidth b, where $b = (N/n_b)$

$$\{B_1, B_2, \ldots, B_{n_b}\} = \Psi(\mathbf{X}, n_b). \tag{2}$$

$\Phi(\cdot)$ is a function that applies a feature selection algorithm along the spectral dimension of a $p \times p \times N_{\text{in}}$ HSI volume $\mathbf{X}$ and produces another $p \times p \times N_{\text{out}}$ output volume $\mathbf{Y}$. Out of the many different possibilities, for this function, we have explored the identity function $I(\cdot)$ and 1×1 spatial convolution in this paper. Let $\mathbf{X} = [X^{(i)}]_{i=1}^{N_{\text{in}}}$ and $\mathbf{Y} = [Y^{(j)}]_{j=1}^{N_{\text{out}}}$, i.e., let X^i and Y^j be the input and output channels along the spectral dimension. If $\Phi(\cdot)$ be the identity function, then $\mathbf{Y} = \Phi(\mathbf{X}) = I(\mathbf{X}) = \mathbf{X}$. If $\Phi(\cdot)$ is implemented using 1×1 spatial convolution then it effectively performs the operation described by

$$Y^j = \sum_{i=1}^{N_{\text{in}}} w_{ji} X^i. \tag{3}$$

$\forall j = 1, 2, \ldots, N^{\text{out}}$. n_b is a hyperparameter that can be tuned to improve performance on the validation set. The set of bands $\{B_1, B_2, \ldots, B_{n_b}\}$ is passed as input to Block 2.

2) Block 2 (Band-Specific Spectral-Spatial Feature Learning): Block 2 applies n_b parallel networks, one on each band. Table I explores a variety of choices for these networks. Each convolutional and fully connected layer is followed by a rectified linear unit (ReLU) layer [40], which applies the following operation elementwise on the input volume

$$y = \text{ReLU}(x) = \max(0, x). \tag{4}$$

The outputs of the parallel networks are concatenated and fed into Block 3.

3) Block 3 (Summarization and Classification): Block 3 summarizes the concatenated outputs of the band-specific networks of Block 2 by using a set of fully connected layers, each of which is followed by an ReLU layer. A C-way softmax layer does the final classification by calculating the conditional probabilities of the C output classes, $\mathbf{p} = [p_1, p_2, \ldots, p_C]$ as

$$p_i = \frac{e^{z_i}}{\sum_{i=1}^{C} e^{z_i}} \tag{5}$$

where $\mathbf{z} = [z_1, z_2, \ldots, z_C]$ is the input to the softmax layer.

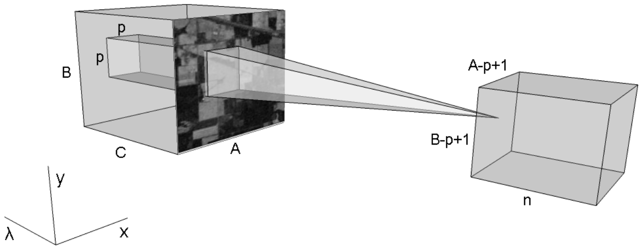


Fig. 2. Diagrammatic representation of $\text{conv}_{xy} - p, n$ on an $A \times B \times C$ input volume.

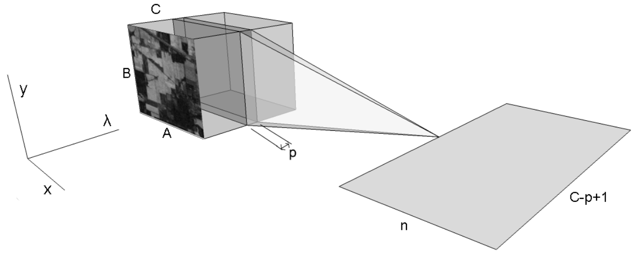


Fig. 3. Diagrammatic representation of $\text{conv}_{\lambda} - p, n$ on an $A \times B \times C$ input volume.

