2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DDF-Net: A Dual-Branch Deep Feature Fusion Network for Few-Shot Hyperspectral Image Classification

Ke Chen[1] and AiGuo Chen[2]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Wuxi City, Jiangsu Province, China
[2] School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Wuxi City, Jiangsu Province, China
agchen@jiangnan.edu.cn

Abstract. In recent years, various deep learning frameworks have been introduced for hyperspectral image (HSI) classification. However, the proposed network models often exhibit high model complexity and fail to provide high classification accuracy when applied in few-shot learning scenarios. In this paper, we propose a Dual-Branch Deep Feature Fusion Network (DDF-Net) for few-shot hyperspectral image classification. DDF-Net extracts multi-layer features from hyperspectral images using a pre-trained CNN model and applies Principal Component Analysis (PCA) for dimensionality reduction. Subsequently, non-overlapping image patches are extracted from the reduced-dimensional features, and processed through two parallel streams: a 3D-CNN stream for spatial feature extraction and a CV-CNN stream for spectral feature extraction. Additionally, to enhance model performance, the Squeeze-and-Excitation (SE) mechanism is incorporated. Finally, the features from the two branches are effectively integrated through concatenation fusion and enhancement by the SE module, and then input into an SVM for classification. Experiments conducted on multiple datasets demonstrate the effectiveness and efficiency of DDF-Net in hyperspectral image classification, outperforming state-of-the-art methods.

Keywords: Hyperspectral Data;Few-Shot Learning, Deep Features, Convolutional Kernels, Dual-Branch.

1 Introduction

Hyperspectral image (HSI) classification addresses critical remote sensing tasks like tree species identification [1] and crop yield estimation [2]. However, limited labeled data in practical scenarios often triggers the Hughes phenomenon, risking overfitting due to the spectral-band-to-sample imbalance [3]. This has spurred research on small-sample HSI classification.

Feature extraction is pivotal in HSI classification. Traditional shallow features (e.g., local binary patterns [4], morphological features [5]) require intensive domain expertise, prompting the shift to deep learning for automatic hierarchical feature learning [3]. Convolutional neural networks (CNNs), despite success in spatial-spectral feature extraction, face challenges in jointly modeling spatial structures and spectral characteristics efficiently.

Early deep models, such as stacked autoencoders (SAE) [6] and deep belief networks (DBN) [7], lost spatial information by flattening spectral vectors. Later methods integrated principal component analysis (PCA) with 2D-CNNs to reduce dimensionality [8], but separated spatial-spectral processing. 3D-CNNs emerged to directly process high-dimensional data, enabling joint spatial-spectral feature extraction [9]. Notable frameworks like spectral-spatial feature-based classification (SSFC) [10] and supervised spatial-spectral residual networks (SSRN) [9] demonstrated improved performance, yet high computational complexity from 3D convolutions hindered edge deployment. Transformer-based approaches [11] introduced redundant computations by encoding all spectral/spatial tokens, underperforming 3D-CNNs in accuracy while escalating complexity.

To balance efficiency and performance, complex-valued neural networks (CVNNs) [12], effective for radar/medical data, offer potential for HSI classification by leveraging complex-valued features. This study proposes a Dual-Branch Deep Feature Fusion Network (DDF-Net) for small-sample HSI classification, integrating CNN-based multi-layer feature extraction, PCA dimensionality reduction, and fast Fourier transform (FFT) spectral analysis via a dual-branch architecture. The 3D-CNN stream extracts spatial features from non-overlapping patches, while the CV-CNN stream processes spectral information. Features are fused via concatenation and enhanced by a squeeze-and-excitation (SE) module before classification by support vector machine (SVM).

The main contributions of this study are as follows:

(1) A novel model, DDF-Net, is proposed for hyperspectral image classification in limited sample scenarios. This model improves classification accuracy and efficiency by processing both spatial and spectral information simultaneously through a dual-branch architecture.

(2) PCA dimensionality reduction and FFT spectral analysis methods are introduced, effectively reducing computational complexity and eliminating redundant information, while capturing the spectral characteristics of images.

(3) An advanced SE module is adopted to enhance the interdependencies among channels, further improving the model's performance.

Through the research in this paper, we aim to provide new theoretical support and technical solutions for the development of hyperspectral image classification, pushing research in this field to a higher level. The remainder of this paper is organized as follows: Section 2 describes the proposed method; Section 3 presents experimental results on different datasets; finally, we conclude and propose suggestions for future work in Section 4.

2 Method

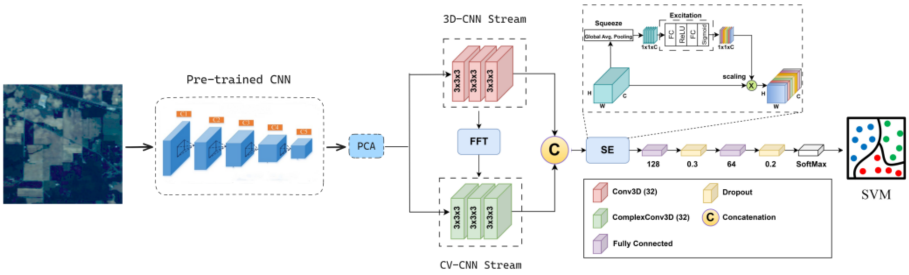


Figure 1. The overall framework of the proposed DDF-Net for scene classification, which consists of four main components: 1) multi-layer convolutional feature extraction using a pre-trained CNN; 2) the DBFF dual-branch structure; 3) the SE attention block; and 4) scene classification using a linear SVM classifier.

