Deep Hyperspectral Shots: Deep Snap Smooth Wavelet Convolutional Neural Network Shots Ensemble for Hyperspectral Image Classification

Farhan Ullah, Yaqian Long, Irfan Ullah, Rehan Ullah Khan, Salabat Khan, Khalil Khan, Maqbool Khan, *Senior Member, IEEE*, and Giovanni Pau, *Member, IEEE*

***Abstract*—The deployment of convolutional neural networks (CNNs) to classify hyperspectral images is extensively discussed in the research study. A number of different algorithms and approaches are applied, including 2-D CNN, 3-D CNN, support vector machine (SVM), regression models, and other state-of-the-art deep learning models, although these methods do not show good performance for hyperspectral image classification. Furthermore, 3-D CNNs require a lot of computational power and are not mainly employed, whereas 2-D CNNs do not constitute multiresolution image processing and exclusively focus on spatial features. However, 3D–2D CNNs aim to incorporate spectral and spatial features, and their efficiency while being evaluated on various datasets tends to be limited. Moreover, a number of deep learning models have been proposed recently, but their performance is still limited. In order to solve these problems, in this article, we propose a novel deep hyperspectral shot, a deep smooth wavelet CNN shots ensemble for hyperspectral image classification. A deep smooth wavelet CNN utilizes layers of wavelet transform to extract spectral features. The computation of a wavelet transform is less intensive as compared to the computation of a 3-D CNN. After that, the extracted spectral features are integrated into 2-D CNN, which generates spatial features, as a result, generates a spatial–spectral feature vector for classification. Furthermore, we introduce the snapshots generation method and employ a cyclic annealing schedule to converge to several local minima along its optimization path and save the models. We build several snapshots of the deep hyperspectral shots model to enhance the performance of our proposed method. We propose the snapshots optimization and ensemble selection approach in order to solve the optimization problem within ensemble creation and further enhance the performance. In addition, we also introduce a novel activation function called Relish to increase spatial–spectral feature propagation and advance for smoother gradients. Overall, we ensemble the snapshots of our proposed method and achieved that can classify multiresolution HSI data with high accuracy. Experiments performed on benchmark datasets, our proposed method, deep hyperspectral shots, achieved overall accuracies of 99.96%, 97.91%, and 99.49% on the Salinas, Indian Pines, and Pavia University datasets against the state-of-the-art methods.**

***Index Terms*—Deep learning (DL), ensemble, hyperspectral image classification (HSIC), snapshot, wavelet convolutional neural network (CNN).**

I. Introduction

HYPERSPECTRAL images (HSIs) have received a lot of focus in recent years because of the valuable features and high spectral resolution [1], [2]. Using these comprehensive HSI data, HSIs are used effectively in a variety of application domains, some examples of applications include urban development, land monitoring, scene interpretation, and resource exploration [3], [4], [5], [6]. For these applications, HSI classification (HSIC) is a popular approach that provides the opportunity for the analysis of earth feature properties, thus also making it easier for HSI applications.

Over the last several decades, a large number of classification techniques based on spectral and spatial characteristics have been developed to classify the pixels in the HSI dataset; those techniques incorporate support vector machines (SVMs) [7], [8] *K*-nearest neighbors [9], random forest [10], logistic regression [11], extreme learning machine [12], and deep learning (DL) models, etc. Nevertheless, as such techniques consider pixel values with very related spectral data but not the same label, the classification performance is often limited, because the leverage of spatial information is rarely taken into consideration. Following that, a number of techniques have used spatial characteristics as a secondary approach, based on spectral characteristics, to improve the illustration of hyperspectral data. For example, the fine-tuned classification maps generated by SVMs, Markov random field, and edge-preserving filtering are employed to accept spatial contextual data into account [13], [14]. Morphological profiles [15], [16], [17] have been advanced to transform the spatial feature extraction of high-dimensional hyperspectral data, which has been proven to be an effective method to analyze spatial data. In [18] and [19], the spatial data of each pixel's

Manuscript received 18 July 2023; revised 21 August 2023; accepted 2 September 2023. Date of publication 13 September 2023; date of current version 23 November 2023. This work was supported by Kore University of Enna and IEEE mutual agreement. *(Corresponding author: Giovanni Pau.)*

Farhan Ullah, Yaqian Long, and Salabat Khan are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: farhan.marwat@gmail.com; longyaqian@szu.edu.cn; salabatwazir@gmail.com).

Irfan Ullah is with the School of Computer Science, Chengdu University of Technology, Chengdu 610059, China (e-mail: irfan.ee@cqu.edu.cn).

Rehan Ullah Khan is with the Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia (e-mail: re.khan@qu.edu.sa).

Khalil Khan is with the Department of computer science, Nazarbayev University Kazakhstan, Astana 010000, Kazakhstan (e-mail: khalil.khan@fecid.paf-iast.edu.pk).

Maqbool Khan is with the Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur 22620, Pakistan (e-mail: maqbool@163.com).

Giovanni Pau is with the Kore University of Enna, 94100 Enna, Italy (e-mail: giovanni.pau@unikore.it).

Digital Object Identifier 10.1109/JSTARS.2023.3314900

neighborhood are presented to a sparse representation model in order to find the best representation approach using a sequence of standard training instances. Moreover, diverse image processing techniques, for instance, Gabor filtering [20], compressive sensing [21], and discriminant analysis [22], [23], are also employed for HIC with the add-on of spectral and spatial information.

Despite the fact that the aforementioned techniques have yielded appropriate results, the validity of their classification largely rely on the reliability of the hand-crafted features, which are regarded as superficial features. The experiments of the aforesaid shallow models are vulnerable to overfitting problems due to the Hughes phenomenon [24] and the smaller sample size. Furthermore, due to erratic environmental forces, there is commonly severe spectral diversity in HSI, resulting in a massive intraclass distance and intense interclass correlation. Therefore, hand-crafted patterns are not suitable for dealing with these issues. The extraction of dynamic spectral and spatial characteristics for HSIC is a commonly known commercial need in the associated industry.

DL has experienced significant growth with the growing scalability of hardware, DL has experienced significant growth in computer vision (CV) tasks (e.g., image classification [25], [26], [27], scene segmentation [28], target detection [29], and natural language processing (NLP) [30], [31]), etc. A number of DL-based algorithms have been proposed to compute the HSIC datasets, which can be further classified into two types, i.e., spectral-based technique, and spectral–spatial-based technique, based on the method of data processing. The spectral-based techniques exclusively employ spectral data, e.g., to explore the subtle spectral disparities among various classes, a high-dimensionality spectrum was fed into a deep neural network (DNN) [32]. Rather than computing each band separately, a recurrent neural network (RNN) was employed to maximize the utilization of the spectral similarity that occurs in the specific bands [33]. To reduce the computational complexity of overlapping data between adjacent bands and improve classification performance, a cascaded RNN approach with two RNN layers was presented, whereas one intends to minimize concurrency and another to learn complementarity [34]. Gradients could come to a halt throughout the RNN training process due to the trait of long-term reliance. Hence, long short-term memory (LSTM) networks, a novel model of RNN, are introduced to obtain the context-specific spectral features efficiently to address this gap [35], [36]. However, such spectral-based techniques have enhanced classification performance, and their efficiency in complex scenes still needs to be improved.

Unlike the previously mentioned, spectral–spatial-based techniques intend to retrieve both spectral and spatial features for classification performance concurrently. Numerous pieces of research have been performed on this concept so far. In another study the author employed [37], principal component analysis (PCA) to reduce the high-dimensionality of the HSI datasets, then every pixel and the flattened vector of the correlating neighbors were fed into the multilayer stacked autoencoders to retrieve spectral–spatial features. In the course of feature learning, Ma et al. [38] introduced a spatially updated deep autoencoder that considered contextual metadata to achieve maximum interclass distances. Deep belief networks were also used in order to acquire representative spectral features and score neighboring pixels' estimates [39], [40]. Even so, these approaches primarily infuse spatial input data into flat vectors that can suppress spatial structure.

Convolutional neural networks (CNNs) have illustrated the potential of extracting features and influenced the research area of HISC tasks due to the distinctive potential benefits of general perception and parameter tuning. The basic two categories of CNN architectures, such as 1-D CNN and 2-D CNN, were constructed to acquire the robust spectral–spatial features and subsequently successfully obtain the classification result through the decision fusion approaches [41], [42], [43]. Moreover, to acquire robust spectral–spatial features, HSI cubes include the central pixels and its neighbor pixels were chosen as the training instance of a DNN. Due to the sheer intense spectral correlation among the center pixels and the neighboring pixels in a limited area, such a method suggests that the label of the whole HSI cube can be illustrated via the label of the central pixel. According to this supposition, 3-D has been the best suitable DL architecture for extracting spectral–spatial features [43]. Furthermore, the Jeffries–Matusita distance was combined with 3-D CNN to adopt efficient bands for the classification of distinct closely related objects [44]. To tackle the problems of significant hyperparameters and extreme time complexity of the training model, such 3-D convolutional layers at depth levels are replaced via employing 2-D convolutional layers to efficaciously optimize DNNs and integrate features at various levels [45], [46].

The general perspective is that deeper DNN architectures can more efficiently extract higher-level abstract features. While greater depth allows for more non-linear transformations between layers, it also increases the risk of the vanishing gradient problem. To address this, a residual network (ResNet) was introduced as a framework that utilizes shortcut connections between residual blocks. This allows gradients to bypass several layers and propagate back directly to shallower layers, facilitating optimization through stochastic gradient descent [47]. In [48], to alleviate the decreasing-accuracy problem, a deep spectral–spatial ResNet was proposed with serial residual blocks. In [49], Paoletti et al. presented a pyramid bottleneck residual block to incorporate extra feature map locations in the deeper network. In [50], Zhang et al. integrated the spectral–spatial fractal residual CNN. Moreover, to solve the issue of limited labeled data, they integrate a data balance augmentation approach to enhance the accuracy of the limited sample size. Furthermore, a dual-channel residual network with a noise-robust loss function was presented to thoroughly exploit the valuable data from incorrectly labeled samples to improve the DNN model effectiveness in unique scenes [51]. Due to the previously significant research progress, ResNet has become the dominant architectural style of spectral–spatial-based techniques for HSIC.

Nevertheless, there is one typical major downside that has yet to be addressed. HSI generally carries a wide range of spectral and spatial details, although not all of it is useful for classification [52]. Moreover, the spectral bands and salient spatial locations that are useful for feature representation and

classification should be illustrated. In order to achieve this, the attention method has been successfully applied to natural language translation [53] and CV tasks [54], [55] have also been proposed to detect the majority of salient bands and locations in HSIs. Among application areas, the attention method is typically integrated as an individual block in networks to improve feature maps by unevenly weighting bands, pixels, or channels. For example, in the initial phases, lightweight spectral attention blocks constructed of global average pooling layers and convolutional layers were introduced at the top of the networks to enhance the impactful spectral bands to perform the main function in resulting feature extraction [56], [57].

In addition to the spectral attention block, the spatial attention block was intended to increase the importance of the related spatial regions. In [58], Shamsolmoali et al. used spatial attention to improve the network's generalization capacity during feature fusion. In a sequential manner, integrating the spectral and spatial attention blocks into the residual connections, effective spectral–spatial features are calculated to enhance the classification of overall majority outputs [52], [59], [60], [61]. Nevertheless, in the aforementioned techniques, the spectral and spatial attention blocks are generated individually, preventing the fusion of spectral and spatial properties. To enhance the association between spectral and spatial attention, similarity matrices produced by the spectral and spatial attention sections were dynamically assigned to all locations and bands [62]. Li and Wang [63] introduced a spectral and spatial fused attention block to cross-apply attention masks, intending to thoroughly explore the significant association between spectral bands, spatial locations, neighborhoods, and classification outcomes. Furthermore, self-attention (SA) was used to analyze the associations among pixels. An SA methodology was developed in order to retrieve distinguishing spectral and spatial features [64]. Zhao et al. [65] presented an approach to calculate the correlation coefficients between the center pixel and its neighbors to improve the impact of the center pixel. The spectral and spatial transformers [66], [67] were used to predict the correlation between spectral bands and spatial positions by incorporating various SA components. In addition, the transformer was used to acquire the enhanced inputs for computation [68]. Even so, the computational complexity is immense because the transformer typically contains multiple SA modules.

In [69] and [70], the spectral and spatial nonlocal blocks retrieve the global salient spectral bands and spatial areas. Both are integrated with the spectral and spatial blocks, respectively, to optimize the features. Recently, graph CNN (GCNN) [71] has been proposed to extract similar features from the spatial and spectral features of HSI and improve the performance issue of HSIC. Many spatial–spectral proposed hybrid methods [72], [73], [74] have been developed by combining spatial correlations and spectral similarity [75], and by connecting the integrated correlations and spectral similarity of nearby pixels, a spatial–spectral hypergraph was created within the class. Zhang et al. [76] proposed a new multireceptive field-based neural (MARP) method for HSIC. In this method, a novel approach called the adaptive receptive path aggregation mechanism is introduced to mitigate the influence of noise nodes on classification. This mechanism also enables the automatic exploration of an adaptable receptive field. To achieve this, a graph attention (GAT) neural network is utilized to learn the significance of neighborhoods with varying sizes. Additionally, the method involves the integration of an LSTM technique to update nodes and retain the inherent local convolutional features of these nodes. Similarly in [77], Ding et al. proposed a novel multiscale receptive fields GAT neural network (MRGAT) for HSI. In [78], an adaptive graph convolution approach termed (AF2GNN) is introduced for HSIC. It incorporates superpixel segmentation to refine local spatial features and a two-layer 1-D CNN for spectral feature transformation. A linear function combines various graph filters through learned weight matrices. Furthermore, degree-scalers unify multiple filters, outlining the graph structure. The AF2GNN network integrates adaptive filters and aggregator fusion, enabling comprehensive HSIC. In [79], Chen et al. introduced a new approach named spectral-induced aligned superpixel segmentation, which simultaneously employs segmentation outcomes from raw and deeply abstracted spectral features in HSI. Deep spectral features are extracted through a deep autoencoder. Minimum and maximum fusion approaches are intuitively examined to combine these segmentation results. Additionally, they introduce a local aggregation and global attention block, integrating graph sampling, aggregation, and graph transformer to hierarchically examine local and global spatial characteristics. The aforementioned methods typically retrieve spatial features based on coordinate distance, which prevents them from capturing key features. Additionally, the majority of them only take these spatial features into account when preprocessing the HSI data and do not take them into account when building the graph [80], [81]. Furthermore, the aforementioned techniques are built on a solo graph structure to guide the weak diversity learning of intrinsic properties found in HSI data. In order to alleviate the phenomenon, ensemble learning [82], [83] draws much attention. However, the existing ensemble learning algorithms are prone to computational complexity problems.