B. Architectures Explored

Table I shows four different network configurations (1–4) and their validation accuracies on the Indian Pines data set (see Section III-A) in an attempt to demonstrate the effect of different architectural design choices on the performance of the network. Only weight layers have been shown to avoid clutter. In all the four configurations, the parallel networks in Block 2 have identical architecture. Block 2 row shows the architecture of one of the parallel networks. Each conv and fc layer (except the last one in Block 3) is followed by an ReLU layer. Cells with an asterisk (*) in the beginning mark the salient points of difference of the corresponding configuration from the one to the left of it. PS = ON/OFF indicates whether parameter-sharing is ON/OFF among the networks of Block 2. When PS = ON, the corresponding weights and biases of the networks are tied and their update-values are averaged across all the networks during back-propagation. Thus, all the networks remain identical. When PS = OFF, the networks are allowed to train independently. $\text{conv}_{xy} - p, n$ represents a spatial convolutional layer with a receptive field size of $p \times p$ and n output spectral-channels. $\text{conv}_{\lambda} - p, n$ represents a spectral convolutional layer with a spectral receptive field of size p and n output spatial-channels. Each convolutional layer, spatial or spectral, consists of a set of 3-D filters, one corresponding to each output channel. Each filter in a $\text{conv}_{xy} - p, n$ layer has a spatial extent of $p \times p$ and extends throughout the entire spectral axis of the input volume (Fig. 2). On the other hand, a filter in a $\text{conv}_{\lambda} - p, n$ layer has a spectral extent of p and extends throughout the spatial extent of the input volume (Fig. 3). All convolutions used in our networks are "valid," which means that there is no zero-padding at the boundaries of the input volume during convolution. As shown in Figs. 2 and 3, if we have an $A \times B \times C$ input volume then the output volumes of a $\text{conv}_{xy} - p, n$ layer and a $\text{conv}_{\lambda} - p, n$ layer with valid convolutions will, respectively, be $(A-p+1) \times (B-p+1) \times n$ and $n \times 1 \times (C - p + 1)$. $fc - n$ denotes a fully connected layer with n nodes. Spectral pooling [31] layers, which tend to induce spectral invariance, are avoided, because our networks are supposed to be discriminative to spectral location.

Significances of different design choices, described in Table I, are as follows.

1) Parameter sharing of the parallel networks in Block 2 (PS = ON) yields an improvement of validation accuracy by at least 1% over PS = OFF in all the four configurations. This confirms that reducing the number of free parameters through parameter sharing leads to better generalization by reducing chances of overfitting.
2) Configuration 2 is constructed by replacing the first fully connected layer in Block 2 in Configuration 1 with two spectral convolution layers. Higher validation accuracy of Configuration 2 can be attributed to fewer parameters in Block 2 than Configuration 1.
3) Configurations 1 and 2 have $\Phi(\cdot) = I(\cdot)$. Configuration 3 is constructed by replacing $I(\cdot)$ in Block 1 in Configuration 2 with a 1×1 spatial convolution followed by an ReLU. An improvement in validation accuracy is observed. This demonstrates the importance of a nontrivial spectral feature selection function. Such a function increases the discriminative power of the network by adding more parameters and nonlinearity.
4) Configuration 4 is constructed by replacing the last fully connected layer in Block 2 in Configuration 3 with two spectral convolution layers and removing the first fully connected layer of Block 3. This construction improves the validation accuracy further by 1% with parameter sharing in Block 2 and 0.5% without. This shows that in the presence of a nontrivial spectral feature selection function in Block 1, reducing the number of parameters in Blocks 2 and 3 can help achieve better generalization by reducing overfitting. This also shows that adding more spectral convolution layers in Block 2 and reducing the number of fully connected layers in Block 3 lead to better performance.

We use Configuration 4 with input patch-size 3×3 and PS = ON in all the experiments of our comparative study in Sections III-D and III-E with some minor modifications for the Salinas and U. Pavia data sets as listed in the following.

1) The 1×1 spatial convolution layer of Block 1 has 224 and 100 output channels for Salinas and U. Pavia, respectively.
2) The number of parallel networks in Block 2, n_b, is 14 and 5 for Salinas and U. Pavia, respectively.
3) In the case of Salinas data set, the last layer of Block 3 is $fc - 16$ as the number of output classes is 16.