For small-sample hyperspectral image (HSI) classification, the proposed Dual-Branch Deep Feature Fusion Network (DDF-Net) achieves efficient spatial-spectral feature integration by combining CNN-based multi-layer feature extraction, PCA dimensionality reduction, FFT spectral analysis, and SE attention mechanism. Comprising parallel 3D-CNN (spatial processing) and CV-CNN (spectral modeling) branches, DDF-Net extracts non-overlapping patch features, fuses them via concatenation at the feature fusion layer, enhances channel dependencies through SE weighting, and classifies using support vector machine (SVM).

Figure 1 illustrates the overall framework of the proposed DBFF for scene classification. The details of this architecture are described below.

2.1 Multi Layer Convolutional Feature Extraction

Convolutional operations are the core component of CNNs, used to extract features from input data. In two-dimensional image processing, a convolutional operation can be represented as the dot product between the input image X and a convolutional kernel K, with a sliding window traversal of the entire image. Mathematically, the convolutional operation can be expressed as:

$$\mathrm{Y(i,j)} = (\mathrm{X} \otimes \mathrm{K})(\mathrm{i,j}) = \sum_{\mathrm{m}} \sum_{\mathrm{n}} \mathrm{X(i+m, j+n)} \cdot \mathrm{K(m,n)} \tag{1}$$

Where Y(i,j) is the pixel value at the i-th row and j-th column of the convolved feature map, X⊗K represents the convolutional operation, $\sum_m \sum_n$ indicates the summation over all elements of the convolutional kernel K, X(i+m,j+n) is the pixel value at the (i+m)-th row and (j+n)-th column of the input image X, and K(m,n) is the parameter value at the m-th row and n-th column of the convolutional kernel K.

In CNNs, convolutional layers are often stacked to form a multi-layer convolutional structure. Each layer uses multiple different convolutional kernels to extract different features[13]. As the network depth increases, the convolutional layers can further combine the simple features extracted from previous layers to extract more complex features.

Specifically, in a multi-layer convolutional structure, the output feature map of each layer is the result of the convolution operation between the input feature map of the previous layer and the convolution kernel of the current layer. Taking the ℓ layer as an example, its output feature map x_j^ℓ can be represented as:

$$\mathrm{x}_{\mathrm{j}}^{\ell} = \mathrm{f}(\sum_{\mathrm{i}} \mathrm{x}_{\mathrm{i}}^{\ell-1} \otimes \mathrm{k}_{\mathrm{ij}}^{\ell} + \mathrm{b}_{\mathrm{j}}^{\ell}) \tag{2}$$

Where $x_i^{\ell-1}$ represents the output feature map of the previous layer (layer $\ell-1$),k_{ij}^{ℓ} represents the convolutional kernel of the current layer (layer ℓ),b_j^{ℓ} represents the bias term, and $\mathrm{f}(\cdot)$ represents the activation function.

2.2 DDF-Net Dual Branch Structure

The DDF-Net leverages a distinctive dual-branch architecture, with one branch dedicated to processing the spatial structure of the original image patches features and the other branch focusing on the spectral properties of the image patches features. This design allows the network to capture both spatial context and spectral information in the image, enhancing its ability to comprehend the image.

(1)Spatial Weighting Branch (3D-CNN Stream).

The 3D-CNN architecture in our framework incorporates a hardwired layer, three convolutional layers, two downsampling layers, and a fully connected layer. Each 3D convolutional kernel convolves over a cube consisting of seven consecutive frames, with each frame having a patch size of 60x40, as illustrated in Figure 2.

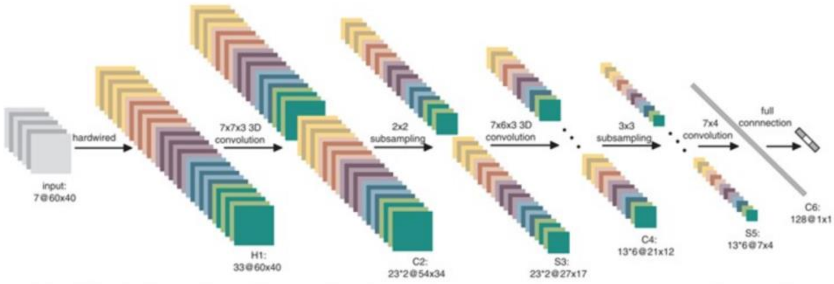


Figure 2. A 3D-CNN architecture for processing the spatial structure of the original image patch features.

In the first layer, we apply a fixed, hardwired kernel to process the original frames, generating information across multiple channels. Each of these channels is then processed separately. Finally, the information from all channels is combined to obtain the final feature representation, as depicted in Figure 3.

