remote sensing

MDPI

Article

Multi-Level Feature Extraction Networks for Hyperspectral Image Classification

Shaoyi Fang , Xinyu Li , Shimao Tian, Weihao Chen and Erlei Zhang *

School of Information Engineering, Northwest A&F University, Xi'an 712100, China;
shaoyi.fang@nwafu.edu.cn (S.F.); xinyuli@nwafu.edu.cn (X.L.); tsm2023056178@nwafu.edu.cn (S.T.);
weihao.chen@nwafu.edu.cn (W.C.)
* Correspondence: erlei.zhang@nwafu.edu.cn

Abstract: Hyperspectral image (HSI) classification plays a key role in the field of earth observation missions. Recently, transformer-based approaches have been widely used for HSI classification due to their ability to model long-range sequences. However, these methods face two main challenges. First, they treat HSI as linear vectors, disregarding their 3D attributes and spatial structure. Second, the repeated concatenation of encoders leads to information loss and gradient vanishing. To overcome these challenges, we propose a new solution called the multi-level feature extraction network (MLFEN). MLFEN consists of two sub-networks: the hybrid convolutional attention module (HCAM) and the enhanced dense vision transformer (EDVT). HCAM incorporates a band shift strategy to eliminate the edge effect of convolution and utilizes hybrid convolutional blocks to capture the 3D properties and spatial structure of HSI. Additionally, an attention module is introduced to identify strongly discriminative features. EDVT reconfigures the organization of original encoders by incorporating dense connections and adaptive feature fusion components, enabling faster propagation of information and mitigating the problem of gradient vanishing. Furthermore, we propose a novel sparse loss function to better fit the data distribution. Extensive experiments conducted on three public datasets demonstrate the significant advancements achieved by MLFEN.

Keywords: hyperspectral image classification; convolutional neural networks; vision transformer

check for updates

Citation: Fang, S.; Li, X.; Tian, S.; Chen, W.; Zhang, E. Multi-Level Feature Extraction Networks for Hyperspectral Image Classification. *Remote Sens.* **2024**, *16*, 590. https://doi.org/10.3390/rs16030590

Academic Editor: Paul Scheunders

Received: 4 November 2023
Revised: 24 January 2024
Accepted: 29 January 2024
Published: 4 February 2024

1. Introduction

Hyperspectral image (HSI) is composed of abundant spatial and spectral data, enabling effective differentiation between various types of land cover. This technology finds extensive applications in urban planning [1,2], geological exploration [3,4], precision agriculture [5,6], and other domains. HSI classification plays a fundamental role in remote sensing as it offers a robust means of analyzing and interpreting information embedded within HSI.

Conventional machine learning approaches typically prioritize the analysis of spectral information in tasks such as *k*-nearest neighbor [7], support vector machine [8], random forest [9], and sparse representation [10]. While these methods are straightforward and easily scalable, they struggle when working with high-dimensional data and small sample sizes, which is known as the Hughes phenomenon [11]. To overcome the challenge, band selection is often adopted to select the most useful bands and thus reduce redundant information, such as the local feature descriptor network [12] and similarity-based ranking method [13]. In addition, dimensionality reduction techniques like principal component analysis (PCA) [14] and linear discriminant analysis [15] are commonly employed to map high-dimensional HSI data into lower-dimensional spaces. However, recent studies demonstrated that relying solely on spectral information and disregarding spatial information makes it difficult for classification algorithms to accurately capture surface object characteristics, such as spatial distribution and morphological features. This ultimately impacts classification accuracy [16]. To address this, various models that extract both spatial and spectral features were proposed, including the Markov random field [17], extended multi-attribute profile [18], Gabor filters [19], and hypergraph structure [20]. Nevertheless, these conventional machine learning methods are limited in their ability to leverage deep nonlinear features, which can result in suboptimal performance.

The advancement of deep learning techniques has greatly contributed to the progress of HSI processing [21]. Numerous models based on deep learning were proposed for HSI classification tasks. Some of the well-known baseline networks include recurrent neural networks [22], graph convolutional neural networks [23], autoencoders [24], generative adversarial networks [25], capsule networks [26], long short-term memory networks [27], and convolutional neural networks (CNNs) [28]. Among these methods, CNNs are the most widely used and can be categorized into 1D convolutional neural network (1D-CNN) [29], 2D convolutional neural network (2D-CNN) [30], and 3D convolutional neural network (3D-CNN) [31] based on their dimensions. Hu et al., introduced 1D-CNN to capture features from the spectral dimension and achieved better accuracy compared to traditional machine learning approaches. The 2D-CNN method is commonly used to extract spatial features, while the 3D-CNN is employed to capture spatial–spectral features. The hybrid spectral CNN (HybridSN) [32] combines the benefits of both 2D-CNN and 3D-CNN, leading to superior classification performance. Furthermore, Paoletti et al., utilized pyramidal bottleneck residual cells [33] to enhance performance by increasing the dimension of spectral and spatial attributes layer by layer. On the other hand, Ma et al., proposed a deconvolution network with skip connections to address issues such as the loss of important information during down-sampling and the lack of training samples [34]. While these methods are more efficient than conventional machine learning approaches, they often fail to fully leverage features at different levels. In recent years, many deep learning methods using representation learning applied to tasks such as HSI classification, segmentation and image super-resolution have also emerged, including [35–38]. For improving the performance of HSI classification tasks, researchers proposed methods based on the joint extraction of spectral–spatial features for representation learning. A central vector-oriented self-similarity network (CVSSN) [39] enhances characterization of the model by considering the association between internal pixels and their neighbors.

To emphasize regions of interest in an image while suppressing irrelevant background regions, researchers have suggested attention mechanisms, taking inspiration from the visual mechanisms of humans. Noteworthy examples of such works include the squeeze-and-excitation network (SENet) [40] and the dual attention network (DANet) [41]. SENet adjusts the scale of channel features and enables the model to focus more on important channel information, while DANet improves the model's perception of contextual information by introducing a dual-attention mechanism. Additionally, the spectral–spatial attention network (SSAN) [42] combines the capabilities of CNN for modeling spatial interior dependence and recurrent neural network for characterizing spectral sequences. The multi-attention fusion network (MAFN) [43] incorporates attention modules that cater to spatial and spectral perspectives, mitigating issues like band redundancy and pixel interference. Furthermore, the multimodal transfer feature fusion network [44] utilizes the local attention mechanism and a multi-task learning strategy to facilitate the learning of HSIs across different domains. However, it is important to note that these attention-based methods often necessitate a substantial amount of annotated data for effective training, posing a challenge when dealing with HSIs that have limited annotated samples.

Transformer models are gaining popularity in the realm of natural language processing [45]. Researchers apply the transformer architecture, originally developed for language processing, to the image domain with the introduction of vision transformer (ViT) [46]. ViT achieves impressive results in image classification by dividing images into smaller patches and treating them as sequential data. In addition to the basic ViT model, there have been advancements in this area with the development of variants such as pyramid vision transformer [47], swin transformer [48], and data-efficient image transformers [49]. Another approach called tokens-to-token ViT [50] improves tokenization in ViT by utilizing a soft-split operation. Researchers also explored the application of ViT in HSI classification. The novel backbone network rethinking HSI classification from a sequential perspective with transformers [51] introduces ViT to the field of HSI classification, but it largely focuses on mining spectral information and lacks the utilization of spatial information. To address this limitation, various methods were proposed to combine CNNs with ViT to extract both spatial and spectral features, including the convolutional transformer network [52], spatial–spectral transformer [53], and neighborhood enhancement hybrid transformer network [54]. Other approaches such as spectral–spatial feature tokenization transformer [55] and hyperspectral image transformer (HiT) [56] were developed to extract spectral–spatial features and semantic features. However, these ViT-based methods often overlook the importance of feature reuse and information transfer.

In summary, the above-mentioned methods have faced two issues: (1) Treating HSI as linear vectors, compromising 3D attributes and spatial structure. (2) Repeatedly concatenating encoders, leading to information loss and gradient vanishing.

To overcome these challenges, on the one hand, hybrid convolution operations are utilized in order to capture the 3D characteristics and spatial structure of HSI. Specifically, to address any potential edge effects caused by the convolution operation, a novel band shift strategy is developed. Furthermore, an attention module is employed to explore the relationship between the spatial and spectral aspects of the data, thereby emphasizing the features with high discriminative capability. On the other hand, a modification is made to the arrangement of encoders in order to enhance the adaptability to the HSI classification task's small-scale dataset. This adjustment involves eliminating the simple concatenation method to ensure that excessive information loss is avoided. To enable feature reuse, an adaptive feature fusion component is incorporated.

The major contributions in this study can be outlined as follows.

(1) A multi-level feature extraction network (MLFEN) for HSI classification is proposed. By combining the capability of CNN's local spatial–spectral feature capture and ViT's global sequence modeling, MLFEN achieves effective fusion of shallow to deep features of HSIs.

(2) A sophisticated hybrid convolutional attention module (HCAM) is suggested, which incorporates band shift, hybrid convolution, and attention mechanisms to efficiently capture and enhance multidimensional features. By seamlessly combining these essential techniques, HCAM can empower the model to gain a comprehensive understanding of the intricate details present within the image.

(3) A novel variation of the ViT called enhanced dense vision transformer (EDVT) is introduced. EDVT is specifically designed for characterizing HSI data. To address the issue of information loss due to deep networks, EDVT incorporates a modified architecture for its encoders. Additionally, EDVT includes an adaptive feature fusion (AFF) component that enables effective information transfer and feature reuse.

(4) A new sparse loss function for HSI classification is developed, combining the benefits of both the cross-entropy loss function and a sparsity regularization operator. This loss function contributes to sparse representation, thus effectively solving the overfitting problem and improving the robustness and accuracy of classification.

The remainder of the article is organized as follows. In Section 2, the methodology of the proposed network is presented. Section 3 details the datasets, provides specific implementation steps and parameter settings, and analyzes the experimental findings. Section 4 compares the computational cost and performance of MLFEN with other methods. Finally, Section 5 provides a summary of this work and outlines future directions.

2. Method

Figure 1 presents a visual representation of the MLFEN architecture. To begin with, the original HSIs undergo preprocessing using PCA. This process converts the HSIs into patch cubes which are then fed into two distinct modules: the hybrid convolutional attention module (HCAM) and the enhanced dense vision transformer (EDVT). For the classification task, a multi-layer perception network (MLP) is implemented as the classifier.

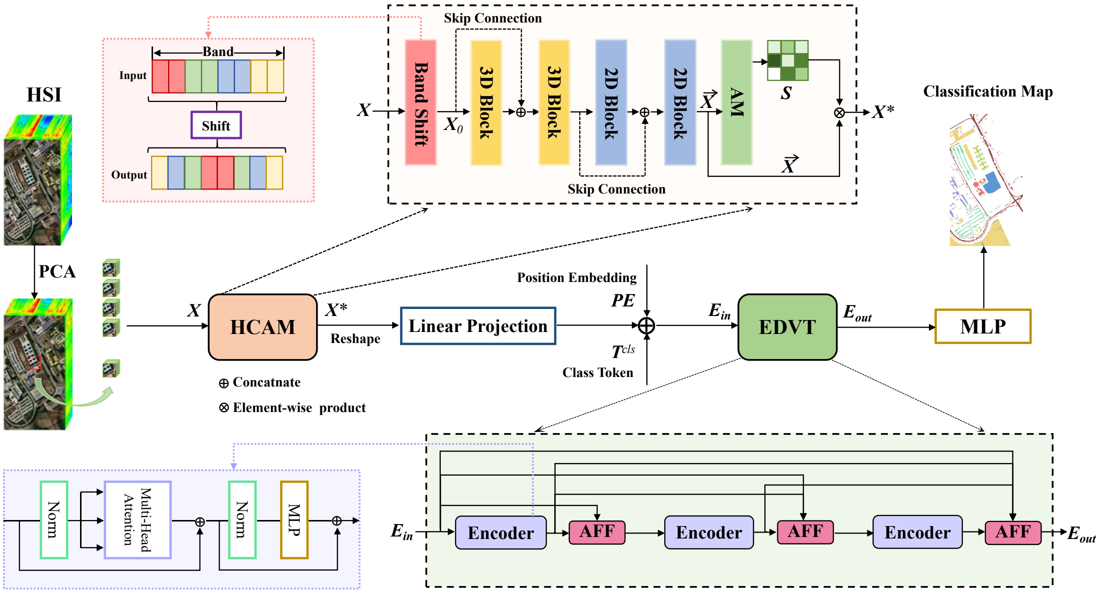


Figure 1. The overall framework of the proposed MLFEN. MLFEN is composed of two primary sub-networks, including HCAM and EDVT. HCAM consists of three key components, which are band shift, convolutional operations (e.g., 3D Block and 2D Block), and AM. While EDVT incorporates two design elements, namely dense connection and AFF.

2.1. *Hybrid Convolutional Attention Module (HCAM)*

The main role of the HCAM module is to acquire the shallow joint spatial–spectral information of HSI. In order to achieve superior performance, the structure of the HCAM is both complete and compact, containing a band shift component, four convolutional blocks and an attention module (AM).

2.1.1. Band Shift

The PCA algorithm ranks the principal components according to the magnitude of the variance. The larger the variance, the more information it contains. Therefore, after PCA processing, the feature information of each band of the HSI image is arranged in descending order, which results in the most informative features being at the edge. However, the pixel points at the edges during the convolution operation are often not fully covered by the convolution kernel. This edge effect can make the mining of spectral information insufficient and further affect the convolution effect.

