Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.056318

ARTICLE

Improving Generalization for Hyperspectral Image Classification: The Impact of Disjoint Sampling on Deep Models

Muhammad Ahmad[1,*], Manuel Mazzara[2], Salvatore Distefano[3], Adil Mehmood Khan[4] and Hamad Ahmed Altuwaijri[5]

[1]Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot, 35400, Pakistan

[2]Institute of Software Development and Engineering, Innopolis University, Innopolis, 420500, Russia

[3]Dipartimento di Matematica e Informatica—MIFT, University of Messina, Messina, 98121, Italy

[4]School of Computer Science, University of Hull, Hull, HU6 7RX, UK

[5]Department of Geography, College of Humanities and Social Sciences, King Saud University, Riyadh, 11451, Saudi Arabia

*Corresponding Author: Muhammad Ahmad. Email: mahmad00@gmail.com

Received: 11 July 2024 Accepted: 11 August 2024 Published: 15 October 2024

ABSTRACT

Disjoint sampling is critical for rigorous and unbiased evaluation of state-of-the-art (SOTA) models e.g., Attention Graph and Vision Transformer. When training, validation, and test sets overlap or share data, it introduces a bias that inflates performance metrics and prevents accurate assessment of a model's true ability to generalize to new examples. This paper presents an innovative disjoint sampling approach for training SOTA models for the Hyperspectral Image Classification (HSIC). By separating training, validation, and test data without overlap, the proposed method facilitates a fairer evaluation of how well a model can classify pixels it was not exposed to during training or validation. Experiments demonstrate the approach significantly improves a model's generalization compared to alternatives that include training and validation data in test data (A trivial approach involves testing the model on the entire Hyperspectral dataset to generate the ground truth maps. This approach produces higher accuracy but ultimately results in low generalization performance). Disjoint sampling eliminates data leakage between sets and provides reliable metrics for benchmarking progress in HSIC. Disjoint sampling is critical for advancing SOTA models and their real-world application to large-scale land mapping with Hyperspectral sensors. Overall, with the disjoint test set, the performance of the deep models achieves 96.36% accuracy on Indian Pines data, 99.73% on Pavia University data, 98.29% on University of Houston data, 99.43% on Botswana data, and 99.88% on Salinas data.

KEYWORDS

Hyperspectral image classification; disjoint sampling; Graph CNN; spatial-spectral transformer

1 Introduction

Hyperspectral Imaging (HSI) plays a pivotal role in various domains such as remote sensing [1], earth observation [2,3], urban planning [4], agriculture [5,6], forestry [7], target/object detection

[8,9], mineral exploration [10], environmental monitoring [11,12], climate change [13] food processing, bakery products, bloodstain identification, and meat processing.

Hyperspectral (HS) remote sensing plays a crucial role in urban planning by providing detailed insights and tools for efficient and informed decision-making [14]. HS remote sensors capture and analyze high-resolution spectral data across numerous narrow and contiguous spectral bands, offering comprehensive information on the composition and characteristics of urban environments [15]. HS data enables precise identification and mapping of various urban land cover types, such as vegetation, impervious surfaces, and soil [16]. Additionally, HS Imagining (HSI) facilitates the detection and monitoring of changes in land use, vegetation health, and pollution levels within urban areas [17]. These capabilities enhance urban planners' ability to assess the impact of urbanization, analyze urban metabolism, and evaluate the effectiveness of sustainability measures. By covering the entire processing chain, from data acquisition to analysis, HS remote sensing serves as a valuable tool for urban planners seeking a deeper understanding of urban environments and their dynamics.

HSI presents both challenges and opportunities for effective classification [18,19]. In recent years, Convolutional Neural Networks (CNNs) [20], Attention Graph [21,22], and Spatial-Spectral Transformers [23,24] have demonstrated remarkable success in various tasks, prompting researchers to explore their potential in HSI analysis [25]. However, achieving robust and reliable classification results requires careful consideration of data sampling techniques [26]. Random sampling for data splitting can lead to several issues. It can result in non-representative training, validation, and test sets, causing models to overfit or underfit. Different random splits produce inconsistent results, making it hard to draw meaningful conclusions [27]. Random sampling offers no control over data distribution, introducing bias in imbalanced datasets [28]. It hinders the reproducibility of experimental results and limits the exploration of data relationships. To address these challenges, disjoint sampling is a crucial yet often overlooked consideration when evaluating spatial-spectral Hyperspectral Image Classification (HSIC) models. As demonstrated by the works [1,22,29–33], traditional evaluations using overlapping training and test samples can lead to biased results and unfair assessments of model performance.

Even though several methodologies meticulously employ disjoint sets for training and testing their models, there's a notable inconsistency in their approach when it comes to generating land-cover maps [22,30,32]. Specifically, many of these methods deviate from the disjoint sampling principle by utilizing the entire dataset for HSIC (Thematic Maps). This practice introduces a conflict between the reported accuracy and the methodology employed. To address this inconsistency, it is essential to advocate for the use of a disjoint test set exclusively for generating land-cover maps. By doing so, the evaluation process aligns more closely with the principles of unbiased model assessment. It ensures that the model is confronted with truly unseen data during the map generation phase, fostering a more accurate representation of its real-world performance.

Moreover, disjoint sampling is essential for training and evaluating deep models [34,35]. This method involves carefully selecting diverse and representative samples from various regions, land cover types, and environmental conditions to overcome biased or non-representative training data limitations [36]. It ensures the model learns robust features, enhancing classification performance and adaptability to unseen data. Additionally, disjoint sampling facilitates fair and accurate model evaluation by keeping training, validation, and testing samples separate [37]. Furthermore, disjoint sampling is crucial in training SOTA models for HSIC, notably for CNN and Spatial-Spectral Transformer-based models. It enhances generalization, ensures fair evaluation, and enables result

interpretability. The use of disjoint training, validation, and test samples is imperative in HSIC for various reasons, such as:

Unbiased Evaluation: It is crucial to evaluate HSIC models using completely separate and disjoint data for training, validation, and testing in order to properly assess a model's true ability to generalize to new unknown examples [1,29].

Preventing Data Leakage and Mitigating Overfitting: Maintaining disjoint samples for training, validation, and testing is crucial to obtaining an accurate evaluation of a model's true generalization performance [22,37]. Employing disjoint subsets of the data at each stage of model development is pivotal in augmenting generalization performance. Through iterative training on distinct partitions, the model is compelled to infer underlying patterns shared across diverse examples, rather than being influenced by potentially misleading idiosyncrasies within a single fixed training sample [38,39]. This practice discourages the memorization of irrelevant characteristics specific to individual data samples. Instead, it fosters the capability to effectively process a wider range of presentations, including both seen and unseen examples.

Therefore, considering the above, this paper made the following contributions:

1. This paper presents a novel approach for disjoint train, validation, and test splits for HSIC. Ensuring the disjoint splits eliminates data leakage between subsets, which can bias performance evaluations. The proposed technique provides a practical implementation for creating disjoint train, validation, and test splits from ground truth data. This allows researchers to obtain unbiased performance evaluations and reliable comparisons between HSIC models.
2. By offering a standardized approach for creating evaluation splits, the proposed technique enhances the reproducibility and transparency of HSIC research. It fosters a more rigorous and standardized evaluation of classification models. The source code can be accessed at: https://github.com/mahmad00/Disjoint-Sampling-for-Hyperspectral-Image-Classification (accessed on 14 June 2024).

2 Mathematical Formulation

Let's consider HSI composed of B spectral bands, each with a spatial resolution of $H \times W$ pixels. The HSI data cube, denoted as $X \in \mathbb{R}^{(H \times W \times B)}$, is initially partitioned into overlapping 3D patches [24,40]. Each patch is centered at a spatial location (α, β) and covers a spatial extent of $PS = S \times S$ (PS = Patch size) pixels across all B bands. The total number of 3D patches (N) extracted from X (i.e., $X \in \mathbb{R}^{(S \times S \times B)}$) is given by $(H - S + 1) \times (W - S + 1)$. A patch located at (α, β) is represented as $P_{\alpha,\beta}$ and spans spatially from $\alpha - \frac{S-1}{2}$ to $\alpha + \frac{S-1}{2}$ in width and $\beta - \frac{S-1}{2}$ to $\beta + \frac{S-1}{2}$ in height. The labeling of these patches is determined by the label assigned to the central pixel within each patch as described in Algorithm 1.

Algorithm 1: Create 3D HSI Patches

Input: HSI, GT, Patch Size (PS)
1 H, W, B ← dimensions of HSI;
2 margin ← PS / 2;
3 Pad ← pad HSI with zeros on all sides;
4 Cubes ← create an array of size (H × W, PS, PS, B);
5 Labels ← create an array of size (H × W);

(Continued)

Algorithm 1 (continued)

```
patchIndex ← 0;
for hh ← margin to hh + margin do
    for ww ← margin to ww + margin do
        cube ← select a sub-array from Pad Cubes[patchIndex, :, :, :] ← cube;
        Labels[patchIndex] ← GT[hh - margin, ww - margin];
        patchIndex ← patchIndex + 1;
return Cubes, Labels;
```

The 3D patches extracted from the HSI are used to generate separate training, validation, and test sets using the proposed splitting algorithm. The key algorithm, titled "Disjoint Train, Validation, and Test Split", handles dividing the HSI data into the respective portions. It takes the Ground Truth (GT) labels and ratios for the test and validation sets (teRatio and vrRatio) as inputs. The unique values in the GT labels and their frequency counts are identified, excluding zeros (background pixel labels). An iterative process is then used to create disjoint training, validation, and test sets based on these unique values and their indices. The resulting indices are utilized to extract and organize the corresponding Hyperspectral cubes and labels for each set. This ensures the subsets are separate while maintaining the integrity of spectral classes during model training and evaluation. The algorithm outputs the training, validation, and test samples along with their matching class labels. This partitioning approach contributes to the robustness and reliability of the subsequent analysis.

Algorithm 2: Disjoint Train, Validation, and Test Split

```
Input: GT, p, m
flattened ← GT.flatten();
unique ← np.unique(flattened);
nonzero_indices ← np.where(unique ≠ 0) [0];
unique ← unique[nonzero_indices];
TrInd, VaInd, TeInd ← [], [], [];
for value ← unique do
    C_ind ← np.where(flattened == value) [0];
    Tr_ind, Te_ind ← train_test_split(C_ind, p);
    Tr_ind, V_ind ← train_test_split(Tr_ind, m);
    TrInd.extend(Tr_ind);
    VaInd.extend(V_ind);
    TeInd.extend(Te_ind);
```

Let us consider that n, m, and p represent the finite numbers of labeled training, validation, and test samples, respectively, selected from patch data to form the training, validation, and test sets as shown in Eqs. (1)–(3):

$$D_{TR} = (x_i, y_i)_{i=1}^{n} \tag{1}$$

$$D_{V} = (x_i, y_i)_{i=1}^{m} \tag{2}$$

$$D_{TE} = (x_i, y_i)_{i=1}^{p} \tag{3}$$

where n and m are the total number of training and validation samples. The remaining p samples constitute the test set. It is important to note that the intersection of the training set, validation set,

and test set is an empty set (ϕ), ensuring the distinctiveness of the samples in each set as shown in Algorithm 2, Fig. 1 and Eq. (4).

$$D_{TR} \cap D_V \cap D_{TE} = \phi \tag{4}$$

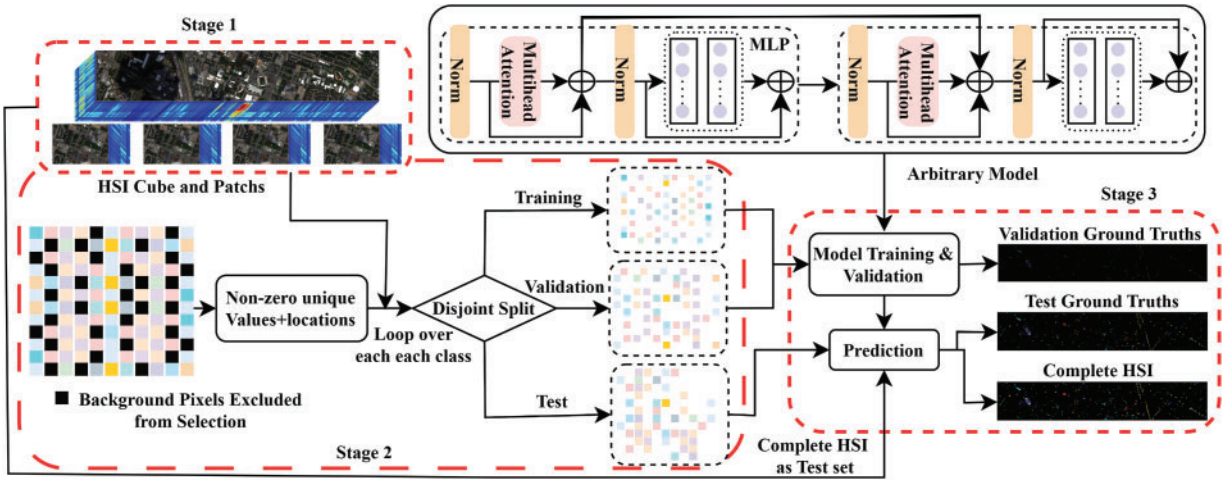

Figure 1: Initially, the HSI cube is divided into overlapping 3D patches, as detailed in Algorithm 1 and Stage 1. Each patch is centered at a spatial point and spans a $PS = S \times S$ pixel extent across all spectral bands. These patches are then utilized in Algorithm 2 to create disjoint training, validation, and test splits based on the geographical locations of the HSI samples, as outlined in Stage 2. The selected samples are fed into various models for feature learning and optimization. The processed features are subsequently passed through a fully connected layer for classification, and the softmax function is applied to generate class probability distributions. These distributions are used to create the final ground truth maps for the disjoint validation, disjoint test, and full HSI test sets, as illustrated in Stage 3. The red dotted lines delineate the stages in the proposed workflow

As shown in Fig. 1, **Stage 1: 3D Patch Extraction:** The HSI cube is initially divided into overlapping 3D patches. Each patch is centered at a spatial point and spans a $PS = S \times S$ pixel extent across all spectral bands. This process is outlined in Algorithm 1. **Stage 2: Data Splitting:** The extracted patches are used in Algorithm 2 to create disjoint training, validation, and test splits. This splitting is based on the geographical locations of the HSI samples, ensuring that each set covers distinct areas. **Stage 3: Feature Learning and Classification:** The samples from each split (training, validation, and test) are fed into various models for feature learning and optimization. The learned features are passed through a fully connected layer for classification. The softmax function is then applied to generate class probability distributions. The class probability distributions are used to generate the final ground truth maps for the disjoint validation set, disjoint test set, and the full HSI test set. Each of these stages is marked with red dotted lines in Fig. 1, facilitating a clear and systematic replication of the workflow.

