Hyperspectral Target Detection-Based 2D-3D Parallel Convolutional Neural Networks for Hyperspectral Image Classification

Shih-Yu Chen, *Member*, *IEEE,* Kai-Hsun Hsu and Tzu-Hsien Kuo

***Abstract*—This paper presents a novel hyperspectral target detection based 2D-3D parallel convolutional neural network (HTD 2D-3D-PCNN) model, which integrates the hyperspectral target detection (HTD) technique to achieve outstanding performance in hyperspectral image classification (HSIC). The proposed model effectively leverages both spectral and spatial information through a dual-branch architecture in hyperspectral imaging (HSI). In the first branch, HTD is utilized to enhance the spectral features of targets of interest, while suppressing background. The resulting enhanced image is then inputted to a 2D-CNN, augmented with an additional deconvolution layer to highlight spatial characteristics. Concurrently, the second branch employs dimensionality reduction via PCA, and a 3D-CNN is employed to capture both spectral and spatial attributes. Subsequently, the feature maps from both CNNs are combined and processed through fully connected layers for classification. To validate the effectiveness of the proposed HTD-2D-3D-PCNN, extensive experiments are conducted on five widely used hyperspectral public datasets (Indian Pines, Pavia University, Salinas Scene, Kennedy Space Center, and Botswana) with a consistent training sample ratio of either 10% or 5%. The results show that HTD-2D-3D-PCNN can achieve overall accuracy values of 98.41%, 99.85%, 99.92%, 99.82%, and 98.82% for the respective datasets, surpassing the performance of recent methodologies.**

***Index Terms*—Convolutional neural networks (CNN), Hyperspectral image classification (HSIC), Hyperspectral target detection (HTD).**

I. Introduction

OVER the past two decades, hyperspectral remote sensing has witnessed significant advancements. Leveraging the unique spectral characteristics of different materials, such as their reflectance and absorption properties, enables more precise detection and identification of various substances. Hyperspectral image classification (HSIC) technology serves as the foundation for numerous applications, including precision agriculture [1–3], forest monitoring [4], urban planning [5], food safety [6], and industrial production [7].

In the past, conventional classification methods such as Support Vector Machine (SVM) [8], K Nearest Neighbor (KNN) [9], Random Forest [10], Logistic Regression [11], and sparse representation [12] were commonly used in HSIC. However, recent years have brought significant advancements in deep learning for various computer vision tasks. In hyperspectral image classification, techniques like convolutional neural networks (CNN) [13–15], recurrent neural networks (RNN) [16–18], generative adversarial networks (GAN) [19], graph convolutional networks (GCN) [20], deep belief networks (DBN) [21], and autoencoders (AE) [22] have been widely adopted and have produced impressive classification results. Notably, 2D-CNN has become a prevalent approach in hyperspectral image classification (HSIC) for capturing spatial features. Several studies, including the work by M. Ahmad et al. [23], have compiled various classification techniques utilizing 2D-CNN.

2D-CNN has become a common approach for spatial feature extraction. For instance, Ding et al. [24] introduced an adaptive kernel approach combined with 2D-CNN, which improved the classification accuracy through learned convolution kernels. Chen et al. [25] addressed overfitting using a combination of Gabor filters and 2D-CNN. [26] used 2D-CNN with multiscale covariance maps for effective feature extraction. Other methods, like Pyramidal Residual Network (pResNet) [27] and CNNDH [28], integrate 2D-CNN with different techniques to enhance feature extraction.

However, solely relying on 2D-CNN might not effectively capture spectral features. The incorporation of 3D-CNN to simultaneously extract spectral and spatial features has shown enhanced performance. For instance, [29] introduced the 3D-Deep CNN model for deep feature extraction. Another study by [30] integrated stacked auto encoders with a spatial-spectral feature learning network for improved classification. Furthermore, [31] combined 3D-CNN with GAN [32] to process overlapping 3D patches with a 3D kernel function, and [33] proposed a multi-scale 3D CNN for end-to-end spatial and spectral feature learning. SSRN [34] designed an end-to-end spectral-spatial residual network.

Previous studies have identified limitations when using either 2D-CNN or 3D-CNN independently. In response,

This work received financial support from the "Intelligent Recognition Industry Service Center" as part of the Featured Areas Research Center Program under the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, and the National Science and Technology Council, Taiwan, under grants [109-2628-E-224-001-MY3] and [112-2622-E-224-003]. Additionally, partial support was provided by Isuzu Optics Corporation. (Corresponding author: Shih-Yu Chen).

S.-Y. Chen, Kai-Hsun Hsu and Tzu-Hsien Kuo are with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan, ROC (e-mail: sychen@yuntech.edu.tw)

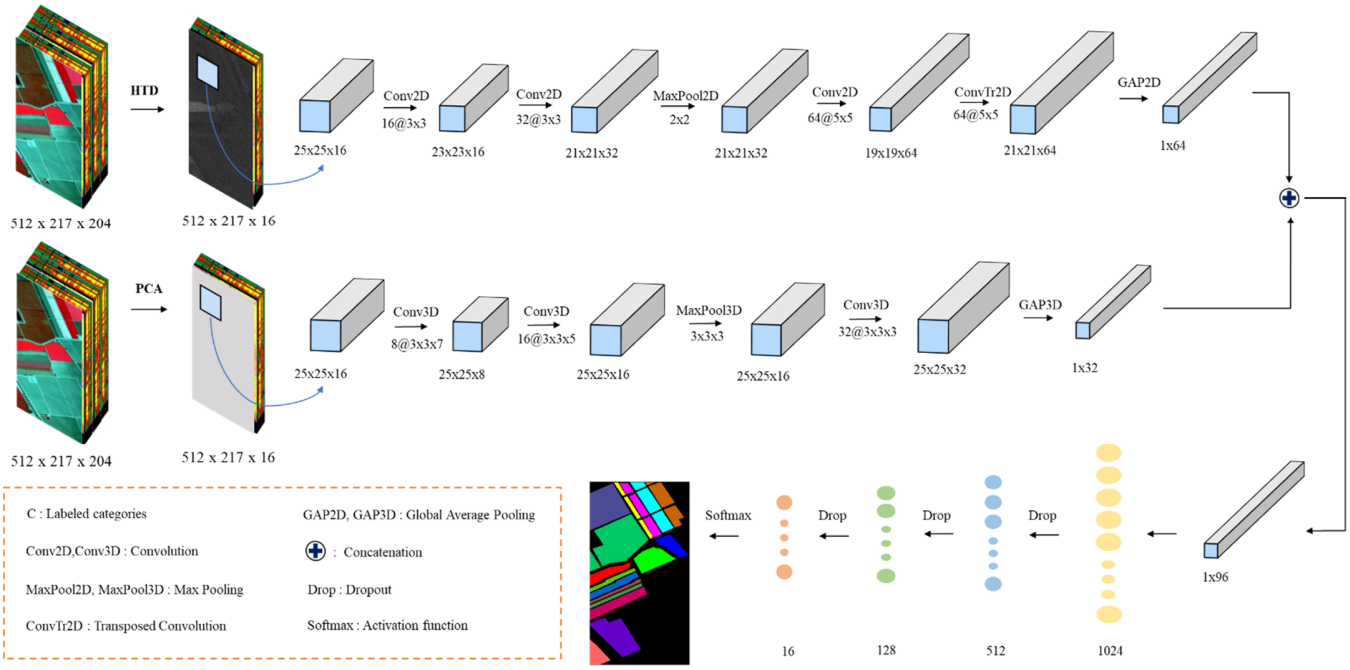

Fig. 1. Proposed HTD-2D-3D-PCNN architecture.

researchers have proposed various hybrid models. For instance, some models adopt a 3D in series 2D hybrid approach, such as Attention-fused hybrid network (AfNet) [35], HybridSN [36], Hybrid 3D-2D CNN [37], and hybrid 3D/2D CNN [38]. Others utilize a multi-branch approach, combining multiple channels into parallel or serial 3D in series 2D structures, like Multibranch 2D–3D CNN [39]. Conversely, some models employ a 2D in series 3D architecture, as seen in simplified 2D-3D CNN [40]. Additionally, end-to-end multilevel hybrid attention networks, such as DMCN [41], combine both types of CNNs, incorporating attention mechanisms to overcome limitations. Models like Collaborative Attention CNN (CACNN) [42], Feedback Attention-Based Dense CNN (FADCNN) [43], and FFDN [44] integrate attention mechanisms or fusion strategies to enhance hyperspectral image classification results. These hybrid models offer promising solutions by integrating both types of CNNs and attention mechanisms, leading to improved classification results in hyperspectral image analysis.

Recent progress in hyperspectral remote sensing image classification, as demonstrated in studies like MGSNet [45], has introduced novel approaches such as the Target-Background Separation Strategy. In MGSNet, background information beyond the effective target is utilized as a decision aid, enhancing distinguishability among samples with similar targets but varying backgrounds. This strategic approach aligns conceptually with the Hyperspectral Target Detection (HTD) methodology proposed in this paper, which emphasizes and highlights the target category. The key difference lies in the utilization of the HTD algorithm in this paper, reinforcing the spectral features of the target, with each category requiring only one target spectral feature for implementation.

In a similar vein, SLA-Net [46] adopts a multistream design that incorporates morphological transformations through trainable structuring elements, specifically aiming to extract fine-grained spatial details. SLA-Net focuses on enhancing tree species classification primarily through spatial features rather than spectral features. In the context of classifying complex objects, the utilization of spatial morphological differences proves beneficial in delineating boundaries for fine-grained categories.

Similarly, SpectralGPT [47] introduces a revolutionary Multi-Target Reconstruction Strategy to effectively capture spatial-spectral characteristics and spectrally sequential information. SpectralGPT excels in learning spatial-spectral features. However, the approach proposed in this paper diverges from these methodologies by initially enhancing spectral features through HTD. Subsequently, a parallel deployment of 2D-CNN and 3D-CNN is employed to simultaneously extract spatial-spectral features, resulting in a notable enhancement in classification performance.

In the case of RSRNet [48], to alleviate classifier bias and maintain the stability of the decision boundary, a status replay strategy (SRS) is constructed to regulate the learning and optimization of the classifier. For HighDAN [49], the solution involves a high-resolution domain adaptation network (HighDAN) that bridges feature and category differences between different cities.

The proposed architecture in this paper distinguishes itself from previous research in several ways:

1) The innovative HTD-2D-3D-PCNN model features a dual-branch architecture. In one branch, the HTD algorithm, in conjunction with the optimal signature generation process (OSGP), is designed to enhance spectral features, suppress background noise, reduce bands, and optimize feature

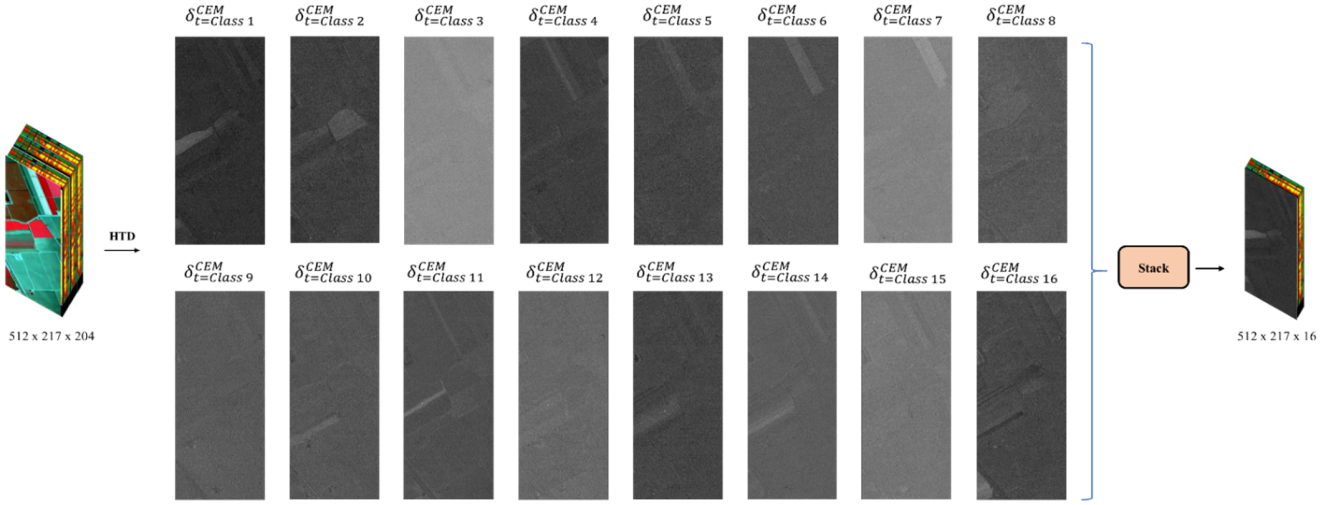

Fig. 2. Data processing of HTD.

extraction and spatial learning efficiency in subsequent 2D CNN models.

