A Disjoint Samples-based 3D-CNN with Active Transfer Learning for Hyperspectral Image Classification

Muhammad Ahmad, Usman Ghous, Danfeng Hong, *Senior Member, IEEE*, Adil Mehmood Khan, Jing Yao, *Member, IEEE*, Shaohua Wang, Jocelyn Chanussot, *Fellow, IEEE*

***Abstract*—Convolutional Neural Networks (CNNs) have been extensively studied for Hyperspectral Image Classification (HSIC). However, CNNs are critically attributed to a large number of labeled training samples, which outlays high costs in terms of time and resources. Moreover, CNNs are trained on some samples and have been tested on the entire HSI. Perhaps, the entire HSI is taken into account at test time to appropriately generate the ground truth maps. In order to obtain a higher accuracy while considering the limited availability of training samples and disjoint validation and test samples, this work proposes a fast and compact 3D CNN-based Active Learning (AL) for HSIC that integrates both deep transfer learning and AL into a unified framework. In the proposed methodology, a 3D CNN model is trained with very few training samples (i.e., 5%, only) and in the next phase, the most informative and heterogeneous samples are queried from the validation set (candidate set) based on the fuzziness, mutual information and breaking ties of the trained model. The 3D CNN model is later fine-tuned (rather retraining from scratch) with the new training samples (i.e., 200 samples are selected in each iteration) to reduce the computational cost. The proposed method has been compared with the state-of-the-art traditional and deep models proposed for HSIC. Experimental results proved the superiority of our proposed method on several benchmark HSI datasets with significantly fewer labeled samples.**

Matlab demo can be accessed on GitHub: github.com/mahmad00

***Index Terms*—Hyperspectral Image Classification (HSIC); Transfer Learning; Active Learning (AL); 3D Convolutional Neural Network (3D CNN); Spatial-Spectral Information.**

I. Introduction

HYPERSPECTRAL IMAGING (HSI) involves extraction of useful spectral-spatial information from the object of interest. This is done by acquiring the radiance at short or long distances without contact using appropriate sensors [1], [2]. HSI can obtain very rich spectral information captured from the electromagnetic spectrum covering a wide range $400nm - 2400nm$, i.e. $400nm - 700nm$ (visible region), $700nm-2400nm$ (short wave infrared). This region is divided into hundreds of narrow and contiguous spectral bands. HSI can explore the light emission properties of objects in mid to long-infrared regions.

HSI Classification (HSIC) process aims to discriminate each spectral pixel and assign a unique class label according to the HSI content [3]. HSIC has been extensively studied and showed promising results for a number of applications, for instance, land cover classification, land use mapping, forest inventory, health sciences, unmixing, and urban areas [4]–[17]. HSIC has been broadly divided into two categories; 1): Spectral Classification and 2): Spatial-Spectral Classification [18].

Spectral-based methods only make use of spectral information and ignore the spatial correlation while classification, thus cannot obtain excellent performance. Whereas, Spatial-Spectral-based methods do consider both information (i.e., spectral information along with the spatial correlation) to overcome the limitations of spectral-based methods [19], [20]. The performance of these methods is much higher as compared to the former because they use a patch-based process that extracts the features in a local window.

In recent years, Deep Learning (DL)-based methods have been proposed for HSIC [21]. DL-based methods outperformed in a purely data-driven manner, however, their performance is entirely based on a large number of labeled training data. Without that, DL-based methods usually underperform in many cases. Here we have presented an example in which a 3D Convolutional Neural Network (CNN) has been trained on 5% disjoint training samples and the model is validated on 60% disjoint samples and finally tested on 35% disjoint test samples. We carefully make sure that $(Train \, \cap \, Validation \, \cap \, Test \, = \, \emptyset)$ and $(Train \, \cup \, Validation \, \cup \, Test \, = \, HSI)$. Moreover, the

This work was supported by the National Natural Science Foundation of China under Grant 42271350 and Grant 62201553. (*Corresponding author: Jing Yao*)

M. Ahmad and U. Ghous are with the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan. (e-mail: mahmad00@gmail.com; usman.ghous@nu.edu.pk).

D. Hong and J. Yao are with the Key Laboratory of Computational Optical Imaging Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (email: hongdf@aircas.ac.cn; yaojing@aircas.ac.cn).

A. Khan is with the Institute of Data Science and Artificial Intelligence, Innopolis University, Innopolis, Russia, and with the Department of Computer Science, University of Hull, United Kingdom, (email: a.khan@innopolis.ru).

S. Wang is with the International Research Center of Big Data for Sustainable Development Goals, Beijing 100094, China, with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, also with the State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (email: wangshaohua@aircas.ac.cn).

J. Chanussot is with the Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France, also with the Aerospace Information Research Institute, Chinese Academy of Sciences, 100094 Beijing, China. (e-mail: jocelyn@hi.is)

same model has been tested on the entire HSI data and the experimental results are presented in Table I and Figure 1.

The aforementioned experimental results confirm the claims, i.e., DL does not perform well when there are not enough training samples available. Moreover, it has been observed that the performance is significantly reduced when the model is tested on disjoint test samples as compared to the entire dataset. Furthermore, the same model has been trained using 50% of training samples and validated on 25% data, and tested on remaining 25% data samples. The comparative accuracies and ground truths are presented in Figure 1 and Table I. One can conclude from these experimental results that the model's performance has been significantly improved with a higher number of training samples as compared to the less number of training samples. Thus, the question arises, is there any way to get a similar kind of generalization performance and accuracy for the same model with less number of labeled training samples, and more importantly, will the model work the same way in disjoint train/validation/test samples case?

As discussed above, it is impractical to assume that each HSI under process must have enough labeled data to train a DL model. Another way around, the labeling process always comes with a cost in terms of time and money, more specifically, requiring experts to hold certain domain knowledge to annotate HSI in many real-life applications. Thus, this paper addresses the aforementioned issue by automating the annotation process with the guarantee of accuracy, specifically when using DL for HSIC.

To effectively address the aforementioned issues, Active Learning (AL) can be considered a promising method that systematically selects the most informative and dissimilar samples for the user to label and train a classifier. Since there is a proven fact that all the samples are not equally important for training, thus only a few samples (e.g., informative, less redundant, dissimilar, etc.) define the hyperplane (separating surface) and the rest of the samples can be considered redundant. Therefore, carefully selecting the important samples that define the hyperplane can significantly reduce the sampling cost, avoid redundancy, and more importantly, guarantee good performance. These are a few facts that motivate us to combine CNN with AL.

Therefore, this article proposed an AL-integrated 3D CNN method into a unified framework by fully utilizing the benefits of both domains, such as the labeling efficiency of AL and the strong discriminative ability of DL. There have been many works that combine AL with DL for HSIC [22]–[26], however, the proposed method has its specific characteristics such as:

1) The proposed method adopts 3D CNN architecture and inexpensive multi-class sample selection criteria to actively select the most informative and heterogeneous samples. The higher fuzziness-based misclassified samples selection concept is used to reduce the labeling cost. Higher fuzziness-based misclassified samples are most likely neither adjacent nor from the same class with the same fuzziness magnitude. Moreover, mutual information and breaking ties-based sample selection methods have been compared.
2) Irrespective of the traditional AL integrated DL, this work makes use of fine-tune concepts in the AL process. Rather than training the 3D CNN in each iteration which is quite expensive in terms of computational cost, we simply fine-tune the model in each iteration, which significantly reduces the retraining cost.
3) The proposed method considers disjoint training, validation, and test samples to train, validate, and test the model, different from the previous studies. The experimental results have been shown in all possible cases, i.e., disjoint train/validation/test and the same model has been tested on the entire HSI dataset, respectively. In supervised HISC, traditional experimental designs are often improperly used in the spatial-spectral processing context, leading to unfair or biased performance evaluation. The widely adopted sampling methods are not always suitable to evaluate spatial-spectral methods, because it is difficult to determine whether the improvement of classification accuracy is caused by incorporating spatial information into the classifier or by increasing the overlap between training and testing samples [27]. To handle this problem, we used a controlled non-overlapping sampling strategy for spatial-spectral HSIC which eliminate the overlap between training and test samples and provides a more objective and accurate evaluation.

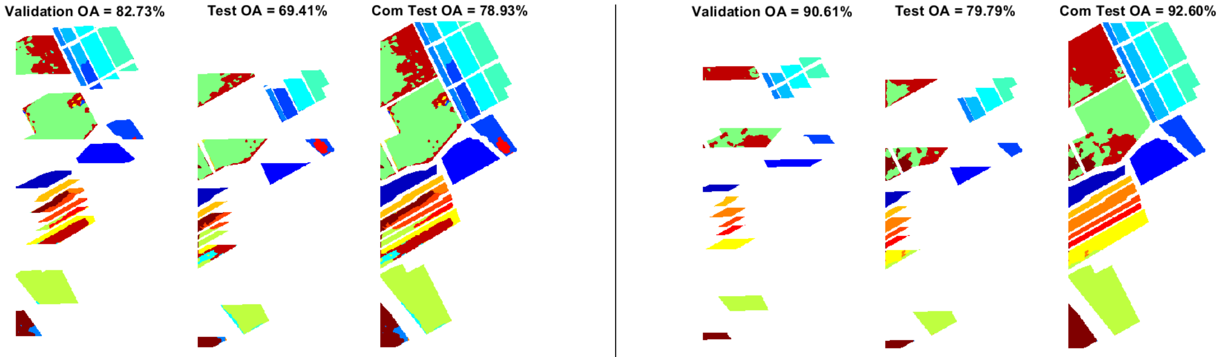
The proposed method attempts to further strengthen AL-based DL with more contextual information to reduce the labeling cost. The proposed method also helps to reduce the number of labeled samples required to train a 3D CNN model and produces higher accuracy.

The rest of the paper is structured as follows. Section II provides a comprehensive review of state-of-the-art (SOA) works published in recent years. Section III describes the problem formulation and proposed methodology. Section IV presents the experimental settings, datasets, and results with discussion. Furthermore, the sections IV-C (Experimental Results), IV-D (Statistical Tests and Computational Time), and IV-E (Comparison with SOA) provides a detailed discussion on results with different experimental settings. Finally, section V concludes the paper with possible future research directions.

II. Literature Review

In recent years, DL methods have been extensively studied for HSIC, for instance, Stacked Autoencoder (SAE) [28]–[32], Multi-layer Extreme Learning Machine (ML-ELM) [33], [34], Deep Boltzmann Machine (DBM) [35], CNN [21], [36]–[40], Cross-Modality and Coupled CNN's [41]–[44], and Deep Belief Network (DBN) [45]–[47].

SAEs are unsupervised feature extraction methods used to extract both spatial as well as spectral features by stacking a series of AEs. A modified CNN framework was proposed in [48] that uses 3-dimensional patches as input to process both spatial and spectral information at the same time. In contrast to the work proposed in [48], the work [49] proposed a combined spatial pyramid pooling strategy that fully considered spatial information.



(a) 5% training, 60% Validation, 35% Test Samples, and complete dataset as test.

(b) 50% training, 25% Validation, 25% Test Samples, and complete dataset as test.

Fig. 1: Classification accuracy of 3D CNN model trained with 5% and 25% disjoint training samples, 60% and 25% disjoint Validation samples, 35% and 50% disjoint Test samples, respectively. Moreover, the same model has been tested on the complete Salinas Dataset (Com Test OA). The number of training, validation, and test samples, class names, and class-wise accuracies for both percentages of training/validation/test samples are presented in Table I

TABLE I: Per-class classification accuracy of 3D CNN model trained with 5% training samples, 60% Validation samples, 35% Test Samples. Moreover the same model have been tested on complete Salinas Dataset. The same model with same settings has been retrained on 50% training samples and validated and tested on 25%, and 25% data samples, respectively. One can observe from the results that the accuracies have increased but not that significant as the number of training samples increased. This is the point claimed in this research to obtain significantly higher accuracies with the least computations over the 3D CNN model.