Although notable significant progress has been made in the research area of spectral–spatial HSIC technique, it remains crucial to improve discriminative features and performance in contexts of accuracy prospects. In this article, we proposed deep hyperspectral shots, which are presented to characterize the spatial and spectral features of HSIC. The proposed method's goal is to be capable to develop discriminative representations for spectral and spatial characteristics, as well as to improve the robustness of features through embedded learning via dense concatenation, snapshots, and ensemble learning. To accomplish the aforementioned task, we build four distinct parts in the proposed deep hyperspectral shots architecture. First, to efficiently compute the feature extraction and reduce the time complexity issue, factor analysis (FA) and wavelet decomposition methods have been employed, which reduces the training and time complexity issue efficiently to exploit the CNN. Second, the excessively dense features emphasize the overfitting and redundancy issues, which reduce classification performance as the number of dense features increases. Recently, the ensemble approach has been extensively applied to a variety of tasks, such as CV and NLP,

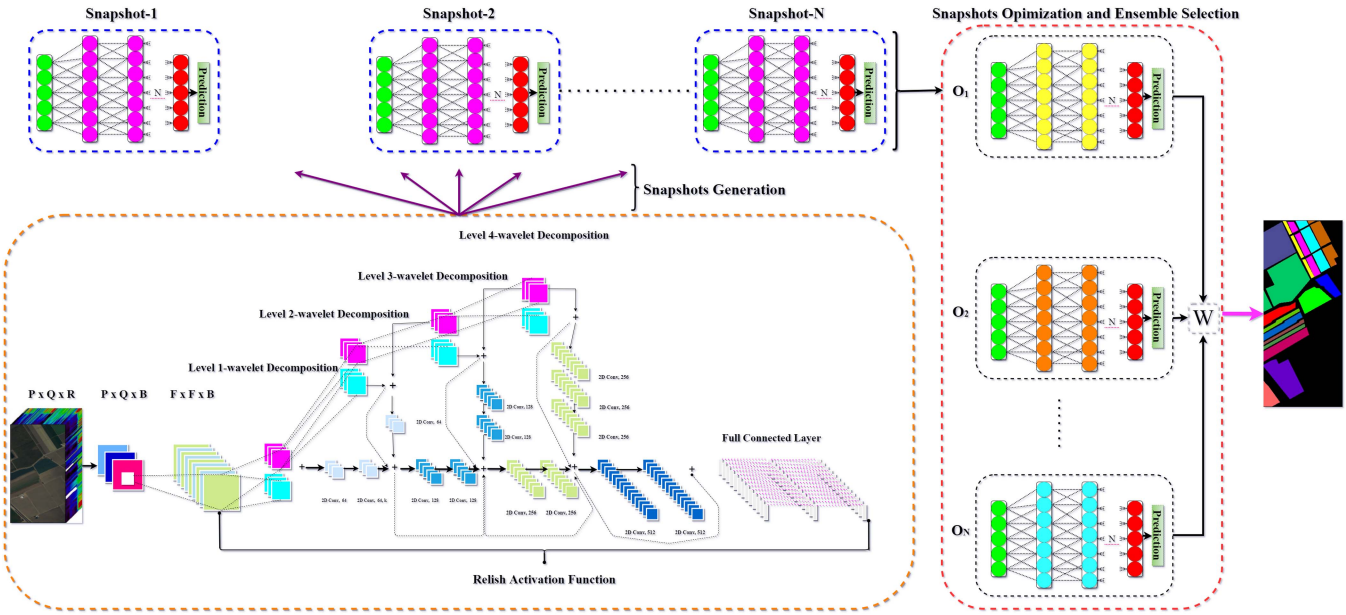


Fig. 1. Deep hyperspectral shots architecture. The orange rectangle box consists of the base model, the blue boxes exhibit the generated snapshots of the base model, the vertical red rectangle box consists of the snapshots optimization and selection process, and the final part is the output of the proposed method. (Please use the zoom-in option to explore the figure.)

[84], [85], which is motivated by the human team task efficiency. Motivated by the fact that the snapshot ensembling can improve representation discrimination and help improve the time complexity generally associated with ensemble approach creation. Moreover, we proposed a snapshot selection technique to optimize the ensemble approach and efficiently compute the spatial and spectral features of HSI. In this article, the 2-D and 3-D spatial and spectral modules are built individually to yield productive feature representation. Third, in order to enhance the performance and low computation complexity problem, we design an improved activation function. Finally, to further extract dense features from the HSI and enhance classification performance on datasets with diverse spatial and spectral characteristics, an approach was developed with the aim of optimizing the representation of data patterns to boost classification accuracy.

This article's key contributions are described as follows.

1) We propose the deep hyperspectral shots method, which extracts the spatial and spectral features efficiently. Moreover, we proposed the snapshot ensemble approach to enhance the performance.
2) We incorporate the novel snapshot optimization and ensemble selection approach to obtain additional accurate classification performance.
3) We also introduce a novel Relish activation function that has been integrated into the proposed method to boost spatial–spectral feature propagation and advance for smoother gradients. This tends to make the landscape simpler to optimize, allowing for smoother and faster generalization of the proposed method.
4) An end-to-end deep hyperspectral method, which incorporates the snapshot, the ensemble, and deep hyperspectral shots, is proposed to stress the relevant spatial areas and extract the discriminating spectral–spatial features for HSIC.
5) A number of experiments were carried out on real-world datasets to evaluate the efficacy of our proposed method and compare it to existing methods. Experimental performance exhibits that our proposed method not only obtains enhanced performance but also can efficiently extract spectral–spatial features.

The rest of this article is organized as follows. Section II introduces the proposed deep hyperspectral shots method in a comprehensive manner. Section III presents the experiment and results analysis of three standard HSIC datasets. Finally, Section IV concludes this article.

II. Proposed Methodology

In this section, the overview of the proposed deep hyperspectral shots architecture is first introduced. Then, the key parts of the framework are described comprehensively. Finally, the activation function of the network and the algorithm's steps are described subsequently.

A. Architecture

As exhibited in Fig. 1, the deep hyperspectral shots architecture is a multisnapshot ensemble feedforward neural network that consists of five explicit functional parts, such as the preprocessing part, the base or core model, snapshots generation, snapshot optimization and ensemble selection, and, finally, the output part. From the forward propagation process, the output of every individual snapshot is utilized as the output of the core model generation. For example, we employ core-model to produce several snapshots required by the deep hyperspectral shots

and final output part. The snapshot optimization and ensemble selection approach is applied for the fusion part to train the acquired spectral and spatial features.

The basic notations of each part in Fig. 1 are described as follows: The orange rectangle box indicates the base%core model, which includes the preprocessing part that converts the original map into low-dimensional features, and the middle part contains all neurons of the deep hyperspectral snapshots and wavelet transform framework. The black line of the core model indicates the activate function infuse through a core model. The purple lines are the snapshots generation process. The blue boxes show the snapshots from $1\ldots, 1+n$, generated by a core model. The red vertical rectangle box is the snapshot optimization and ensemble selection procedure to optimize the snapshots and generate a prediction for the final output of the proposed method. The following operational parts will be thoroughly discussed ahead.

B. Feature Extractions

In order to enhance computational efficacy, the input HSI cube with dimension $P \times Q \times R$ is initially forwarded to an FA technique to minimize the dimension to $P \times Q \times B$. Minimizing the dimension of the original HSI cube helps in decreasing the overall training time complexity by 60% [86]. The output vector $\tilde{Y}$, with a dimension of $1 \times PQ$, selects one of the accessible land cover classes defined by C. The spectral dimensions are saved in FA, i.e., $P \times Q$, only the bands are scaled down from R to B. Using FA as a preprocessing phase in HSI is immensely effective because it can characterize the variations among the distinct correlated and intersecting spectrum bands, allowing the model to effectively classify related instances. However, the frequently utilized PCA-based dimension reduction does not thoroughly focus on this goal in HSI. PCA generates an approximate value for the desired factors, but it does not enable categorizing resembling instances very efficiently. Following the completion of the FA phase, intersecting 3-D patches of size $F \times F \times B$ are retrieved from the preprocessed HSI and issued to the core model of the deep hyperspectral shots. $F \times F$ is the window size for patch extraction, for the Indian Pines dataset, University of Pavia, and Salinas Scene dataset, and the window size has been set at 24×24. The truth scores for the above patches are measured by the class category of the central pixel. The scores were assigned through the experimental work in order to optimize overall accuracy (OA).

C. Deep Smooth Wavelet Convolution Neural Network

The traditional 2-D CNN can be stated as a variant of a multiresolution CNN, which can take spatial and spectral information into account [87]. Past studies have been effective in proposing the convolution and pooling functions in a 2-D CNN as filtering and downsampling [88]. The conventional CNN can be mathematically expressed as the weighted sum of nearest neighbors with the addition of constant bias.

Given an input vector X_n with corresponding labels Y_n from the R^n space. In (1), Y_i is a label from Y_n labels and X_i is the corresponding sample from X_n. W_j is the weight defined by a filtering kernel. N_i are neighboring i data

$$y_i = \sum_{j \in N_i} W_j X_j. \tag{1}$$

Equation (1) can be merely termed as the convolution of X_i and kernel W_j and can be expressed as $Y = X \times W$. This is called the convolution layer of a CNN, where W is in R^o. Convolution layers' output is usually huge and must be pooled down prior to being fed to the following layer. Pooling layers are incorporated in between convolution layers to initiate a filtering process and reduce the total amount of outputs. This forms the foundation for the multiresolution CNN where the convolution is executed by a pair of kernels k_{low} and k_{high}, which generate X_{low} and X_{high}. The multiresolution CNN performs the hierarchical decomposition of the $X_{(\text{low},t)}$ into $X_{(\text{low},t+1)}$ and $X_{(\text{high},t+1)}$ with different kernels at each step t.

For deep hyperspectral shots, we utilized the wavelet transform. The wavelet kernel $K_{\text{high},t}$ is Haar wavelets and $K_{\text{low},t}$ is a scaling function [89]. The two-dimensional Haar wavelets employ the four kernels (fv, vfv, ufu, vfu, u) for the wavelet transform [90] as given in

$$f_{v,v} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} f_{v,u} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$
$$f_{u,v} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} f_{u,u} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{2}$$

After passing patch x with $F \times F$ dimensions via a Haar transform, the (i_v, j_v)th spectrum position value can be expressed as Haar

$$(i_v, j_v) = x(2i_v - 1, 2j_v - 1) + x(2i_v - 1, 2j_v) + x(2i_v, 2j_v - 1) + x(2i_v, 2j_v). \tag{3}$$

The wavelet transform transforms the HSI patch that has been obtained as an input into subbands, which are then forwarded through a convolution layer to learn the spectral and location features. Please note that the subbands designated as high-pass and low-pass filters do not essentially filter the spectral band with these filters. The wavelet transform subdivides the subband portion once more in the following layer before sending it to the convolution layer. This procedure is repeated in each layer, with the CNN continuing to learn spectral and spatial features from the HSI patch.

The base model of the proposed method architecture is shown in Fig. 1. The model is initialized with 3×3 convolution kernels and 1×1 padding. To replace pooling layers in between convolution, a stride of 2 has been employed. In order to avoid the model overfitting issue, a global mean pooling has been used for all convolution layers before passing data into the dense layer. Furthermore, to make better use of the wavelet transformed feature, dense connections and projection shortcuts approach have been adopted [47], [91]. To ensure that all the features stream through to the end of the model, dense connections are used with a channelwise concatenation of the decomposed data. The base model is embedded with two dropout layers as well

along with batch normalization to eliminate the overfitting issue. Due to the small number of samples in HSI datasets, there is a high likelihood of overfitting. It is imperative to take every precaution to keep the base model from the overfitting problem. Relish has been employed as the smooth activation function, the proposed activation function has been explained in detail in Section II-G.

D. Snapshot Generation

From a singular training phase, deep hyperspectral shots generate an ensemble of accurate and vibrant snapshot models. An optimization technique is the core of the deep hyperspectral shot, which explores multiple local minima until settling on optimum minima. In order to achieve the maximum advantage, at the training phase, we choose model snapshots at multiple different minima.

Generally, deep hyperspectral shots perform efficiently if the snapshot models, first, have low test error and, second, do not overlap in the set of examples they misclassify. The weight assignments of a neural network do not always correlate to low test inaccuracy across the optimization curve. In addition, it is frequently observed that the validation loss of the deep hyperspectral shots ensemble gradually declines only after the learning rate is lower, where it occurs after several hundred epochs. During our analysis, the training of our proposed method snapshots ensemble for very lesser epochs, and dropping the learning rate earlier has a slight effect on the end test error [92]. This appears to imply that local minima across the optimization curve indeed be convincing (in aspects of prediction error) only after a couple of fewer epochs.