Based on the above analysis, a band shift strategy is proposed for mitigating the undesirable consequences of edge effects. Band shift aims to move key spectral channels to the central position across all data, while relocating less critical spectral channels towards the edges of the data. The benefits of doing so include increasing the likelihood of effective extraction of spectral features through convolutional operations, and maintaining critical spectral channels at the core position of the receptive field. With the shift strategy, the negative impact due to the edge effect of the convolution operation is mitigated, and the separability of the bands is improved thus enhancing the feature representation of the model. The mapping function $g(\cdot)$ of the band shift strategy can be expressed as:

$$\begin{cases} g(v) = 2(\lfloor \frac{p}{2} \rfloor - v) - 1, v \in [0, \frac{p}{2}) \\ g(v) = 2(v - \lfloor \frac{p}{2} \rfloor), v \in [\frac{p}{2}, p-1] \end{cases} \tag{1}$$

where p represents the number of principal components after PCA, $g(v)$ and v denote the band index before and after shift. Applying this strategy can enhance the discrimination of target features, alleviate the band correlation problem, and help improve the accuracy and reliability of classification.

2.1.2. Multidimensional Feature Extraction

As illustrated in Figure 1, HCAM contains two 3D convolution blocks and two 2D convolution blocks with a convolution kernel size of 3 × 3 × 3 and 3 × 3, respectively. This design first extracts the joint spatial–spectral features by 3D convolution and adaptively learns more abstract and advanced feature representations using 2D convolution. Batch normalization layers and rectified linear unit (ReLU) activation functions are added after each convolutional layer to avoid the gradient disappearance phenomenon and improve generalization ability. Compared to the single 3D-CNN method, this hybrid approach not only enables multi-level features to be obtained, but also dramatically reduces the computational cost.

To further enhance the interaction and fusion of features between different layers and save time and computational resources, the skip connection mechanism [57] is employed in HCAM. By doing so, it allows the model to more effectively learn and capture relevant features, ultimately leading to improved performance. Additionally, the implementation of skip connection in the HCAM module helps to improve computational efficiency while still maintaining robust performance.

Let $X_0 \in R^{k \times k \times d}$ (k and d are the patch size and number of bands of the 3D cubes) denote 3D cubes after the band shift, and X_i is the output of ith block. Then X_i can be expressed by:

$$\begin{cases} X_i = \delta(BN(Conv(X_{i-1}))), i = 2m-1, m = [1, \ldots, m] \\ X_i = \delta(BN(Conv(Cat(X_{i-2}, X_{i-1})))), i = 2m, m = [1, \ldots, m] \end{cases} \tag{2}$$

where *Cat* denotes the concatenation function, *Conv* means the convolutional operation and *BN* is the batch normalization layer. The symbol δ is the ReLU activation. When i = 4, X_i is $\vec{X}$, the input to the AM component. Through hybrid convolutional operations described above, the model comprehensively extracts shallow multidimensional spatial and spectral features.

2.1.3. Attention Module (AM)

The attention component is designed to incorporate broader contextual information into local contextual features. Inspired by the attention module proposed in the DANet [41], the local spatial–spectral features $\vec{X}$ is processed through convolution to obtain three feature mappings, namely $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$. These mappings can be represented as:

$$\begin{cases} \vec{X}_1 = Conv(\vec{X}, w_1) + b_1 \\ \vec{X}_2 = Conv(\vec{X}, w_2) + b_2 \\ \vec{X}_3 = Conv(\vec{X}, w_3) + b_3 \end{cases} \tag{3}$$

where the symbol w_1, w_2, and w_3 are weights of convolution layers while b_1, b_2, and b_3 denote biases. Then the pixel correlation is calculated by a matrix multiplication of the deformed $\vec{X}_1$ with $\vec{X}_2$ and the attention map S is obtained by using a softmax layer, which is defined as:

$$S = \frac{exp(\vec{X}_1 \vec{X}_2)}{\sum_{j=1}^{n} exp(\vec{X}_1 \vec{X}_2)} \tag{4}$$

where n is the number of features. We perform a matrix multiplication between $\vec{X}_3$ and S and then multiply it by a scale parameter α. Finally, the spatial attention feature is added to the input $\vec{X}$ to obtain the final result X^*:

$$X^* = \alpha \sum_{j=1}^{n} S\vec{X}_3 + \vec{X} \tag{5}$$

where α is initialized to 0 and continuously updated during the training process. We can deduce that the final feature X^* is obtained by taking a weighted sum of the features at all positions including the original features. As a result, X^* incorporates a global contextual view and selectively integrates contextual information based on the attention map.

2.2. Enhanced Dense Vision Transformer (EDVT)

Despite the benefits of CNNs in extracting nearby spatial contextual information, they face difficulties in capturing delicate variations in spectral data over extended distances. As a result, ViT is frequently utilized to augment the feature maps produced by CNNs, enriching the global spectral correlation and disparity information. ViT employs a conventional transformer encoder methodology, which includes MSA and MLP modules. To normalize each encoder block, layer normalization (LN) [58] is employed. The standard transformer encoder block can be represented by the following equation:

$$\begin{cases} E_l' = MSA(LN(E_{l-1})) + E_{l-1} \\ E_l = MLP(LN(E_l')) + E_l' \end{cases} \tag{6}$$

where E_l denotes the output features of lth encoder.

It is important to note that the original ViT consists of six encoders that are connected in a straightforward manner. However, there are various issues that still need to be addressed. Firstly, the connection between encoders is too simplistic for efficient feature reuse, which hinders the capturing of multi-level spectral information. Secondly, the training datasets used for HSI classification tasks are typically small and considered as limited samples in many research studies. Consequently, the network of six encoders can be seen as overly deep, resulting in a substantial loss of information.

In order to address the aforementioned issues, EDVT is proposed to minimize information loss and facilitate the reuse of multi-level features. As shown in Figure 2, EDVT comprises three encoder blocks, each consisting of an encoder and an AFF component. Taking inspiration from dense convolutional network [59], the two encoder blocks are interconnected with dense connections. These dense connections enable every encoder in the network to directly access the feature maps from all preceding encoders, thereby enhancing the extent of feature sharing and reuse. Additionally, the AFF component adaptively fuses varying numbers of features through a two-step process involving concatenation and convolution. The AFF_l operation can be defined as:

$$\widehat{E_l} = \ddot{w}(Cat(E_{in}, E_{l-1}, E_l)) \tag{7}$$

where $\widehat{E_l}$ is the fused feature after the lth AFF, $\ddot{w}$ denotes a network weight parameter that can be learned, and Cat represents the concatenation operation.

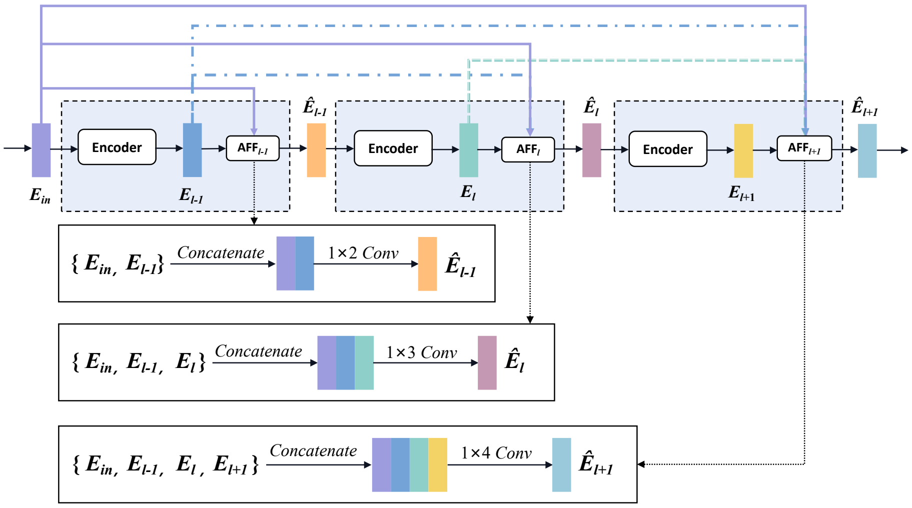

E_{in}
Encoder
E_{l-1}
AFF_{l-1}
$\hat{E}_{l-1}$
Encoder
E_l
AFF_l
$\hat{E}_l$
Encoder
E_{l+1}
AFF_{l+1}
$\hat{E}_{l+1}$
$\{E_{in}, E_{l-1}\}$ Concatenate 1×2 Conv $\hat{E}_{l-1}$
$\{E_{in}, E_{l-1}, E_l\}$ Concatenate 1×3 Conv $\hat{E}_l$
$\{E_{in}, E_{l-1}, E_l, E_{l+1}\}$ Concatenate 1×4 Conv $\hat{E}_{l+1}$

Figure 2. The illustration of the EDVT.

2.3. Sparse Loss Function

Supervised learning focuses on reducing the error by regularizing the parameters. Reducing the error ensures that the model fits the training data well, while parameter regularization prevents the model from overfitting the training data. Based on these principles, a new loss function for HSI classification tasks is proposed. This loss function utilizes the cross-entropy loss function L_{ce} [60] to minimize the error, and incorporates a sparsity regularization operator L_{sro} [61] as the regularization term.

The aim of the L_{ce} function is to reduce the difference between the predicted output and the ground truth, thereby bringing the predicted value closer to the actual label. L_{ce} is highly responsive to any inconsistency between the model output and the true label, enabling efficient gradient propagation and expediting model convergence. The calculation of L_{ce} is as follows:

$$L_{ce} = -\frac{1}{C}\sum_{i=1}^{C} y_i log(p_i) \tag{8}$$

where the symbol y_i is the probability value of the ith category in the true label while p_i denotes that in the predicted result. C is the number of categories.

However, when there is an imbalance in the number of samples across different categories in a dataset, the use of L_{ce} may result in the model being biased towards predicting the more frequent categories and disregarding the less frequent ones. Additionally, L_{ce} updates the model parameters by minimizing the prediction error rate during training, which can lead to overfitting if the training data are insufficient or the model is too complex. To address these issues, sparse regularization loss (L_{sro}) can be employed, which encourages the model to learn a more sparse feature representation. L_{sro} can be expressed as:

$$L_{sro} = \sum_{i=1}^{C} y_i - p_i \tag{9}$$

On one side, because of the scarcity of L_{sro}, it diminishes the preference for a larger number of categories and thus enhances the model's performance when handling datasets with imbalances. On the other side, by minimizing the impact of noisy or irrelevant input features, L_{sro} as a regularization term can effectively decrease the risk of overfitting and improve the model's ability to make generalizations. Therefore, we propose a new sparse loss function L that combines L_{sro} with L_{ce}, resulting in a more robust and easier to interpret model, as demonstrated below:

$$L = L_{ce} + \lambda L_{sro} \tag{10}$$

where λ represents the coefficient of L_{sro}. The combination of L_{ce} and L_{sro} can control the parameters while characterizing the probability distribution of the sample classification.

3. Results

3.1. Dataset Description

In order to evaluate the efficacy of MLFEN, five publicly available datasets are employed: Pavia University (PU), Kennedy Space Center (KSC), Salinas (SA), Indian Pines (IP), and Houston University (HU). Illustrations in Figure 3 depict the false color maps, ground truth maps, and color bars corresponding to the PU, KSC, SA, IP and HU datasets, respectively. Table 1 shows the basic information of the five datasets, which include image size, the number of categories and bands, band range, resolution, sensor and location. The land cover categories for the five datasets, along with the number of training and test set samples, are shown in Table 2. To ensure an unbiased evaluation, we randomly choose 1% of the data from each category of the datasets as the training set, while the remaining data are kept for testing purposes.

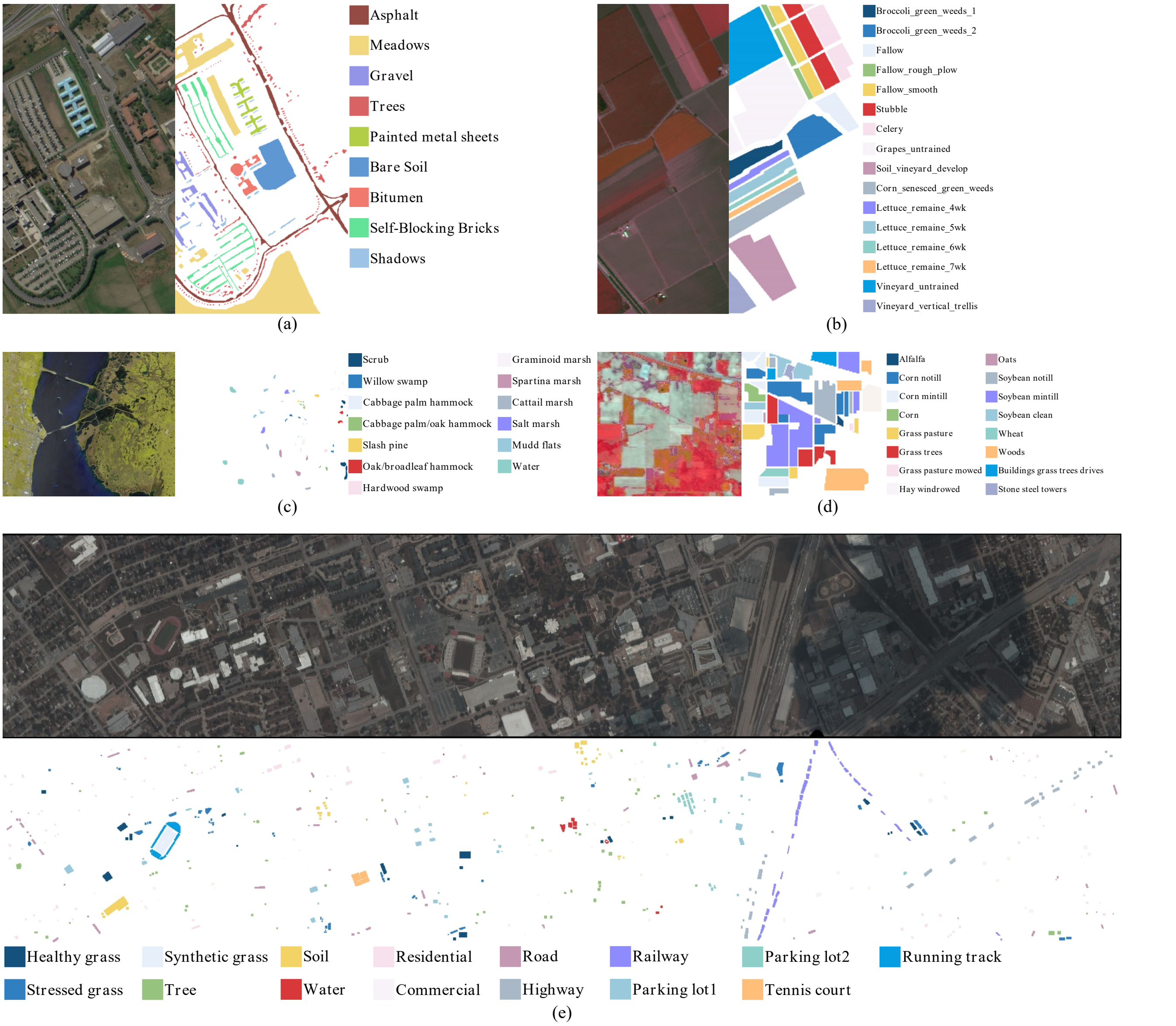


Figure 3. Visualization of five datasets. (**a**) PU dataset (false color image (RGB-R: 56, G: 33, B: 13)), (**b**) SA dataset (false color image (RGB-R: 38, G: 5, B: 20)), (**c**) KSC dataset(false color image (RGB-R: 59, G: 40, B: 23)), (**d**) IP dataset (false color image (RGB-R: 50, G: 30, B: 20)), and (**e**) HU dataset (false color image (RGB-R: 50, G: 30, B: 20)).