C. Learning Algorithm

The networks are trained by minimizing the cross-entropy loss function [22]. If $\mathcal{C}$ be the total number of output classes, $\{X_i, y_i\}_{i=1}^{N}$ be the training set, $P_{\text{data}}(\text{class} = c|X)$ and

$P_{\text{model}}(\text{class} = c|X)$, $\forall c = 1, 2, \ldots, \mathcal{C}$ be the observed and model conditional distributions, respectively, then the cross entropy loss function, $\mathcal{L}_{\times-\text{entropy}}$ is given by

$$\mathcal{L}_{\times-\text{entropy}} = -\sum_{i=1}^{N}\sum_{c=1}^{\mathcal{C}} P_{\text{data}}(c|X_i)\log(P_{\text{model}}(c|X_i)). \quad (6)$$

In our data sets, the observed conditional distribution P_{data} is a one-hot distribution, that is

$$P_{\text{data}}(\text{class} = i|X) = \begin{cases} 1, & \text{if } y = i \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Hence, the expression of cross-entropy loss function becomes

$$\mathcal{L}_{\times-\text{entropy}} = -\sum_{i=1}^{N} \log(P_{\text{model}}(y_i|X_i)). \quad (8)$$

Thus, minimizing this expression is equivalent to maximizing the log-likelihood of the target labels given the inputs.

The Adam optimizer [41] is used for making the parameter updates. It computes adaptive learning rates for each parameter. The base learning rate is set to 0.0005 and batch-size to 200. Dropout, with probability 0.5, is applied to the fully connected layers of Block 3. Dropout is an effective method of regularizing neural networks by preventing coadaptation of features [42]. Batchnorm [43] is observed to degrade the performance of our network and, hence, is not used.

III. Experimental Results

We first present the details of the data set used; followed by the classification performances.

A. Data Sets

The experiments are performed on three popular HSI classification data sets—Indian Pines [44], Salinas, and Pavia University scene (U. Pavia).[2] Some classes in the Indian Pines data set have very few samples. We reject those classes and select the top nine classes by population for experimentation. The problem of insufficient samples is less severe for Salinas and U. Pavia and all the classes are taken into account; 200 labeled pixels from each class are randomly picked to construct a training set. The rest of the labeled samples constitute the test set. A validation set is extracted from the available training set for tuning the hyperparameters of the model. As different frequency channels have different dynamic ranges, their values are normalized to the range [0, 1] using the transformation $f(\cdot)$ defined in (9), where x denotes the random variable corresponding to the pixel values of a given channel

$$f(x) = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (9)$$

B. Evaluation Metrics

We evaluate the proposed architecture in terms of the following metrics.

TABLE II
Data Sets Used

	Indian Pines	Salinas	U. Pavia
Sensor	AVIRIS	AVIRIS	ROSIS
Place	Northwestern Indiana	Salinas Valley California	Pavia, Northern Italy
Frequency Band	0.4-0.45μm	0.4-0.45μm	0.43-0.86μm
Spatial Resolution	20m	20m	1.3m
No. of Channels	220	224	103
No. of Classes	16	16	9

1) Class-Specific Accuracy: Class specific accuracy for class C_i is calculated as the fraction of samples from class C_i, which were correctly classified.

2) Overall Accuracy: Overall accuracy (OA) is the ratio of the total number of correctly classified samples to the total number of samples of all classes.

3) Macroaveraged and Microaveraged Precision, Recall and F-Score: Let TP, TN, FN, and FP denote, respectively, the number of true positive, true negative, false negative, and false positive samples. Then

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (10)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

$$F - \text{score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \quad (12)$$

Let M(TP, FP, TN, FN) be an evaluation metric, e.g., precision, recall, F-score. The macroaveraged and microaveraged values of the metric can be calculated as

$$M_{\text{micro}} = M\left(\sum_{c=1}^{N} TP_c, \sum_{c=1}^{N} FP_c, \sum_{c=1}^{N} TN_c, \sum_{c=1}^{N} FN_c\right) \quad (13)$$

$$M_{\text{macro}} = \frac{1}{N}\sum_{c=1}^{N} M(TP_c, FP_c, TN_c, FN_c) \quad (14)$$

where N is the total number of output classes. A significantly lower value of the microaverage of a metric than the macroaverage indicates that the less populated labels are correctly classified, while the most populated labels have been grossly misclassified and vice versa [45].