E. Schedule Selection

We employ a cyclic annealing schedule (CAS) to converge to numerous local minima as in [92]. We minimize the learning rate rapidly, letting the model converge to its initial local minima in as little as 50 epochs. The optimization process is then repeated at a slightly large learning rate, perturbing the model and causing it to escape from the minimum. This procedure is reiterated numerous times in order to achieve various convergences. The learning rate is expressed in mathematical terms as

$$\alpha(\hat{\mathrm{r}}) = \mathfrak{f}(\mathrm{mod}(\hat{\mathrm{r}} - 1, \lceil N/N_i \rceil)) \tag{4}$$

where $\hat{\mathrm{r}}$ denotes the number of iterations, N represents the cumulative number of training iterations, and $\mathfrak{f}$ denotes a constant declining function. Specifically, we partition the training phase into M cycles, each of which begins with a high learning rate and gradually reduces to a lower rate. The high learning rate $\alpha = \mathfrak{f}(0)$ allows the model to detach from a critical point, whereas the low learning rate $\alpha = \mathfrak{f}(\lceil N/N_i \rceil)$ leads to an effective local minimum. In our experimental studies, we initialize $\mathfrak{f}$ to be the shifted cosine function:

$$\alpha(\hat{\mathrm{r}}) = \frac{\alpha_0}{2}\left(\cos\left(\frac{\pi \bmod (\hat{\mathrm{r}} - 1, \lceil N/N_i \rceil)}{\lceil N/N_i \rceil}\right) + 1\right) \tag{5}$$

Algorithm 1: Snapshots Generation and Schedule Selection.

Require: **Z**: HSIs data; **N**: total number of classes; **M**: total number of training samples; **L**: Snapshot; **SCB**: Snapshot Callback; **epc**: epoch per cycle; n_epoch: total number of epochs of each cycle.

```
Procedure
Apply FA to execute the dimensionality reduction process for N of HSI datasets Z
Obtain the training samples M by arbitrary selection N pixels from the spectral–spatial dataset Z, generate the testing samples MT by the remaining pixels
Retrieve the spatial features from the HSIs datasets Z and acquire the spectral–spatial fusion information by using the wavelet transform method
for CallBack = SCB do:
    initialize L
    Execute above Equations (4), and (5)
    after each iteration n_epoch save learning rate
    if epoch ! = 0 & (epoch+1) % epc == 0: do:
        Generate and Save train L model.
    end if
end for
```

where α_0 is the initial learning rate. Implicitly, this function anneals the learning rate from its initial value α_0 to $f(\lceil N/N_i \rceil) \approx 0$ over the duration of a cycle. However after that, instead of updating the learning rate at each epoch, we update it at each iteration. This helps to enhance the convergence of short cycles even since a high early learning rate is chosen.

The proposed deep hyperspectral shots model appears to approach a local minimum in terms of training loss at the end of each training cycle. As a result, leading up to increasing the learning rate, we start taking a "snapshot" of the proposed model weights. After training *M* cycles, we have *M* model snapshots, $\emptyset_1, ..., \emptyset_M$, sequentially every snapshot will be incorporated into the final ensemble. It is worth noting that the entire training time of the *M* snapshots is constant as training a model on a typical schedule. In certain situations, the basic learning rate schedule generates lower training loss as compared to the cyclic schedule. The algorithmic details of the proposed method are given in Algorithm 1.

F. Snapshots Optimization and Ensemble Selection

The snapshot selection plays a key role in the construction of a robust and enhanced performance ensemble in our proposed method. Generally, the basic approach of ensemble creation is to integrate the total number of base models by using the bagging [93], boosting [94], and stacking [95] approaches. Although our proposed method can acquire fairly good performance, we take further steps to optimize and minimize the loss in order to acquire the robust and enhanced generalized performance of the ensemble. Since we are considering a nonlinear programming (NLP) problem, we employ the sequential least square programming approach to solve this issue. Optimizing O, the snapshots,

and selecting the best ensemble to allow us to further slightly enhance the performance by using the equations as

$$\begin{aligned} &\min f(q), \text{ over } q \in R^n \\ &\text{subject to } k(q) = 0 \\ &\qquad l(q) \leq 0 \end{aligned} \tag{6}$$

where $f : R^n \to R$ is an objective function, functions $k : R^n \to R^m$ and $l : R^n \to R^z$ describe the equality and inequality constraints, n denotes the total number of variables that are subject to optimization, m is the number of equality, and z is the number of inequality constraints function. For an assigned iterative process $q[p]$, the NLP is scaled down to the QP subproblems, and the following iteration $q[p+1]$ is then derived from the QP solution. The sequence $q[p]$ is designed in such a format that it eventually converges to a local minima $f(q^*)$ of the NLP as $q \to \infty$, where $*$ relates to the parameter associated with the local minimum point. The evaluation and execution of NLP become further challenging in the existence of constraints. The values that fulfill equality and inequality constraints are referred to as the NLPs viable set and are described by

$$\mathcal{F} = \{q \in R^n \mid k(q) = 0, g(q) \leq 0\}. \tag{7}$$

The Lagrangian for this problem is

$$\mathcal{L}(q, \lambda, \mu) = f(q) + \lambda^T k(q) + \mu^T l(q). \tag{8}$$

where vectors $\lambda \in R^m$ and $\mu \in R^z$ are referred to as Lagrangian multipliers. Active constraints are an index set for $q \in R^n$ when

$$\mathcal{I}_{\mathrm{a\,c}} = \{i \in \{1, \ldots, z\} \mid l_i(q) = 0\}. \tag{9}$$

Let $f(q^*)$ be a local minimum of the NLP, then the condition

$$\begin{aligned} l_i(q^*)\, \mu_i^* = 0, \quad 1 \leq i \leq z \\ \mu_i^* > 0, \quad i \in \mathcal{I}_{\mathrm{a\,c}} \end{aligned} \tag{10}$$

is called strict complementary slackness at q^*. Matrix L is described with the derivative of active constraints as follows:

$$L(q) = \left(\nabla k_1(q), \ldots, \nabla k_m(q), \nabla l_{i_1}(q), \ldots, \nabla l_{i_j}(q)\right) \tag{11}$$

where ∇ represents the gradient operator and j represents the number of active constraints. For $f(q^*)$, which is a local minimum of the NLP assumed existing of λ^* and μ^* such that

$$\begin{aligned} \nabla \mathcal{L}(q^*, \lambda^*, \mu^*) = \nabla f(q^*) + \nabla k(q^*)\, \lambda^* \\ + \nabla l(q^*)\, \mu^* = 0. \end{aligned} \tag{12}$$

If (16) carries true, hence the condition is known as the first-order necessary optimality condition. Suppose the following criteria are encountered.

1) The columns of $L(q^*)$ are linearly independent.
2) Strict complementarity slackness holds at q^*.
3) The Hessian of the Lagrangian with respect to q is positive definite on the null space of $L(q^*)^T$.

Thereafter, (6) and these prerequisites are referred to as the second-order enough optimality conditions of the NLP, and the Lagrangian multipliers λ and μ^* are distinctively defined.

After describing the constraints that must be fulfilled in order to take some value as a minimum, the QP for each iteration step is requisite to be constructed. The QP subproblems represent the NLPs local properties with regard to the existing iteration $p[k]$. As a result, the objective function f is substituted with its local quadratic probability distribution

$$\begin{aligned} f(p) \approx f(q[p]) + \nabla f(q[p])(q - q[p]) \\ + \frac{1}{2}(q - q[p]) K f(q[p])(q - q[p]) \end{aligned} \tag{13}$$

and constraint functions l and k are replaced with their local affine approximations

$$l(q) \approx l(q[p]) + \nabla l(q[p])(q - q[p]) \tag{14}$$

$$(q) \approx k(q[p]) + \nabla k(q[p])(q - q[p]) \tag{15}$$

where $\nabla f(q)$ represents the gradient of function and $Kf(q)$ Hessian of function. For optimization purposes, we assign the bounds parameter between 0 and 1. As shown in Fig. 2, in the initial iteration, it takes the total number of Snapshots *T*, and we reiterate *T* to the total number of 25 times $(O_1, ..., O_N)$ to acquire the local optimum and generate the best ensemble selection.

G. Activation Function

Activation functions have been a main field of research in order to fully understand the training complexity of neural networks. While activation functions have played a significant part, in practical application paradigms, they have been regarded as a sole element, whereas in theoretical areas such as mean-field theory, they have been regarded as a primary component; the discussion over activation functions is becoming more volatile. Whereas ReLU and Leaky ReLU have transformed the area due to their ease of formulation and low computational complexity, several studies have presented smoother versions to enhance optimization and information propagation, such as ELU, Softplus, and Swish [96]. However, they have largely been one-hit adventures, failing to adopt versatile datasets. In this article, in order to tackle the versatile spatial–spectral feature datasets, we propose a novel activation function called Relish. The graphical implementation of the Relish activation function has been shown in Fig. 3. The details of the equation of the proposed activation function are given ahead. To recap, Relish is defined as

$$Y = X \times \max(0, \beta x) \tag{16}$$

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \tag{17}$$

where $f(x) = \max(0, x)$ is the ReLU activation function and β is either a constant or a trainable parameter. Fig. 3 shows the graph of Relish for the value of $\beta = 0.1$. As seen in Fig. 3, it has a few key subtly different characteristics that distinguish it from other activation functions. First, Relish is a smooth continuous function like Swish. Since Relish threshold values all negative weights to zero, whereas Swish enables a few negative weights to pass through. This Relish characteristic is critical to

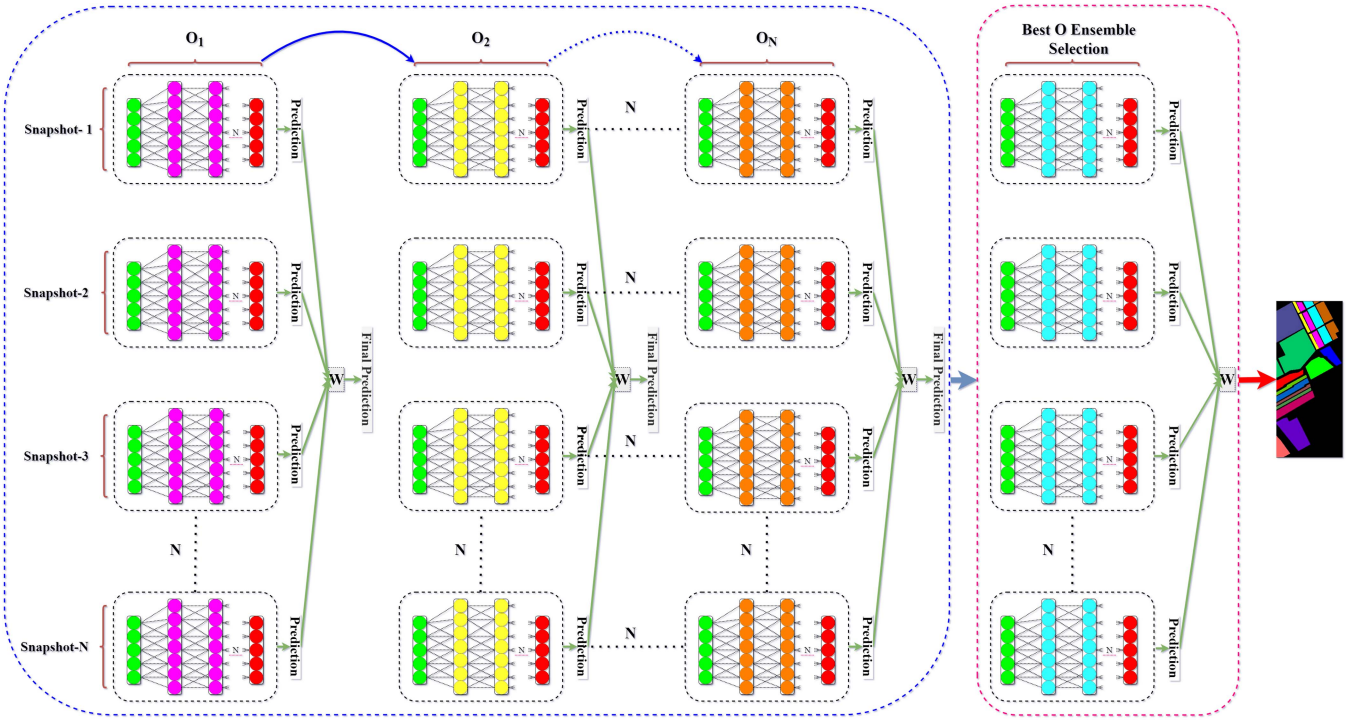


Fig. 2. Snapshots optimization and ensemble selection process. The blue box shows the optimization of T snapshots and the red box shows the best ensemble selection and prediction.

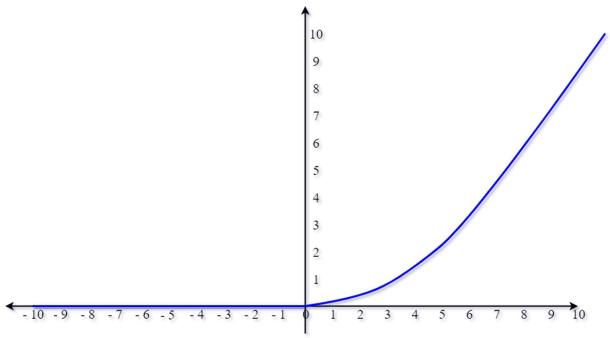


Fig. 3. Plot of the proposed Relish activation function.

the effectiveness of nonmonotonic smooth activation functions when employed in our deep hyperspectral shots method to classify spatial–spectral features. Finally, the trainable parameter ensures finer tuning of the activation function in order to boost spatial–spectral feature propagation and advance for smoother gradients, this tends to make the landscape simpler to optimize, allowing for smoother and faster generalization of the proposed method. The complete algorithmic description of our proposed deep hyperspectral shots is expressed in Algorithm 2.