Table 1. Descriptions of the five HSI datasets.

Dataset	Image Size	Categories	Bands	Band Range	Resolution	Sensor	Location
PU	610 × 340	9	103	430~860 nm	1.3 m	ROSIS	Pavia, Italy
KSC	512 × 614	13	176	400~2500 nm	18 m	AVIRIS	Florida, USA
SA	512 × 217	16	204	360~2500 nm	3.7 m	AVIRIS	Salinas Valley, USA
IP	145 × 145	16	200	400~2500 nm	20 m	AVIRIS	North-Western Indiana, USA
HU	349 × 1905	15	144	364~1046 nm	2.5 m	ITRES CASI-1500	Texas, USA

Table 2. Different categories and corresponding numbers of samples for training and testing on the PU, KSC, HU, IP and SA dataset.

No.	Pavia University (PU) (1%)			Kennedy Space Center (KSC) (1%)			Houston University (HU) (1%)		
	Class	**Train**	**Test**	**Class**	**Train**	**Test**	**Class**	**Train**	**Test**
1	Asphalt	66	6565	Scrub	7	747	Healthy grass	13	1238
2	Meadows	186	18,463	Willow swamp	2	239	Stressed grass	13	1241
3	Gravel	20	2079	Cabbage palm hammock	2	252	Synthetic grass	7	690
4	Trees	30	3034	Cabbage palm/oak hammock	2	248	Tree	12	1232
5	Painted metal sheets	13	1332	Slash pine	2	157	Soil	12	1230
6	Bare Soil	50	4979	Oak/broadleaf hammock	2	225	Water	3	322
7	Bitumen	13	1317	Hardwood swamp	2	101	Residential	13	1255
8	Self-Blocking Bricks	36	3646	Graminoid marsh	4	423	Commercial	12	1232
9	Shadows	9	938	Spartina marsh	5	510	Road	13	1239
10				Cattail marsh	4	396	Highway	12	1215
11				Salt marsh	4	411	Railway	12	1223
12				Mudd flats	5	493	Parking lot1	12	1221
13				Water	9	909	Parking lot2	5	464
14							Tennis court	4	424
15							Running track	7	653
	Total	423	42353	Total	50	5111	Total	150	14,879

No.	Indian Pines (IP) (1%)			Salinas (SA) (1%)		
	Class	**Train**	**Test**	**Class**	**Train**	**Test**
1	Alfalfa	1	45	Broccoli_green_weeds_1	20	1889
2	Corn notill	14	1414	Broccoli_green_weeds_2	37	3689
3	Corn mintill	8	822	Fallow	19	1957
4	Corn	2	235	Fallow_rough_plow	13	1381
5	Grass pasture	48	435	Fallow_smooth	26	2652
6	Grass trees	73	657	Stubble	39	3920
7	Grass pasture mowed	1	27	Celery	35	3544
8	Hay windrowed	5	473	Grapes_untrained	112	11,159
9	Oats	1	19	Soil_vineyard_develop	62	6141
10	Soybean notill	10	962	Corn_senesced_green_weeds	32	3246
11	Soybean mintill	25	2430	Lettuce_remaine_4wk	10	1058
12	Soybean clean	6	587	Lettuce_remaine_5wk	19	1908
13	Wheat	2	203	Lettuce_remaine_6wk	9	907
14	Woods	13	1252	Lettuce_remaine_7wk	10	1060
15	Buildings grass trees drives	39	347	Vineyard_untrained	72	7196
16	Stone steel towers	9	84	Vineyard_vertical_trellis	18	1789
	Total	257	9992	Total	533	53,496

3.2. Experimental Setup

3.2.1. Implementation Details

All experiments are conducted in the pytorch framework using a server equipped with Ubuntu 20.04, Intel(R) Xeon(R) Platinum 8375C CPU, 250 GB RAM and NVIDIA GeForce RTX 3090 GPU.

After performing PCA, 100 principal components are retained. During the training phase, 200 iterations are undertaken with a learning rate of 0.001 and a batch size of 64. The sparse loss function consists of L_{ce} and L_{sro} with a coefficient of λ set to 0.01. The adam optimizer is employed with $\beta 1$ and $\beta 2$ parameters set to 0.1 and 0.99, respectively.

3.2.2. Evaluation Indicators

The performance of the proposed method is assessed using three commonly employed metrics: overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa). Each of these evaluation indicators positively reflects the classification accuracy. To ensure reliability, the reported values for these three metrics are averaged over ten independent runs in all experiments.

3.3. Parameter Analysis

3.3.1. The Influence of the Selection of Dimensionality Reduction Method

HSI contains an abundance of redundant and noisy information in the spectral channel, which can create interference. Therefore, it is very challenging to adequately extract discriminative spectral information from the images. In this paper, comparative experiments between PCA and LDA are conducted on the PU dataset and the results are shown in Table 3. It can be concluded from the comparison that PCA is slightly better for categorization than LDA, so PCA is chosen to extract the principal spectral bands.

Table 3. Classification results using PCA and LDA on the PU dataset.

	OA (%)	AA (%)	Kappa (%)
PCA	**97.37 ± 0.35**	**95.58 ± 0.66**	**96.41 ± 0.41**
LDA	97.25 ± 0.65	95.43 ± 1.32	96.34 ± 0.86

To explore the effect of the number of principal components on performance, parametric experiments are conducted on the PU dataset. As shown in Table 4, it can be found that both the network parameters and the running time increase with the increase in number of principal components, and the best classification results are obtained when the number is 100. Therefore, the number of principal components is set to 100.

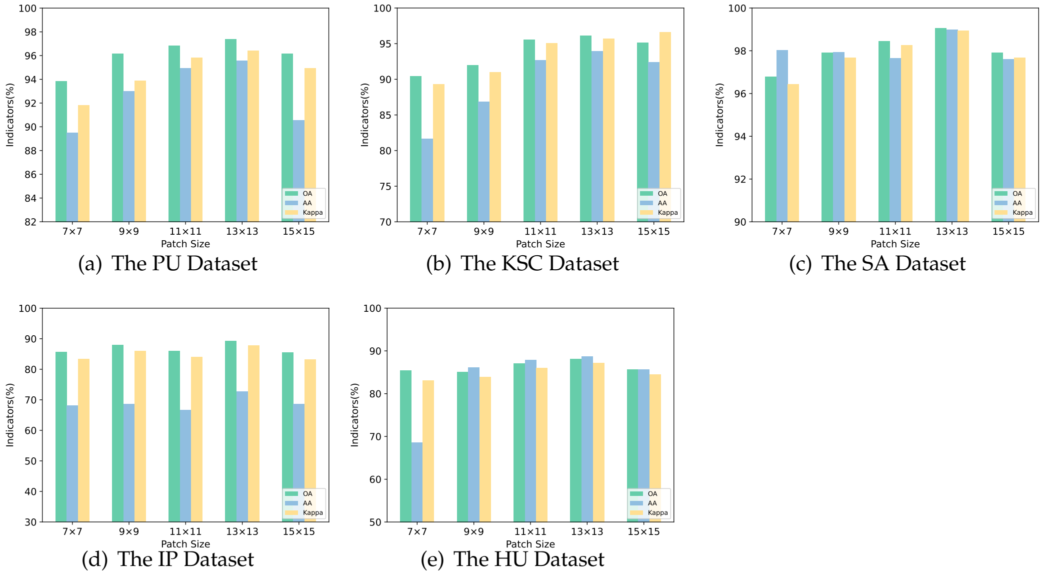
Table 4. Experimental results with different PCA components on the PU dataset.

PCA	OA (%)	AA (%)	Kappa (%)	Train(s)	Params (M)
30	96.51	94.61	95.36	**180.93**	**20.12**
50	96.59	93.78	95.48	183.18	22.13
80	97.20	94.54	96.28	209.68	25.13
100	**97.37**	**95.58**	**96.41**	253.96	27.14

3.3.2. The Influence of Patch Size

Patch size plays a crucial role in determining how well a classification model performs. In order to find the optimal patch size, we conducted a series of experiments within the range of $\{7, 9, 11, 13, 15\}$. Figure 4 displays the effect of parameter k on the classification performance across all three datasets.

The analysis illustrated in Figure 4 demonstrates that a patch size of 13 is the most effective. All evaluation metrics reach their peak values for three datasets when k is set to 13. When the patch size k is small, the image blocks contain less spatial neighborhood information and are vulnerable to noise interference. Conversely, if k is large, there may be an excessive amount of spatial neighborhood information, which disrupts the central pixel features. In order to achieve a balance between performance and efficiency, the value of the patch size k was established as 13.



(a) The PU Dataset (b) The KSC Dataset (c) The SA Dataset

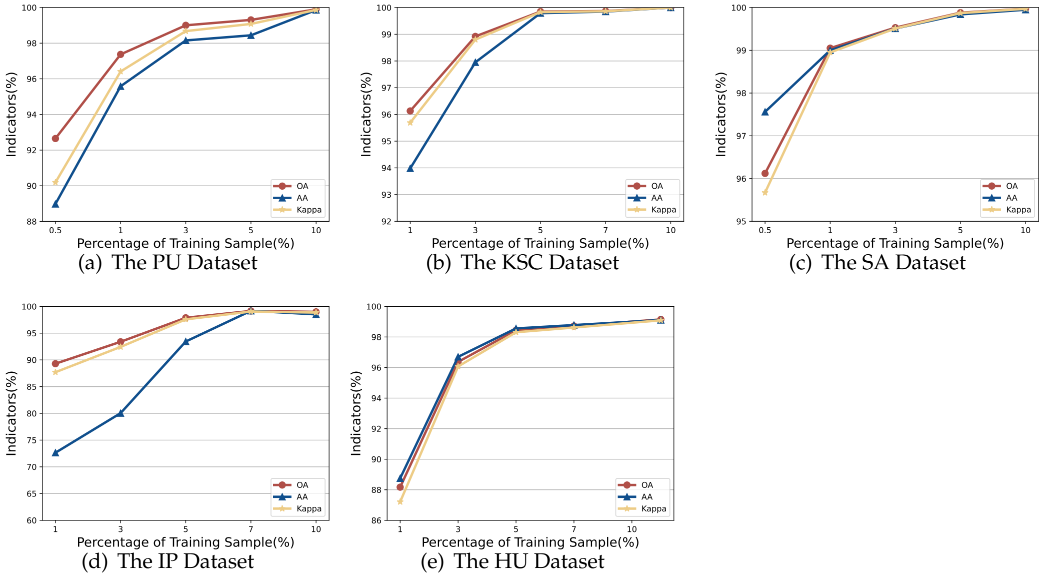
(d) The IP Dataset (e) The HU Dataset

Figure 4. Impact of patch size *k* on the performance of MLFEN.

3.3.3. The Influence of the Number of Training Samples

The performance of the model is greatly influenced by the number of training samples. Therefore, an analysis of the classification results is conducted using different sizes of training sets. To ensure the stability and robustness of the MLFEN method, we randomly selected 0.5%, 1%, 3%, 5%, and 10% labeled samples as the training data for the PU and SA datasets. For the KSC, IP and HU datasets, we selected 1%, 3%, 5%, 7%, and 10% labeled samples as the training data.

In Figure 5, we observe the impact of varying the number of training samples on the performance of the MLFEN approach. Through the experimental findings, it becomes evident that by using a mere 0.5% of training data from the PU dataset, the model can achieve an OA surpassing 92%. Remarkably, this advantage extends further to over 96% on the SA dataset. Similarly, on the KSC dataset, the model reaches an OA metric of more than 96% with the utilization of only 1% of the training samples. These results highlight the robustness of the model in handling inter-class imbalances.



(a) The PU Dataset (b) The KSC Dataset (c) The SA Dataset

(d) The IP Dataset (e) The HU Dataset

Figure 5. Impact of the number of training samples on the performance of MLFEN.

