Study of Spatial–Spectral Feature Extraction Frameworks With 3-D Convolutional Neural Network for Robust Hyperspectral Imagery Classification

Bishwas Praveen, *Graduate Student Member, IEEE*, and Vineetha Menon, *Student Member, IEEE*

***Abstract*—Advances in hyperspectral remote sensing have instigated multitude of applications for better understanding of our planet through remote data acquisition and observation of natural phenomena such as weather monitoring and prediction to include tornado, wild fires, global warming, etc. For this, data analysis methods that exploit the rich spectral and spatial information in hyperspectral data are often employed to gain insights about the natural phenomenon. This work presents a new deep learning based hyperspectral data analysis framework, which efficiently utilizes both spatial and spectral information present in the data to achieve superior classification performance. Gabor filtering is used for spatial feature extraction in conjunction with sparse random projections for spectral feature extraction and dimensionality reduction. Finally, supervised classification using a 3-D convolutional neural network was employed to perform a volumetric hyperspectral data analysis. Experimental results reveal that the proposed spatial–spectral hyperspectral data analysis frameworks outperform the conventional 2-D convolution neural network-based spectral–spatial feature extraction techniques.**

***Index Terms*—3-D convolutional neural network (3-D CNN), deep learning, dimensionality reduction (DR), feature extraction, Gabor filtering, Gaussian filtering, hyperspectral classification, principal component analysis (PCA), sparse random projection (RP), support vector machine (SVM).**

I. Introduction

TECHNOLOGICAL advancements in the hyperspectral remote sensing domain in conjunction with the ease at which data is acquired utilizing various platforms such as drones have motivated the inception of various hyperspectral remote sensing applications in the area of earth monitoring and observation such as land cover classification for agriculture [1], city planning [2], airborne surveillance [3], weather monitoring [4], climate change observations [5], etc. With the advancements in hyperspectral sensors technology pacing ahead in the direction of light-weight and portable sensors with significant increase in their spectral, spatial, and temporal resolutions capabilities, there is a crucial necessity for smart Big Data Analysis techniques that can stay abreast of sensor technology advancements. In order for these limited hardware platform-centered applications to deliver performance-oriented results, there is a crucial necessity for more efficient methods of data analysis for feature/information extraction, detection, and classification methods for hyperspectral remote sensing applications.

Although there are numerous innovative hyperspectral data analysis techniques that have been developed over the last decade, most conventional hyperspectral data analysis approaches make an implicit assumption for availability of the necessary hardware to meet the Big Data processing demands. As a result, the novelty and motivation of our proposed hyperspectral data analysis techniques are to address the need for novel efficient data analysis methods for feature extraction and classification and detection in hyperspectral remote sensing applications. As the nonlinearity, volume, and dimensionality of hyperspectral remote sensing data that needs to be processed increase, it highlights the quandary of Big Data caveats that need to be addressed, such as increased need of computational resources, limited number of training samples, high data-learning overhead, and curse of dimensionality [7]. As a solution, dimensionality reduction (DR) has emerged as an essential prerequisite data science technique because of its abilities to circumvent the curse of dimensionality issue, provide reduction in redundancy of information, computation time and memory requirements.

Conventional frameworks that use high-dimensional data such as hyperspectral imagery acquire data in its full dimensionality using some form of remote signal-acquisition platform (e.g., a satellite, drone, etc.). Consequently, it is always beneficial if such nonlinear/high-dimensional data can be reduced to lower dimensions in order to dramatically reduce the hardware requirements and computational time of the classification system [9]. Some of such DR techniques proposed in literature are principal component analysis (PCA), which is an unsupervised learning technique that transforms correlated data variables into linearly uncorrelated parameters called principal components (PCs; eigenvectors) that are mathematically abstracted [10]. Linear discriminant analysis is a technique in which the data is projected in directions that maximize the between-class scatter and minimize the within-class scatter for discrimination between various classes in the data [11]. Local fisher discriminant analysis is a localized variant of fisher discriminant analysis that takes the local structure of the data into account so that multimodality

Manuscript received September 9, 2020; revised November 12, 2020; accepted December 4, 2020. Date of publication December 23, 2020; date of current version January 15, 2021. This work was supported by the Gulf Research Program of the National Academies of Sciences, Engineering, and Medicine. *(Corresponding author: Bishwas Praveen.)*

The authors are with the Department of Computer Science, University of Alabama in Huntsville, Huntsville, 35899 AL USA (e-mail: bp0052@uah.edu; vineetha.menon@uah.edu).

Digital Object Identifier 10.1109/JSTARS.2020.3046414

in data can be embedded appropriately [12]. Alternatively, sparse random projections (RPs) have been extensively used in recent years as a DR technique due to its demonstrated computational efficacy and ability to yield a data-independent representation and preserve essential information present in the hyperspectral data [13], [14]. Hence, this work utilizes sparse RPs as not only an effective but also light-weight DR component for the purpose of spectral feature extraction.

Various spatial filtering techniques have also been employed for spatial feature extraction in hyperspectral data analysis applications to include Gaussian filtering [16], sparse filtering [17], local binary patterns [15], etc., that can capture local neighborhood information and hence aid in better classification of hyperspectral data. Gabor filtering is used for the purpose of spatial feature extraction for the proposed framework in this journal. A Gabor filter can effectively extract the local image “textures” or “edges” present in the image, which is achieved through convolution of the input hyperspectral image with a Gabor filter bank. The intuition behind Gabor filtering from the hyperspectral remote sensing aspect is that elements/pixels belonging to the same class tend to have uniform frequency components, whereas high frequency components are exhibited by the border pixels or at transitions between multiple classes or groups of elements [20]. As different Gabor filters construct features along various modes of scale and orientation, a unique frequency response is captured that corresponds to the local orientation of edges in the image [18], [19]. In fact, Yosinski *et al.* [39] have shown that deep neural networks trained on images tend to learn first layer features resembling Gabor filters. This further corroborates our intuition of using Gabor filters to extract important spatial information even before the network is trained with our proposed 3-D convolutional neural network (CNN) architecture. This combination creates a balanced system that gives better training performance with respect to computation time, compared to the stand-alone CNN architecture. Thus, the proposed deep learning based spatial–spectral frameworks aim to preserve spatial dependencies between neighboring pixels via Gabor-based spatial feature extraction along all the spectral bands of hyperspectral data in conjunction with sparse RPs for effective spectral feature extraction and DR paradigm for hyperspectral remote sensing applications.

With the emergence of deep learning, hyperspectral remote sensing applications have widely started to incorporate deep learning models for the purpose of classification, object recognition, etc., and have produced exceptional results compared to state-of-the-art techniques [21], [22]. Deep learning models include supervised classification techniques such as CNNs and recurrent neural networks which progressively learn data patterns and construct features based on the information presented in the data. Of late, the CNNs have gained a huge popularity because of their drastic performance gain over the hand-designed features. 2-D CNNs make use of 2-D convolutional kernels to construct feature maps for a single slice/spectral band at once separately which means that 2-D convolutional kernels are unable to leverage context between spectral kernels and hence only take spatial context into consideration. In contrast, 3-D CNNs address this issue by using 3-D convolutional kernels to make segmentation predictions for a volumetric patch of a scan. The ability to leverage interslice/interspectral bands context can lead to improved performance in terms of classification accuracy for a reasonable tradeoff with computation time [27]. Thus, in this work, various 3-D CNN-based spatial-spectral feature extraction and supervised classification frameworks are introduced which deliver enhanced volumetric hyperspectral data-learning. This work aims to study the effects of volumetric hyperspectral data-learning that are underscored by the incorporation of different information modes (spatial and spectral) on the performance of automated hyperspectral data analysis.

Therefore, the novel contributions of the proposed work are summarized as follows.