III. Experiment and Result Analysis

In this section, the comprehensive description of three different HSI datasets acquired by using various imaging technologies that include Indian Pines (IP), University of Pavia (PU), and Salinas (SA),[1] the experimental configuration is first thoroughly stated. Then, the hyperparameters setting of the method, the results, the discussion, and the comparison between the proposed and the state-of-the-art methods are investigated and discussed. Furthermore, the experimental depiction of the different hyperparameter and analyses are illustrated and evaluated.

A. Data Description

In this section, we employed three various types of widely employed datasets to investigate the effectiveness of our proposed method. The source images and ground truths of three various hyperspectral datasets are illustrated in Figs. 4–6.

1) The first dataset SA was captured by the AVIRIS sensor through the geographical area of Salinas Valley, CA, USA, with a spatial resolution of 3.7 m/pixel. The Salinas scene is made up of 224 spectral bands composed of the wavelength range of 360 to 2500 nm and 512 × 217 pixels, as shown in Table I. The dataset includes 16 classes concerning vegetables, vineyard fields, and bare soil. Table II depicts the number of classes and includes training and test data of this dataset.
2) The second dataset PU was collected by an unmanned aerial optical device, i.e., the Reflective Optics Spectrographic Imaging System over Pavia, Italy, in 2002 (ROSIS). The aircraft was managed by the German Aerospace Center as part of the HySens venture, which was funded by

[1] http://lesun.weebly.com/hyperspectral-data-set.html

Algorithm 2: Deep Hyperspectral Shots.

Require: **Z**: HSIs data; **N**: total number of classes; **M**: total number of training samples; **L**: snapshot; **SCB**: snapshot callback; **epc**: epoch per cycle; n_epoch: total number of epochs of each cycle; **T**: total number of snapshots; **V**: number of snapshots required; y_v: prediction of snapshots; $E = \phi$: an ensemble.

Procedure
Apply FA to execute the dimensionality reduction process for **N** of HSI datasets **Z**
Obtain the training samples **M** by arbitrary selection **N** pixels from the spectral–spatial dataset **Z**, generate the testing samples **MT** by the remaining pixels
Retrieve the spatial features from the HSIs datasets **Z** and acquire the spectral–spatial fusion information by using the wavelet transform method.
Training Phase:
  for CallBack = **SCB do:**
    initialize **L**
    Execute (4) and (5)
    After each iteration n_epoch save learning rate
    if epoch ! = 0 & (epoch+1) % **epc** == 0: **do:**
      Generate and Save trained **L** model.
    end if
    if T = **V :**
      Execute **T** to optimize
    end if
    Testing Phase:
      Generate prediction y_v by feeding the testing **MT** into the trained **T** model and E
end for
  Ensemble **T**: $E = T_1, T_2, ..., T_n$
  Final prediction: $y^* = \arg\max \sum_{t=1}^{T} 1(y_v = g), g = 1, 2, ..., g$

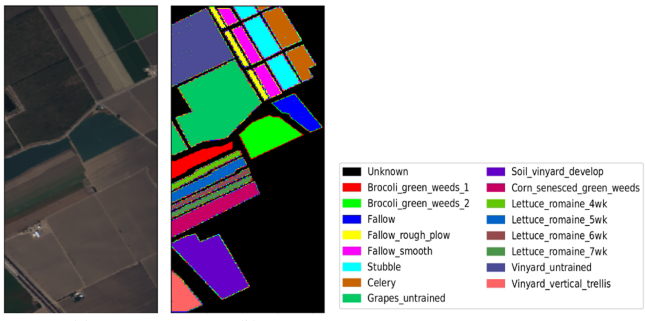


Fig. 4. SA dataset. (a) Source image. (b) Ground truth.

the European Union. After discarding 12 noisy channels, the dataset mainly comprises 640 × 340 pixels with a 1.3 m/pixel spatial resolution and 103 bands containing the wavelength range of 430–860 nm, as shown in Table I. Moreover, the whole dataset is comprised of nine classes.

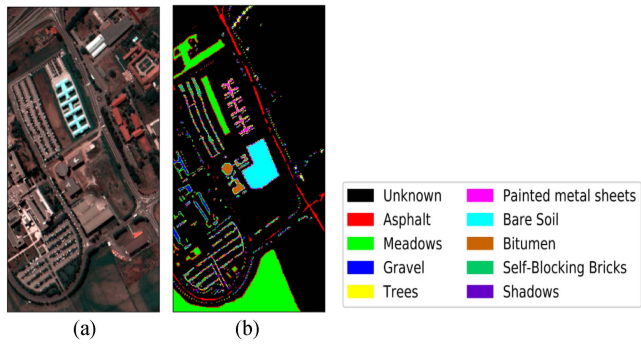


Fig. 5. PU dataset. (a) Source image. (b) Ground truth.

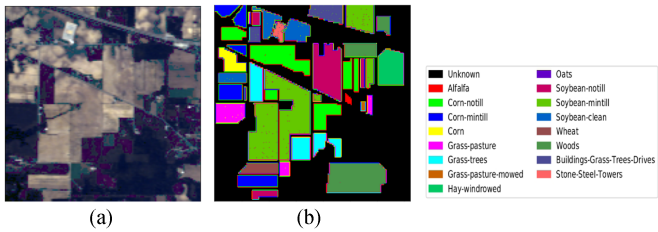


Fig. 6. IP dataset. (a) Source image. (b) Ground truth.

TABLE I
DETAILED DESCRIPTION OF EACH DATASET USED DURING EXPERIMENT

Name	Spatial Dimension	Spectral Bands	Wavelength Range	Classes
SA	512×217	224	360 – 2500 nm	16
UP	610×340	103	430 – 860 nm	9
IP	145×145	224	400 – 2500 nm	16

TABLE II
AMOUNTS OF TRAINING AND TEST DATA OF THE SALINAS DATASET

Class No.	Class Name	Training	Test
1	Brocoli_green_weeds_1	201	1808
2	Brocoli_green_weeds_2	373	3353
3	Fallow	198	1778
4	Fallow_rough_plow	140	1254
5	Fallow_smooth	268	2410
6	Stubble	396	3563
7	Celery	358	3221
8	Grapes_untrained	1128	10 143
9	Soil_vinyard_develop	621	5582
10	Corn_senesced_green_weeds	328	2950
11	Lettuce_romaine_4wk	107	961
12	Lettuce_romaine_5wk	193	1734
13	Lettuce_romaine_6wk	92	824
14	Lettuce_romaine_7wk	107	963
15	Vinyard_untrained	727	6 541
16	Vinyard_vertical_trellis	181	1626
	Total	**5418**	**48 711**

The bold face values shows the higher values in the comparative analysis.

TABLE III
AMOUNTS OF TRAINING AND TEST DATA OF THE PAVIA UNIVERSITY DATASET

Class No.	Class Name	Training	Test
1	Asphalt	664	5967
2	Medows	1865	16 784
3	Gravel	210	1889
4	Tress	307	2757
5	Painted Metal sheets	135	1210
6	Self-Blocking Bare soil	503	4526
7	Bitumen	133	1197
8	Bricks	369	3313
9	Shadows	664	5967
	Total	**4850**	**43 610**

The bold face values shows the higher values in the comparative analysis.

TABLE IV
AMOUNTS OF TRAINING AND TEST DATA OF THE INDIAN PINES SCENE DATASET

Class No.	Class Name	Training	Test
1	Alfalfa	5	41
2	Corn-notill	143	1285
3	Corn-mintill	83	747
4	Corn	24	213
5	Grass-pasture	49	434
6	Grass-trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	Soybean-notill	98	874
11	Soybean-mintill	246	2209
12	Soybean-clean	60	533
13	Wheat	21	184
14	Woods	127	1138
15	Buildings-grass-trees	39	347
16	Stone-steel-towers	10	83
	Total	**1031**	**9218**

The bold face values shows the higher values in the comparative analysis.

The comprehensive details of different classes are given in Table III.

3) The third dataset is the Purdue Indiana Indian Pines scene IP, which was acquired from the Indian Pines testing site in North-Western Indiana. The data resolution is 145 × 145, and it incorporates 200 spectral bands. It covers 145 × 145 pixels with a 20-m/pixel spatial resolution and 224 spectral bands comprised from 400 to 2500 nm. The ground truth appends 16 classes of interest, which are largely diverse crops in various growth stages as given in Tables I and IV.

B. Experimental Configuration

To investigate the efficiencies of the proposed HSIC model, a number of the-state-of-the-art methods were compared with DHS, including SVM-RBF,[2] CCF-200,[3] 2-D CNN [45], GCNN [71], FADCNN [97], NL-GCNN [98], DWR-DEL [99], and E-CNN-PPF-SSF [100]. Moreover, we also compare the proposed activation function with other techniques on the three datasets. We set the total number of epochs to 500, number of depths of CNN and Wavelet and Wavelet decomposition are considered 4, the learning rate of activation function is set to 0.1, the number of components of FA is set to 75, and the standard test ratio is considered 0.9, with the window size 24. The total number of ensemble size is set to 10 and n cycles per epoch is fixed at 50.

The following evaluation metrics are used to quantitatively compare various methods for hyperspectral data classification computations from various perspectives. Detailed information on each evaluation metric is given ahead.

1) *OA:* To evaluate the performance of HSIC methods the first standard evaluation metric is OA. The metric is calculated as the fraction of test samples that are differentiated correctly.
2) *Average Accuracy (AA):* In order to assess the performance of the method with respect to each class in the dataset, we compute per-class access performance with respect to each category in a dataset, and we also compute per-class accuracy. This metric is calculated as the average of all per-class accuracies.
3) *Kappa Coefficient (K):* The most common evaluation metric used in the HSIC is K; this statistical evaluation metric is an aspect of the declaration of validity standard measure.

C. Performance Comparison of Methods

1) *SVM-RBF:* An SVM method with the extensively utilized radial basis function (RBF) kernel. We employ fivefold cross-validation to find the optimum hyperparameters and C (maintaining the sensitivity of penalization throughout model optimization) in the range of $y = 2^{-3}, 2^{-2}, ..., 2^4$ and $C = 10^{-2}, 10^{-1}, ..., 10^4$.
2) *CCF-200:* This technique is a traditional spectral–spatial HSIC method that, for the first time, employs EPF to design feature representations of HSIs.
3) *2-D CNN:* This method used iterative feedback to initiate HSIC, capturing and integrating spatial contextual information to enhance classification accuracy. The spectral signature of the particular class analyzed with the label, employed in the Gaussian filter, is set to 0.5 with a window size of 5, and the stopping threshold is 0.99 in this method.
4) *GCNN:* It generated a sparse feature map for HSI data incorporating numerous features extracted using various spatial-based filters. In order to implement the replica, we use four types of features with a sparsity level of three in our method.
5) *FADCNN:* The bidirectional RNN is constructed by training in both positive and negative time directions at the same time. Instead of using spectrum data as a training instance, we used spatial patches with a size of 13 × 13 in

[2]https://www.csie.ntu.edu.tw/ cjlin/libsvm/
[3]https://github.com/twgr/ccfs

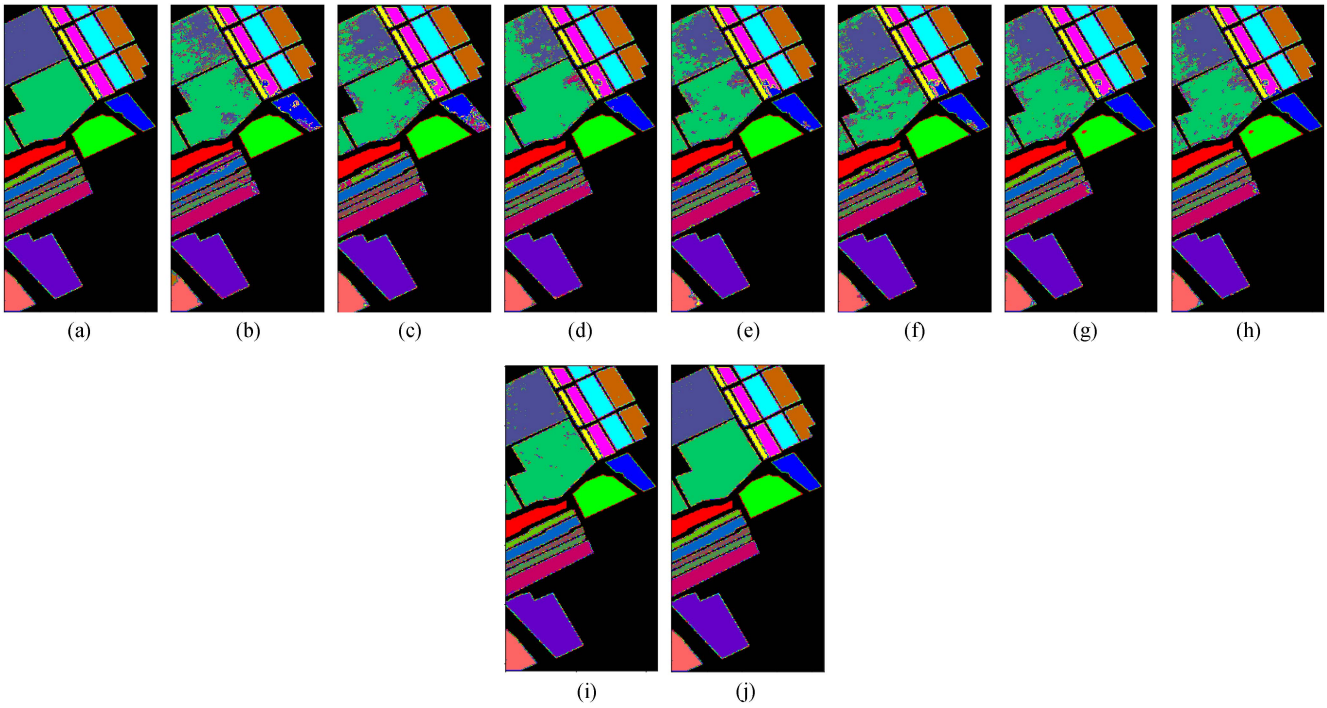


Fig. 7. Classification maps of various methods for the Salinas dataset. (From Left to Right) (a) Groundtruth. (b) SVM-RBF. (c) CCF-200. (d) 2-D CNN. (e) GCNN. (f) FADCNN. (g) NL-GCNN. (h) DWR-DEL. (i) E-CNN-PPF-SSF. (j) DHS.

the experiment. In our deployment, there are three hidden layers, each with 60 nodes. The other set contains a batch size of 20, 5000 iterations, and a 1e-4 learning rate.