3.3.4. The Influence of Coefficient λ

To examine the impact of the coefficient λ in the proposed sparse loss function on the classification performance, a series of experiments are conducted on the PU dataset. Initially, the performance of the standard L_{ce} is confirmed. Following this, several sets of experiments are designed with varying values of λ. The outcomes are summarized in Table 5. To ensure fairness and scientific rigor in the experimental findings, all parameters except λ remained consistent with those described in Section 3.2.1.

Table 5. Impact of the coefficient λ on the performance of MLFEN.

Indicators	Lce	Loss (λ Value)						
		0.001	**0.005**	**0.01**	**0.02**	**0.05**	**0.1**	**0.2**
OA (%)	95.15	96.46	96.85	**97.37**	96.96	96.81	96.44	95.93
AA (%)	92.35	94.36	94.66	**95.58**	94.70	94.68	93.84	91.97
Kappa (%)	93.56	95.31	95.82	**96.41**	95.97	95.77	95.94	94.59

As shown in Table 5, the lowest classification performance is achieved in the proposed MLFEN using the commonly used L_{ce}. The addition of the L_{sro} regularization term to the loss function results in a higher classification accuracy than L_{ce} alone. This can be explained by the fact that the regularization term prevents overfitting to some extent and is resistant to noise and interference. With the gradual increase of the λ hyperparameter, all three evaluation indicators show an upward trend and reach the maximum value at the value of 0.01 for the λ hyperparameter. When the value of λ is greater than 0.01, OA, AA and Kappa decrease. In view of the above experimental results, the value of λ is set to 0.01.

3.3.5. The Influence of the Depth of EDVT

To investigate the impact of the network depth of the EDVT module on the performance of the model, we conduct quantitative experiments on the PU dataset using networks of varying depths and the results are presented in Table 6.

Table 6. Impact of the number of the depth of EDVT on the performance of MLFEN.

Module	Depth	Indicators				
		OA (%)	**AA (%)**	**Kappa (%)**	**Train (s)**	**Test (s)**
EDVT	1	89.28	81.40	85.16	**146.07**	**5.87**
	2	93.23	87.86	90.92	154.19	6.34
	3	**97.37**	**95.58**	**96.41**	253.96	8.04
	4	96.68	94.67	95.60	261.12	8.52
	5	95.97	92.77	94.64	310.23	9.41
	6	95.69	92.22	94.27	320.74	9.92

From the table, it is evident that the performance of the short-range EDVT is relatively poor, particularly at a depth of 1 where the OA value reaches only 89.28%. This can be attributed to the difficulty in fully capturing and integrating deep global features with only one or two encoders. In contrast, the mid-range EDVT (three or four encoders) tends to yield better results, with the optimal depth parameter being three. However, as the number of network layers increases further, the classification performance experiences some regression. For instance, when there are six layers, the OA decreases to 95.69%.

In addition, we evaluate the impact of EDVT depth on computational complexity by counting the training time and testing time. The results show that the training time and testing time are the shortest when the depth is 1, which are 146.07 s and 5.87 s, respectively. The computational complexity increases gradually as the network depth increases. It can be seen that shortening the depth as much as possible can help reduce the computational complexity while ensuring high classification accuracy.

It is important to highlight that the classical ViT utilizes six encoders. However, our experiments have shown that the model can achieve good accuracy with only three encoders. This relatively shallow network design helps prevent excessive information loss and showcases the effectiveness of the EDVT module in facilitating deep feature learning.

3.4. Ablation Study

To validate the effectiveness of the band shift module, we performed ablation experiments on the PU dataset. Table 7 shows the mean values of the results of ten experiments. From these findings, we can observe that OA, AA, Kappa are improved by 1.41%, 2.76% and 1.78%, respectively, which proves that the band shift module has a positive effect on the classification results.

Table 7. Ablation study of Band Shift on the PU dataset. Optimal results are shown in bold.

Cases	OA	AA	Kappa
without Band Shift	95.96	92.82	94.63
with Band Shift	**97.37**	**95.58**	**96.41**

This paper introduces the MLFEN method and its main technical contributions. These include the HCAM module, which models and enhances the 3D spatial–spectral joint features, the EDVT module for mining deep spectral features, and the sparse loss function for reducing the fitting variance. In this section, our research goal is to analyze the impact of these structures on the PU dataset both qualitatively and quantitatively. To achieve this, we conduct several ablation experiments to test the applicability and effectiveness of these structures in MLFEN for HSI classification. The specific classification results for different components are presented in Table 8.

Table 8. Ablation study of the proposed MLFEN with a combination of different components on the PU dataset. Optimal results are shown in bold.

Cases	**Components**			**Indicators**		
	HCAM	**EDVT**	**Loss**	**OA (%)**	**AA (%)**	**Kappa (%)**
1	×	×	×	90.25	80.66	86.93
2	×	✓	✓	91.29	84.38	88.36
3	✓	×	✓	95.70	91.66	94.28
4	✓	✓	×	95.15	92.35	93.56
5	✓	✓	✓	**97.37**	**95.58**	**96.41**

Table 8 demonstrate the influence of different components on the performance of MLFEN. When all three components are utilized (Case 5), the best classification results are obtained on the PU dataset. Conversely, when none of these components are employed (case 1), the worst performance is observed. Removing HCAM (Case 2) leads to a significant decrease in accuracy, with OA, AA, and Kappa decreasing by 6.08%, 11.2%, and 8.05%, respectively. This highlights the importance of extracting local joint spatial–spectral features. Similarly, removing EDVT (Case 3) results in a decrease in all three evaluation indicators, indicating that the improved EDVT model is more advantageous for HSI classification tasks compared to the original ViT model. Furthermore, it is worth noting that neither EDVT without HCAM assistance nor HCAM without EDVT support can achieve excellent performance. This suggests that HCAM and EDVT mutually enhance each other, enabling the fusion of shallow and deep features. Lastly, the inclusion of the loss function (Case 4) leads to improvements in the classification metrics OA, AA, and Kappa by 2.22%, 3.23%, and 2.85%, respectively. Despite its simplicity, the design of the sparse loss function helps mitigate overfitting and enhances the accuracy of the classification results.

3.5. Comparative Experiments

Extensive experiments were conducted on the PU, KSC, SA, IP and HU datasets to analyze and evaluate the performance of various methods. The comparison experiments include refining the evaluation of representative methods such as 2D-CNN [30], 3D-CNN [31], HybridSN [32], SSAN [42], MAFN [43], ViT [46], HiT [56], CVSSN [39], and SSFTT [53].

Regarding 2D-CNN, the network structure comprises four 2D convolutional blocks and an average pooling layer. The initial two 2D convolutional blocks consist of a 2D convolutional layer, a batch normalization layer, and a ReLU activation function layer. Meanwhile, the last two 2D convolutional blocks incorporate a dropout layer in addition to the aforementioned three layers. Each of the four convolutional kernels has a size of 3 × 3, with corresponding quantities of 30, 32, 64, and 128, respectively.

Concerning 3D-CNN, the model consists of three 3D convolutional blocks and an average pooling layer. Similar to the 2D-CNN, the first 3D convolutional block includes a 3D convolutional layer, a batch normalization layer, and a ReLU activation function layer, whereas the last two 3D convolutional blocks encompass a dropout layer in addition to the aforementioned three layers. The number of 3D convolutional kernels are 32, 64, and 128, and their size is 3 × 3 × 3.

For the other seven methods, including HybridSN, SSAN, MAFN, ViT, HiT, CVSSN and SSFTT, the networks follow the setup described in respective references.

For the proposed MLFEN method, the number of components after PCA dimensionality reduction is set to 100. The patch size after the data-processing stage is set to 13 × 13. Within the EDVT module, a 2D convolution is performed using a convolution kernel size of 3 × 3, while a 3D convolution is performed using a convolution kernel size of 3 × 3 × 3. The HCVT module utilizes three encoders.

3.5.1. Quantitative Evaluation

Tables 9–13 provide the comparative results of MLFEN and other state-of-the-art methods on the PU, KSC, SA, IP and HU datasets, respectively. Based on the evaluation metrics of OA, AA, and Kappa, the MLFEN approach proposed in this study displays superior performance compared to other models for all five datasets.

Table 9. The mean and standard deviation of the ten results obtained by different methods on the PU dataset. Optimal results are shown in bold.

No.	2D-CNN	3D-CNN	HybridSN	SSAN	MAFN	ViT	HiT	CVSSN	SSFTT	MLFEN
1	94.66 ± 2.06	93.62 ± 0.77	93.80 ± 1.92	94.14 ± 3.01	**98.06** ± **3.20**	85.92 ± 0.90	89.14 ± 3.22	94.89 ± 1.80	93.68 ± 1.97	97.10 ± 0.80
2	99.11 ± 0.49	98.30 ± 0.39	98.80 ± 0.45	99.24 ± 0.25	99.07 ± 0.93	98.30 ± 2.22	97.43 ± 1.50	98.16 ± 0.76	99.86 ± 0.08	**99.88** ± **0.11**
3	69.86 ± 6.04	58.56 ± 2.82	65.48 ± 9.85	75.84 ± 8.78	78.10 ± 6.96	78.58 ± 0.10	74.18 ± 8.47	87.97 ± 7.48	85.62 ± 4.17	**88.21** ± **3.16**
4	86.12 ± 6.08	92.67 ± 1.70	91.53 ± 1.27	88.79 ± 4.42	95.79 ± 3.53	77.84 ± 1.49	86.82 ± 2.29	**96.86** ± **2.26**	89.70 ± 1.31	91.96 ± 1.89
5	**99.92** ± **0.16**	99.71 ± 0.28	99.79 ± 0.23	97.16 ± 5.07	98.63 ± 0.51	98.27 ± 0.48	99.14 ± 0.68	97.97 ± 2.61	99.76 ± 0.22	99.74 ± 0.32
6	88.86 ± 2.55	74.23 ± 2.76	84.94 ± 5.79	93.14 ± 4.73	97.39 ± 1.34	97.79 ± 1.13	79.78 ± 2.99	95.93 ± 2.28	98.56 ± 0.91	**99.77** ± **0.22**
7	70.50 ± 9.33	74.07 ± 4.17	80.84 ± 5.41	91.52 ± 4.50	94.05 ± 5.89	85.94 ± 0.88	78.76 ± 7.20	91.98 ± 6.74	97.43 ± 1.77	**98.02** ± **2.43**
8	88.96 ± 4.76	88.52 ± 4.20	89.46 ± 5.57	**93.44** ± **0.91**	91.28 ± 3.44	71.48 ± 9.21	85.97 ± 4.73	90.11 ± 4.06	88.25 ± 2.42	89.04 ± 2.33
9	99.48 ± 0.49	**99.93** ± **0.09**	99.46 ± 0.50	98.30 ± 1.97	99.93 ± 0.52	65.27 ± 0.71	98.69 ± 1.15	97.84 ± 1.90	87.98 ± 3.42	95.88 ± 3.30
OA (%)	92.68 ± 1.35	90.30 ± 0.68	92.93 ± 1.11	95.25 ± 0.69	96.33 ± 1.54	89.26 ± 0.32	90.59 ± 0.47	95.86 ± 0.83	95.98 ± 0.51	**97.37** ± **0.35**
AA (%)	87.61 ± 3.33	85.33 ± 1.81	89.34 ± 2.00	92.62 ± 0.76	94.84 ± 0.64	85.13 ± 1.45	87.48 ± 1.03	94.63 ± 1.29	93.43 ± 0.96	**95.58** ± **0.66**
Kappa (%)	90.21 ± 1.82	86.98 ± 0.92	90.54 ± 1.51	93.68 ± 0.93	95.73 ± 0.98	87.45 ± 0.93	87.88 ± 0.53	94.51 ± 1.09	94.66 ± 0.68	**96.41** ± **0.41**

Table 10. The mean and standard deviation of the ten results obtained by different methods on the KSC dataset. Optimal results are shown in bold.