Class Name	Samples (Tr/Val/Te)		Disjoint Validation		Disjoint Test		Complete Test	
	5%	50%	5%	50%	5%	50%	5%	50%
Brocoli Green Weeds 1	(101,1205,703)	(1005,502,502)	1	1	1	0.9940	1	0.9985
Brocoli Green Weeds 2	(186,2236,1304)	(1862,932,932)	1	1	1	1	1	1
Fallow	(98,1186,692)	(988,494,494)	0.9856	1	0.5014	1	0.8168	1
Fallow Rough Plow	(70,836,488)	(696,349,349)	1	1	0.8442	0.9770	0.9454	0.9942
Fallow Smooth	(134,1607,937)	(1338,670,670)	0.5849	0.9985	0	0.9940	**0.4010**	0.9981
Stubble	(198,2375,1386)	(1979,990,990)	0.9915	1	0.9942	1	0.9929	1
Celery	(179,2147,1253)	(1789,895,895)	0.9990	1	1	1	0.9994	1
Grapes Untrained	(563,6763,3945)	(5635,2818,2818)	0.8854	0.6149	0.8724	0.4308	0.8866	0.7614
Soil Vinyard Develop	(310,3722,2171)	(3101,1551,1551)	0.9970	1	0.9249	0.9974	0.9719	0.9993
Corn Senesced Green Weeds	(164,1967,1147)	(1638,820,820)	0.4295	1	0.1595	0.6829	**0.3636**	0.9206
Lettuce Romaine 4wk	(53,641,374)	(534,267,267)	1	0.9925	1	1	1	0.9981
Lettuce Romaine 5wk	(97,1156,674)	(963,482,482)	0.1608	1	0.0044	1	**0.1484**	1
Lettuce Romaine 6wk	(45,550,321)	(458,229,229)	0.9581	1	0.4267	0.9956	0.7740	0.9989
Lettuce Romaine 7wk	(53,642,375)	(534,268,268)	0.5716	1	0	0.9664	**0.3925**	0.9915
Vinyard Untrained	(363,4361,2544)	(3634,1817,1817)	0.6487	0.8992	0.4127	0.5376	**0.5836**	0.8592
Vinyard Vertical Trellis	(91,1084,632)	(903,452,452)	0.8210	1	0.8797	0.9955	0.8505	0.9988
Average	—	—	0.8146	0.9690	0.6262	0.9107	0.7579	0.9699
Overall	—	—	0.8273	0.9061	0.6940	0.7979	0.7893	0.9259
kappa (κ)	—	—	0.8069	0.8960	0.6566	0.7773	0.7641	0.9179
Time	Training – 1355 Sec.	7.3409e+03 Sec	10 Sec.	7.5 Sec	6 Sec.	4.9 Sec	17 Sec.	22.7 Sec

Moreover, the works [50] proposed a framework, combining CNN with hand-crafted features along with Conditional Random Field (CRF) and Markov Random Field (MRF). A dual-channel CNN i.e., a combined 1 and 2-dimensional CNN model has been proposed in [51]. A fast and compact 3-dimensional CNN model has been proposed in [52] which significantly reduces the computational cost and improves the experimental results for several Hyperspectral datasets. In this hierarchy, the works [37], [38], [53], [54] proposed Hybrid 3-dimensional followed by 2-dimensional CNN layers for a better spatial-spectral feature hierarchy for end classification. The proposed Hybrid models significantly improve the beam search which helps to get better accuracy. Such models provide statistical significance and better generalization performance of the CNN model in a reduced time.

In recent years, CNN coupled with Active Learning (AL) has been studied for HSIC. For instance, the work [55] proposed a semi-supervised multinomial logistic regression model combined with an entropy-based sample selection strategy for AL. Later on, the works [56], [57] proposed a Loopy belief propagation and Bayesian classification approaches combined with AL. Moreover, the work [58] proposed a model-based

AL method where SVM is used for classification, along with six different sample selection methods.

There are several other AL methods proposed in the literature for HSIC while considering the limited availability of training samples and iteratively selecting the most informative and heterogeneous samples to query for HSIC [2], [34], [59]–[62]. More recently, the work [63] proposed to combine multiclass-level uncertainty-based sample selection method with an SAE-based neural network. Whereas, the work [64] presented a weighted incremental dictionary learning criterion with the RBM method. Moreover, the work [65] presented a method that combined six different sample selection methods including maximum entropy, random sampling, breaking ties, modified breaking ties, mutual information, etc., with the BCNN method.

The aforesaid methods have achieved excellent performance for HSIC while considering the limited availability of training samples, however, the proposed method is different than the ones discussed above. First, the proposed method adopts a 3-dimensional CNN architecture rather than SAE, RBM and BCNN, etc. Secondly, the proposed method uses several integrated multiclass sample selection criteria to select the most informative and spectral-spatially heterogeneous samples. Third, the proposed method employs the transfer learning concept to accelerate the training process of 3D CNN and reduce the computational cost of retraining a 3D CNN. Finally, the proposed method integrates contextual information using prior probabilities. The aforementioned aspects are mainly considered different from the existing related works proposed in recent years.

III. Problem Formulation

An HSI cube can be expressed as $X = \{x_i, y_i\} \in \mathcal{R}^L$ where each $x_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,L}\} \in \mathcal{R}^L$ and y_i be the class label of each x_i. Here we first randomly select $X_{train} = 0.05\%$ training samples, $X_{val} = 0.60\%$ validation samples (pool set), and $X_{test} = 0.35\%$ test samples. We make sure that $|X_{train}| \ll |X_{val}|; |X_{train}| \ll |X_{test}|$ and $X_{train} \cap X_{val} \cap X_{test} = \emptyset$ for each iteration i.e., training, validation and test sets must not contain any single samples which is overlapped with other set. The training, validation, and test sets must need to be disjointed to avoid biases.

A. Convolutional Neural Network (CNN)

1D and 2D CNN models have been studied for HSIC, however, these models are unable to cater to both spatial-spectral information together, thus 3D CNN models are capable to address aforesaid issues, i.e., 3D CNN can extract the spectral information correlated with spatial characteristics of HSI at the same time. In general, the network architecture of 2D and 3D CNN is quite similar except for the convolutional process followed by an activation function (non-linearity induction process). The major difference is a convolutional kernel, e.g., the 2D CNN model uses a 2D kernel function whereas, the 3D CNN model uses a 3D kernel function. Moreover, 3D CNN's performance is much higher than 2D CNN because it uses a patch of an image to extract both spatial-spectral local features. 3D CNN performs operations on the spatial-spectral dimensions at the same time to extract both features at the same time. Figure 2 shows an example of the 3D CNN process adopted in this work.

The 3D convolutional process initially computes the sum of the dot product between the input patches and the 3D kernel function. This is done by convolving the 3D input patch with the 3D kernel function and results in a 3D feature map. The feature map produced is then passed on to an activation function to induce non-linearity in it. In such kind of convolutional process, the activation value of spatial location (x, y, z) at the i^{th} layer and j^{th} feature map can be formulated as:

$$v_{i,j}^{x,y} = ReLu(b_{i,j} + \sum_{\tau=1}^{d_{l-1}} \sum_{\sigma=-\delta}^{\delta} \sum_{\lambda=-v}^{v} \sum_{\rho=-\gamma}^{\gamma} w_{i,j,\tau}^{\sigma,\rho,\lambda} \times v_{i-1,\tau}^{x+\sigma,y+\rho,z+\lambda}) \tag{1}$$

where d_{l-1}, $b_{i,j}$, and $w_{i,j}$ represent the number of feature maps, the bias parameter, and depth of kernel for j^{th} feature map at $(l-1)^{th}$ layer, respectively. $2v+1$, $2\gamma+1$, and $2\sigma+1$ is the depth, width and height of the kernel. ReLu defines the activation function.

ReLu can converge faster than other activation functions such as the Sigmoid and Tanh functions. The form of ReLu used here is $f(x) = max(0, x)$. Finally, a softmax classifier is used to classify HSI features. Softmax loss used to train the model makes use of random admiral descent of backpropagation to minimize the loss of the network. The details of 3D convolutional layers are as follows: $layer\_1 = 60 \times 3 \times 3 \times 7$ i.e. $K_1^1 = 3, K_2^1 = 3$ and $K_3^1 = 7$. $layer\_2 = 30 \times 3 \times 3 \times 5$ i.e. $K_1^2 = 3, K_2^2 = 3$ and $K_3^2 = 5$. $layer\_3 = 10 \times 3 \times 3 \times 3$ i.e. $K_1^3 = 3, K_2^3 = 3$ and $K_3^3 = 3$. In total, three convolutional layers are stacked for low and high-level feature learning i.e., to increase the number of spatial-spectral feature maps and to distinguish the spatial/spectral features while preserving the spectral information. The convolutional process produces zero filling thus it does not require the use of batch normalization or data enhancement. Moreover, the weights are initially randomized and later optimized using backpropagation with Adam optimizer using softmax loss function. The entire network is trained over 50 epochs using a mini-batch of 256.

B. Active Learning (AL)

Active Learning (AL) has been considered an effective method to reduce the labeling cost as well as acquire a large number of labeled training samples [66]. AL is based on three main aspects; 1): The availability of initial training set X_{train}, 2): The availability of pool set (validation set in this work) X_{val}, 3): Query function e.g., informative sample selection or acquisition function.

Let us consider $X_{train} = [X, Y] = \{x_i, y_i\}_{i=1}^{l}$ as a training set consisting l samples where $x_i \in \mathcal{R}^d = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,L}\}$ and $y_i = \{1, 2, 3, \dots, Y\}$ and $X_{val} = [X] = \{x_i\}_{i=l+1}^{u} \in \mathcal{R}^d$ be the validation (pool set) set, i.e., a set of u samples and $l \ll u$. AL methods are

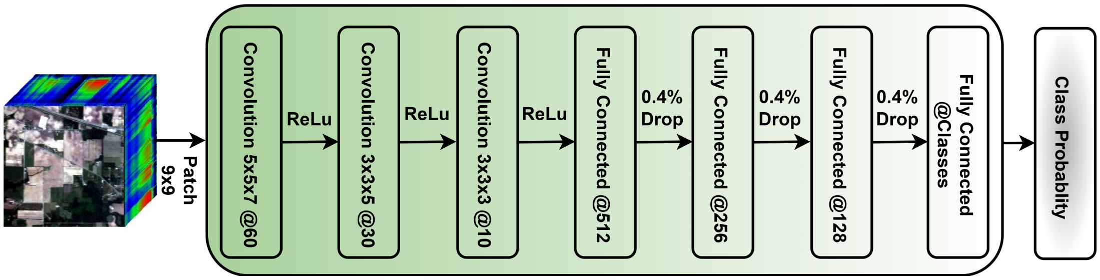


Fig. 2: 3D CNN network structure for HSIC. The input patch is with size $9 \times 9 \times d$. The first 3D convolutional layer contains 60 filters with $3 \times 3 \times 7$, second 3D convolutional layer includes 30 filters with $3 \times 3 \times 5$, third 3D convolutional layer includes 10 filters with $3 \times 3 \times 3$. The first fully connected layer contains 512 units with 0.4% dropout, the second fully connected layer contains 256 units with the same dropout, the third fully connected layer contains 128 units with the same dropout, and the fourth fully connected layer contains the number of classes existed in HSI.

composed of a learner (3D CNN in this study) which is trained on a small number of training samples and iteratively selects new training samples from the validation set. The process provides maximal information about the dataset and improves the model's performance. As a result of the AL process, the final classification results given by the selected training set are much higher than the ones obtained by randomly selected training samples.

The sample selection function (i.e., sample acquisition or query function), in particular, the user-defined heuristic is a crucial point for any AL method. Here in this research, we rely on the posterior probability-based AL method, i.e., fuzziness computed from the membership function (i.e., posterior probabilities $p(y|x)$) produced by the classifier to rank the candidates in X_{val}. Moreover, two other query functions i.e., Breaking ties and Mutual Information are used for comparison purposes.