4) κ-Score: The κ-score or κ-coefficient is a statistical measure of the degree of agreement among different evaluators [46]. Suppose there are two evaluators that classify N items into C mutually exclusive classes. Then, the κ-score is given by the following equation:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \quad (15)$$

where p_0 is the relative observed probability of agreement and p_e is the hypothetical probability of chance agreement. $\kappa = 1$ indicates complete agreement between the evaluators, while $\kappa \leq 0$ means there is no agreement at all.

[2] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

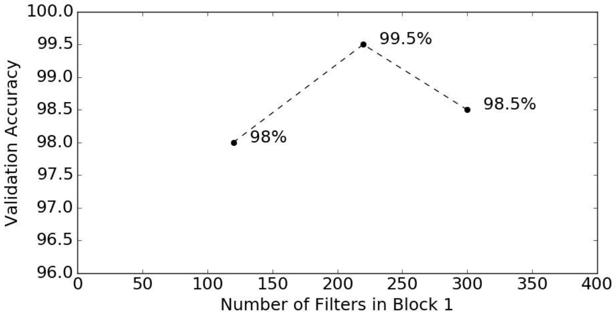


Fig. 4. Variation of validation accuracy on the Indian Pines data set with the number of output channels in Block 1 in Configuration 4.

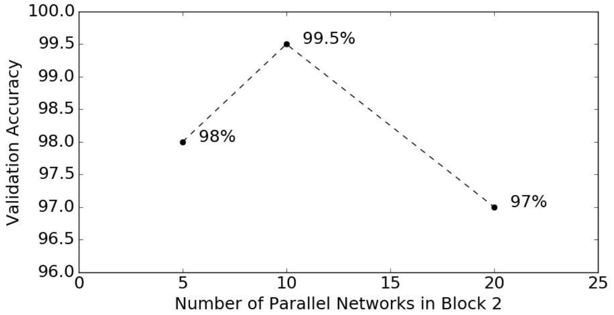


Fig. 5. Variation of validation accuracy on the Indian Pines data set with the number of parallel networks in Block 2 in Configuration 4.

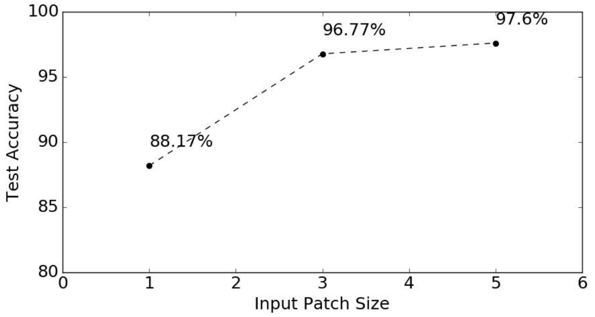


Fig. 6. Variation of test accuracy on the Indian Pines data set with input patch-size in Configuration 4.

C. Implementation Platform

The networks are implemented in Torch,[3] a popular deep learning library written in Lua. The models are trained on an NVIDIA Tesla K20c GPU.

D. Comparison of Different Hyperparameter Settings

Figs. 4 and 5 show the effect of changing the number of output channels of 1 × 1 spatial convolution in Block 1 and the number of networks in Block 2 in Configuration 4 on validation accuracy on Indian Pines. Fig. 6 shows test accuracies on Indian Pines for different choices of input patch-size. Increasing the patch-size gives more spatial context, which results in a marginally better accuracy of classification. However, due to an increased number of parameters, the model might tend to learn the data set bias and fail to generalize to samples outside the image region from which the training and testing samples were extracted.

TABLE III

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE INDIAN PINES DATA SET

Class	k-NN	SVM	ELM	MLP	CNN	PPF	BASS
1	61.83	88.73	86.06	77.77	78.58	92.99	96.09
2	72.65	91.20	88.19	79.05	85.23	96.66	98.25
3	95.65	97.52	96.07	94.70	95.75	98.58	100
4	98.90	99.86	99.73	98.11	99.81	100	99.24
5	100	100	100	99.64	99.64	100	100
6	80.76	91.67	90.02	83.68	89.63	96.24	94.82
7	59.39	78.79	71.00	79.60	81.55	87.80	94.41
8	75.72	93.76	95.62	89.31	95.42	98.98	97.46
9	94.86	98.74	98.66	98.12	98.59	99.81	99.90
OA	76.24	89.83	87.33	85.48	86.44	94.34	**96.77**