- A new mode of localized spatial feature extraction using Gabor filters in conjunction with efficient spectral features extraction using sparse RPs is proposed for a framework for hyperspectral data analysis. Gabor filtering offers an effective means to capture physical traits present in hyperspectral images, such as specific orientation and textural information. Gabor wavelets are utilized to construct Gabor filters that are designed to accommodate a variety of image transformations in the form of rotations and dilations, thus aiding in restoring and enhancing the spatial relationship between data points in HSI data.
- The new Gabor-based spatial and sparse RP-based spectral feature extraction model offers a new reduced dimensional data representation which preserves both spatial–spectral information in lower dimensional subspace and also provides substantial reduction in computational time.
- The extracted spatial–spectral features are windowed to preserve the local neighborhood information which is followed by supervised classification using 3-D CNN. Here, the proposed 3-D CNN based spatial–spectral feature extraction framework is driven toward improved automated hyperspectral data analysis.
- This work introduces various novel 3-D CNN based spatial–spectral feature extraction frameworks and conducts a comprehensive study on the effects of incorporation of various spatial–spectral feature extraction techniques on volumetric hyperspectral data analysis.

The remainder of this article is outlined as follows. Section II presents the theoretical description of the individual components in the proposed architecture. This is followed by discussion of the proposed architecture in Section III along with the generalized framework of other models that are used as a basis of comparison, which are then validated under the experimental results in Section IV. Finally, the efficacy of this work is summarized in Section V.

II. Approach Overview

A. Gabor Filtering

Gabor filtering has shown irrefutable prospects for hyperspectral data analysis due to its ability to capture important physical or spatial information present in hyperspectral images such as specific orientation and textural information [28]. Gabor wavelets are utilized to construct Gabor filters that are designed to accommodate a variety of image transformations in the form of rotations and dilations. A particular advantage of Gabor filters is their degree of invariance to scaling, rotation, and translation, while preserving the spatial relationship between neighboring pixels in the hyperspectral data. Hence, spatial features extracted from Gabor filtering have been especially effective in many image processing applications [28]. A Gabor filter-based spatial feature extraction can be achieved through a 2-D Gabor filter function which is given by

$$\psi(\mathbf{p},\mathbf{q}) = \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}\mathbf{p}'^2 + \frac{f^2}{\eta^2}\mathbf{q}'^2\right)} e^{j2\pi f \mathbf{p}'} \tag{1}$$

where $\mathbf{p}' = \mathbf{p}\cos\theta + \mathbf{q}\sin\theta$ and $\mathbf{q}' = -\mathbf{p}\sin\theta + \mathbf{q}\cos\theta$. In spatial domain [as described in (1)], the Gabor filter represents a complex plane wave which is multiplied by a Gaussian function centered at origin. f is the filter's central frequency, θ is the angle of rotation, γ is the sharpness along the Gaussian major axis, and η is the sharpness along the minor axis. The aspect ratio of the Gaussian function is given by $\frac{\eta}{\gamma}$. Therefore, the corresponding frequency domain representation of Gabor filter is given by

$$\boldsymbol{\Psi}(\mathbf{u},\mathbf{v}) = e^{-\frac{\pi^2}{f^2}(\gamma^2(\mathbf{u}'-f)^2 + \eta^2\mathbf{v}'^2)} \tag{2}$$

where $\mathbf{u}' = \mathbf{u}\cos\theta + \mathbf{v}\sin\theta$ and $\mathbf{v}' = -\mathbf{u}\sin\theta + \mathbf{v}\cos\theta$. In frequency domain [as described in (2)], the Gabor filter denotes a single valued Gaussian distribution centered at f. Hence, the spatial (or time) and frequency domain Gabor filter representations, as described in (1) and (2), respectively, provide a streamlined rendition of the general 2-D structure devised by Daugman from the Gabor's original 1-D elementary function [19]. This implied function enforces a set of self-similar filters, i.e., scaled and rotated variants of one another (Gabor wavelets), regardless of the orientation θ and frequency f, that are capable of capturing the distinct orientation changes and textural information present in the hyperspectral image.

B. Principal Component Analysis

PCA is an unsupervised learning technique that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called PCs. Hence, PCA transformation is defined such that the first PC has the largest possible variance and each successive component has the next highest variance, where all the PCs (new basis vectors) are mutually orthogonal and uncorrelated. PCA achieves DR by projecting a high-dimensional data to its lower dimensions such that all the essential and cardinal information in data is preserved by means of rotation of an existing axis to a new location in space as defined by the new PC-basis vectors. PCA chooses a direction of projection such that the mean squared error between the original data and the transformed or projected data is minimized while the variance of the projected data is maximized [29], [30].

Consider a data, $\mathbf{P} \in \mathcal{R}^{n\times d}$, whose columnwise empirical mean is 0, where n rows represent the number of data vectors and d rows give the features or dimensions of the data. Mathematically, the PCA transformation is defined by the linear combination of a set of d-dimensional basis vectors $\mathbf{u}_{(k)} = (u_1,\ldots,u_d)_{(k)}$ that map each row vector $\mathbf{p}_{(i)}$ of $\mathbf{P}$ to a new vector of PC scores $\mathbf{t}_{(i)} = (t_1,\ldots,t_l)_{(i)}$ computed as

$$\boldsymbol{t}_{k(i)} = \mathbf{p}_{(i)}.\mathbf{u}_{(k)} \tag{3}$$

where $i = \{1,\ldots,n\}$, $k = \{1,\ldots,l\}$, and $l \leq d$. In order to maximize variance, the first weight vector $\mathbf{u}_{(1)}$ which corresponds to the first PC thus has to satisfy

$$\begin{aligned} \mathbf{u}_{(1)} &= \arg\max_{\|\mathbf{u}\|=1}\{\Sigma(\mathbf{t}_1)^2_{(i)}\} \\ &= \arg\max_{\|\mathbf{u}\|=1}\{\Sigma(\mathbf{p}_{(i)}.\mathbf{u})^2\}. \end{aligned} \tag{4}$$

Equivalently, (4) can be expressed in matrix form as

$$\begin{aligned} \mathbf{u}_{(1)} &= \arg\max_{\|\mathbf{u}\|=1}\{\| \mathbf{Pu} \|^2\} \\ &= \arg\max_{\|\mathbf{u}\|=1}\{\mathbf{u}^T\mathbf{P}^T\mathbf{Pu}\}. \end{aligned} \tag{5}$$

Since $\mathbf{u}_{(1)}$ has been defined to be a unit vector, it equivalently also satisfies (6) as given below

$$\mathbf{u}_{(1)} = \arg\max\left\{\frac{\mathbf{u}^T\mathbf{C}_m\mathbf{u}}{\mathbf{u}^T\mathbf{u}}\right\} \tag{6}$$

where $\mathbf{C}_m = \mathbf{P}^T\mathbf{P}$ represents the covariance matrix of the data and it gives the largest Eigenvalues (in descending order) corresponding to the largest Eigenvectors $\mathbf{u}$ of the data. For DR purposes, only the first d_r PCs of the computed d are retained, where $d_r \ll d$.

C. Sparse Random Projection

Although PCA is one of the most widely used DR techniques, during PCA transformation, data must be centered first and processed beforehand for the new data projection to be calculated. This process turns out to be computationally very expensive mainly in scenarios where the size of data being processed is extremely large which is the case with hyperspectral data. This also makes PCA impossible to use if the goal is to compute projections for a stream of data points in real time, which is pertinent to the "velocity" attribute of remote sensing Big Data acquired by drones or satellites. In such scenarios, we need more efficient and portable DR techniques such as sparse RPs which are computationally fast and highly efficient basis functions since it uses a randomly distributed generic function to project a high-dimensional data onto a random lower dimensional subspace, while ensuring that all the vital details in the data are preserved with high probability. The generic nature of sparse RP basis facilitates better portability and computational efficiency since there is no data-learning involved as in traditional DR techniques like PCA. Moreover, sparse RPs are an excellent alternative to conventional Gaussian RPs, where the former guarantees similar embedding quality while being substantially more memory efficient than the dense Gaussian projections [29]. If a is defined as $\frac{1}{\text{density}}$ and $density$ gives the ratio of nonzero components in the RP matrix, the elements of the sparse RP matrix $\boldsymbol{R}$ are drawn as described in the following equation:

$$\boldsymbol{R}_{i,j} = \begin{cases} -\sqrt{\frac{a}{d_r}} & \text{with probability} \quad \frac{1}{2a} \\ 0 & \text{with probability} \quad 1-\frac{1}{a} \\ +\sqrt{\frac{a}{d_r}} & \text{with probability} \quad \frac{1}{2a} \end{cases} \tag{7}$$

where d_r is the size of the lower dimensional subspace and density is set to $\frac{1}{\sqrt{d}}$ and $\boldsymbol{R}_{i,j}$ is an orthogonal sparse RP matrix which satisfies the property of spherical symmetry, normality, and orthogonality.