C. Query Function

The query function for any AL method can be represented as $\alpha(x, \mathcal{M})$ of a model $\mathcal{M}$ with X_{val} data and inputs $x \in X_{val} \in \mathcal{R}^d$ decides which samples x will be queried by an external oracle. This process is being led by a human expert however, in this work, we systematically performed the work of classifying the unlabeled data to be added to the original training set. In this research, we performed a comparison of three different query functions that have been adopted to AL taking into account different measurements, such as breaking ties, mutual information, and fuzziness.

1) **Breaking Ties (BT)** [67] focuses on the boundary region between two different classes intending to obtain more diversity in the composition of the training set. The samples x_{BT} are selected from X_{val} by;

$$x_{BT} = \underset{x_i \in X_{val}}{\operatorname{argmax}} \{ \max_{y \in Y} \; p(y_i = y | x_i, \mu) - \max_{y \in Y / y^+} \; p(y_i = y | x_i, \mu) \} \quad (2)$$

where $y^+ = \underset{y \in Y}{\operatorname{argmax}} \; p(y_i = y | x_i, \mu)$ are the most probable label class for sample x_i.

2) **Mutual Information (MI)** [68]: It computes the mutual dependencies among the samples and only selects the samples x_{MI} that maximizes the MI between the actual class labels and obtained results as follows:

$$x_{MI} = \underset{x_i \in X_{val}}{\operatorname{argmax}} \; I(\mu; y_i | x_i) \quad (3)$$

where

$$I(\mu; y_i | x_i)) = \frac{1}{2} log(H_{MI}/H) \quad (4)$$

The above expression computes the MI between the class label y_i and obtained results, where H represents the posterior precision matrix and H_{MI} represents the posterior precision matrix after including the new samples.

3) **Fuzziness** [2]: Any trained probabilistic classification model produces the output matrix ($\mu = \mu_{ij}$) which is being used to compute the membership matrix with the following properties $\sum_{j=1}^{C} \mu_{ij} = 1$ and $0 < \sum_{i=1}^{M \times N} \mu_{ij} < 1$ where $\mu_{ij} = \mu_j(x_i) \in [0, 1]$. μ_{ij} represents the membership of x_i sample belongs to y_j class [59]. μ_{ij} is used to compute the fuzziness of $m = (l + 1 \rightarrow u)$ samples for Y classes as;

$$x_{\mathcal{F}} = \frac{-1}{m \times Y} \sum_{i=1}^{m} \sum_{j=1}^{Y} [\mu_{ij} log(\mu_{ij}) + (1 - \mu_{ij}) log(1 - \mu_{ij})] \quad (5)$$

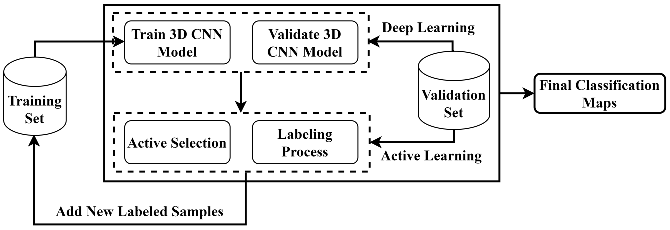


Fig. 3: Proposed Active DL Method for HSIC.

Algorithm 1: A fast and compact 3D CNN based AL

```
Data: $X_{train}, X_{val}, N, \varepsilon$
$X^{\varepsilon}_{train} = \{x_i, y_i\}^{l}_{i=1} \in \mathcal{R}^d,\ \varepsilon = 1 \rightarrow$ Initial Disjoint Training Set;
$X^{\varepsilon}_{val} = \{x_i, y_i\}^{u}_{i=l+1} \in \mathcal{R}^d,\ \varepsilon = 1 \rightarrow$ Initial Disjoint Validation Set (Pool Set);
$N \rightarrow$ Number of spectral samples to add in training set at each iteration until to reach a final batch of selected spectral samples i.e., $X_{Selected}$;
$\varepsilon \rightarrow$ Iteration number;
while $|X_{train}| \leq Threshold$ do
    if $if\ \varepsilon = 1$ then
        Train the Model with $X^{\varepsilon=1}_{train}$ and evaluate on $X^{\varepsilon=1}_{val}$ and get $\mu$ (membership matrix);
    else
        Fine-tune the Model with $X^{\varepsilon+1}_{train}$ and evaluate on $X^{\varepsilon+1}_{val}$ and get $\mu$ (membership matrix);
    end
    $x_{BT} = \underset{x_i \in X_{val}}{\text{argmax}}\{\max_{y \in Y}\ p(y_i = y|x_i, \mu) - \max_{y \in Y/y^+}\ p(y_i = y|x_i, \mu)\} \rightarrow$ Compute Breaking Ties;
    $x_{MI} = \underset{x_i \in X_{val}}{\text{argmax}}\ I(\mu; y_i|x_i) \rightarrow$ Compute Mutual Information;
    $x_{\mathcal{F}} = \frac{-1}{m \times Y} \sum_{i=1}^{m} \sum_{j=1}^{Y} [\mu_{ij} log(\mu_{ij}) + (1 - \mu_{ij}) log(1 - \mu_{ij})] \rightarrow$ Compute fuzziness magnitude;
    Rank the candidates $x_i$ in $X^{\varepsilon}_{val}$ according to $x_{\mathcal{F}}, x_{MI}, x_{BT}$;
    $X^{\varepsilon}_{Selected} = \{x_k\}^{N}_{k=1} \rightarrow$ select $N$ spectral samples which were misclassified with higher fuzziness magnitude, same number of samples are selected from $x_{MI}$ and $x_{BT}$, respectively;
    $X^{\varepsilon}_{Selected} = \{x_k, y_k\}^{N}_{k=1} \rightarrow$ assigned true class labels to the selected samples;
    $X^{\varepsilon+1}_{train} = X^{\varepsilon}_{train} \cup X^{\varepsilon}_{Selected} \rightarrow$ Add new batch of samples to the training set;
    $X^{\varepsilon+1}_{val} = X^{\varepsilon}_{validation} - X^{\varepsilon}_{Selected} \rightarrow$ Remove batch of samples from the validation set;
    $\varepsilon = \varepsilon + 1 \rightarrow$ Update iteration index
    Repeat until $|X_{train}| > Threshold$ or Maximum number of Iterations meet;
end
```

Figure 3 provides a detailed illustration of the proposed method and the complete pipeline is presented in the Algorithm. Overall, the proposed method combines 3D CNN with AL strategy in order to reduce the labeling cost and required number of labeled training samples. The proposed method consists of the following steps. 1): construct an initialized training patch set corresponding to a limited number of randomly selected labeled samples. 2): 3D CNN is trained on randomly selected training samples. 3): Actively select the most informative and heterogeneous samples from the validation set based on the class probabilities (fuzziness, Mutual Information, and Breaking ties, respectively) obtained from trained 3D CNN. Later the patches of the selected samples are labeled and added to the training set, which is regarded as a new training set for the next iteration. 4): To overcome the computational cost of retraining the 3D CNN, we freeze the convolutional and the first two fully connected layers of the model. The last fully connected layer along with the output layer is used to fine-tune the model in each iteration except the first.

IV. Experimental Evaluation

This section presents experimental datasets with their ground truths, class names, and total samples in each class. Ground truth maps are essential for supervised classification however, this work considers a scenario in which the ground labels are limited.

A. Experimental Datasets

Table II presents the details of each dataset used in the experiments and Table III provides the number of disjoint training, validation, and test samples selected from each class to train, validate and test the proposed and comparative methods. Moreover, the geographical maps for disjoint training, validation, and tests samples are shown in Figure 4. We stress the point that the number of training, validation, and test samples and their geographical locations remain the same for all methods used for experimental evaluation, So that unbiased and fair evaluations can be presented.

TABLE II: HSI datasets description used for experimental evaluation.

Data	PU	KSC	SA
Source	ROSIS-03	AVIRIS	AVIRIS
Sensor	Aerial	Aerial	Aerial
Resolution	$1.3\ m$	$10\ nm$	$3.7\ m$
Spatial Information	610×610	512×614	340×1905
Spectral Bands	115	176	224
Wavelength	$430 - 860$	$400 - 2500$	$0.35 - 1.05$
Classes	9	13	16
Samples	207400	314368	54129

Kennedy Space Center (KSC) data cube has been acquired by NASA using an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) instrument over Florida, on March 23, 1996. KSC data cube consists of 224 bands of 10 nm width with center wavelengths from 400-2500 nm with an altitude of approximately 20 km with a spatial resolution of 18 m. The low resolution (low SNR) and water absorption bands were removed prior to the experiments.

Salinas (SA) data cube was collected by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Salinas Valley, California. SA cube consists of 224 bands and is characterized by high spatial resolution i.e., 3.7-meter pixels. The total spatial lines comprise 512×217 samples. 20 most noisy and water absorption bands i.e., [108-112], [154-167], 224, were removed prior to the experiments. SA cube is available only as at-sensor radiance data, and it includes bare soils, vineyard fields, and vegetables. In total, the SA cube contains samples of 16 different classes, i.e., ground truths consist of 16 classes.

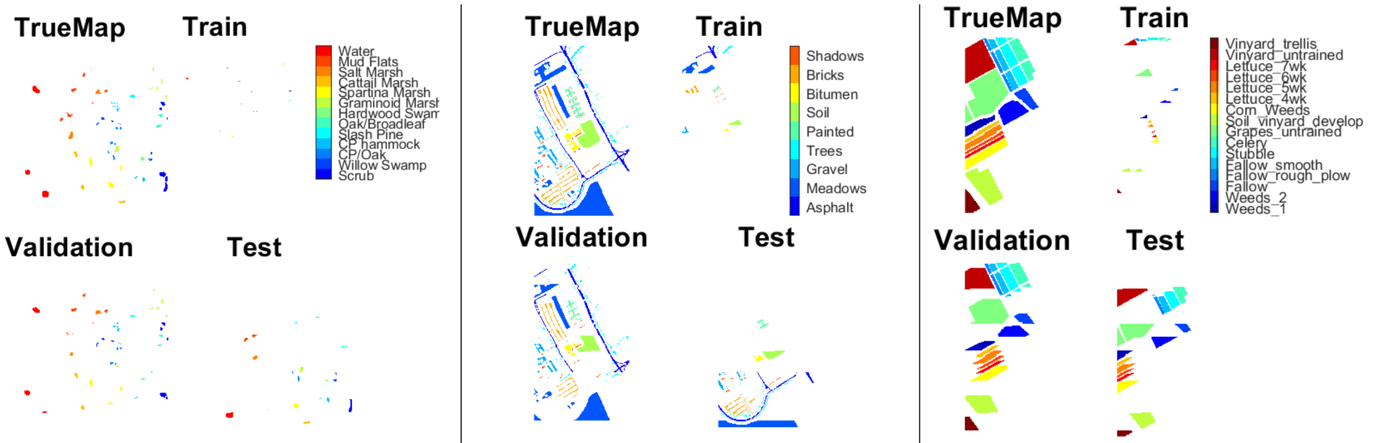
Pavia University (PU) data cube acquired by Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia Northern Italy with a geometric resolution of 1.3 meters. PU consists of 103 spectral bands and 610×610 spatial lines (spatial pixels), however, some of the samples contain no information and thus have to be discarded before the experiments. In total, the PU cube contains samples of 9 different classes, i.e., ground truths consist of 9 classes.

B. Experimental Settings

There are many ways to analyze the performance of any classification model such as overall (OA), average (AA), and

TABLE III: Detailed Description of Experimental Datasets along with the class names and number of disjoint samples used to train/validation/test 3D-CNN model. The percentages are as follows: 5% disjoint training samples, 65% disjoint validation map, and 35% disjoint Test samples.

