DATA LEAKAGE IN PATCH-BASED HYPERSPECTRAL IMAGE CLASSIFICATION: QUANTIFYING THE IMPACT OF SPATIAL OVERLAP

Mohammed Q. Alkhatib

College of Engineering and IT, University of Dubai, Dubai, 14143, UAE
mqalkhatib@ieee.org

ABSTRACT

Patch-based learning improves hyperspectral image (HSI) classification by exploiting local spectral-spatial information, but random train-test sampling from the same image can cause spatial patch overlap, leading to data leakage and optimistic performance estimates. This paper investigates same-class train-test spatial overlap in patch-based HSI classification using two measures: overlap percentage (OP), which quantifies the global amount of overlapped testing patch pixels, and average overlap ratio (AOR), which measures the local severity among affected testing patches. Experiments on the Pavia University dataset compare random and non-random spatial sampling using SVM, MLP, 2D-CNN, 3D-CNN, ViT, and MorpMamba. The results show that deep patch-based models achieve high accuracy under random sampling, with 3D-CNN reaching 96.17% Overall Accuracy (OA), but drop substantially under non-random spatial sampling, where 3D-CNN decreases to 55.20% and ViT and 2D-CNN drop by 40.71 and 38.81 percentage points (PP), respectively. Patch-size analysis further shows that increasing the patch size from 5×5 to 19×19 raises the random-sampling overlap percentage from 23.28% to 77.02%. These findings demonstrate that random patch-based evaluation can substantially inflate classification performance, especially for models that strongly exploit spatial context. The code associated with this paper is available at: https://github.com/mqalkhatib/Data_Leakage_in_HSI_Classification.

***Index Terms*—** HSI classification, Data Leakage, Spatial Overlap, Deep Learning, Patch-Based Learning

1. INTRODUCTION

Hyperspectral image (HSI) classification assigns semantic labels to pixels using their spectral responses across many contiguous bands. Compared with Optical imagery, HSI captures richer material-specific information, which improves the discrimination of visually similar land-cover classes. This makes HSI classification important for fine-grained scene understanding, particularly when different materials exhibit similar spatial appearance but distinct spectral signatures. However, the task remains challenging due to high dimensionality, spectral redundancy, spatial variability within the same class, and the limited availability of labeled samples.

Traditional methods, such as Support Vector Machines (SVMs) [1] and Multilayer Perceptrons (MLPs) [2], have been widely used for HSI classification, but they often depend mainly on spectral features and have limited ability to exploit spatial context. Deep learning methods address this limitation by learning spectral-spatial representations directly from HSI data [3]. Convolutional neural network (CNN)-based models have progressed from spectral 1D-CNNs [4] to spatial 2D-CNNs [5], joint spectral-spatial 3D-CNNs [6], and hybrid 2D-3D CNNs [7]. More recently, transformer- and Mamba-based models have been introduced to capture long-range dependencies and improve spectral-spatial modeling [8, 9].

Despite these advances, many HSI classifiers still rely on patch-based learning, where a spatial neighborhood around each labeled pixel is used as input. Although this improves spectral-spatial representation, it can introduce data leakage when training and testing samples are randomly selected from the same image. In such cases, their patches may spatially overlap, causing shared pixels to appear in both sets, weakening train-test independence and producing overly optimistic performance estimates [10, 11]. This leakage becomes more severe with larger patch sizes and higher training ratios, and is further amplified by spatial autocorrelation, where neighboring pixels often have similar spectra [11]. Although spatially disjoint sampling can reduce this issue [12], it may further reduce the already limited number of labeled training samples.

To better understand and quantify data leakage, this paper investigates the relationship between same-class train-test spatial overlap and classification performance across six classifiers and two sampling protocols. Unlike prior studies that mainly identify or mitigate leakage, this study jointly quantifies its global extent and local severity using overlap percentage and average overlap ratio. SVM, MLP, 2D-CNN, 3D-CNN, Vision Transformer (ViT), and MorpMamba are evaluated under random and non-random spatial sampling to assess model sensitivity and quantify the resulting accuracy inflation.