D. Support Vector Machines

Support vector machines (SVMs) are intrinsically binary classifiers. However, in practical image classification tasks, it often needs the simultaneous discrimination of several different classes. Generally, the multiclass extension of SVM classifiers can be done by combining several binary classifiers, among which are two classical strategies one-against-all and one-against-one [31].

The main goal of SVM is to maximize the margin between the two classes and to minimize the risk of generalization errors. Given a set of N training examples with input vectors $\{p_k \in \mathcal{R}^n\}_{k=1}^N$ and corresponding labels $\{q_k \in \{-1, 1\}\}_{k=1}^N$, SVM classification algorithm [32] builds a linear classifier in the feature space given by the following equation:

$$\hat{q}(p) = \text{sign}(q^T \varphi(p) + b) \tag{8}$$

where the feature map $\varphi(p)$ maps the input data into a feature space. According to Mercer's theorem [33], any positive-definite kernel function can be expressed as $K(p, r) = \varphi(p)^T \varphi(r)$. Due to the kernel trick, it is made possible to implicitly work in huge dimensional feature spaces without having to incur the associated computational costs of higher dimensional analysis. Few popular nonlinear kernel functions used with SVM are radial basis function (RBF) kernel defined as $K(p, p_k) = \exp(- \parallel p - p_k \parallel^2)/\sigma^2$ and polynomial kernel $K(p, p_k) = (p^T p_k + \tau)^d$ with $\tau \geq 0$. The predicted label of a new data vector can be described by the following equation as

$$\hat{q}(p) = \text{sign}\left(\sum_{k,l=1}^{N} q_k q_l K(p, p_k) + b\right). \tag{9}$$

There exists another family of kernels for SVM in which two or more kernels are combined together as a weighted sum and are called composite kernels. Let the spatial feature vector of hyperspectral pixel vector $\mathbf{Y_i}$ be denoted as $\mathbf{Y_i^s}$ and its spectral feature vector as $\mathbf{Y_i^\omega}$. Then the weighted-summation composite kernel can be defined as shown in the following equation by

$$K(\mathbf{Y_i}, \mathbf{Y_j}) = \alpha K_s(\mathbf{Y_i^s}, \mathbf{Y_j^s}) + (1 - \alpha) K_\omega(\mathbf{Y_i^\omega}, \mathbf{Y_j^\omega}) \tag{10}$$

where $K_s(\cdot, \cdot)$ represents spatial kernel, $K_\omega(\cdot, \cdot)$ stands for spectral kernel, and $0 < \alpha < 1$. The proposed 3-D CNN based spatial–spectral supervised classification architectures in this work are compared against a composite kernel SVM for hyperspectral classification.

E. Gaussian Filtering

Generally, data acquisition of any kind often introduces some process noise artifacts. As such, hyperspectral remote sensing is not impervious to various types of process noises that are introduced during the data acquisition process such as atmospheric noise, turbulence, cloud cover, etc. Hence, denoising is another core motivation for pursuing DR techniques as a prerequisite to hyperspectral data analysis. Other filtering-based denoising techniques used include Gaussian filtering [16], sparse filtering [17], and median filtering, and other kinds of filters are applied to the data as a preprocessing step to minimize the effects on noise on the data. Therefore, Gaussian filtering is adopted as another spatial feature extraction technique that is used in conjunction with a CNN-based framework to minimize the effects of noise on data analysis and evaluate the robustness of the model in the presence of inherent noise.

The idea behind Gaussian filtering is to use a 2-D point-spread distribution function and convolve it with the input hyperspectral band. Since input data is a collection of pixels with discrete values, a discrete approximation to the Gaussian function needs to be generated before convolving it with the input set of pixels. Theoretically, the Gaussian distribution is nonzero throughout, which brings the necessity of an infinitely large convolution kernel, but in real-world scenarios, it is chosen to be zero for values more than approximately 2–3 standard deviations from the mean. This makes it possible for the kernel to be truncated at a distance of three standard deviations from the mean. Thus, a 2-D circularly symmetric Gaussian kernel can be formulated as

$$\boldsymbol{G}(\mathbf{p}, \mathbf{q}) = \frac{1}{2\pi\sigma^2} e^{-\frac{\mathbf{p}^2 + \mathbf{q}^2}{2\sigma^2}} \tag{11}$$

where σ is the standard deviation and represents the spread of the kernel. The kernel should normally be selected large enough so that the kernel coefficients of the border rows and columns contribute very little to the sum of coefficients. By selecting a kernel size parameter six times the standard deviation, the border parameters will be 1% or lower than the center parameter which is ideal for our application.

F. 3-D Convolutional Neural Network

Similar to traditional deep learning architectures, CNN models are hierarchical architectures where several convolutional layers are stacked on top of each other. Traditional CNNs have 2-D convolutional kernels for applications on 2-D images. However, convolutions in a 2-D CNN can only capture two-dimensional spatial information and neglect the information along the third dimension. To address this concern, Ji *et al.* [35] extended the idea of 2-D CNN used for 2-D images to a 3-D convolution in both space (2-D) and time for video classification. Similarly, low level features for high-dimensional hyperspectral data can be extracted using 3-D convolutions as hyperspectral data is generally generated as a three-dimensional data cube. Therefore, in this work, 3-D CNN is considered to perform 3-D convolutions in order to learn and capture the 3-D local patterns and preserve the spatial–spectral neighboring pixel dependencies across the whole hyperspectral data cube for a volumetric data analysis which in turn aids in better classification performance. 2-D CNNs are applied on 2-D features maps to extract the spatial features, whereas 3-D kernels are convolved over 3-D feature cubes to detect and preserve the 3-D local patterns in the hyperspectral data cube. More specifically, for the case of 3-D CNN, the value at position (p, q, r) on the jth feature map in the ith layer is obtained as shown in the following equation:

$$h^{i,j}_{p,q,r} = f((W_{i,j} * V^{i-1})_{p,q,r} + b_{i,j}) \tag{12}$$

where $W_{i,j}$ and $b_{i,j}$ are the weights and the bias for jth feature map, respectively, V^{i-1} denotes the sets of input feature maps from the $(i-1)^{th}$ layer connected to the current layer, f is the nonlinear function, and $*$ is the convolution operation. The complexity of learned local patterns is closely related to the numbers of 3-D convolutional kernels in the network. With more kernels, the network can learn deeper and more powerful features, but, on the other hand, it will be more susceptible to overfitting. A general principle is that a network should have sufficient convolutional layers to learn deeper features and fewer numbers of feature maps in each layer to reduce the overall computation complexity.

III. Classification Methodologies

A. *Proposed Architecture (GAB-RP-S-3DCNN)*

In our proposed model, Gabor filtering is first utilized for spatial feature extraction directly on the input hyperspectral data cube. Gabor filters are convolved with individual bands of hyperspectral cube separately and features are constructed based on the orientation and frequency responses from the individual Gabor filters. These responses widely vary with changes in values of frequency f and orientation θ, and it is, hence, crucial to select the parameters that give the best outcome when introduced in the framework of the proposed model. Therefore, a value of $f = 0.8$ and $\theta = 0$ was chosen empirically as they produced the optimum results in this work.

Training CNNs requires that every last one of the network's 2-D inputs be convolved with all trainable filters. But hundreds of bands along the spectral dimensions of a hyperspectral image increase the computational cost of training and prediction process. Consequently, sparse RPs are employed with the end goal of computationally efficient DR and spectral feature extraction in order to condense the size of the original high-dimensional hyperspectral data. Additionally, a fascinating view of kernel functions can be provided by the use of sparse RPs. Furthermore, they are also capable of providing a fast basis technique that can map a kernel function into an explicit feature space. In this work, sparse RPs were used to reduce the original data dimensions to $d_r = 30$ for all the datasets that were used to validate the effectiveness of the proposed hyperspectral data analysis methodologies.