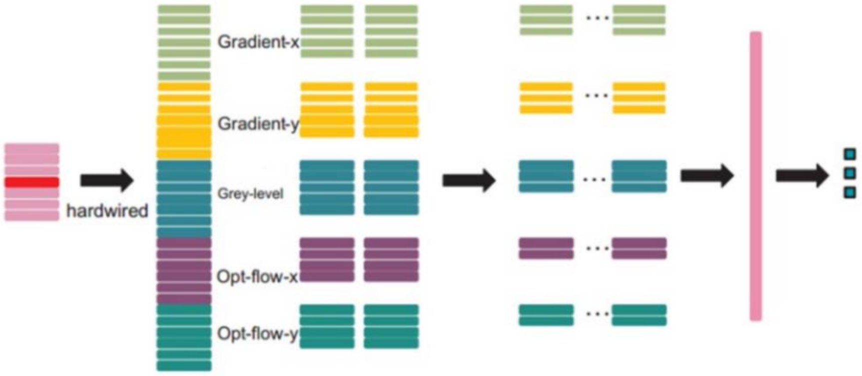


Figure 3. A fixed, hardwired kernel is applied to process the original frames, generating information across multiple channels.

For each frame, five channels of information are extracted: grayscale, gradients in the x and y directions, and optical flow in the x and y directions. The first three can be computed for each frame individually. However, the horizontal and vertical optical flow fields require two consecutive frames to be determined. Therefore, the total number of feature maps is calculated as 7x3 + (7-1) x 2 = 33.

The spatial weighting is achieved through a well-designed 3D Convolutional Neural Network (3D-CNN) stream. Here, the 3D-CNN captures the structural information in the image patches across spatial dimensions.Assuming X is the input original image patch, W_s is the weight of the 3D-CNN, and b_s is the bias term, then the spatially weighted feature F_s can be expressed as:

$$\mathrm{F_s} = \mathrm{f_s}(\mathrm{X}; \mathrm{W_s}; \mathrm{b_s}) \tag{3}$$

Where f_s represents the mapping function of the 3D-CNN.

(2)Spectral Property Branch (FFT + CV-CNN Stream)

A complex-valued neural network (CV-NN) maintains an inter-layer fully connected structure with complex-valued inputs, weights, and biases. In complex-valued convolutional neural networks (CV-CNNs), forward propagation applies nonlinear activation to the real and imaginary components of weighted sums separately, producing complex-valued outputs at each layer. Backward training employs complex-domain gradient descent for parameter optimization.

CV-CNNs utilize complex-valued convolutional layers and pooling operations, where both input feature maps and filter kernels are complex-valued—distinguishing them from real-valued CNNs—to directly process complex-valued data for spectral-spatial feature extraction in hyperspectral image (HSI) classification.

The inputs and kernels for complex-valued convolutions can be expressed as：

input：$\mathrm{V} = \mathrm{V_r} + \mathrm{jV_i}$

Convolutional Kernel：$\mathrm{K} = \mathrm{K_r} + \mathrm{jK_i}$

$$\begin{aligned}\mathrm{K} * \mathrm{V} &= (\mathrm{K_\tau} + \mathrm{jK_i}) * (\mathrm{V_\tau} + \mathrm{jV_i}) \\ &= (\mathrm{K_\tau} * \mathrm{V_\tau} - \mathrm{K_i} * \mathrm{V_i}) + \mathrm{j}(\mathrm{K_\tau} * \mathrm{V_i} + \mathrm{K_i} * \mathrm{V_\tau}) \qquad (4)\end{aligned}$$

First, the Fast Fourier Transform (FFT) is applied to the image patch to obtain its spectral representation, known as a complex patch. Then, these patches undergo channel weighting through an efficient Channel-wise Convolutional Neural Network (CV-CNN) stream, mining the spectral properties of the image.

FFT: Assuming X is the input original image patch, its spectral representation X_{fft} can be expressed as:

$$\mathrm{X_{fft}} = \mathrm{FFT(X)} \qquad (5)$$

CV-CNN: Assuming W_c is the weight of the CV-CNN and b_c is the bias term, then the channel-weighted spectral feature F_c can be expressed as:

$$F_c = f_c(X_{fft}; W_c; b_c) \qquad (6)$$

Where f_c represents the mapping function of the CV-CNN.

2.3 SE Attention Blocks

The SE attention mechanism, is a channel-based attention mechanism designed to enhance the network's focus on different feature channels Its core idea lies in explicitly modeling the interdependencies between feature channels to adaptively recalibrate channel-wise feature responses[14].

The SE module's workflow comprises two main steps: Squeeze and Excitation[15].

Squeeze: Each channel's feature map is compressed into a scalar value using global average pooling, capturing global information for that channel.Assuming the input feature map X has dimensions (H,W,C), where H and W are the height and width, and C is the number of channels, the output after global average pooling has dimensions (1,1,C).

Excitation: One or more fully connected layers are used to capture the dependencies between channels. This typically involves two fully connected layers, the first for dimensionality reduction and the second to restore the original channel count.Between these two fully connected layers, a ReLU activation function is often added for nonlinearity.Finally, a sigmoid activation function is used to limit the output to values between 0 and 1, representing the importance of each channel.