2. RELATED WORK

2.1. Patch-Based Learning in HSI

Patch-based learning is widely used to incorporate spatial context by extracting a local neighborhood around each labeled pixel and using it as the model input. This strategy allows classifiers to exploit local spectral-spatial correlations and has become common in CNN, Transformer, and Mamba-based HSI classification frameworks [5, 6, 7, 8, 9].

However, patch extraction introduces dependencies between samples from the same scene. Since neighboring patches may share many pixels, training and testing samples can become highly correlated even when their center pixels are different [10, 11]. This dependency is mainly controlled by the patch size and sampling strategy. Larger patches increase spatial coverage and therefore raise the probability of train-test overlap, while random sampling from a single image can place training and testing pixels close to each other due to spatial continuity [10, 11].

2.2. Data Leakage in Remote Sensing and Computer Vision

Data leakage occurs when testing information becomes indirectly available during training, leading to overly optimistic performance estimates. In patch-based HSI classification, random sampling from the same image may place training and testing samples in spatially correlated regions, causing their patches to overlap and share pixels [10, 11]. Spatially disjoint, iterative, and augmentation-based sampling strategies have been proposed to reduce this dependence [12, 13, 10], but they may reduce the number of usable labeled samples and primarily focus on mitigation.

Comparatively little attention has been given to jointly quantifying the global extent and local severity of same-class train-test patch overlap and relating these characteristics to performance degradation across different classifier families under matched experimental conditions. This work addresses this gap using two complementary overlap measures evaluated across traditional machine-learning, CNN-based, Transformer-based, and Mamba-based classifiers under random and leakage-reduced spatial sampling protocols.

3. METHODOLOGY

3.1. Patch-Based HSI Classification

The general architecture of most deep learning frameworks for hyperspectral image classification is illustrated in Fig. 1. The process begins with a hyperspectral image $\mathbf{I}_{\text{Original}} \in \mathbb{R}^{H\times W\times C}$. To reduce spectral redundancy and computational cost, Principal Component Analysis (PCA) is applied, resulting in a compressed image $\mathbf{I}_{\text{Reduced}} \in \mathbb{R}^{H\times W\times P}$, where $P \ll C$. Spatial patches of size $M \times M \times P$ are then extracted and used as inputs to a deep learning model, such as CNN, ViT, or Mamba-based architectures. The extracted features are subsequently passed through a classification head

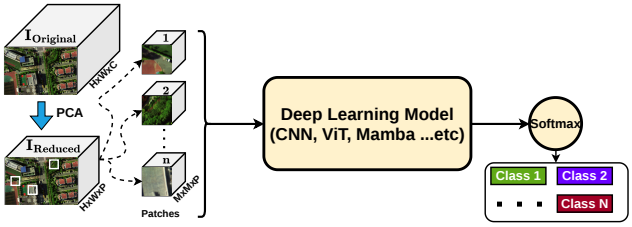


Fig. 1: General Patch-based hyperspectral image classification framework using PCA and deep learning models.

with a softmax activation function to produce the final class predictions.

3.2. Quantifying Train-Test Spatial Overlap

To measure the amount of same-class spatial overlap between training and testing patches, two simple measures are used. Let T_{tr} and T_{te} denote the training and testing center pixels, respectively. For any pixel p, the patch extracted around it is denoted by $\Omega_M(p)$, where M is the patch size.

For a class c, the spatial coverage of its training patches is defined as

$$\Omega_{\text{tr}}^{(c)} = \bigcup_{\substack{p\in T_{\text{tr}}\\ c_p=c}} \Omega_M(p), \tag{1}$$

where c_p denotes the class label of the training center pixel p.

A testing patch is considered overlapped when it shares pixels with the training patch coverage of the same class. This same-class condition is used because overlap between patches from different classes does not represent direct class-consistent leakage.

The first measure is the overlap percentage (OP). It measures how many pixels inside the testing patches are also covered by training patches from the same class. It is computed as:

$$OP = \frac{\sum_{q\in T_{\text{te}}} \left|\Omega_M(q) \cap \Omega_{\text{tr}}^{(c_q)}\right|}{\sum_{q\in T_{\text{te}}} |\Omega_M(q)|} \times 100, \tag{2}$$

where c_q is the class label of the testing center pixel q. A higher value of OP means that a larger portion of all testing-patch pixels is also contained in same-class training patches.