After the Gabor filtering-based spatial feature extraction and DR process for spectral feature extraction is accomplished, the proposed model extracts spatial patches of pixels to preserve the local neighborhood information in the input data before being fed into a CNN. So the input which was earlier in the shape of $(h \times w \times d_r)$, where h and w are the height and width of the input data, respectively, and d_r is the new reduced spectral bands of hyperspectral data, is now changed to $(s_j \times s_j \times d_r)$ where s_j is now the size of the input patch that is fed into CNN for training purposes. In this article, the value of s_j is set to 21 (i.e., $s_j \times s_j = 21 \times 21$), which implies that every time a data pixel is trained, the information around its 440 neighbors are also considered. Other values for s_j from 9 through 23 were also experimented, but for larger values of s_j such as 23, the number of neighbors that were accounted for went up drastically which had an adverse effect on the computational time and deteriorated the classification performance of CNN. Conversely, when the size of s_j was too small, for instance, $s_j = 9$, not enough neighborhood information was integrated for data analysis. Thus, the spatial/local neighborhood information was not effectively exploited and resulted in a lower classification performance than that of $s_j = 21$.

Finally, the number of layers of 3-D CNN, the dimensions, and number of trainable filters, dropouts, and fully connected layers play a crucial role in deciding the number of trainable parameters for the network which directly impacts the computational time of the model [36]. In this proposed model and other comparison models, the first layer of 3-D CNN is a convolutional layer of size 8 and trainable filters of dimension $(3 \times 3 \times 7)$ which is followed by a pooling layer where pool size is specified to be $(2 \times 2 \times 2)$. This is followed by two more 3-D convolutions of sizes 16 and 32 with kernel dimensions $(3 \times 3 \times 5)$ and $(3 \times 3 \times 3)$, respectively. This part of the model is trailed by a 2-D convolution layer of size 64 with kernel dimension (3×3). The final part of the network has three dense layers of sizes 256, 128, and C, respectively, with a dropout of 40%, where C is the number of classes or unique elements present in the dataset. The overall 3-D CNN based spatial–spectral feature extraction framework GAB-RP-S-3DCNN is illustrated in Fig. 1.

B. *RP-S-3DCNN*

This approach has no spatial feature extraction component. The aim here is to explore the performance of using a lightweight spectral-only based data analysis framework. Hence, this model directly employs sparse RPs on the input data for spectral feature extraction and DR purposes. The number of reduced dimension components $d_r = 30$ was empirically selected. The resultant reduced dimensional data is windowed to extract the spatial patches for CNN-based classification, where the window size $s_j = 21$ was chosen. The generalized framework for this 3-D CNN based spectral feature extraction model RP-S-3DCNN is depicted in Fig. 2.

C. *PCA-S-3DCNN*

As in the case of former RP-S-3DCNN model, no spatial feature extraction is implemented in this approach. The emphasis here is to understand the effect of conventional spectral feature extraction techniques like PCA in conjunction with CNN-based supervised classification on hyperspectral data analysis. In this approach, PCA is used as for spectral feature extraction and DR. The first 30 PCs ($d_r = 30$) were retained for further processing in all the datasets used for experimentation. The spectral features extracted using PCA where the data is projected on to a lower dimensional subspace by rotating existing axis to a new location in space is defined by the Eigenvectors. These data points are then windowed into size of $s_j \times s_j = 21 \times 21$ before being input into CNN model for supervised classification.

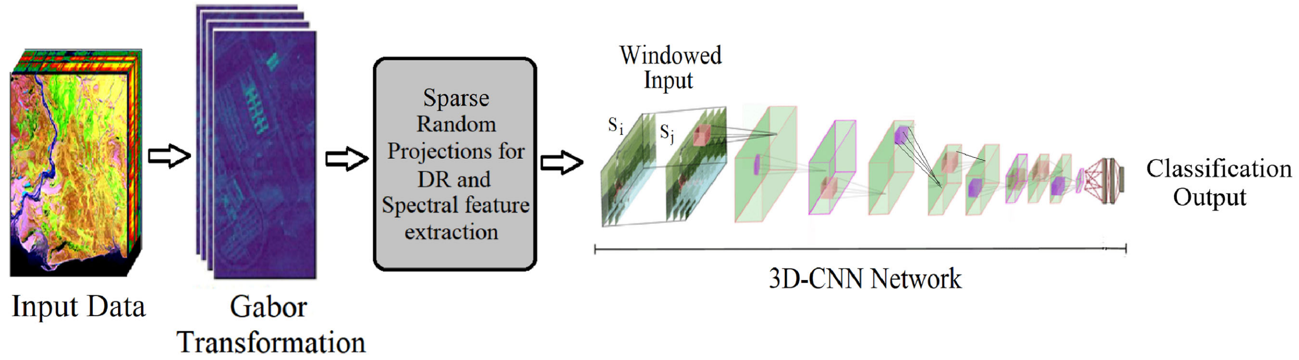

Fig. 1. Proposed 3-D CNN-based spatial–spectral feature extraction system architecture GAB-RP-S-3DCNN.

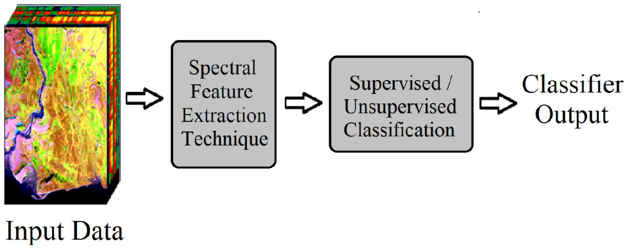
Input Data
Spectral Feature Extraction Technique
Supervised / Unsupervised Classification
Classifier Output

Fig. 2. Generalized spectral feature extraction framework for RP-S-3DCNN and PCA-S-3DCNN data analysis models.

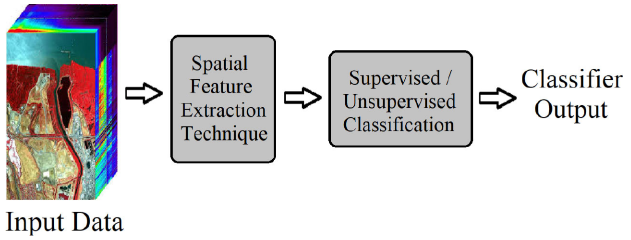
Input Data
Spatial Feature Extraction Technique
Supervised / Unsupervised Classification
Classifier Output

Fig. 3. Generalized spatial feature extraction framework for GAB-S-3DCNN and GAU-S-3DCNN data analysis models.

The generalized framework of this 3-D CNN based spectral feature extraction model PCA-S-3DCNN is as shown in Fig. 2. The numbers of layers and hyperparameters of the 3-D CNN used for classification in RP-S-3DCNN and PCA-S-3DCNN are identical to the one used in GAB-RP-S-3DCNN.

D. GAB-S-3DCNN

In this model, we investigate the effects and contributions of spatial feature extraction alone on hyperspectral data analysis. In GAB-S-3DCNN, Gabor filters are employed for the purpose of spatial feature extraction. Here Gabor filters are convolved with the input hyperspectral bands and features are constructed based on the responses provided by these Gabor filters set at a frequency $f = 0.8$ and orientation $\theta = 0$. Once the spatial features are extracted, the resultant data is windowed ($s_j \times s_j = 21 \times 21$) and given as an input to the supervised CNN classifier. Thus, the generalized framework for GAB-S-3DCNN model is as illustrated in Fig. 3.

E. GAU-S-3DCNN

In this exclusive spatial feature extraction model, Gaussian filtering is utilized for noise reduction purposes. Hyperspectral data pixels are usually contaminated with inherent noise that is introduced during the data acquisition process. As a consequence, this intuitive noise reduction model will characterize the robustness of the proposed model in the presence of noisy conditions. Hence, in this approach, Gaussian filtering was implemented on all the hyperspectral bands which reduced the effect of noise. The resultant denoised data was given as an input to 3-D CNN for classification. The point spread function was set to $\sigma = 0.4$ in this framework. The generalized CNN-based spatial feature extraction framework for GAU-S-3DCNN model is shown in Fig. 3. The number of layers and hyperparameters of the 3-D CNN used for classification in GAB-S-3DCNN and GAU-S-3DCNN are identical to the one used in GAB-RP-S-3DCNN.