No.	**2D-CNN**	**3D-CNN**	**HybridSN**	**SSAN**	**MAFN**	**ViT**	**HiT**	**CVSSN**	**SSFTT**	**MLFEN**
1	95.85 ± 2.59	95.84 ± 0.70	97.92 ± 0.73	90.75 ± 4.86	96.56 ± 2.97	96.94 ± 4.67	95.04 ± 3.86	93.37 ± 4.08	**99.80 ± 0.22**	97.42 ± 2.51
2	71.31 ± 7.57	56.12 ± 4.90	64.52 ± 9.92	59.62 ± 7.69	61.44 ± 9.53	90.30 ± 5.57	**92.25 ± 7.32**	62.89 ± 16.14	66.27 ± 4.19	66.92 ± 3.86
3	92.91 ± 5.10	58.40 ± 9.72	89.16 ± 9.77	89.60 ± 6.92	77.91 ± 9.24	62.38 ± 5.77	83.27 ± 9.92	78.36 ± 13.58	**100.00 ± 0.00**	99.16 ± 1.28
4	19.57 ± 6.56	21.95 ± 0.60	25.89 ± 6.09	28.26 ± 9.91	39.53 ± 7.90	37.92 ± 9.37	41.06 ± 7.89	54.51 ± 9.01	86.23 ± 8.45	**91.35 ± 7.55**
5	14.62 ± 4.89	16.37 ± 2.57	15.80 ± 8.25	47.02 ± 9.42	38.50 ± 5.90	44.06 ± 7.43	41.76 ± 2.67	52.04 ± 21.45	**81.76 ± 0.00**	79.87 ± 4.62
6	67.62 ± 9.02	52.47 ± 1.86	71.97 ± 7.64	50.31 ± 9.89	33.19 ± 3.13	38.36 ± 6.56	51.86 ± 7.31	79.53 ± 15.25	60.00 ± 11.51	**95.83 ± 5.20**
7	85.74 ± 7.93	94.06 ± 8.77	88.82 ± 8.73	85.80 ± 6.72	78.60 ± 8.03	55.90 ± 3.60	56.00 ± 5.70	79.52 ± 19.96	**100.00 ± 0.00**	**100.00 ± 0.00**
8	84.23 ± 8.79	47.27 ± 4.03	80.41 ± 9.11	78.47 ± 5.87	81.79 ± 7.97	80.79 ± 9.33	84.40 ± 8.07	59.45 ± 8.98	97.92 ± 1.61	**99.66 ± 0.51**
9	86.73 ± 2.14	86.22 ± 1.37	87.54 ± 1.19	73.92 ± 8.96	90.87 ± 8.55	76.62 ± 9.21	87.92 ± 5.63	86.78 ± 8.84	86.54 ± 3.45	**94.67 ± 4.65**
10	91.35 ± 2.86	84.30 ± 2.35	85.19 ± 5.22	70.54 ± 9.30	96.50 ± 4.15	96.76 ± 4.12	92.75 ± 8.28	71.17 ± 17.22	99.85 ± 0.32	**99.92 ± 0.16**
11	99.86 ± 0.26	99.27 ± 1.18	100.00 ± 0.00	98.09 ± 2.56	90.30 ± 6.45	92.74 ± 7.05	95.31 ± 4.97	95.21 ± 5.83	**100.00 ± 0.00**	**100.00 ± 0.00**
12	96.98 ± 1.07	93.50 ± 1.02	93.23 ± 4.30	80.62 ± 9.72	91.50 ± 7.58	78.47 ± 8.26	83.90 ± 7.95	89.64 ± 7.51	98.43 ± 1.44	**99.98 ± 0.06**
13	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	99.49 ± 1.02	99.53 ± 1.47	99.98 ± 0.07	99.92 ± 0.25	99.07 ± 1.90	**100.00 ± 0.00**	**100.00 ± 0.00**
OA (%)	85.29 ± 2.24	78.81 ± 0.60	84.31 ± 2.42	77.82 ± 2.66	82.65 ± 5.22	79.29 ± 4.87	81.63 ± 4.66	81.39 ± 2.82	93.73 ± 0.80	**96.13 ± 0.75**
AA (%)	77.65 ± 3.11	68.90 ± 0.55	75.60 ± 3.35	69.88 ± 3.38	72.66 ± 6.53	68.17 ± 7.89	72.26 ± 8.38	77.04 ± 3.54	90.52 ± 1.13	**93.98 ± 1.03**
Kappa (%)	83.65 ± 2.47	76.36 ± 0.67	82.52 ± 2.70	75.30 ± 2.98	80.48 ± 5.98	76.58 ± 5.67	79.28 ± 5.39	79.26 ± 3.15	93.01 ± 0.89	**95.69 ± 0.84**

Table 11. The mean and standard deviation of the ten results obtained by different methods on the SA dataset. Optimal results are shown in bold.

No.	**2D-CNN**	**3D-CNN**	**HybridSN**	**SSAN**	**MAFN**	**ViT**	**HiT**	**CVSSN**	**SSFTT**	**MLFEN**
1	98.59 ± 2.45	97.31 ± 2.68	97.41 ± 3.13	**99.97 ± 0.02**	99.56 ± 0.91	99.27 ± 1.59	97.53 ± 2.94	99.19 ± 1.14	99.82 ± 0.30	99.96 ± 0.05
2	99.91 ± 0.18	99.65 ± 0.64	99.91 ± 0.08	99.40 ± 1.04	**100.00 ± 0.00**	99.89 ± 0.24	99.74 ± 0.54	99.98 ± 0.04	99.99 ± 0.04	99.99 ± 0.02
3	99.49 ± 0.17	97.82 ± 0.99	97.43 ± 3.51	96.17 ± 1.98	99.97 ± 0.08	99.24 ± 1.66	97.04 ± 4.85	96.86 ± 1.58	**100.00 ± 0.00**	99.98 ± 0.03
4	99.60 ± 0.27	99.59 ± 0.12	97.31 ± 4.48	96.50 ± 3.46	99.78 ± 0.33	85.76 ± 8.91	88.04 ± 8.04	95.87 ± 2.66	**99.88 ± 0.12**	99.11 ± 1.44
5	97.58 ± 0.45	90.58 ± 2.81	94.22 ± 3.96	92.66 ± 2.58	95.94 ± 5.93	97.92 ± 2.19	94.52 ± 3.43	98.12 ± 1.63	98.65 ± 0.31	**99.38 ± 0.40**
6	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	99.96 ± 0.07	99.98 ± 0.03	99.86 ± 0.40	98.37 ± 2.84	99.97 ± 0.07	99.74 ± 0.73	99.90 ± 0.17
7	98.85 ± 0.80	99.58 ± 0.19	99.24 ± 0.98	99.30 ± 0.42	99.86 ± 0.07	99.73 ± 0.35	98.96 ± 1.75	99.29 ± 0.86	99.96 ± 0.09	**99.96 ± 0.04**
8	85.21 ± 2.42	79.57 ± 1.99	84.01 ± 3.04	84.23 ± 2.20	96.38 ± 1.96	92.58 ± 3.87	88.93 ± 7.55	92.92 ± 1.76	98.93 ± 0.74	**99.19 ± 0.66**
9	99.94 ± 0.05	99.91 ± 0.10	99.97 ± 0.03	98.64 ± 2.36	99.99 ± 0.02	99.88 ± 0.30	99.77 ± 0.47	99.67 ± 0.34	**100.00 ± 0.00**	**100.00 ± 0.00**
10	95.78 ± 0.75	90.96 ± 1.87	92.10 ± 1.99	87.90 ± 3.00	98.82 ± 0.51	97.79 ± 2.17	91.94 ± 7.53	95.76 ± 2.13	98.63 ± 0.55	**99.45 ± 0.35**
11	84.23 ± 2.38	84.42 ± 0.64	86.56 ± 2.52	89.77 ± 4.19	98.87 ± 0.83	99.11 ± 1.13	96.83 ± 3.39	94.54 ± 5.67	**99.90 ± 0.09**	98.81 ± 1.95
12	99.81 ± 0.11	**99.96 ± 0.04**	99.84 ± 0.12	98.64 ± 0.97	99.83 ± 0.47	95.97 ± 4.06	94.34 ± 7.19	99.66 ± 0.50	99.92 ± 0.08	99.94 ± 0.10
13	98.44 ± 0.60	97.28 ± 0.54	93.44 ± 5.94	94.95 ± 3.63	93.25 ± 4.61	92.13 ± 8.78	95.95 ± 3.79	**99.18 ± 1.27**	92.18 ± 3.22	93.08 ± 4.22
14	98.13 ± 0.75	97.80 ± 0.50	98.85 ± 0.51	97.35 ± 0.58	98.44 ± 0.70	90.09 ± 8.79	75.74 ± 8.17	98.20 ± 1.45	**99.32 ± 0.29**	99.23 ± 0.65
15	70.37 ± 3.86	71.29 ± 3.52	79.70 ± 2.71	82.12 ± 3.13	89.12 ± 4.18	66.40 ± 6.77	78.49 ± 5.58	89.59 ± 2.68	94.42 ± 1.50	**96.15 ± 2.21**
16	96.92 ± 0.74	96.67 ± 1.32	96.70 ± 1.21	95.22 ± 4.05	99.42 ± 0.23	98.69 ± 0.58	94.38 ± 2.47	98.68 ± 1.44	99.76 ± 0.35	**99.88 ± 0.16**
OA (%)	91.91 ± 0.20	90.11 ± 0.36	92.34 ± 0.41	92.20 ± 0.72	97.28 ± 0.67	89.80 ± 2.11	88.94 ± 3.11	96.17 ± 0.19	98.69 ± 0.26	**99.05 ± 0.45**
AA (%)	95.18 ± 0.23	93.90 ± 0.35	94.79 ± 0.53	94.55 ± 1.01	98.08 ± 0.42	91.78 ± 1.84	89.44 ± 3.94	97.34 ± 0.45	98.82 ± 0.29	**99.00 ± 0.39**
Kappa (%)	90.99 ± 0.22	88.99 ± 0.40	91.48 ± 0.46	91.31 ± 0.80	96.97 ± 0.75	88.60 ± 2.39	88.23 ± 3.33	95.73 ± 0.21	98.54 ± 0.29	**98.95 ± 0.50**

Table 12. The mean and standard deviation of the ten results obtained by different methods on the IP dataset. Optimal results are shown in bold.

No.	2D-CNN	3D-CNN	HybridSN	SSAN	MAFN	ViT	HiT	CVSSN	SSFTT	MLFEN
1	1.34 ± 1.17	0.91 ± 2.20	0.23 ± 0.72	0.46 ± 1.44	15.00 ± 19.47	2.91 ± 1.12	1.70 ± 1.26	47.41 ± 5.55	**44.67 ± 29.72**	6.67 ± 1.93
2	71.80 ± 1.73	40.48 ± 8.31	49.56 ± 6.43	41.36 ± 4.75	81.21 ± 4.66	51.50 ± 2.73	66.00 ± 11.17	59.64 ± 0.97	68.80 ± 3.50	**86.59 ± 4.96**
3	29.88 ± 7.38	23.22 ± 2.20	44.42 ± 3.33	13.69 ± 7.03	74.56 ± 1.82	34.71 ± 4.15	5.67 ± 4.34	53.49 ± 9.51	**94.96 ± 1.73**	90.74 ± 5.10
4	4.32 ± 8.77	6.08 ± 8.84	16.87 ± 5.86	26.43 ± 1.78	34.74 ± 5.65	20.33 ± 12.19	40.18 ± 14.34	62.68 ± 5.64	46.00 ± 6.41	**85.58 ± 2.62**
5	18.12 ± 3.73	25.21 ± 3.04	49.46 ± 5.87	36.37 ± 3.96	87.82 ± 1.69	30.66 ± 6.16	81.73 ± 3.02	62.11 ± 7.84	75.44 ± 1.29	**96.82 ± 3.82**
6	97.74 ± 1.52	92.59 ± 3.65	92.29 ± 3.9	90.16 ± 8.63	95.14 ± 2.44	85.20 ± 1.38	91.25 ± 2.71	80.68 ± 6.39	95.64 ± 1.44	**98.45 ± 1.40**
7	0.00 ± 0.00	0.00 ± 0.00	0.77 ± 2.43	1.83 ± 5.37	1.72 ± 2.43	17.95 ± 5.39	3.82 ± 3.23	**25.89 ± 2.46**	0.00 ± 0.00	0.00 ± 0.00
8	98.38 ± 0.81	95.07 ± 3.00	93.97 ± 8.28	87.45 ± 7.38	97.53 ± 1.82	80.95 ± 14.93	93.16 ± 4.37	97.90 ± 3.88	**99.49 ± 0.42**	97.49 ± 0.42
9	0.00 ± 0.00	1.67 ± 5.27	2.94 ± 1.73	3.05 ± 1.95	1.30 ± 1.84	7.41 ± 6.93	1.85 ± 0.62	19.20 ± 9.80	0.00 ± 0.00	**21.52 ± 1.08**
10	40.28 ± 8.46	22.78 ± 7.21	61.99 ± 2.90	36.83 ± 6.75	65.30 ± 16.17	61.20 ± 3.57	49.94 ± 14.19	64.58 ± 0.13	75.17 ± 1.93	**87.85 ± 3.69**
11	69.16 ± 6.36	83.38 ± 8.53	72.62 ± 5.00	85.34 ± 3.06	81.70 ± 9.51	74.16 ± 6.79	71.38 ± 8.02	75.06 ± 5.27	92.67 ± 1.83	**95.21 ± 1.22**
12	23.34 ± 1.53	25.12 ± 7.36	40.51 ± 3.13	16.92 ± 2.16	68.35 ± 9.51	21.67 ± 12.01	10.18 ± 6.70	41.28 ± 5.23	63.05 ± 9.53	**75.88 ± 4.21**
13	73.62 ± 7.6	59.59 ± 2.65	79.59 ± 9.82	80.20 ± 8.36	98.19 ± 0.68	63.78 ± 5.24	98.28 ± 1.28	76.18 ± 3.86	86.75 ± 6.32	**98.90 ± 1.05**
14	96.78 ± 4.84	96.05 ± 4.10	94.41 ± 6.90	93.75 ± 3.87	90.66 ± 7.59	90.53 ± 9.53	81.17 ± 7.81	86.10 ± 5.52	**99.32 ± 0.61**	99.17 ± 0.68
15	23.73 ± 7.94	40.94 ± 3.14	30.48 ± 2.33	28.60 ± 4.78	65.13 ± 5.94	31.22 ± 9.68	43.53 ± 9.47	63.29 ± 9.04	90.10 ± 2.59	**92.39 ± 4.01**
16	36.07 ± 4.50	53.37 ± 9.22	57.55 ± 4.65	20.45 ± 4.88	**87.95 ± 7.81**	76.76 ± 3.45	77.65 ± 7.72	80.44 ± 0.65	46.20 ± 6.96	86.81 ± 6.80
OA (%)	61.32 ± 2.52	58.56 ± 2.36	64.68 ± 3.38	59.07 ± 2.98	80.19 ± 1.64	60.98 ± 2.19	62.05 ± 3.74	67.26 ± 2.59	84.33 ± 1.20	**89.29 ± 2.48**
AA (%)	42.70 ± 4.44	41.65 ± 4.32	48.69 ± 6.09	41.13 ± 4.03	65.38 ± 2.44	46.75 ± 5.33	50.75 ± 2.75	62.25 ± 2.70	67.39 ± 2.35	**72.64 ± 6.31**
Kappa (%)	55.41 ± 2.69	51.34 ± 3.07	59.50 ± 3.96	52.04 ± 3.73	77.31 ± 1.85	55.06 ± 2.54	56.10 ± 4.50	62.70 ± 2.90	82.06 ± 1.37	**87.70 ± 2.88**

Table 13. The mean and standard deviation of the ten results obtained by different methods on the HU dataset. Optimal results are shown in bold.