Pavia University (PU)		Kennedy Space Center (KSC)		Salinas (SA)	
Class	**Tr/Val/Te**	**Class**	**Tr/Val/Te**	**Class**	**Tr/Val/Te**
Shadows	95/568/284	Swap	5/63/37	Lettuce romaine 6wk	45/550/321
Bitumen	133/798/399	Oak/Broadleaf	8/97/56	Lettuce romaine 4wk	53/641/374
Painted metal sheets	134/807/404	Hardwood	12/137/80	Lettuce romaine 7wk	53/642/375
Gravel	210/1259/630	Willow swamp	12/146/85	Fallow rough plow	70/836/488
Trees	307/1838/919	Slash pine	12/154/90	Vinyard vertical trellis	91/1084/632
Self-Blocking Bricks	368/2209/1105	CP hammock	13/151/88	Lettuce romaine 5wk	97/1156/674
Bare Soil	503/3017/1509	Graminoid marsh	21/259/151	Fallow	98/1186/692
Asphalt	663/3979/1989	Salt marsh	21/251/147	Brocoli green weeds 1	101/1205/703
Meadows	1865/11189/5595	Cattail marsh	21/242/141	Fallow smooth	134/1607/937
		Mud flats	25/302/176	Corn senesced green weeds	164/1967/1147
		Spartina marsh	26/312/182	Celery	179/2147/1253
		Scrub	38/457/266	Brocoli green weeds 2	186/2236/1304
		Water	47/556/324	Stubble	198/2375/1386
				Soil vinyard develop	310/3722/2171
				Vinyard untrained	363/4361/2544
				Grapes untrained	563/6763/3945



(a) KSC Ground Truths.

(b) Pavia University Ground Truths.

(c) Salinas Ground Truths.

Fig. 4: Geographical maps of true ground truths, disjoint training map (5%), disjoint validation map (65%), and 35% disjoint Test samples respectively. The number of training, validation, and test samples, class names, and percentages of training/validation/test samples are presented in Table I.

kappa (κ) coefficient along with several other statistical tests. OA tells us more about which samples are mapped correctly and is usually computed in percentage. OA is easy to compute and understand, however only provides the map user and producer with basic classification information. Whereas, the Kappa (κ) coefficient is generated from the statistical test to evaluate the classification accuracy. κ coefficient evaluates how well the classification model performed as compared to the random values, for instance, the κ coefficient varies between -1 to 1 in which -1, 0, and 1 indicate the classification is significantly worse than random, equal to or better than random, respectively. The κ coefficient is computed as follows:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{6}$$

where p_o and p_e present the OA accuracy and measures of the agreement among the actual and predicted class labels as it happening by chance. Moreover, $p_o - p_e$ accounts for the difference between the observed OA accuracy of the model as well as the OA accuracy obtained by chance. $1 - p_e$ computes the maximum value for this difference. For any model to be considered as good, the maximum and observed difference must need to be close to each other, thus $\kappa = 1$. However, for a random model, the numerator turns to 0 thus $\kappa = 0$ or maybe negative. Therefore, in this case, the OA accuracy of the model will be even lower than what could have been obtained by a random guess.

In all the experiments, we started evaluating all the conventional as well as the state-of-the-art models with 5% of randomly selected samples, and then in each iteration, 200 samples have been selected using Fuzziness, Mutual Information, and Breaking Ties based sample selection methods until 2000 samples have been selected. For long, different variants of CNN have been used and proposed for HSIC, however, CNN requires a large number of labeled training samples for learning, whereas the collection of such a huge number of labeled training samples is challenging for HSI datasets, due to overlapping and nested regions, human efforts, and time in many real problems. Moreover, the limited availability of training samples deters the classification performance. Therefore, to get higher accuracy, an appropriate number of training samples are required and are considered an important factor for classification performance.

We intentionally did not select 1-4% of training samples because there are some classes in all the datasets which have quite a lower number of samples, thus will bring only 1 or 2 samples from such classes. In the meantime some classes have 100's or 1000's samples which at the same time have more information, this can lead to the class imbalance issue, which is not the problem under investigation in this work. There is an option to avoid such a problem is to select the number of training samples rather than selecting the percentage of randomly selected training samples. Thus, with any of the above options, one can opt for a re-examination of the work.

For experimental results, a 3D-CNN architecture is adopted. The details of 3D convolutional layers are as follows: $layer\_1 = 60 \times 3 \times 3 \times 7$ i.e. $K_1^1 = 3, K_2^1 = 3$ and $K_3^1 = 7$. $layer\_2 = 30 \times 3 \times 3 \times 5$ i.e. $K_1^2 = 3, K_2^2 = 3$ and $K_3^2 = 5$. $layer\_3 = 10 \times 3 \times 3 \times 3$ i.e. $K_1^3 = 3, K_2^3 = 3$ and $K_3^3 = 3$. In total, three convolutional layers are stacked for low and high-level feature learning i.e., to increase the number of spatial-spectral feature maps and to distinguish the spatial/spectral features while preserving the spectral information. The convolutional process produces zero filling thus it does not require the use of batch normalization or data enhancement. Moreover, the weights are initially randomized and later optimized using backpropagation with Adam optimizer using softmax loss function. The entire network is trained over 50 epochs using a mini-batch of 256. The rest of the competing methods have been implemented as per the settings mentioned in their respective works (i.e, MLP [69], MLR [70], RF [71], SVM [72], 1D CNN [73] and 2D CNN [74]).

C. Experimental Results and Discussion

This section presents the experimental results and a discussion on the obtained results with possible pros and cones. The obtained accuracies are from disjoint training, disjoint validation, disjoint test, and complete (as similar to the traditional works published in the literature) datasets. The obtained accuracies for disjoint validation, disjoint test, and complete dataset as test are shown in Figures 5, 6 7, and 8.

The comparative methods mostly misclassify samples having similar spatial structures (i.e., Meadows and Bare Soil classes for Pavia University dataset) as shown in Figure 7. Moreover, the overall accuracy for Grapes Untrained is lower as compared to other classes due to the reasons mentioned above. In a nutshell, it can be said that higher accuracy can be achieved using more number of labeled training samples as shown in Figure 5, Therefore a higher number of labeled training samples (not as high as claimed in the literature, i.e., only a few carefully selected new samples can produce better/higher accuracy as compared to the bulk amount of randomly selected samples) produces better accuracy for all competing methods. Generally, we pay much attention to the accuracy only while considering the limited availability of training samples, however, the computational time is also quite important especially when one deals with deep models. Thus, the higher accuracies of a generalized model trained on a limited number of training samples in less computational time could be considered an innovative and important contribution to the domain.

Figure 5 presents the classification performance in terms of OA, AA, and κ accuracy with different numbers of training samples selected using Fuzziness, Breaking Ties, and Mutual Information based sample selected methods, respectively for disjoint validation, disjoint test, and complete dataset as a test set. As earlier explained, initially 5% of randomly selected training samples are used and in each iteration and 200 samples are systematically selected using all three sample selection methods. One can observe from the results as the number of training samples increases, the classification performance improves to a certain number, and then got stable. This is because there is no new information added in the training samples, thus, only the redundancy is being increased rather the information, i.e., the new samples added into the training set are either geographically similar or have similar patterns.

The experimental results in Figures 5, 6 7, and 8 show the quality of spectral-spatial features learned by 3D CNN and active learning framework. To observe the number of training samples required to train a 3D CNN model with or without active learning, 5 to 7 iterations are enough as shown in Figure 5. All the experimental results explained in this work are obtained using $9 \times 9 \times B$ spatial dimensions, and all other training parameters remain the same except for a number of training samples in each iteration which have already been discussed in detail. Moreover, from a computational time point of view, a detailed discussion has been done in the former sections, however, similar to accuracy trends (i.e., gradually increasing), the computational time is also increasing.

D. Statistical Tests and Computational Time

Overall, Average, Kappa (κ) accuracies may not be the only good measures especially when the datasets are not balanced i.e., with different numbers of samples in each class. Let us see an example to understand it. Let us consider a case where 10 individuals are not healthy (i.e., have some disease, $+ve$ class) and 90 healthy individuals ($-ve$ class). Moreover, assume that the machine learning model correctly predicts 90% individuals as healthy, however, it also predicts the unhealthy people as healthy. What will be the best accuracy in this case?

Thus, there are 90, 0, 10, and 0 samples are identified as "True Negative", "False Positive", "False Negative", and "True Positive", respectively. Thus, in this case, the accuracy is 90% i.e., $\frac{90+0}{100} = 0.9$. As identified, the accuracy is 90% however, the model is highly biased since all 10 individuals who are not healthy are predicted as healthy, i.e., only accuracy measure in a such scenario can be misleading or maybe misinterpret the results. Thus, accuracy is not the only measure or maybe not the best measure the evaluate a machine learning model. On top of accuracy, statistical analyses are worth discussing to validate any machine learning model.

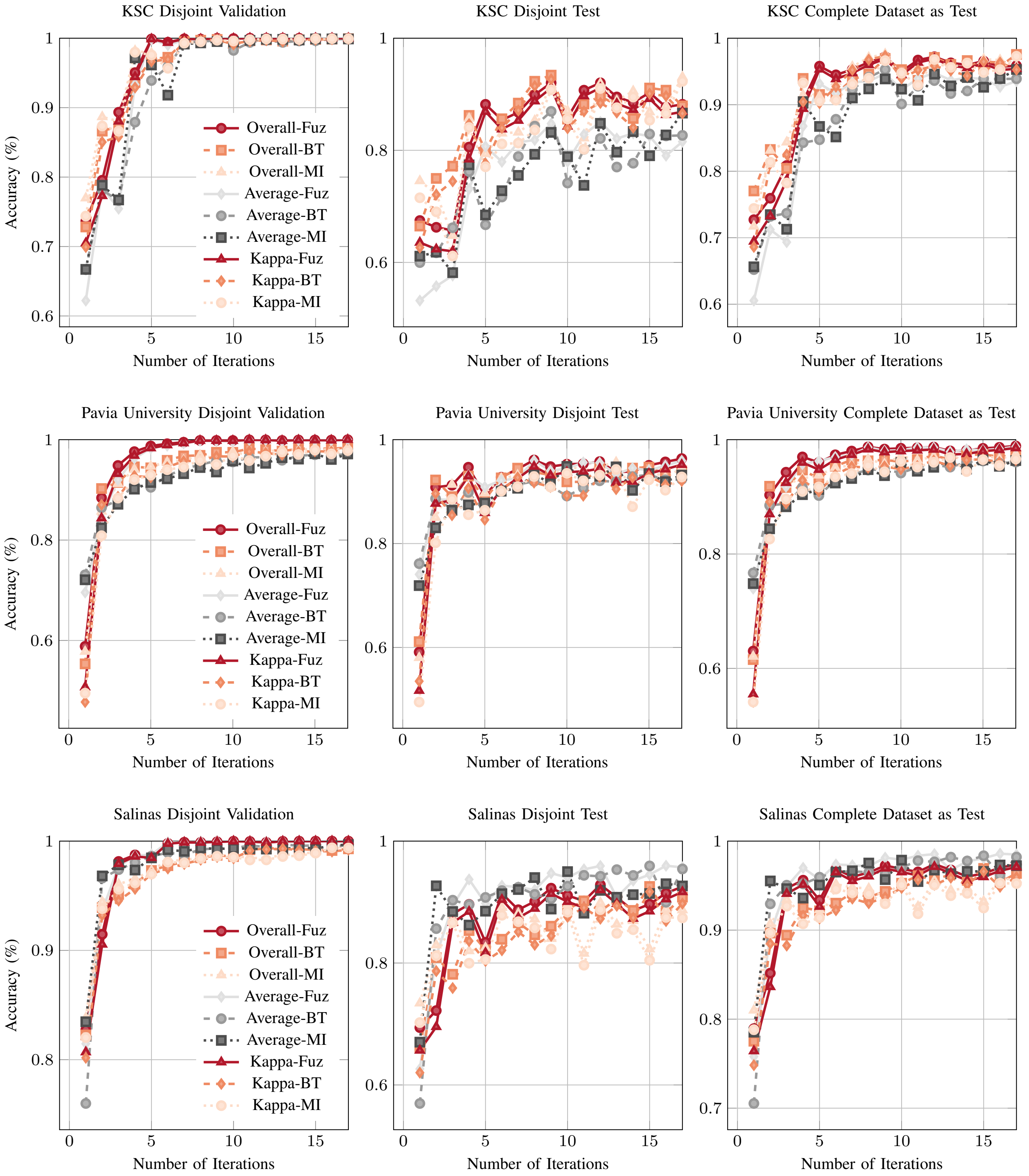
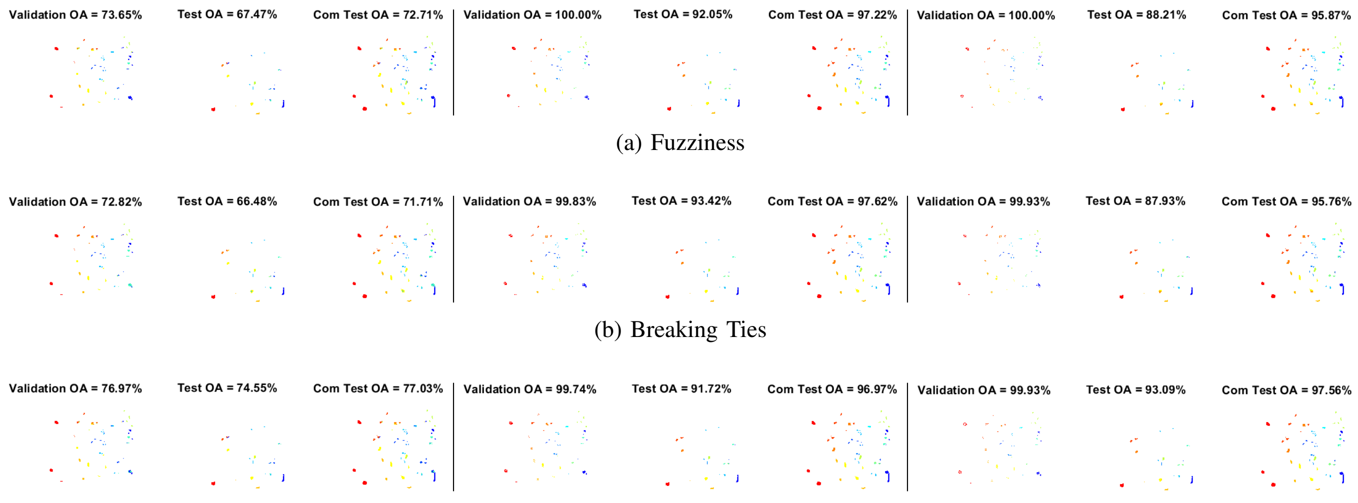