F. GAB-RP-S-2DCNN

In this model, Gabor filtering is first utilized for spatial feature extraction directly on the hyperspectral data cube and features are constructed based on the orientation and frequency responses from the individual Gabor filters. A value of $f = 0.8$ and $\theta = 0$ was chosen empirically as they produced the optimum results in this work. Additionally, sparse RPs are employed with the end goal of computationally efficient DR and spectral feature extraction in order to reduce the size of the original high-dimensional hyperspectral data. In this model, sparse RPs were used to reduce the original data dimensions to $d_r = 50$ for all the datasets that were used to validate the effectiveness of the proposed hyperspectral data analysis methodologies.

For this model, the size of the input window for the 2D-CNN is set to 5 (i.e., $s_j \times s_j = 5 \times 5$), which implies that every time a data pixel is trained, the information around its 24 neighbors are also considered. Finally, the number of layers of CNN, the dimensions, and number of trainable filters, dropouts, and fully connected layers play a crucial role in deciding the number of trainable parameters for the network which directly impacts the computational time of the model. In this model, the principal layer of CNN is a convolutional layer of size 150 and trainable filters of dimension (3×3). As opposed to a conventional CNN, no pooling layer is utilized after the convolution layer in this model as the input size is already small and is subsequently trailed by one more convolutional layer. The size of the first layer is 450, the size of the second layer is 150 neurons, and the size of trainable filters is (3×3) in both the layers. This is then forwarded to the two dense convolution layers of size 450 and C, respectively, where C is the number of classes or unique elements present in the dataset [38].

G. SVM-CK

In this work, we validate our proposed spatial–spectral feature extraction-based 3-D CNN model architectures against a traditional SVM with a composite-kernel SVM for an inclusive spatial–spectral information extraction with a window-based spatial mean kernel with a window size of (3×3) coupled with an RBF spectral kernel [denoted by SVM-CK as in (10)]. For spectral features, the hyperspectral pixel vectors are directly used as spectral feature vector and RBF is used as the spectral kernel which makes SVM-CK incorporate both spatial and spectral features present in the hyperspectral data to enhance its classification performance. All the experiments related to SVM model-based hyperspectral classification were conducted using LIBSVM on raw hyperspectral data without the use of any DR techniques.

IV. Experimental Results

In this section, the efficiency of the proposed spatial–spectral feature extraction model GAB-RP-S-3DCNN is validated and compared against six other models, namely RP-S-3DCNN, GAB-S-3DCNN, PCA-S-3DCNN, GAU-S-3DCNN, GAB-RP-S-2DCNN, and SVM-CK as described in Section III. All experiments were conducted on an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) dataset—Indian Pines and a Reflective Optics System Imaging Spectrometer (ROSIS) dataset—University of Pavia [37]. The Indian Pines dataset was acquired by an AVIRIS sensor over the Indian Pines test site in North-Western Indiana. This dataset has a spatial dimension of 145 × 145 and 224 spectral bands (202 after removal of water-absorption bands) with a spatial resolution of 20 m spanning 16 land-cover classes. University of Pavia dataset was acquired by ROSIS sensor over an urban area of Pavia in North Italy. The dataset has 103 spectral bands each having a spatial dimension of 610×340 with a spatial resolution of 1.3 m spanning nine classes of land-cover classes. For each dataset, training set was randomly chosen spanning from 5% through 50%. All parameters in the proposed approach were empirically chosen and optimized. To avoid any bias induced by random sampling of pixels, the classification results are averaged over five trials and the average accuracies along with execution time of the models are tabulated. The objective function used for all our experimentation is categorical cross-entropy with a learning rate of 0.001 and a decay of 10^{-6}. The total number of trainable parameters turned out to be 313 472 and a batch size of 32 was chosen empirically. Table I gives a detailed information of the 3-D CNN architecture used for classification in all our experimentation where n is 16 in the case of Indian Pines dataset and 9 in the case of Pavia University dataset. The networks were trained for 80 epochs on both the datasets. Also, no GPUs were made use of to train any of the proposed frameworks in our work. All experiments were implemented using python on a Intel(R) Core(TM) i7-7700HQ processor with 16-GB RAM machine.

TABLE I
3-D CNN Architecture of Our Proposed Frameworks

Layer (Type)	Output Shape	# Parameters
conv3d (Conv3D)	(None, 19, 19, 24, 8)	512
max_pooling3d (MaxPooling3D)	(None, 9, 9, 12, 8)	0
conv3d (Conv3D)	(None, 7, 7, 8, 16)	5776
conv3d (Conv3D)	(None, 5, 5, 6, 32)	13856
reshape (Reshape)	(None, 5, 5, 192)	0
conv2d (Conv2D)	(None, 3, 3, 64)	110656
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 256)	147712
dropout (Dropout)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, n)	2064

TABLE II
Total Number of Class-Specific Training and Testing Samples Used for Indian Pines Dataset for 10% Training Data

#	Class Name	# of Training Samples	# of Testing Samples
1	Alfalfa	5	41
2	Corn-notill	140	1288
3	Corn-mintill	81	749
4	Corn	24	213
5	Grass-pasture	48	435
6	Grass-trees	72	658
7	Grass-pasture-mowed	3	25
8	Hay-Windrowed	47	431
9	Oats	2	18
10	Soybean-notill	95	877
11	Soybean-mintill	232	2223
12	Soybean-clean	58	535
13	Wheat	21	184
14	Woods	124	1141
15	Buildings-Grass-Trees-Drives	38	348
16	Stone-Steel-Towers	10	83
	Total	**1000**	**9249**

TABLE III
Total Number of Class-Specific Training and Testing Samples Used for University of Pavia Dataset for 10% Training Data

#	Class Name	# of Training Samples	# of Testing Samples
1	Asphalt	663	5968
2	Meadows	1865	16784
3	Gravel	210	1889
4	Trees	306	2758
5	Painted Metal Sheets	134	1211
6	Bare Soil	503	4526
7	Bitumen	133	1197
8	Self-Blocking Bricks	368	3314
9	Shadows	95	852
	Total	**4277**	**38499**

Fig. 4 gives the classification maps for 10% of training data across all the proposed 3-D CNN based spatial–spectral hyperspectral feature extraction and data analysis models for the Indian Pines dataset along with the models used for comparison. Fig. 5 describes the classification maps for all methods using 10% of training data for Pavia University dataset. Tables II and III give the total number of training and testing samples per class for 10% of training data used in experimentation for Indian Pines and Pavia University datasets, respectively.

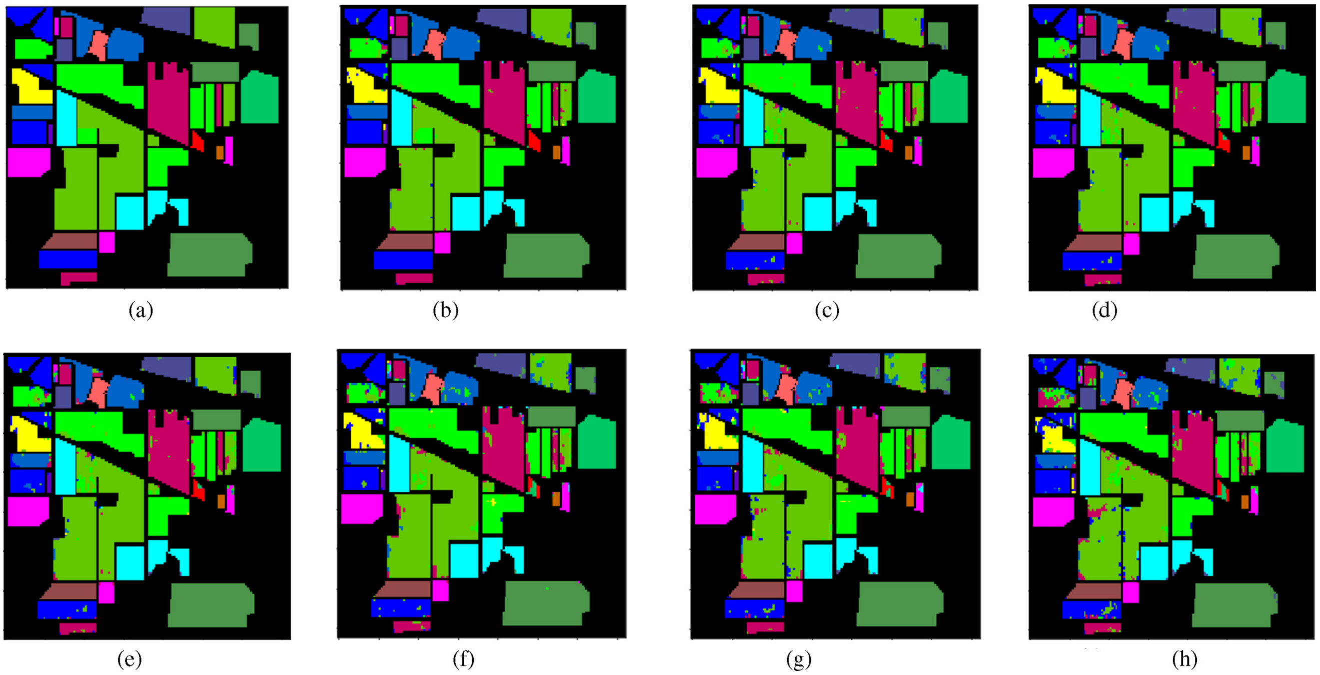
(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Classification maps of Indian Pines for various proposed 3-D CNN based spatial–spectral feature extraction models using 10% training data. (a) Ground truth. (b) GAB-RP-S-3DCNN (96.91%). (c) RP-S-3DCNN (94.01%). (d) GAB-S-3DCNN (95.38%). (e) PCA-S-3DCNN (95.32%). (f) GAU-S-3DCNN (94.47%). (g) GAB-RP-S-2DCNN (91.47%). (h) SVM-CK (90.53%).