Let the input feature map X have dimensions (H,W,C). The channel weights W_c for the output feature map X′ after the SE module can be calculated as follows:

Squeeze:

$$\mathrm{z_c} = \mathrm{F_{sq}}(\mathrm{u_c}) = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum_{\mathrm{i=1}}^{\mathrm{H}} \sum_{\mathrm{j=1}}^{\mathrm{W}} \mathrm{u_c}(\mathrm{i}, \mathrm{j}) \tag{7}$$

Where u_c is the feature map of the c-th channel in X, and z_c is the global feature descriptor (i.e., the compressed scalar value) for that channel.

Excitation:

$$\mathrm{s} = \mathrm{F_{ex}}(\mathrm{z}, \mathrm{W}) = \sigma\big(\mathrm{g}(\mathrm{z}, \mathrm{W})\big) = \sigma\big(\mathrm{W_2}\delta(\mathrm{W_1 z})\big) \tag{8}$$

Where $W_1 \in R^{\frac{C}{T} \times C}$ and $W_2 \in R^{C \times \frac{C}{T}}$ are the weight matrices of the two fully connected layers, r is the reduction ratio, δ is the ReLU activation function, and σ is the sigmoid activation function. s is a vector of length C, representing the importance (i.e., weight) of each channel.

Feature Rescaling: Finally, the obtained channel weights s are multiplied channel-wise with the original feature map X to obtain the output feature map X′:

$$\mathrm{X'_c} = \mathrm{F_{scale}}(\mathrm{u_c}, \mathrm{s_c}) = \mathrm{s_c} \cdot \mathrm{u_c} \tag{9}$$

Where s_c is the weight corresponding to the c-th channel in s, and X'_c is the feature map of the c-th channel in the output feature map X′.

The principle of the compression excitation mechanism for SE attention is shown in Figure 4：

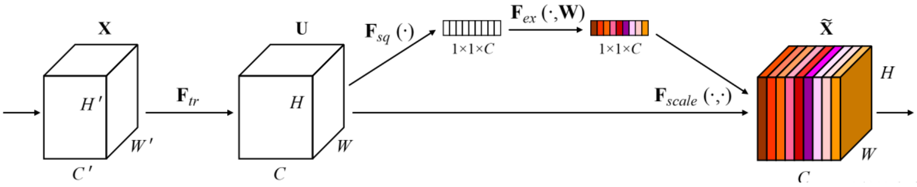


Figure 4. Schematic diagram of the compression excitation mechanism for SE attention

2.4 SVM Classification

Through the self-attention layer, the feature vectors with a dimension of K are derived for final classification. Before scene image classification, normalization is used to avoid overfitting. For the whole data set, the common operation is applied. Then, we randomly select feature vectors as training samples to train a linear SVM, while the rest of the feature vectors are used to test the effectiveness of our method.

3 Experiment and Analysis

3.1 Dataset

To evaluate the feasibility and effectiveness of our method, we conducted experiments on several popular hyperspectral scene datasets.

(1) Indian pines dataset

The Indiana Pines dataset was collected in June 1992 by the AVIRIS Spectral Imager in the Indiana Pine Forest Experimental Area in northwestern Indiana, USA. The data image measures 145x145 pixels, with a spatial resolution of 20 meters. It encompasses 220 spectral bands within the wavelength range of 0.4-2.5 micrometers. Twenty bands corresponding to water vapor absorption and low signal-to-noise ratios were removed, leaving the remaining 200 bands for experimentation. This dataset includes 16 types of ground objects, such as grasslands, buildings, and crops. The spatial distribution of its samples is illustrated in Figure 5.

(2) Pavia University dataset

The Pavia University dataset was collected in 2001 by the ROSIS spectral imager in the Pavia region of northern Italy. The image measures 610x340 pixels and has a spatial resolution of 1.3 meters. It comprises 115 spectral bands within the wavelength range of 0.43-0.86 micrometers. Twelve bands containing strong noise and water vapor absorption were removed, leaving the remaining 103 bands for experimentation. This dataset includes nine ground objects, such as roads, trees, and rooftops, with the spatial distribution of different classes illustrated in Figure 6.

(3) KSC dataset

The Kennedy Space Center (KSC) dataset was collected on March 23, 1996, by the AVIRIS spectral imager at the Kennedy Space Center in Florida. AVIRIS acquired 224 bands with a width of 10 nm, centered between 400-2500 nm. The spatial resolution of the KSC data, acquired from an altitude of approximately 20 kilometers, is 18 meters. After removing water absorption and low signal-to-noise ratio bands, 176 bands were used for analysis. The dataset defines 13 classes. The spatial distribution of its samples is illustrated in Figure 7.

3.2 Experimental Setup

In our experiments, we utilized the pre-trained CNN model VGG-VD16 to extract multi-layer deep features. The proposed method was compared with 3D-CNN [16], CA-GAN [17], DCFSL [18], 3D VS-CNN [19], S-DMM [20], TC-GAN [21], and the state-of-the-art method RPNet-RF [22]. Additionally, to evaluate the impact of the Attention SE block on model performance, experiments were conducted by removing this block and assessing the model's performance without it. Overall Accuracy (OA), Average Accuracy (AA), and Kappa statistic (Kappa) were reported to assess the performance of the proposed model. Furthermore, classification accuracy for each class was also provided. The experiments were conducted and repeated 10 times to validate the effectiveness of the 10 trials. For OA, AA, and Kappa, the average and standard deviation of all 10 trials were recorded. Three widely used hyperspectral datasets were employed: the Indian Pines dataset, the Pavia University dataset, and the KSC dataset. Figures 6, 7, and 8 display the reference data for these three datasets.

For these three datasets, 15 labeled samples from each class were randomly selected for training, while the remaining samples were used for testing and evaluation. The patch size used was 13×13, and the number of principal components was set to 15, which were determined to be the optimal choices through experimentation. The model was trained for 100 epochs with a batch size of 16. During the model training process, an early stopping strategy was adopted. Specifically, if the model's performance did not improve within consecutive 10 epochs, the training process was terminated and the model was restored to its best weights. The optimization algorithm used was Adam, with a learning rate set to 10^{-3}. The loss function employed was categorical cross-entropy. All models were implemented using the Python Keras framework and run with the TensorFlow backend. To ensure a fair comparison, all models were trained under the same conditions.

Table 1. Classification accuracies for the proposed and compared HSI classification methods on the IP dataset (the best accuracy in each row is shown in bold). Fifteen labeled samples per class were randomly selected for training

Class No	3D-CNN	CA-GAN	DCFSL	3D VS-CNN	S-DMM	TC-GAN	RPNet-RF	DDF-Net (no SE)	DDF-Net
1	83.87	100.0	100.0	90.32	91.67	100.0	93.48	94.32	95.64
2	38.08	61.78	60.79	75.94	47.18	78.77	81.30	85.46	**87.06**
3	41.84	68.22	78.77	85.03	44.88	**92.15**	85.66	90.24	91.08
4	52.70	92.34	94.59	95.95	33.04	**99.10**	83.31	92.14	93.87
5	74.79	82.69	85.68	91.03	78.44	95.30	94.14	95.62	**97.54**
6	87.27	89.51	96.64	97.34	92.50	95.94	95.15	96.38	**97.65**
7	100.0	100.0	100.0	100.0	100.0	100.0	43.98	92.78	94.08
8	94.38	**99.78**	92.22	97.84	85.26	100.0	97.76	97.12	98.26
9	100.0	100.0	100.0	100.0	100.0	100.0	63.83	87.35	89.46
10	64.26	76.28	71.89	80.88	66.74	**86.00**	83.07	82.04	84.68
11	41.43	64.22	65.66	73.32	70.39	81.39	**94.44**	92.17	94.06
12	41.70	78.72	73.18	**88.41**	40.82	73.18	82.96	81.04	83.37
13	99.47	99.47	100.0	98.95	99.49	100.0	99.10	99.24	99.64
14	84.24	82.32	93.28	84.24	81.35	97.28	**99.77**	97.65	98.23
15	70.89	92.99	87.87	86.52	68.35	83.83	98.45	97.58	**98.66**
16	97.44	92.31	100.0	98.72	98.80	100.0	97.60	97.50	98.76
OA(%)	58.94	75.52	77.45	83.06	67.04	87.47	90.23	92.41	**93.87**
AA(%)	73.27	81.21	87.54	90.28	74.93	92.68	87.12	89.22	**90.86**
Kappa(%)	54.06	72.69	74.65	80.89	62.44	85.78	88.87	91.02	**92.24**

Table 2. Classification accuracies for the proposed and compared HSI classification methods on the PU dataset (the best accuracy in each row is shown in bold). Fifteen labeled samples per class were randomly selected for training.

Class No	3D-CNN	CA-GAN	DCFSL	3D VS-CNN	S-DMM	TC-GAN	RPNet-RF	DDF-Net (no SE)	DDF-Net
1	70.41	60.16	74.55	83.27	96.97	89.07	96.37	97.06	**98.12**
2	73.10	72.83	97.20	76.96	81.15	97.57	97.37	97.56	**98.30**
3	73.80	98.03	80.57	81.91	92.69	67.08	97.19	97.49	**98.45**
4	89.37	89.44	94.62	86.86	**97.50**	88.03	78.86	78.75	80.67
5	96.39	99.70	100.0	99.55	100.0	100.0	97.85	97.23	98.80
6	69.68	79.94	90.37	82.81	84.73	93.80	98.92	98.47	**99.05**
7	86.46	90.04	92.47	77.94	97.71	**99.47**	93.82	91.78	93.23
8	77.09	81.95	81.62	**93.58**	93.23	93.07	85.67	88.31	89.48
9	86.05	97.32	**100.0**	71.84	99.89	96.67	98.58	98.80	99.67
OA(%)	75.24	76.81	90.71	81.63	88.30	93.20	94.60	93.94	**95.09**
AA(%)	80.26	76.94	90.20	83.86	93.76	91.60	93.96	93.32	**95.86**
Kappa(%)	68.43	71.02	87.73	76.46	84.90	91.00	93.27	93.06	**94.08**