Fig. 5: Overall, Average and kappa accuracy with different percentages of training samples selected in each iteration from Kennedy Space Center, Pavia University, and Salinas datasets. It is perceived from the above figure that by including the samples back to the training set, the classification results are significantly improved. Moreover, it can be seen that Fuzziness-based samples selection method is more robust than Breaking Ties, and Mutual Information sampling criteria. Furthermore, it is clear from the results, the disjoint test samples produce lower accuracies than the ones obtained on complete datasets.

Many metrics can be used to validate the results, and from those, precision (Positive predictive values), recall (sensitivity or true positive rate), and F1 score (both precision and recall are considered) are considered in this research. Precision should be 1.00 for the ideal classification model; happens only once the denominator and numerator come equal, i.e.,

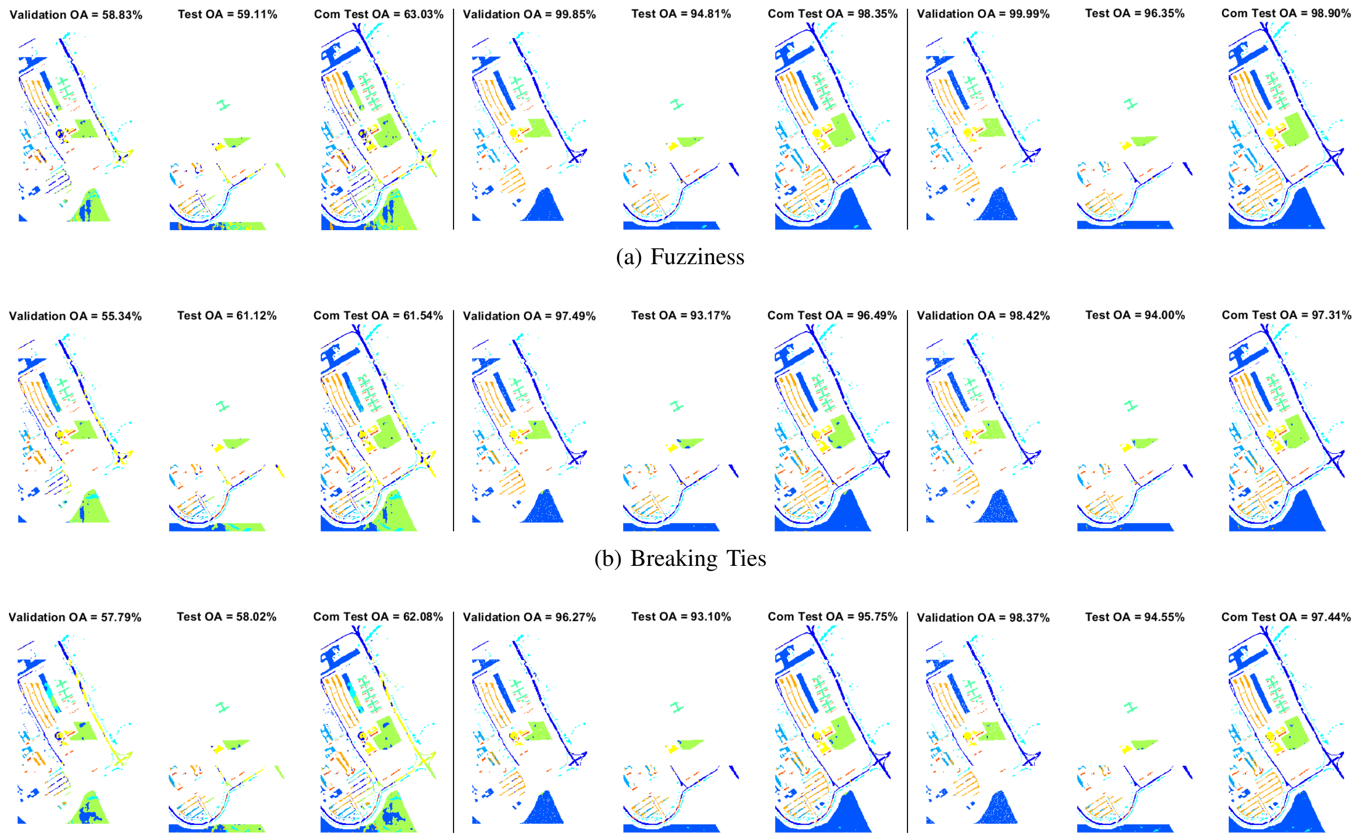


(a) Fuzziness

(b) Breaking Ties

(c) Mutual Information

Fig. 6: Geographical maps for KSC dataset for 1st, 9th, and 17th iteration.



(a) Fuzziness

(b) Breaking Ties

(c) Mutual Information

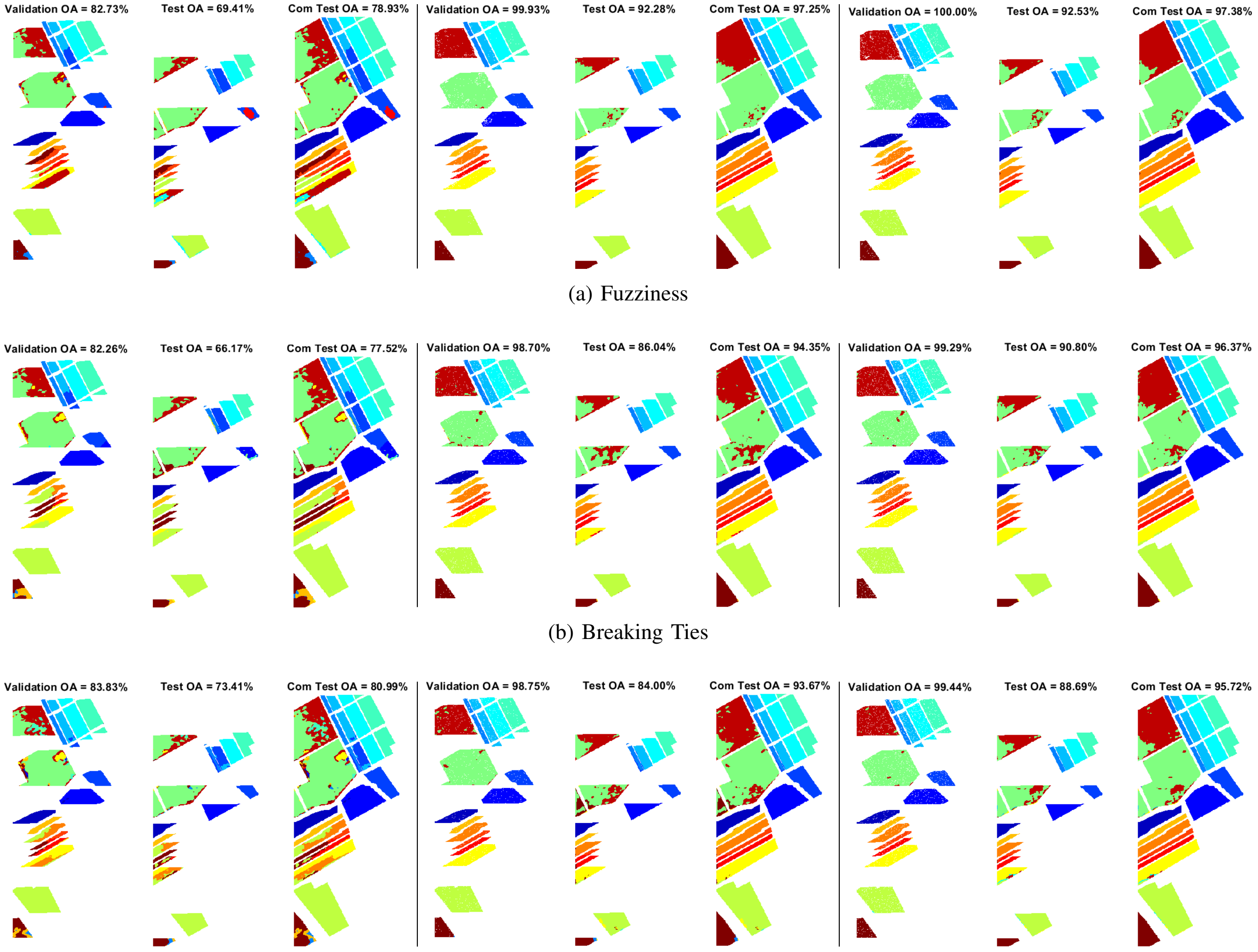
Fig. 7: Geographical maps for Pavia University dataset for 1st, 9th, and 17th iteration.

true positive (TP) = TP + false positive (FP), in such case FP becomes zero. However, as FP increases, the overall precision decreases which reflect an inappropriate classification model. Similar behavior is suggested for Recall where only False negative (FN) is replaced with FP. Precision and recall are defined as follows:

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

In a nutshell for a good classifier, both recall and precision need to be high i.e., both FN and FP needs to be quite low in value. Thus, on top of precision and recall, one needs to have an F1 score that considered both precision and recall at the same time and provides more insight into a classifier's



(a) Fuzziness

(b) Breaking Ties

(c) Mutual Information

Fig. 8: Geographical maps for Salinas dataset for 1st, 9th, and 17th iteration.

generalization performance and statistical significance. F1 score can be computed as follows:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

The higher the values of precision, recall, and F1 score the better the classification model is. Moreover, these measures are way better than only accuracy to justify the performance of any proposed method. The statistical performance of our proposed method is presented in Table IV. The results presented in Table IV show the statistical significance of our proposed method and have achieved above 90% in most of the cases. To avoid paper over-length issues, we only presented the statistical results of the Fuzziness-based sample selected method, however, breaking ties, and mutual information-based sample selection methods do produce the same results.