TABLE IV
Class-Specific Accuracies of Indian Pines Dataset for 10% of Training Data for the Proposed 3-D CNN Models Used for Comparison

#	Class Name	GAB-RP-S-3DCNN	RP-S-3DCNN	GAB-S-3DCNN	PCA-S-3DCNN	GAU-S-3DCNN	GAB-RP-S-2DCNN	SVM-CK
1	Alfalfa	98.8	86.1	50.9	89.2	91.3	83.8	92.9
2	Corn-notill	93.2	94.2	94.9	93.3	93.6	81.7	92.4
3	Corn-mintill	89.6	83.2	93.0	92.4	97.8	79.3	92.5
4	Corn	91.3	92.1	83.8	97.7	96.9	78.4	91.0
5	Grass-pasture	93.7	94.3	93.5	97.3	96.3	95.8	92.3
6	Grass-trees	97.8	97.8	99.5	99.0	98.4	97.7	83.1
7	Grass-pasture-mowed	95.8	47.8	78.0	95.8	69.2	88.9	95.4
8	Hay-Windrowed	100	99.4	97.3	99.7	97.0	97.5	90.0
9	Oats	72.2	68.3	75.9	36.4	43.7	53.6	98.1
10	Soybean-notill	92.0	95.6	95.9	95.5	95.1	82.4	92.4
11	Soybean-mintill	94.1	92.1	97.8	96.5	96.3	84.8	98.3
12	Soybean-clean	95.3	95.6	90.8	91.1	96.4	99.7	87.5
13	Wheat	96.1	98.4	99.5	93.2	96.4	97.6	96.0
14	Woods	97.2	97.7	95.7	98.7	95.7	86.1	92.2
15	Buildings-Grass-Trees-Drives	98.8	98.4	95.1	95.2	92.0	88.4	97.7
16	Stone-Steel-Towers	88.4	93.6	94.6	87.0	90.8	97.6	93.9
	OA (%)	**96.91**	**94.01**	**95.38**	**95.32**	**94.47**	**91.47**	**92.97**
	κ (%)	**95.06**	**93.15**	**94.74**	**95.02**	**94.00**	**90.37**	**91.83**

From Figs. 4 and 5, it can be noted that our proposed 3-D CNN-based spatial–spectral feature data analysis method GAB-RP-S-3DCNN had more coherent classification regions and fewer misclassifications when compared to other methods. Tables IV and V present the class-specific accuracies for the proposed architecture GAB-RP-S-3DCNN along with all the other frameworks used for performance comparison for Indian Pines and Pavia University datasets, respectively. It can be inferred from Tables IV and V that GAB-RP-S-3DCNN gave superior classification performance over other frameworks that are discussed. The effectiveness of our proposed approach GAB-RP-S-3DCNN can be further affirmed from overall classification accuracy as depicted in Figs. 6 and 7 for Indian Pines and Pavia datasets, respectively. From Figs. 6 and 7, it can be noted that GAB-RP-S-3DCNN outperformed all other methods of comparison, especially against the conventional PCA-based spectral feature extraction model-PCA-S-3DCNN, a 2-D CNN-based hyperspectral data classification model namely GAB-RP-S-2DCNN, and a conventionally used SVM-based spatial–spectral information inclusion model namely SVM-CK. The major reason for the proposed approach GAB-RP-S-3DCNN to produce better results when compared to GAB-RP-S-2DCNN is that the 3-D CNN based methodology leverages the context that exists between spectral bands by building filter maps by volumetric convolutions in contrast to two-dimensional convolutions in the case of 2DCNN-based classification approach. Our method GAB-RP-S-3DCNN gave the best classification performance even when just 10% of training samples were used.