The above disjoint samples (as explained in Algorithm 2) are then processed by the baseline 2D, 3D CNN, and Spatial-Spectral Transformer models [41,42]. In a 2D CNN, the input data undergoes convolution with a 2D kernel function, resulting in the computation of the dot product between the input and the kernel function. The kernel is then applied in a strided manner over the input to cover the entire spatial dimension. Subsequently, the convolved features are subjected to an activation function, which introduces non-linearity into the model, aiding in the learning of non-linear features from the

data. For 2D convolution, the activation value of the j^{th} feature map at spatial location (x, y) in the i^{th} layer, denoted by $v_{i,j}^{x,y}$, can be expressed as shown in Eq. (5).

$$v_{i,j}^{x,y} = \mathcal{F}\left(\sum_{\tau=1}^{d_{l-1}} \sum_{\rho=-\gamma}^{\gamma} \sum_{\sigma=-\delta}^{\delta} w_{i,j,\tau}^{\sigma,\rho} \times v_{i-1,\tau}^{x+\sigma,y+\rho} + b_{i,j}\right) \tag{5}$$

where $\mathcal{F}$ represents the activation function, d_{l-1} signifies the number of feature maps at the $(l-1)^{th}$ layer, while the depth of the kernel $w_{i,j}$ pertains to the j^{th} feature map at the i^{th} layer. Additionally, $b_{i,j}$ denotes the bias parameter for the j^{th} feature map at the i^{th} layer, with $2\gamma + 1$ and $2\sigma + 1$ representing the width and height of the kernel, respectively. In contrast, the 3D convolutional process initially calculates the sum of the dot product between input patches and the 3D kernel function, wherein the 3D input patches undergo convolution with the 3D kernel function [40,43]. Subsequently, these feature maps are subjected to an activation function to introduce non-linearity. The Hybrid model generates feature maps of the 3D convolutional layer by applying the 3D kernel function across B spectral bands, which are extracted post-dimensionality reduction, in the input layer. For the 3D convolutional process, the activation value at spatial location (x, y, z) in the i^{th} layer and j^{th} feature map can be expressed as in Eq. (6):

$$v_{i,j}^{x,y} = \mathcal{F}\left(\sum_{\tau=1}^{d_{l-1}} \sum_{\lambda=-v}^{v} \sum_{\rho=-\gamma}^{\gamma} \sum_{\sigma=-\delta}^{\delta} w_{i,j,\tau}^{\sigma,\rho,\lambda} \times v_{i-1,\tau}^{x+\sigma,y+\rho,z+\lambda} + b_{i,j}\right) \tag{6}$$

where all the parameters are the same as defined in Eq. (5) except $2v + 1$ which is the depth of the 3D kernel along a spectral dimension.

For the Spatial-Spectral Transformer model [44–46], consider $f_{i,j} \in \mathbb{R}^{N \times D}$ as the input tensor fed into the Transformer, where N signifies the number of patches, and D denotes the dimensionality of each patch post convolutional processing. This encoding process is fused with the input embeddings, enriching the model with spatial arrangement details. At the core of the Transformer lies its foundational architecture, the encoder, which comprises multiple layers housing multimodal attention mechanisms and a feed-forward network. The attention mechanism assumes a pivotal role in facilitating the model's ability to capture intricate relationships between diverse patches. More specifically, for a given input $v_{i,j}$, each layer within the Transformer encoder encompasses layer normalization, cross attention, and Multi-layer Perceptron (MLP).

Given a query matrix $Q \in \mathcal{R}^{(n_q \times d_k)}$, key matrix $K \in \mathcal{R}^{(n_k \times d_k)}$, value matrix $V \in \mathcal{R}^{(n_v \times d_v)}$, and mask matrix $M \in \mathcal{R}^{(n_q \times n_k)}$, where n_q, n_k, n_v, and d_v represent the number of queries, keys, values, the dimensionality of keys/queries, and the dimensionality of values, respectively. Lets define the weight matrices for each head i as $\mathcal{W}_i^Q \in \mathcal{R}^{d_{model} \times d_k}$, $\mathcal{W}_i^K \in \mathcal{R}^{d_{model} \times d_k}$, and $\mathcal{W}_i^V \in \mathcal{R}^{d_{model} \times d_v}$. Apply linear transformation as: $Q_i = Q \times \mathcal{W}_i^Q$, $K_i = K \times \mathcal{W}_i^K$, and $V_i = V \times \mathcal{W}_i^V$. Later compute the attention scores for each head and apply the Softmax to compute the weighted sum of the values as shown in Eq. (7).

$$Atten_W = Softmax\left(\frac{Q_i \times K_i^T}{\sqrt{d_k}} + M\right) \times V \tag{7}$$

Then calculate the concatenate the outputs from all heads as $\mathcal{H} = Concat(\mathcal{H}_1, \mathcal{H}_2, \vdots, \mathcal{H}_h)$ and apply a final linear transformation as $\mathcal{H}_{att}^l$. Then given an input $x \in \mathcal{R}^{d_{model}}$, layer normalization is computed as shown in Eq. (8).

$$Norm(x) = \frac{x - \mu}{\sqrt{\nu^2 + \eta}}\gamma + \lambda \tag{8}$$

where μ is the mean and ν^2 is the variance of x, η is a small constant for numerical stability and γ and λ are learnable parameters. Later the information is processed through MLP that consists of two linear transformations with a ReLU activation function as shown in (9).

$$MLP(x) = ReLu(x\mathcal{W}_1 + b_1)\mathcal{W}_2 + b_2 \tag{9}$$

where $\mathcal{W}_1 \in \mathcal{R}^{d_{model} \times d_{ff}}$, $b_1 \in \mathcal{R}^{d_{ff}}$, $\mathcal{W}_2 \in \mathcal{R}^{d_{ff} \times d_{model}}$, and $b_2 \in \mathcal{R}^{d_{model}}$. The output of the multi-head self-attention H^l_{att} is added to the input and normalized as presented in Eq. (10).

$$\hat{\mathcal{H}}^l_{att} = Norm\left(\mathcal{H}^l_{att} \oplus \mathcal{H}^{(l-1)}\right) \tag{10}$$

Eq. (10) adds the attention output $\mathcal{H}^l_{att}$ to the previous layer's output $\mathcal{H}^{(l-1)}_{att}$ and normalizes the results.

$$\mathcal{H}^l_{MLP} = MLP(\hat{\mathcal{H}}^l_{att}) \tag{11}$$

Eq. (11) computes the output of the attention mechanism in the l-th layer through an MLP. The MLP consists of two linear layers with a ReLU activation function applied between them. The output of the MLP is added to the normalized attention output and normalized again as:

$$\mathcal{H}^l = Norm\left(\mathcal{H}^l_{MLP} \oplus \hat{\mathcal{H}}^l_{att}\right) \tag{12}$$

Eq. (12) adds the output $\mathcal{H}^l_{MLP}$ to the normalized attention output and then normalizes the result once more. Afterward, $\mathcal{H}^l$ is flattened into $(\mathcal{H}^l, (b \times h \times w, 1))$, where b, h, and w denote the batch size, height, and width, respectively. Finally, a Softmax function is utilized to produce the GT maps.

3 Experimental Results and Discussion

3.1 Experimental Datasets

In order to highlight the importance and the proposed procedure of disjoint sampling in HSIC, the following datasets are used.

The University of Houston: The University of Houston dataset consists of 144 spectral bands spanning wavelengths from 380 to 1050 nm, the dataset encompasses an imaged spatial region measuring 349 × 1905 pixels at a resolution of 2.5 meters per pixel. Additionally, the dataset annotates 15 labeled classes pertaining to urban land use and land cover types. The disjoint train, validation, and test samples are presented in Table 1 and Fig. 2.

Table 1: University of Houston Dataset: Disjoint sets of training (Tr), validation (Va), and test (Te) samples were chosen, with their geographical locations (Excluding background samples) illustrated in Fig. 2, to train various SOTA models

Class	Tr	Va	Te	Class	Tr	Va	Te
Healthy grass	75	300	876	Road	75	300	877
Stressed grass	75	301	878	Highway	73	295	859
Synthetic grass	41	168	488	Railway	74	296	865
Trees	74	299	871	Parking lot 1	73	296	864

(Continued)

Table 1 (continued)

Class	Tr	Va	Te	Class	Tr	Va	Te
Soil	74	298	870	Parking lot 2	28	112	329
Water	19	78	228	Tennis court	25	103	300
Residential	76	304	888	Running track	39	159	462
Commercial	74	299	871	–	–	–	–

Figure 2: The University of Houston Dataset: Geographical locations of the disjoint train, validation, and test samples presented in Table 1

Indian Pines: The Indian Pines dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over an agricultural site in Northwestern Indiana. It consists of 145 × 145 pixels with spectral information across 224 narrow bands ranging from 0.4 to 2.5 micrometers. The major land cover classes in the dataset included agricultural land, forest, highways, rail lines, low-density housing, and built structures separated by smaller roads. Crops such as corn and soybeans covered less than 5% of typical growing areas as the June image showed early stages of development. Ground truths designate 16 non-mutually exclusive classes. The number of bands was reduced to 200 by removing wavelengths associated with water absorption. The disjoint train, validation, and test samples are presented in Table 2 and Fig. 3.

Table 2: Indian Pines Dataset: Disjoint sets of training (Tr), validation (Va), and test (Te) samples were chosen, with their geographical locations (Excluding background samples) illustrated in Fig. 3, to train various SOTA models

Class	Tr	Va	Te	Class	Tr	Va	Te
Alfalfa	6	7	33	Oats	3	3	14
Corn-notill	214	214	1000	Soybean-notill	145	146	681
Corn-mintill	124	125	581	Soybean-mintill	368	368	1719
Corn	35	36	166	Soybean-clean	88	89	416
Grass-pasture	72	72	339	Wheat	30	31	144
Grass-trees	109	110	511	Woods	189	190	886
Grass-mowed	4	4	20	Buildings	57	58	271
Hay-windrowed	71	72	335	Stone-Steel	13	14	66

Pavia University: The Pavia University dataset was captured using the Reflective Optics System Imaging Spectrometer (ROSIS), this dataset consists of an image with 610 × 340 pixels and 115 spectral bands. It has 9 classes of urban materials-including asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, brick, and shadows-comprising 42,776 labeled samples in total. The disjoint train, validation, and test samples are presented in Table 3 and Fig. 4.