Figure 9 presents the computational time to process/evaluate the Hyperspectral datasets used in this study. As shown in the figure, the training time gradually increases as the number of training samples increases however the increment in the training in each iteration is significantly lesser than what is needed to train a 3D Convolutional Neural Network (CNN). This is due to the concept of fine-tuning rather than retraining the model from scratch, this work proposed the idea to fine-tune i.e., instead of training the entire model, the last layers are fine-tuned with new parameters along with the weights frozen in the previous iteration. unlike the training time, validation and testing times are more stable.

E. Comparison with State-of-the-art

This section presents a detailed discussion of experimental results obtained as compared to the state-of-the-art works published in recent years. Most of the research carried out in recent years presents comprehensive experimental results to pin the advantages/disadvantages of their works. However, to some extent, the experimental results presented in the literature may have adopted different experimental protocols such as randomly selected training, validation, and test samples may have the same percentage but may have different geographical locations of each model as those models have been trained, validated, and tested in different times (the comparative models have been executed in multiple times, one after each other,

TABLE IV: Precision, Recall and F1 Score test for each iteration. The higher the values of precision, recall and F1 scores, the better the performance, generalization, and statistical significance is.

Iteration #	Kennedy Space Center Dataset			Pavia University Dataset			Salinas Dataset		
	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score
1	0.6054±0.06	0.6087±0.05	0.5709±0.05	0.7395±0.04	0.6863±0.05	0.6698±0.03	0.7579±0.03	0.8029±0.02	0.7396±0.02
2	0.7121±0.04	0.7472±0.04	0.6879±0.04	0.8729±0.02	0.8867±0.01	0.8741±0.01	0.8962±0.02	0.9144±0.02	0.8841±0.02
3	0.6933±0.06	0.7102±0.07	0.6623±0.05	0.9176±0.01	0.9172±0.02	0.9168±0.02	0.9489±0.01	0.9615±0.01	0.9533±0.01
4	0.8678±0.03	0.8858 ±0.02	0.8639±0.02	0.9572±0.01	0.9494±0.01	0.9524±0.01	0.9699±0.01	0.9727±0.01	0.9708±0.01
5	0.9316±0.02	0.9304±0.02	0.9222±0.01	0.9626±0.01	0.9373±0.01	0.9489±0.01	0.9569±0.01	0.9409±0.01	0.9459±0.01
6	0.9168±0.02	0.9215±0.02	0.9127±0.01	0.9696±0.01	0.9638±0.01	0.9665±0.01	0.9738±0.01	0.9789±0.01	0.9757±0.01
7	0.9335±0.02	0.9377±0.02	0.9278±0.01	0.9735±0.01	0.9622±0.01	0.9673±0.01	0.9728±0.01	0.9766±0.01	0.9739±0.01
8	0.9356±0.02	0.9422±0.02	0.9278±0.02	0.9867±0.01	0.9737±0.01	0.9798±0.01	0.9717±0.01	0.9772±0.01	0.9737±0.01
9	0.9468±0.02	0.9535±0.01	0.9453±0.01	0.9807±0.01	0.9691±0.01	0.9746±0.01	0.9814±0.01	0.9846±0.01	0.9826±0.01
10	0.9227±0.02	0.9219±0.02	0.9146±0.01	0.9814±0.01	0.9756±0.01	0.9784±0.01	0.9790±0.01	0.9808±0.01	0.9797±0.01
11	0.9393±0.02	0.9434±0.02	0.9334±0.01	0.9864±0.01	0.9766±0.01	0.9812±0.01	0.9836±0.01	0.9776±0.01	0.9803±0.01
12	0.9478±0.01	0.9508±0.02	0.9437±0.01	0.9863±0.0	0.9806±0.01	0.9834±0.01	0.9854±0.01	0.9882±0.01	0.9865±0.01
13	0.9368±0.01	0.9387±0.02	0.9317±0.01	0.9788±0.01	0.9697±0.01	0.9741±0.01	0.9676±0.01	0.9724±0.01	0.9688±0.01
14	0.9407±0.01	0.9408±0.02	0.9339±0.01	0.9829±0.01	0.9530±0.02	0.9654±0.01	0.9773±0.01	0.9726±0.01	0.9746±0.01
15	0.9386±0.02	0.9430±0.02	0.9330±0.01	0.9816±0.01	0.9794±0.01	0.9803±0.01	0.9807±0.01	0.9766±0.01	0.9785±0.01
16	0.9267±0.02	0.9298±0.02	0.9201±0.02	0.9841±0.01	0.9816±0.01	0.9828±0.01	0.9859±0.01	0.9837±0.01	0.9847±0.01
17	0.9355±0.02	0.9409±0.02	0.9301±0.01	0.9877±0.01	0.9749±0.01	0.9810±0.01	0.9849±0.01	0.9878±0.01	0.9862±0.01

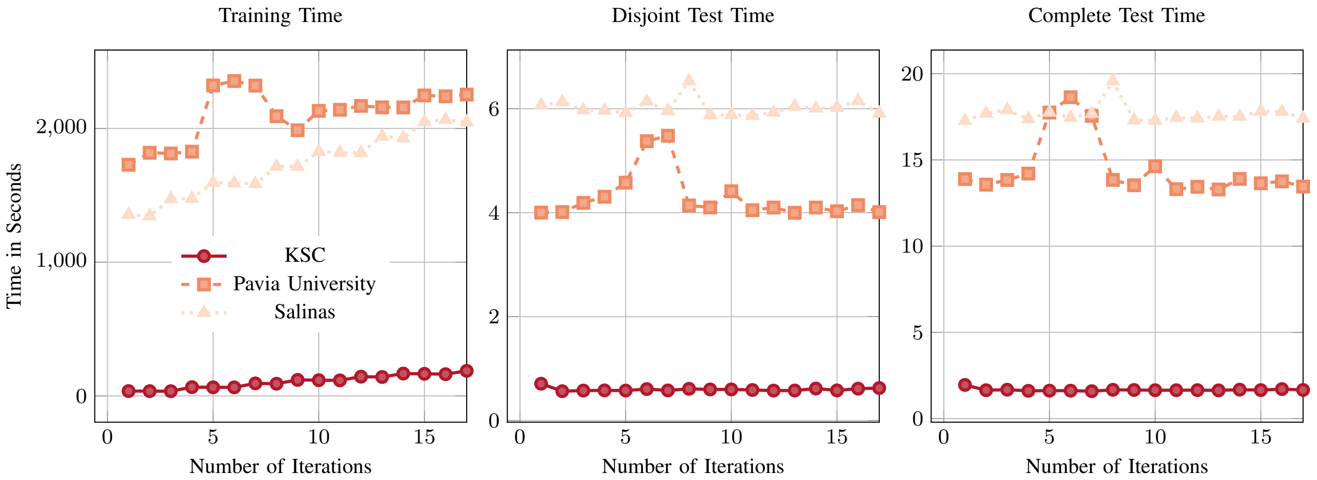


Fig. 9: Training, Disjoint Test, and Complete dataset test time for Kennedy Space Center, Pavia University, and Salinas Time. The training time is significantly lower than the usual 3D CNN training time because the proposed model adopts the fine-tuning process rather than retraining the entire network for each iteration i.e., the last few classification layers are retrained rather than the entire network.

or in parallel which brings a new set of training, validation, and test samples with the same number or percentage) as initial samples have been chosen randomly [75]. Therefore, to make the comparison fair between the works proposed in the literature and current, one must need to have the same experimental settings and must need to be executed with the same set of training, validation, and test samples.

Another issue with most of the literature proposed in recent years is overlapping training/test samples. As the training/validation samples are randomly selected (including or excluding the above point) the data split contains overlapping samples. This results in a biased model (as overlapping means the model has already seen the training and validation samples) and produces higher accuracy. To avoid it from happening, this study ensures that although the samples are chosen randomly, the intersection between training, test, and validation samples remains empty and constant for all competing methods.

The proposed fast and compact 3D CNN with an active transfer learning method has been compared with several state-of-the-art methods. The comparative methods includes Multi-layer Perceptron (MLP) [69], Multinomial Logistic Regression (MLR) [70], Random Forest (RF) [71], Support Vector Machine (SVM) [72], 1D CNN [73] and 2D CNN [74]. All these methods are retrained using a fuzziness-based sample selection method to make the comparison fair and reliable. The comparative models have been implemented as per the parameters explained in the cited works. The detailed experimental results are enlisted in Tables V and VI. Focusing on the Salinas dataset, one can see that the performance of the pixel-wise classifiers such as RF and MLR provide lower accuracy but better than SVM. However, the spectral classifier such as 1D CNN is way better than other spectral classifiers, whereas the spatial classifier, for instance, 2D CNN produces much better results than SVM, RF, MLP, MLR, and 1d CNN method, but underperforms spatial-spectral classifier i.e., 3D CNN. From these results, one can observe that after adding spectral-spatial information, the classifier significantly improves the accuracy as compared to the individual information, i.e., alone spectral

TABLE V: Salinas Dataset: Average accuracy for 17 iterations and in each iteration 200 samples are selected using the predefined sample selection method with 5% initially randomly selected training samples. The comparative methods includes MLP [69], MLR [70], RF [71], SVM [72], 1D CNN [73] and 2D CNN [74]. All these methods are retrained using a fuzziness-based sample selection method to make the comparison fair and reliable.

Class	Tr/Val/Te Samples	MLP Fuz	MLR Fuz	RF Fuz	SVM Fuz	1D CNN Fuz	2D CNN Fuz	3D CNN Fuz	3D CNN BT	3D CNN MI
Brocoli green weeds 1	101/1205/703	98.16±1.94	98.26±0.61	97.71±1.94	97.59±1.31	99.00±0.42	99.85±0.15	99.90±0.01	99.95±0.01	99.99±0.01
Brocoli green weeds 2	186/2236/1304	99.48±0.40	99.78±0.07	99.83±0.07	99.35±0.45	99.55±0.00	94.15±1.45	98.85±0.03	99.81±0.01	99.51±0.02
Fallow	98/1186/692	96.89±1.76	94.94±1.82	93.74±3.59	96.88±2.08	97.79±0.56	99.62±0.03	97.82±0.03	97.59±0.01	99.32±0.03
Fallow rough plow	70/836/488	99.44±0.31	99.24±0.38	97.06±3.00	98.98±0.61	98.76±0.96	99.86±0.14	96.55±0.04	98.32±0.01	95.71±0.05
Fallow smooth	134/1607/937	97.50±1.15	97.36±1.21	96.25±0.99	97.87±0.72	96.98±1.18	99.79±0.06	95.71±0.02	94.05±0.08	96.00±0.03
Stubble	198/2375/1386	99.52±0.22	99.57±0.18	98.73±0.99	99.43±0.40	99.80±0.13	99.73±0.21	99.92±0.01	99.90±0.01	99.99±0.01
Celery	179/2147/1253	99.27±0.33	99.66±0.16	99.09±0.41	99.44±0.21	99.68±0.09	99.09±0.15	99.84±0.01	99.91±0.01	99.77±0.01
Grapes untrained	563/6763/3945	81.16±5.33	81.89±3.01	81.85±2.60	97.53±1.78	83.43±3.15	92.31±1.01	91.96±0.12	88.55±0.05	87.18±0.06
Soil vinyard develop	310/3722/2171	99.34±0.43	99.86±0.07	98.90±0.44	99.39±0.52	99.26±0.43	99.84±0.06	98.75±0.02	98.96±0.02	98.98±0.01
Corn senesced green weeds	164/1967/1147	89.33±2.19	88.50±2.12	85.53±1.96	91.13±1.74	93.49±2.15	96.19±2.81	89.84±0.05	91.43±0.05	92.23±0.02
Lettuce romaine 4wk	53/641/374	90.02±3.76	91.95±3.05	88.16±4.53	93.93±1.83	94.48±1.99	96.82±0.84	98.12±0.07	99.82±0.01	99.94±0.01
Lettuce romaine 5wk	97/1156/674	97.21±2.40	99.03±0.73	97.19±1.37	99.14±0.56	99.97±0.05	99.82±0.18	94.70±0.01	94.26±0.02	96.22±0.03
Lettuce romaine 6wk	45/550/321	97.66±1.32	94.39±8.09	97.79±1.37	97.39±2.38	98.25±0.62	98.42±1.15	90.62±0.17	89.20±0.07	83.88±0.10
Lettuce romaine 7wk	53/642/375	91.38±2.33	92.26±1.34	90.88±3.21	91.92±3.07	91.03±1.75	96.82±0.00	94.86±0.03	93.36±0.01	95.94±0.01
Vinyard untrained	363/4361/2544	64.87±8.76	60.89±3.55	59.21±4.36	64.20±2.91	66.41±7.54	84.74±1.35	85.69±0.02	79.83±0.08	81.15±0.03
Vinyard vertical trellis	91/1084/632	96.36±1.20	95.29±2.48	92.92±2.26	96.70±1.84	98.34±0.57	85.78±0.39	98.55±0.01	96.77±0.01	96.93±0.03
OA		89.57±0.41	89.20±0.30	88.22±0.29	91.07±0.37	90.85±0.77	94.95±0.07	98.17±0.06	97.13±0.04	97.24±0.04
AA		93.60±0.56	93.30±0.59	92.18±0.28	94.43±0.38	94.79±0.64	96.43±0.23	98.22±0.04	97.61±0.06	98.00±0.04
κ		88.38±0.47	89.20±0.33	86.86±0.33	90.03±0.41	90.85±0.87	94.95±0.08	97.96±0.05	96.79±0.05	96.92±0.04