6) *NL-GCNN:* This method consists of a baseline 34-layer ResNets for HSIC. The applied settings of each residual block are used with the following hyperparameters, set the total number of epochs to 5000, and the learning rate of 1e-4.
7) *DWR-DEL:* This method consists of two dynamic ensemble learning techniques using local weighted residual (LWR-DEL) and double-weighted residual (DWR-DEL) for HSIC. The range of regularization parameters λ in the classification models is fixed as (1e-2, 1e-4, 1e-6). The framework consists of 15 core classification models.
8) *E-CNN-PPF-SSF:* This method used a deep ensemble CNN based on sample expansion for HSIC. In order to implement the method, we set the total number of epochs to 900 and the learning rate decays every 300 epochs with a decay rate of 0.01, and the ensemble total size is set to 5.
9) *DHS:* This is our proposed method; the proposed method is a multisnapshot ensemble feedforward neural network that consists of five explicit functional parts, such as the preprocessing part, the base or core model, snapshots generation, snapshot optimization, and ensemble selection, and finally the output part. The details of hyperparameters are given in Section III-B.

We employ OA, AA, and K as objective measures to analyze the effectiveness of all the comparison methodologies in the subsequent experiments. Each network implementation has been reiterated five times, and the OA calculation has been described. All of the approaches were built on the TensorFlow framework, with the running environment consisting of an NVidia P100 Tesla with 16 GB, and 128 GB RAM. The batch size is set to 1024, and the proposed networks are trained for 500 Epochs with a 0.1 learning rate. The cyclic annealing algorithm is chosen as an optimizer, with the HSI data dropout rate set to 0.4, and the training sample size is 64×64. Furthermore, Tables II– IV mentioned the total number of training and testing sets samples for the three HSI datasets.

The first experiment is being carried out on the Salinas dataset. After the FA block, the data size is converted to $64 \times 64 \times 3$ and the training ratio is assigned to 10% similarly to the previous two datasets. The ground truth and predicted classification maps are shown in Fig. 7, and classification results based on the OAs metric are depicted in Table V. We can observe that OAs produced by our proposed method are higher with a score of 99.96% comparably with the state-of-the-art methods E-CNN-PPF-SSF with a score of 99.18%, DWR-DEL with a score of 93.27%, NL-GCNN with a score of 92.28%, FADCNN with a score of 90.58%, GCNN with a score of 90.37%, and also against the conventional methods 2D-CNN, CCF-200 and SVM-RBF with a score of 90.25%, 82.87%, and 74.24%, respectively.

Similarly, the proposed DHS shows higher AA scores with 99.97% against the state-of-the-art and traditional methods with scores of 99.58%, 96.82%, 92.28%, 97.80%, 97.80%, 85.43%, 85.43%, 90.25%, 82.87%, and 74.24%, respectively, as illustrated in Table V. Moreover, we also compared the results of our proposed method DHS-based K, as we can see, DHS has a

TABLE V
CLASSIFICATION ACCURACIES OF PROPOSED DEEP HYPERSPECTRAL SHOTS IN TERMS OF OA, KAPPA, AND AA AGAINST THE STATE-OF-THE-ART METHODS SALINAS DATASET

No.	Class Names	SVM-RBF	CCF-200	2-D CNN	GCNN	FADCNN	NL-GCNN	DWR-DEL	E-CNN-PPF-SSF	DHS
1	Brocoli_green_weeds_1	98.98	99.49	71.57	99.59	88.26	99.69	99.61	**100**	**100**
2	Brocoli_green_weeds_2	99.67	99.95	99.86	98.07	98.04	99.21	99.45	**100**	**100**
3	Fallow	98.70	99.43	88.89	91.95	98.04	99.79	99.59	**100**	**100**
4	Fallow_rough_plow	97.77	99.33	98.14	97.84	97.13	98.29	99.01	**100**	**100**
5	Fallow_smooth	98.33	98.82	98.17	98.06	99.13	99.28	99.35	**100**	99.97
6	Stubble	99.72	99.80	**100**	99.00	99.10	99.80	99.21	**100**	**100**
7	Celery	99.46	99.66	97.00	99.29	99.21	99.04	99.23	**100**	99.97
8	Grapes_untrained	70.37	67.56	70.79	82.25	75.72	79.11	84.12	98.21	**99.98**
9	Soil_vinyard_develop	98.59	99.19	99.45	97.11	**100**	97.74	97.10	**100**	**100**
10	Corn_senesced_green_weeds	93.74	93.80	96.19	91.60	79.00	95.01	94.91	99.33	**99.93**
11	Lettuce_romaine_4wk	94.70	95.87	96.37	90.77	95.3	94.60	96.01	**100**	**100**
12	Lettuce_romaine_5wk	99.89	99.95	**100**	**100**	98.34	**100**	**100**	**100**	**100**
13	Lettuce_romaine_6wk	97.81	98.15	**100**	98.96	95.12	98.96	98.67	**100**	**100**
14	Lettuce_romaine_7wk	97.35	96.86	98.33	97.35	98.86	99.41	99.54	99.37	**100**
15	Vinyard_untrained	71.53	80.77	91.22	70.44	99.23	84.26	85.11	97.23	**99.80**
16	Vinyard_vertical_trellis	98.18	98.18	93.00	97.10	83.53	98.01	98.21	**100**	**100**
OA	-	88.82	89.72	90.25	90.37	90.58	92.28	93.27	99.18	**99.96**
AA	-	94.67	95.43	93.69	94.34	94.56	96.39	96.82	99.58	**99.97**
K	-	87.57	88.58	89.18	89.28	90.25	89.64	91.97	98.17	**99.95**

The bold face values shows the higher values in the comparative analysis.

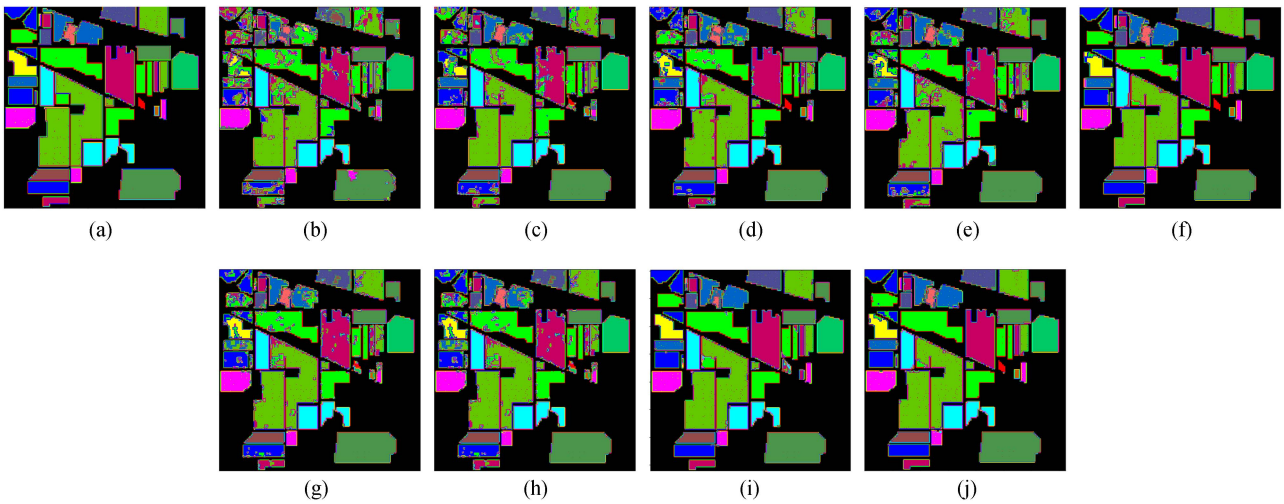


Fig. 8. Classification maps of various methods for the Indian Pines dataset. (From Left to Right and from top to bottom) (a) Groundtruth. (b) SVM-RBF. (c) CCF-200. (d) 2-D CNN. (e) GCNN. (f) FADCNN. (g) NL-GCNN. (h) DWR-DEL. (i) E-CNN-PPF-SSF. (j) DHS.

comparatively robust and well-balanced classification impact for each class and produces the highest scores and outperforms the state-of-the-art and traditional methods with scores of 99.95%.

In the second experiment Purdue Indiana Indian Pines scene, 10% of the training samples were randomly chosen, and the input data dimensions after FA are $64 \times 64 \times 3$. The classification result is based on the IP dataset of AA, OA, and K; a comparison with the state-of-the-art methods is shown in Table IV. The ground truth and predicted maps are illustrated in Fig. 8. As demonstrated in Table IV, the DHS obtained the highest OA of 97.91%. It can be observed that the proposed model has a significant impact on the HSI dataset performance. Furthermore, we can observe that DHS outperforms E-CNN-PPF-SSF at 97.91%, DWR-DEL at 89.13%, NL-GCNN at 87.92%, FADCNN at 97.80%, and GCNN at 85.43% indicating that the DHS approach is useful and necessary for the HSIC.

Based on the AA evaluation metric, the proposed DHS 97.14% method also performed well against the state-of-the-art methods with lower scores of E-CNN-PPF-SSF 97.14%, DWR-DEL 93.81%, NL-GCNN 93.79%, FADCNN 93.96%, and GCNN 91.87%, it shows that DHS can effectively classify the HIC dataset. The experimental results of all methods are shown in Table VI.

The final experiment is being carried out on the PU dataset to validate the performance of our proposed method. The training sample percentage of the PU is set to 10%, we kept the sample size of the PU dataset after FA similar to IP $64 \times 64 \times 3$. Fig. 9 shows the ground truth and predicted classification maps of the proposed method and other methods.

As illustrated in Table VII, it can be observed that the DHS method acquired the leading OA of 99.49%, which is higher than E-CNN-PPF-SSF at 99.10%, DWR-DEL at 98.41%, NL-GCNN at 90.04%, GCNN at 87.08%, and slightly higher than FADCNN at 99.19%. In contrast, the other methods only produce lower scores of OA. The main factor is that the PU dataset has a comparatively sparse feature density, so the conventional

TABLE VI
CLASSIFICATION ACCURACIES OF PROPOSED DEEP HYPERSPECTRAL SHOTS IN TERMS OF OA, KAPPA, AND AA AGAINST THE STATE-OF-THE-ART METHODS THE INDIAN PINES SCENE DATASET

No.	Class Names	SVM-RBF	CCF-200	2-D CNN	GCNN	FADCNN	NL-GCNN	DWR-DEL	E-CNN-PPF-SSF	DHS
1	Alfalfa	71.39	76.37	54.77	76.66	88.26	83.09	95.45	91.35	**98.01**
2	Corn-notill	71.05	77.93	96.94	86.10	**98.04**	89.03	72.19	95.50	95.50
3	Corn-mintill	86.96	94.57	99.46	**100**	97.04	**100**	84.45	97.51	95.02
4	Corn	91.72	94.41	96.87	93.06	96.03	93.51	81.41	97.32	**97.32**
5	Grass-pasture	85.80	91.39	94.12	92.06	**99.34**	94.12	88.29	98.10	98.10
6	Grass-trees	93.85	97.04	96.81	96.81	99.48	98.18	99.42	99.53	**99.53**
7	Grass-pasture-mowed	75.38	90.96	91.29	88.24	75.72	88.24	**100**	**100**	**100**
8	Hay-windrowed	59.88	69.48	93.05	76.80	**100**	78.78	**100**	**100**	**100**
9	Oats	76.24	89.01	87.59	80.85	78.00	86.70	**100**	92.30	**92.30**
10	Soybean-notill	96.91	98.77	**100**	99.38	96.50	99.38	86.80	96.12	96.12
11	Soybean-mintill	79.58	93.73	68.57	93.89	98.49	94.94	92.46	98.64	**98.64**
12	Soybean-clean	74.84	74.55	88.48	93.64	95.45	**97.27**	79.46	94.56	94.56
13	Wheat	97.78	**100**	**100**	**100**	99.02	**100**	**100**	**100**	**100**
14	Woods	79.49	97.44	82.05	92.31	99.48	97.44	99.43	99.78	**99.78**
15	Buildings-Grass-Trees-Drives	**100**	90.91	**100**	**100**	99.74	**100**	94.81	98.70	98.70
16	Stone-Steel-Towers	**100**	**100**	**100**	**100**	82.80	**100**	93.22	90.60	90.60
OA	-	74.24	82.87	90.25	85.43	97.80	87.92	89.13	97.91	**97.91**
AA	-	83.80	89.78	93.69	91.87	93.96	93.79	93.81	97.14	**97.14**
K	-	70.93	80.59	89.18	83.42	86.10	86.25	88.18	96.26	**96.26**

The bold face values shows the higher values in the comparative analysis.