TABLE IV

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE SALINAS DATA SET

Class	k-NN	SVM	ELM	MLP	CNN	PPF	BASS
1	98.71	99.55	99.75	99.67	97.34	100	100
2	99.65	99.92	99.87	99.77	99.29	99.88	99.97
3	99.09	99.44	99.60	98.37	96.51	99.60	100
4	99.78	99.86	99.64	99.75	99.66	99.49	99.66
5	95.28	98.02	98.81	98.83	96.97	98.34	99.59
6	99.49	99.70	99.67	99.68	99.60	99.97	100
7	99.55	99.69	99.66	99.29	99.49	100	99.91
8	63.53	84.85	84.04	75.96	72.25	88.68	90.11
9	95.94	99.58	99.89	99.27	97.53	98.33	99.73
10	91.98	96.49	95.03	96.07	91.29	98.60	97.46
11	98.41	98.78	96.82	97.93	97.58	99.54	99.08
12	99.84	100	100	100	100	100	100
13	98.69	99.13	98.25	99.58	99.02	99.44	99.44
14	97.38	98.97	97.94	98.96	95.05	98.96	100
15	65.66	76.38	72.96	75.93	76.83	83.53	83.94
16	99.00	99.56	99.06	98.51	98.94	99.31	99.38
OA	86.29	93.15	92.42	90.78	89.28	94.80	**95.36**

TABLE V

CLASS-SPECIFIC ACCURACY (%) AND OA OF DIFFERENT TECHNIQUES FOR THE PAVIA UNIVERSITY SCENE DATA SET

Class	k-NN	SVM	ELM	MLP	CNN	PPF	BASS
1	77.70	87.95	81.32	91.73	88.38	97.42	97.71
2	75.30	91.17	90.91	94.79	91.27	95.76	97.93
3	77.27	86.99	85.09	85.41	85.88	94.05	94.95
4	92.46	95.50	96.61	94.13	97.24	97.52	97.80
5	99.63	99.85	99.63	99.65	99.91	100	100
6	79.50	94.31	94.33	90.87	96.41	99.13	96.60
7	92.86	94.74	95.94	92.56	93.62	96.19	98.14
8	76.45	85.89	82.65	83.19	87.45	93.62	95.46
9	99.62	99.89	99.79	99.73	99.57	99.60	100
OA	79.45	91.10	89.86	92.54	92.27	96.48	**97.48**

E. Comparison With Other Methods

The test accuracies of the BASS Net architecture (BASS) for the Indian Pines, Salinas, and U. Pavia data sets are compared with other traditional and deep learning-based classifiers in Tables III–V. All the classifiers are trained on the same training set and tested on the same test set for a fair comparison. Among traditional classifiers *k*-NN, SVM with random feature selection [11] and ELM [17] are compared. *k*-NN is implemented in *scikit learn*[4] with *k* equal to the number of classes for each data set. SVM with random feature selection is implemented as described by Waske *et al.* [11] with Gaussian (RBF) kernel, ensemble size 50, 200 training samples per class, and feature subset sizes (percentage of total

[3] http://torch.ch

[4] http://scikit-learn.org

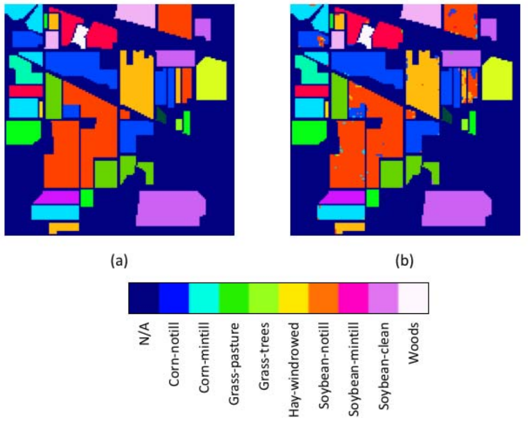


Fig. 7. Thematic maps resulting from classification for the Indian Pines data set with nine classes. (a) Ground-truth map. (b) Decoded output from our model.