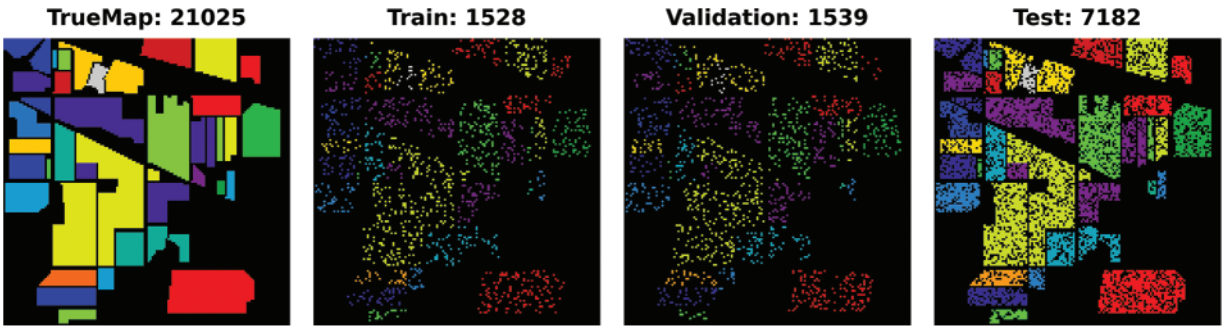

Figure 3: Indian Pines: Geographical locations of the disjoint train, validation, and test samples presented in Table 2

Table 3: Pavia University Dataset: Disjoint sets of training (Tr), validation (Va), and test (Te) samples were chosen, with their geographical locations (Excluding background samples) illustrated in Fig. 4, to train various SOTA models

Class	Tr	Va	Te	Class	Tr	Va	Te
Asphalt	994	995	4642	Soil	754	754	3521
Meadows	2797	2797	13055	Bitumen	199	200	931
Gravel	314	315	1470	Bricks	552	552	2578
Trees	459	460	2145	Shadows	142	142	663
Painted	201	202	942	–	–	–	–

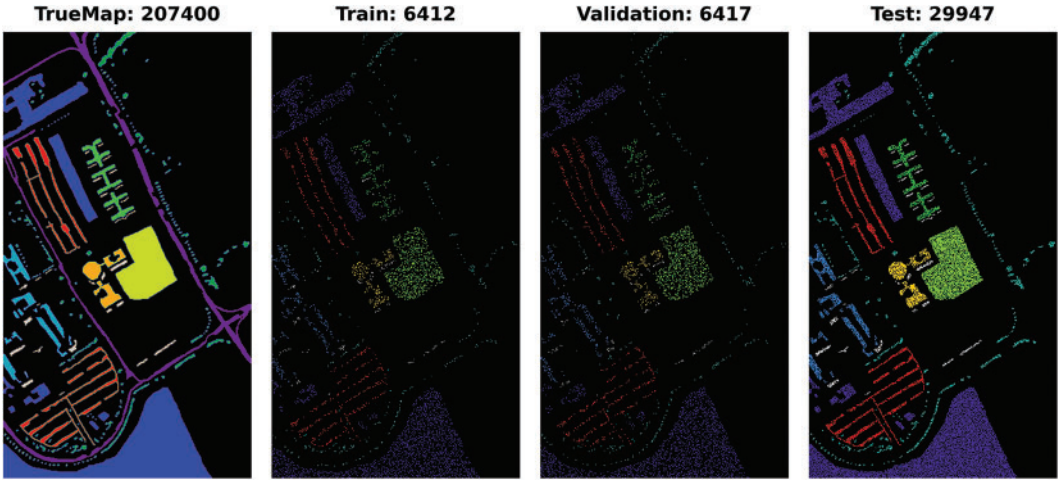

Figure 4: Pavia University Dataset: Geographical locations of the disjoint train, validation, and test samples presented in Table 3

Salinas: The Salinas dataset is collected using the 224-band AVIRIS sensor over Salinas Valley, California, this dataset is characterized by high spatial resolution at 3.7 m per pixel. The study area encompasses 512 lines by 217 samples after removing 20 bands obscured by water absorption. Land cover types within the dataset include vegetables, bare soils, and vineyard fields. The Salinas ground

truth annotates 16 classes. The disjoint train, validation, and test samples are presented in Table 4 and Fig. 5.

Table 4: Salinas Dataset: Disjoint sets of training (Tr), validation (Va), and test (Te) samples were chosen, with their geographical locations (Excluding background samples) illustrated in Fig. 5, to train various SOTA models

Class	Tr	Va	Te	Class	Tr	Va	Te
Weeds 1	301	301	1407	Soil vinyard develop	930	930	4343
Weeds 2	558	559	2609	Corn weeds	491	492	2295
Fallow	296	296	1384	Lettuce 4 wk	160	160	748
Fallow rough plow	209	209	976	Lettuce 5 wk	289	289	1349
Fallow smooth	401	402	1875	Lettuce 6 wk	137	137	642
Stubble	593	594	2772	Lettuce 7 wk	160	161	749
Celery	536	537	2506	Vinyard untrained	1090	1090	5088
Grapes untrained	1690	1691	7890	Vinyard trellis	271	271	1265

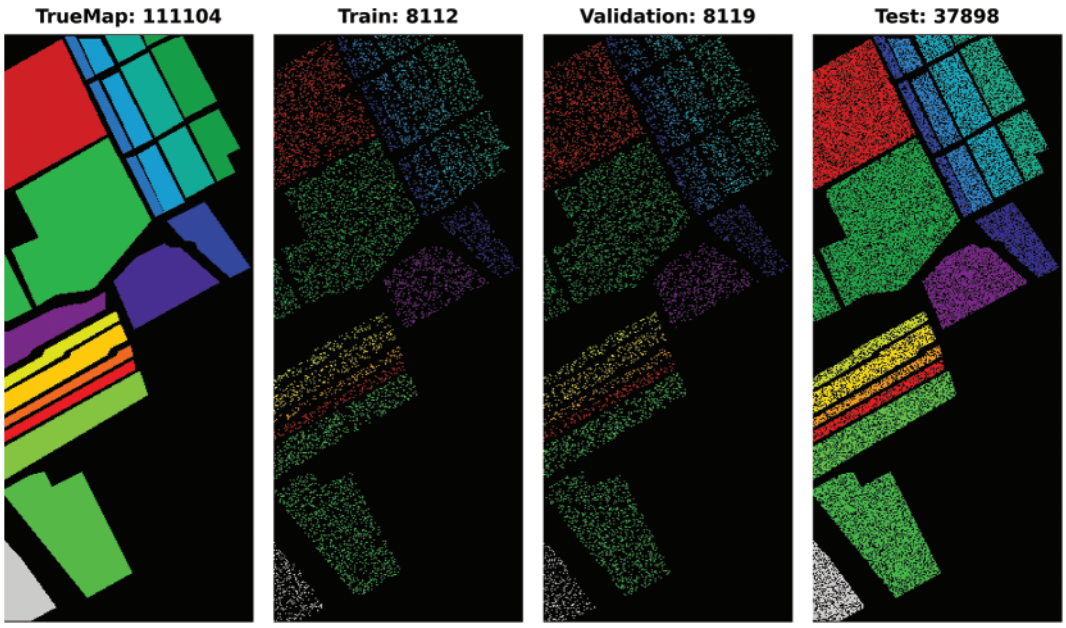

Figure 5: Salinas Dataset: Geographical locations of the disjoint train, validation, and test samples presented in Table 4

Botswana: The NASA EO-1 satellite acquired Hyperspectral imagery of the Okavango Delta region in Botswana from 2001–2004 using the Hyperion sensor to collect 30 m resolution data across 242 bands from 400–2500 nm over a 7.7 km strip. The data analyzed from 31 May, 2001, consisted of observations of 14 land cover classes representing seasonal swamps, occasional swamps, and drier woodlands in the distal delta region after preprocessing removed uncalibrated and noisy bands covering water absorption and retaining 145 bands. The disjoint train, validation, and test samples are presented in Table 5 and Fig. 6.

Table 5: Botswana Dataset: Disjoint sets of training (Tr), validation (Va), and test (Te) samples were chosen, with their geographical locations (Excluding background samples) illustrated in Fig. 6, to train various SOTA models

Class	Tr	Va	Te	Class	Tr	Va	Te
Water	40	41	189	Island interior	30	30	143
Hippo grass	15	15	71	Woodlands	47	47	220
Floodplain grasses 1	37	38	176	Acacia Shrublands	37	37	174
Floodplain grasses 2	32	32	151	Acacia Grasslands	45	46	214
Reeds 1	40	40	189	Short Mopane	27	27	127
Riparian	40	40	189	Mixed Mopane	40	40	188
Firescar 2	38	39	182	Exposed soils	14	14	67

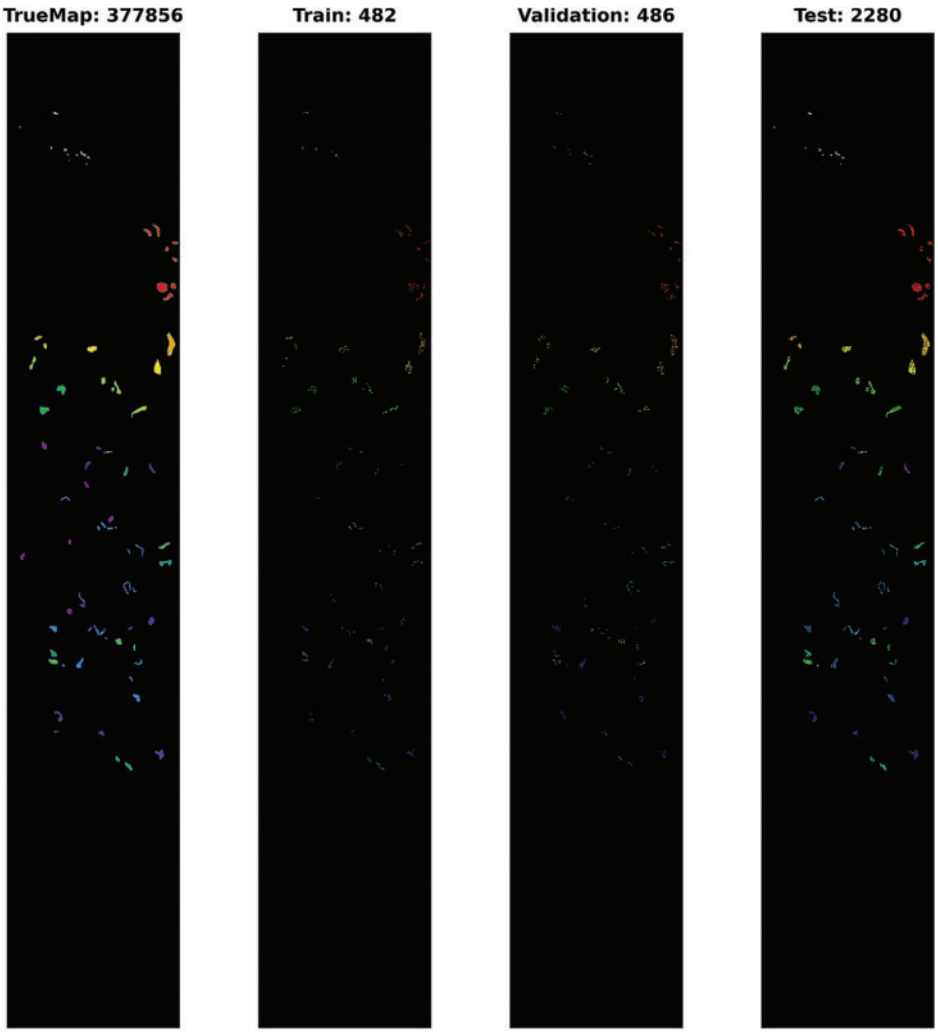

Figure 6: Botswana Dataset: Geographical locations of the disjoint train, validation, and test samples presented in Table 5

3.2 Experimental Settings

This section presents comprehensive experimental settings for various deep learning models, including 3D CNN [20], Hybrid Inception Net [47], 3D Inception Net [48], 2D Inception Net [49], 2D CNN [50], Hybrid CNN [51], Attention Graph CNN [22]. Spatial-spectral Transformer [24]. Prior to training, 3D overlapped patches are extracted using an 8×8 window size, as outlined in Algorithm 1. All models in this study are trained using the Adam optimizer with a learning rate of 0.0001, a decay rate of 1e-06, and a batch size of 56 for 50 epochs. The loss and accuracy trend is presented in Fig. 7 for all the competing methods.

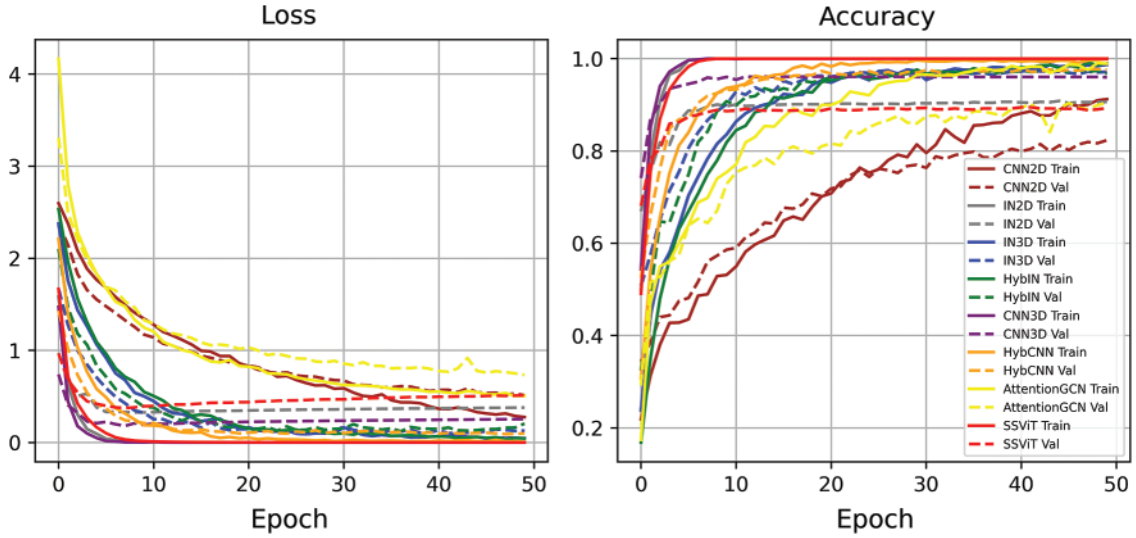

Figure 7: Loss and Accuracy trends for all the competing methods for Indian Pines dataset

All evaluations were conducted on a Google Colab, using a Jupyter Notebook. Colab works on online resources and requires a fast and stable internet connection. Colab works on a Python 3 notebook with a graphic processing unit (GPU) for data analysis, offering 25 GB of random access memory (RAM) and 358 GB of storage.

All the competing methods are tested with a patch size of 8×8, using the 15 most informative bands selected through principal component analysis. The dataset is split into 70% test samples, with the remaining 30% equally divided into training and validation sets (15% each). For effective learning, all models are trained using the Adam optimizer with a learning rate of 0.0001 and a decay of $1e^{-06}$ over 50 epochs, with a batch size of 56.

The 2D CNN model is trained using four convolutional layers with kernel sizes of $(3 \times 3 \times 8, 16, 32, 64)$ and the same padding with $(8, 16, 32, 64)$ number of filters, respectively. Following the convolutional layers, two dense layers are utilized with a dropout rate of 0.4%. Finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset. The 3D CNN model is trained using four convolutional layers with kernel sizes of $(3 \times 3 \times 7, 5, 3, 3)$ with $(8, 16, 32, 64)$ number of filters, respectively. Following the convolutional layers, two dense layers are utilized and finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset. The Hybrid CNN model is trained using three 3D convolutional layers with kernel sizes of $(3 \times 3 \times 7, 5, 3)$, followed by a reshaped layer to transform the features into 2D to learn spatial features using (3×3) kernel with 64 filters. Following the

convolutional layers, two dense layers are utilized with a dropout rate of 0.4%. Finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset.

The 2D Inception Net architecture consists of three blocks with the following configurations. In the first block, three 2D convolutional layers are used. The first layer employs a (1×1) kernel with 30 filters, the second layer uses a (3×3) kernel with 20 filters, and the third layer utilizes a (1×1) kernel with 10 filters. In the second block, three 2D convolutional layers are utilized. The first layer has a (1×1) kernel with 40 filters, the second layer employs a (5×5) kernel with 20 filters, and the third layer uses a (1×1) kernel with 10 filters. The third block begins with a 2D max pooling operation using a (3×3) kernel and the same padding. This is followed by two 2D convolutional layers with (1×1) kernels and the same padding. The filters for these layers are set to 20 and 10, respectively. Afterward, the outputs from all three blocks are concatenated, and a convolutional layer with a (1×1) kernel and 128 filters is applied. Following the convolutional layer, two dense layers are deployed. Finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset.

The 3D Inception Net architecture consists of three blocks with the following configurations. In the first block, three 3D convolutional layers are used. The first layer employs a $(5 \times 5 \times 7)$ kernel with 30 filters, the second layer uses a $(3 \times 3 \times 5)$ kernel with 20 filters, and the third layer utilizes a $(3 \times 3 \times 3)$ kernel with 10 filters and the same padding in all three layers. In the second block, three 3D convolutional layers are utilized. The first layer has a $(5 \times 5 \times 7)$ kernel with 40 filters, the second layer employs a $(3 \times 3 \times 5)$ kernel with 20 filters, and the third layer uses a $(3 \times 3 \times 3)$ kernel with 10 filters and the same padding in all three layers. The third block begins with three 3D convolutional layers with $(5 \times 5 \times 7)$ kernel with 60 filters, the second layer uses a $(3 \times 3 \times 5)$ kernel with 30 filters, and the third layer utilizes a $(3 \times 3 \times 3)$ kernel with 10 filters and the same padding in all three layers. Afterward, the outputs from all three blocks are concatenated, and a convolutional layer with a $(1 \times 1 \times 1)$ kernel and 128 filters is applied. Following the convolutional layer, two dense layers are deployed with a 0.4% dropout rate. Finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset.

The hybrid Inception Net architecture consists of three blocks with the following configurations. In the first block, three 3D convolutional layers are used. The first layer has a $(5 \times 5 \times 7)$ kernel with 30 filters, the second layer uses a $(3 \times 3 \times 5)$ kernel with 20 filters, and the third layer employs a $(3 \times 3 \times 3)$ kernel with 10 filters. The same padding is applied in all three layers. Following the convolutional layer, a reshaped layer is used to convert the features from 3D to 2D. Next, a 2D max-pooling layer with a (3×3) filter is applied, followed by three 2D convolutional layers. Each of these layers uses a (1×1) kernel with 16, 32, and 64 filters, respectively. The same padding is used for all three layers. The same configuration is repeated for the second and third blocks, with the numbers of filters set to 40, 20, and 10 for the 3D convolutional layers, and 16, 32, and 64 for the 2D convolutional layers in the second block, and 60, 30, and 10 for the 3D convolutional layers, and 16, 32, and 64 for the 2D convolutional layers in the third block, respectively. Afterward, the outputs from all three blocks are concatenated, and a convolutional layer with a $(1 \times 1 \times 1)$ kernel and 128 filters is applied. Following the convolutional layer, two dense layers are deployed with a 0.4% dropout rate. Finally, a classification layer with Softmax is added with the number of output units corresponding to the number of classes in the dataset.

The Attention Graph CNN [22] and Spatial-Spectral Transformer [24] models are trained according to the settings specified in their respective papers. The Transformer model, in particular,

is used without the wavelet transformation and consists of 4 layers with 8 heads to compute the final maps. A dropout rate of 0.1 is applied to the classification layers. For more detailed information, please refer to the original papers.

3.3 Qualitative and Quantitative Results and Discussion

This section provides a detailed exploration of experimental results in comparison to the state-of-the-art (SOTA) works published in recent years. While many recent research endeavors present extensive experimental outcomes to highlight the strengths and weaknesses of their approaches, it is noteworthy that the experimental results in the literature may follow diverse protocols. For instance, the selection of training, validation, and test samples might be randomly done, and the percentage distribution may be identical. However, there could be variations in the geographical locations of each model, as these models may have undergone training, validation, and testing at different times. Comparative models may have been executed in multiple instances, either sequentially or in parallel, introducing a new set of training, validation, and test samples with the same number or percentage. Consequently, to ensure a fair comparison between the works proposed in the literature and the current study, it is imperative to employ identical experimental settings and execute them with the same set of training, validation, and test samples. This approach ensures a consistent and unbiased evaluation of the proposed methodologies against existing benchmarks.

A prevalent concern in the majority of recent literature is the presence of overlapping training and test samples. When training and validation samples are randomly selected, with or without considering the point mentioned earlier, the data split often includes overlapping samples. This situation introduces bias to the model, as overlapping implies the model has already encountered the training and validation samples, leading to inflated accuracy metrics. To prevent this issue, this study ensures that, despite the random selection of samples, the intersection between training, test, and validation samples remains consistently empty for all competing methods. This measure aims to maintain the integrity of the model evaluation process and uphold the reliability of accuracy assessments.

To ensure a robust and fair evaluation, the datasets are split into disjoint training, validation, and test sets. Following the proposed method, we begin by dividing the HSI dataset into disjoint training, validation, and test sets. Each model is then trained on the training set and tuned on the validation set to optimize performance. Subsequently, the models are evaluated on the disjoint test set and the complete HSI dataset to assess their generalization capabilities. The experimental results demonstrate the effectiveness of the proposed method in improving the classification accuracy of HSIC as shown in Tables 6–10 and Figs. 8–12. Among the deep learning models considered, 3D CNN [20] and Hybrid Inception Net [47] achieve the highest classification accuracy, indicating their suitability for HSIC. Additionally, the results highlight the importance of using a large and diverse training dataset to achieve optimal performance.

Table 6: Indian Pines Dataset: Per class comparative results of various SOTA models are showcased on disjoint validation and test sets. Additionally, results on the entire HSI dataset serving as the test set are also presented. The comparative methods include 3D CNN [20], Hybrid Inception Net (Hybrid IN) [47], 3D Inception Net (3D IN) [48], 2D Inception Net (2D IN) [49], 2D CNN [50], Hybrid CNN [51], Attention Graph CNN (Attention GCN) [22], and Spatial-Spectral Transformer [24]. The geographical maps for each model for disjoint validation, test, and complete test are presented in Fig. 8

Class	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI
Alfalfa	57.14	69.69	99.87	85.71	75.75	99.92	100.00	96.97	99.99	100.00	93.94	99.98	100.00	90.91	99.97	100.00	93.94	99.98	71.43	60.61	99.86	100.00	90.91	99.97
Corn-notill	74.76	75.4	78.36	92.52	94.10	94.75	92.53	92.80	93.77	84.58	84.20	86.62	97.66	97.30	97.76	95.33	96.00	96.50	76.64	71.30	75.70	85.51	84.70	87.11
Corn-mintill	76.8	81.06	82.77	96.00	97.76	97.83	97.60	99.83	99.52	84.80	86.40	88.19	96.80	98.45	98.43	97.60	99.66	99.40	60.80	61.96	66.14	82.4	84.34	86.39
Corn	58.33	45.78	54.85	94.44	89.16	91.56	94.44	95.18	95.78	88.89	79.52	83.97	100.00	98.80	99.16	88.89	88.55	90.30	63.89	53.61	62.03	58.33	48.80	57.81
Grass-pasture	94.44	94.10	95.03	97.22	96.76	97.31	98.61	97.94	98.34	95.83	95.28	96.07	98.61	97.05	97.72	95.83	96.76	97.10	90.28	89.97	91.30	94.44	96.17	96.48
Grass-trees	100.	99.41	99.58	99.09	99.41	99.45	100.00	99.80	99.86	98.18	99.41	99.32	100.00	99.80	99.86	100.00	100.00	100.00	98.18	98.63	98.77	99.09	99.22	99.32
Grass-mowed	0.	0.	7.14	100.00	100.00	100.00	100.00	100.00	100.00	75.00	70.00	75.00	75.00	70.00	75.00	25.00	20.00	25.00	25.00	5.00	14.29	75.00	75.00	78.57
Hay-windrowed	98.611	98.50	98.74	100	99.70	99.79	100.00	100.00	100.00	100.00	98.21	98.74	100.00	100.00	100.00	100.00	100.00	100.00	98.61	95.52	96.65	98.61	99.40	99.37
Oats	0.	0.	5.0	66.66	71.43	75.00	66.67	85.71	85.00	33.33	71.43	70.00	0.00	28.57	35.00	0.00	28.57	30.00	0.00	21.43	30.00	0.00	78.57	70.00
Soybean-notill	75.34	74.44	77.77	92.47	91.34	92.80	91.78	93.10	93.93	86.30	81.64	85.08	91.78	94.42	94.86	88.36	91.48	92.18	81.51	79.00	82.41	82.19	82.97	85.39
Soybean-mintill	85.05	88.24	89.16	97.83	97.27	97.76	99.73	99.71	99.76	89.40	92.15	92.91	98.10	98.49	98.66	98.91	98.72	98.94	86.14	87.73	89.33	91.58	92.03	93.16
Soybean-clean	58.42	63.22	67.11	92.13	96.88	96.63	93.26	96.88	96.80	91.01	90.87	92.24	89.89	97.12	96.46	97.75	96.88	97.47	50.56	49.76	56.16	79.78	80.53	83.31
Wheat	83.87	95.13	94.14	100.00	100.00	100.00	100.00	100.00	100.00	96.77	99.31	99.02	100.00	97.92	98.54	100.00	98.61	99.02	96.77	92.36	94.15	93.55	99.31	98.54
Woods	97.36	97.40	97.78	96.32	97.29	97.55	98.95	99.32	99.37	98.42	99.10	99.13	98.95	98.53	98.81	98.95	99.21	99.30	97.37	97.18	97.63	97.89	96.73	97.39
Buildings	75.86	81.91	83.67	100	97.79	98.45	100.00	98.89	99.22	86.21	92.25	92.49	100.00	98.15	98.70	100.00	97.42	98.19	77.58	75.65	79.53	93.10	94.83	95.34
Stone-Steel	50.	83.33	79.56	92,86	96.97	96.77	100.00	100.00	100.00	92.86	98.48	97.85	100.00	100.00	100.00	100.00	100.00	100.00	78.57	83.33	84.95	92.86	100.00	98.92
Kappa	79.85	81.91	90.23	95.41	95.75	97.78	96.74	97.36	98.58	89.35	89.88	94.74	96.74	97.38	98.59	96.22	96.68	98.22	79.67	78.29	88.76	87.86	88.29	93.93
OA	82.33	84.17	93.10	95.97	96.27	98.43	97.14	97.69	99.00	90.64	91.13	96.29	97.14	97.70	99.00	96.69	97.09	98.74	82.20	81.06	92.08	89.34	89.74	95.71
AA	67.88	71.73	75.66	93.95	93.85	95.97	95.85	97.26	97.58	87.60	89.51	91.04	90.42	91.59	93.06	87.86	87.86	88.96	72.08	70.19	76.18	87.72	87.72	89.19
Time (s)	1.54	1.68	8.74	0.45	1.39	9.80	0.53	0.81	8.64	0.85	1.39	10.96	1.48	3.14	14.84	0.89	2.68	12.75	1.18	1.38	10.27	1.28	2.24	14.72

Table 7: Pavia University Dataset: Per class comparative results of various SOTA models are showcased on disjoint validation and test sets. Additionally, results on the entire HSI dataset serving as the test set are also presented. The comparative methods include 3D CNN [20], Hybrid Inception Net (Hybrid IN) [47], 3D Inception Net (3D IN) [48], 2D Inception Net (2D IN) [49], 2D CNN [50], Hybrid CNN [51], Attention Graph CNN (Attention GCN) [22], and Spatial-Spectral Transformer [24]. The geographical maps for each model for disjoint validation, test, and complete test are presented in Fig. 9

Class	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI
Asphalt	99.70	99.87	99.99	99.80	99.78	99.99	100.00	99.83	99.99	98.59	98.97	99.96	99.40	99.05	99.97	100.00	100.00	100.00	98.29	98.32	99.94	98.29	98.15	99.94
Meadows	99.96	99.95	99.96	99.96	100.00	99.99	99.96	99.99	99.99	99.89	99.96	99.96	99.96	100.00	99.99	99.89	99.91	99.92	99.82	99.90	99.89	100.00	98.91	99.94
Gravel	94.29	94.35	95.19	98.41	97.55	98.04	98.41	98.50	98.71	93.65	93.67	94.62	97.46	97.76	98.05	97.14	97.69	97.95	87.30	85.17	86.04	89.21	88.76	90.52
Trees	98.91	99.07	99.18	99.13	99.39	99.25	99.13	99.49	99.51	96.96	97.86	98.04	98.26	98.97	99.02	99.78	99.86	99.87	95.43	96.41	96.57	97.39	98.60	98.63
Painted	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.89	99.93	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Soil	99.87	99.20	99.42	100.00	100.00	100.00	100.00	100.00	100.00	99.34	98.64	98.95	100.00	99.94	99.96	100.00	100.00	100.00	100.00	99.43	99.60	99.73	99.03	99.28
Bitumen	98.00	98.82	98.87	99.50	99.36	99.47	99.50	99.79	99.77	96.50	97.96	98.05	99.50	99.79	99.77	100.00	99.68	99.77	94.00	93.34	94.29	97.00	96.24	96.92
Bricks	96.56	97.05	97.42	98.73	99.19	99.24	99.46	99.34	99.46	96.56	96.47	97.01	99.28	99.50	99.54	99.28	98.80	99.02	96.56	97.40	97.64	94.38	93.91	94.89
Shadows	100.00	98.79	99.16	99.30	99.40	99.47	99.30	99.85	99.79	100.00	98.49	98.94	100.00	99.85	99.94	100.00	99.70	99.79	99.30	99.25	99.37	99.30	99.55	99.58
Kappa	98.95	98.95	99.55	99.57	99.60	99.83	99.69	99.73	99.88	98.31	98.40	99.31	99.42	99.48	99.77	99.65	99.62	99.84	97.62	97.61	98.90	97.81	97.69	99.02
OA	99.21	99.21	99.86	99.67	99.70	99.95	99.77	99.79	99.96	98.72	98.79	99.79	99.56	99.61	99.93	99.74	99.72	99.95	98.21	98.20	99.66	98.39	98.26	99.70
AA	98.59	98.57	98.80	99.43	99.41	99.52	99.53	99.64	99.69	97.94	97.99	98.38	99.32	99.43	99.58	99.57	99.51	99.59	96.74	96.58	97.04	97.26	97.13	97.74
Time (s)	1.05	2.91	85.48	1.42	3.52	90.00	0.80	5.47	83.94	1.47	3.81	85.17	2.98	11.53	139.04	2.90	7.13	105.40	1.80	5.01	100.10	2.49	10.61	144.50

Table 8: University of Houston Dataset: Per class comparative results of various SOTA models are showcased on disjoint validation and test sets. Additionally, results on the entire HSI dataset serving as the test set are also presented. The comparative methods include 3D CNN [20], Hybrid Inception Net (Hybrid IN) [47], 3D Inception Net (3D IN) [48], 2D Inception Net (2D IN) [49], 2D CNN [50], Hybrid CNN [51], Attention Graph CNN (Attention GCN) [22], and Spatial-Spectral Transformer [24]. The geographical maps for each model for disjoint validation, test, and complete test are presented in Fig. 10

Class	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI
Healthy grass	97.33	96.69	99.99	97.00	97.95	99.99	99.33	99.09	99.99	98.67	98.63	99.99	99.33	98.86	99.99	100.00	99.54	99.99	94.33	94.18	99.99	98.00	98.17	99.99
Stressed grass	99.67	99.20	99.36	99.67	98.86	99.12	99.67	98.97	99.20	99.67	98.97	99.20	99.67	99.32	99.44	99.34	98.41	98.72	98.67	98.41	98.56	99.34	98.75	98.96
Synthetic grass	98.81	99.39	99.28	95.83	96.31	96.41	97.02	97.95	97.85	95.24	92.83	93.83	98.21	98.16	98.28	98.81	98.57	98.71	97.62	97.54	97.70	97.02	96.11	96.56
Trees	100.00	99.77	99.84	98.33	98.85	98.79	98.99	99.54	99.44	99.33	98.74	98.95	99.67	99.89	99.84	99.33	99.66	99.60	90.30	89.44	90.03	97.32	97.93	97.91
Soil	100.00	100.00	100.00	99.66	100.00	99.92	100.00	100.00	100.00	100.00	99.89	99.92	100.00	100.00	100.00	100.00	100.00	100.00	99.66	99.77	99.76	100.00	99.89	99.92
Water	78.21	82.46	82.46	92.31	95.18	94.77	97.44	99.12	98.77	79.49	85.09	84.62	92.31	91.23	92.00	97.44	97.81	97.85	75.64	75.44	76.92	94.87	95.61	95.69
Residential	95.07	94.14	94.72	96.05	96.28	96.45	98.36	98.54	98.58	95.39	94.26	94.87	98.36	97.75	98.03	98.68	98.09	98.34	84.54	86.26	86.67	92.76	94.37	94.32
Commercial	85.28	85.19	85.85	97.66	97.70	97.83	97.66	97.82	97.91	99.33	95.64	96.78	94.98	97.82	97.27	96.32	97.36	97.27	81.94	84.27	84.65	93.98	96.90	96.38
Road	91.67	89.28	90.50	94.33	92.82	93.61	97.33	95.78	96.41	94.33	91.33	92.57	94.00	92.93	93.61	94.33	92.93	93.69	83.00	80.62	82.35	91.66	91.33	91.93
Highway	93.90	93.83	94.21	98.98	98.72	98.86	95.93	95.81	95.93	97.63	96.97	97.31	99.32	99.30	99.35	99.32	99.30	99.35	91.53	93.95	93.72	97.63	97.21	97.47
Railway	95.27	96.30	96.28	98.31	98.15	98.30	100.00	99.77	99.84	99.66	99.65	99.68	99.32	99.31	99.35	99.66	98.03	98.54	93.24	91.56	92.47	98.99	98.38	98.62
Parking lot 1	94.93	95.49	95.62	98.99	99.88	99.68	98.65	99.54	99.35	97.64	98.03	98.05	98.99	99.77	99.59	98.99	99..88	99.68	95.27	93.52	94.32	98.65	99.19	99.10
Parking lot 2	63.39	62.92	65.25	79.46	81.46	82.09	93.75	96.96	96.38	85.71	88.15	88.27	94.64	94.83	95.10	96.43	96.05	96.38	50.00	52.88	55.01	59.82	68.99	68.65
Tennis court	97.09	96.67	96.96	99.03	98.00	98.36	99.03	98.67	98.83	90.29	87.00	88.55	99.03	100.00	99.77	96.12	100.00	99.07	90.29	90.00	90.65	97.09	95.67	96.26
Running track	100.00	99.78	99.85	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.92	99.24	100.00	100.00	100.00	100.00	100.00	100.00	95.27	92.42	94.09	100.00	100.00	100.00
Kappa	93.91	93.7	96.99	97.00	97.14	98.61	98.29	98.38	99.20	96.85	95.93	98.16	98.05	98.17	99.11	98.38	98.23	99.17	89.35	89.13	94.79	95.44	96.01	98.02
OA	94.37	94.18	99.88	97.23	97.36	99.94	98.42	98.50	99.97	97.09	96.24	99.92	98.20	98.31	99.96	98.50	98.37	99.97	90.16	89.96	99.79	95.79	96.31	99.92
AA	92.71	92.74	93.34	96.37	96.68	96.95	98.21	98.50	98.56	95.49	94.94	95.46	97.86	97.94	98.11	98.32	98.37	99.48	88.23	88.02	89.13	94.48	95.23	95.45
Time (s)	1.63	2.54	320.98	0.76	1.58	323.42	0.47	1.12	309.30	0.63	1.38	362.58	1.78	4.55	542.21	1.24	2.85	449.60	2.07	2.73	342.90	2.04	3.76	533.75

Table 9: Botswana Dataset: Per class comparative results of various SOTA models are showcased on disjoint validation and test sets. Additionally, results on the entire HSI dataset serving as the test set are also presented. The comparative methods include 3D CNN [20], Hybrid Inception Net (Hybrid IN) [47], 3D Inception Net (3D IN) [48], 2D Inception Net (2D IN) [49], 2D CNN [52], Hybrid CNN [51], Attention Graph CNN (Attention GCN) [22], and Spatial-Spectral Transformer [24]. The geographical maps for each model for disjoint validation, test, and complete test are presented in Fig. 11

Class	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI
Water	100.00	100.00	100.00	100.00	98.94	99.99	97.56	100.00	99.99	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	97.56	99.47	99.99	100.00	100.00	100.00
Hippo grass	53.33	53.52	60.40	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	80.00	50.70	62.38	100.00	100.00	100.00
Floodplain grasses 1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	89.47	87.50	89.64	100.00	100.00	100.00
Floodplain grasses 2	96.88	99.34	99.07	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	93.75	99.34	98.60	100.00	100.00	100.00
Reeds 1	87.5	92.06	92.19	100.00	99.47	99.63	95.00	97.35	97.40	87.5	90.48	91.45	90.00	94.18	94.42	97.50	100.00	99.63	92.50	95.24	95.54	87.50	93.12	93.31
Riparian	82.50	82.54	85.13	95.00	94.71	95.54	87.50	92.06	92.57	90.00	95.24	95.17	95.00	95.24	95.91	90.00	94.71	94.80	87.50	86.77	88.85	92.50	94.71	95.17
Firescar 2	100.00	98.35	98.84	100.00	98.90	99.23	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Island interior	96.67	98.60	98.52	100.00	95.10	96.55	100.00	96.50	97.54	100.00	95.80	97.04	100.00	100.00	100.00	100.00	100.00	100.00	86.67	93.01	93.10	100.00	100.00	100.00
Woodlands	82.98	85.00	86.94	97.87	96.36	97.13	100.00	100.00	100.00	100.00	95.45	96.82	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.64	99.04	100.00	97.73	98.41
Acacia Shrublands	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	97.30	88.51	91.53	100.00	100.00	100.00
Acacia Grasslands	93.48	97.66	97.38	100.00	100.00	100.00	100.00	100.00	100.00	93.48	95.79	96.07	97.83	99.07	99.02	100.00	100.00	100.00	93.48	85.51	88.85	97.83	99.07	99.02
Short mopane	88.89	87.40	89.50	100.00	100.00	100.00	100.00	100.00	100.00	92.59	97.64	97.24	100.00	100.00	100.00	100.00	100.00	100.00	85.19	90.55	99.16	100.00	99.21	99.44
Mixed mopane	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.47	99.63	100.00	100.00	100.00	100.00	100.00	100.00	95.00	94.15	95.15	100.00	99.47	99.62
Exposed soils	92.86	89.55	91.58	92.86	97.01	96.84	100.00	100.00	100.00	92.85	92.54	93.68	92.86	95.52	95.79	92.86	97.01	96.84	71.43	70.15	74.74	92.86	97.01	96.84
Kappa	91.97	92.96	96.88	99.11	98.48	99.39	98.22	98.81	99.44	96.66	97.10	98.72	98.22	98.81	99.44	98.66	99.43	99.7	91.97	90.48	96.02	97.77	98.38	99.26
OA	92.59	93.51	99.95	99.18	98.60	99.99	98.35	98.90	99.99	96.91	97.32	99.98	98.35	98.90	99.99	98.77	99.47	99.99	92.59	91.23	99.94	97.94	98.51	99.98
AA	91.08	91.72	92.83	98.98	98.61	98.92	98.58	98.99	99.11	96.89	97.32	97.65	98.26	98.86	98.94	98.60	99.41	99.38	90.70	88.54	90.61	97.91	98.59	98.70
Time (s)	1.67	0.71	91.08	0.50	0.73	173.25	0.20	0.33	169.01	0.81	0.52	153.69	0.78	1.34	269.21	0.61	1.34	231.29	0.71	0.70	185.74	1.01	1.35	276.85

Table 10: Salinas Dataset: Per class comparative results of various SOTA models are showcased on disjoint validation and test sets. Additionally, results on the entire HSI dataset serving as the test set are also presented. The comparative methods include 3D CNN [20], Hybrid Inception Net (Hybrid IN) [47], 3D Inception Net (3D IN) [48], 2D Inception Net (2D IN) [49], 2D CNN [50], Hybrid CNN [51], Attention Graph CNN (Attention GCN) [22], and Spatial-Spectral Transformer [24]. The geographical maps for each model for disjoint validation, test, and complete test are presented in Fig. 12

Class	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI	Va	Te	HSI
Weeds 1	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Weeds 2	100	100	100	99.64	100	99.95	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Fallow	100	100	100	99.32	99.64	99.65	100	100	100	99.66	99.78	99.80	100	100	100	100	100	100	98.65	99.71	99.60	99.32	99.71	99.70
Fallow rough plow	100	99.90	99.93	100	100	100	100	100	100	100	99.80	99.86	100	100	100	99.52	100	99.93	98.56	99.28	99.21	99.52	99.90	99.86
Fallow smooth	97.26	98.19	98.32	99.25	99.20	99.33	98.76	99.20	99.07	97.26	97.55	97.87	99.75	100	99.96	99.75	99.95	99.93	98.01	97.87	98.21	98.76	98.51	98.77
Stubble	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Celery	100	100	100	100	100	100	100	100	100	99.81	99.80	99.83	100	100	100	100	100	100	100	100	100	100	100	100
Grapes untrained	99.53	99.21	99.38	99.59	99.54	99.62	99.88	99.68	99.76	98.11	98.18	98.45	99.88	99.85	99.88	99.82	99.67	99.74	98.17	98.16	98.35	97.99	97.68	98.09
Soil vinyard develop	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	99.98	99.98	100	100	100
Corn weeds	100	100	100	100	100	100	100	100	100	100	100	100	100	99.96	99.97	100	100	100	100	99.91	99.94	100	100	100
Lettuce 4wk	99.38	99.73	99.72	100	100	100	100	100	100	100	99.86	99.91	100	100	100	100	100	100	98.75	98.80	98.88	100	100	100
Lettuce 5 wk	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Lettuce 6 wk	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	99.28	99.84	99.78	100	100	100
Lettuce 7 wk	99.38	100	99.91	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Vinyard untrained	97.25	96.91	97.37	99.82	99.84	99.86	99.91	99.86	99.89	95.41	95.48	96.15	99.72	99.88	99.88	100	99.72	99.81	95.60	94.99	95.46	97.80	96.76	97.40
Vinyard trellis	99.63	99.53	99.61	100	99.84	99.89	99.26	99.76	99.72	99.26	99.68	99.67	100	100	100	100	100	100	99.89	99.13	99.22	100	100	100
Kappa	99.29	99.23	99.59	99.78	99.81	99.89	99.86	99.85	99.92	98.67	98.72	99.31	99.92	99.94	99.97	99.93	99.88	99.94	98.63	98.60	99.20	99.09	98.88	99.42
OA	99.36	99.31	99.71	99.80	99.83	99.93	99.88	99.87	99.94	98.81	98.85	99.52	99.93	99.95	99.98	99.94	99.89	99.96	98.77	98.75	99.45	99.19	98.99	99.60
AA	99.53	99.59	99.64	99.85	99.88	99.89	99.86	99.91	99.90	99.35	99.38	99.47	99.96	99.98	99.98	99.94	99.96	99.96	99.12	99.23	99.29	99.59	99.53	99.61
Time (s)	1.23	5.56	44.41	1.45	5.56	47.61	0.82	5.56	47.67	2.84	5.67	43.78	5.33	14.52	72.51	2.85	9.15	61.40	3.07	5.93	49.99	2.87	20.92	78.67

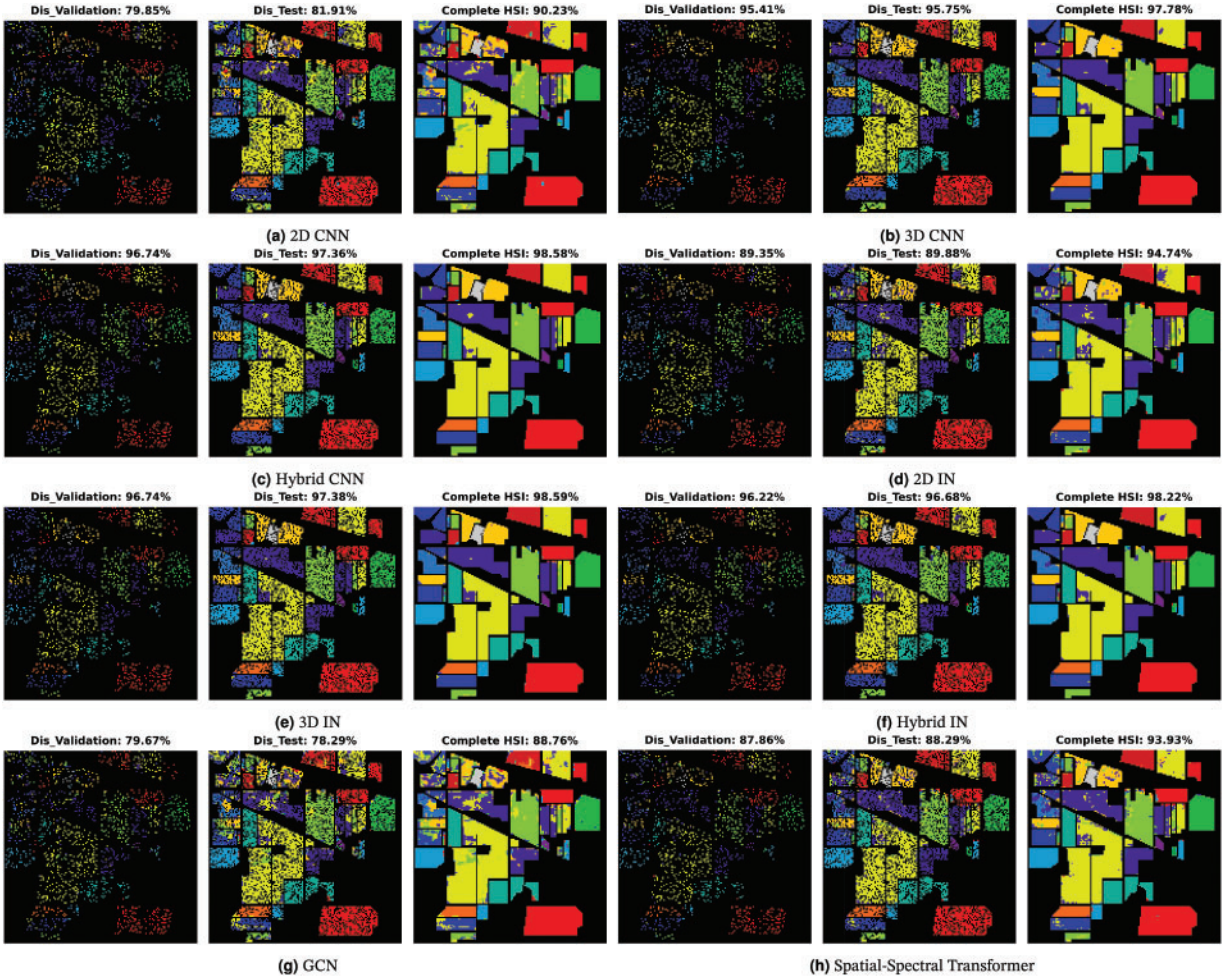

Figure 8: Indian Pines Dataset: Land cover maps for disjoint validation, test, and the entire HSI used as a test set are provided. Comprehensive class-wise results can be found in Table 6

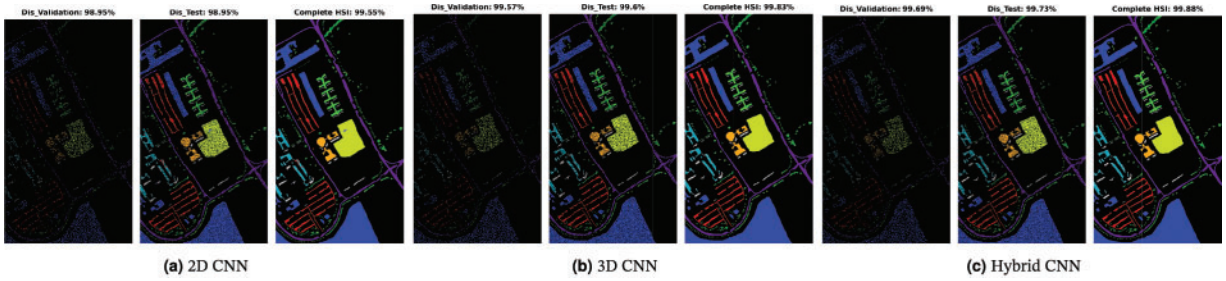

Figure 9: (Continued)

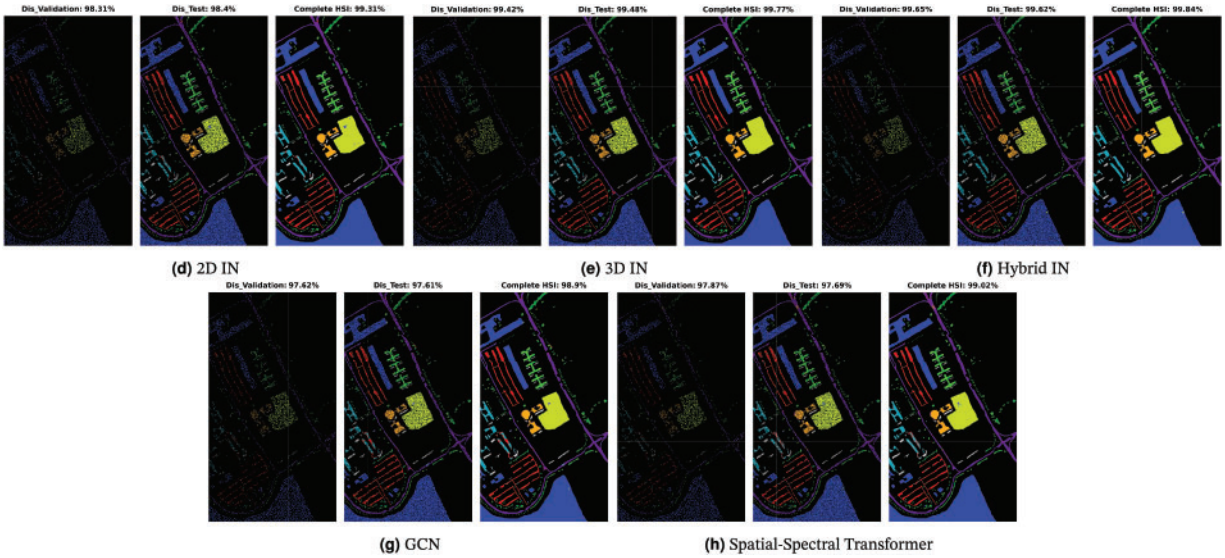

Figure 9: Pavia University Dataset: Land cover maps for disjoint validation, test, and the entire HSI used as a test set are provided. Comprehensive class-wise results can be found in Table 7

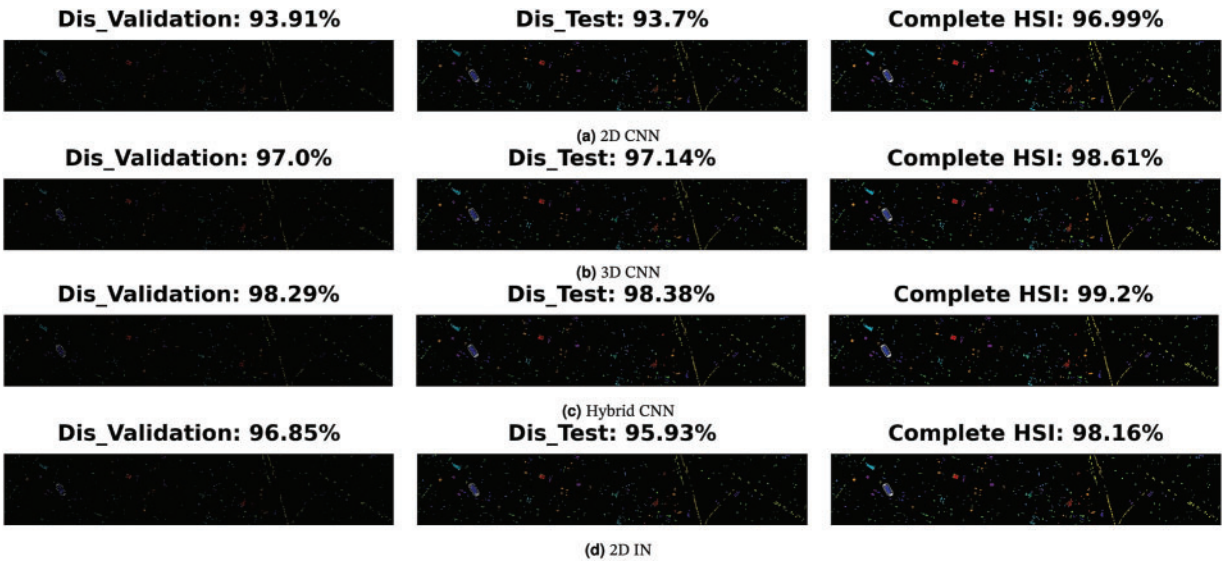

Figure 10: (Continued)

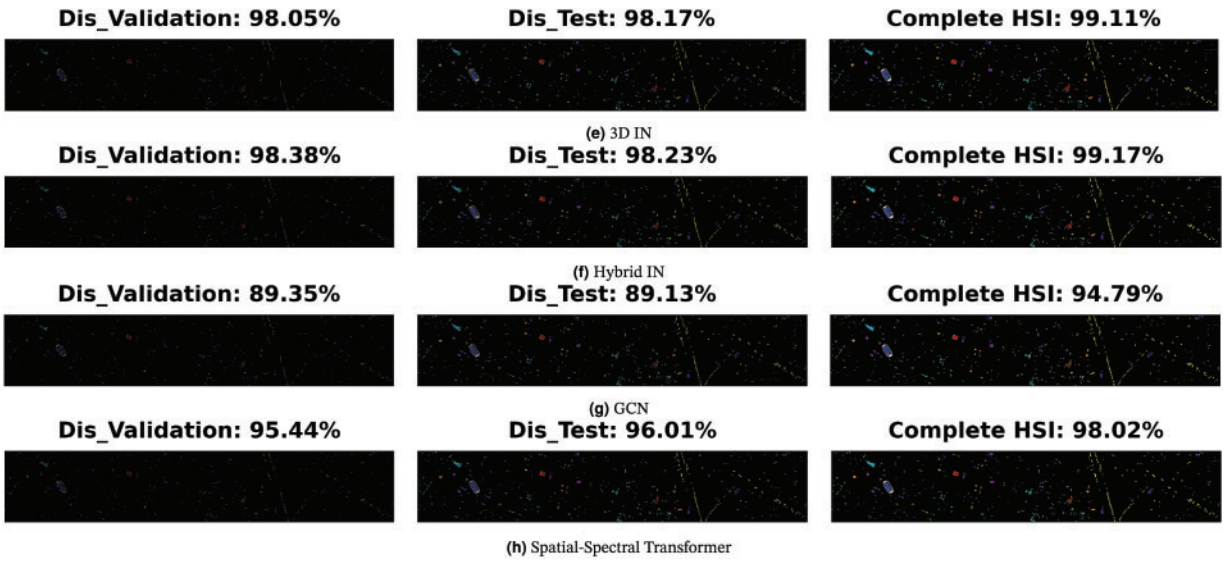

Figure 10: University Houston Dataset: Land cover maps for disjoint validation, test, and the entire HSI used as a test set are provided. Comprehensive class-wise results can be found in Table 8

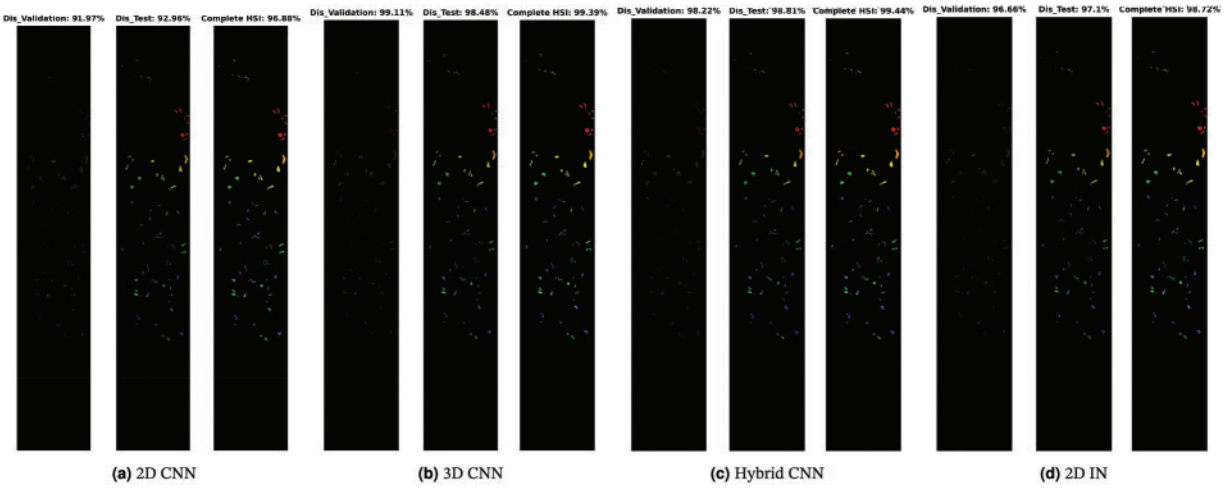

Figure 11: (Continued)

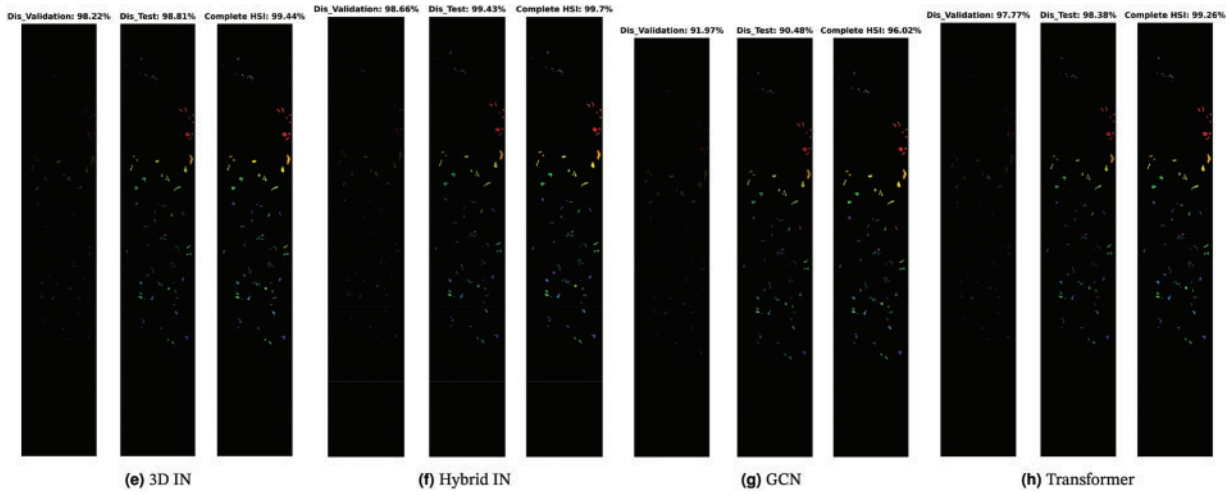

Figure 11: Botswana Dataset: Land cover maps for disjoint validation, test, and the entire HSI used as a test set are provided. Comprehensive class-wise results can be found in Table 9

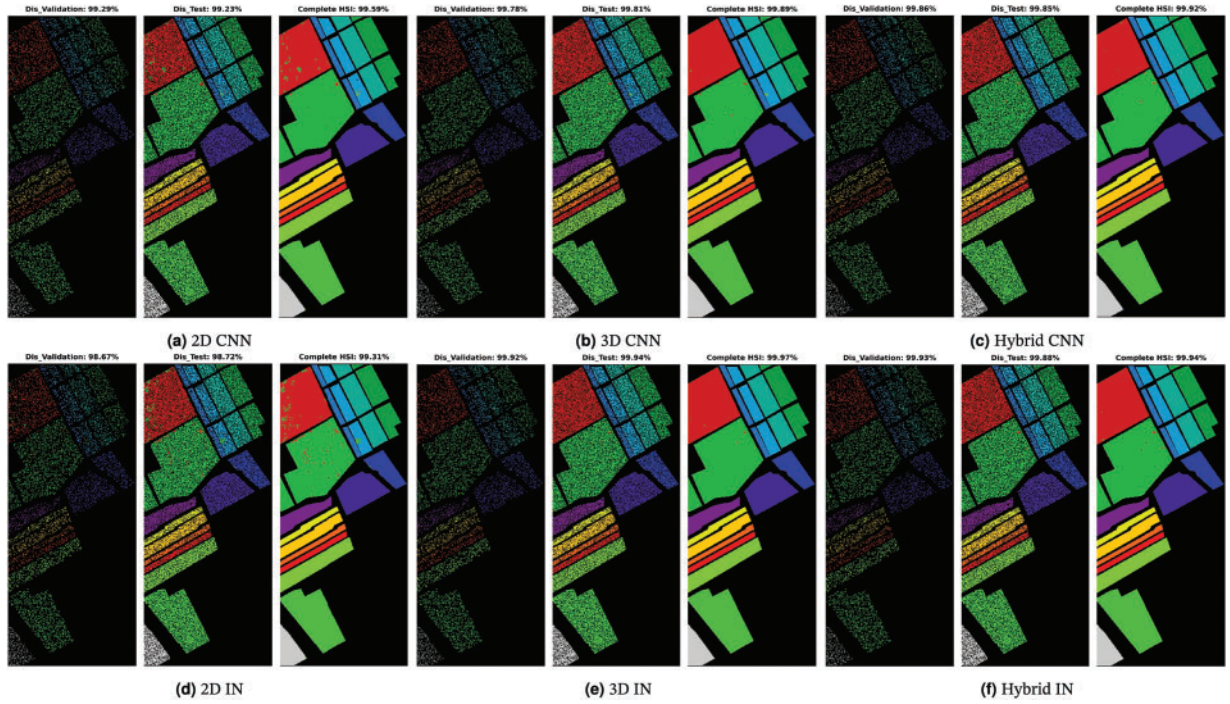

Figure 12: (Continued)

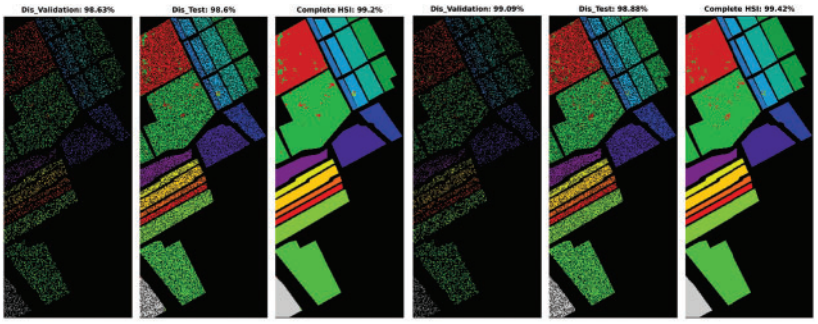

Figure 12: Salinas Dataset: Land cover maps for disjoint validation, test, and the entire HSI used as a test set are provided. Comprehensive class-wise results can be found in Table 10

The comparative methods frequently misclassify samples with similar spatial structures, exemplified by the misclassification of Meadows and Bare Soil classes in the Pavia University dataset, as illustrated in Fig. 9. Furthermore, the overall accuracy (OA) for the Grapes Untrained class is lower compared to other classes due to the aforementioned reasons as shown in Table 7. In summary, higher accuracy can be attained by employing a higher number of labeled samples (complete HSI dataset as the test set), as depicted in Figs. 8–12 and Tables 6–10, nevertheless, the elevated accuracy is accompanied by the drawbacks of bias, redundancy, and diminished generalization performance. Tables 6–10 also illustrate the computational time required to process and evaluate the HSI datasets used in this study. As depicted in the Tables, the time exhibits a gradual increase with the growing number of samples, i.e., Disjoint validation, disjoint test, and complete HSI dataset as a test set.

4 Statistical Tests

Average, overall, and Kappa accuracy may not always be appropriate measures, especially when there are significant differences in the number of samples in each class within a dataset. To clarify this point, consider the following scenario. Suppose we have a dataset with 90 healthy (positive) individuals and 10 not-healthy (negative) individuals. If a conventional model correctly predicts 90% of individuals as healthy, it might still predict the not-healthy individuals as healthy. What would be the best accuracy in this scenario?

In this setting, the model identifies 10 individuals as "False Negative", 0 as "True Positive", 0 as "False Positive", and 90 as "True Negative". Thus, the average accuracy would be 90%, i.e., $\frac{90+0}{100} = 0.9$. However, the model is highly biased since it predicts all the not-healthy individuals as healthy. In such scenarios, overall and average accuracies can be misleading or misinterpreted, indicating that these measures alone are not sufficient for evaluating a machine learning model. Therefore, it is important to consider additional statistical measures to validate the model beyond simple accuracy metrics.

Several statistical tests can be used to validate the results. For this work, we consider Recall (True Positive Rate or Sensitivity), Precision (Positive Predictive Value, PPV), and F1 score (a harmonic mean of precision and recall). In an ideal scenario, PPV should be 1, which occurs when the numerator and denominator are equal, i.e., when True Positive (TP) equals TP + False Positive (FP), making FP equal to 0. As FP increases, PPV decreases, leading to an inappropriate model. A similar trend can be observed for Recall, where False Negative (FN) replaces FP. Recall and PPV can be computed as

follows:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

For a classification model to be effective, both Precision (Positive Predictive Value, PPV) and Recall need to be high, which means that both False Positives (FP) and False Negatives (FN) must be low. In addition to Recall and PPV, the F1 score should also be computed, as it combines both Recall and PPV to provide a single metric that offers statistical significance and deeper insight into the classifier's generalization performance. The F1 score can be calculated as follows:

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

A model is considered effective if it achieves high values for PPV, Recall, and the F1 score. These metrics provide a more comprehensive evaluation of the model's performance compared to using accuracy alone. The detailed statistical results are presented in Table 11.

Table 11: Macro Average of Statistical results for various SOTA models on several benchmark Hyperspectral Datasets over disjoint test set. P = Precision; R = Recall; F1 = F1-score

Data	2D CNN			3D CNN			Hybrid CNN			2D IN			3D IN			Hybrid IN			Attention GCN			SSViT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IP	0.84%	0.84%	0.84%	0.97%	0.94%	0.95%	0.98%	0.97%	0.98%	0.92%	0.90%	0.90%	0.98%	0.92%	0.93%	0.97%	0.88%	0.89%	0.77%	0.70%	0.72%	0.90%	0.88%	0.88%
PU	0.99%	0.99%	0.99%	1.00%	0.99%	1.00%	1.00%	1.00%	1.00%	0.98%	0.99%	0.99%	1.00%	0.99%	0.99%	1.00%	1.00%	1.00%	0.98%	0.97%	0.97%	0.98%	0.97%	0.97%
SA	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	0.99%	0.99%	1.00%	1.00%	1.00%	1.00%	1.00%	1.00%	0.99%	0.99%	0.99%	1.00%	1.00%	1.00%
UH	0.95%	0.93%	0.94%	0.98%	0.97%	0.97%	0.99%	0.99%	0.99%	0.97%	0.95%	0.96%	0.98%	0.98%	0.98%	0.98%	0.98%	0.98%	0.91%	0.88%	0.89%	0.97%	0.95%	0.96%
BS	0.95%	0.92%	0.92%	0.98%	0.99%	0.98%	0.99%	0.99%	0.99%	0.97%	0.97%	0.97%	0.99%	0.99%	0.99%	1.00%	0.99%	0.99%	0.92%	0.89%	0.90%	0.99%	0.99%	0.99%

5 Conclusion

This paper introduced a novel technique for generating disjoint train, validation, and test splits in Hyperspectral Image Classification (HSIC). By efficiently partitioning the ground truth data, the proposed technique ensured unbiased performance evaluations and facilitated reliable comparisons between classification models. It proved to be a valuable tool for creating disjoint splits, guaranteeing that the subsets were representative of the entire dataset and that the classification results were not skewed by data leakage. While the technique demonstrated significant advantages, limitations were acknowledged, and opportunities for further improvement were identified. Future research could investigate alternative data-splitting strategies that incorporate additional factors, such as class imbalance or spatial coherence, to further enhance the representativeness and generalizability of the subsets. Addressing these aspects could lead to the development of even more robust and effective data-splitting techniques for HSIC.

Acknowledgement: Not applicable.

Funding Statement: The authors extend their appreciation to the Researchers Supporting Project number (RSPD2024R848), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and desing: Muhammad Ahmad, Manuel Mazzara, Salvatore Distefano; Analysis and interpretation of results: Muhammad Ahmad, Hamad Ahmed Altuwaijri, Adil Mehmood Khan; manuscript preparation: Muhammad Ahmad, Manuel Mazzara, Salvatore Distefano, Adil Mehmood Khan, Hamad Ahmed Altuwaijri. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study is publicly available and can be accessed from the corresponding data repository https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 2 January 2024). The source code for the experiments conducted in this paper can be accessed at https://github.com/mahmad00/Disjoint-Sampling-for-Hyperspectral-Image-Classification (accessed on 2 May 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Ahmad *et al.*, "Hyperspectral image classification-traditional to deep models: A survey for future prospects," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 968–999, 2022. doi: 10.1109/JSTARS.2021.3133021.

[2] S. Aksoy, E. Sertel, R. Roscher, A. Tanik, and N. Hamzehpour, "Assessment of soil salinity using explainable machine learning methods and landsat 8 images," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 130, 2024, Art. no. 103879. doi: 10.1016/j.jag.2024.103879.

[3] V. Lodhi, D. Chakravarty, and P. Mitra, "Hyperspectral imaging for earth observation: Platforms and instruments," *J. Indian Inst. Sci.*, vol. 98, no. 4, pp. 429–443, 2018. doi: 10.1007/s41745-018-0070-8.

[4] Y. Li, D. Hong, C. Li, J. Yao, and J. Chanussote, "HD-Net: High-resolution decoupled network for building footprint extraction via deeply supervised body and boundary decomposition," *ISPRS J. Photogramm. Remote Sens.*, vol. 209, no. 1, pp. 51–65, 2024. doi: 10.1016/j.isprsjprs.2024.01.022.

[5] T. Jiang, H. van der Werff, F. van Ruitenbeek, A. Dijkstra, C. Lievens and M. van der Meijde, "The influence of changing moisture content on laboratory acquired spectral feature parameters and mineral classification," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 130, 2024, Art. no. 103884. doi: 10.1016/j.jag.2024.103884.
[6] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sens.*, vol. 12, no. 16, 2020, Art. no. 2659. doi: 10.3390/rs12162659.
[7] T. Adão *et al.*, "Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1110. doi: 10.3390/rs9111110.
[8] S. Chen *et al.*, "CDasXORNet: Change detection of buildings from bi-temporal remote sensing images as an XOR problem," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 130, no. 1, 2024, Art. no. 103836. doi: 10.1016/j.jag.2024.103836.
[9] C. Li, B. Zhang, D. Hong, J. Yao, and J. Chanussot, "LRR-Net: An interpretable deep unfolding network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023. doi: 10.1109/TGRS.2023.3279834.
[10] E. Bedini, "The use of hyperspectral remote sensing for mineral exploration: A review," *J. Hyperspectr. Remote Sens.*, vol. 7, no. 4, pp. 189–211, 2017. doi: 10.29150/jhrs.v7.4.p189-211.
[11] C. Weber *et al.*, "Hyperspectral imagery for environmental urban planning," in *IGARSS 2018—2018 IEEE Int. Geosci. Remote Sens. Symp.*, IEEE, 2018, pp. 1628–1631.
[12] M. B. Stuart, A. J. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, 2019, Art. no. 3071. doi: 10.3390/s19143071.
[13] C. B. Pande and K. N. Moharir, "Application of hyperspectral remote sensing role in precision farming and sustainable agriculture under climate change: A review," in *Climate Change Impacts on Natural Resources, Ecosystems and Agricultural Systems*, Springer International Publishing, 2023, pp. 503–520. doi: 10.1007/978-3-031-19059-9_21.
[14] S. Wang, W. Han, X. Huang, X. Zhang, L. Wang, and J. Li, "Trustworthy remote sensing interpretation: Concepts, technologies, and applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 209, no. 3, pp. 150–172, 2024. doi: 10.1016/j.isprsjprs.2024.02.003.
[15] S. S. Deshpande and A. B. Inamdar, *Hyperspectral Remote Sensing in Urban Environments*. UK: Routledge Taylor & Francis Group, 2023.
[16] P. Sajadi *et al.*, "Automated pixel purification for delineating pervious and impervious surfaces in a city using advanced hyperspectral imagery techniques," *IEEE Access*, vol. 12, no. 1, pp. 82560–82583, 2024. doi: 10.1109/ACCESS.2024.3408805.
[17] A. Chauhan *et al.*, "Chapter 10-earth observation applications for urban mapping and monitoring: Research prospects, opportunities and challenges," in *Earth Observation in Urban Monitoring, Earth Observation*, A. Kumar, P. K. Srivastava, P. Saikia, R. K. Mall, Eds., Elsevier, 2024, pp. 197–229.
[18] D. Hong *et al.*, "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, 2024. doi: 10.1109/TPAMI.2024.3362475.
[19] I. Ahmad, G. Farooque, Q. Liu, F. Hadi, and L. Xiao, "MSTSENet: Multiscale spectral-spatial transformer with squeeze and excitation network for hyperspectral image classification," *Eng. Appl. Artif. Intell.*, vol. 134, no. 3, 2024, Art. no. 108669. doi: 10.1016/j.engappai.2024.108669.
[20] 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 Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022. doi: 10.1109/LGRS.2020.3043710.
[21] 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. doi: 10.1109/TGRS.2020.3015157.
[22] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Attention graph convolutional network for disjoint hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, 2024. doi: 10.1109/LGRS.2024.3356422.

[23] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (ExViT) for land use and land cover classification: A multimodal deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023. doi: 10.1109/TGRS.2023.3284671.

[24] M. Ahmad, U. Ghous, M. Usama, and M. Mazzara, "Waveformer: Spectral-spatial wavelet transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024. doi: 10.1109/LGRS.2024.3441938.

[25] M. Ahmad and M. Mazzara, "SCSNet: Sharpened cosine similarity-based neural network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–4, 2024. doi: 10.1109/LGRS.2024.3365611.

[26] T. Lu, Y. Fang, W. Fu, K. Ding, and X. Kang, "Dual-stream class-adaptive network for semi-supervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–11, 2024. doi: 10.1109/TGRS.2024.3357455.

[27] M. Ahmad, M. Usama, A. M. Khan, S. Distefano, H. A. Altuwaijri and M. Mazzara, "Spatial spectral transformer with conditional position encoding for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, 2024, Art. no. 5508205. doi: 10.1109/LGRS.2024.3431188.

[28] H. Tan *et al.*, "Data pruning via moving-one-sample-out," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine, Eds., Curran Associates, Inc., 2023, vol. 36.

[29] M. Ahmad *et al.*, "A disjoint samples-based 3D-CNN with active transfer learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022. doi: 10.1109/TGRS.2022.3209182.

[30] C. Geiß, P. Aravena Pelizar, H. Schrade, A. Brenning, and H. Taubenböck, "On the effect of spatially non-disjoint training and test samples on estimated model generalization capabilities in supervised classification with spatial features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2008–2012, 2017. doi: 10.1109/LGRS.2017.2747222.

[31] M. Ahmad *et al.*, "Spatial prior fuzziness pool-based interactive classification of hyperspectral images," *Remote Sens.*, vol. 11, no. 9, 2019, Art. no. 1136. doi: 10.3390/rs11091136.

[32] P. Zhang, H. Yu, P. Li, and R. Wang, "TransHSI: A hybrid CNN-transformer method for disjoint sample-based hyperspectral image classification," *Remote Sens.*, vol. 15, no. 22, 2023. doi: 10.3390/rs15225331.

[33] D. Hong, C. Li, B. Zhang, N. Yokoya, J. A. Benediktsson and J. Chanussot, "Multimodal artificial intelligence foundation models: Unleashing the power of remote sensing big data in earth observation," *Innov. Geosci.*, vol. 2, no. 1, 2024, Art. no. 100055. doi: 10.59717/j.xinn-geo.2024.100055.

[34] J. Yao, D. Hong, H. Wang, H. Liu, and J. Chanussot, "UCSL: Toward unsupervised common subspace learning for cross-modal image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023. doi: 10.1109/TGRS.2023.3282951.

[35] J. Yao, D. Hong, C. Li, and J. Chanussot, "SpectralMamba: Efficient mamba for hyperspectral image classification," 2024, *arXiv:2404.08489*.

[36] D. Hong *et al.*, "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," *Remote Sens. Environ.*, vol. 299, 2023, Art. no. 113856. doi: 10.1016/j.rse.2023.113856.

[37] J. Yao *et al.*, "Semi-active convolutional neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, no. 2, pp. 1–15, 2022. doi: 10.1109/TGRS.2022.3230411.

[38] U. Ghous, M. S. Sarfraz, M. Ahmad, C. Li, and D. Hong, "(2+1)D extreme xception net for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 5159–5172, 2024. doi: 10.1109/JSTARS.2024.3362936.

[39] S. K. Adari and S. Alla, "Introduction to machine learning," in *Beginning Anomaly Detection Using Python-Based Deep Learning: Implement Anomaly Detection Applications with Keras and PyTorch*. Berkeley, CA: Apress, 2024, pp. 105–134.

[40] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, 2020. doi: 10.1109/LGRS.2019.2918719.

[41] Y. Wang, X. Yu, X. Wen, X. Li, H. Dong and S. Zang, "Learning a 3-D-CNN and convolution transformers for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024. doi: 10.1109/LGRS.2024.3365615.

[42] L. Song, Z. Feng, S. Yang, X. Zhang, and L. Jiao, "Interactive spectral-spatial transformer for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, 2024. doi: 10.1109/TCSVT.2024.3386578.

[43] X. Yang *et al.*, "Synergistic 2D/3D convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 12, 2020, Art. no. 2033. doi: 10.3390/rs12122033.

[44] W. Liao, F. Wang, and H. Zhao, "Hyperspectral image classification using attention-only spatial-spectral network based on transformer," *IEEE Access*, vol. 12, pp. 93677–93688, 2024. doi: 10.1109/ACCESS.2024.3424674.

[45] T. Arshad and J. Zhang, "A light-weighted spectral-spatial transformer model for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 12008–12019, 2024. doi: 10.1109/JSTARS.2024.3419070.

[46] R. Khan, T. Arshad, X. Ma, W. Chen, H. F. Zhu and Y. N. Wu, "Deep spectral spatial feature enhancement through transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024. doi: 10.1109/LGRS.2024.3424986.

[47] H. Fırat, M. E. Asker, M. I. Bayındır, and D. Hanbay, "Hybrid 3D/2D complete inception module and convolutional neural network for hyperspectral remote sensing image classification," *Neural Process. Lett.*, vol. 55, no. 2, pp. 1087–1130, 2023. doi: 10.1007/s11063-022-10929-z.

[48] X. Zhang, "Improved three-dimensional inception networks for hyperspectral remote sensing image classification," *IEEE Access*, vol. 11, pp. 32648–32658, 2023. doi: 10.1109/ACCESS.2023.3262992.

[49] Z. Xiong, Y. Yuan, and Q. Wang, "AI-NET: Attention inception neural networks for hyperspectral image classification," in *IGARSS 2018—2018 IEEE Int. Geosci. Remote Sens. Symp.*, IEEE, 2018.

[50] X. Wu, D. Hong, and J. Chanussot, "Convolutional neural networks for multimodal remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–10, 2022. doi: 10.1109/TGRS.2022.3228927.

[51] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq, "Hyperspectral image classification using a hybrid 3D-2D convolutional neural networks," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 7570–7588, 2021. doi: 10.1109/JSTARS.2021.3099118.

[52] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, 2018. doi: 10.1109/TGRS.2018.2815613.