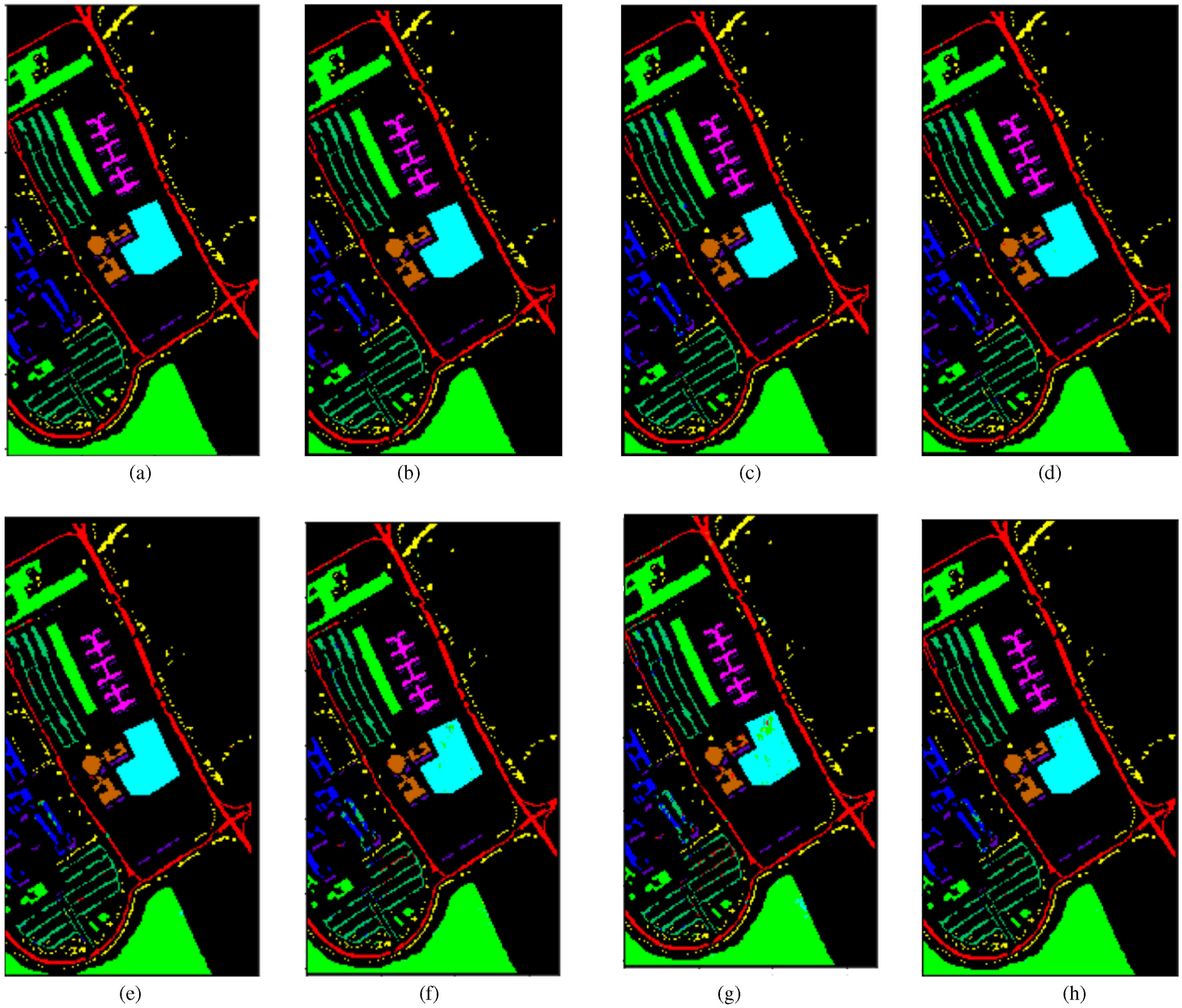

Fig. 5. Classification maps of University of Pavia dataset for various proposed models using 10% training data. (a) Ground truth. (b) GAB-RP-S-3DCNN (99.26%). (c) RP-S-3DCNN (98.42%). (d) GAB-S-3DCNN (98.64%). (e) PCA-S-3DCNN (98.83%). (f) GAU-S-3DCNN (98.30%). (g) GAB-RP-S-2DCNN (95.33%). (h) SVM-CK (94.01%).

TABLE V
Classwise Accuracies of University of Pavia Dataset for 10% of Training Data for the Proposed Approach Along With Models Used for Comparison

#	Class Name	GAB-RP-S-3DCNN	RP-S-3DCNN	GAB-S-3DCNN	PCA-S-3DCNN	GAU-S-3DCNN	GAB-RP-S-2DCNN	SVM-CK
1	Asphalt	98.9	99.8	98.9	99.0	98.1	94.2	93.7
2	Meadows	100	100	99.9	99.9	99.8	98.1	96.4
3	Gravel	97.9	97.0	95.9	95.8	91.3	81.3	88.9
4	Trees	97.6	98.7	97.8	98.7	97.1	97.7	93.7
5	Painted Metal Sheets	100	99.8	99.9	100	99.8	99.3	99.4
6	Bare Soil	99.9	99.9	99.9	99.7	99.5	92.9	98.4
7	Bitumen	99.4	99.7	98.4	99.5	98.4	91.5	97.1
8	Self-Blocking Bricks	98.0	97.7	96.7	96.4	94.4	84.6	92.8
9	Shadows	96.4	97.4	94.3	96.8	96.2	98.9	99.8
	OA (%)	**99.26**	**98.42**	**98.64**	**98.83**	**98.30**	**95.33**	**94.01**
	κ (%)	**99.02**	**97.93**	**98.50**	**98.42**	**97.75**	**94.72**	**93.26**

Tables VI and VII tabulate the overall execution time of all the proposed models for Indian Pines and University of Pavia datasets, respectively. Figs. 8 and 9 present the effect of varying window sizes on classification performance of all methods and substantiates our argument for the choice of window size as (21 × 21) as a tradeoff between computational efficiency and desired classification performance. It can further be inferred from Tables VI and VII that GAB-RP-S-3DCNN indeed provided superior classification at a reasonable tradeoff between computational time and classification performance compared to other 3-D CNN-based spatial- and spectral-only feature extraction models.

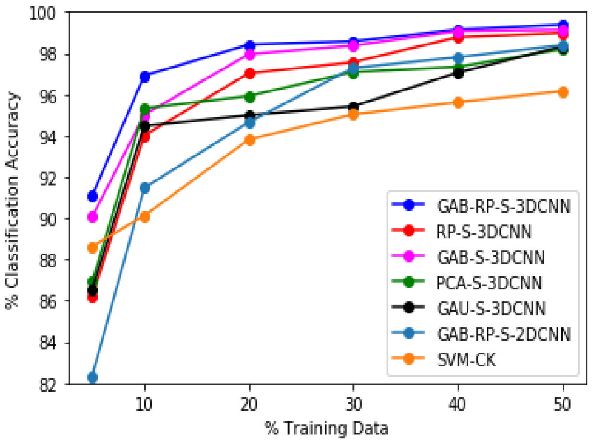

Fig. 6. Overall classification accuracies of Indian Pines dataset for varying sizes of training samples.

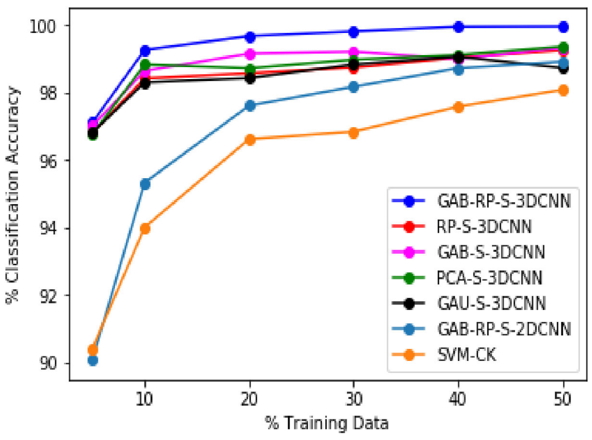

Fig. 7. Overall classification accuracies of University of Pavia dataset for varying sizes of training samples.

TABLE VI
OVERALL EXECUTION TIME OF THE PROPOSED MODELS FOR INDIAN PINES IN MINUTES

% Training data	GAB-RP-S-3DCNN	RP-S-3DCNN	GAB-S-3DCNN	PCA-S-3DCNN	GAU-S-3DCNN	GAB-RP-S-2DCNN	SVM-CK
5	21.75	22.97	21.55	42.07	16.59	12.11	15.66
10	22.02	32.61	54.36	33.64	38.50	20.95	27.11
20	50.60	61.23	70.82	71.98	62.55	41.65	45.62
30	91.35	102.33	92.05	112.37	93.08	62.39	65.92
40	143.56	126.96	150.92	117.46	125.74	65.56	72.04
50	134.73	137.39	183.11	125.76	132.01	68.22	81.33

TABLE VII
OVERALL EXECUTION TIME OF THE PROPOSED MODELS FOR UNIVERSITY OF PAVIA DATASET IN MINUTES

% Training data	GAB-RP-S-3DCNN	RP-S-3DCNN	GAB-S-3DCNN	PCA-S-3DCNN	GAU-S-3DCNN	GAB-RP-S-2DCNN	SVM-CK
5	66.20	59.51	58.26	46.61	46.81	8.06	14.29
10	97.99	85.81	88.18	99.05	58.60	12.76	21.54
20	210.90	145.34	187.18	154.26	87.32	21.98	28.71
30	290.36	190.88	216.33	173.99	129.05	40.05	66.25
40	320.71	215.63	230.12	201.57	191.76	54.65	71.95
50	290.22	247.55	254.78	235.83	258.60	47.90	96.72

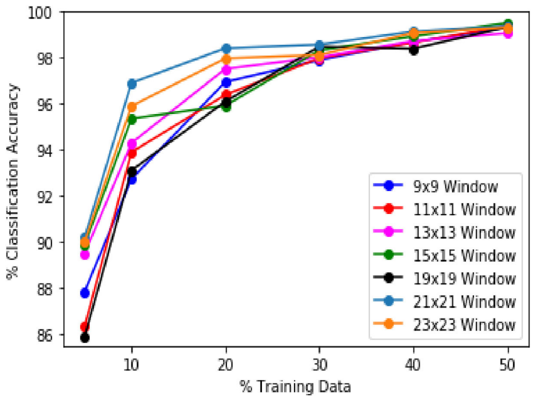

Fig. 8. Overall performance of GAB-RP-S-3DCNN over a range of window size in terms of classification accuracy for Indian Pines dataset.

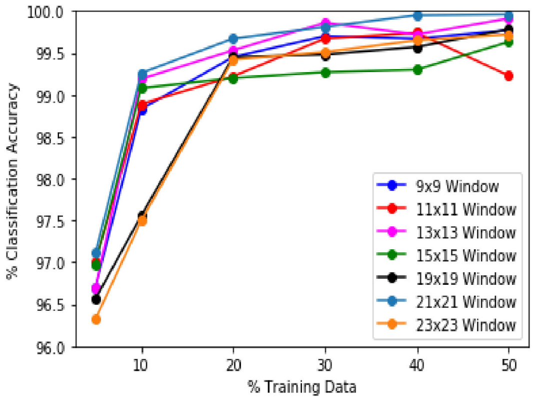

Fig. 9. Overall performance of GAB-RP-S-3DCNN over a range of window size in terms of classification accuracy for University of Pavia dataset.