TABLE VI: Kennedy Space Center Dataset: Average accuracy for 17 iterations and in each iteration 200 samples are selected using the predefined sample selection method with 5% initially randomly selected training samples. The comparative methods includes MLP [69], MLR [70], RF [71], SVM [72], 1D CNN [73] and 2D CNN [74]. All these methods are retrained using a fuzziness-based sample selection method to make the comparison fair and reliable.

Class	Tr/Val/Te Samples	MLP Fuz	MLR Fuz	RF Fuz	SVM Fuz	1D CNN Fuz	2D CNN Fuz	3D CNN Fuz	3D CNN BT	3D CNN MI
Scrub	38/457/266	96.35±0.79	95.90±0.87	94.95±1.39	95.32±1.44	97.33±0.16	95.93±0.53	98.66±0.04	97.53±0.08	98.74±0.04
Willow swamp	12/146/85	89.63±4.04	88.81±1.75	87.94±1.68	94.49±3.20	93.42±1.16	87.65±0.82	74.77±0.11	75.11±0.10	76.74±0.11
Slash pine	12/154/90	91.52±2.46	87.97±4.75	89.49±2.29	91.88±1.47	86.85±8.15	86.72±0.00	82.61±0.18	73.69±0.22	75.53±0.24
CP hammock	13/151/88	75.32±6.34	67.70±11.4	75.60±2.71	78.25±4.55	83.86±9.71	89.29±0.40	89.24±0.19	90.78±0.11	84.85±0.18
Oak/Broadleaf	8/97/56	66.58±7.63	62.11±8.24	59.25±6.92	75.03±4.73	70.60±6.92	97.83±2.17	63.17±0.25	53.71±0.31	63.94±0.24
Hardwood	12/137/80	69.74±4.59	71.35±4.48	58.12±6.80	80.39±5.88	83.11±2.09	94.10±3.28	89.42±0.21	89.13±0.11	91.78±0.17
Swap	5/63/37	87.81±5.32	84.19±5.51	85.90±4.74	88.19±4.05	92.38±6.07	78.57±0.48	83.42±0.31	84.54±0.17	77.76±0.36
Graminoid marsh	21/259/151	93.76±1.81	90.35±1.26	87.24±2.12	94.99±3.25	95.13±1.00	89.91±1.04	94.54±0.11	95.62±0.09	95.71±0.08
Spartina marsh	26/312/182	97.58±0.89	96.81±0.63	93.65±3.03	97.58±0.93	98.65±0.16	96.54±0.38	86.72±0.19	95.38±0.06	89.06±0.11
Cattail marsh	21/242/141	97.45±1.72	94.90±2.79	89.60±2.53	98.42±0.72	97.69±1.69	96.91±1.86	98.57±0.06	1.00±0.00	1.00±0.00
Salt marsh	21/251/147	98.07±1.23	96.95±0.71	97.42±0.97	98.00±0.99	98.65±0.92	98.21±0.12	95.28±0.19	91.42±0.21	95.35±0.18
Mud flats	25/302/176	94.63±1.16	92.41±1.19	90.97±1.98	95.84±1.40	96.69±1.53	94.73±3.28	94.35±0.18	94.50±0.11	96.83±0.06
Water	47/556/324	100.00±0.00	100.00±0.00	99.69±0.13	100.00±0.00	99.50±0.64	99.68±0.32	98.68±0.05	1.00±0.00	98.01±0.08
OA		93.14±0.49	91.49±0.43	89.99±0.28	94.40±0.50	94.84±0.21	94.77±0.47	96.27±0.08	95.96±0.07	96.67±0.06
AA		89.11±0.74	86.88±0.60	85.37±0.72	91.41±0.92	91.83±0.29	92.77±0.41	94.58±0.11	93.72±0.10	94.33±0.10
κ		92.35±0.55	91.49±0.47	88.85±0.31	93.76±0.56	94.84±0.24	94.77±0.52	95.84±0.09	95.50±0.081	96.29±0.07

or spatial information. From the results, we can see that the 3D CNN classifier is able to attain good classification results with fewer training samples than other active transfer learning-based classifiers. A similar trend can be seen in the Kennedy Space center dataset.

V. Conclusion

Traditionally, Convolutional Neural Network (CNN) is trained on a large number of labeled training samples and tested on the entire HSI cube to generate accurate thematic maps which produce high accuracy. Indeed, this includes bias, as many of the test samples have already been seen by the model while training. However, in this work, a disjoint Train/Validation/Test samples split-based unified 3D CNN and Active Transfer Learning method is proposed. A 3D CNN model is initially trained with 5% labeled training samples and validated on 65% samples. In the next phase, high fuzziness magnitude, Mutual Information, and Breaking Ties-based 200 misclassified samples have been selected to include in the original training set to fine-tune the model rather than retraining from scratch to reduce the computational cost. To prove the superiority of our proposed method, three different types of experiments have been conducted as follows: 1): Disjoint train and validation test only, 2): Disjoint Train/Validation and Test set are all evaluated together, and finally, 3): Disjoint Train/Validation/Test and complete HSI cube as Test set to compare the experimental results of the disjoint test and complete HSI cube as a test set. The proposed model significantly improves the classification results as compared to the state-of-the-art models with a significantly fewer number of labeled training samples.

References

[1] D. Hong, W. He, N. Yokoya, J. Yao, L. Gao, L. Zhang, J. Chanussot, and X. Zhu, "Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, pp. 52–87, Jun. 2021.

[2] M. Ahmad, A. Khan, A. M. Khan, M. Mazzara, S. Distefano, A. Sohaib, and O. Nibouche, "Spatial prior fuzziness pool-based interactive classification of hyperspectral images," *Remote Sensing*, vol. 11, no. 9, May. 2019.

[3] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, Nov. 2022. doi: 10.1109/TGRS.2021.3130716.

[4] F. Xiong, J. Zhou, and Y. Qian, "Hyperspectral restoration via *l*_0 gradient regularized low-rank tensor factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10410–10425, 2019.

[5] R. Sarić, V. D. Nguyen, T. Burge, O. Berkowitz, M. Trtílek, J. Whelan, M. G. Lewsey, and E. Čustović, "Applications of hyperspectral imaging in plant phenotyping," *Trends in Plant Science*, 2022.

[6] H. Ayaz, M. Ahmad, A. Sohaib, M. N. Yasir, M. A. Zaidan, M. Ali, M. H. Khan, and Z. Saleem, "Myoglobin-based classification of minced meat using hyperspectral imaging," *Applied Sciences*, vol. 10, no. 19, p. 6862, 2020.

[7] H. Ayaz, M. Ahmad, M. Mazzara, and A. Sohaib, "Hyperspectral imaging for minced meat classification using nonlinear deep features," *Applied Sciences*, vol. 10, no. 21, p. 7783, 2020.

[8] M. Zulfiqar, M. Ahmad, A. Sohaib, M. Mazzara, and S. Distefano, "Hyperspectral imaging for bloodstain identification," *Sensors*, vol. 21, no. 9, p. 3045, 2021.

[9] A. ul Rehman and S. A. Qureshi, "A review of the medical hyperspectral imaging systems and unmixing algorithms' in biological tissues," *Photodiagnosis and Photodynamic Therapy*, vol. 33, p. 102165, 2021.

[10] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, pp. 3791–3808, Jun. 2020.

[11] Z. Saleem, M. H. Khan, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, "Prediction of microbial spoilage and shelf-life of bakery products through hyperspectral imaging," *IEEE Access*, vol. 8, pp. 176986–176996, 2020.

[12] M. H. Khan, Z. Saleem, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, "Hyperspectral imaging for color adulteration detection in red chili," *Applied Sciences*, vol. 10, no. 17, p. 5955, 2020.

[13] M. H. Khan, Z. Saleem, M. Ahmad, A. Sohaib, H. Ayaz, M. Mazzara, and R. A. Raza, "Hyperspectral imaging-based unsupervised adulterated red chili content transformation for classification: Identification of red chili adulterants," *Neural Computing and Applications*, pp. 1–15, 2021.

[14] M. Wang, Q. Wang, D. Hong, S. K. Roy, and J. Chanussot, "Learning tensor low-rank representation for hyperspectral anomaly detection," *IEEE Transactions on Cybernetics*, 2022. doi: 10.1109/TCYB.2022.3175771.

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

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

[17] F. Xiong, J. Zhou, S. Tao, J. Lu, and Y. Qian, "Snmf-net: Learning a deep alternating neural network for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[18] M. Kavitha, R. Gayathri, K. Polat, A. Alhudhaif, and F. Alenezi, "Performance evaluation of deep e-cnn with integrated spatial-spectral features in hyperspectral image classification," *Measurement*, vol. 191, p. 110760, 2022.

[19] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "Orsim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 5146–5158, 2019.

[20] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 302–306, 2019.

[21] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, J. Yao, A. M. Khan, M. Mazzara, S. Distefano, and J. Chanussot, "Hyperspectral image classification—traditional to deep models: A survey for future prospects," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 968–999, 2022.

[22] Z. Lei, Y. Zeng, P. Liu, and X. Su, "Active deep learning for hyperspectral image classification with uncertainty learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[23] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 60–88, 2020.

[24] B. Liu, A. Yu, P. Zhang, L. Ding, W. Guo, K. Gao, and X. Zuo, "Active deep densely connected convolutional network for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 42, no. 15, pp. 5915–5934, 2021.

[25] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, oct 2021.

[26] A. Tasissa, D. Nguyen, and J. M. Murphy, "Deep diffusion processes for active learning of hyperspectral images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 3665–3668, 2021.

[27] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao, "On the sampling strategy for evaluation of spectral-spatial methods in hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 862–880, 2017.

[28] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral-spatial classification of hyperspectral image using autoencoders," in *2013 9th International Conference on Information, Communications & Signal Processing*, pp. 1–5, 2013.