No.	2D-CNN	3D-CNN	HybridSN	SSAN	MAFN	ViT	HiT	CVSSN	SSFTT	MLFEN
1	94.50 ± 4.70	**98.96 ± 1.17**	98.80 ± 1.55	92.75 ± 5.32	91.93 ± 1.17	89.86 ± 5.51	91.67 ± 6.30	89.15 ± 6.94	95.70 ± 1.49	93.34 ± 1.52
2	83.20 ± 3.86	84.66 ± 2.89	86.21 ± 3.64	83.79 ± 4.58	90.99 ± 5.65	79.72 ± 1.15	**93.83 ± 6.41**	92.33 ± 5.06	91.39 ± 3.64	92.52 ± 2.98
3	71.40 ± 21.16	91.34 ± 1.84	91.49 ± 5.20	95.12 ± 5.35	97.16 ± 1.37	94.06 ± 7.61	**98.80 ± 1.78**	94.22 ± 5.86	96.02 ± 0.89	93.24 ± 2.36
4	**92.50 ± 3.72**	91.38 ± 2.84	87.36 ± 8.99	91.81 ± 1.58	78.56 ± 5.09	73.25 ± 1.49	90.70 ± 6.23	86.62 ± 12.28	86.57 ± 2.06	84.44 ± 6.53
5	**100.00 ± 0.00**	98.76 ± 1.22	99.60 ± 0.57	94.63 ± 5.51	99.72 ± 0.57	97.76 ± 5.89	98.19 ± 2.20	93.12 ± 2.27	**100.00 ± 0.00**	**100.00 ± 0.00**
6	67.07 ± 5.71	69.06 ± 6.05	70.12 ± 3.69	62.39 ± 14.03	80.92 ± 3.17	72.63 ± 6.92	72.26 ± 12.88	83.49 ± 17.15	81.76 ± 2.59	**83.57 ± 3.44**
7	65.33 ± 8.03	58.87 ± 3.06	67.84 ± 7.80	54.45 ± 3.01	69.08 ± 9.78	73.94 ± 4.28	**84.37 ± 8.13**	82.26 ± 4.26	75.28 ± 2.10	78.62 ± 3.86
8	59.71 ± 4.86	56.35 ± 3.31	71.61 ± 6.01	58.42 ± 9.90	74.15 ± 5.46	64.46 ± 2.31	61.49 ± 11.88	76.77 ± 7.58	**79.70 ± 6.62**	77.27 ± 1.52
9	73.75 ± 6.08	77.38 ± 4.04	58.80 ± 9.29	77.86 ± 5.09	67.99 ± 13.68	52.99 ± 8.57	76.88 ± 10.79	**82.33 ± 6.56**	79.64 ± 3.53	79.80 ± 14.88
10	43.30 ± 10.46	68.96 ± 6.93	78.29 ± 5.69	78.17 ± 13.18	88.37 ± 6.62	71.73 ± 4.15	68.60 ± 14.45	73.30 ± 7.66	88.72 ± 4.42	**93.33 ± 4.96**
11	70.04 ± 9.62	71.38 ± 8.66	77.17 ± 8.21	72.25 ± 5.07	75.56 ± 10.25	68.79 ± 1.18	75.19 ± 9.15	79.90 ± 8.38	76.67 ± 5.67	**85.65 ± 7.11**
12	69.34 ± 10.86	65.57 ± 8.99	80.37 ± 10.07	60.05 ± 11.81	80.91 ± 4.40	73.30 ± 1.94	71.46 ± 12.75	77.23 ± 8.31	83.38 ± 3.63	**89.13 ± 5.87**
13	67.56 ± 7.65	**91.26 ± 1.55**	63.09 ± 8.55	71.83 ± 8.06	74.58 ± 11.49	42.66 ± 5.33	58.15 ± 15.62	87.66 ± 7.88	90.22 ± 3.32	80.72 ± 7.77
14	62.93 ± 19.29	91.58 ± 3.06	86.07 ± 10.24	81.42 ± 13.39	99.63 ± 0.43	97.07 ± 2.23	91.34 ± 18.01	74.98 ± 5.29	**100.00 ± 0.00**	99.67 ± 0.49
15	98.91 ± 0.61	98.77 ± 0.73	98.39 ± 1.46	92.57 ± 2.82	99.91 ± 0.18	97.93 ± 3.45	98.95 ± 1.33	89.05 ± 7.43	**100.00 ± 0.00**	99.98 ± 0.05
OA (%)	74.89 ± 1.18	79.29 ± 1.50	80.81 ± 1.73	77.40 ± 1.86	83.25 ± 2.79	75.23 ± 3.45	81.93 ± 2.24	83.04 ± 2.20	87.09 ± 1.03	**88.17 ± 1.81**
AA (%)	74.64 ± 1.84	80.95 ± 1.40	81.01 ± 1.69	77.83 ± 2.06	84.63 ± 2.51	76.01 ± 2.99	82.13 ± 2.79	84.16 ± 2.09	88.34 ± 0.88	**88.75 ± 1.80**
Kappa (%)	72.82 ± 1.28	77.62 ± 1.62	79.25 ± 1.87	75.58 ± 2.01	81.90 ± 3.02	73.30 ± 3.65	80.45 ± 2.43	81.67 ± 2.39	86.05 ± 1.11	**87.21 ± 1.96**

The PU dataset contains fewer categories and is mainly used to validate the model's classification performance in the presence of multi-scale coarse-graining. The OA of all methods except ViT exceeds 90%, among which the OA of MAFN method reaches 96.33%, indicating that this dataset is not too difficult to classify. However, MLFEN is the most superior in all three classification metrics, with an OA improvement of 1.04% over MAFN. This could potentially be attributed to the fact that although MAFN applies the attention mechanism for spatial and spectral information, respectively, it ignores the link between them. In contrast, MLFEN considers the correlation between spatial and spectral features through its HCAM module, resulting in higher classification accuracy.

The number of samples used for training on the KSC dataset is only 50, making it suitable for evaluating the model's classification performance with limited data. Unfortunately, the classification performance of convolution-based 2D-CNN, 3D-CNN, HybridSN, CVSSN methods, attention-mechanism-based SSAN, MAFN methods, and transformer-based ViT, HiT, SSFTT methods need to be improved. However, the MLFEN method exhibited absolute dominance, achieving the best accuracy in terms of OA, AA, Kappa, and 9 out of the 13 classes. The OA of MLFEN outperformed the other methods by margins ranging from 2.4% to 18.31%. These results indicate that MLFEN demonstrates excellent performance even with limited sample data.

The SA dataset contains 16 classes with a large disparity in the number of samples between classes. The primary purpose of using this dataset is to evaluate how well the model can address these class imbalances. However, both the transformer-based ViT and HiT methods performed poorly on the SA dataset, suggesting that they struggle to overcome the effects of class imbalances. Another transformer-based network, SSFTT, performs slightly worse for some of the classes even though it performs well on five classes. The proposed MLFEN achieves remarkably high performance on various evaluation metrics. Specifically, it achieved 99.05% OA, 99.00% AA, and 98.95% Kappa. These results clearly demonstrate that the proposed method successfully addresses the differences between classes and significantly improves classification performance.

The IP dataset contains up to 16 categories where there is a significant imbalance in the distribution of samples between categories. The main objective of utilising this dataset is to assess the ability of the models to handle these category imbalances. Strikingly, traditional transformer-based models, such as ViT and HiT, perform poorly on the IP dataset, implying that they struggle to overcome the challenges posed by category imbalances. In addition, another transformer-based network, SSFTT, performs slightly weakly for some categories, although it performs well in handling four categories. In contrast, the proposed MLFEN demonstrates significant efficacy, achieving excellent results on various evaluation metrics. Of particular note, it achieved an impressive 89.29% OA, 72.64% AA and 87.70% Kappa. These results highlight that the method successfully copes with the differences between categories, improving classification performance on the IP dataset.

After quantitative analysis of the classification result of HU, our model achieved 88.17% accuracy across the entire dataset, demonstrating classification accuracy for multiple categories. Specifically, for a specific category "Fallow_smooth", we observe that the model has a precision of 100.00%. Similarly, MLFEN achieves 88.75% for AA and 87.21% for Kappa, the highest among the selected state-of-the-art methods.

3.5.2. Visual Analysis

The classification maps can vividly visualize the difference in effect between the proposed method MLFEN and other comparative methods. Therefore, we show in Figures 6–10 the classification effects of MLFEN and nine different methods on the five datasets, respectively.

In general, the classification maps obtained by the MLFEN model best fit the ground truth maps and are the clearest. Specifically, as shown in Figure 6, the category that is harder to distinguish is the blue region located in the middle of the image. The classification results of most methods for this region are adulterated with many yellow pixels, and the MAFN and the proposed MLFEN methods are the most accurate for this region. However, for the yellow region at the bottom of the image, the MAFN method incorrectly classifies many pixels as blue, whereas the MLFEN method recognizes the yellow color to a greater extent. These two regions on the image of the PU dataset confirm the superior performance of our method. Moreover, MLFEN successfully mitigated the problem of the category "Self-Blocking Bricks" being misclassified as other classes and other classes being misclassified as "Self-Blocking Bricks" that occurs in SSFTT.

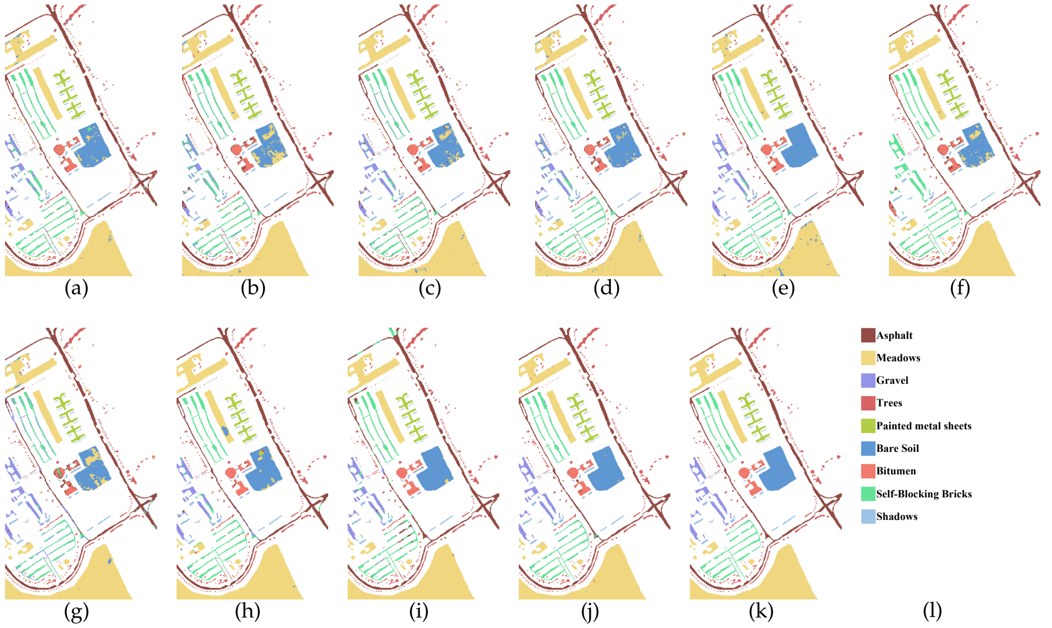


Figure 6. Classification maps on the PU dataset. (**a**) 2D-CNN. (**b**) 3D-CNN. (**c**) HybridSN. (**d**) SSAN. (**e**) MAFN. (**f**) ViT. (**g**) HiT. (**h**) CVSSN. (**i**) SSFTT. (**j**) MLFEN (Ours). (**k**) Ground truth. (**l**) Color bar.

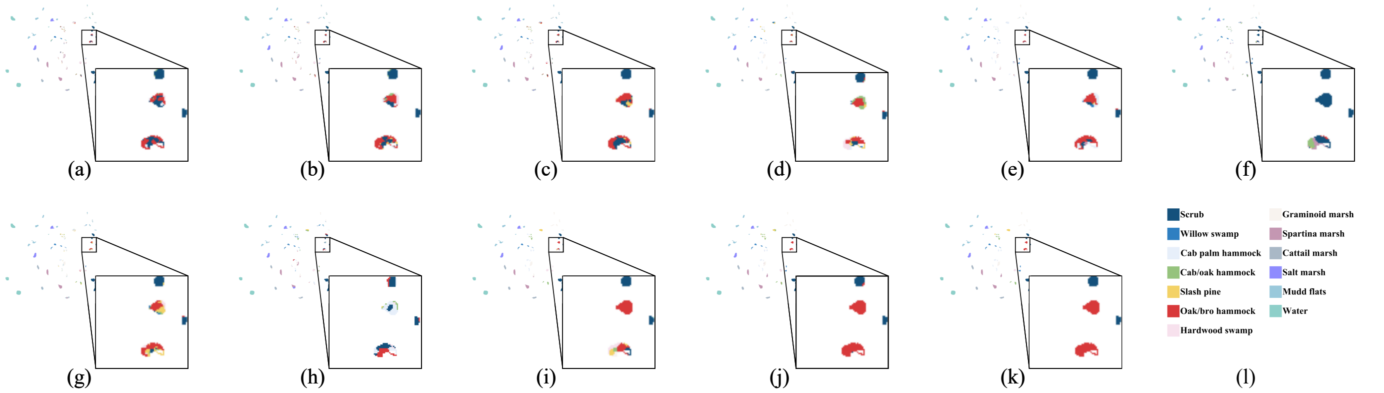


Figure 7. Classification maps on the KSC dataset. (**a**) 2D-CNN. (**b**) 3D-CNN. (**c**) HybridSN. (**d**) SSAN. (**e**) MAFN. (**f**) ViT. (**g**) HiT. (**h**) CVSSN. (**i**) SSFTT. (**j**) MLFEN (Ours). (**k**) Ground truth. (**l**) Color bar.