number of features) sampled from a uniform distribution over $\{10, 20, \ldots 90\}$. The regularization parameter C and the kernel parameter g are determined by grid search using threefold cross validation. *Libsvm*[5] is used as a platform. ELM is implemented in the exact same way as described by Li *et al.* [17] with the same data sets, sample size, and hyperparameter values using code downloaded from the Web page.[6] The number of selected bands of linear prediction error is set to 11. (m, r) and patch-size for the LBP operator are set to $(8, 2)$ and 21×21, respectively. Bandwidth of the Gabor filter and Gaussian kernel parameters are optimized by fivefold cross validation. Among deep learning-based classifiers, an N_c-150-100-50-C multilayer perceptron (MLP), the CNN architecture of Hu *et al.* [31], and the CNN with PPFs of Li *et al.* [35] are implemented in Torch. The MLP is trained by Adam optimizer with base learning rate 0.0005 and batch-size 200. Parameters of the learning algorithm of the CNN as well as the training set size are set equal to the values mentioned in [31] with the only exception that the number of output classes for Indian Pines is chosen as 9 as opposed to 8 in the original paper. PPF is implemented with the same architecture, learning algorithm and hyperparameter values as mentioned by Li *et al.* [35].

F. Results and Discussion

Thematic maps resulting from the classification of Indian Pines, Salinas, and U. Pavia scenes using our network are presented alongside ground truth in Figs. 7–9, respectively. Tables III–V show the results of the comparison of the proposed framework with traditional and deep learning-based methods. The proposed framework outperforms all the other methods on all the three data sets in terms of OA of classification. For example, on Indian Pines, the test accuracy of our network exceeds SVM, CNN, and PPF by 6.94%, 10.33%, and 2.43%, respectively. Fig. 10 compares the variation of validation accuracy over epochs of training on Indian Pines. Our network converges faster than MLP and CNN. Table VI gives microaveraged and macroaveraged precision, recall, F-score, and κ-score for our models trained on the three data sets. High values of both macroaveraged and microaveraged precision, recall, and F-score suggest that the classifier is effective for both scarce and abundant classes. To further validate our claim, we trained our Configuration 4 network on all 16 classes of Indian Pines, taking 200 samples from each of the nine most populated classes and 80% of the samples from each of the remaining seven classes. The results obtained are corroborative and are presented in Table VII.

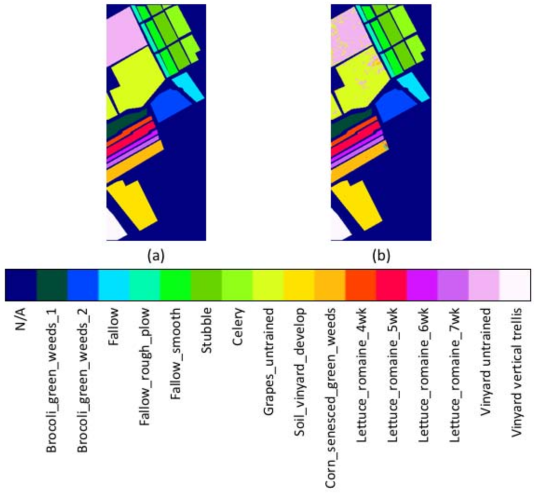


Fig. 8. Thematic maps resulting from classification for the Salinas data set with 16 classes. (a) Ground-truth map. (b) Decoded output from our model.

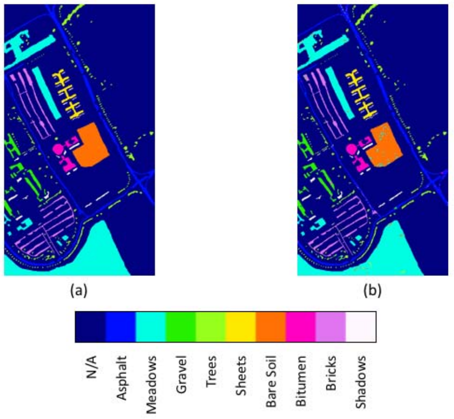


Fig. 9. Thematic maps resulting from classification for the Pavia University scene data set with nine classes. (a) Ground-truth map. (b) Decoded output from our model.