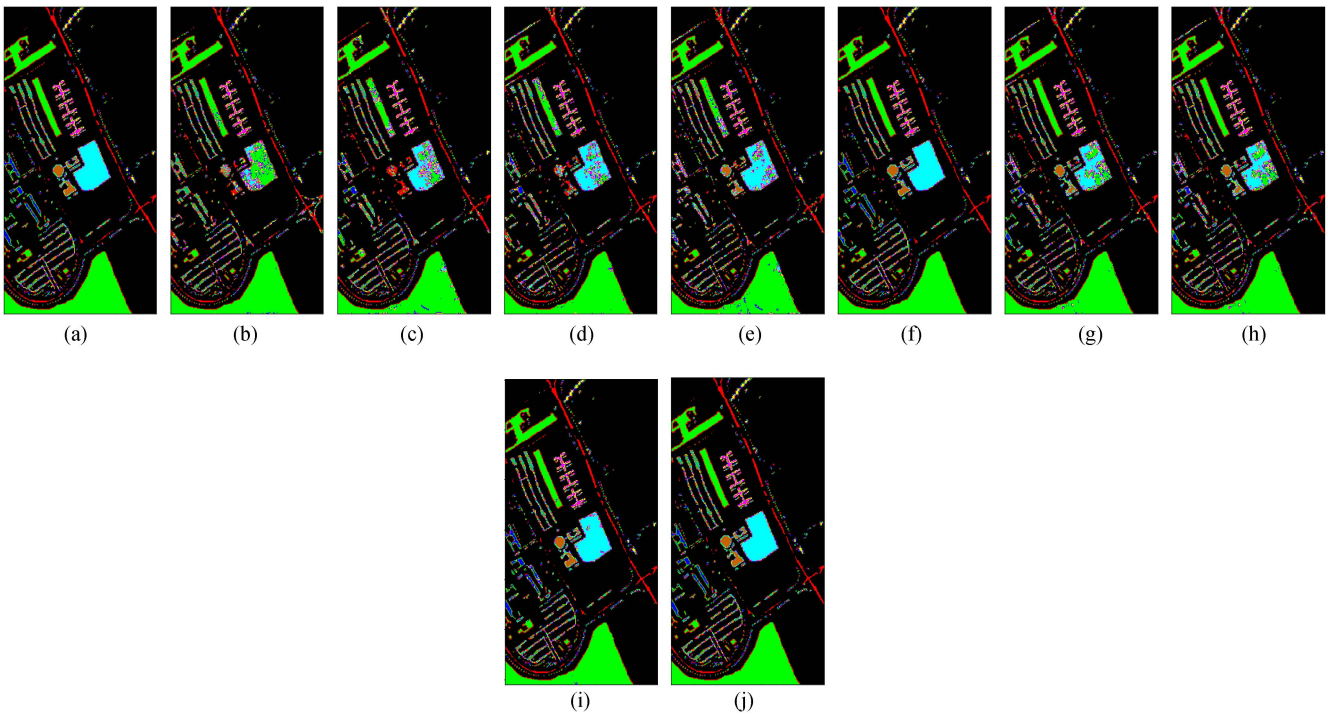


Fig. 9. Classification maps of various methods for the Pavia University dataset. (From Left to Right) (a) Ground truth. (b) SVM-RBF. (c) CCF-200. (d) 2-D CNN. (e) GCNN. (f) FADCNN. (g) NL-GCNN. (h) DWR-DEL. (i) E-CNN-PPF-SSF. (j) DHS.

methods have more complexity in representing the spectral–spatial feature. While with the DHS with snapshot and Relish execution, the proposed method acquired higher accuracy compared to the other state-of-the-art methods, which validates that the proposed DHS has the capabilities with the enhanced feature description. Furthermore, the PCA-based proposed methods performed low, with OA indicating that the representation of the primary components has a troubling impact on the dense feature extraction for the HSI data with the feature of high sparsity. In comparison, our method with the FA, snapshot ensemble, and Relish activation function module has better sparsity durability. We also performed experiments based on AA and K accuracy metrics, as Table VII shows that our proposed method outperformed the state-of-the-art methods with scores of 99.36% and 99.31%, respectively. The DHS method demonstrates the high performance and high reliability of HSIC in all of the experiments performed on the three datasets.

IV. DISCUSSION

To thoroughly validate the effectiveness of our proposed method, we perform a number of experiments, incorporating the effect of each part and scalability.

TABLE VII
CLASSIFICATION ACCURACIES OF PROPOSED DEEP HYPERSPECTRAL SHOTS IN TERMS OF OA, KAPPA, AND AA AGAINST THE STATE-OF-THE-ART METHODS THE PAVIA UNIVERSITY DATASET

No.	Class Names	SVM-RBF	CCF-200	2-D CNN	GCNN	FADCNN	NL-GCNN	DWR-DEL	E-CNN-PPF-SSF	DHS
1	Asphalt	82.37	86.59	83.85	78.89	**99.70**	86.80	97.69	99.08	99.57
2	Meadows	67.87	72.33	96.09	90.50	99.75	88.74	98.01	**100**	**100**
3	Gravel	69.18	71.75	81.47	71.70	94.31	70.84	98.33	94.45	**98.43**
4	Trees	98.37	99.09	96.12	98.76	99.10	98.43	99.07	99.09	**99.50**
5	Painted metal sheets	99.41	99.78	98.74	99.93	99.82	99.85	**100**	**100**	**100**
6	Bare Soil	93.64	97.26	49.79	79.08	99.92	94.37	99.52	**100**	**100**
7	Bitumen	91.20	91.88	79.32	71.20	96.97	86.24	**100**	99.09	98.41
8	Self-Blocking Bricks	92.59	94.92	88.89	92.83	97.90	96.74	98.05	96.15	**99.78**
9	Shadows	96.94	98.73	94.19	97.47	98.97	95.78	**100**	99.95	98.53
OA	-	78.89	83.36	86.93	87.08	99.19	90.04	98.41	99.10	**99.49**
AA	-	87.95	90.26	85.38	86.71	98.49	90.87	98.99	98.41	**99.36**
K	-	74.91	79.05	82.42	83.07	95.25	87.06	98.04	95.37	**99.31**

The bold face values shows the higher values in the comparative analysis.

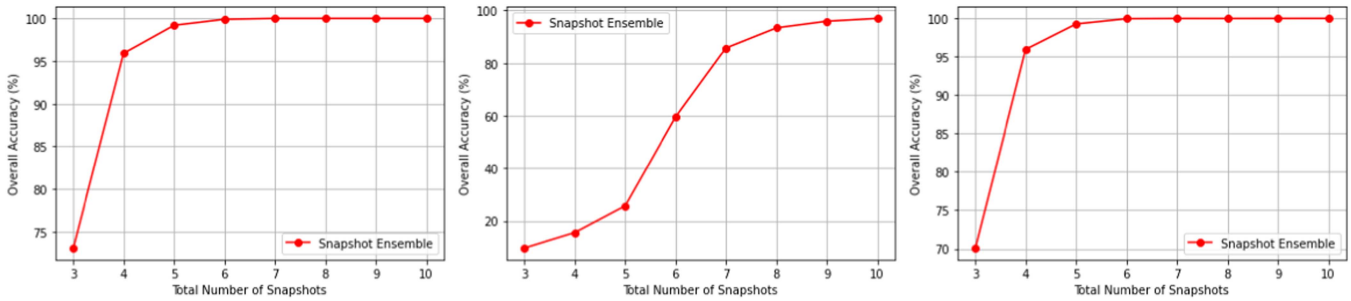


Fig. 10. OA effect of snapshots ensemble of the proposed method on datasets of (a) SA, (b) IP, and (c) PU.

A. Analysis of the Effect of Snapshots Ensemble

Fig. 10 demonstrates the effectiveness of DHS with various numbers of Snapshot models ensemble on three datasets. The snapshot models are chosen from the DHS intermediary models trained after every 50 iterations. As the results show, the efficiency variability is not very substantial when the range of snapshot models is increased from 6 to 8, indicating that DHS can reach fair accuracy without optimized snapshot model preference.

Furthermore, in this experiment, choosing 1–10 snapshot models can obtain relatively improved results than others. This suggests that utilizing numerous snapshot models, e.g., 15, could reduce prediction performance if several low-performance models are chosen. This inference is aligned with findings from other ensemble learning research findings, in which researchers discovered that if the efficiency of the core models is too terrible, the ensemble efficiency may suffer [101], [102]. However, thanks to the strong ensemble selection Even so, because of DHS's robust ensemble sampling adaptability, the ensemble efficiency does not deteriorate significantly even when we choose 10 snapshot models. As we can see in Fig. 10(a)–(c), the performance of the DHS is consistently increasing on SA, and PU datasets after 1–4 with the increasing number of snapshots while the performance on the IP dataset increased after 5–6. Hence, this demonstrates that our proposed DHS method is effective for HSIC datasets.

B. Analysis of the Depth of Snapshots

The depth of the base models is very important for any ensemble-based method. In order to optimize the efficiency of our proposed DHS, we performed a number of experiments to validate the efficiency. In Fig. 11, we can see that the performance of the model was lower at the beginning, but the performance increased with an increasing number of the depth of snapshots.

In this article, we select the total number of snapshots depth of only 10 and left the remaining sizes for future work. The performance of the snapshot at the depth of 3 is 99.88% OA on the dataset SA, 99.03% OA with the dataset PU, and 9.53% OA with the dataset IP, when the number of snapshot depth increased to 7 and the performance of the model increased to 99.89% OA on the SA dataset and 99.12% OA on the PU dataset, and also on 90.74% OA on the IP dataset. Moreover, when we kept the final snapshot depth as 10, the performance of the method increases to 99.89% OA on the SA dataset, 99.13% OA on the PU dataset, and, finally, 97.20% OA on the IP dataset. Hence, it shows that the depth of snapshots has a significant role in enhancing the performance of the method in terms of OA.

C. Analysis of Depth of Model

Numerous latest findings have illustrated that DNNs can substantially enhance the efficacy of a wide range of artificial intelligence applications, including CV and NLP. ResNeXt [103]

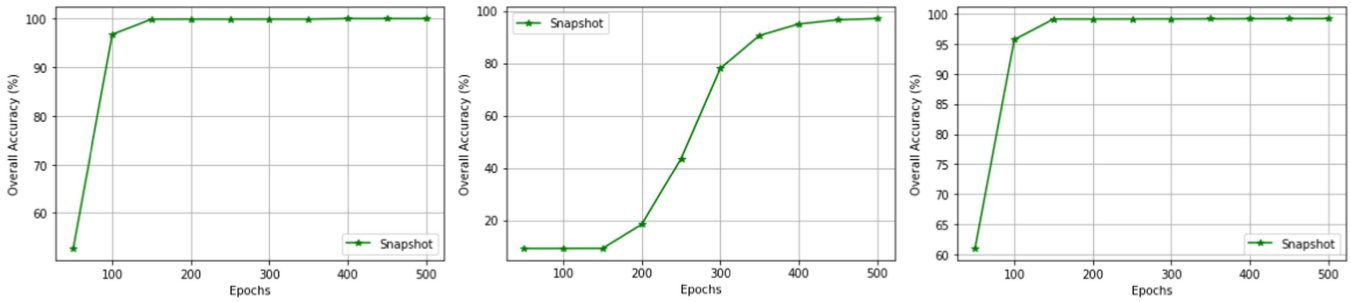


Fig. 11. OA depth of snapshot results based on datasets of (a) SA, (b) IP, and (c) PU.

is the most recent promising DNN for CV issues, and it has demonstrated outstanding results.

A ResNeXt model has been built upon a total of 50 and 101 layers in order to acquire top performance. In our proposed method, we use a DNN tower structure to allow the model to learn additional spatial–spectral features of three different dataset information. We utilize DHS with 28 CNN layers within the framework of the tower formation, and each layer has 256, 128, and 64, respectively, with the concatenation of wavelet transform levels. If the network topology map is designed by 64 neurons with the first layer, it can be expressed as (64). Same as if the depth of the DHS second part of layers with 128, it can be expressed as $< 128, 64 >$. Similarly, the third part of the CNN layer with 256 (256, 128, 64) and the fourth part of the CNN layers also with same layers of parameters of 256 (256, 256, 128, 64) respectively. As a result, if the depth of the CNN layers is I, thereafter the DHS Topological architecture is as described in the following: $[24^{2i1}, 23^{2i2}, \ldots, 24]$.

Fig. 12 shows the confusion matrix of various depths of a model that when the total number of CNN layers is increased from 4, 5, and 6, efficiency slowly increases, indicating that the depth of a DNN can increase the model's efficiency. Moreover, in the IP dataset, the proposed model performed poorly with the increasing number of CNN and wavelet transforms; the core reason for that is the total number of the training sample size of the IP dataset is very less. Furthermore, we also increased the wavelet transform decomposition layers number, which increases the sparsity of the IP dataset leading to the performance of IP dataset in decline. Although the performance on remaining datasets boosts the performance of DHS additionally, DNN capabilities must not be underestimated.

D. Analysis of Hyperparameter Effect

In this section, the effects of three various parameters on the DHS model's efficiency are investigated: 0.1, 0.01, and 0.001. Because of the vital spatial feature engaged, the learning rate of the DL training sample is critical for improving HSIC accuracy. In general, the small-scale learning rate relatively increased efficiency. Consequently, a lower learning rate of the training sample eventually results in the train-test data overlapping. On the one hand, we suppose that the size specification is appropriate for exploiting the plentiful local feature. On the other hand, we believe that this specification is not substantial to cause overlapping problems.

According to the comprehensive experimental findings, Fig. 13(a)–(c) shows qualitative analyses of classification results with an identical training sample size for all three different datasets. Particularly, the OA initially enhances with a learning rate of 0.001 and decreases for the Indian Pines dataset. As we can see, when the learning rate of the proposed method changed to 0.01, the performance was enhanced with an improved accuracy score of 95.96% on the IP dataset, 99.21% on the Pu dataset, and an accuracy of 99.95% score on the Salinas dataset. Specifically, the OA gets the highest value, when we integrate a 0.01 learning rate in our method; the performance significantly improved on three datasets. The OA accuracy rates are mainly steadily improving, with a more influential learning rate of 0.1, and the classification accuracy achieves its pinnacle. As we can observe in Fig. 13, the OA obtains the highest score for the Salinas dataset at 99.96%. When the learning rate is set to 0.1, our method generates the best accuracy of 99.49% achieved with the Purdue University dataset. Moreover, the accuracy score for the Indian Pines dataset is also improved with an accuracy score of 97.91%. Comparatively, it demonstrates that the larger the learning rate in our proposed method with spatial information implied is beneficial for improving accuracy.