Furthermore, by observing and comparing the inter-class boundary regions presented in Figure 7, it can be observed that the MLFEN classification map has the clearest boundaries. Meanwhile, we chose a zoomed-in region of interest (ROI) to further highlight the classification performance of different models more visually. The ROI can be analysed to intuitively see that the classification effect of our method is at its best.

In addition, on the SA dataset, for the category "Broccoli-green-weeds-2" in blue and the category "Corn-senesced-green-weeds" in gray, the comparison methods contain significantly more noise, indicating that these methods are unable to accurately identify feature classes. The MLFEN method, on the other hand, has less noise on category "Broccoli-green-weeds-2", while the category in gray is clean. In the "Grapes_untrained" class (light grey area in Figure 8), some misclassification pixels appears, which indicates that the MLFEN method is slightly less effective than SSFTT in this class. However, SSFTT exhibits inadequacies in the blue area, which allows MLFEN to achieve superior results in comparison.

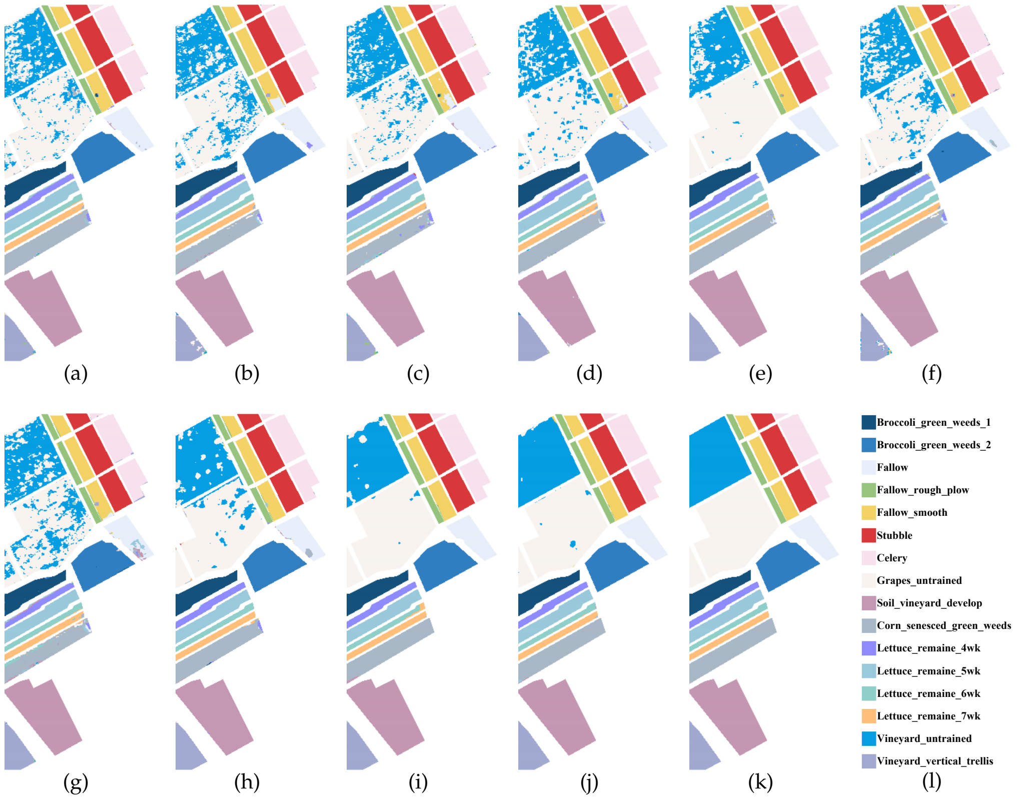


Figure 8. Classification maps on the SA dataset. (**a**) 2D-CNN. (**b**) 3D-CNN. (**c**) HybridSN. (**d**) SSAN. (**e**) MAFN. (**f**) ViT. (**g**) HiT. (**h**) CVSSN. (**i**) SSFTT. (**j**) MLFEN (Ours). (**k**) Ground truth. (**l**) Color bar.

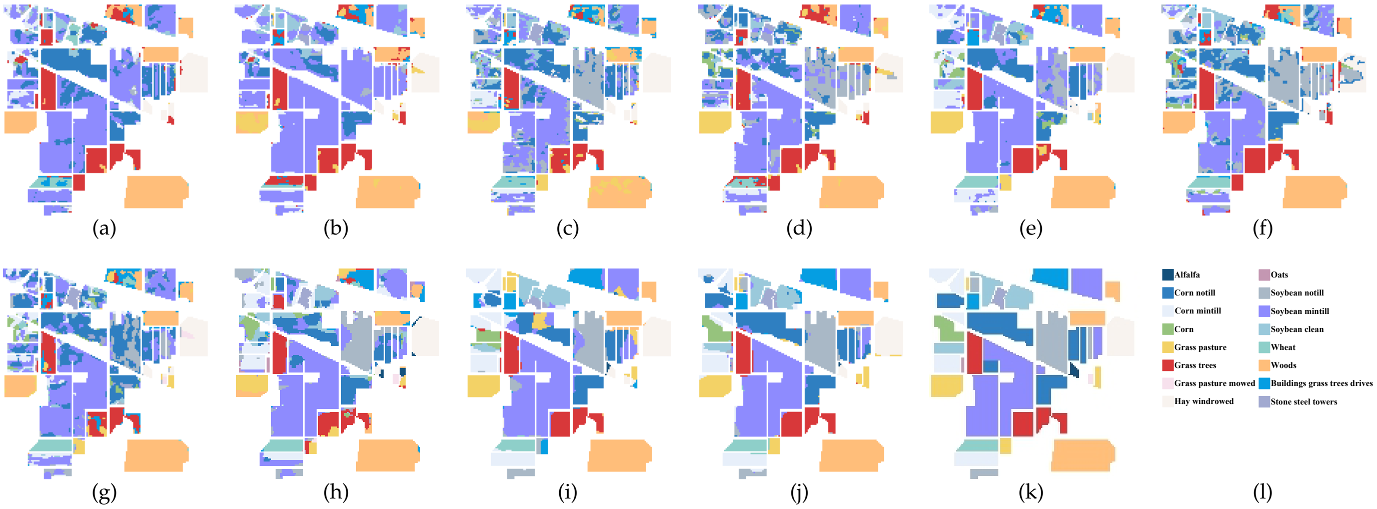


Figure 9. Classification maps on the IP dataset. (**a**) 2D-CNN. (**b**) 3D-CNN. (**c**) HybridSN. (**d**) SSAN. (**e**) MAFN. (**f**) ViT. (**g**) HiT. (**h**) CVSSN. (**i**) SSFTT. (**j**) MLFEN (Ours). (**k**) Ground truth. (**l**) Color bar.

On the IP dataset, MLFEN achieved the best classification results for the upper middle part “Corn notill” and the right middle part “Soybean notill”, which are easily misclassified by other methods. Although MLFEN exhibits some noise, the classification outcomes for various categorical areas are generally accurate. Unlike other methods, it does not suffer from the issue where the majority of pixels within categorical regions are misclassified into incorrect categories.

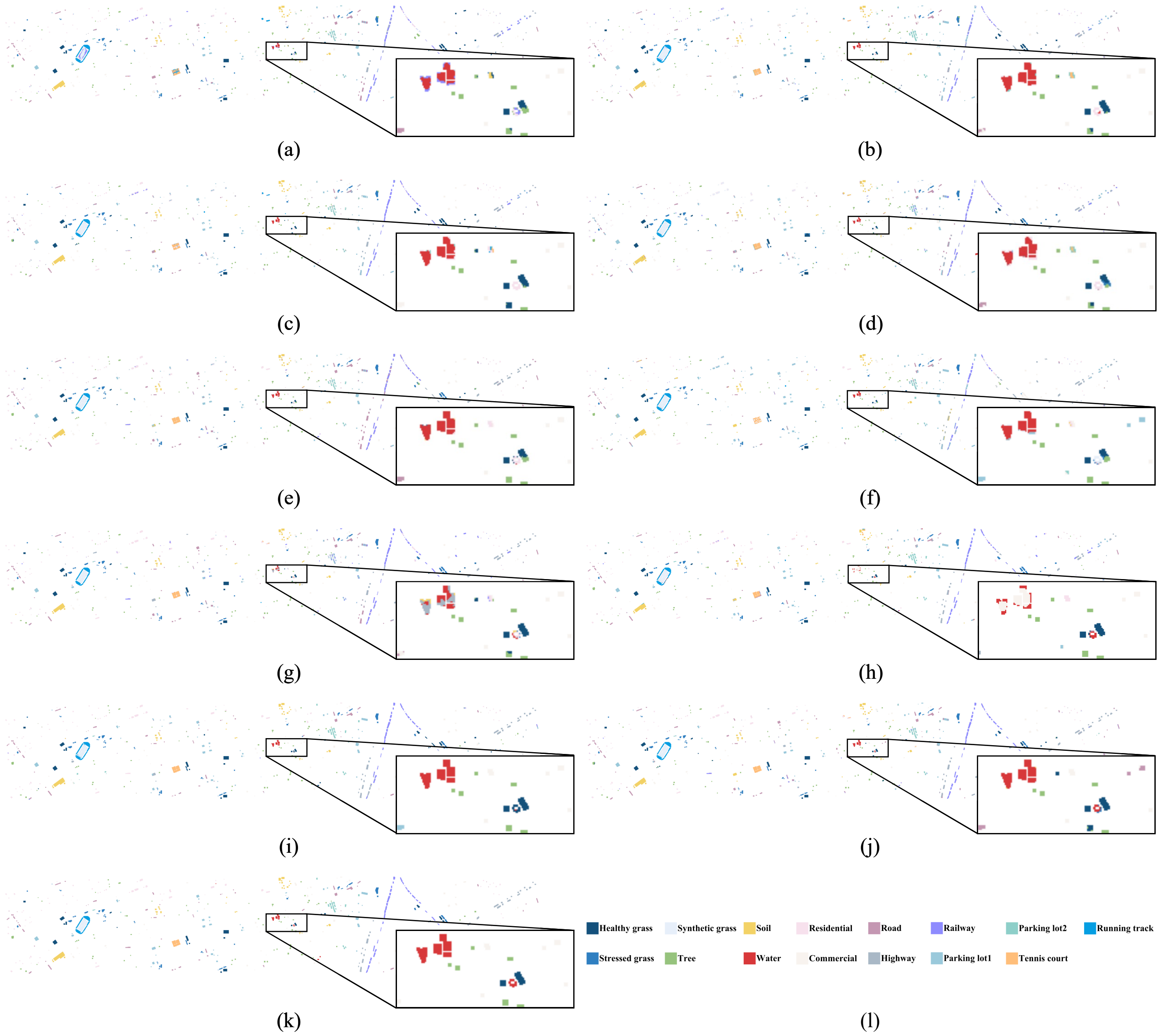
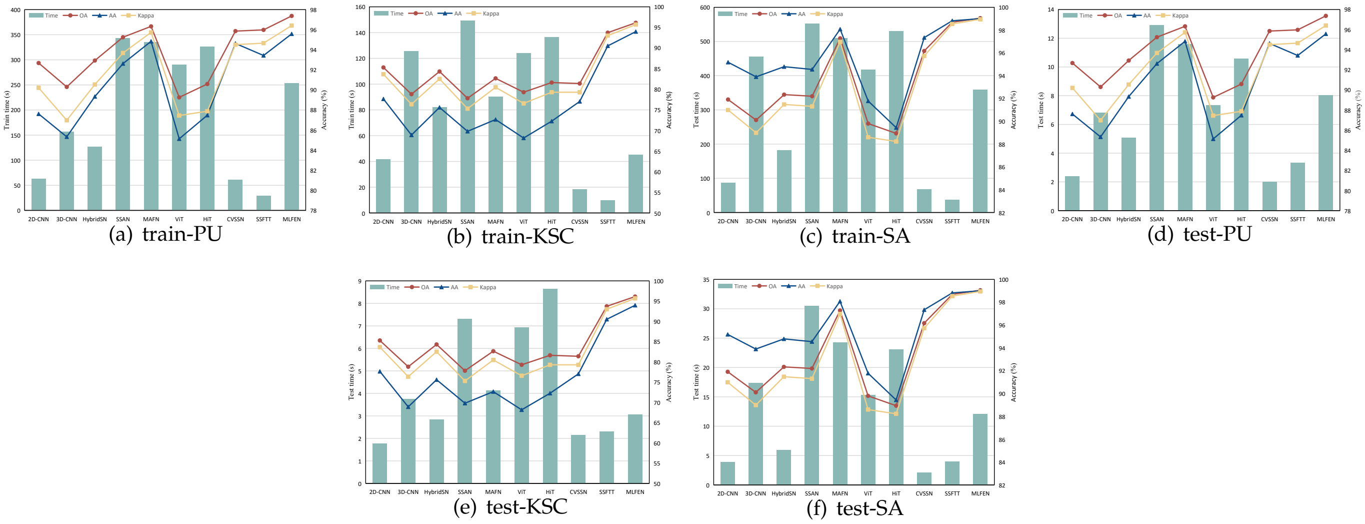


Figure 10. Classification maps on the HU dataset. (**a**) 2D-CNN. (**b**) 3D-CNN. (**c**) HybridSN. (**d**) SSAN. (**e**) MAFN. (**f**) ViT. (**g**) HiT. (**h**) CVSSN. (**i**) SSFTT. (**j**) MLFEN (Ours). (**k**) Ground truth. (**l**) Color bar.