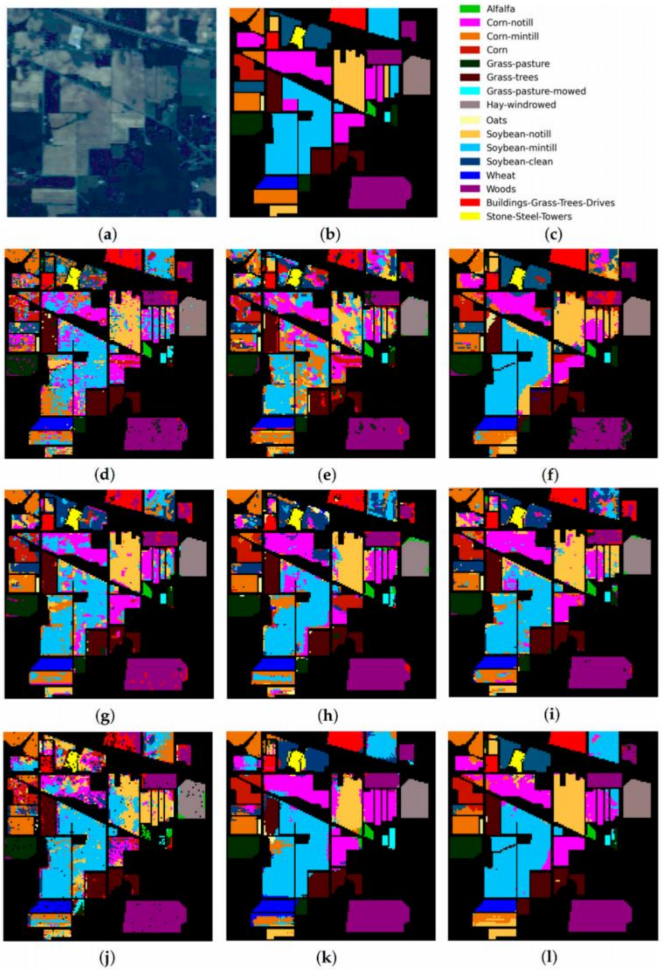
3.3 Experimental Results

Tables 1, 2, and 3 present the classification accuracies for the three datasets, including the accuracies for each class, OA, AA, and Kappa. It can be observed that the proposed model achieves superior results compared to other classification models. Compared with the latest state-of-the-art (SOTA) method, RPNet-RF, the proposed model demonstrates an accuracy improvement of nearly 3.5% on the India dataset and significant improvements on both the PU dataset and the KSC dataset. Additionally, by incorporating the attention mechanism, the model's performance improves by nearly 1% compared to the model without the SE block. Furthermore, the results show less variability or fluctuation compared to other models, indicating that the proposed model performs well and produces reliable and accurate predictions.

Table 3. Classification accuracies for the proposed and compared HSI classification methods on the KSC dataset (the best accuracy in each row is shown in bold). Fifteen labeled samples per class were randomly selected for training.

Class No	3D-CNN	CA-GAN	DCFSL	3D VS-CNN	S-DMM	TC-GAN	RPNet-RF	DDF-Net (no SE)	DDF-Net
1	89.41	88.20	96.92	97.15	96.01	99.87	**99.46**	97.75	98.87
2	86.40	85.53	86.40	91.28	88.84	**100.0**	96.20	96.87	98.12
3	85.06	95.02	98.76	80.09	**99.19**	96.68	97.54	97.61	98.54
4	54.01	90.72	82.28	42.29	54.96	86.08	98.32	97.73	**98.67**
5	83.56	90.41	91.78	58.09	80.79	93.84	95.90	97.88	**99.23**
6	76.64	94.39	97.66	70.59	96.35	**100.0**	91.53	96.12	97.25
7	100.0	100.0	100.0	70.00	100.0	97.78	100.0	97.89	99.29
8	92.55	86.78	**100.0**	62.81	99.29	96.63	98.14	97.07	98.04
9	60.59	86.73	100.0	74.55	100.0	99.60	98.52	98.68	99.11
10	93.32	86.12	99.74	61.48	**100.0**	99.49	99.06	97.87	98.83
11	93.07	92.57	100.0	78.68	100.0	100.0	99.80	97.18	98.75
12	93.85	88.52	99.18	78.24	98.99	97.95	99.16	97.93	**99.22**
13	100.0	100.0	100.0	99.89	100.0	100.0	99.95	98.27	99.34
OA(%)	87.18	91.17	97.59	80.15	95.83	98.39	98.51	97.68	**98.71**
AA(%)	85.27	84.64	96.36	74.24	93.42	97.53	97.97	96.24	**98.18**
Kappa(%)	85.73	90.20	97.31	77.81	95.35	98.20	98.33	97.65	**98.68**

To conduct an ablation study, we evaluated the data from each stream separately. Specifically, we used standard real-valued data in the CV-CNN stream and FFT data in the CVNN stream. The results show that the 3D-CNN stream achieves an accuracy of 95.1% on the PU dataset and 95.53% on the India dataset. On the other hand, the CV-CNN stream obtains a score of 94.74% on the PU dataset and 95.02% on the India dataset. However, when these features are fused together, the model's performance improves, with accuracy increases of more than 1% for all three datasets. Figures 5, 6, and 7 display the classification results for the three datasets. Compared to other methods, the proposed model exhibits visual quality that is closest to the real situation.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 5. IP dataset: (a) false-color image, (b) ground truth map, (c) legend. Classification maps obtained by the compared methods on the IP dataset: (d) 3D-CNN, (e) CA-GNN, (f)DCFSL, (g) 3D VS-CNN(h) S-DMM, (i) TC-GAN, (j) RPNet-RF (k) DDF-Net(no SE),and (I) DDF-Net. Fifteen labeled samples per class were randomly selected for training.

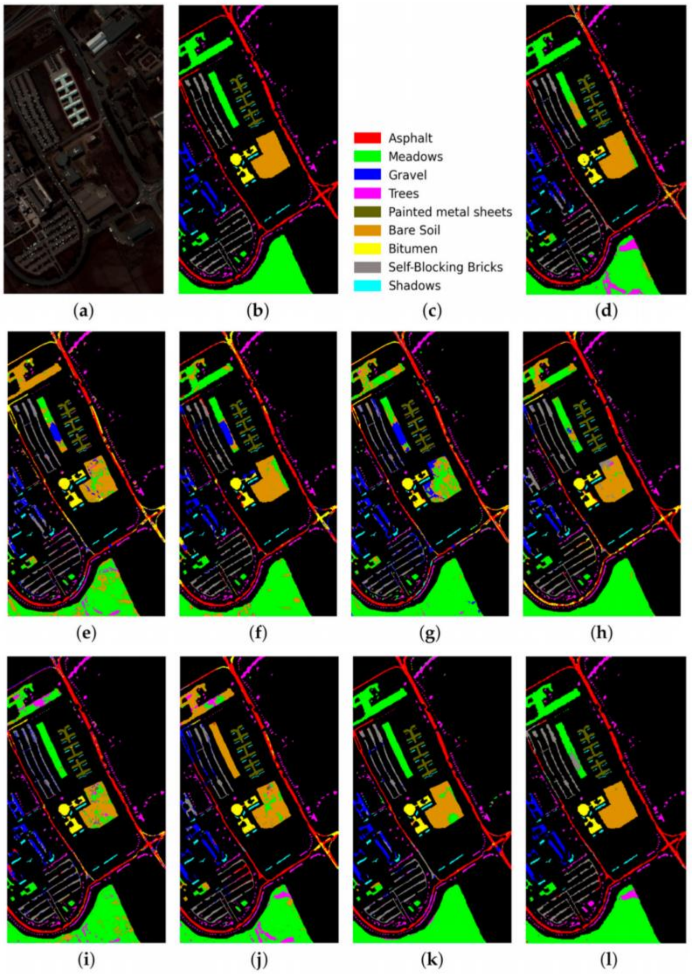


Figure 6. PU dataset: (a) false-color image, (b) ground truth map, (c) legend.Classification maps obtained by the compared methods on the PU dataset: (d) 3D-CNN, (e) CA-GNN, (f)DCFSL, (g) 3D VS-CNN(h) S-DMM, (i) TC-GAN, (j) RPNet-RF (k) DDF-Net(no SE),and (I) DDF-Net. Fifteen labeled samples per class were randomly selected for training.

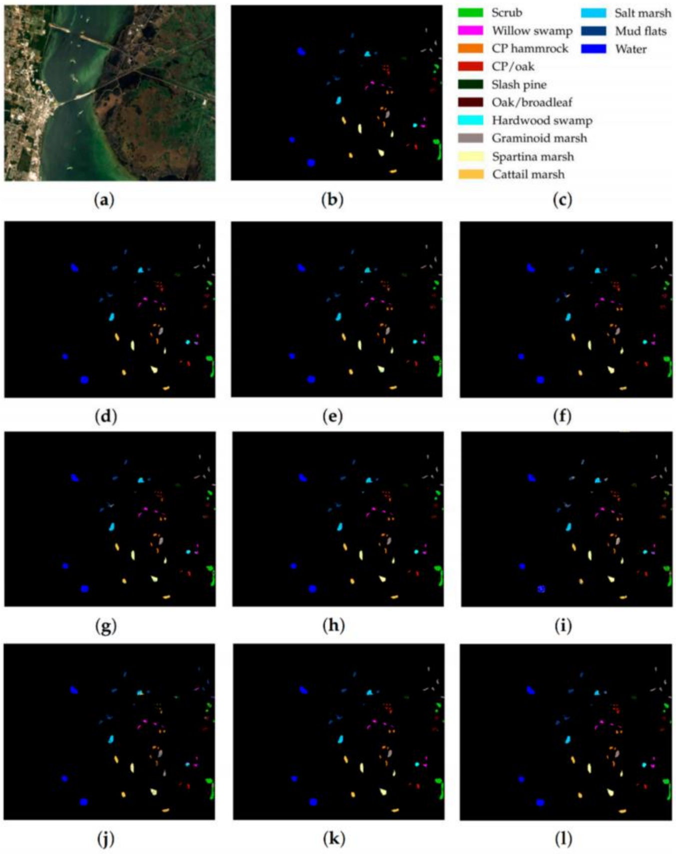


Figure 7. KSC dataset: (a) false-color image, (b) ground truth map, (c) legend. Classification maps obtained by the compared methods on the KSC dataset: (d) 3D-CNN, (e) CA-GNN, (f)DCFSL, (g) 3D VS-CNN(h) S-DMM, (i) TC-GAN, (j) RPNet-RF (k) DDF-Net(no SE),and (I) DDF-Net. Fifteen labeled samples per class were randomly selected for training.

4 Conclusion

This paper presents DDF-Net, a dual-branch deep feature fusion network with SE attention for few-shot hyperspectral image (HSI) classification. Integrating pre-trained CNN for multi-layer feature extraction, PCA for dimensionality reduction, and FFT for spectral processing of image patches, DDF-Net fuses spatial (3D-CNN stream) and frequency-domain (CV-CNN stream) information efficiently. Leveraging its dual-branch architecture and SE blocks to enhance channel interdependencies, DDF-Net advances few-shot HSI classification performance.

Experiments on three benchmark datasets show that DDF-Net outperforms state-of-the-art (SOTA) few-shot methods under limited training samples (5–25 per class), validating its design effectiveness and providing new methodologies for HSI classification research. Future work will focus on optimizing architecture and parameters to further improve generalization and performance.

References

1. Hycza, T.; Stere ´nczak, K.; Bałazy, R. Potential Use of Hyperspectral Data to Classify Forest Tree Species. N. Z. J. For. Sci. 2018,48, 18.
2. Vaidya, R.; Nalawade, D.; Kale, K. Hyperspectral Imagery for Crop Yield Estimation in Precision Agriculture Using Machine Learning Approaches: A Review. Int. J. Creat. Res.Thoughts 2022, 9, a777–a789.
3. Suriguga; Bao, Y.; Bao, Y.; Jin, E. Application of Hyperspectral Remote Sensing in the Detection of Marine Oil Spill. Nat. Inn. Asia 2019, 4, 93–99.
4. Li, Y.; Tang, H.; Xie, W.; Luo, W. Multidimensional Local Binary Pattern for Hyperspectral Image Classification. IEEE Trans. Geosci. Remote Sens. 2022, 60, 1–13.
5. Dalla Mura, M.; Atli Benediktsson, J.; Waske, B.; Bruzzone, L. Extended Profiles with Morphological Attribute Filters for the Analysis of Hyperspectral Data. Int. J. Remote Sens. 2010, 31, 5975–5991.
6. Feng J, Liu L, Zhang X, et al. Hyperspectral image classification based on stacked marginal discriminative autoencoder[C]//2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017: 3668-3671.
7. Li J, Xi B, Li Y, et al. Hyperspectral classification based on texture feature enhancement and deep belief networks[J]. Remote Sensing, 2018, 10(3): 396.
8. Lee, H.; Kwon, H. Going Deeper with Contextual CNN for Hyperspectral Image Classification. IEEE Trans. Image Process. 2017, 26, 4843–4855.
9. Li G D, Zhang C J, Gao F, Zhang X Y . Doubleconvpool-structured 3D-CNN for hyperspectral remote sensing image classification[J]. Journal of Image and Graphics, 2019, 24(4): 639-654.
10. Zhao W, Du S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach[J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(8): 4544-4554.

11. He X, Chen Y, Lin Z. Spatial-spectral transformer for hyperspectral image classification[J]. Remote Sensing, 2021, 13(3): 498.
12. ChiYan Lee, Hideyuki Hasegawa, and Shangce Gao,“Complex-valued neural networks: A comprehensive survey,” IEEE/CAA Journal of Automatica Sinica, vol.9, no. 8, pp. 1406–1426, 2022.
13. F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” Remote Sens., vol. 7, no. 11, pp. 14680–14707, Nov. 2015.
14. Li Wang, Jiangtao Peng, and Weiwei Sun, “Spatial– spectral squeeze-and-excitation residual network for hyperspectral image classification,” Remote Sensing, vol. 11, no. 7, pp. 884, 2019.
15. Y. Liu, Y. Zhong, and Q. Qin, “Scene classification based on multiscale convolutional neural network,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 12, pp. 7109–7121, Dec. 2018.
16. Amina Ben Hamida, Alexandre Benoit, Patrick Lambert, and Chokri Ben Amar, “3-d deep learning approach for remote sensing image classification,” IEEE Transactions on geoscience and remote sensing, vol. 56, no. 8, pp. 4420–4434, 2018.
17. Feng, J.; Feng, X.; Chen, J.; Cao, X.; Zhang, X.; Jiao, L.; Yu, T. Generative Adversarial Networks Based on Collaborative Learning and Attention Mechanism for Hyperspectral Image Classification. Remote Sens. 2020, 12, 1149.
18. Li, Z.; Liu, M.; Chen, Y.; Xu, Y.; Li, W.; Du, Q. Deep Cross-Domain Few-Shot Learning for Hyperspectral Image Classification.IEEE Trans. Geosci. Remote Sens. 2022, 60, 1–18.
19. Hu, L.; Luo, X.; Wei, Y. Hyperspectral Image Classification of Convolutional Neural Network Combined with Valuable Samples.J. Phys. Conf. Ser. 2020, 1549, 052011.
20. Deng, B.; Jia, S.; Shi, D. Deep Metric Learning-Based Feature Embedding for Hyperspectral Image Classification. IEEE Trans.Geosci. Remote Sens. 2020, 58, 1422–1435.
21. Bai, J.; Lu, J.; Xiao, Z.; Chen, Z.; Jiao, L. Generative Adversarial Networks Based on Transformer Encoder and Convolution Block for Hyperspectral Image Classification. Remote Sens. 2022, 14, 3426.
22. Uchaev, D.; Uchaev, D. Small Sample Hyperspectral Image Classification Based on the Random Patches Network and Recursive Filtering. Sensors 2023, 23, 2499.