E. Scalability

We evaluated the scalability of our proposed method by employing two criteria: 1) Response time and 2) throughput. We run experimental studies on the SA, PU, and IP datasets at three various densities of 10%, 20%, and 30% training samples. Fig. 14(a)–(d) shows that the proposed method performed well on a density of 10% at the SA dataset, with scores of 99.96% OA, and time of 12 min and 25 s. The score of the density of 10% on the PU dataset is 99.49% OA with a total time of 3 min and 50 s. Similarly, the score on the IP dataset is 97.91% OA with a total time of 55 s. We also conducted experiments on the 20% density of three datasets. The performance score of 20% on the SA dataset is 100% OA with a total time of 22 min and 21 s. The performance score of the PU dataset also acquired the 99.91% OA with a total time of 6 min and 55 s. The performance score of IP is 98.51% OA with a total time of 2 min and 18 s. Furthermore, we also evaluated experiments on 30% density

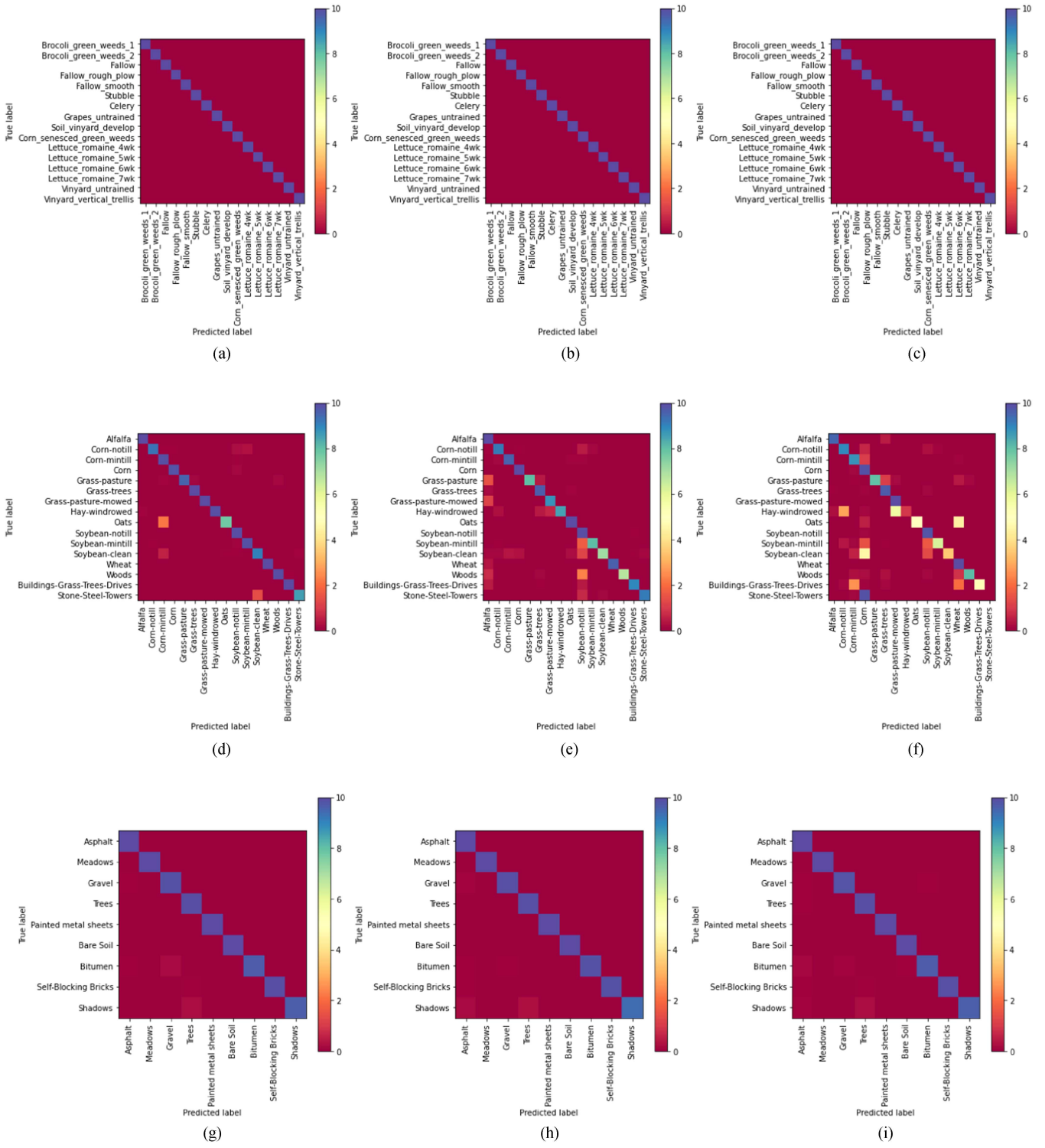


Fig. 12. Confusion matrix of the impact of the depth of the proposed method on datasets of (a)–(c) SA, (d)–(f) IP, and (g)–(i) PU.

on three datasets. The OA accuracy value of the SA dataset is 100% with a total time of 33 min and 58 s. The OA score on PU is also 99.93% with a total time of 10 min and 2 s. Finally, the IP dataset acquires 99.75% OA with a total time of 2 min and 25 s. In particular, the scalability of machine learning (ML) techniques relates to a collection of ML and DL models that can handle various amounts of data and parameters. This highly variable number of datasets can either be building the model, which is known as the training step, or after the model has been built, which is known as the testing or execution step. When the model is developed, scalability corresponds to the model's ability to perform well across a range of test datasets and

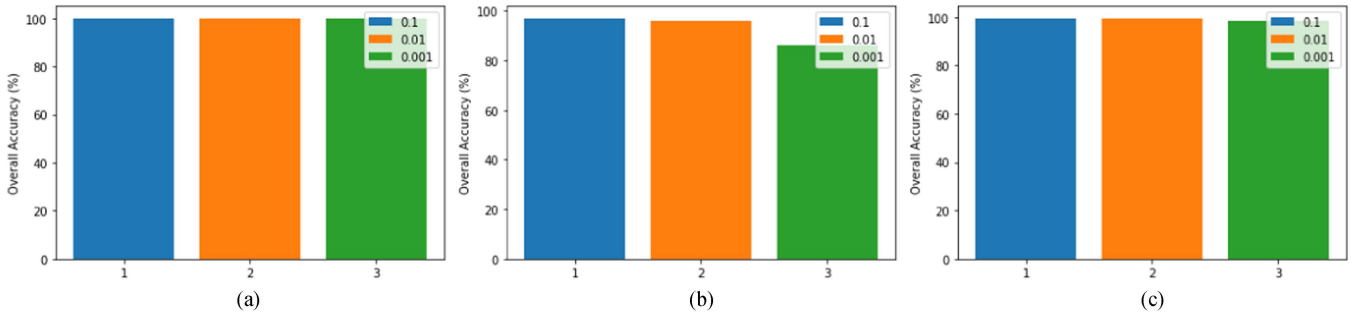


Fig. 13. Impact of different hyperparameter effects on the proposed method on data sets of (a) SA, (b) IP, and (c) PU.

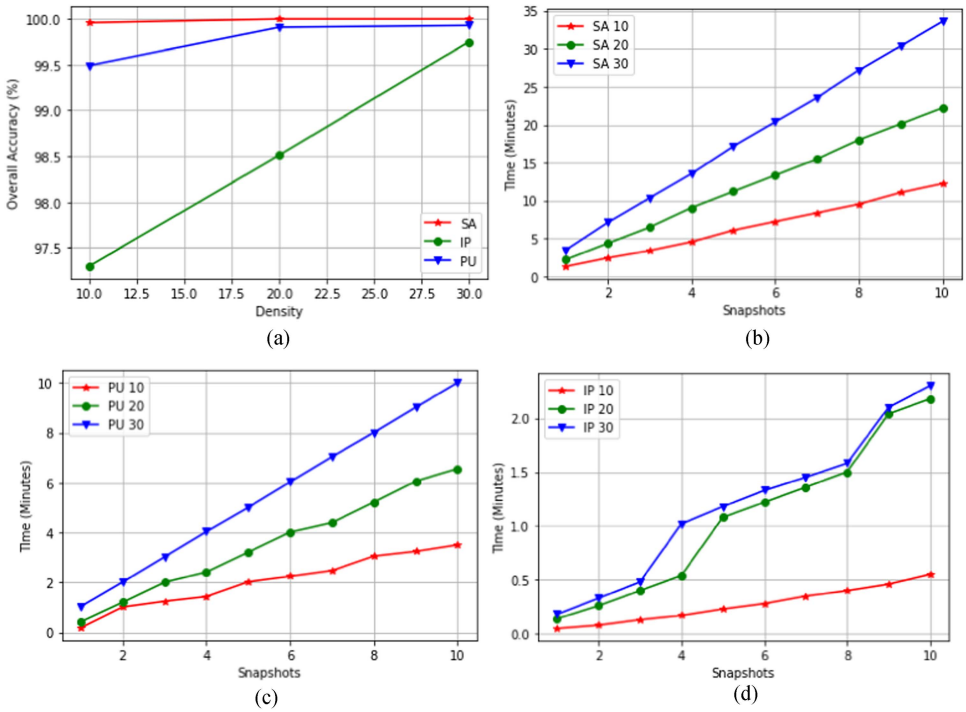


Fig. 14. Scalability results of the proposed method on three hyperspectral datasets. (a) Response time of the proposed method of SA, PU, and IP datasets. (b) Throughput on the SA dataset. (c) Throughput on the PU dataset. (d) Throughput on IP dataset.

hyperparameters. The effectiveness of the DL approach is also highly subjective and depends on a variety of hyperparameters. For instance, performance can be improved through optimization of memory consumption or improved recognition performance regardless of the testing data sizes and characteristics. Furthermore, the best scalable DL techniques should generalize effectively and enable for computational efficiency of large datasets.

In terms of the deep hyperspectral snapshots and the above interpretations of scalability, our method is very effective. Because the method is constructed with training- and testing-based validation, hence, the efficiency noted is the outcome of a significant percentage of iterations that mainly portray a realistic real-world scenario. In terms of ensemble integration, deep hyperspectral snapshots enable a very scalable method of DL where the number of new models is integrated by just training a single core model. This not only minimizes computation iterations but also improves OA results. DL generates numerous real-time applications for scalability in the massive dataset. DL has been shown to be effective in a wide range of practical and real-world issues when it comes to testing new and diverse datasets. As a result, we believe our method takes benefit of the latest advances in DL techniques, resulting in a plausible, effective, and scalable approach. Based on the experimental results in Fig. 14, we observed that the proposed method presents a scalable approach.

V. Conclusion

In this article, we propose a novel deep hyperspectral shot, a deep smooth wavelet CNN shots ensemble for HSIC. A deep smooth wavelet CNN incorporates the layers of wavelet transform to extract spectral features. Furthermore, computing

a wavelet transform is computationally lighter than 3-D CNN. After spectral features extraction, it is concatenated to the 2-D CNN, which extracts the spatial features, thus establishing a spatial–spectral feature vector for classification. Furthermore, we present the snapshots generation method and employ CAS to converge to various local minimums along its optimization path and save the base model. We build several snapshots of the deep hyperspectral shots model to enhance performance. We introduce the snapshots optimization and ensemble selection approach in order to solve the optimization problem within ensemble creation and further enhance the performance. Furthermore, we also present a novel Relish activation function that has been integrated into the proposed method to boost spatial–spectral feature propagation and advance for smoother gradients. Experimental results and analyses show that our proposed method DHS achieved OAs of 99.96%, 97.91%, and 99.49% on the Salinas, Indian Pines, and Pavia University datasets, respectively, against the state-of-the-art methods.

In future work, we will improve the proposed method by integrating the latest approaches, thereby expanding its suitability for HSI classification tasks. Additionally, we will build a lightweight ensemble that maintains classification performance while lessening computational complexity. Moreover, we will explore how incorporating state-of-the-art vision transformer models into our proposed method can enhance the interpretability of HSI data features. Alongside this, there is potential for extending the application of the DHS method to various tasks such as semantic segmentation and object detection, opening new avenues for research.

REFERENCES

[1] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.

[2] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[3] S. Roessner, K. Segl, U. Heiden, and H. Kaufmann, "Automated differentiation of urban surfaces based on airborne hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1525–1532, Jul. 2001.

[4] 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.

[5] J. Li, I. Dópido, P. Gamba, and A. Plaza, "Complementarity of discriminative classifiers and spectral unmixing techniques for the interpretation of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2899–2912, May 2014.

[6] F. F. Sabins, "Remote sensing for mineral exploration," *Ore Geol. Rev.*, vol. 14, no. 3/4, pp. 157–183, 1999.

[7] 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.

[8] 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.

[9] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

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

[11] M. Khodadadzadeh, J. Li, A. Plaza, and J. M. Bioucas-Dias, "A subspace-based multinomial logistic regression for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2105–2109, Dec. 2014.

[12] 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, 2015.

[13] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[14] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2013.

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

[16] P. Quesada-Barriuso, F. Argüello, and D. B. Heras, "Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1177–1185, Apr. 2014.

[17] B. Hou, T. Huang, and L. Jiao, "Spectral–spatial classification of hyperspectral data using 3-D morphological profile," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2364–2368, Dec. 2015.

[18] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.

[19] J. Liu, Z. Wu, Z. Wei, L. Xiao, and L. Sun, "Spatial-spectral kernel sparse representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2462–2471, Dec. 2013.