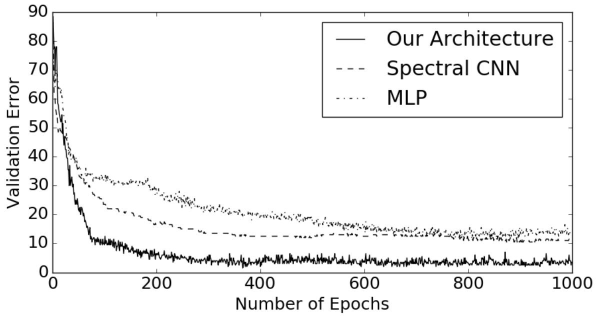


Fig. 10. Variation of validation error over epochs of training on the Indian Pines data set for the proposed architecture and other popular deep neural networks.

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[6] http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

TABLE VI

CLASSIFICATION PERFORMANCE STATISTICS

		Indian Pines	Salinas	U. Pavia
micro-averaged	precision	0.9677	0.9536	0.9748
	recall	0.9677	0.9536	0.9748
	F-score	0.9677	0.9536	0.9748
macro-averaged	precision	0.9713	0.9730	0.9680
	recall	0.9779	0.9802	0.9762
	F-score	0.9745	0.9764	0.9719
κ- score		0.9612	0.9480	0.9662

TABLE VII

CLASSIFICATION PERFORMANCE STATISTICS FOR ALL 16 CLASSES OF INDIAN PINES

test accuracy		0.9503
micro-averaged	precision	0.9503
	recall	0.9503
	F-score	0.9503
macro-averaged	precision	0.9447
	recall	0.9781
	F-score	0.9591
κ-score		0.9411

TABLE VIII

OA STATISTICS OVER 20 INDEPENDENT EXPERIMENTS

	Indian Pines	Salinas	Pavia
mean	95.17	94.26	96.81
std	0.304	0.19	0.13

High values of κ-score for all the data sets show that the proposed classifier has a high degree of agreeability with the ground truth generating mechanism. In order to test whether our network architecture gives high performance consistently across different choices of the training set, we repeat the experiments 20 times with disjoint training sets. In each experiment, we train the Configuration 4 network on a unique sample of 200 pixels from each class and test on the remaining pixels. Table VIII shows the mean and standard deviation of OA over these 20 experiments for each data set. Mean OA is high and standard deviation is low for all three data sets, which show that the proposed architecture performs consistently and the superior performance is not specific to a cherry-picked selection of training and test sets.

IV. CONCLUSION

In this paper, an end-to-end deep learning neural network architecture has been proposed to directly address the problems of the curse of dimensionality, scarcity of labeled data, and spatial variability of spectral signature pertaining to HSI classification. Band-specific spectral-spatial feature learning and extensive parameter sharing in the neural network help achieve superior classification performance and faster convergence than other popular deep learning-based methods on benchmark HSI classification data sets.

REFERENCES

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

[2] J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction*. New York, NY, USA: Springer, 2013.

[3] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson. (2013). "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods." [Online]. Available: https://arxiv.org/abs/1310.5107

[4] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[5] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.

[6] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[7] W. Li, Q. Du, F. Zhang, and W. Hu, "Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 389–393, Feb. 2015.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[9] 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.

[10] L. Gao *et al.*, "Subspace-based support vector machines for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 349–353, Feb. 2015.

[11] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.

[12] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 894–898, Sep. 2011.

[13] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.

[14] F. A. Mianji and Y. Zhang, "Robust hyperspectral classification using relevance vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2100–2112, Jun. 2011.

[15] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, "E^{2}LMs: Ensemble extreme learning machines for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, Apr. 2014.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[17] 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.

[18] T. Lu, S. Li, L. Fang, L. Bruzzone, and J. A. Benediktsson, "Set-to-set distance-based spectral–spatial classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7122–7134, Dec. 2016.

[19] Y. Yuan, J. Lin, and Q. Wang, "Dual-clustering-based hyperspectral band selection by contextual analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1431–1445, Mar. 2016.

[20] Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, Jun. 2016.

[21] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[23] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[24] 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.

[25] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 2015, p. 20, Jul. 2015.

[26] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[27] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[28] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[30] 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.

[31] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, 2015, Art. no. 258619, doi:10.1155/2015/258619.

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

[33] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, Jan. 2017.