V. Conclusion

In this work, several novel 3-D CNN based spatial–spectral feature extraction and hyperspectral classification models were introduced. Compared with the traditional 2DCNN-based models, 3-D convolution was used to construct features by exploring both spatial context in neighboring areas and spectral correlation in neighboring bands. Experimental results demonstrate that the proposed 3-D CNN based spatial–spectral feature extraction methodology GAB-RP-S-3DCNN yields outstanding classification performance, while being robust under limited training samples scenarios, when compared to other spatial- and spectral-only based feature extraction approaches. 3-D CNN based classification framework further demonstrates the effectiveness and ability to learn the 3-D patterns in the data and extract features both along spatial and spectral dimensions which aids in better classification performance of hyperspectral data; thereby prompting new exploration avenues for automated hyperspectral data analysis in remote sensing applications.

Acknowledgment

The content is solely the responsibility of the authors and does not necessarily represent the official views of the Gulf Research Program or the National Academies of Sciences, Engineering, and Medicine.

REFERENCES

[1] K. Zhou *et al.*, "Assessment of spectral variation between rice canopy components using spectral feature analysis of near-ground hyperspectral imaging data," in *Proc. 8th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2016 pp. 1–4.

[2] G. Abbate, L. Fiumi, C. De Lorenzo, and R. Vintila, "Evaluation of remote sensing data for urban planning. Applicative examples by means of multispectral and hyperspectral data," in *Proc. 2nd GRSS/ISPRS Joint Workshop Remote Sens. Data Fusion Over Urban Areas*, 2003.

[3] M. I. Vakil, D. B. Megherbi, and J. A. Malas, "An efficient multi-stage hyper-spectral aerial image registration technique in the presence of differential spatial and temporal sensor uncertainty with application to large critical infrastructures and key resources (CIKR) surveillance," in *Proc. IEEE Int. Symp. Technol. Homeland Secur.*, 2015.

[4] L. M. Wickert, J. B. Percival, W. A. Morris, and J. R. Harris, "XRD and infrared spectroscopic validation of weathering surfaces from ultramafic and mafic lithologies examined using hyperspectral imagery, cross lake area, cape smith belt," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Northern Quebec, Canada, 2008.

[5] W. Heldens, T. Esch, and U. Heiden, "Supporting urban micro climate modelling with airborne hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012.

[6] A. Marinoni and H. Clenet, "Identification of mafic minerals on mars by nonlinear hyperspectral unmixing," in *Proc. 8th Workshop Hyperspectral Image Signal Process. : Evol. Remote Sens.*, 2016.

[7] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[8] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[9] J. E. Fowler, Q. Du, W. Zhu, and N. H. Younan, "Classification performance of random-projection-based dimensionality reduction of hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2009, pp. V-76–V-79

[10] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[12] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, no. 5, pp. 1027–1061, May 2007.

[13] V. Menon, Q. Du, and S. Christopher, "Improved random projection with K-means clustering for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, Mar. 2017.

[14] N. Hoai, Q. Du, and J. E. Fowler, "Reconstruction from random projections of hyperspectral imagery with spectral and spatial partitioning," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 466–472, Apr. 2013.

[15] Z. Ye, J. E. Fowler, and L. Bai, "Spatial-spectral hyperspectral classification using local binary patterns and Markov random fields," *J. Appl. Remote Sens.*, vol. 11, no. 3, Jul. 2017.

[16] M. Khodadadzadeh, L. Bruzzone, J. Li, and A. Plaza, "A Gaussian approach to subspace based classification of hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 3278–3281.

[17] H. K. Aggarwal and A. Majumdar, "Sparse filtering based hyperspectral unmixing," in *Proc. Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2016, pp. 2158–2162.

[18] C. Li, S. Li, X. Kang, and T. Lu, "Gabor filtering based deep network for hyperspectral image classification," *Int. Geosci. Remote Sens. Symp.*, 2017, pp. 1808–1811.

[19] K. Wu and S. Jia, "2D Gabor-based sparse representation classification for hyperspectral imagery," in *Proc. 6th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, 2014, pp. 1–4.

[20] Z. Ye, L. Bai, and Y. Nian, "Decision fusion for hyperspectral image classification based on multiple features and locality-preserving analysis," *Eur. J. Remote Sens.*, pp. 166–178, Jan. 2017.

[21] 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.

[22] Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using convolutional neural networks multiple feature learning," *Remote Sens.*, vol. 10, 2018, Art. no. 299.

[23] S. Valero, P. Salembier, and J. Chanussot, "Object recognition in urban hyperspectral images using binary partition tree representation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 4098–4101.

[24] C. Dai, Q. Li, J. Liu, and C. Dai, "Band selection for biomedical hyperspectral data studies using genetic algorithms," in *Proc. 3rd Int. Conf. Bioinf. Biomed. Sci.*, 2009.

[25] L. M. Wickert, J. B. Percival, W. A. Morris, and J. R. Harris, "XRD and infrared spectroscopic validation of weathering surfaces from ultramafic and mafic lithologies examined using hyperspectral imagery, cross lake area, cape smith belt," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Northern Quebec, Canada, 2008.

[26] K. Zhou *et al.*, "Assessment of spectral variation between rice canopy components using spectral feature analysis of near-ground hyperspectral imaging data," in *Proc. Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2016.

[27] S. K. Roy, G. Krishna, R. S. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[28] Z. Zhu, S. Jia, S. He, Y. Sun, Z. Ji, and L. Shen, "Three-dimensional gabor feature extraction for hyperspectral imagery classification using a memetic framework," *Inf. Sci.*, 2014. doi: 10.1016/j.ins.2014.11.045.

[29] F. Wang, R. Zhang, and Q. Wu, "Hyperspectral image classification based on PCA network," in *Proc. 8th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2016, pp. 1–4.

[30] P. Deepa and K. Thilagavathi, "Data reduction techniques of hyperspectral images: A comparative study," in *Proc. 3rd Int. Conf. Signal Process., Commun. Netw.*, 2015

[31] G. Camps-Valls and L. Bruzzone, "Kernelbased methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1–12, Jun. 2005.

[32] V. Vapnik, *Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1999.

[33] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. R. Soc. (A)*, vol. 83, no. 559, pp. 69–70, Nov. 1909.

[34] J. Kleesiek *et al.* "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping," *NeuroImage*, vol. 129, pp. 460–469, Apr. 2016.

[35] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[36] F. Deng, S. Pu, X. Chen, Y. Shi, and T. Y. S. Pu, "Hyperspectral image classification with capsule network using limited training samples," *Sensors*, vol. 18, 2018, Art. no. 3153.

[37] P. Gamba, "A collection of data for urban area characterization," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2004.

[38] B. Praveen and V. Menon, "Novel deep-learning-based spatial-spectral feature extraction for hyperspectral remote sensing applications," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2019.

[39] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Adv. Neural Inf. Process. Syst.*, pp. 3320–3328, 2014.

Bishwas Praveen (Graduate Student Member, IEEE) received the bachelor's degree in computer science from BNM Institute of Technology, Bangalore, India, in 2017, and the master's degree in computer and information sciences from The University of Alabama in Huntsville, Alabama, USA, in 2019. He is currently working toward the Ph.D. degree in computer and information sciences at The University of Alabama in Huntsville.

His research interests include hyperspectral data analysis, deep learning techniques for remote sensing data processing and classification, and interpretable machine learning.

Vineetha Menon (Student Member, IEEE) received the bachelor's degree in electronics and communication engineering from Amrita Vishwa Vidyapeetham, Bangalore, India, in 2010, and the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2016.

She is an Assistant Professor of Computer Science and Director with the Big Data Analytics Lab, University of Alabama in Huntsville, Alabama, USA. Her multidisciplinary research interests include deep learning, remote sensing, computational biology, cybersecurity, explainable artificial intelligence, and interpretable machine learning for human–machine interaction domains.