As for the HU dataset, by analysing the ROI in Figure 10, it can be found that the MLFEN method achieves optimal classification in the red and blue clustered regions and achieves efficient classification. Our method demonstrates superior boundary classification capabilities, further corroborating the exceptional classification performance of the MLFEN approach.

4. Discussion

Figure 11 shows the training and testing times of the different methods on the three datasets. In order to facilitate a comprehensive and objective evaluation of the different models with regard to both computational cost and performance, the three indicators are presented in the figure in the form of broken lines. Owing to the properties of convolution such as parameter sharing and local connectivity, 2D-CNN has relatively less computation time. However, due to the increase in dimensions and parameters, the running time of 3D-CNN is substantially longer. HybridSN extracts the joint spatial–spectral features of the image by combining the above two convolutions, and the running time is shorter than pure 3D-CNN, but naturally longer than 2D-CNN. It is worth noting that the CVSSN and SSFTT methods maintain high accuracy while also having relatively short time costs. The computation times of ViT and HiT are relatively long due to the introduction of the self-attention mechanism and the fully connected layers, which require a large number of matrix multiplication and addition operations at each layer of the model. These operations require more computational resources and time. The method proposed in this paper requires significantly less time than the ViT and HiT methods, while achieving the best classification performance on all three datasets. The above analysis shows that, on the one hand, a certain extension of the computation time is within a reasonable range, considering the significant increase in performance. On the other hand, investigations toward further reducing computational costs while maintaining superior classification performance represent a direction worth exploring in the future.



(a) train-PU (b) train-KSC (c) train-SA (d) test-PU (e) test-KSC (f) test-SA

Figure 11. Running time and classification performance of different models.

5. Conclusions

In this paper, MLFEN network is proposed for mining multiple sources of information from HSIs at multiple levels. On the one hand, in order to obtain 3D attributes of HSIs and maintain the spatial structure of HSIs, the HCAM module employs hybrid convolution to mine shallow features. HCAM integrates the band shift strategy, convolution operations and attention mechanisms to achieve focusing, mining and enhancement of multidimensional features. On the other hand, in order to avoid excessive information loss and increase model adaptability to the HSI classification task, EDVT reorganizes the encoder connections and adds the AFF module to achieve adaptive feature reuse and fusion. In addition, a novel sparse loss function is proposed to enhance the sparse representation ability of the model, providing prediction results that are closer to the real data distribution. Extensive experiments conducted on five datasets fully validate the superior effectiveness of MLFEN.

In the future, we will introduce more advanced techniques to optimize the model and make it more suitable for HSI classification tasks, such as transfer learning and band selection. In addition, the fusion strategy of the two modules, EDVT and HCAM, may be an important factor affecting performance thus warranting further attention and research.

Author Contributions: S.F.; Conceptualization, Methodology, Writing—original draft. X.L.; Software, Writing—review and editing. S.T.; Supervision, Resources. W.C.; Investigation, Data curation. E.Z.; Formal analysis, Visualization, Project administration. All authors have reviewed and consented to the final version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62376225, 62006188), the QinChuangyuan High-Level Innovation and Entrepreneurship Talent Program of Shaanxi (No. 2021QCYRC4-50), and Chinese Universities Scientific Fund (No. 2452022341).

Data Availability Statement: Data is included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HSI	Hyperspectral image
MLFEN	Multi-Level Feature Extraction Network
HCAM	Hybrid Convolutional Attention Module
EDVT	Enhanced Dense Vision Transformer
PCA	Principal Component Analysis
CNN	Convolutional Neural Network
1D-CNN	1D Convolutional Neural Network
2D-CNN	2D Convolutional Neural Network
3D-CNN	3D Convolutional Neural Network
HybridSN	Hybrid Spectral Convolutional Neural Network
CVSSN	Central Vector-oriented Self-Similarity Network
SSFTT	Spectral–Spatial Feature Tokenization Transformer
SENet	Squeeze-and-Excitation Network
DANet	Dual Attention Network
SSAN	Spectral-Spatial Attention Network
MAFN	Multi Attention Fusion Network
ViT	Vision Transformer
HiT	Hyperspectral image Transformer
AFF	Adaptive Feature Fusion
MSA	Multi-head Self-Attention mechanism
MLP	Multi-Layer Perception network
AM	Attention Module
ReLU	Rectified Linear Unit
LN	Layer Normalization
PU	Pavia University dataset
KSC	Kennedy Space Center dataset
SA	Salinas dataset
IP	Indian Pines dataset
HU	Houston University
ROSIS	Reflective Optics System Imaging Spectrometer
AVIRIS	Airborne Visible/InfraRed Imaging Spectrometer
OA	Overall Accuracy
AA	Average Accuracy
Kappa	Kappa Coefficient
ROI	Region Of Interest

References

1. Liao, X.; Liao, G.; Xiao, L. Rapeseed Storage Quality Detection Using Hyperspectral Image Technology-An Application for Future Smart Cities. *J. Test. Eval.* **2023**, *51*, 1740–1752. [CrossRef]
2. Ghandehari, M.; Aghamohamadnia, M.; Dobler, G.; Karpf, A.; Cavalcante, C.; Buckland, K.; Qian, J.; Koonin, S. Ground based Hyperspectral Imaging of Urban Emissions. In Proceedings of the 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Los Angeles, CA, USA, 21–24 August 2016; pp. 1–3. [CrossRef]

3. Wang, J.; Zhang, L.; Tong, Q.; Sun, X. The Spectral Crust project—Research on new mineral exploration technology. In Proceedings of the 2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Shanghai, China, 4–7 June 2012; pp. 1–4. [CrossRef]
4. Contreras, C.; Khodadadzadeh, M.; Tusa, L.; Loidolt, C.; Tolosana-Delgado, R.; Gloaguen, R. Geochemical and Hyperspectral Data Fusion for Drill-Core Mineral Mapping. In Proceedings of the 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019; pp. 1–4. [CrossRef]
5. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
6. Ang, K.L.M.; Seng, J.K.P. Big data and machine learning with hyperspectral information in agriculture. *IEEE Access* **2021**, *9*, 36699–36718. [CrossRef]
7. Samaniego, L.; Bárdossy, A.; Schulz, K. Supervised classification of remotely sensed imagery using a modified *k*-NN technique. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2112–2125. [CrossRef]
8. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
9. Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
10. Xue, Z.; Nie, X.; Zhang, M. Incremental Dictionary Learning-Driven Tensor Low-Rank and Sparse Representation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–19. [CrossRef]
11. Hughes, G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory.* **1968**, *14*, 55–63. [CrossRef]
12. Wu, Z.; Yan, Z. Selection of optimal bands for hyperspectral local feature descriptor. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 5511205. [CrossRef]
13. Xu, B.; Li, X.; Hou, W.; Wang, Y.; Wei, Y. A similarity-based ranking method for hyperspectral band selection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 9585–9599. [CrossRef]
14. Licciardi, G.; Marpu, P.R.; Chanussot, J.; Benediktsson, J.A. Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles. *IEEE Geosci. Remote Sens. Lett.* **2011**, *9*, 447–451. [CrossRef]
15. Ye, Q.; Yang, J.; Liu, F.; Zhao, C.; Ye, N.; Yin, T. L1-norm distance linear discriminant analysis based on an effective iterative algorithm. *IEEE Trans. Circ. Syst. Video Technol.* **2016**, *28*, 114–129. [CrossRef]
16. Luo, F.; Zhang, L.; Zhou, X.; Guo, T.; Cheng, Y.; Yin, T. Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 1082–1086. [CrossRef]
17. Ghamisi, P.; Benediktsson, J.A.; Ulfarsson, M.O. The spectral-spatial classification of hyperspectral images based on Hidden Markov Random Field and its Expectation-Maximization. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, Australia, 21–26 July 2013; pp. 1107–1110. [CrossRef]
18. Dalla Mura, M.; Villa, A.; Benediktsson, J.A.; Chanussot, J.; Bruzzone, L. Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 542–546. [CrossRef]
19. He, L.; Liu, C.; Li, J.; Li, Y.; Li, S.; Yu, Z. Hyperspectral image spectral–spatial-range Gabor filtering. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4818–4836. [CrossRef]
20. Ji, R.; Gao, Y.; Hong, R.; Liu, Q.; Tao, D.; Li, X. Spectral-spatial constraint hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 1811–1824. [CrossRef]
21. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
22. Ma, A.; Filippi, A.M.; Wang, Z.; Yin, Z.; Huo, D.; Li, X.; Güneralp, B. Fast sequential feature extraction for recurrent neural network-based hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5920–5937. [CrossRef]
23. Yu, Q.; Wei, W.; Pan, Z.; He, J.; Wang, S.; Hong, D. GPF-Net: Graph-Polarized Fusion Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5519622. [CrossRef]
24. Cai, Y.; Zhang, Z.; Cai, Z.; Liu, X.; Jiang, X. Hypergraph-structured autoencoder for unsupervised and semisupervised classification of hyperspectral image. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 5503505. [CrossRef]
25. Qin, A.; Tan, Z.; Wang, R.; Sun, Y.; Yang, F.; Zhao, Y.; Gao, C. Distance Constraints-based Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5511416. [CrossRef]
26. Paoletti, M.E.; Haut, J.M.; Fernández-Beltran, R.; Plaza, J.; Plaza, A.J.; Li, J.Y.; Pla, F. Capsule Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2145–2160. [CrossRef]
27. Qi, W.; Zhang, X.; Wang, N.; Zhang, M.; Cen, Y. A spectral-spatial cascaded 3D convolutional neural network with a convolutional long short-term memory network for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 2363. [CrossRef]
28. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 3–13. [CrossRef]
29. Wei, H.; Yangyu, H.; Li, W.; Fan, Z.; Hengchao, L. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
30. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.; Zhang, X.; Huang, X. Hyperspectral image classification with deep learning models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [CrossRef]

31. Hamida, A.B.; Benoit, A.; Lambert, P.; Amar, C.B. 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]
32. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
33. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 740–754. [CrossRef]
34. Ma, X.; Fu, A.; Wang, J.; Wang, H.; Yin, B. Hyperspectral image classification based on deep deconvolution network with skip architecture. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4781–4791. [CrossRef]
35. Nalepa, J.; Myller, M.; Imai, Y.; Honda, K.I.; Takeda, T.; Antoniak, M. Unsupervised Segmentation of Hyperspectral Images Using 3-D Convolutional Autoencoders. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1948–1952. [CrossRef]
36. Zhang, S.; Zhang, X.; Li, T.; Meng, H.; Cao, X.; Wang, L. Adversarial Representation Learning for Hyperspectral Image Classification with Small-Sized Labeled Set. *Remote Sens.* **2022**, *14*, 2612. [CrossRef]
37. Sellami, A.; Tabbone, S. Deep neural networks-based relevant latent representation learning for hyperspectral image classification. *Pattern Recognition* **2022**, *121*, 108224. [CrossRef]
38. Zhang, K.; Zhu, D.; Min, X.; Zhai, G. Implicit Neural Representation Learning for Hyperspectral Image Super-Resolution. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6. [CrossRef]
39. Li, M.; Liu, Y.; Xue, G.; Huang, Y.; Yang, G. Exploring the Relationship Between Center and Neighborhoods: Central Vector Oriented Self-Similarity Network for Hyperspectral Image Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1979–1993. [CrossRef]
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
41. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154. [CrossRef]
42. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral–spatial attention network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3232–3245. [CrossRef]
43. Li, Z.; Zhao, X.; Xu, Y.; Li, W.; Zhai, L.; Fang, Z.; Shi, X. Hyperspectral Image Classification with Multiattention Fusion Network. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5503305. [CrossRef]
44. Yan, H.; Zhang, E.; Wang, J.; Leng, C.; Peng, J. MTFFN: Multimodal Transfer Feature Fusion Network for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6008005. [CrossRef]
45. Lyutikova, M.N.; Korobeynikov, S.M.; Rao, U.M.; Fofana, I. Mixed Insulating Liquids With Mineral Oil for High-Voltage Transformer Applications: A Review. *IEEE Trans. Dielect. Electr. Insul.* **2022**, *29*, 454–461. [CrossRef]
46. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2021**, arXiv:2010.11929v2.
47. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 11–17 October 2021; pp. 568–578. [CrossRef]
48. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 11–17 October 2021; pp. 10012–10022. [CrossRef]
49. Touvron, H.; Cord, M.; Jégou, H. Deit iii: Revenge of the vit. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 516–533. arXiv.2204.07118. [CrossRef]
50. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.H.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 11–17 October 2021; pp. 558–567. [CrossRef]
51. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5518615. [CrossRef]
52. Zhao, Z.; Hu, D.; Wang, H.; Yu, X. Convolutional Transformer Network for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6009005. [CrossRef]
53. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
54. Ma, C.; Jiang, J.; Li, H.; Mei, X.; Bai, C. Hyperspectral Image Classification via Spectral Pooling and Hybrid Transformer. *Remote Sens.* **2022**, *14*, 4732. [CrossRef]
55. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [CrossRef]
56. Yang, X.; Cao, W.; Lu, Y.; Zhou, Y. Hyperspectral image transformer classification networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5528715. [CrossRef]
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]

58. Liu, F.; Ren, X.; Zhang, Z.; Sun, X.; Zou, Y. Rethinking skip connection with layer normalization. In Proceedings of the International Conference on Computational Linguistics (COLING), Barcelona, Spain, 8–13 December 2020; pp. 3586–3598. [CrossRef]
59. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 4700–4708. [CrossRef]
60. Ho, Y.; Wookey, S. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access* **2019**, *8*, 4806–4813. [CrossRef]
61. Kumar, A.; Shaikh, A.M.; Li, Y.; Bilal, H.; Yin, B. Pruning filters with L1-norm and capped L1-norm for CNN compression. *Appl. Intell.* **2021**, *51*, 1152–1160. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.