[20] L. He, J. Li, A. Plaza, and Y. Li, "Discriminative low-rank Gabor filtering for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1381–1395, Mar. 2017.

[21] C. J. D. Porta, A. A. Bekit, B. H. Lampe, and C.-I. Chang, "Hyperspectral image classification via compressive sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8290–8303, Oct. 2019.

[22] Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2077–2081, Nov. 2017.

[23] H. Yuan, Y. Y. Tang, Y. Lu, L. Yang, and H. Luo, "Spectral-spatial classification of hyperspectral image based on discriminant analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2035–2043, Jun. 2014.

[24] D. L. Donoho et al., "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, vol. 1, no. 2000, 2000, Art. no. 32.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.

[26] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[27] F. Ullah, B. Zhang, and R. U. Khan, "Image-based service recommendation system: A JPEG-coefficient RFs approach," *IEEE Access*, vol. 8, pp. 3308–3318, 2020.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[29] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[30] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing" in *Proc. Artif. Intell. Stat.*, 2012, pp. 127–135.

[31] F. Ullah et al., "Deep Edu: A deep neural collaborative filtering for educational services recommendation," *IEEE Access*, vol. 8, pp. 110915–110928, 2020.

[32] E. Merényi, W. H. Farrand, J. V. Taranik, and T. B. Minor, "Classification of hyperspectral imagery with neural networks: Comparison to conventional tools," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, pp. 1–19, 2014.

[33] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[34] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[35] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.

[36] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, 2019.

[37] 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.

[38] 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, 2016.

[39] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[40] 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.

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

[42] N. Li and Z. Wang, "Hyperspectral image ship detection based upon two-channel convolutional neural network and transfer learning," in *Proc. IEEE 5th Int. Conf. Signal Image Process.*, 2020, pp. 88–92.

[43] 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.

[44] C. Wang, N. Ma, Y. Ming, Q. Wang, and J. Xia, "Classification of hyperspectral imagery with a 3D convolutional neural network and J-M distance," *Proc. Adv. Space Res.*, vol. 64, no. 4, pp. 886–899, Aug. 2019.

[45] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral image classification method based on 2D–3D CNN and multibranch feature fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5776–5788, 2020.

[46] J. Zheng, Y. Feng, C. Bai, and J. Zhang, "Hyperspectral image classification using mixed convolutions and covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 522–534, Jan. 2021.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

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

[49] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.

[50] X. Zhang et al., "Spectral–spatial fractal residual convolutional neural network with data balance augmentation for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10473–10487, Dec. 2021.

[51] Y. Xu et al., "Dual-channel residual network for hyperspectral image classification with noisy labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5502511.

[52] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.

[53] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang, and J. Xie, "A hierarchy-to-sequence attentional neural machine translation model," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 623–632, Mar. 2018.

[54] F. Wang et al., "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.

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

[56] L. Mou and X. X. Zhu, "Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 110–122, Jan. 2020.

[57] L. Zhao, J. Yi, X. Li, W. Hu, J. Wu, and G. Zhang, "Compact band weighting module based on attention-driven for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9540–9552, Nov. 2021.

[58] P. Shamsolmoali, M. Zareapoor, J. Chanussot, H. Zhou, and J. Yang, "Rotation equivariant feature image pyramid network for object detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5608614.

[59] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 449–462, Jan. 2021.

[60] Z. Lu, B. Xu, L. Sun, T. Zhan, and S. Tang, "3-D channel and spatial attention based multiscale spatial–spectral residual network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4311–4324, 2020.

[61] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1307.

[62] L. Li, J. Yin, X. Jia, S. Li, and B. Han, "Joint spatial–spectral attention network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1816–1820, Oct. 2021.

[63] N. Li and Z. Wang, "Spectral-spatial fused attention network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 3832–3836.

[64] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3232–3245, May 2020.

[65] Z. Zhao, D. Hu, H. Wang, and X. Yu, "Center attention network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3415–3425, Jan. 2021.

[66] Z. Zhong, Y. Li, L. Ma, J. Li, and W.-S. Zheng, "Spectral–spatial transformer network for hyperspectral image classification: A factorized architecture search framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5514715.

[67] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Jan. 2020.

[68] X. He and Y. Chen, "Optimized input for CNN-based hyperspectral image classification using spatial transformer network," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1884–1888, Dec. 2019.

[69] H. Yan et al., "A 3D cascaded spectral–spatial element attention network for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 13, pp. 1–24, 2021.

[70] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sens.*, vol. 12, no. 3, p. 582, 2020.

[71] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[72] W. Liao, M. D. Mura, J. Chanussot, and A. Pižurica, "Fusion of spectral and spatial information for classification of hyperspectral remote-sensed imagery by local graph," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 583–594, Feb. 2016.

[73] H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality reduction of hyperspectral imagery based on spatial–spectral manifold learning," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2604–2616, Jun. 2020.

[74] H. Huang, Y. Duan, H. He, and G. Shi, "Local linear spatial–spectral probabilistic distribution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1259–1272, Feb. 2020.

[75] Z. Feng, S. Yang, M. Wang, and L. Jiao, "Learning dual geometric low-rank structure for semisupervised hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 346–358, Jan. 2021.

[76] Z. Zhang et al., "Multireceptive field: An adaptive path aggregation graph neural framework for hyperspectral image classification," *Expert Syst. Appl.*, vol. 217, 2023, Art. no. 119508.

[77] Y. Ding et al., "Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification," *Expert Syst. Appl.*, vol. 223, 2023, Art. no. 119858.

[78] Y. Ding et al., "Af2GNN: Graph convolution with adaptive filters and aggregator fusion for hyperspectral image classification," *Inf. Sci.*, vol. 602, pp. 201–219, 2022.

[79] Z. Chen, G. Wu, H. Gao, Y. Ding, D. Hong, and B. Zhang, "Local aggregation and global attention network for hyperspectral image classification with spectral-induced aligned superpixel segmentation," *Expert Syst. Appl.*, 2023, Art. no. 120828.
[80] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.
[81] Y. Zhou, J. Peng, and C. L. P. Chen, "Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1082–1095, Feb. 2015.
[82] Z. Feng, S. Yang, S. Wang, and L. Jiao, "Discriminative spectral–spatial margin-based semisupervised dimensionality reduction of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 224–228, Feb. 2015.
[83] E. Zhang, X. Zhang, L. Jiao, L. Li, and B. Hou, "Spectral–spatial hyperspectral image ensemble classification via joint sparse representation," *Pattern Recognit.*, vol. 59, pp. 42–54, 2016.
[84] Y. Zhang, G. Cao, A. Shafique, and P. Fu, "Label propagation ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3623–3636, Sep. 2019.
[85] F. Gao, Q. Wang, J. Dong, and Q. Xu, "Spectral and spatial classification of hyperspectral images based on random multi-graphs," *Remote Sens.*, vol. 10, no. 8, 2018, Art. no. 1271.
[86] N. Laban, B. Abdellatif, H. M. Ebeid, H. A. Shedeed, and M. F. Tolba, "Reduced 3-D deep learning framework for hyperspectral image classification," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.*, 2019, pp. 13–22.
[87] S. Fujieda, K. Takayama, and T. Hachisuka, "Wavelet convolutional neural networks," 2018, *arXiv:1805.08620v1*.
[88] S. Fujieda, K. Takayama, and T. Hachisuka, "Wavelet convolutional neural networks for texture classification," 2017, *arXiv:1707.07394*.
[89] X. Wang, "Moving window-based double Haar wavelet transform for image processing," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2771–2779, Sep. 2006.
[90] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 886–88609.
[91] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[92] L. N. Smith, "No more pesky learning rate guessing games," *CoRR*, vol. 5, p. 363, 2015.
[93] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via Bootstrap learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1884–1892.
[94] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen, and Y. Wu, "AdaBoost-based face detection for embedded systems," *Comput. Vis. Image Understanding*, vol. 114, no. 11, pp. 1116–1125, 2010.
[95] C. Zhang, J. Yan, C. Li, and R. Bie, "Contour detection via stacking random forest learning," *Neurocomputing*, vol. 275, pp. 2702–2715, Sep. 2018.
[96] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
[97] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "Feedback attention-based dense CNN for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5501916.
[98] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.
[99] H. Lu, H. Su, P. Zheng, Y. Gao, and Q. Du, "Weighted residual dynamic ensemble learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6912–6927, Jan. 2022.
[100] S. Dong, W. Feng, Y. Quan, G. Dauphin, L. Gao, and M. Xing, "Deep ensemble CNN method based on sample expansion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5531815.
[101] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1/2, pp. 239–263, 2002.
[102] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Proc. Neural Inf. Process. Syst.*, 1994, vol. 7.
[103] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.

Farhan Ullah received the B.Sc. degree in computer science from the University of Peshawar, Peshawar, Pakistan, in 2013, and the M.C.S. degree in computer science from the Sarhad University of Science and Information Technology, Peshawar, in 2015, and the Ph.D. degree in computer science and technology from the Shanghai University, Shanghai, China, in 2020.

He is currently a Postdoctoral Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include deep learning, hyperspectral image classification, and computer vision.

Yaqian Long received the Ph.D. degree in geosciences from the University of Alberta, Edmonton, AB, Canada, in 2019.

She is currently an Assistant Professor with Shenzhen University, Shenzhen, China. Her study area is hyperspectral remote sensing in geological, ecological, and medical applications. Her primary research interests include feature selection and classification using spectral mixture analyses, and state-of-the-art machine learning techniques.

Irfan Ullah received the B.Sc. and M.Sc. degrees in electrical engineering from the Sarhad University of Science and Information Technology, Peshawar, Pakistan, in 2012 and 2015, respectively, the Ph.D. degree in electrical engineering from the State Key Laboratory of Power Transmission Equipment and System Security and New Technology, School of Electrical Engineering, Chongqing University, Chongqing, China, in 2019, and the Postdoctoral degree in AI & New Technologies from the School of Engineering, Chulalongkorn University, Bangkok, Thailand, in 2021.

He is currently an Associate Professor with the School of Computer Science, Chengdu University of Technology, Chengdu, China. He has authored or coauthored several papers in conferences and journals, such as IEEE TRANSACTIONS ON POWER ELECTRONICS, IEEE TRANSACTIONS ON MAGNETICS, *Applied Thermal Engineering*, and IEEE ACCESS.

Rehan Ullah Khan received the B.Sc. and M.Sc. degrees in information systems from the University of Engineering and Technology at Peshawar, Peshawar, Pakistan, in 2004 and 2006, respectively, and the Ph.D. degree in artificial intelligence from the Vienna University of Technology, Vienna, Austria, in 2011.

He is currently an Associate Professor with IT Department, CoC, Qassim University, Buraydah, Saudi Arabia. His current research interests include machine learning, segmentation, and image-based object recognition.

Salabat Khan received the M.Sc. degree in computer science from the University of Peshawar, Peshawar, Pakistan, in 2009, the M.S. degree in computer science from International Islamic University, Islamabad, Pakistan, in 2014, and the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2020.

He is currently a Postdoctoral Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His main research includes machine learning, deep learning, security, and privacy issues on the Internet, IoT, and IoV.

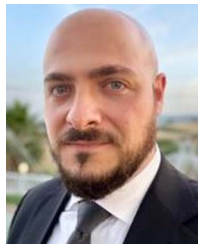
Khalil Khan received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2007 and 2012, respectively, and the Ph.D. degree in intellegence and machine learning from the Signals and Communication Laboratory, University of Brescia, Brescia, Italy, in 2016.

He is currently an Assistant Professor with the Department of Computer Science and Information Technology, Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan.

Maqbool Khan (Senior Member, IEEE) received the M.S. degree in information security from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2011, the Ph.D. degree in computer science and technology from Nanjing University, Nanjing, China, in 2013, and the Postdoctoral degree in AI & data science from Software Competence Center Hagenberg (SCCH), Hagenberg im Mühlkreis, Austria, in 2023.

He is currently an Assistant Professor with Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology (PAF-IAST), Haripur, Pakistan, and an Adjunct Researcher with SCCH. He worked in multinational companies, such as Siemens and Atos. He is an active researcher working on various projects with multiple collaborators and is currently working on a European Union project titled "Human-AI Teaming Platform for Maintaining and Evolving AI Systems in Manufacturing." He has more than 10 years of professional experience while working both in industry and academia. He has multidisciplinary expertise and working experience on diverse topics of big data analytics, cloud computing, predictive maintenance, explainable AI, knowledge graphs, data science, and machine learning.

Dr. Khan obtained a fully funded scholarship for the M.S. and Ph.D. degrees, awarded by the China Scholarship Council (CSC) of China. He is a certified Google Cloud Professional Architect. He also won a project during the Pakistan Scientific Foundation (PSF) CRP4 call as a Principal Investigator.



Giovanni Pau (Member, IEEE) received the bachelor's degree in telematic engineering from the University of Catania, Catania, Italy, in 2008, and the master's degree (*cum Laude*) in telematic engineering and the Ph.D. degree in technologies and management of aeronautical infrastructure from Kore University of Enna, Enna, Italy, in 2010 and 2015, respectively.

He is currently an Associate Professor with the Faculty of Engineering and Architecture, Kore University of Enna. He is the author/coauthor of more than 80 refereed articles published in journals and conferences proceedings. His research interests include wireless sensor networks, fuzzy logic controllers, intelligent transportation systems, Internet of Things, and network security.

Dr. Pau is a Member of IEEE (Italy Section) and has been involved in the organization of several international conferences as a Session Co-Chair and a Technical Program Committee member. He serves/served as a leading Guest Editor in special issues of several international journals and is an editorial board member as an Associate Editor for IEEE ACCESS, *Wireless Networks* (Springer), *EURASIP Journal on Wireless Communications and Networking* (Springer), *Wireless Communications and Mobile Computing* (Hindawi), and *Future Internet* (MDPI).