To compute the second measure, the overlap ratio of each testing patch q is first calculated as:

$$OR(q) = \frac{\left|\Omega_M(q) \cap \Omega_{\text{tr}}^{(c_q)}\right|}{|\Omega_M(q)|}. \tag{3}$$

The second measure, termed the average overlap ratio (AOR), is then obtained by averaging $OR(q)$ over only the testing patches affected by same-class overlap:

$$AOR = \frac{1}{|Q|} \sum_{q\in Q} OR(q), \tag{4}$$

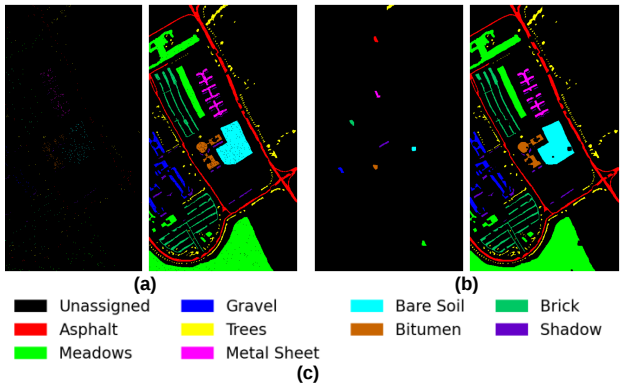


Fig. 2: Examples of the sampling settings used in this study: (a) random sampling, where training and testing patches may overlap; (b) non-random spatial sampling, where training and testing regions are selected with greater spatial separation; (c) Class Labels.

where $Q = \{q \in T_{\text{te}} : OR(q) > 0\}$ is the set of testing patches with non-zero same-class overlap.

Therefore, OP measures the global prevalence of leakage across the complete testing set, whereas AOR measures the conditional severity of leakage among only the affected testing patches. Consequently, a sampling protocol may produce a low OP but a relatively high AOR when only a small number of testing patches overlap substantially with same-class training patches.

3.3. Non-Random Spatial Sampling Protocol

To examine the effect of spatial overlap in patch-based HSI classification, two sampling protocols are considered, as illustrated in Fig. 2. The first follows the conventional random strategy, where labeled pixels are divided into training, validation, and testing subsets. Since each pixel is represented by a spatial patch, nearby training and testing samples may share pixels, introducing spatial dependency and potentially inflating classification performance.

The second protocol uses non-random spatial sampling, where samples are selected with greater spatial separation. This reduces direct mixing between training and testing regions and provides a stricter evaluation setting. Since testing pixels are not explicitly excluded after patch extraction, this setting is considered leakage-reduced rather than fully leakage-free. Training samples are selected first. Validation samples are then drawn from the remaining labeled pixels and used only for model monitoring and selection, while all other labeled pixels form the test set used for the final performance evaluation.

Comparing the two protocols allows the sensitivity of each classifier to train-test spatial overlap to be evaluated. A large performance drop under non-random spatial sampling suggests that the model may have benefited from spatial dependency under random patch-based evaluation.

3.4. Experimental Setup

The experiments are conducted on the Pavia University hyperspectral dataset, which contains nine labeled land-cover classes [14]. Before patch extraction, PCA is applied to reduce the spectral dimensionality to $P = 15$ components, and each labeled pixel is represented by a $9 \times 9 \times 15$ spatial-spectral patch.

For both sampling protocols, a class-balanced training set is constructed by selecting 94 samples per class, corresponding to approximately one-tenth of the smallest class, Shadow, which contains 947 labeled samples. In addition, 18 samples per class are used for validation and reserved only for model monitoring and selection, while all remaining labeled pixels are used for testing.

The evaluated classifiers include SVM [1], MLP [2], 2D-CNN [5], 3D-CNN [6], ViT [15], and MorpMamba [9]. The same training configuration is used under both sampling protocols so that performance differences are mainly attributed to the sampling strategy. All experiments are repeated over 10 independent runs, and the results are reported as mean$\pm$standard deviation using Overall Accuracy (OA), Average Accuracy (AA), Kappa coefficient, and class-wise accuracy.

4. RESULTS AND DISCUSSION

Tables 1 and 2 present the classification results under random and non-random patch-based sampling, respectively. Under random sampling, deep spectral-spatial models clearly outperform traditional classifiers. The 3D-CNN achieves the highest OA of 96.17%, followed by ViT, 2D-CNN, and MorpMamba with 91.78%, 86.97%, and 81.07%, respectively, whereas SVM and MLP obtain 62.15% and 69.50%. A similar trend is observed for AA and Kappa, indicating that deep patch-based models benefit strongly when training and testing samples are randomly distributed within the same image.

This advantage decreases substantially under the non-random spatial setting. As summarized in Table 3, all models experience an OA reduction, with the largest drops observed for 3D-CNN, ViT, and 2D-CNN at 40.97, 40.71, and 38.81 percentage points, respectively. MorpMamba decreases by 32.62 percentage points, while SVM and MLP show smaller drops of 10.90 and 19.35 percentage points. These results indicate that models explicitly exploiting spatial neighborhoods are more sensitive to the sampling protocol.

The class-wise results show a similar pattern. For 3D-CNN, accuracy decreases from 94.28% to 46.27% for Asphalt, from 97.63% to 49.78% for Meadows, from 91.59% to 43.99% for Gravel, and from 96.38% to 47.89% for Bare Soil. These large reductions mainly occur in spatially variable classes. In contrast, compact and spectrally distinctive classes, such as Metal Sheet and Shadow, remain comparatively easier to classify under non-random sampling.

Table 1: Class-wise and overall classification performance under the random patch-based evaluation setting with potential same-class train-test spatial overlap. The reported values are presented as mean±standard deviation over 10 runs.

No	Name	Train	Val	Test	Total	SVM	MLP	2D-CNN	3D-CNN	ViT	MorpMamba
1	Asphalt	94	18	6,519	6,631	62.83±1.01	64.51±3.11	83.07±8.19	94.28±1.95	89.26±3.32	75.67±6.86
2	Meadows	94	18	18,537	18,649	49.46±4.91	64.17±8.45	84.98±5.47	97.63±1.17	91.32±4.04	78.05±7.50
3	Gravel	94	18	1,987	2,099	21.40±8.34	69.26±12.62	83.04±4.28	91.59±3.10	88.94±3.86	74.49±16.57
4	Trees	94	18	2,952	3,064	95.32±1.48	96.14±1.50	96.98±1.04	97.74±1.11	97.71±1.11	95.45±1.64
5	Metal Sheet	94	18	1,233	1,345	99.63±0.15	99.65±0.20	99.50±0.53	99.99±0.02	99.99±0.02	99.94±0.06
6	Bare Soil	94	18	4,917	5,029	67.18±5.16	64.71±11.94	88.51±3.45	96.38±1.56	89.44±4.82	88.41±6.44
7	Bitumen	94	18	1,218	1,330	87.40±1.55	87.23±5.45	92.13±3.65	97.04±1.60	96.49±1.68	85.34±7.43
8	Brick	94	18	3,570	3,682	84.74±5.07	67.43±8.04	87.20±4.04	90.51±4.16	92.44±2.03	76.15±11.94
9	Shadow	94	18	835	947	99.77±0.13	99.72±0.30	99.64±0.23	99.92±0.11	99.71±0.34	99.08±0.79
		OA (%)				62.15±1.68	69.50±3.22	86.97±3.27	96.17±0.80	91.78±1.30	81.07±3.65
		AA (%)				74.19±0.54	79.20±1.90	90.56±1.60	96.12±0.59	93.92±0.83	85.84±2.43
		Kappa × 100				54.01±1.56	61.77±3.37	83.10±4.02	94.90±1.05	89.19±1.61	75.83±4.30

Table 2: Class-wise and overall classification performance under the non-random spatial sampling setting designed to reduce same-class train-test spatial overlap. The reported values are presented as mean±standard deviation over repeated runs.

No	Name	Train	Val	Test	Total	SVM	MLP	2D-CNN	3D-CNN	ViT	MorpMamba
1	Asphalt	94	18	6,519	6,631	61.04±12.66	64.43±9.44	50.30±18.10	46.27±16.32	46.63±14.61	49.11±15.48
2	Meadows	94	18	18,537	18,649	38.72±12.41	33.66±14.46	41.56±10.45	49.78±18.88	50.26±17.19	37.60±13.61
3	Gravel	94	18	1,987	2,099	29.85±12.74	36.51±22.81	34.83±22.72	43.99±20.93	31.36±17.46	30.97±20.54
4	Trees	94	18	2,952	3,064	76.88±19.95	80.39±13.05	62.17±18.31	82.36±11.12	57.62±21.13	73.17±19.35
5	Metal Sheet	94	18	1,233	1,345	99.60±0.22	99.45±0.49	92.99±4.96	99.76±0.47	98.90±1.57	97.31±2.93
6	Bare Soil	94	18	4,917	5,029	45.81±11.61	48.65±12.35	38.41±6.55	47.89±13.47	39.51±11.01	42.46±14.88
7	Bitumen	94	18	1,218	1,330	66.79±19.21	74.52±17.45	72.00±16.20	89.92±5.38	64.78±20.57	77.78±15.34
8	Brick	94	18	3,570	3,682	63.36±12.02	57.38±24.16	53.63±17.66	56.55±20.72	56.75±17.02	64.94±15.13
9	Shadow	94	18	835	947	99.40±0.39	99.70±0.11	93.08±3.74	96.51±5.36	80.85±12.76	88.28±7.01
		OA (%)				51.25±5.43	50.15±6.91	48.16±5.96	55.20±9.21	51.07±7.42	48.45±9.26
		AA (%)				64.60±3.76	66.08±3.22	59.89±4.78	68.12±3.02	58.52±3.57	62.40±4.92
		Kappa × 100				41.84±5.09	41.27±6.44	38.06±6.45	45.96±8.64	40.54±6.81	39.15±9.06

Table 3: Overall accuracy degradation, in percentage points, from random sampling to non-random spatial sampling.

Model	Random OA (%)	Non-Random OA (%)	OA Drop (pp)
SVM	62.15	51.25	10.90
MLP	69.50	50.15	19.35
2D-CNN	86.97	48.16	38.81
3D-CNN	96.17	55.20	40.97
ViT	91.78	51.07	40.71
MorpMamba	81.07	48.45	32.62

Overall, Tables 1 and 2 indicate that the high accuracy obtained under random patch-based evaluation is partly influenced by same-class train-test spatial dependency. Neighboring patches may share pixels or highly similar local structures, allowing information from training regions to be indirectly reflected in the test set. Random sampling may therefore produce optimistic performance estimates, whereas non-random spatial sampling provides a stricter assessment of model generalization.

4.1. Effect of Patch Size on Same-Class Spatial Leakage

Table 4 shows the effect of patch size on the same-class OP and AOR, computed using Eqs. (2) and (4), respectively. Under random sampling, OP increases from 23.28% with a 5×5 patch to 77.02% with a 19×19 patch, showing that larger patches substantially increase the portion of testing regions covered by training patches. AOR follows the same trend, increasing from 47.29% to 79.91%, which indicates that larger patches also increase the severity of overlap among the affected testing patches. In contrast, the non-random spatial setting maintains a much lower overlap percentage, increasing only from 0.83% to 4.66% as the patch size grows from 5×5 to 19×19. This confirms that the non-random sampling strategy greatly reduces the global amount of same-class train-test overlap. However, AOR remains relatively high, decreasing from 45.26% to around 39% for larger patch sizes. This is because Eq. (4) is computed only over testing patches with non-zero overlap; thus, although few testing patches are affected, the remaining overlap cases may still contain considerable local overlap.

Table 4: Effect of patch size on same-class overlap percentage and average overlap ratio under random and non-random spatial sampling settings. The values are computed from the training and testing patch locations, not from model predictions. Only mean values are reported.

Patch Size	Overlap Percentage (%)		Average Overlap Ratio (%)	
	Random	Non-Random	Random	Non-Random
5×5	23.28	0.83	47.29	45.26
7×7	35.03	1.28	52.98	43.79
9×9	45.29	1.75	58.45	42.26
11×11	54.04	2.26	63.61	40.50
13×13	61.48	2.80	68.31	38.98
15×15	67.73	3.39	72.70	38.86
17×17	72.86	4.01	76.60	39.13
19×19	77.02	4.66	79.91	39.42

Overall, patch size directly controls both the extent and severity of spatial leakage. OP measures the global amount of leakage, whereas AOR provides complementary information about the severity of the affected cases. This distinction

is important in leakage-reduced settings, where the number of affected testing patches may be small even when their local overlap remains high.

5. CONCLUSION

This paper investigated same-class train-test spatial overlap in patch-based hyperspectral image classification using two measures: OP and AOR. The results showed that random patch-based sampling can substantially inflate classification performance, especially for deep models that exploit spatial context. Under non-random spatial sampling, 3D-CNN, ViT, and 2D-CNN showed large OA drops of 40.97, 40.71, and 38.81 percentage points, respectively, while SVM and MLP showed smaller reductions. The patch-size analysis further confirmed that larger patches increase spatial leakage, with the random-sampling overlap percentage rising from 23.28% for 5×5 patches to 77.02% for 19×19 patches.

Future work will extend this analysis to additional hyperspectral datasets and different training ratios to further assess the generality of the observed leakage effects. More rigorous spatial separation strategies will also be investigated to reduce train-test dependency while preserving sufficient labeled samples for training. In addition, future studies may examine how leakage affects newer deep architectures and whether model design choices can improve robustness under leakage-reduced evaluation settings.

6. REFERENCES

[1] Farid Melgani and Lorenzo 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.

[2] Javier Plaza, Antonio Plaza, Rosa Pérez, and Pablo Martínez, "Parallel classification of hyperspectral images using neural networks," in *Computational intelligence for remote sensing*, pp. 193–216. Springer, 2008.

[3] Mohammed Q. Alkhatib, "ConvViTMamba: efficient multiscale convolution, Transformer, and Mamba-based sequence modelling for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 0, no. 0, pp. 1–40, 2026.

[4] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, pp. 1–12, 2015.

[5] Konstantinos Makantasis, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE international geoscience and remote sensing symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[6] Amina Ben Hamida, Alexandre Benoit, Patrick Lambert, and Chokri Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[7] Swalpa Kumar Roy, Gopal Krishna, Shiv Ram Dubey, and Bidyut B Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2019.

[8] Mohammed Q Alkhatib and Ali Jamali, "HSIFormer: An Efficient Vision Transformer Framework for Enhanced Hyperspectral Image Classification Using Local Window Attention," in *2024 14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2024, pp. 1–5.

[9] Muhammad Ahmad, Muhammad Hassaan Farooq Butt, Adil Mehmood Khan, Manuel Mazzara, Salvatore Distefano, Muhammad Usama, Swalpa Kumar Roy, Jocelyn Chanussot, and Danfeng Hong, "Spatial–spectral morphological mamba for hyperspectral image classification," *Neurocomputing*, vol. 636, pp. 129995, 2025.

[10] Hao Feng, Yongcheng Wang, Zheng Li, Ning Zhang, Yuxi Zhang, and Yunxiao Gao, "Information leakage in deep learning-based hyperspectral image classification: A survey," *Remote Sensing*, vol. 15, no. 15, pp. 3793, 2023.

[11] Kaizhe Zhan, Zijie Huo, Zhijian Yin, and Zhen Yang, "Rethinking data leakage in patch-based hyperspectral image classification with traditional deep networks," *Remote Sensing*, vol. 18, no. 7, pp. 979, 2026.

[12] Muhammad Ahmad, Usman Ghous, Danfeng Hong, Adil Mehmood Khan, Jing Yao, Shaohua Wang, and Jocelyn Chanussot, "A disjoint samples-based 3d-cnn with active transfer learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[13] Chein-I Chang, Chia-Chen Liang, and Peter Fuming Hu, "Iterative random training sampling convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–26, 2023.

[14] "Hyperspectral remote sensing scenes," https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, Accessed: 31-May-2026.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.