2) Concurrently, the second branch integrates PCA and 3D-CNN for spectral and spatial feature extraction. The resulting feature maps from both branches are merged, thereby preserving the advantageous features of the original PCA-3D-CNN model. Additionally, the inclusion of HTD-2DCNN further enhances the learning of spectral and spatial characteristics.

3) The proposed approach adeptly addresses class imbalance and enhances learning effectiveness, with a particular emphasis on benefiting smaller class sizes and capturing fine details at the edges. These enhancements significantly contribute to the outstanding performance observed, further boosting the classification accuracy of the network model.

4) Deconvolution layers were thoughtfully inserted within the 2D-CNN to elevate the prominence of spatial features in images. Moreover, global average pooling (GAP) replaced Flatten layer to counteract overfitting during training.

In our experiments, we conducted evaluations on five well-established hyperspectral image (HSI) public datasets. These evaluations employed a consistent training sample ratio across all datasets. The outcomes of our experiments unequivocally establish the superiority of our proposed architecture compared to recent methodologies of a similar nature. Notably, our approach excels in scenarios involving smaller class sizes and the nuanced details at the edges, as evident from the improvements in overall accuracy, average accuracy, and the KAPPA coefficient.

II. Hyperspectral Target Detection

Hyperspectral imaging contains valuable spectral information about different materials, allowing researchers to identify the types of materials represented by each pixel. This process, known as hyperspectral target detection (HTD), can be carried out using active or passive methods. Active HTD methods can be broadly classified into three categories [50]: hypothesis testing-based detection, signal to noise ration-based detection, and spectral angle-based detection. Among these, the SNR-based detectors, such as constrained energy minimization (CEM) and normalized adaptive matched detector (AMD), stand out for their effective background suppression and improved target detectability. The squared version of these detectors further enhances the target's spectral features and increases the separation between target and background. In our experiments, we chose CEM, CEM2, NAMD, and NAMD2 as the target detectors, and their impact on the model's performance were thoroughly explored and analyzed. This chapter delves into the details of the HTD algorithms used in our experiments and emphasizes the significance of extracting spectral features for accurate material identification.

A. Constrained Energy Minimization

The constrained energy minimization (CEM) algorithm in active hyperspectral target detection [50] demonstrates stability and excellence in subpixel detection. One of the main advantages of CEM is its ability to perform target detection with only one spectral feature (also known as the target signature or region of interest), set as parameter t, without requiring prior knowledge of other targets or background information. This simplicity is a key strength of CEM.

Another advantage of CEM lies in its approach of transposing the data's correlation matrix **R**. The correlation matrix is defined as $\mathbf{R} = (1/N)\sum_{i=1}^{N} r_i r_i^T$, where r denotes the spectral response of each pixel in the data. By utilizing correlation matrix **R**, CEM achieves background suppression and enhances detection capabilities through the customization of finite impulse response (FIR) filters, which are matched with target feature t. The formula for the CEM detector is as follows:

$$\delta^{\mathrm{CEM}}(\mathrm{r}) = \frac{\mathbf{t}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}}{\mathbf{t}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{t}} \quad (1)$$

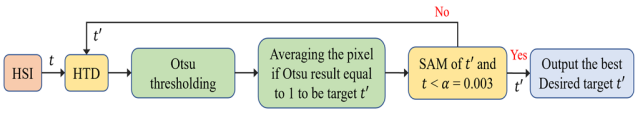

Fig. 3. Flowchart of the optimal signature generation process.

If we square the entire formula, it becomes the CEM^2 detector:

$$\left(\delta^{\mathrm{CEM}}(\mathrm{r})\right)^2 = \left[\frac{\mathbf{t}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}}{\mathbf{t}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{t}}\right]^2 \tag{2}$$

B. Normalized Adaptive Matched Detector

The normalized adaptive matched detector (AMD) [50] is an SNR-based detection method and a commonly used adaptive matched filter. It utilizes data sphered (DS) whitening of the spatial data to eliminate the first and second-order statistics in the original data space. It requires the target signal t, all data samples (r), covariance matrix K, and the background mean μb, which are described by Gaussian distributions. The formula for the NAMD detector is as follows:

$$\delta^{\mathrm{NAMD}}(\mathrm{r}) = \frac{(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})^{\mathrm{T}}\,\mathbf{K}^{-1}\,(\mathbf{r}-\boldsymbol{\mu}_{\mathrm{b}})}{(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})^{\mathrm{T}}\,\mathbf{K}^{-1}\,(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})} \tag{3}$$

When the entire formula is squared, it becomes the NAMD^2 detector:

$$\left(\delta^{\mathrm{NAMD}}(\mathrm{r})\right)^2 = \left[\frac{(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})^{\mathrm{T}}\,\mathbf{K}^{-1}\,(\mathbf{r}-\boldsymbol{\mu}_{\mathrm{b}})}{(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})^{\mathrm{T}}\,\mathbf{K}^{-1}\,(\mathbf{t}-\boldsymbol{\mu}_{\mathrm{b}})}\right]^2 \tag{4}$$

III. Hyperspectral Target Detection-Based 2D-3D Parallel Convolutional Neural Networks

A. Network Structure of the Proposed Framework

In this paper, we proposed a novel network architecture called HTD-2D-3D-PCNN, which is depicted in Figures 1 and 2. The network comprises two main convolutional channels: 2D-CNN and 3D-CNN. The upper channel first processes the original data using the HTD algorithm, which enhances the spectral features and then extracts rich spatial features through the 2D-CNN. This combination effectively integrates spectral and spatial information for improved classification performance.

On the other hand, the lower channel handles the original data after dimensionality reduction through PCA, and utilizes the 3D-CNN to simultaneously extract spectral and spatial features. Pooling layers are introduced between the convolutional layers to maintain feature invariance and reduce the parameter count, ensuring robustness in feature extraction even after feature map scaling. Additionally, to enhance feature representation and retain excellent classification performance, we added a deconvolutional (upsampling) layer after the 2D-CNN. This up-ampling operation helped improve feature representation To address overfitting, we implemented a GAP layer and applied Dropout method in the three fully connected layers (FC1, FC2, FC3). The GAP layer calculated the average of each feature map, reducing the risk of overfitting. Dropout was applied in the fully connected layers to prevent the model from becoming overly reliant on specific features during training, thus enhancing generalization. Further details about the CNN architecture are elaborated in subsequent chapters, thereby providing a comprehensive understanding of the proposed HTD-2D-3D-PCNN model.

B. Data processing of HTD

The previous section introduced the hyperspectral target detection (HTD) algorithm used in our experiments. This paragraph provides a detailed description of how the original data is transformed via HTD before being input into 2D-CNN shown in Fig 2. The HTD process served multiple purposes, including image enhancement, background suppression, and data dimensionality reduction, resulting in improved efficiency during model training due to decreased data volume.

The success of HTD relies on obtaining appropriate signatures, as the accuracy of the target information is crucial. Incorrect signatures can lead to false alarms and missed target detections. Therefore, obtaining the correct target signature is a critical step in the process. To achieve this, we utilized the optimal signature generation process (OSGP) [51], which facilitated obtaining optimal target feature **t**. OSGP involves an iterative process that enhances the accuracy of target selection, leading to effective target detection and stability, as confirmed by the experimental results. In this study, we combined OSGP with HTD to iteratively determine the best target feature **t**.

Figure 3 illustrates the flowchart of the OSGP process. Below is a step-by-step outline of the OSGP process, using Class 1 of HTD as an example:

Step 1) A random pixel is selected from Class 1 of HSI ground truth as the initial target feature **t**.

Step 2) HTD is performed on **t**, and Otsu's method [52] is applied for binary segmentatiuon.

Step 3) All pixels with a binary value of 1 in the binary segmentation result are averaged to obtain the new target feature t'.

Step 4) The spectral angle mapper (SAM) [53] between t' and **t** is computed. If the SAM value is smaller than preset condition α, the iteration stops. Otherwise, t' is used as the new target and iteratively input into the HTD detector. Steps 2 to 4 are repeated until the condition is met.

Step 5) Finally, t' is employed as the optimal signature for the HTD detector.

Once the optimal t' is obtained for HTD computation using Class 1 of the dataset as the target, an image of size 512 x 217

is generated. This image enhances the spectral features of Class 1, highlights their differences from the other 15 classes, and uses $\mathbf{R}^{-1}$ for background suppression. Similar results can be achieved when detecting other classes as targets. By generating 16 different HTD results using various t', the subsequent 2D-CNN can learn spatial information with enhanced spectral features of specific classes, achieving the fusion of spectral and spatial information. After detecting all 16 classes as targets, the 16 images are stacked together to form a 512 x 217 x 16 image, which serves as the input to the 2D-CNN in our proposed model.

C. 2D-CNN

In the proposed model, 2D-CNN is utilized to extract spatial feature information from the data. Convolutional layers with activation functions perform non-linear transformations to obtain feature maps, while deconvolutional layers expand the size of the feature maps, acting as the inverse operation of the convolution. Pooling layers are employed to reduce the size of feature maps while retaining important features, resulting in reduced model parameters and improved computational efficiency. Global average pooling (GAP) is used to calculate the average of each feature map and reduce the risk of overfitting.

To enhance the desired target class in the image based on spectral features, the data is processed through the HTD algorithm. Subsequently, 2D-CNN is used to learn spatial features from the image. This approach demonstrates improved classification performance compared to simultaneously extracting spectral and spatial features using 3D-CNN. The experimental results indicated that the former yielded a slightly higher overall accuracy compared to the latter, by approximately 0.01% to 1.21%. Therefore, in this study, we employed the HTD algorithm to detect targets on the data and then used 2D-CNN to learn spatial features.

For example, in the case of the Indian Pines dataset with an original data size of 145 x 145 x 200, after using the CEM detector, we obtained a size of 145 x 145 x 16, as this dataset contained 16 classes. The objective was to predict the label of each pixel in the image. To achieve this, we generated cubes of size L x W x B as the input to the model, where L represents the length, W represents the width, and B represents the number of bands, and B represents the number of bands in each cube.

The detailed architecture of the 2D-CNN used in this experiment was as follows: the 25 x 25 x 16 cube was input into the 2D-CNN, passing through two consecutive convolutional layers with a kernel size of 3 x 3 and a stride of 1 x 1. Next, it entered a max pooling layer, followed by a convolutional layer with a kernel size of 5 x 5 and a deconvolutional layer with a kernel size of 5 x 5. Finally, it underwent global average pooling. This architecture allowed the 2D-CNN to effectively learn spatial features from the data and contributed to the improved classification performance.

D. Principle component analysis processing of HSI data

Due to the high dimensionality of HSI, which may contain hundreds of spectral bands, PCA was used to identify the main elements in the data. The objective was to maximize the variance of the data after projection, thereby removing redundant information and achieving dimensionality reduction while preserving the original characteristics of the data. In this experiment, PCA was applied to reduce the dimensionality of the original data, which was then input into the 3D-CNN to simultaneously extract spectral and spatial information. The experimental results demonstrated that the best classification performance was achieved when the data dimensionality was reduced to the number of classes present in the dataset. For instance, in the case of the Indian Pines dataset, which contained a total of 16 classes, reducing the original data size from 145 x 145 x 200 to 145 x 145 x 16 yielded the optimal overall accuracy.

E. 3D-CNN

The main difference between 3D-CNN and 2D-CNN lies in the ability of 3D-CNN to effectively extract both spectral and spatial information simultaneously, leading to improved classification performance compared to 2D-CNN. While 2D-CNN focuses solely on spatial features, 3D-CNN considers both spectral and spatial characteristics, making it more suitable for hyperspectral image classification tasks.

When data is dimensionally reduced through PCA, PCA method does not consider the inherent spectral and spatial features of the data. The primary objective of PCA is to maximize the variance of the data after projection, without considering the specific information contained in different spectral bands. However, when the reduced data is then processed through 3D-CNN, that can effectively extract both spectral and spatial features together, leading to better classification performance. The experimental results showed an overall accuracy increase of approximately 3.83% compared to using 2D-CNN, which only learns spatial features. In this study, the dimensionality of the original data was first reduced using PCA, and then training was performed using 3D-CNN. The detailed architecture of the 3D-CNN in this experiment was as follows: the data, of size 25 x 25 x 16, entered a convolutional layer with a kernel size of 3 x 3 x 7, followed by another convolutional layer with a kernel size of 3 x 3 x 5. Subsequently, a max-pooling was is applied. After that, the data entered a convolutional layer with a kernel size of 3 x 3 x 3 and then went through GAP before being passed to the fully connected layers for classification. This architecture allowed the 3D-CNN to effectively capture both spectral and spatial information from the reduced data.