[29] M. Ahmad, M. A. Alqarni, A. M. Khan, R. Hussain, M. Mazzara, and S. Distefano, "Segmented and non-segmented stacked denoising autoencoder for hyperspectral band reduction," *Optik*, vol. 180, pp. 370–378, 2019.

[30] J. Zhao, L. Hu, Y. Dong, L. Huang, S. Weng, and D. Zhang, "A combination method of stacked autoencoder and 3d deep residual network for hyperspectral image classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 102, p. 102459, 2021.

[31] E. Pan, Y. Ma, X. Mei, F. Fan, and J. Ma, "Unsupervised stacked capsule autoencoder for hyperspectral image classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1825–1829, 2021.

[32] D. Hong, L. Gao, J. Yao, N. Yokoya, J. Chanussot, U. Heiden, and B. Zhang, "Endmember-guided unmixing network (egu-net): A general deep learning framework for self-supervised hyperspectral unmixing," *IEEE Transactions on Neural Networks and Learning Systems*, May 2021. doi: 10.1109/TNNLS.2021.3082289.

[33] M. Ahmad, A. M. Khan, M. Mazzara, and S. Distefano, "Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification," in *VISAPP'19*, 2019.

[34] M. Ahmad, "Ground truth labeling and samples selection for hyperspectral image classification," *Optik*, vol. 230, p. 166267, 2021.

[35] Y. Yuan, C. Wang, and Z. Jiang, "Proxy-based deep learning framework for spectral–spatial hyperspectral image classification: Efficient and robust," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[36] X. Wu, D. Hong, and J. Chanussot, "Convolutional neural networks for multimodal remote sensing data classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–10, 2022. doi: 10.1109/TGRS.2021.3124913.

[37] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, S. K. Roy, and X. Wu, "Hybrid dense network with attention mechanism for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3948–3957, 2022.

[38] M. Ahmad, S. Shabbir, R. A. Raza, M. Mazzara, S. Distefano, and A. M. Khan, "Artifacts of different dimension reduction methods on hybrid cnn feature hierarchy for hyperspectral image classification," *Optik*, vol. 246, p. 167757, 2021.

[39] V. Kumar, R. S. Singh, and Y. Dua, "Morphologically dilated convolutional neural network for hyperspectral image classification," *Signal Processing: Image Communication*, vol. 101, p. 116549, 2022.

[40] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 4340–4354, May 2021.

[41] R. Hang, X. Qian, and Q. Liu, "Cross-modality contrastive learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[42] R. Hang, Q. Liu, and Z. Li, "Spectral super-resolution network guided by intrinsic properties of hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 30, pp. 7256–7265, 2021.

[43] R. Hang, Z. Li, P. Ghamisi, D. Hong, G. Xia, and Q. Liu, "Classification of hyperspectral and lidar data using coupled cnns," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4939–4950, 2020.

[44] D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, and X. X. Zhu, "X-modalnet: A semi-supervised deep cross-modal network for classification of remote sensing data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 12–23, 2020.

[45] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected*

Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381–2392, 2015.

[46] J. Sun, Y. Cao, X. Zhou, M. Wu, Y. Sun, and Y. Hu, "Detection for lead pollution level of lettuce leaves based on deep belief network combined with hyperspectral image technology," *Journal of Food Safety*, vol. 41, no. 1, p. e12866, 2021.

[47] N. Balakrishnan, A. Rajendran, D. Pelusi, and V. Ponnusamy, "Deep belief network enhanced intrusion detection system to prevent security breach in the internet of things," *Internet of Things*, vol. 14, p. 100112, 2021.

[48] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.

[49] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sensing Letters*, vol. 7, no. 9, pp. 875–884, 2016.

[50] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Van-Den Hengel, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 36–43, 2015.

[51] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sensing Letters*, vol. 8, no. 5, pp. 438–447, 2017.

[52] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-d cnn for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[53] D. Hong, J. Yao, D. Meng, Z. Xu, and J. Chanussot, "Multimodal gans: Toward crossmodal hyperspectral–multispectral image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5103–5113, 2020.

[54] M. Ahmad, M. Mazzara, and S. Distefano, "Regularized cnn feature hierarchy for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 12, p. 2275, 2021.

[55] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, 2010.

[56] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, pp. 809–823, 2012.

[57] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new bayesian approach with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3947–3960, 2011.

[58] S. Sun, P. Zhong, H. Xiao, and R. Wang, "An mrf model-based active learning framework for the spectral-spatial classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 1074–1088, 2015.

[59] M. Ahmad, S. Protasov, A. M. Khan, R. Hussain, A. M. Khattak, and W. A. Khan, "Fuzziness-based active learning framework to enhance hyperspectral image classification performance for discriminative and generative classifiers," *PLoS ONE*, vol. 13, p. e0188996, January 2018.

[60] M. Ahmad, S. Shabbir, D. Oliva, M. Mazzara, and S. Distefano, "Spatial-prior generalized fuzziness extreme learning machine autoencoder-based active learning for hyperspectral image classification," *Optik*, vol. 206, p. 163712, 2020.

[61] M. Ahmad, M. Mazzara, R. A. Raza, S. Distefano, M. Asif, M. S. Sarfraz, A. M. Khan, and A. Sohaib, "Multiclass non-randomized spectral–spatial active learning for hyperspectral image classification," *Applied Sciences*, vol. 10, no. 14, p. 4739, 2020.

[62] M. Ahmad, "Fuzziness-based spatial-spectral class discriminant information preserving active learning for hyperspectral image classification," *arXiv preprint arXiv:2005.14236*, 2020.

[63] J. Li, "Active learning for hyperspectral image classification with a stacked autoencoders based neural network," in *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–4, 2015.

[64] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 712–724, 2017.

[65] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6440–6461, 2018.

[66] X. Li, Z. Cao, L. Zhao, and J. Jiang, "Alpn: Active-learning-based prototypical network for few-shot hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[67] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new bayesian approach with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3947–3960, 2011.

[68] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[69] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[70] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, 2010.

[71] Y. Zhang, G. Cao, X. Li, B. Wang, and P. Fu, "Active semi-supervised random forest for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 24, p. 2974, 2019.

[72] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[73] 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, 2021.

[74] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IGARSS*, pp. 4959–4962, IEEE, 2015.

[75] M. Ahmad, *Deep Learning for Hyperspectral Image Classification*. PhD thesis, Università DEGLI Studi DI Messina, 2021.

Muhammad Ahmad received his BS degree in Mathematics from GC University, Pakistan in 2007, MS degree in Electronics Engineering from International Islamic University, Islamabad, Pakistan in 2011, a Ph.D. degree in Computer Science and Engineering, from Innopolis University, Innopolis, Russia in 2019, and another Ph.D. degree in Cyber-Physical Systems from the University of Messina, Messina, Italy, in 2021.

He is currently working at the National University of Computer & Emerging Sciences (FAST-NUCES). He authored and co-authored over 70 scientific contributions to international journals, conferences, and books. He has delivered a number of invited and keynote talks and reviewed (reviewing) the technology-leading articles for journals. His research interest includes Hyperspectral Imaging, Remote Sensing, Machine Learning, Computer Vision, and Wearable Computing.

Usman Ghous completed his BS and MS degree in Computer Science from National University of Computer and Emerging Sciences (FAST) in 2016 and 2019 respectively. Currently, he is enrolled in PhD. Computer Science where his area of research is Hyperspectral Image classification and its problems. His alternate research interests includes mutation testing in deep neural networks and Recommendation Systems.

Danfeng Hong (S'16–M'19–SM'21) received the M.Sc. degree (summa cum laude) in computer vision from the College of Information Engineering, Qingdao University, Qingdao, China, in 2015, the Dr. -Ing degree (summa cum laude) from the Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich, Germany, in 2019.

He is currently a Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences (CAS). Before joining CAS, he has been a Research Scientist and led a Spectral Vision Working Group at the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He was also an Adjunct Scientist at GIPSA-lab, Grenoble INP, CNRS, Univ. Grenoble Alpes, Grenoble, France. His research interests include signal / image processing, hyperspectral remote sensing, machine / deep learning, artificial intelligence, and their applications in Earth Vision.

Dr. Hong is an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing (TGRS), an Editorial Board Member of Remote Sensing, an Editorial Advisory Board Member of ISPRS Journal of Photogrammetry and Remote Sensing. He was a recipient of the Best Reviewer Award of the IEEE TGRS in 2021 and 2022, and the Best Reviewer Award of the IEEE JSTARS in 2022, the Jose Bioucas Dias Award for recognizing the outstanding paper at WHISPERS in 2021, the Remote Sensing Young Investigator Award in 2022, the IEEE GRSS Early Career Award in 2022, and a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters) in 2022.

Adil Mehmood Khan received his B.S. degree in Information Technology from the National University of Science and Technology (NUST), Pakistan, in 2005. He completed his M.S. leading to Ph.D. degree in Computer Engineering from Kyung Hee University, South Korea, in 2011. He is currently a Professor at the Institute of Artificial Intelligence and Data Science, Innopolis University, Russia, and at the Department of Computer Science, the University of Hull, United Kingdom. His research interests are machine learning, deep learning, continual learning, domain adaptation, and fair machine learning.

Jing Yao (M'22) received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 2021.

He is currently an Assistant Professor with the Key Laboratory of Computational Optical Imaging Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. From 2019 to 2020, he was a visiting student at Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich, Germany, and at the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He put his recent research interests on hyperspectral and multimodal remote sensing image analysis, mainly including optimization and deep learning-based methods for image processing and interpretation tasks.

He was the recipient of the Jose Bioucas Dias Award for recognizing the outstanding paper at the Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS) in 2021. He also serves as a Guest Editor of Remote Sensing.

Shaohua Wang received a Diploma degree in mathematics from Beijing University of Chemical Technology, Beijing, China, in 2006. He received Ph.D. degree in the field of Cartography and GIS from the University of Chinese Academy of Science in 2013. From 2013 to 2016, he was Postdoctoral Research Assistant at Institute of Geographic Sciences and Natural Resources Research, CAS. From 2016 to 2021, he worked as Postdoctoral Fellow in the University of California Santa Barbara, Arizonal State University, and University of Illinois Urbana-Champaign, respectively.

He is currently an Innovation Professor with and International Research Center of Big Data for Sustainable Development Goals and the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences (CAS). His major research interests include geospatial artificial intelligence, spatiotemporal big data analytics, high-performance geospatial computing and their applications in Earth Vision.

Jocelyn Chanussot (M'04–SM'04–F'12) received the M.Sc. degree in electrical engineering from the Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree from the Université de Savoie, Annecy, France, in 1998. Since 1999, he has been with Grenoble INP, where he is currently a Professor of signal and image processing. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning and artificial intelligence. He has been a visiting scholar at Stanford University (USA), KTH (Sweden) and NUS (Singapore). Since 2013, he is an Adjunct Professor of the University of Iceland. In 2015-2017, he was a visiting professor at the University of California, Los Angeles (UCLA). He holds the AXA chair in remote sensing and is an Adjunct professor at the Chinese Academy of Sciences, Aerospace Information research Institute, Beijing.

Dr. Chanussot is the founding President of IEEE Geoscience and Remote Sensing French chapter (2007-2010) which received the 2010 IEEE GRS-S Chapter Excellence Award. He has received multiple outstanding paper awards. He was the Vice-President of the IEEE Geoscience and Remote Sensing Society, in charge of meetings and symposia (2017-2019). He was the General Chair of the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS). He was the Chair (2009-2011) and Cochair of the GRS Data Fusion Technical Committee (2005-2008). He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society (2006-2008) and the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing (2009). He is an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing and the Proceedings of the IEEE. He was the Editor-in-Chief of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2011-2015) and an Associate Editor for IEEE Transactions on Image Processing. In 2014 he served as a Guest Editor for the IEEE Signal Processing Magazine. He is a Fellow of the IEEE, a member of the Institut Universitaire de France (2012-2017) and a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters).