A Multi-branch Feature Fusion Model Based on Convolutional Neural Network for Hyperspectral Remote Sensing Image Classification

Jinli Zhang[1], Ziqiang Chen[2], Yuanfa Ji[3*], Xiyan Sun[4], Yang Bai[5*]
Information and Communication School, Guilin University of Electronic Technology, Guilin 541004, China[1, 2, 3, 4, 5]
Guangxi Key Laboratory of Precision Navigation Technology and Application, Guilin University of Electronic Technology, Guilin 541004, China[1, 3, 4, 5]
Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China[2]
GUET-Nanning E-Tech Research Institute Co., Ltd., Nanning 530031, China[4]

***Abstract*—Hyperspectral image classification constitutes a pivotal research domain in the realm of remote sensing image processing. In the past few years, convolutional neural networks (CNNs) with advanced feature extraction capabilities have demonstrated remarkable performance in hyperspectral image classification. However, the challenges faced by classification methods are compounded by the difficulties of "dimensional disaster" and limited sample distinctiveness in hyperspectral images. Despite existing efforts to extract spectral spatial information, low classification accuracy remains a persistent issue. Therefore, this paper proposes a multi-branch feature fusion model classification method based on convolutional neural networks to fully extract more effective and adequate high-level semantic features. The proposed classification model first undergoes PCA dimensionality reduction, followed by a multi-branch network composed of three-dimensional and two-dimensional convolutions. Convolutional kernels of varying scales are utilized for multi-feature extraction. Among them, the 3D convolution not only adapts to the cube of hyperspectral data but also fully exploits the spectral-spatial information, while the 2D convolution learns deeper spatial information. The experimental results of the proposed model on three datasets demonstrate its superior performance over traditional classification models, enabling it to accomplish the task of hyperspectral image classification more effectively.**

***Keywords*—Hyperspectral image classification; convolutional neural network (CNN); multi-branch network; feature fusion**

I. INTRODUCTION

Hyperspectral remote sensing is a cutting-edge technology that utilizes imaging spectrometry to remotely acquire the electromagnetic properties of objects, representing a revolutionary advancement in the area of remote sensing. The key to this technology lies in the utilization of a narrow and continuous spectral channel for remote sensing imaging of objects [1, 2], which can detect the two-dimensional spatial image and the third-dimensional spectral image of the object on earth at the same time and is a cube with the spectral and spatial information, and is also developed based on imaging and spectroscopy. Nowadays, hyperspectral imagery has found extensive application in the field of agriculture [3], military [4], chemistry [5], mineral identification [6], human health [7], and other fields, playing an indispensable role in the development and progress of human society. The objective of hyperspectral remote sensing image classification is to accurately categorize target ground objects [8], integrate the categories with actual ground object information, and obtain specific category information for the target region [9]. This field of study represents a specialized application of image classification within the realm of remote sensing. However, hyperspectral images are plagued by "dimension disaster" [9], "Hughes phenomenon" [10, 11], the limited quantity of labeled training samples [12], and the inequality of data sample types, which will make hyperspectral images encounter great hardships in the course of extracting features and performing classification.

In the initial exploration of hyperspectral image classification, researchers primarily focused on the spectral information contained within these images, which can effectively capture and reflect the internal mechanisms and chemical composition of ground objects. Specifically, traditional classification methods have harnessed the abundance of bands in hyperspectral images to execute machine learning algorithms for classification purposes with great efficacy, including random forest [13], decision trees [14], support vector machine [15] and K-nearest neighbor [16] algorithms. Relying solely on spectral information, these methods are capable of performing simple classification without the need for feature extraction. Meanwhile, the problem of data redundancy has led subsequent researchers to focus their attention on dimensionality reduction and feature extraction methods. As a preliminary step to classification, the primary techniques of dimensionality reduction can be classified into feature selection and feature extraction [17]. The aim of feature selection is to identify representative spectral information from redundant hyperspectral data while preserving as much original band information as possible [18, 19]. Commercial feature selection methods, including principal component analysis (PCA) [20], independent component analysis (ICA) [21], and linear discriminant analysis (LDA) [22], are commonly used in hyperspectral image processing. PCA method is the most favored linear dimensionality reduction technique. With the continuous advancement and widespread application of deep learning in image processing,

target detection, and speech recognition, it has become a crucial tool for hyperspectral image classification research[23], some typical deep neural network models mainly include stacked autoencoders (SAE) [24], deep belief networks (DBN) [25], and convolutional neural networks (CNN) [26]. Compared to machine learning methods, deep learning models possess a hierarchical structure that enables the extraction of high-level semantic information during feature extraction. This allows for better approximation of the nonlinear structure present in hyperspectral image data and enhances algorithmic effectiveness and robustness [27], thereby facilitating the extraction of complex and high-level features. So far, several deep learning-based approaches have been accomplished within the field of hyperspectral image classification. Just as the application of stacked encoders (SAEs) [28] in hyperspectral image classification. PCA is used to reduce the spectral dimensions of the original data and expanded the data into one-dimensional (1D) vectors as the input of SAE model. Finally, the hyperspectral images were classified by SVM classifier. In 2015, a hyperspectral image classification method based on deep belief network (DBN) was proposed, which also combined PCA method, and used the hierarchical feature extraction and logistic regression classifier to complete the classification of hyperspectral images [29]. As the mentioned two methods expand the spatial neighborhood into a 1D vector, which destroys the correlation of spatial information, they cannot effectively extract the spectral-spatial information to achieve high precision classification of hyperspectral images.

Fortunately, convolutional neural networks, another major branch of deep learning, have demonstrated superior performance in handling hyperspectral data due to their ability to directly address high dimensionality and automatically extract hierarchical image features compared to SAE and DBN [30, 31]. In 2015, the first hyperspectral image classification algorithm based on CNN was introduced. Despite utilizing only the spectral dimension information of the image, its initial application in hyperspectral classification demonstrated superiority over traditional methods such as Support Vector Machine (SVM) [32]. The authors in [33] augmented the number of samples by rotating labelled training data. However, their model's feature extraction process solely relies on spatial domain information while disregarding spectral dimension information. Spectral information is complementary and essential as it indicates that adjacent pixels may belong to the same class [33]. Then, the 1D-CNN+2D-CNN network [34] utilizes a double-branch structure to connect spectral features learned from one-dimensional CNN (1D-CNN) and spatial features learned from two-dimensional CNN (2D-CNN), extracting joint spectral-spatial features for classification. However, this method fails to consider the interdependence between spectral and spatial features. The appearance of three-dimensional convolution just solves the problem of the above model. Using three-dimensional (3D) convolution to work concurrently on information in 3D directions could be more appropriate for the dimensionality of hyperspectral data. A three-dimensional CNN (3D-CNN) model proposed by [35] does not perform any pre-processing on hyperspectral data and uses the full spectral band as input, which retains complete information but actually has high band-to-band correlation and much redundant information. In recent years, an eight-layer 3D-CNN network structure [36] was also proposed for hyperspectral image classification, in which the convolutional layer and the pooling layer were placed alternately.

In this paper, we propose a multi-branch feature fusion model based on CNN for extracting features from hyperspectral images and achieving ground object classification. The present study makes noteworthy contributions in the following aspects:

1) In this paper, a multi-branch feature fusion classification model is proposed for hyperspectral image classification. 3D convolution operations are preferentially used to process special hyperspectral 3D data and extract features from different degrees of spectral and spatial dimensions by utilizing different scales of convolution kernels and number of filters. In addition, 2D convolution was added after 3D convolution to reduce the complexity of the neural network while still efficiently extracting deeper spatial features. Meanwhile, PCA method is used to solve "curse of dimension" of the hyperspectral image.

2) The model framework presented adopts a multi-branch feature fusion structure to integrate features extracted from different branches. By connecting the features extracted from various branches using the Concatenate function, network features can be more comprehensively supplemented, thereby addressing issues of inadequate feature extraction and low precision associated with single branch models, ultimately leading to improved classification performance.

3) The proposed method's effectiveness is demonstrated on three datasets, and the results indicate that it outperforms several other classical methods. The experiments validate that multi-branch feature fusion can significantly enhance classification accuracy. Additionally, various experiments were conducted to determine the model parameters' effects, such as patch size, learning rate, percentage of training samples, and number of branches.

II. METHODS

The experimental study in this research primarily involves the acquisition of public hyperspectral datasets, preprocessing them, and randomly dividing them into training, validation, and testing sets. During model training, the validation set is utilized for verification to determine whether parameter retuning or training cessation is necessary. Finally, the test set input is used for prediction to obtain results. The specific classification process can be seen in Fig. 1.

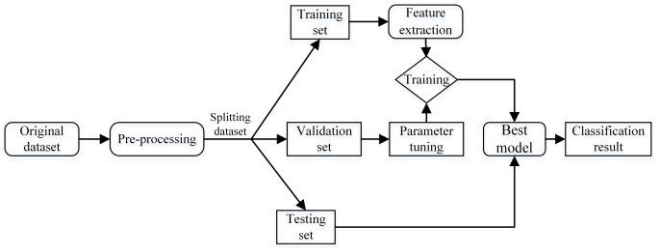

Fig. 1. The hyperspectral image classification flow chart.

A. Principal Component Analysis (PCA)

Hyperspectral images contain abundant spectral information, but a large number of bands exhibit strong correlation, leading to potential redundancy in the data. Therefore, employing PCA can effectively reduce dimensionality while retaining sufficient information for subsequent feature extraction and classification tasks. This approach not only saves time during model training and testing but also ensures that valuable information is preserved. Meanwhile, as sufficient information is preserved, the discarded band data is essentially superfluous and repetitive, thus exerting negligible influence on the ultimate classification outcomes. For further criteria and a comprehensive overview, refer to [37-39] and relevant literature therein.

To determine the appropriate number of principal components k, we analyzed the graph depicting the relationship between spectral information and the number of principal components after dimensionality reduction. The aim was to identify a value for k that would eliminate redundant bands while retaining most of the information in this experiment. Fig. 2 illustrates that even without PCA, the corrected removal of some noise bands does not result in a 100% retention of information across the three downloaded public datasets. After analyzing the outcome plots of these datasets following PCA and ensuring the proposed model's generality, we determined k to be 30.

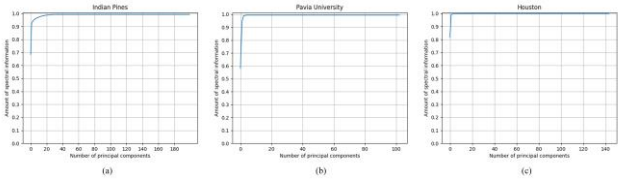

Fig. 2. Relationship between the number of principal components and the amount of retained spectral information.

B. Convolutional Neural Networks (CNN)

In 1989, LeCun introduced the concept of convolutional neural networks and proposed a multi-layer CNN model for handwritten digit recognition [40]. Since then, CNN, one of the typical feedforward deep neural network architectures, has seen extensive use in numerous computer vision domains. With its inherent advantages in local connectivity and weight sharing, the Convolutional Neural Network (CNN) has proven to be a powerful tool for image classification as well as other related fields. Consisting of an input layer, multiple hidden layers, and an output layer, deep CNN are capable of extracting features at different levels with remarkable efficacy.

As the most crucial operation in CNN, convolution realizes feature extraction of input data by utilizing various convolution kernels to perform sliding pixel extraction on the input image matrix. The nonlinear structure of activation function is then employed to enhance the similarity between image features and real features.

In 2D-CNN [41], both the convolutional kernel and input are in 2D format. When applied to hyperspectral image classification, the network typically takes the neighborhood block surrounding a center pixel as input, with the label of said center pixel serving as that of the entire block. 2D convolution can effectively utilize neighborhood information, fuse the features of neighborhood samples, and extract spatial information. Its basic principle is to carry out weighted summation of image center pixel and neighborhood pixel according to the weights of convolution kernels and use the output of activation function as the value of center pixel. The output of jth feature map at (x, y) of the ith layer can be expressed as [42]:

$$v_{ij}^{xy} = f\left(b_{ij} + \sum_{k=1}^{m}\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} w_{ijk}^{pq} \cdot v_{(i-1)k}^{(x+p)(y+q)}\right) \quad (1)$$

Where in (1), f is the activation function, w and b are the weight and bias of the jth feature graph in the ith layer, respectively.

The 3D-CNN [41] is an extension of the 2D-CNN, where convolution is performed along three dimensions of input data simultaneously. This means that convolution is not only carried out in the height and width directions but also in the spectral channel. The output of the jth feature graph at (x, y, z) of the ith layer can be obtained by the formula [42]:

$$v_{ij}^{xyz} = f\left(b_{ij} + \sum_{k=1}^{m}\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1}\sum_{r=0}^{R-1} w_{ijk}^{pqr} \cdot v_{(i-1)k}^{(x+p)(y+q)(z+r)}\right) (2)$$

Same as (1), f is the activation function, w and b are the weight and bias of the jth feature graph in the ith layer, respectively.

In the convolution operation, utilizing an activation function can enhance the nonlinearity of the network. In this study, ReLU activation function[30] is employed for both convolution and full connection layers, while SoftMax classification function is exclusively used for final output classification layer. The formula is presented as follows:

$$f(x) = \max(0, x) \quad (3)$$

The ReLU function is relatively simple compared to other functions, yet it boasts faster operational efficiency and convergence speed. Consequently, it is widely utilized in deep learning models due to its ease of obtaining the required model.

C. The proposed CNN Classification Model

In this research, a multi-branch feature fusion model based on CNN for extracting more profound and expressive spectral-spatial features of hyperspectral remote sensing images was discussed. To extract both spectral and spatial features from hyperspectral images, 3D convolutional operations are given priority to achieve this goal. This entails the utilization of convolutional kernels with varying scales and numbers to effectively capture features of different degrees, ensuring a comprehensive and efficient feature extraction process. As such, employing a network solely consisting of 3D convolution operations presents challenges in directly computing 3D data, resulting in excessive hyperparameters and prolonged feature extraction time due to its complexity. To augment the model's capacity for extracting spatial information features from data while simultaneously mitigating its complexity, the 3D data

resulting from convolution is transformed into simpler 2D flat data and subsequently subjected to additional 2D convolution operations.

On one hand, the conventional methods for enhancing the classification performance of the entire model involve deepening it, such as augmenting the number of convolutional layers. However, this often results in an increase in training parameters and complexity, as well as higher computational costs. While the classification performance may become better with a deeper network structure, the training difficulty also becomes greater. With consideration of these factors, this paper adopts a multi-branch convolutional neural network structure for the reason of improving the classification performance of the model as much as possible without increasing the overall model complexity and network depth. On the other hand, during the feature extracting procedure, the textural elements such as edge background of the hyperspectral image are mainly extracted by the low-level network, the regions of the image are extracted in the middle-level network, and the overall feature is partially extracted by the upper-level network, consequently, some essential feature contents are lost in the convolution process, which affects the final classification accuracy. In this paper, a multi-branch feature fusion approach is employed, whereby the same hyperspectral data is fed into multiple branches for processing and obtaining multi-scale feature information. Subsequently, the information obtained from each convolutional layer is integrated together. Compared to a single-branch structure, this architecture can capture a more diverse and comprehensive range of information by incorporating low-, middle-, and high-level features, thereby enhancing the overall classification performance of the model. Finally, a Dropout layer is appended after the fully connected layer to forestall overfitting. The specific model framework is illustrated in Fig. 3.

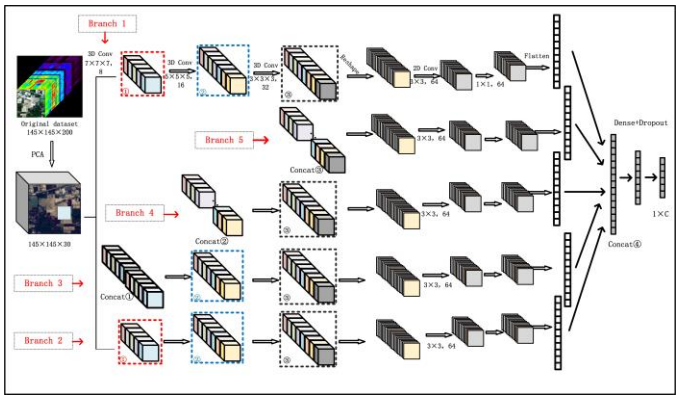

Fig. 3. The proposed network framework.

The proposed network framework is demonstrated using the representative Indian Pines dataset. The first stage of classification involves pre-processing the original hyperspectral dataset, wherein the spectral dimensionality is reduced, and redundant information is eliminated by applying PCA to reduce 200 spectral bands to 30. Afterwards, the reduced-dimensional dataset is fed into small cubes of size s × s × 30 and slid from left to right and top to bottom as input for branch 1 and branch 2 of the same design in a convolutional network. These branches utilize three distinct 3D convolutional layers with kernel sizes of 7×7×7, 5×5×5, and 3×3×3 respectively, along with additional 2D convolutions using kernels of size 3×3 and 1×1. As illustrated in Fig. 3, the output of the first convolution layer of the two branches, which is the feature map marked as number ① in Fig. 2, is connected together with the Concatenate function, and it is marked as Concat ①, and it is used as the input of branch 3. The two 3D convolution layers of the third branch are designated as 5×5×5 and 3×3×3, respectively. 2D convolution kernel size is 3×3 and 1×1, and the fourth branch in the same way.

The three output feature maps marked with number ② in the figure are also connected and marked as Concat ②, which is the input of branch 4, consisting of the second output feature map of branch 1 and branch 2 and the first output feature map of branch 3. The input of branch 5 is Concat ③, which is obtained by connecting the four outputs numbered ③. In this model, all branches finalize with two two-dimensional convolutional layers. Eventually, the one-dimensional vectors obtained by flattening all the branches are connected together again with the fully connected layer and the Dropout layer in turn. The extracted features are multi-classified and compared with the actual ground object map using the Softmax function in the final fully linked layer. The parameters of the complete model framework branches and convolutional layers are shown in Table I.

TABLE I. NETWORK STRUCTURES

Input	*Hidden Layer*	*Kernel Size*	*Filters*
25×25×30,1	Conv3D_branch1_1, Conv3D_branch2_1	7×7×7	8
19×19×24,8 (19×19×24,16)	Conv3D_branch1_2, Conv3D_branch2_2, (Conv3D_branch3_1)	5×5×5	16
15×15×20,16 (15×15×20,48)	Conv3D_branch1_3, Conv3D_branch2_3, Conv3D_branch3_2 (Conv3D_branch4_1)	3×3×3	32
13×13,576 (13×13,2304)	Conv2D_branch1_4 Conv2D_branch2_4, Conv2D_branch3_3 Conv2D_branch4_2 (Conv2D_branch5_1)	3×3	64
11×11,64	Conv2D_branch1_5 Conv2D_branch2_5, Conv2D_branch3_4 Conv2D_branch4_3 Conv2D_branch5_2	1×1	64

III. EXPERIMENT

The datasets, performance measurements, and experimental setting used in this work are briefly described in this section. It includes the partitioning of three publicly available datasets - Indian Pines, Pavia University, and Houston; as well as an explanation of the three objective evaluation metrics used in our experiments: OA, AA, and Kappa coefficients. Finally, we give a thorough explanation of the experimental setup and variables used in this research.

A. Datasets

To execute the proposed model, three hyperspectral image datasets were utilized, namely Indian Pines, Pavia University and Houston, which differ in terms of band number, pixel count, feature classes and spatial resolution.

1) Indian Pines(IP): The initial dataset comprises of Indian pine trees, captured by the infrared imaging spectrometer sensor AVRIS in northwestern Indiana, USA. This image boasts a total of 220 bands, with 20 noise bands being eliminated to enhance its quality. Each individual band has a pixel size of 145 × 145 and spatial resolution of 20 meters. It encompasses an impressive array of 16 feature species. In this paper, each feature class found in the Indian Pines dataset is painstakingly split into a training set, a validation set, and a testing set in the ratios of 1:1:8 in this research.

2) Pavia University(PU): The second dataset captured by the ROSIS sensor over the University of Pavia is a stunning hyperspectral remote sensing image measuring 512×614 with an impressive spatial resolution of 1.3 m. The image was imaged continuously in the wavelength range of 0.43-0.86 μm, and after removing noise12 bands that were severely affected by noise, the remaining 103 bands were used for classification. The dataset contains a total of nine categories of real features for classification, in the experiments of this paper 3% of the samples are selected as the training set and 3% as the validation set and the rest are used for testing.

3) Houston (HT): The Houston dataset is acquired by the ITRES CASI-1500 sensor for ground feature information on the University of Houston campus and adjacent urban areas, and it is provided by the 2013 IEEE GRSS Data Fusion Competition with a spatial resolution of 2.5 meters. It contains 349 x 1905 pixels, and this hyperspectral image consists of 144 spectral bands in the range of 380 nm to 1050 nm and contains 15 feature classes. As with the Indian Pine dataset, each class of features is divided into training set, validation set, and testing sets in the ratio of 1:1:8.

B. Classification Index

To evaluate the effectiveness of our proposed model accurately and scientifically for classification, subjective perception alone is insufficient. Therefore, this paper employs evaluation indices including Overall Accuracy, Average Accuracy, Kappa coefficient [45].

- Overall Accuracy (OA): refers to the proportion of correctly classified test samples to the total number of test samples, reflecting the precision and effectiveness of classification performance.
- Average Accuracy (AA): refers to the ratio of the sum of classification accuracy of each type of ground object in hyperspectral images to the number of ground object classes.
- Kappa coefficient: It is an evaluation index used to test the consistency. It is used to test the consistency between the actual results and the predicted results in hyperspectral images. Its value is generally between -1 and 1, and generally greater than 0.

C. Experimental Environment

To verify the effectiveness of the proposed algorithm, this study was conducted in Python 3.8 environment with code written in TensorFlow framework and experiments on three publicly available datasets, including Indian Pines, Pavia University, and Houston. All experiments were run on NVIDIA RTX A6000 GPU servers.

IV. Experimental Parameters and Results

In this section, we first conduct experiments on the model's parameter settings, which involve adjusting parameters such as learning rate and epoch based on validation set results. We also analyse the effects of various settings on experimental outcomes, including training sample ratios across three datasets, input spatial region size for the model, and number of branches in the proposed multi-branch feature fusion model. When all the parameters were set, the results of the method proposed in this paper were analysed and compared with seven other different methods, including SVM[43], 1D-CNN[32], CDCNN[44], 3D-CNN[36], HybridSN[45], M3D-DCNN[46] and DBMA[47]. All the methods run in the same environment and use the same number of training set samples.

A. Experimental Parameters Setting and Analysis

In this section, the primary objective is to optimize network parameters and determine the optimal configuration for the classification network by comparing experimental results, in order to achieve superior classification results. We ran experiments on three different datasets and picked the best network parameters after comparing them all. This resulted in the most accurate classification results for our network. The following comparative analysis presents learning rate, epochs, spatial size, training set ratio, and number of branches respectively.

1) Learning rate and epochs: It is crucial to ascertain the suitable learning rate for the model during training, as it is arguably the most critical hyperparameter to configure. The learning rate represents the magnitude of each parameter update in the network and dynamically adjusts during training with changes in epoch, following a specific update formula:

$$\omega_{i+1} = \omega_i - lr\nabla \tag{4}$$

Where ω_{i+1} and ω_i are the weight values of the $i+1$st epoch and the ith epoch, respectively. lr is the learning rate and ∇ is the decay exponent.

As depicted in Fig. 4(a), with Adam optimizer and an initial learning rate of 0.001, both training accuracy and validation accuracy gradually improve as the number of epoch increases. The rising trend of both accuracies is consistent with the convergence trend, but there are certain fluctuations during the intermediate process, resulting in less stability. In Fig. 4(b), the relationship between epoch time and loss function is depicted, where an increase in epoch time leads to a gradual decrease and convergence of the loss function; however, it is evident that there exists a significant degree of fluctuation.

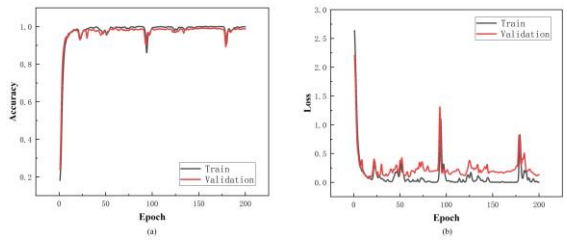

Fig. 4. Learning rate equals to 0.001. (a): Training and validation accuracy curves. (b): Training and validation of loss function curves.

With a reduced learning rate of 0.0005, the results depicted in Fig. 5 demonstrate faster convergence of both training and validation accuracy with smaller fluctuations, as well as quicker convergence of the loss function with less fluctuation compared to a learning rate of 0.001.

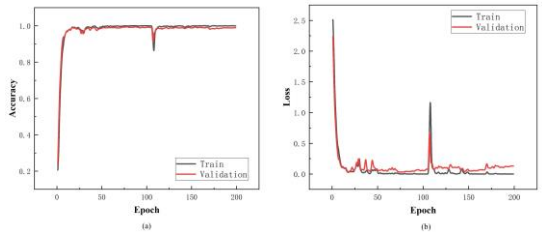

Fig. 5. Learning rate equals to 0.0005. (a): Training and validation accuracy curves. (b): Training and validation of loss function curves.

Furthermore, adjusting the learning rate to 0.0001 results in a clear convergence of the training and validation accuracy curves, with minimal fluctuations and improved overlap as shown in Fig. 6. This indicates a more stable convergence of the loss function. Therefore, following a comprehensive analysis of the relationship between the three learning rates and epochs, we have selected a more effective learning rate of 0.0001 and an epoch of 200 for experimentation in this paper. Based on these findings, we have set the optimizer's learning rate to 0.001 and the epoch to 200.

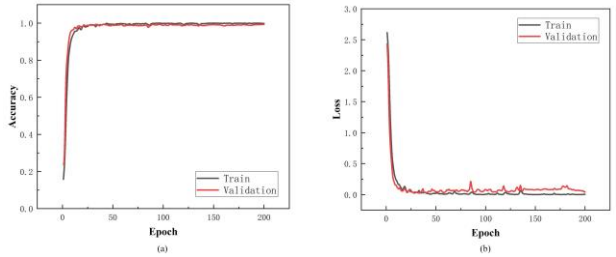

Fig. 6. Learning rate equals to 0.0001. (a): Training and validation accuracy curves. (b): Training and validation of loss function curves.

2) Spatial size: The spatial size refers to the dimension of the input sample after segmentation and dimensionality reduction of small cubes. For 3D CNN classification, the input data size is a crucial parameter that affects feature extraction and classification performance. Increasing spatial size captures more information but also introduces redundancy, which may affect final classification results. The experiments were conducted by setting 15×15、17×17、19×19、21×21、23×23、25×25、27×27、29×29.

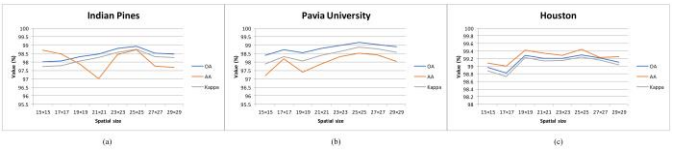

Fig. 7. OA, AA, and Kappa values of three datasets at different spatial sizes. (a): Indian Pines. (b): Pavia University. (c): Houston.

The Fig. 7 shows the results of our experiments on the three datasets. As the space size increases, the OA, AA, and Kappa of the three datasets also increase. After the space size is larger than 25, the classification effect of the three datasets suddenly becomes worse, probably because the selected space is too large leading to more spatial contextual information, which brings redundancy leading to misclassification. And when the $spatial\ size = 25 \times 25$, the datasets Indian Pines, Pavia University and Houston all reach the highest OA values of 98.92%, 99.16% and 99.30%, respectively. Therefore, combining the experimental results of the three datasets, the optimal parameter $spatial\ size = 25 \times 25$ was chosen in this study.

3) Training set and ratio: Deep learning-based classification models are highly reliant on the ratio of training samples. Generally, adding samples leads to improved performance in both training and testing. However, a significant challenge with hyperspectral data is the less labeled training samples. Furthermore, augmenting the size of the training dataset also results in prolonged training durations, which adversely affects model performance. Bearing these factors in mind, we will examine the impact of training set occupancy on classification outcomes. For the two datasets Indian Pines and Houston, training sets were used with 1%, 3%, 5%, 10%, 15%, 20%, 25%, and 30%, respectively; whereas for the larger dataset Pavia University, training sets with 0.1%, 0.5%, 1%, 3%, 5%, 10%, 15%, and 20% were used ratios were performed for the test.

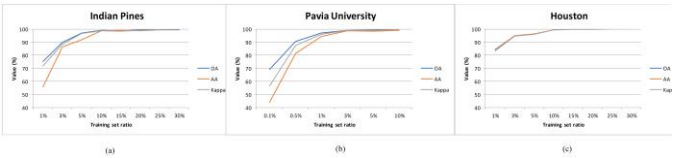

Fig. 8. OA, AA, Kappa values of three datasets with different training set ratio. (a): Indian Pines. (b): Pavia University. (c): Houston.

From the Fig. 8, it is observed that as the training sample gradually increases, the OA, AA, and Kappa predicted by the classification model also improve, and when the training sample reaches 10%, the three evaluation indexes OA, AA, and Kappa of Indian Pines are 98.92%, 98.72%, and 98.76%, respectively, and the classification results of Houston were 99.30%, 99.44%, and 99.24%. In the case of Pavia University, owing to its large sample size, its OA, AA, and Kappa reached 99.16%, 98.53%, and 98.89%, respectively, when 3% was used for the training ratio, and the classification results were already better. In summary, our goal of minimizing the number of training samples and avoiding lengthy training time, was achieved by setting the training sample ratio to 10% for both

Indian Pines and Houston datasets, and 3% for Pavia University.

4) Number of branches: To validate the efficacy of branches in the proposed CNN, a series of comparative models have been devised to determine the optimal number of branches by assessing their impact on classification accuracy. The proposed model consists of five branches, Branch 1, Branch 2, Branch 3, Branch 4, and Branch 5 labelled in the network framework. This analysis explores the efficacy of branching in feature extraction and classification by examining the correlation between branch quantity and final classification outcomes.

In Fig. 9, where the meanings of the horizontal axes from 1 to 5, respectively, are:

1) $(Branch1)$
2) $(Branch1 \oplus Branch2)$
3) $(Branch1 \oplus Branch2 \oplus Branch3)$
4) $(Branch1 \oplus Branch2 \oplus Branch3 \oplus Branch4)$
5) $(Branch1 \oplus Branch2 \oplus Branch3 \oplus Branch4 \oplus Branch5)$

where the symbol $\oplus$ denotes the Concatenate operation.

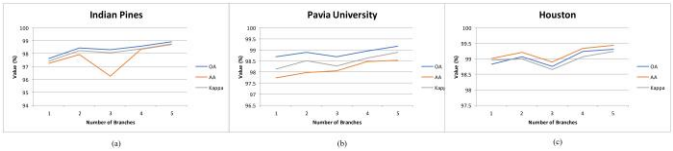

Fig. 9. OA, AA, Kappa values for three datasets with different number of branches. (a): Indian Pines. (b): Pavia University. (c): Houston.

As depicted in Fig. 9, the correlation between the number of branches in feature fusion and the three evaluation metrics (OA, AA, and Kappa) across three public datasets indicates that an increase in extracted features leads to higher OA, AA, and Kappa scores and improved classification performance. In this study, after analysing three sets of data and considering the number of branches and final results, the model ultimately selected five series branches to improve classification accuracy and enhance model robustness.

B. Results

Table II to Table III and Fig. 10 to Fig. 12 show the results of three different datasets in seven classification methods, including the results of OA, AA and Kappa. In addition, Fig. 13 shows the confusion matrix for the three datasets acquired in this paper.

Table II and Fig. 10 reveal that SVM[43] exhibits the poorest classification results when using a training set of only 10% from the Indian Pine dataset, whereas DBMA[47] and HybridSN[45] demonstrate superior experimental outcomes in terms of classification accuracy compared to other methods. Amongst the seven compared methods, DBMA[47] stands out with its exceptional performance. Compared to DBMA[47], the proposed methods in this paper exhibit significant improvements in OA, AA and Kappa. Specifically, the classification accuracy of all types of ground objects is basically improved, eventually, OA is improved by about 0.88%, AA is improved by about 1.02%, and Kappa is improved by about 1.00%. It is worth mentioning that among the 16 features in Indian Pines, the classification results of five categories of features reached 100% under the classification method proposed in this paper.

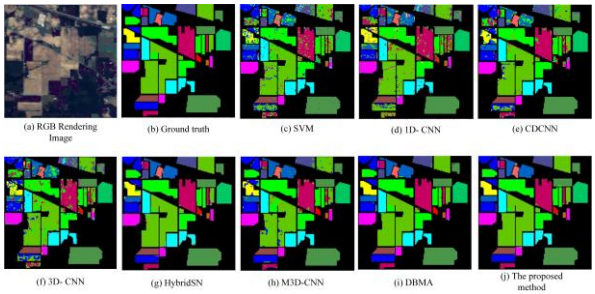

Fig. 10. Classification results of Indian Pines scenes using different methods.

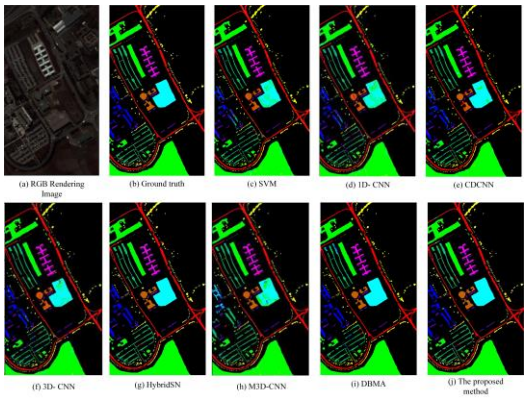

Fig. 11. Classification results of Pavia University scenes using different methods.

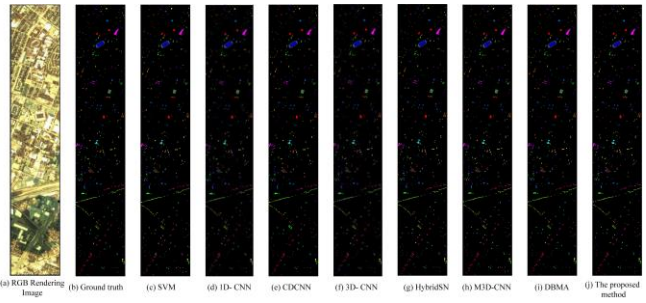

Fig. 12. Classification results of Houston scenes using different methods.

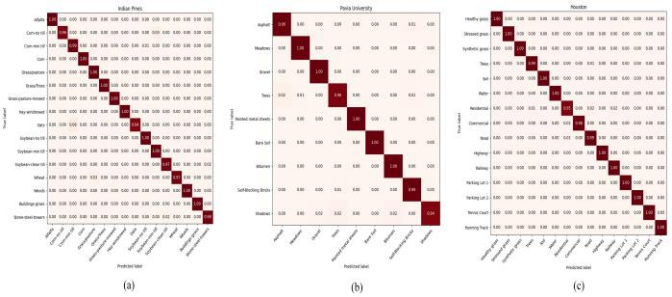

Fig. 13. Confusion matrix of the proposed method for the three datasets. (a): Indian Pines. (b): Pavia University. (c): Houston.

TABLE II. CLASSIFICATION ACCURACY OF INDIAN PINES AND HOUSTON

Classes	Accuracy (%): Indian Pines (IP), Houston (HT)															
	SVM		*1D-CNN*		*CDCNN*		*3D-CNN*		*HybridSN*		*M3D-DCNN*		*DBMA*		*Ours*	
Dataset	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***	***IP***	***HT***
1	66.67	97.69	63.41	97.25	61.90	94.04	65.85	96.00	95.12	96.54	34.15	98.85	100.00	98.68	100.00	99.73
2	71.76	98.14	70.66	98.58	81.28	91.42	81.17	99.38	96.03	100.00	81.40	99.20	99.03	99.40	96.11	99.91
3	74.12	99.52	55.42	99.68	82.62	98.07	64.26	95.37	97.32	100.00	85.54	100.00	99.24	100.00	98.66	99.84
4	70.37	93.21	82.63	97.59	83.80	98.84	58.69	97.23	91.55	98.12	77.00	94.20	95.52	99.50	99.53	99.38
5	90.34	97.67	87.82	97.05	96.82	99.70	83.22	99.82	99.08	100.00	94.94	99.91	94.80	97.28	99.54	100.00
6	89.28	98.29	96.35	98.63	99.31	100.00	93.30	84.25	99.85	97.60	98.63	94.52	99.31	99.26	99.54	100.00
7	85.71	90.36	80.00	82.82	88.89	96.42	60.00	87.99	100.00	96.06	64.00	92.20	90.91	98.91	100.00	95.44
8	87.94	83.84	99.53	83.21	91.81	95.08	95.35	86.61	100.00	92.59	100.00	80.98	99.74	100.00	100.00	98.12
9	55.56	80.48	66.67	84.65	82.35	94.81	38.89	89.62	100.00	95.39	83.33	90.06	100.00	97.41	94.44	99.47
10	75.32	90.22	80.80	82.16	83.08	83.75	80.69	77.08	96.69	99.82	73.60	93.48	98.45	99.70	99.89	100.00
11	78.51	78.42	88.01	77.25	85.71	89.48	82.31	86.78	98.19	100.00	85.48	77.79	97.54	98.90	99.50	100.00
12	75.78	79.55	72.85	69.55	75.41	95.22	68.73	81.35	98.31	96.04	83.52	87.39	97.41	95.68	97.38	99.64
13	89.50	37.20	99.46	32.94	99.40	94.61	95.68	64.22	100.00	94.79	98.92	89.10	100.00	98.94	97.30	100.00
14	92.16	96.88	96.22	96.36	95.81	100.00	93.77	95.58	99.47	100.00	98.68	99.22	98.34	100.00	100.00	100.00
15	70.10	99.66	61.38	99.83	89.00	98.16	69.16	98.32	99.14	100.00	79.25	100.00	94.25	98.34	100.00	100.00
16	98.57		90.48		93.59		100.00		96.43		100.00		98.63		97.62	
OA	**80.55**	**88.75**	**82.48**	**87.02**	**87.47**	**94.41**	**81.77**	**90.03**	**97.98**	**97.67**	**87.05**	**92.42**	**98.04**	**98.66**	**98.92**	**99.30**
AA	**79.79**	**88.08**	**80.73**	**86.50**	**86.92**	**95.31**	**76.94**	**89.31**	**97.95**	**97.80**	**83.65**	**93.13**	**97.70**	**98.80**	**98.72**	**99.44**
Kappa	**77.72**	**87.82**	**79.90**	**85.95**	**85.67**	**93.96**	**79.19**	**89.21**	**97.70**	**97.48**	**85.20**	**91.80**	**97.76**	**98.55**	**98.76**	**99.24**

TABLE III. CLASSIFICATION ACCURACY OF PAVIA UNIVERSITY

Classes	Accuracy (%): Pavia University (PU)							
	SVM	*1D-CNN*	*CDCNN*	*3D-CNN*	*HybridSN*	*M3D-DCNN*	*DBMA*	*Ours*
Dataset	***PU***	***PU***	***PU***	***PU***	***PU***	***PU***	***PU***	***PU***
1	92.06	92.88	92.92	93.11	97.51	94.56	98.12	98.52
2	97.94	96.98	97.75	98.01	99.95	99.31	99.90	99.86
3	72.74	83.06	89.44	91.01	97.69	78.88	90.70	99.51
4	93.78	85.77	97.81	89.70	92.50	94.68	95.94	95.63
5	99.46	99.46	100.00	99.16	99.92	100.00	99.61	100.00
6	83.68	75.34	91.30	88.60	100.00	72.67	100.00	99.94
7	85.04	78.22	94.07	82.64	100.00	82.64	100.00	100.00
8	89.83	82.50	91.26	89.36	97.31	98.40	99.30	99.08
9	100.00	99.67	99.44	90.42	82.05	99.78	99.54	94.23
OA	92.81	90.62	95.26	93.85	98.31	93.54	98.78	99.16
AA	90.50	88.21	94.89	91.34	96.32	91.21	98.12	98.53
Kappa	90.42	87.43	93.72	91.83	97.76	91.32	98.39	98.89

Based on Table III and Fig. 11, it is evident that among the six compared methods for the Pavia University dataset with only 3% training samples, 1D-CNN[32] exhibits the poorest classification performance while DBMA[47] remains superior in terms of classification accuracy. Furthermore, our proposed algorithm has demonstrated significant improvement compared to DBMA. OA, AA, and Kappa reached 99.16%, 98.53%, and 98.89%. Compared with the DBMA[47] algorithm, our method increases by about 0.38%, 0.41% and 0.50% for OA, AA and Kappa, respectively.

Table II and Fig. 12 present the experimental findings for the Houston dataset with a training sample of only 10%. It is evident that Houston's classification performance in 1D-CNN[32] was subpar, while it excelled in DBMA[47]. It is worth mentioning that the results obtained using the proposed method show that seven of the fifteen classes of objects in the Houston dataset can achieve 100% classification accuracy, and compared with the best method DBMA[47], OA, AA and Kappa respectively increased by about 0.64%, 0.64% and 0.69%. Although the improvement results are modest, the classification metrics have reached 99.30%, 99.44%, and 99.24%.

From the above experimental results, we can see that DBMA [47] shows great superiority among the seven compared methods, which is the result of the development and application of the attention mechanism in recent years. Meanwhile, SVM [43] and 1D-CNN [32] exhibit poor results, confirming that the idea of using only spectral information is not enough, and the 3D convolution operation proposed in this paper is designed to make good use of both spatial and spectral information to improve accuracy. Therefore, it also enlightens

us to research and study the attention mechanism in the field of hyperspectral remote sensing image classification afterwards. And the model still has shortcomings and still needs to be studied and improved in depth. For the characteristics of small training samples of hyperspectral remote sensing data markers, the use of semi-supervised and unsupervised methods for classification in subsequent experiments is also one of the research directions. Moreover, with the development of attention mechanisms, the ability to suppress unimportant information as another improved feature of the model is one of the main points for continued learning in the future.

V. Conclusions

A hyperspectral remote sensing image classification method based on multi-branch feature fusion is proposed in this paper to effectively extract spectral-spatial features of hyperspectral images and achieve efficient classification of ground objects. The proposed method, composed of multiple branches, yields more comprehensive and accurate extracted features. The 2D convolution layers are added to reduce the complexity brought by the 3D convolution, which makes the network not only more concise but also more deeply to extract spatial information. The experimental comparison results also demonstrate the preeminence of the proposed model over other methods, surpassing not only traditional classification techniques but also exhibiting significant advancements compared to other deep learning approaches. To sum up, the model approach proposed in this paper yields excellent classification outcomes across most datasets. Not only does it leverage 3D convolutional layers to simultaneously extract spectral-spatial features, but also reduces network complexity through the inclusion of 2D convolutional layers. Moreover, the multi-branch feature fusion structure enhances feature extraction adequacy and improves classification accuracy.

VI. Funding Statement

The research was financially supported by the Guangxi Key Laboratory of Precision Navigation Technology and Application at Guilin University of Electronic Technology, under grant number DH202208.

References

[1] W. Lv, and X. J. J. o. S. Wang, “Overview of hyperspectral image classification,” vol. 2020, 2020.

[2] H. L. Lu, H. J. Su, J. Hu, and Q. Du, “Dynamic Ensemble Learning With Multi-View Kernel Collaborative Subspace Clustering for Hyperspectral Image Classification,” Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 2681-2695, 2022.

[3] B. Lu, P. D. Dao, J. Liu, Y. He, and J. J. R. S. Shang, “Recent advances of hyperspectral imaging technology and applications in agriculture,” vol. 12, no. 16, pp. 2659, 2020.

[4] E. M. Winter, "Detection of surface mines using hyperspectral sensors." pp. 1597-1600.

[5] Y.-N. Chen, D.-W. Sun, J.-H. Cheng, and W.-H. J. F. E. R. Gao, “Recent advances for rapid identification of chemical information of muscle foods by hyperspectral imaging analysis,” vol. 8, pp. 336-350, 2016.

[6] Z. Ting-ting, and L. Fei, "Application of hyperspectral remote sensing in mineral identification and mapping." pp. 103-106.

[7] M. Cihan, M. Ceylan, and A. H. J. S. L. Ornek, “Spectral-spatial classification for non-invasive health status detection of neonates using hyperspectral imaging and deep convolutional neural networks,” vol. 55, no. 5, pp. 336-349, 2022.

[8] X. F. Shen, W. X. Bao, H. B. Liang, X. W. Zhang, and X. Ma, “Grouped Collaborative Representation for Hyperspectral Image Classification Using a Two-Phase Strategy,” Ieee Geoscience and Remote Sensing Letters, vol. 19, pp. 5, 2022.

[9] L. M. Bruce, C. H. Koger, J. J. I. T. o. g. Li, and r. sensing, “Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction,” vol. 40, no. 10, pp. 2331-2338, 2002.

[10] Z. Yang, L. Zhi-Xin, J. J. J. o. I. o. S. Han, and Mapping, “The hughes phenomenon in hyperspectral analysis and the application of the lowpass filter,” 2004.

[11] Z. H. Xue, X. Y. Nie, and M. X. Zhang, “Incremental Dictionary Learning-Driven Tensor Low-Rank and Sparse Representation for Hyperspectral Image Classification,” Ieee Transactions on Geoscience and Remote Sensing, vol. 60, pp. 19, 2022.

[12] A. Z. Zhang, Z. J. Pan, H. Fu, G. Y. Sun, J. Rong, J. C. Ren, X. P. Jia, and Y. J. Yao, “Superpixel Nonlocal Weighting Joint Sparse Representation for Hyperspectral Image Classification,” Remote Sensing, vol. 14, no. 9, pp. 19, May, 2022.

[13] C. E. Priebe, D. J. Marchette, D. M. J. I. T. o. P. A. Healy, and M. Intelligence, “Integrated sensing and processing decision trees,” vol. 26, no. 6, pp. 699-708, 2004.