F. Concatenated 2D-3D-CNN

After applying the HTD algorithm, the association between feature maps and inter-layer information was significantly reduced during the 2D-CNN convolution. Therefore, no substantial benefits in feature learning would be found after

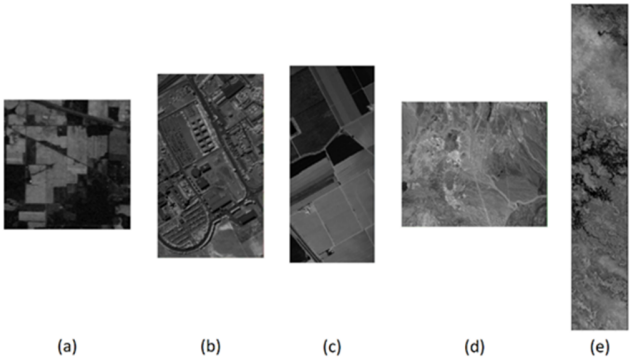

Fig. 4. Grayscale images of (a) Indian Pines; (b) University of Pavia; (c) Salinas Scene; (d) Kennedy Space Center; (e) Botswana.

concatenating the results of 2D-CNN and PCA+3D-CNN in a sequential manner for further convolution using 3D-CNN.

Instead, adopting a parallel architecture that combined the results from 2D-CNN and PCA+3D-CNN and passed them through subsequent fully connected layers for classification led to better overall accuracy compared to the sequential approach. According to the experimental results, this parallel approach yielded an overall accuracy improvement ranging from approximately 0.01% to 29.65%. In this study, the 2D-CNN method and the 3D-CNN method both processed through GAP to obtain the feature maps, were merged and connected. They were then sequentially passed through three fully connected layers with neurons number of 1024, 512, and 128, respectively. A dropout rate of 0.25 was added between each fully connected layer to prevent overfitting. Finally, the softmax activation function was applied for classification.

IV. Experiments

A. Data set

This study validated the proposed model's performance using five publicly available HSI datasets. The grayscale images for each dataset are displayed in Figures 4.

1) The first dataset was the Purdue Indian Pines Scene (IP), captured at the Indiana Pine Tree Test Site in northwestern Indiana. It had a data resolution of 145 x 145 and contained 200 spectral bands. The Purdue Indian Pines Scene consisted of 16 different plant categories.
2) The second dataset was the Pavia University (PU) dataset, acquired by the ROSIS-03 satellite sensor. This scene included 610 x 340 pixels, with a spatial resolution of 1.3 meters and 103 spectral bands.
3) The third dataset was the Salinas Scene (SA), collected by the AVIRIS sensor in the Salinas Valley of California, known for its high spatial resolution (3.7-meter pixels). The data size was 512 x 217, and after removing 20 water absorption bands, it contained 204 spectral bands, with 16 land cover categories.
4) The fourth dataset was the Kennedy Space Center (KSC) dataset, collected from the KSC in Florida. The data size was 512 x 614, and after removing water absorption bands, it had 176 spectral bands.
5) The final dataset was the Botswana (BOT) dataset, acquired by the Hyperion NASA EO-1 satellite over the Okavango Delta in Botswana between 2001 and 2004. After removing the uncalibrated and noisy bands covering water features, the data size was 1476 x 256, with 145 spectral bands.

B. Experimental setup

After applying HTD and PCA dimensionality reduction, the dimensions were reduced to the number of classes corresponding to each dataset (IP and SA had 16 bands, PU had nine bands, KSC had 13 bands, and BOT had 14 bands). To improve the model's performance, the window size for generating image cubes before inputting into the CNN was set to 19 x 19 for the PU dataset, 15 x 15 for the SA dataset, and 25 x 25 for the rest of the datasets. For the IP, PU, SA, and KSC datasets, the samples were randomly split into 10% as the training set, 5% as the validation set, and 85% as the test set. Effective data partitioning methods were employed to ensure there was no sample overlap between the training and testing datasets. The use of stratified sampling further ensured a consistent distribution in the training and test sets, resulting in the best possible global fit. As for the BOT dataset, it was split into 5% for training, 5% for validation, and 90% for testing. The number of epochs for the IP, SA, and KSC datasets was set to 300, while for the PU and BOT datasets, it was set to 100. The initial learning rate was set to 0.001, and the batch size was set to 100 for all datasets. The detailed number of training samples for the five datasets is shown in Tables I to V.

The hardware and software specifications used in this experiment were: a 12th Gen Intel® Core™ i9-12900KF CPU, an NVIDIA GeForce RTX 3090 Ti, 24.0 GB of RAM, and Python 3.9. The evaluation of the model's performance was done using the overall

accuracy, average accuracy, and kappa coefficient. The overall accuracy referred to the percentage of correctly classified pixels. The average accuracy referred to the average of overall accuracy measured for each class. The kappa coefficient was a statistical measure of the agreement between qualitative items.

The experimental results presented in this paper were based on the average of 10 runs. Each experiment was performed 10 times, and the obtained data were averaged to ensure the reliability and robustness of the results. This approach helped to reduce the impact of randomness or variability in the training process and provided more accurate and stable performance metrics for the proposed HTD-2D-3D-PCNN model using the five different datasets.

TABLE I
NUMBER OF IP DATASET SAMPLES

Class No.	Class Name	Total	Training	Validation	Testing
1	Alfalfa	46	5	3	38
2	Corn-notill	1428	143	72	1213
3	Corn-mintill	830	83	42	705
4	Corn	237	24	12	201
5	Grass-pasture	483	49	25	409
6	Grass-trees	730	73	37	620
7	Grass-pasture-mowed	28	3	2	23
8	Hay-windrowed	478	48	48	382
9	Oats	20	2	1	17
10	Soybean-notill	972	98	49	825
11	Soybean-mintill	2455	246	123	2086
12	Soybean-clean	593	60	30	503
13	Wheat	205	21	11	173
14	Woods	1265	127	64	1074
15	Buildings-Grass-Trees-Drives	386	39	20	327
16	Stone-Steel-Towers	93	10	5	78

TABLE II
NUMBER OF PU DATASET SAMPLES

Class No.	Class Name	Total	Training	Validation	Testing
1	Asphalt	6631	664	332	5635
2	Meadows	18649	1865	933	15851
3	Gravel	2099	210	105	1784
4	Trees	3064	307	154	2603
5	Painted metal sheets	1345	135	68	1142
6	Bare Soil	5029	503	252	4274
7	Bitumen	1330	133	67	1130
8	Self-Blocking Bricks	3682	369	185	3128
9	Shadows	947	95	48	804

TABLE III
NUMBER OF SA DATASET SAMPLES

Class No.	Class Name	Total	Training	Validation	Testing
1	Brocoli_green_weeds_1	2009	201	101	1797
2	Brocoli_green_weeds_2	3726	373	187	3166
3	Fallow	1976	198	99	1679
4	Fallow_rough_plow	1394	140	70	1184
5	Fallow_smooth	2678	268	134	2276
6	Stubble	3959	396	198	3365
7	Celery	3579	358	179	3042
8	Grapes_untrained	11271	1128	564	9579
9	Soil_vinyard_develop	6203	621	311	5271
10	Corn_senesced_green_weeds	3278	328	164	2786
11	Lettuce_romaine_4wk	1068	107	54	907
12	Lettuce_romaine_5wk	1927	193	97	1637
13	Lettuce_romaine_6wk	916	92	46	778
14	Lettuce_romaine_7wk	1070	107	54	909
15	Vinyard_untrained	7268	727	364	6177
16	Vinyard_vertical_trellis	1807	181	91	1535

TABLE IV
NUMBER OF KSC DATASET SAMPLES

Class No.	Class Name	Total	Training	Validation	Testing
1	Scrub	761	77	39	645
2	Willow swamp	243	25	13	205
3	Cabbage palm hammock	256	26	13	217
4	Cabbage palm/oak hammock	252	26	13	213
5	Slash pine	161	17	8	136
6	Oak/Broadleaf hammock	229	23	12	194
7	Harwood swamp	105	11	6	88
8	Graminoid marsh	431	44	22	365
9	Spartina marsh	520	52	26	442
10	Cattail marsh	404	41	21	342
11	Salt marsh	419	42	21	356
12	Mud flats	503	51	26	426
13	Water	927	93	47	787

TABLE V
NUMBER OF BOT DATASET SAMPLES

Class No.	Class Name	Total	Training	Validation	Testing
1	Water	270	14	14	242
2	Hippo grass	101	6	6	89
3	Floodplain Grasses 1	251	13	13	225
4	Floodplain Grasses 2	215	11	11	193
5	Reeds 1	269	14	14	229
6	Riparian	269	14	14	241
7	Firescar 2	259	13	13	233
8	Island interior	203	11	11	181
9	Acacia woodlands	314	16	16	282
10	Acacia shrublands	248	13	13	222
11	Acacia grasslands	305	16	16	273
12	Short mopane	181	10	10	161
13	Mixed mopane	268	14	14	240
14	Exposed soils	95	5	5	85

V. RESULTS AND ANALYSIS

In our proposed model, the input data for 2D-CNN was obtained after applying the HTD algorithm, which enhanced the spectral features of the target objects and reduced data while suppressing background noise. For 3D-CNN, the input data was obtained by reducing the original data to the number of classes corresponding to each dataset using PCA. After separate processing in 2D-CNN and 3D-CNN, the feature maps were combined and further processed through convolutional operations before being fed into fully connected layers for classification.

For the experimental setup, we randomly split the samples of the IP, PU, SA, and KSC datasets into 10% for training, 5% for validation, and 85% for testing. As for the BOT dataset, we used a random split of 5% for training, 5% for validation, and 90% for testing. To prevent overfitting, we applied a dropout rate of 25% between each fully connected layer. The number of epochs was set to 300 for the IP, SA, and KSC datasets, while for the PU and BOT datasets, it was set to 100. Additionally, we used a batch size of 100 for all datasets. These parameters were carefully chosen to achieve the best balance between training efficiency and model performance for each dataset. The experimental results presented in this section are averages derived from 5 individual results, and standard deviations are included to provide insights into the stability of the obtained results. It's crucial to note that some references listed in our tables did not provide the standard deviation. As a result, we have selectively included the standard deviation only for our proposed models where applicable, as these values are implementable and contribute to a more comprehensive understanding of the experimental outcomes.

A. Results of the Indian Pines dataset

Figure 5 shows the classification results of different CNN model architectures on the IP dataset. Table 6 presents the performance data of our proposed HTD-2D-3D-PCNN model on the IP dataset, along with a comparison to recent methods, including SVM, SSRN [34], CNNDH [28], CACNN [42], 2D-3D-D [40], and FADCNN [43]. If the KAPPA coefficient result was not provided in the literature, it was marked as blank in the table.

Based on the data presented in the table, it is clear that our HTD-2D-3D-PCNN model surpasses other recent methods in terms of overall accuracy, average accuracy, and the KAPPA coefficient. We achieved scores of 98.41%, 98.23%, and 98.17%, respectively. Notably, our model excels in achieving exceptionally high accuracy in Classes 8, 9, 13, and 15, where accuracy reaches 100%. Furthermore, for Classes 10, 12, and 13, our model outperforms all recent methods in terms of accuracy. The second-highest overall accuracy was recorded by the 2D-3D-D model, at 98.33%.

This superiority is particularly pronounced in classes with limited sample sizes, such as Classes 10, 12, and 13. These classes typically possess restricted spatial information and heavily rely on spectral data. The HTD-2D-3D-PCNN model leverages the strengths of both 2D-CNN and 3D-CNN. Specifically, the 2D-CNN component, in conjunction with the HTD method, enhances spectral features via HTD and learns to extract spatial characteristics. Conversely, the 3D-CNN excels at capturing spatial-spectral features, harnessing the advantages of both networks to enhance accuracy across various classes. Our proposed approach consistently achieves higher accuracy compared to alternative methods.

B. Results of the University of Pavia dataset

Figure 6 shows the classification results of different CNN model architectures on the PU dataset. Table 7 presents the performance data of the proposed HTD-2D-3D-PCNN model on the PU dataset, along with the comparison results of recent methods, including SVM, SSRN [34], AfNet [35], CACNN [42], FFDN [44], and HybridSN [36].

Based on the data presented in the table, our proposed HTD-2D-3D-PCNN model consistently outperforms other recent methods in terms of overall accuracy, average accuracy, and the KAPPA coefficient, achieving scores of 99.85%, 99.73%, and 99.80%, respectively. Notably, our model attains a perfect accuracy of 100% in Classes 3 and 7, and it surpasses all recent methods in accuracy for Classes 2 and 4. The second-highest overall accuracy is observed in the HybridSN [36] model, at 99.69%.

In classes 3 and 4, where the presence of Class 2 surrounding their edges makes them particularly susceptible to

TABLE VI
CLASSIFICATION RESULTS OF THE IP DATASET WITH THE COMPARED METHODS

Class P_{OA}%	SVM	2D-CNN	3D-CNN	SSRN [34]	CNNDH [28]	CACNN [42]	2D-3D-D [40]	FADCNN [43]	**HTD-2D-3D-PCNN**
1	84.78±5.11	**100.0**±0.00	**100.0**±0.00	95.81	60.43	77.78	**100.0**±0.00	88.26±7.73	97.43±0.34
2	70.38±10.37	99.30±0.45	91.67±3.27	**99.58**	92.06	92.31	98.36±0.62	98.04±0.27	96.95±0.59
3	62.05±12.31	96.38±2.81	93.97±4.58	**99.61**	88.05	98.79	97.80±0.76	97.04±1.50	98.72±1.86
4	89.03±2.04	**100.0**±0.00	98.12±0.57	99.91	95.69	**100.0**	97.20±1.86	96.03±3.21	92.53±2.18
5	90.89±3.10	94.84±3.29	95.41±2.64	**100.0**	95.65	94.84	99.30±0.50	99.34±0.44	94.16±2.22
6	95.48±2.26	99.31±0.36	96.19±0.98	96.85	98.82	**100.0**	99.07±0.41	99.48±0.48	97.58±0.54
7	96.43±0.71	**100.0**±0.00	76.0±11.67	93.53	87.86	80.00	**100.0**±0.00	75.72±20.50	87.50±5.63
8	96.86±0.42	**100.0**±0.00	**100.0**±0.00	99.08	**100.0**	**100.0**	99.83±0.18	**100.0**±0.00	**100.0**±0.00
9	85.00±3.39	**100.0**±0.00	66.66±16.74	99.78	93.00	**100.0**	92.72±3.73	78.00±25.42	**100.0**±0.00
10	72.94±11.97	97.42±1.72	89.37±5.38	91.52	94.98	97.93	97.34±0.84	96.50±0.89	**98.18**±0.19
11	65.42±9.08	97.75±2.11	**99.09**±0.48	98.36	96.67	98.98	98.23±0.53	98.49±0.69	98.08±1.50
12	67.79±6.71	90.75±6.93	91.94±4.96	96.29	92.62	94.95	97.66±0.93	95.45±1.77	**98.80**±1.01
13	97.56±0.72	97.56±1.28	90.81±3.75	99.02	99.41	90.24	99.32±0.65	99.02±1.02	**100.0**±0.00
14	89.96±4.62	**100.0**±0.00	99.47±0.23	96.90	99.41	**100.0**	99.01±0.47	99.48±0.66	99.62±0.21
15	71.50±13.26	**100.0**±0.00	99.71±0.16	95.65	94.04	**100.0**	98.60±0.86	99.74±0.16	**100.0**±0.00
16	**100.0**±0.00	94.73±3.69	91.66±5.28	93.95	98.92	**100.0**	92.59±3.08	82.80±9.40	94.93±4.41
OA	73.39±2.44	97.90±0.32	97.02±0.29	96.88±0.26	95.29	97.51	98.33±0.21	97.95±0.15	**98.41±0.17**
AA	83.50±2.38	98.01±1.41	94.07±3.79	97.24±0.59	92.98	95.37	97.94±0.96	93.96±4.63	**98.23±1.29**
KAPPA	68.82±6.65	97.61±1.82	96.61±1.37	96.26±0.30		97.16			**98.17±0.94**

misclassification, our proposed method demonstrates superior accuracy compared to alternative approaches. This is attributed to our utilization of the HTD method, which enhances the visibility of targets while suppressing background noise. Consequently, it highlights features that are relatively small and have blurred edges in the image. These enhanced features serve as input for 2D-CNN, enabling better spatial learning and, consequently, an enhancement in overall accuracy.

C. Results of the Salinas Scene dataset

Figure 7 shows the classification results of different CNN model architectures on the SA dataset. Table 8 presents the performance data of the proposed HTD-2D-3D-PCNN model on the SA dataset, along with the comparison results of recent methods, including SVM, SSRN [34], AfNet [35], CACNN [42], FFDN [44], and pResNet [27].

Based on the data presented in the table, our proposed HTD-2D-3D-PCNN model consistently outperforms other recent methods in terms of overall accuracy, average accuracy, and the KAPPA coefficient, achieving scores of 99.92%, 99.86%, and 99.91%, respectively. Notably, our model attains a perfect accuracy of 100% in multiple classes, specifically in Classes 2, 3, 9, 10, 11, 13, 14, 15, and 16. The second-highest overall accuracy is observed in the 3D-CNN model, at 99.78%. In classes 8-15, our method significantly benefits from the use of the HTD method to enhance spectral information and the subsequent application of 2D-CNN for spatial feature enhancement. This approach effectively highlights the edges of these classes and distinguishes fine details from areas that are similar to other classes. As a result, it excels in handling subtle nuances. While other methods may struggle with edge cases and intricate details, our proposed method achieves an accuracy rate of 100%.

Furthermore, our method incorporates Global Average Pooling (GAP) and Dropout in the fully connected layers, effectively preventing overfitting. These strategies, coupled with the features mentioned above, contribute to the successful improvement in accuracy.

D. Results of the Kennedy Space Center dataset

Figure 8 displays the classification results of different CNN model architectures on the KSC dataset. Table 9 presents the performance data of the proposed HTD-2D-3D-PCNN model on the KSC dataset , along with the comparison results of recent methods, including SVM, CNNDH [28], FADCNN [43], and 2D-3D-D [40]. Blank cells in the table indicate that the corresponding literature did not provide KAPPA coefficient results.

Based on the data in the table, our proposed HTD-2D-3D-PCNN model consistently outperforms other recent methods in terms of overall accuracy, average accuracy, and the KAPPA coefficient, achieving scores of 99.82%, 99.75%, and 99.79%, respectively. Remarkably, our model attains a perfect

TABLE VII
CLASSIFICATION RESULTS OF THE PU DATASET WITH THE COMPARED METHODS

Class P_{OA}%	SVM	2D-CNN	3D-CNN	SSRN [34]	AfNet [35]	CACNN [42]	FFDN [44]	HybridSN [36]	**HTD-2D-3D-PCNN**
1	81.87±2.73	99.44±0.28	99.91±0.03	98.45	99.49	99.09	98.24	**99.96**	99.81±0.16
2	81.81±2.49	99.81±0.07	99.94±0.02	97.90	99.97	99.16	98.90	99.93	**99.99**±0.01
3	82.99±1.66	98.83±1.03	98.88±1.07	96.99	**100.0**	99.19	98.07	99.43	**100.0**±0.00
4	95.48±1.32	98.98±0.84	95.95±2.39	98.36	97.59	98.76	97.82	98.54	**99.56**±0.37
5	99.59±0.05	98.92±1.50	98.01±1.48	99.91	95.63	99.90	**99.93**	99.38	99.17±0.44
6	86.73±1.94	99.42±0.24	**100.0**±0.00	97.48	**100.0**	99.26	99.46	**100.0**	99.97±0.02
7	93.76±0.88	99.41±0.19	**100.0**±0.00	**100.0**	**100.0**	99.77	99.79	**100.0**	**100.0**±0.00
8	82.96±1.92	99.21±0.23	99.51±0.21	99.27	**99.93**	99.03	98.52	99.77	99.48±0.23
9	99.93±0.02	**100.0**±0.00	97.18±1.34	99.55	92.48	99.77	99.69	95.40	99.88±0.08
OA	84.66±1.12	99.53±0.09	99.42±0.11	98.19±0.09	99.43	99.17	98.78	99.69	**99.85**±0.07
AA	89.46±1.45	99.16±0.48	98.72±0.72	98.02±0.17	98.35	99.33	98.93	99.16	**99.73**±0.15
KAPPA	80.03±2.03	99.38±0.17	99.23±0.25	97.57±0.12	99.24	98.89	98.36	99.59	**99.80**±0.12

TABLE VIII
CLASSIFICATION RESULTS OF THE SA DATASET WITH THE COMPARED METHODS

Class P_{OA}%	SVM	2D-CNN	3D-CNN	SSRN [34]	AfNet [35]	CACNN [42]	FFDN [44]	pResNet [27]	**HTD-2D-3D-PCNN**
1	97.84±1.08	99.94±0.02	**100.0**±0.00	99.95	**100.0**	**100.0**	99.96	99.98	99.83±0.11
2	94.38±2.41	**100.0**±0.00	**100.0**±0.00	**100.0**	99.97	99.93	99.96	**100.0**	**100.0**±0.00
3	86.71±3.37	**100.0**±0.00	**100.0**±0.00	99.94	99.25	99.79	99.61	99.99	**100.0**±0.00
4	99.41±0.15	99.91±0.02	98.40±0.97	99.91	98.11	99.79	99.88	**99.96**	99.84±0.06
5	92.98±2.20	99.64±0.14	99.50±0.23	99.57	99.56	**99.83**	99.80	98.81	99.75±0.14
6	99.33±0.26	99.97±0.01	99.57±0.16	**100.0**	99.86	99.96	99.77	**100.0**	99.97±0.01
7	98.93±0.83	99.73±0.17	99.78±0.13	**100.0**	**100.0**	99.89	99.80	99.96	99.78±0.19
8	55.30±9.58	99.92±0.03	99.98±0.01	87.13	**100.0**	95.69	93.77	91.47	99.99±0.00
9	94.29±3.60	**100.0**±0.00	**100.0**±0.00	**100.0**	**100.0**	99.76	99.75	99.99	**100.0**±0.00
10	85.61±5.76	**100.0**±0.00	**100.0**±0.00	99.39	99.51	99.26	99.43	99.64	**100.0**±0.00
11	89.98±4.90	98.78±1.04	94.79±2.66	99.99	94.87	99.86	99.98	**100.0**	**100.0**±0.00
12	99.77±0.06	99.81±0.05	**100.0**±0.00	**100.0**	92.79	99.96	99.95	**100.0**	99.36±0.27
13	97.91±0.75	99.87±0.08	99.51±0.25	**100.0**	97.70	99.92	99.85	**100.0**	**100.0**±0.00
14	91.03±3.09	96.69±1.18	99.48±0.24	99.98	99.21	99.89	99.90	**100.0**	**100.0**±0.00
15	71.02±9.11	99.82±0.09	99.77±0.11	94.06	98.69	97.20	96.63	93.43	**100.0**±0.00
16	91.66±2.86	**100.0**±0.00	**100.0**±0.00	99.99	**100.0**	99.75	99.86	99.97	**100.0**±0.00
OA	82.69±2.55	99.73±0.08	99.78±0.06	96.31	99.27	98.55	98.04	97.15	**99.92**±0.02
AA	90.39±3.12	99.46±0.17	99.61±0.29	98.44	98.72	99.41	99.24	98.95	**99.86**±0.04
KAPPA	80.75±3.64	99.69±0.14	99.76±0.16	95.88	99.19	98.38	97.81	96.81	**99.91**±0.05

100% accuracy in more than half of the classes, specifically Classes 1, 2, 4, 7, 8, 10, 12, and 13, with Classes 4 and 8 surpassing all other recent methods in accuracy. The second-highest overall accuracy is observed in the 2D-CNN model, at 99.34%.In this dataset, many classes have limited sample sizes, with Class 4 posing particular challenges due to its small target size and limited spatial features. To address these challenges, our method employs the HTD technique to enhance spectral features, followed by spatial feature extraction using 2D-CNN. Additionally, we utilize PCA to reduce the dimensionality of the original images and employ them as input for 3D-CNN to extract spatial-spectral features. This strategic fusion of methods significantly enhances the Overall Accuracy (OA).

E. Results of the Botswana dataset

Figure 9 displays the classification results of different CNN model architectures on the BOT dataset. Table 10 presents the

TABLE IX
CLASSIFICATION RESULTS OF THE KSC DATASET WITH THE COMPARED METHODS

Class P_{OA}%	SVM	2D-CNN	3D-CNN	CNNDH [28]	FADCNN [43]	2D-3D-D [40]	**HTD-2D-3D-PCNN**
1	92.16±3.47	**100.0**±0.00	97.37±1.68	98.13	99.97	98.19	**100.0**±0.00
2	86.16±5.79	**100.0**±0.00	84.01±10.27	72.84	98.68	96.75	**100.0**±0.00
3	42.55±9.13	**100.0**±0.00	87.82±5.72	95.39	96.72	98.31	99.56±0.24
4	67.69±10.35	99.55±0.21	73.12±14.81	65.95	92.86	87.50	**100.0**±0.00
5	0.00	**100.0**±0.00	88.27±6.47	82.24	97.14	91.40	97.93±1.17
6	54.71±10.82	**99.51**±0.38	96.11±2.09	86.20	97.73	98.73	98.05±0.49
7	0.00	**100.0**±0.00	94.68±3.44	92.96	99.24	95.87	**100.0**±0.00
8	65.12±9.29	96.64±2.13	93.29±4.15	87.19	98.98	96.32	**100.0**±0.00
9	67.82±8.83	**99.78**±0.12	92.52±4.37	92.08	99.77	97.34	98.51±0.72
10	93.40±2.56	**100.0**±0.00	99.45±0.35	86.29	**100.0**	97.35	**100.0**±0.00
11	**100.0**±0.00	**100.0**±0.00	98.14±1.15	99.76	99.86	99.48	99.73±0.20
12	83.75±6.22	98.89±0.82	**100.0**±0.00	94.83	**100.0**	98.90	**100.0**±0.00
13	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00	**100.0**	**100.0**	99.70	**100.0**±0.00
OA	80.29±3.77	99.34±0.18	94.29±2.88	91.86	99.10	97.47	**99.82**±0.13
AA	65.64±5.11	99.14±0.28	92.19±4.19	88.76	98.53	96.81	**99.75**±0.21
KAPPA	77.98±3.40	99.27±0.23	93.64±3.58				**99.79**±0.11

TABLE X
CLASSIFICATION RESULTS OF THE BOT DATASET WITH THE COMPARED METHODS

Class P_{OA}%	SVM	2D-CNN	3D-CNN	SSRN [34]	AfNet [35]	HybridSN [36]	**HTD-2D-3D-PCNN**
1	92.57±2.03	98.04±0.73	97.65±1.07	97.26	**100.0**	**100.0**	**100.0**±0.00
2	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00	**100.0**	**100.0**	**100.0**	**100.0**±0.00
3	70.17±11.2	68.48±13.64	70.58±11.05	73.17	71.00	70.17	**99.15**±0.30
4	84.80±9.25	83.82±10.30	**100.0**±0.00	**100.0**	**100.0**	**100.0**	**100.0**±0.00
5	34.76±9.97	68.35±15.41	80.46±8.70	87.89	**97.26**	86.32	92.96±3.85
6	64.06±13.7	55.85±6.77	83.98±8.86	**94.92**	82.81	89.84	84.37±10.57
7	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00	**100.0**	**100.0**	**100.0**	**100.0**±0.00
8	81.34±6.24	99.48±0.12	**100.0**±0.00	**100.0**	**100.0**	97.92	**100.0**±0.00
9	82.88±6.01	82.88±8.83	**100.0**±0.00	98.99	97.65	99.66	98.33±0.61
10	**100.0**±0.00	99.15±0.46	**100.0**±0.00	**100.0**	**100.0**	**100.0**	99.15±0.49
11	91.03±3.30	83.79±10.72	**100.0**±0.00	**100.0**	**100.0**	89.65	**100.0**±0.00
12	80.23±7.06	97.67±1.68	**100.0**±0.00	**100.0**	95.34	**100.0**	**100.0**±0.00
13	99.21±0.29	99.21±0.31	**100.0**±0.00	**100.0**	**100.0**	**100.0**	**100.0**±0.00
14	71.11±9.82	86.66±7.04	**100.0**±0.00	95.83	98.88	**100.0**	98.89±0.88
OA	82.02±3.95	86.20±5.09	94.59±2.67	95.92	95.59	94.59	**97.58**±1.34
AA	82.30±5.64	87.39±5.42	95.19±2.12	96.29	95.93	95.26	**97.51**±1.19
KAPPA	80.51±4.38	85.05±6.78	94.14±3.03	95.57	95.23	94.14	**97.54**±0.99

performance data of the proposed HTD-2D-3D-PCNN model on the BOT data set , along with the comparison results of recent methods, including SVM, SSRN [34], AfNet [35], and HybridSN [36].

According to the data in the table, our proposed HTD-2D-3D-PCNN model achieved the highest overall accuracy, average accuracy, and KAPPA coefficient compared to other recent methods, with scores of 97.58%, 97.51%, and 97.54%, respectively. Remarkably, our model achieved 100% accuracy in more than half of the classes, specifically in Classes 1, 2, 4, 7, 8, 11, 12, and 13, with Class 3 exhibiting significantly higher accuracy compared to all other recent methods. The second-highest overall accuracy was achieved by the SSRN [34] model, at 95.92%. Especially in Class 3, where the targets

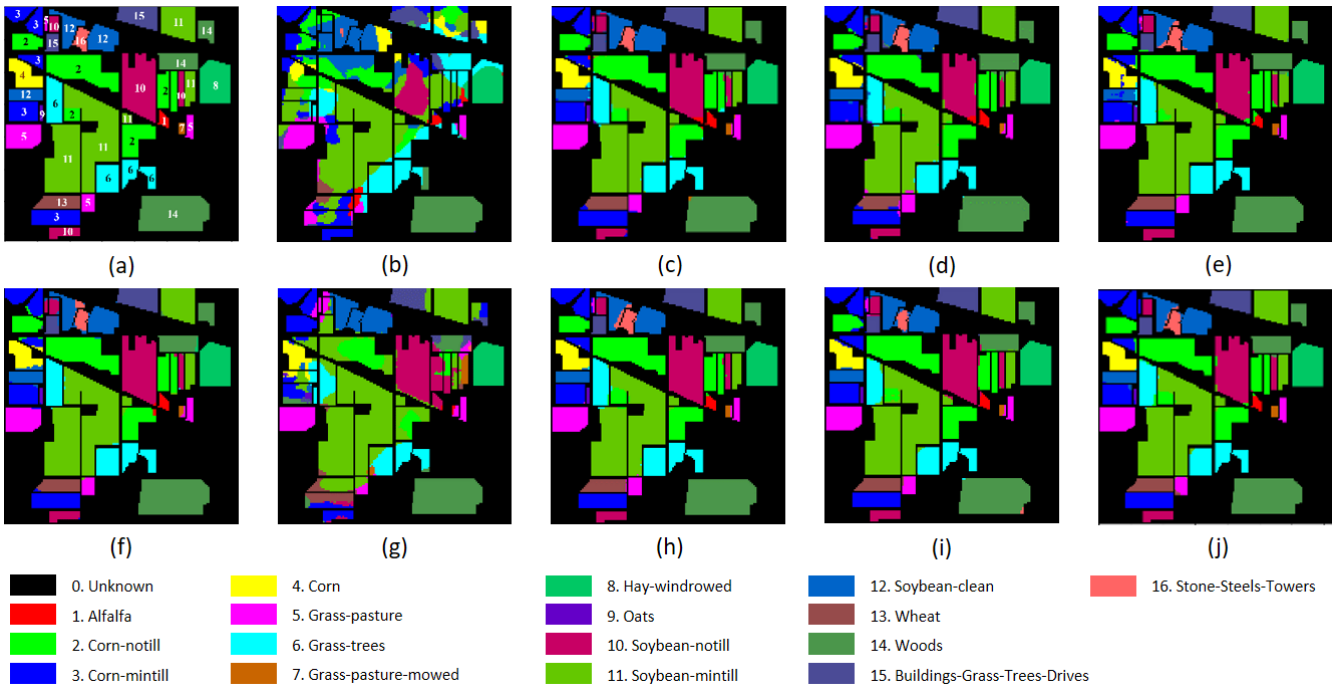

Fig. 5. Land cover categories and classification results of the IP dataset with the compared methods: (a) ground truth; (b) SVM; (c) 2D-CNN; (d) 3D-CNN; (e) SSRN; (f) CNNDH; (g) CACNN; (h) 2D-3D-D; (i) FADCNN; (j) HTD-2D-3D-PCNN.

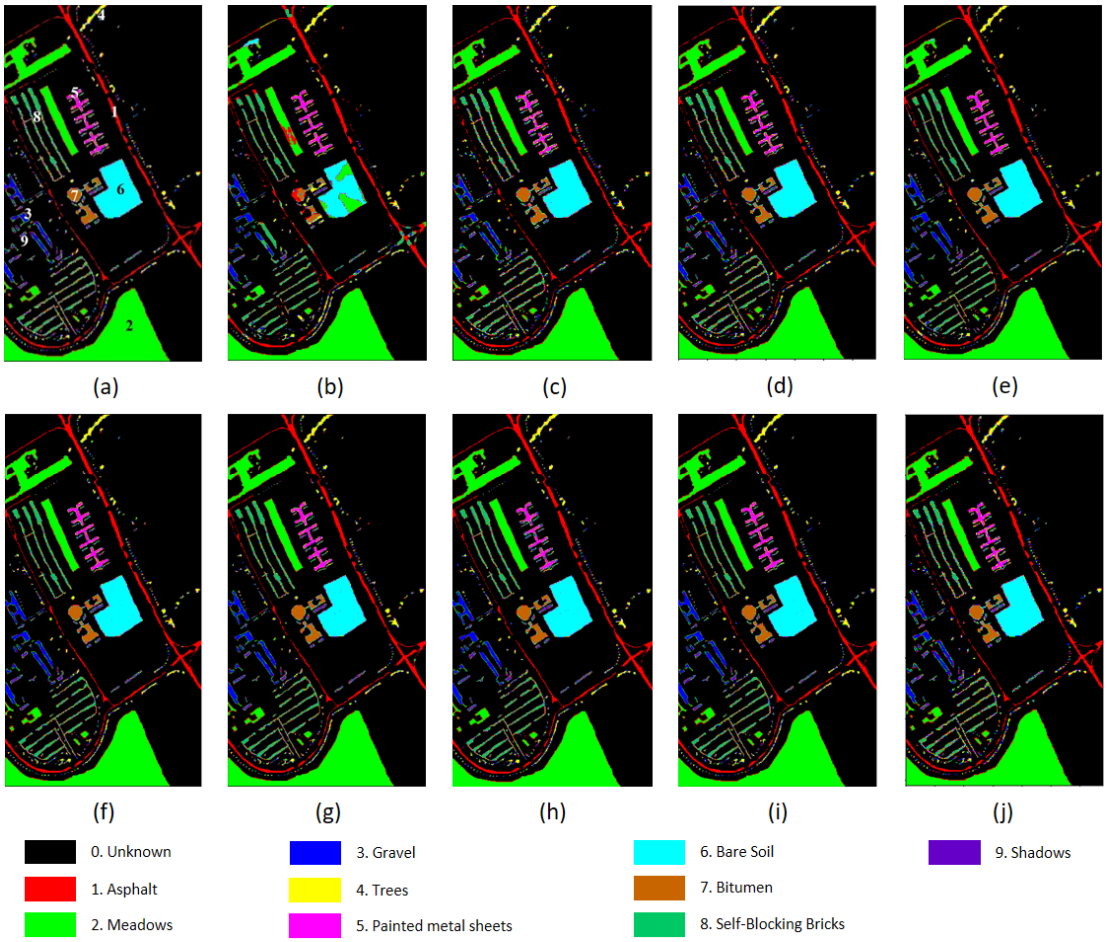

Fig. 6. Land cover categories and classification results of the PU dataset with the compared methods: (a) ground truth; (b) SVM; (c) 2D-CNN; (d) 3D-CNN; (e) SSRN; (f) AfNet; (g) CACNN; (h) FFDN; (i) HybridSN; (j) HTD-2D-3D-PCNN.

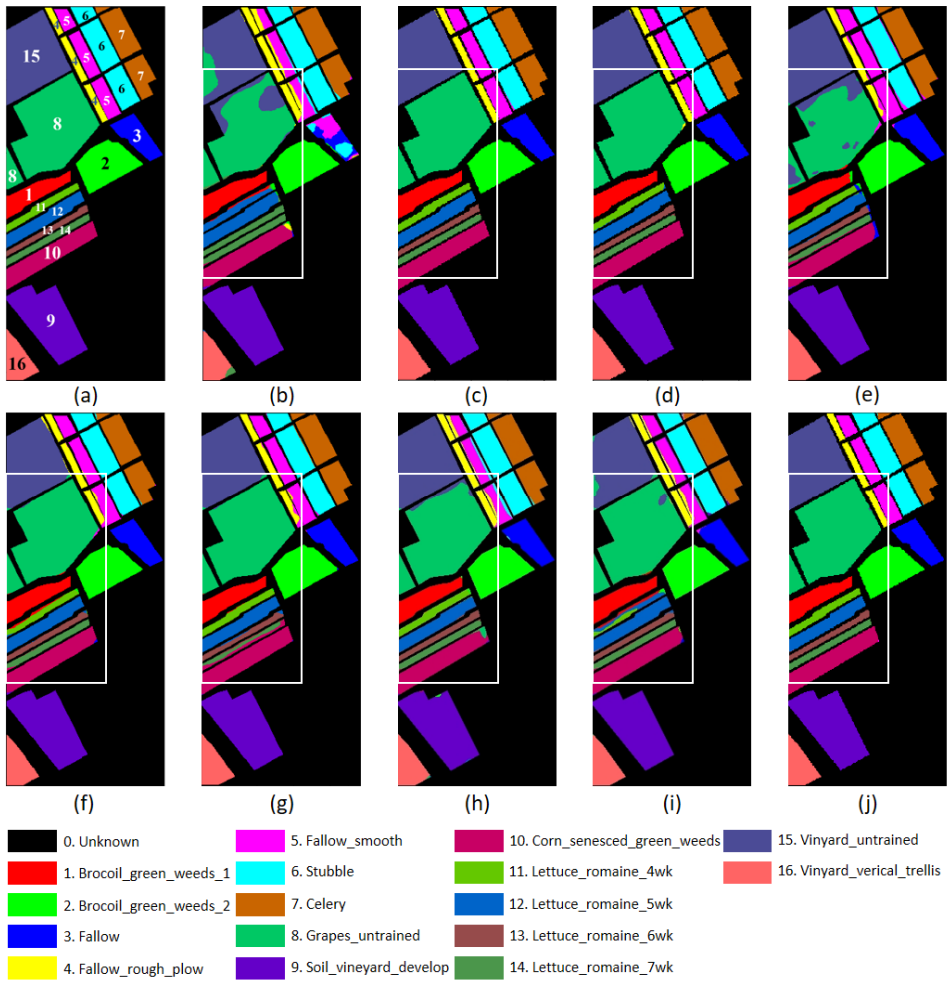

Fig. 7. Land cover categories and classification results of the SA dataset with the compared methods: (a) ground truth; (b) SVM; (c) 2D-CNN; (d) 3D-CNN; (e) SSRN; (f) AfNet; (g) CACNN; (h) FFDN; (i) pResNet; (j) HTD-2D-3D-PCNN.

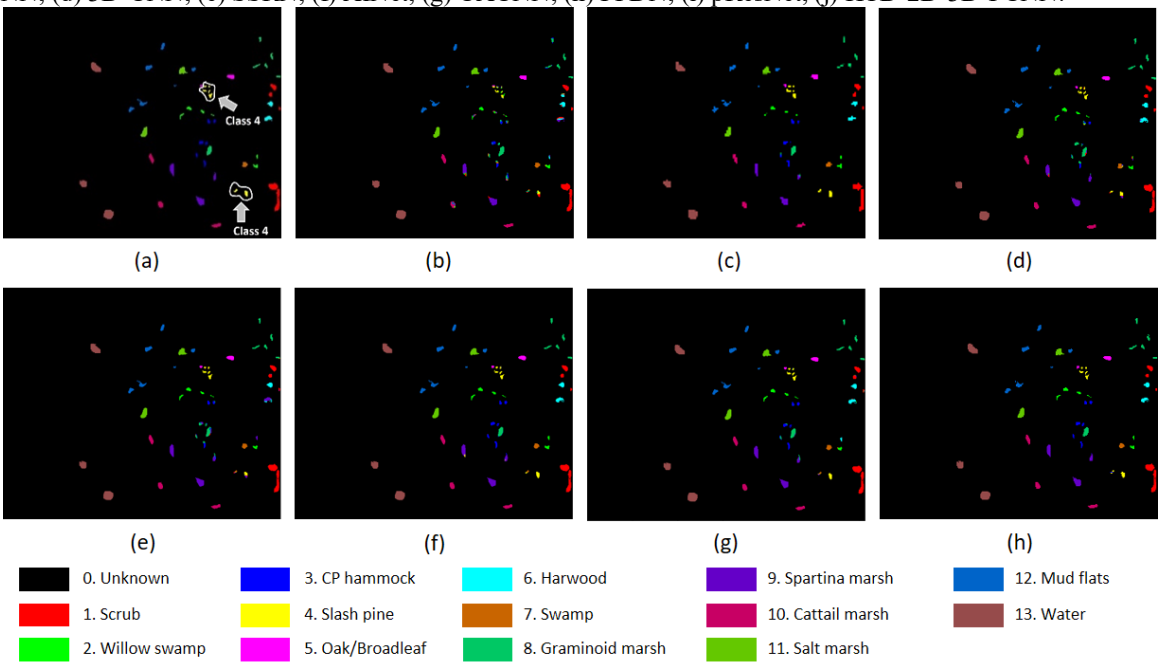

Fig. 8. Land cover categories and classification results of the KSC dataset with the compared methods: (a) ground truth; (b) SVM; (c) 2D-CNN; (d) 3D-CNN; (e) CNNDH; (f) FADCNN; (g) 2D-3D-D; (h) HTD-2D-3D-PCNN.

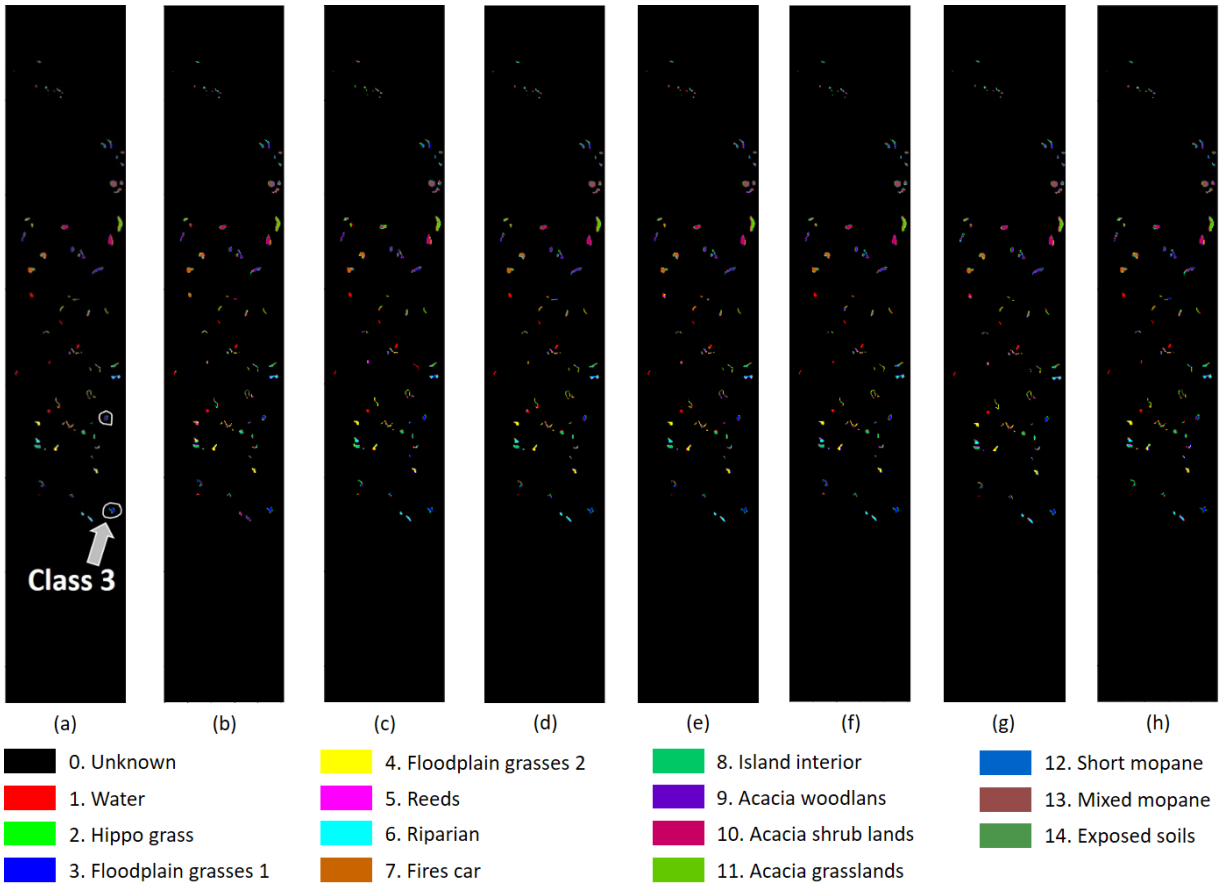

Fig. 9. Land cover categories and classification results of the BOT dataset with the compared methods: (a) ground truth; (b) SVM; (c) 2D-CNN; (d) 3D-CNN; (e) SSRN; (f) AfNet; (g) HybridSN; (h) HTD-2D-3D-PCN.

are small and closely related to other classes, we employ the NAMD2 method. This method enhances the differentiation of closely related classes and serves as input for 2D-CNN to learn spatial feature extraction. Similar to our experiments on previous datasets, this approach combines features extracted by 2D-CNN and 3D-CNN, enabling more effective learning of both spatial and spectral features, ultimately resulting in an improved Overall Accuracy (OA).

VI. Discussion

The introduction of hybrid 2D-3D models in hyperspectral image processing has been extensively studied in previous research [35-40]. Many existing models in this domain utilize Principal Component Analysis (PCA) for dimensionality reduction and are integrated into either a 3D CNN architecture following a 2D CNN (3D in series 2D) [35-38] or a 2D CNN architecture following a 3D CNN (2D in series 3D) [40]. Another approach, as presented in [39], involves a multi-branch strategy where multiple channels are combined into parallel 3D CNN architectures.

In contrast, our proposed HTD-2D-3D-PCNN model represents a departure from these established approaches. It features a dual-branch architecture tailored to overcome the limitations of existing hybrid 2D-3D models. In one branch, recognizing the spatial emphasis of 2D CNNs, we integrate Hyperspectral Target Detection (HTD) for data filtering and dimension reduction, replacing PCA. This methodology enhances spectral features, effectively mitigates background noise, and reduces spectral bands. Furthermore, the inclusion of HTD accelerates training and improves spatial learning efficiency in subsequent 2D CNN layers.

The second branch of our model is constructed upon the foundation of the PCA-3D CNN framework, enabling the learning of spectral and spatial features using 3D kernels. The resultant feature maps from both branches are merged and processed for classification through fully connected layers. Our approach capitalizes on the strengths of the original PCA-3D CNN model while further augmenting spectral and spatial features through HTD-2D-CNN.

Experimental results, presented in Tables 6 to 10, consistently demonstrate the superiority of our HTD-2D-3D-PCNN model across all five diverse datasets. This model achieves the highest overall accuracy, average accuracy, and KAPPA coefficient, outperforming recent CNN models and confirming the effectiveness of our proposed methodology.

By leveraging the HTD algorithm, which is based on spectral features, we effectively enhanced the representation of spectral information for each target class within the images. Our methodology distinguishes itself from existing attention methods, as most attention techniques are an integral part of deep learning network architectures, aiming to focus on learning specific features during the network training process. In contrast, the HTD algorithm utilized in our proposed method operates independently of the deep learning network. The HTD algorithm is applied to the original images through a specific algorithm, enhancing and highlighting the spectral signature of each target class. The outcome of this process serves as the input for network training. Moreover, the HTD algorithm we employ demands minimal prior knowledge or background for different classes, solely requiring the spectral features of a single target class. This is in contrast to many attention methods, which may necessitate a significant amount of prior knowledge or background information for various categories within the dataset. The distinct feature of our proposed approach lies in its pre-processing step using the HTD algorithm, augmenting the visibility of target classes before integrating the data into the network training process.

Similar to LRR-Net [54-55], which integrates the low-rank representation (LRR) model with deep learning techniques to achieve robust separation capabilities for background and target features in hyperspectral anomaly detection (HAD), our paper focuses on enhancing classification performance in hyperspectral image classification (HSIC) using HTD techniques. While HAD typically involves binary classification, HSIC may encompass over ten classes. In such scenarios, the approach in this paper involves leveraging HTD technology to enhance each class individually, achieving improved classification performance across multiple categories.

Subsequently, we utilized 2D-CNN to extract spatial features, thereby achieving a fusion of spectral and spatial information. Furthermore, after reducing the data dimensionality through PCA, both spectral and spatial features played substantial roles in model learning and training. This was particularly beneficial when employing 3D-CNN for extracting spectral and spatial features.

However, it was observed that using HTD followed by 2D-CNN convolution resulted in decreased inter-class correlations, limiting the advantages when concatenating these features and processing them further with 3D-CNN for feature learning. In contrast, adopting a parallel architecture that combined the outputs of 2D-CNN and 3D-CNN proved to be more effective, as it merged two non-overlapping channels. By utilizing subsequent fully connected layers for classification, the parallel approach achieved higher overall accuracy compared to the sequential method.

To substantiate the effectiveness of our proposed model framework, we conducted comparisons with several recent CNN methods, including FADCNN, AfNet, CACNN, SSRN, 2D-3D-D, pResNet, HybridSN, FFDN, and CNNDH. Additionally, the data from Tables 6 to 10 clearly indicate that our HTD-2D-3D-PCNN model, built upon the proposed framework, outperforms traditional supervised learning methods like SVM, 2D-CNN, and 3D-CNN in terms of overall accuracy, showcasing improvements ranging from approximately 1% to 25%. The experimental results and analysis presented in these tables lead to the conclusion that our proposed model architecture consistently demonstrates superior performance across all five different datasets, particularly excelling in smaller classes and effectively handling fine details at the edges of each class.

The advantages of the method proposed in this paper have been delineated earlier. One aspect that requires ongoing refinement is the selection of the HTD algorithm, as it can influence the outcomes across diverse datasets. The choice of the method may need to be tailored based on the characteristics of each specific dataset. Hence, the generalization capabilities of the various HTD algorithms discussed in this paper can be continuously optimized to enhance adaptability across a range of datasets.

Other limitations of the proposed method, such as spectral variability, present inherent challenges in hyperspectral imagery obtained from airborne or satellite sources [56]. In practical applications, the computational resources required for the training process may impose constraints, necessitating higher hardware specifications. Variabilities that warrant further exploration encompass environmental conditions, as scenes captured in different shooting scenarios and with various hyperspectral instruments may yield varying pixel resolutions and spectral intensities. Therefore, when applying the proposed method to different public datasets, it becomes imperative to configure distinct window sizes and employ diverse HTD algorithms to address situations where varying spectral intensities are exhibited across different datasets.

VII. Effect of Parameters

This chapter analyzes the impact of different parameters in the proposed model architecture on its classification performance. Specifically, we investigated the effects of various HTD algorithms, window sizes, and training sample proportions.

A. Different HTD algorithms

Table XI shows the results when the training samples were fixed at 10%. For the IP, SA, and KSC datasets, the NAMD algorithm achieved the best overall accuracy with values of 98.41%, 99.92%, and 99.82%, respectively. For the PU dataset, the CEM algorithm yielded the highest overall accuracy, at 99.85%. In the case of the BOT dataset, the NAMD2 algorithm achieved the best classification performance, with an overall accuracy of 98.82%.

Table XI illustrates that using the results of either HTD or PCA as input for the 2D branch in the network model yields different outcomes. The classification accuracy obtained when employing HTD as the input for the 2D branch is higher

TABLE XI
CLASSIFICATION PERFORMANCE OF DIFFERENT ALGORITHMS AS INPUT TO THE 2D BRANCH

Dataset	Measure		Target Detection Algorithms			
		PCA	CEM	CEM^2	NAMD	NAMD^2
IP	OA	95.71	96.70	97.62	**98.41**	97.53
	AA	94.44	95.72	95.99	98.27	95.58
	Kappa	95.27	96.24	97.29	98.18	97.19
PU	OA	96.42	**99.85**	99.65	99.52	99.32
	AA	96.15	99.73	99.39	99.11	98.82
	Kappa	96.23	99.82	99.54	99.37	99.09
SA	OA	96.87	99.90	99.87	**99.92**	99.78
	AA	96.68	99.81	99.74	99.86	99.48
	Kappa	96.72	99.90	99.85	99.91	99.75
KSC	OA	96.83	98.83	88.54	**99.82**	99.51
	AA	96.57	98.29	85.28	99.75	99.39
	Kappa	96.63	98.71	87.21	99.80	99.45
BOT	OA	95.69	98.69	97.95	98.48	**98.82**
	AA	95.50	98.63	97.72	98.44	98.74
	Kappa	95.37	98.57	97.78	98.36	98.72

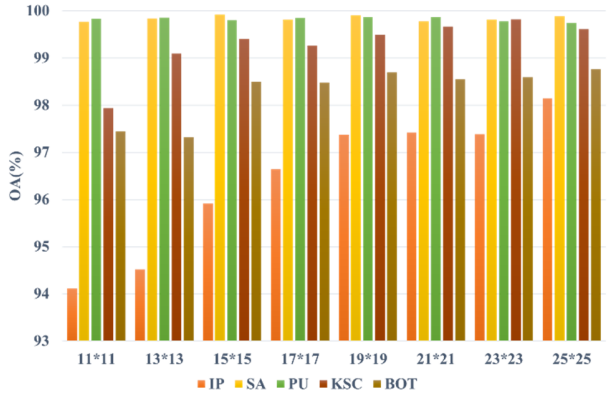

Fig. 10. Classification performance on different window sizes.

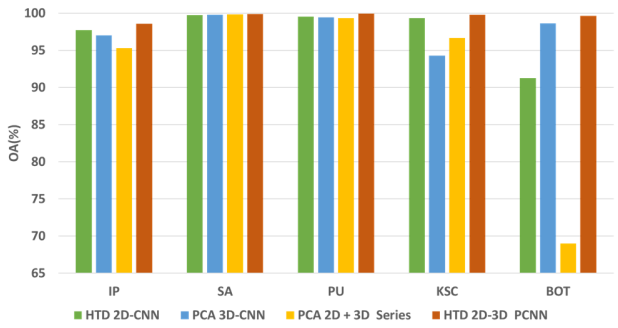

Fig. 11. Classification performance on different network model.

compared to using PCA algorithm as the input. Therefore, it can be inferred that HTD enhances the spectral features of the target class beforehand, allowing the network model to obtain clearer spatial information for learning and classification.

Furthermore, from the experimental results in Table XI for the five datasets using different HTD algorithms, it could be observed that generally, NAMD outperformed CEM in terms of the classification performance. Three main differences were identified between the two HTD algorithms discussed in the previous chapters. First, the calculation of the signal-to-noise

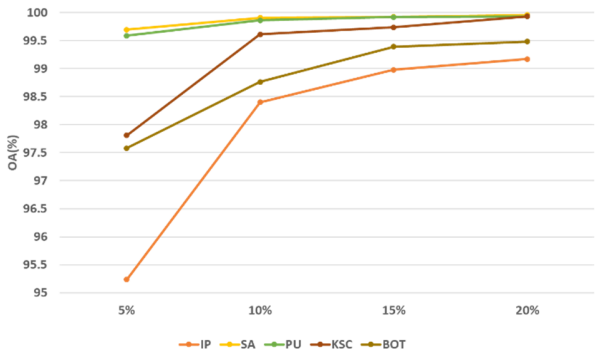

Fig. 12. Classification performance on different percentages of training samples.

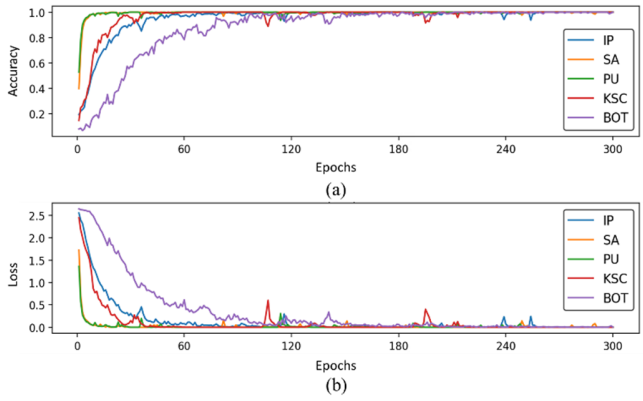

Fig. 13. (a) Training accuracy and (b)Training loss of the HTD-2D-3DPCNN.

ratio (SNR) was different. NAMD performs SNR calculations in the DS-space, while CEM performs it in the R-space, effectively eliminating the correlation matrix representing the background in the original data space. Second, both algorithms employ different methods to eliminate the influence of the background. NAMD uses variance matrix K, while CEM uses correlation matrix R. Third, since NAMD is an algorithm based on sub-pixel target detection, compared to pure pixel target detection, it requires the estimation of abundance scores using the Mahalanobis distance (MD). This led to the need to subtract the background mean (μb) from all pixel samples (r) and target signal (t) in the detector. In Table XI, it becomes evident that, with the exception of the BOT dataset, the majority of datasets do not yield benefits from implementing the HTD algorithm in a squared manner. This specific implementation tends to overly emphasize the already strong spectral features while sacrificing the weaker ones, resulting in a reduction in Overall Accuracy (OA). Hence, it can be concluded that the squared form of the HTD algorithms are not universally suitable for all datasets.

B. Different Window size

Figure 10 shows the overall accuracy results obtained with different window sizes before feeding the data into the CNN model. Eight different window sizes were tested for each of

TABLE XII
TRAINING TIME OF DIFFERENT ALGORITHMS

Time (Seconds)	HTD 2D-CNN	PCA 3D-CNN	PCA 2D+3D series	HTD 2D-3D PCNN
IP	22.743	35.714	28.104	85.524
PU	32.834	42.457	104.215	158.692
SA	82.901	96.846	130.030	193.107
KSC	16.167	18.539	17.603	43.691
BOT	7.041	13.799	12.595	17.201

the five datasets: 25 x 25; 23 x 23; 21 x 21; 19 x 19; 17 x 17; 15 x 15; 13 x 13; and 11 x 11.

The experimental findings demonstrated that for the IP and BOT datasets, setting the window size to 25 x 25 yielded the best performance, achieving overall accuracies of 98.41% and 98.82%, respectively. For the KSC dataset, the highest overall accuracy (99.82%) was achieved when the window size was set to 23 x 23. In the case of the PU dataset, the optimal window size was 19 x 19, resulting in an overall accuracy of 99.85%. As for the SA dataset, the best performance was obtained with a window size of 15 x 15, achieving an overall accuracy of 99.92%.

Furthermore, the experiments revealed that as the window size increased, the overall accuracy showed an upward trend. However, the window size also impacted the training time of the model, with larger window sizes requiring longer training times. Therefore, selecting an appropriate window size was essential, as its balanced accuracy and training efficiency for the specific requirements of the application.

C. Ablation Analysis

In Figure 11, an ablation analysis of network models is presented, comparing 2D-CNN, 3D CNN, 2D+3D Series CNN, and HTD-2D-3D-PCNN. For 3D-CNN and 2D+3D Series CNN, experiments were conducted with PCA dimensionality reduction after HTD, as the spectral features are already enhanced and lack spectral correlations. Following HTD, only spatial features need to be captured through 2D-CNN.

However, PCA, performing a linear transformation, is better suited for the simultaneous extraction of spectral and spatial features using 3D-CNN. Therefore, in our model design, HTD is coupled with 2D-CNN, while PCA is paired with 3D CNN.

We observe that HTD-2D-CNN performs quite well, outperforming PCA-3D CNN in most datasets besides BOT data. This is attributed to the relatively small size of targets in BOT data, where 2D CNN struggles to extract spatial features. In contrast, 3D-CNN, with its capability to simultaneously extract spectral and spatial information, performs better in such scenarios.

However, results indicate that, across five different datasets, our proposed HTD-2D-3D-PCNN achieved the highest classification accuracy when employing a parallel architecture, which is superior to using a sequential architecture. This implies that through this dual-channel parallel approach, it is possible to extract the maximum amount of information, thereby producing optimal results.

D. Different Numbers of Training Samples

Figure 12 illustrates the overall accuracy results obtained with different training sample ratios, while keeping the validation sample ratio fixed at 5%. The experiments were conducted using four different training sample ratios for each of the five datasets (20%, 15%, 10%, and 5%). The window sizes used were 25 x 25 for the IP and BOT datasets, 23 x 23 for the KSC dataset, 19 x 19 for the PU dataset, and 15 x 15 for the SA dataset.

The experimental findings revealed that the proposed HTD-2D-3D-PCNN model achieved the highest overall accuracies on all five datasets when the training sample ratio was set to 20%. Specifically, for the IP dataset, the overall accuracy reached 99.17%; for the PU dataset, the overall accuracy was 99.93%; for the SA dataset, the overall accuracy was 99.96%; for the KSC dataset, the overall accuracy was 99.93%; and for the BOT dataset, the overall accuracy was 99.48%.

Fig 12. demonstrates that as the training sample ratio increased, the overall accuracy also improved. However, the training sample ratio also affected the training time of the model, with higher sample ratios leading to longer training times. Therefore, there was a trade-off between accuracy and training time, and selecting an appropriate training sample ratio based on the specific requirements and constraints of the application was crucial.

E. Convergence and Complexity Analysis

To provide a comprehensive analysis of the proposed method, this paper tracks the variation trends of training accuracy and training loss for four different network models across five datasets. The accuracy and loss curves are illustrated in Figure 13 (a) and Figure 13 (b), where the vertical axis represents the evolution of accuracy or loss, and the horizontal axis represents the number of epochs. From Figure 13 (b), it can be observed that the experiments were conducted with a fixed number of epochs set at 300 for the network models. The experiments on all five datasets showed that the loss value tended towards 0 after approximately 200 epochs, particularly excelling on the SA and PU datasets. This indicates reasonable training and good convergence of the proposed method's network models. Given that the proposed method employs HTD as a preprocessing technique, spatial information holds relative importance. However, the BOT dataset, characterized by sparse images, may affect spatial features, potentially resulting in slower convergence compared to other datasets.

Table XII compares the training time of the proposed method with other experimental models across the five datasets. Training time and computational costs are influenced by factors such as the size of the hyperspectral image, the number of training samples, neural network layers, and branches. The results reveal that the training time cost of the network model proposed in this paper is longer compared to

other experimental models. This is attributed to the proposed network model's increased number of branches and deeper depth, which nonetheless achieves optimal classification performance. Among the datasets, the SA dataset incurs the longest training time cost due to its larger number of samples per class. In contrast, the BOT dataset exhibits the shortest training time cost as it contains fewer samples per class compared to other datasets, with this paper setting its training ratio to only 5%.

VIII. Conclusion

This paper presents a novel network model for hyperspectral image classification, which integrates hyperspectral target detection (HTD) with a hybrid 2D-CNN and 3D-CNN architecture. The model comprised two branches: one utilizing the HTD algorithm to enhance and highlight spectral features of the target class in the image, followed by learning the spatial features using 2D-CNN. The other branch performed PCA dimensionality reduction on the original data, followed by extracting both spectral and spatial features using 3D-CNN. The resulting feature maps from both branches were merged and fused, and classification results were obtained through fully connected layers.

This paper contributes to the field in two significant ways. Firstly, it introduces the utilization of Hyperspectral Target Detection (HTD) in the data preprocessing stage, offering an alternative to conventional methods primarily reliant on Principal Component Analysis (PCA) for dimensionality reduction. HTD enhances spectral features, suppresses background noise, and improves feature extraction and learning efficiency in subsequent models. Additionally, it achieves significant dimension reduction, leading to improved training outcomes.

Secondly, the proposed method combines the strengths of PCA + 3D CNN with HTD-2D-CNN to enhance spectral features in spatial learning. The independent training of these branches synergistically merges feature maps for classification, particularly beneficial for handling smaller class sizes and capturing fine details at the edges. Comparative evaluations across five publicly available hyperspectral datasets demonstrate the superior performance of the proposed approach in terms of overall accuracy, average accuracy, and the KAPPA coefficient, outperforming recent methodologies.

In future research, further exploration of HTD techniques could enhance classification performance and improve our method's generalization capabilities. Additionally, investigating adjustments to the network architecture, such as incorporating attention mechanisms and addressing various variabilities, would be valuable. While prioritizing classification performance, efforts could focus on optimizing the architecture to reduce training time costs. This might involve designing alternative networks to enhance classification efficiency and continually pushing the boundaries of performance.

References

[1] H. Cetin, J. T. Pafford and T. G. Mueller, "Precision agriculture using hyperspectral remote sensing and GIS," *Proceedings of 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST 2005.*, Istanbul, Turkey, 2005, pp. 70-77.

[2] S.-Y. Chen, S.-H. Hsu, C.-Y. Ko and K.-H. Hsu, "Real-time defect and freshness inspection on chicken eggs using hyperspectral imaging," *Food control.* vol. 11, 109716, 2023.

[3] S.-Y. Chen, C.-Y. Chang, C.-S. Ou, and C.-T. Lien, "Detection of Insect Damage in Green Coffee Beans Using VIS-NIR Hyperspectral Imaging," *Remote Sensing*, vol. 12, no. 15, p. 2348, Jul. 2020.

[4] S. C. Kefauver, J. Peñuelas and S. L. Ustin, "Applications of hyperspectral remote sensing and GIS for assessing forest health and air pollution," *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, 2012, pp. 3379-3382.

[5] C. Weber *et al.*, "Hyperspectral Imagery for Environmental Urban Planning," *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, Spain, 2018, pp. 1628-1631.

[6] S.-Y. Chen, M.-F. Chiu, X.-W. Zou, "Real-time defect inspection of green coffee beans using NIR snapshot hyperspectral imaging," *Computers and Electronics in Agriculture.* vol. 197, 106970, 2022

[7] S. -Y. Chen, Y. -C. Cheng, W. -L. Yang and M. -Y. Wang, "Surface Defect Detection of Wet-Blue Leather Using Hyperspectral Imaging," in *IEEE Access*, vol. 9, pp. 127685-127702, 2021.

[8] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," *IGARSS 2003. IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, Toulouse, France, 2003, pp. 288-290.

[9] W. Song, S. Li, X. Kang and K. Huang, "Hyperspectral image classification based on KNN sparse representation," *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Beijing, China, 2016, pp. 2411-2414.

[10] S. R. Joelsson, J. A. Benediktsson and J. R. Sveinsson, "Random forest classifiers for hyperspectral data," Proceedings. *2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, Seoul, Korea (South), 2005, pp. 4 pp.-, doi: 10.1109/IGARSS.2005.1526129..

[11] L. Sun, Z. Wu, J. Liu and Z. Wei, "Supervised hyperspectral image classification using sparse logistic regression and spatial-TV regularization," *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, Melbourne, VIC, Australia, 2013, pp. 1019-1022.

[12] Y. Tang *et al.*, "Hyperspectral image classification using sparse representation-based classifier," *2014 IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC, Canada, 2014, pp. 3450-3453.

[13] X. Liu, Q. Sun, B. Liu, B. Huang and M. Fu, "Hyperspectral image classification based on convolutional neural network and dimension reduction," *2017 Chinese Automation Congress (CAC)*, Jinan, China, 2017, pp. 1686-1690.

[14] M. Chao and G. Meng-Yuan, "Hyperspectral Image Classification Based on Convolutional Neural Network," *2018 5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, Hangzhou, China, 2018, pp. 117-121.

[15] X. Cao, J. Yao, Z. Xu and D. Meng, "Hyperspectral Image Classification With Convolutional Neural Network and Active Learning," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604-4616, July 2020, doi: 10.1109/TGRS.2020.2964627.

[16] L. Mou, P. Ghamisi and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639-3655, July 2017.

[17] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao and H. Zhou, "Spatial Sequential Recurrent Neural Network for Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4141-4155, Nov. 2018.

[18] J. Feng, G. Bai, Z. Gao, X. Zhang and X. Tang, "Automatic Design Recurrent Neural Network for Hyperspectral Image Classification," *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Brussels, Belgium, 2021, pp. 2234-2237.

[19] L. Zhu, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Generative Adversarial Networks for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5046-5063, Sept. 2018.

[20] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza and J. Chanussot, "Graph Convolutional Networks for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966-5978, July 2021.

[21] T. Li, J. Zhang and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 5132-5136.

[22] J. Feng, Z. Ye, D. Li, Y. Liang, X. Tang and X. Zhang, "Hyperspectral Image Classification Based on Semi-Supervised Dual-Branch Convolutional Autoencoder with Self-Attention," *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, Waikoloa, HI, USA, 2020, pp. 1267-1270.

[23] M. Ahmad *et al*., "Hyperspectral Image Classification—Traditional to Deep Models: A Survey for Future Prospects," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 968-999, 2022.

[24] Ding, C.; Li, Y.; Xia, Y.; Wei, W.; Zhang, L.; Zhang, Y. Convolutional Neural Networks Based Hyperspectral Image Classification Method with Adaptive Kernels. *Remote Sens.* 2017, *9*, 618.

[25] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li and L. Tang, "Hyperspectral Images Classification With Gabor Filtering and Convolutional Neural Network," in *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2355-2359, Dec. 2017.

[26] N. He *et al*., "Feature Extraction With Multiscale Covariance Maps for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 755-769, Feb. 2019.

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

[28] C. Yu *et al*., "Hyperspectral Image Classification Method Based on CNN Architecture Embedding With Hashing Semantic Feature," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 6, pp. 1866-1881, June 2019.

[29] M. Kanthi, T. H. Sarma and C. S. Bindu, "A 3d-Deep CNN Based Feature Extraction and Hyperspectral Image Classification," *2020 IEEE India Geoscience and Remote Sensing Symposium (InGARSS)*, Ahmedabad, India, 2020, pp. 229-232.

[30] S. Li, X. Zhu, Y. Liu and J. Bao, "Adaptive Spatial-Spectral Feature Learning for Hyperspectral Image Classification," in *IEEE Access*, vol. 7, pp. 61534-61547, 2019.

[31] Q. Yang, Y. Liu, T. Zhou, Y. Peng and Y. Tang, "3D Convolutional Neural Network for Hyperspectral Image Classification Using Generative Adversarial Network," *2020 13th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Xi'an, China, 2020, pp. 277-283.

[32] 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," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, Art no. 5502205.

[33] M. He, B. Li and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3904-3908.

[34] Z. Zhong, J. Li, Z. Luo and M. Chapman, "Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847-858, Feb. 2018.

[35] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, S. K. Roy and X. Wu, "Hybrid Dense Network With Attention Mechanism for Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3948-3957, 2022.

[36] 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," in *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277-281, Feb. 2020.

[37] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao and A. Tariq, "Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 7570-7588, 2021.

[38] H. Fırat, M. E. Asker and D. Hanbay, "Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D CNN", *Remote Sens. Appl.: Soc. Environ.*, vol. 25, 2022.

[39] Z. Ge, G. Cao, X. Li and P. Fu, "Hyperspectral Image Classification Method Based on 2D–3D CNN and Multibranch Feature Fusion," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5776-5788, 2020

[40] C. Yu, R. Han, M. Song, C. Liu and C. -I. Chang, "A Simplified 2D-3D CNN Architecture for Hyperspectral Image Classification Based on Spatial–Spectral Fusion," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2485-2501, 2020.

[41] J. Xiang, C. Wei, M. Wang and L. Teng, "End-to-End Multilevel Hybrid Attention Framework for Hyperspectral Image Classification," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, Art no. 5511305.

[42] H. Guo, J. Liu, J. Yang, Z. Xiao and Z. Wu, "Deep Collaborative Attention Network for Hyperspectral Image Classification by Combining 2-D CNN and 3-D CNN," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4789-4802, 2020.

[43] C. Yu, R. Han, M. Song, C. Liu and C. -I. Chang, "Feedback Attention-Based Dense CNN for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-16, 2022, Art no. 5501916.

[44] H. Guo, J. Liu, Z. Xiao, and L. Xiao, "Deep CNN-based hyperspectral image classification using discriminative multiple spatial-spectral feature fusion," *Remote Sens. Lett.*, vol. 11, no. 9, pp. 827–836,2020.

[45] J. Wang, W. Li, M. Zhang, R. Tao and J. Chanussot, "Remote-Sensing Scene Classification via Multistage Self-Guided Separation Network," in *IEEE Geoscience and Remote Sensing Letters*, vol. 61, 2023, Art no. 5615312.

[46] M. Zhang, W. Li, X. Zhao, H. Liu, R. Tao and Q. Du, "Morphological Transformation and Spatial-Logical Aggregation for Tree Species Classification Using Hyperspectral Imagery," in *IEEE Geoscience and Remote Sensing Letters*, vol. 61, 2023, Art no. 5501212.

[47] D. Hong, B. Zhang, X. Li, Y. Li, C. Li, J. Yao, N. Yokoya, H. Li, P. Ghamisi, X. Jia, A. Plaza, G. Paolo, J. A. Benediktsson and J. Chanussot, "SpectralGPT: Spectral Foundation Model," in *arXiv Computer Vision and Pattern Recognition*, 2311.07113.

[48] J. Wang, W. Li, Y. Wang, R. Tao, Q. Du, "Representation-Enhanced Status Replay Network for Multisource Remote-Sensing Image Classification," in *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-13, 2023, doi: 10.1109/TNNLS.2023.3286422.

[49] D. Hong, B. Zhang, H. Li, Y. Li, Jing. Yao, Chenyu. Li, M. Werner, J. Chanussot, A. Zipf, X. Zhu, "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," in *ScienceDirect, Remote Sensing of Environment*, vol. 299, 2023, 113856.

[50] C. -I. Chang, "Hyperspectral Target Detection: Hypothesis Testing, Signal-to-Noise Ratio, and Spectral Angle Theories," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-23, 2022, Art no. 5505223.

[51] S.-Y. Chen, C. Lin, C.-H. Tai, and S.-J. Chuang, "Adaptive Window-Based Constrained Energy Minimization for Detection of Newly Grown Tree Leaves," Remote Sensing, vol. 10, no. 2, p. 96, Jan. 2018.

[52] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, p. 62-66, Jan. 1979.

[53] H. Zhuang, K. Deng, H. Fan and M. Yu, "Strategies Combining Spectral Angle Mapper and Change Vector Analysis to Unsupervised Change Detection in Multispectral Images," in *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 681-685, May 2016.

[54] C. Li, B. Zhang, D. Hong, J. Yao and J. Chanussot, "LRR-Net: An Interpretable Deep Unfolding Network for Hyperspectral Anomaly Detection," in *IEEE Geoscience and Remote Sensing Letters*, vol. 61, 2023, Art no. 5513412.

[55] S.-Y. Chen, C. Lin, S.-J. Chuang, and Z.-Y. Kao, "Weighted Background Suppression Target Detection Using Sparse Image Enhancement Technique for Newly Grown Tree Leaves," *Remote Sensing*, vol. 11, no. 9, p. 1081, May 2019.

[56] D. Hong, N. Yokoya, J. Chanussot and X. X. Zhu, "An Augmented Linear Mixing Model to Address Spectral Variability for Hyperspectral Unmixing," in *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1923-1938, April 2019, doi: 10.1109/TIP.2018.2878958.

Shih-Yu Chen (M'15) received his BS degree in Electrical Engineering from Da-Yeh University in 2005, followed by an MS in Electrical Engineering from National Chung Hsing University in 2010, and a Ph.D. in Electrical Engineering from the University of Maryland, Baltimore County (UMBC) in 2014. He served as an Assistant Professor in the Department of Computer Science and Information Engineering at National Yunlin University of Science and Technology, Taiwan, from 2014 to 2018. Since 2018, Dr. Chen has been an Associate Professor in the same department. His research interests include multi/hyperspectral image processing, medical signal processing, image processing, and pattern recognition.

Kai-Hsun Hsu received the MS degrees in Computer Science and Information Engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2023. His current research interests include remote sensing image processing and analysis, pattern recognition, computer vision and machine learning.

Tzu-Hsien Kuo is currently a master student at Yunlin National University of Science and Technology in Yunlin, Taiwan. His current research interests include remote sensing image processing and analysis, pattern recognition, computer vision and machine learning.