[34] 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.

[35] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[36] B. Wu and Z. Xiong, "Adaptive optimally segmentation of spectra for hyperspectral imagery classification," in *Proc. 3rd Int. Congr. Image Signal Process.*, 2010, pp. 2094–2098.

[37] X. Zhou, S. Li, F. Tang, K. Qin, S. Hu, and S. Liu, "Deep learning with grouped features for spatial spectral classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 14, no. 1, pp. 97–101, Jan. 2017.

[38] A. Mughees, X. Chen, and L. Tao, "Unsupervised hyperspectral image segmentation: Merging spectral and spatial information in boundary adjustment," in *Proc. Annu. Conf. SICE*, 2016, pp. 1466–1471.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1989.

[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2011, pp. 315–323.

[41] D. P. Kingma and J. Ba. (2015). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

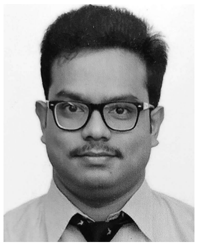
[42] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[43] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[44] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe. (Sep. 2015). *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. [Online]. Available: https://purr.purdue.edu/publications/1947/1

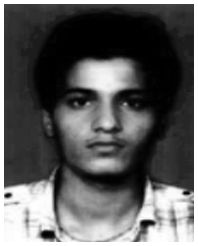
[45] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009.

[46] J. Cohen, "A coefficient of agreement for nominal scales," *Edu. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.



Anirban Santara received the B.Tech. degree in electronics and electrical communication engineering from IIT Kharagpur, Kharagpur, India, in 2015.

He is a currently a Google India Ph.D. Fellow with the Department of Computer Science and Engineering, IIT Kharagpur. His research interests encompass deep learning, reinforcement learning, and computer vision.

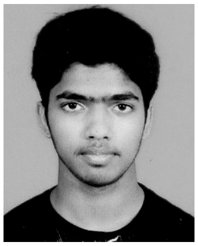


Kaustubh Mani is currently pursuing the integrated M.Sc. degree in exploration geophysics with the Department of Geology and Geophysics, IIT Kharagpur, Kharagpur, India.

He has been actively involved in artificial intelligence and machine learning research. His research interests include deep neural networks, computer vision, and natural language processing.

Pranoot Hatwar was born in India in 1995. He has been pursuing the B.Tech. degree in instrumentation engineering with the Department of Electrical Engineering, IIT Kharagpur, Kharagpur, India, since 2013.

His research interests include deep learning and computer vision.



Ankit Singh was born in India in 1995. He has been pursuing the B.Tech. degree in instrumentation engineering with the Department of Electrical Engineering, IIT Kharagpur, Kharagpur, India, since 2013.

His research interests include deep learning and computer vision.

Ankur Garg is currently pursuing the master's degree in computer science and engineering with IIT Kanpur, Kanpur, India.

He joined the Space Applications Centre, Indian Space Research Organization (ISRO), Ahmedabad, India, in 2014. He has involved in the development of data processing and classification software for ISRO airborne hyperspectral missions. He has developed algorithms and software for closed-loop radiometric quality improvement of medium and high-resolution Indian remote sensing satellites, such as for Cartosat and Resourcesat. His research interests include deep learning, computer vision, and image processing, especially in the fields of classification, blind deconvolution, image deblurring, denoising, and superresolution.

Kirti Padia joined the Space Applications Centre, Indian Space Research Organization (ISRO), Ahmedabad, India, in 1983. He has involved in the development of data processing software for ISRO airborne and spaceborne optical and microwave sensors, such as AHYSI, IRS, MSMR, O2SCAT, RISAT-1, M2S, and ERS-1. He has developed algorithms and software for In-SAR data processing using ERS tandem mission data. His research interests include algorithms and software development for data processing related work for airborne and spaceborne hyper spectral sensors, microwave radiometers, scatterometers, SARs, altimeters, radio occultation payloads, and 3-D visualization.

Pabitra Mitra received the B.Tech. degree in electrical engineering from IIT Kharagpur, Kharagpur, India, and the Ph.D. degree from the Indian Statistical Institute, Kolkata, India.

He is currently an Associate Professor with IIT Kharagpur. His research interests include pattern recognition and machine learning.