[14] J. Xia, P. Ghamisi, N. Yokoya, A. J. I. T. o. G. Iwasaki, and R. Sensing, “Random forest ensembles and extended multiextinction profiles for hyperspectral image classification,” vol. 56, no. 1, pp. 202-216, 2017.

[15] K. S. Ettabaa, M. A. Hamdi, and R. B. Salem, "SVM for hyperspectral images classification based on 3D spectral signature." pp. 42-47.

[16] L. Ma, M. M. Crawford, J. J. I. T. o. G. Tian, and R. Sensing, “Local manifold learning-based $ k $-nearest-neighbor for hyperspectral image classification,” vol. 48, no. 11, pp. 4099-4109, 2010.

[17] B. B. Damodaran, N. Courty, S. J. I. T. o. G. Lefèvre, and R. Sensing, “Sparse Hilbert Schmidt independence criterion and surrogate-kernel-based feature selection for hyperspectral image classification,” vol. 55, no. 4, pp. 2385-2398, 2017.

[18] C. Persello, L. J. I. t. o. g. Bruzzone, and r. sensing, “Kernel-based domain-invariant feature selection in hyperspectral images for transfer learning,” vol. 54, no. 5, pp. 2615-2626, 2015.

[19] N. Audebert, B. Le Saux, S. J. I. g. Lefèvre, and r. s. magazine, “Deep learning for classification of hyperspectral data: A comparative review,” vol. 7, no. 2, pp. 159-173, 2019.

[20] X. Kang, X. Xiang, S. Li, J. A. J. I. T. o. G. Benediktsson, and R. Sensing, “PCA-based edge-preserving features for hyperspectral image classification,” vol. 55, no. 12, pp. 7140-7151, 2017.

[21] A. Villa, J. A. Benediktsson, J. Chanussot, C. J. I. t. o. G. Jutten, and r. sensing, “Hyperspectral image classification with independent component discriminant analysis,” vol. 49, no. 12, pp. 4865-4876, 2011.

[22] S. Yuan, X. Mao, and L. J. I. T. o. I. P. Chen, “Multilinear spatial discriminant analysis for dimensionality reduction,” vol. 26, no. 6, pp. 2669-2681, 2017.

[23] H. B. Liang, W. X. Bao, X. F. Shen, and X. W. Zhang, “HSI-Mixer: Hyperspectral Image Classification Using the Spectral-Spatial Mixer Representation From Convolutions,” Ieee Geoscience and Remote Sensing Letters, vol. 19, pp. 5, 2022.

[24] C. Tao, H. Pan, Y. Li, Z. J. I. G. Zou, and r. s. letters, “Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification,” vol. 12, no. 12, pp. 2438-2442, 2015.

[25] P. Zhong, Z. Gong, S. Li, C.-B. J. I. T. o. G. Schönlieb, and R. Sensing, “Learning to diversify deep belief networks for hyperspectral image classification,” vol. 55, no. 6, pp. 3516-3530, 2017.

[26] A. Krizhevsky, I. Sutskever, and G. E. J. C. o. t. A. Hinton, “Imagenet classification with deep convolutional neural networks,” vol. 60, no. 6, pp. 84-90, 2017.

[27] X. Zhang, Y. Wang, N. Zhang, D. Xu, H. Luo, B. Chen, and G. J. I. A. Ben, “SSDANet: Spectral-spatial three-dimensional convolutional neural network for hyperspectral image classification,” vol. 8, pp. 127167-127180, 2020.

[28] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. J. I. J. o. S. t. i. a. e. o. Gu, and r. sensing, "Deep learning-based classification of hyperspectral data," vol. 7, no. 6, pp. 2094-2107, 2014.

[29] Y. Chen, X. Zhao, X. J. I. j. o. s. t. i. a. e. o. Jia, and r. sensing, "Spectral–spatial classification of hyperspectral data based on deep belief network," vol. 8, no. 6, pp. 2381-2392, 2015.

[30] X. Liu, Q. Sun, Y. Meng, C. Wang, and M. Fu, "Feature extraction and classification of hyperspectral image based on 3D-convolution neural network." pp. 918-922.

[31] X. Tan, and Z. X. Xue, "Spectral-spatial multi-layer perceptron network for hyperspectral image land cover classification," European Journal of Remote Sensing, vol. 55, no. 1, pp. 409-419, Dec, 2022.

[32] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. J. J. o. S. Li, "Deep convolutional neural networks for hyperspectral image classification," vol. 2015, pp. 1-12, 2015.

[33] S. Yu, S. Jia, and C. J. N. Xu, "Convolutional neural networks for hyperspectral image classification," vol. 219, pp. 88-98, 2017.

[34] J. Yang, Y.-Q. Zhao, J. C.-W. J. I. T. o. G. Chan, and R. Sensing, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," vol. 55, no. 8, pp. 4729-4742, 2017.

[35] Y. Li, H. Zhang, and Q. J. R. S. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," vol. 9, no. 1, pp. 67, 2017.

[36] A. B. Hamida, A. Benoit, P. Lambert, C. B. J. I. T. o. g. Amar, and r. sensing, "3-D deep learning approach for remote sensing image classification," vol. 56, no. 8, pp. 4420-4434, 2018.

[37] C. Chatfield, A. J. Collins, C. Chatfield, and A. J. J. I. t. m. a. Collins, "Principal component analysis," pp. 57-81, 1980.

[38] H. Abdi, and L. J. J. W. i. r. c. s. Williams, "Principal component analysis," vol. 2, no. 4, pp. 433-459, 2010.

[39] I. T. Jolliffe, Principal component analysis for special types of data: Springer, 2002.

[40] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. J. N. c. Jackel, "Backpropagation applied to handwritten zip code recognition," vol. 1, no. 4, pp. 541-551, 1989.

[41] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, X. J. I. T. o. G. Huang, and R. Sensing, "Hyperspectral image classification with deep learning models," vol. 56, no. 9, pp. 5408-5423, 2018.

[42] Y. S. Chen, H. L. Jiang, C. Y. Li, X. P. Jia, and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," Ieee Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, Oct, 2016.

[43] F. Melgani, and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 8, pp. 1778-1790, 2004.

[44] H. Lee, and H. Kwon, "Going Deeper With Contextual CNN for Hyperspectral Image Classification," IEEE Trans Image Process, vol. 26, no. 10, pp. 4843-4855, Oct, 2017.

[45] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification," Ieee Geoscience and Remote Sensing Letters, vol. 17, no. 2, pp. 277-281, Feb, 2020.

[46] M. Y. He, B. Li, H. H. Chen, and Ieee, "Multi-Scale 3D Deep Convolutional Neural Network for Hyperspectral Image Classification," IEEE International Conference on Image Processing ICIP. pp. 3904-3908, 2017.

[47] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. J. R. S. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," vol. 11, no. 11, pp. 1307, 2019.