HYPERSPECTRAL IMAGES CLASSIFICATION BASED ON FUSION FEATURES DERIVED FROM 1D AND 2D CONVOLUTIONAL NEURAL NETWORK

Jiang Tao¹, Wang Xinjie^{1,*}

¹ Dept. of Forestry, Beijing Forestry University, Xueyuan Street, Beijing, China - (taoorwell, xinjieW)@bjfu.edu.cn

KEY WORDS: Hyperspectral images, 1D-CNN, 2D-CNN, features fusion

ABSTRACT:

In recent years, deep learning technology has been continuously developed and gradually transferred to various fields. Among them, Convolutional Neural Network (CNN), which has the ability to extract deep features of images due to its unique network structure, plays an increasingly important role in the realm of Hyperspectral images classification. This paper attempts to construct a features fusion model that combines the deep features derived from 1D-CNN and 2D-CNN, and explores the potential of features fusion model in the field of hyperspectral image classification. The experiment is based on the deep learning open source framework TensorFlow with Python3 as programming environment. Firstly, constructing multi-layer perceptron (MLP), 1D-CNN and 2D-CNN models respectively, and then, using the pre-trained 1D-CNN and 2D-CNN models as feature extractors, finally, extracting features via constructing the features fusion model. The general open hyperspectral datasets (Pavia University) were selected as a test to compare classification accuracy and classification confidence among different models. The experimental results show that the features fusion model obtains higher overall accuracy (99.65%), Kappa coefficient (0.9953) and lower uncertainty for the boundary and unknown regions (3.43%) in the data set. Since features fusion model inherits the structural characteristics of 1D-CNN and 2D-CNN, the complementary advantages between the models are achieved. The spectral and spatial features of hyperspectral images are fully exploited, thus getting state-of-the-art classification accuracy and generalization performance.

1. INTRODUCTION

With the development of sensor technology, remote sensing data sources are gradually moving towards high spatial resolution and spectral resolution. The improvement of data source resolution provides a basis for remote sensing interpretation. Compared with multispectral images, the hyperspectral images has a narrower band width, so it has more continuous spectral bands within the same spectral range. Because of the spectral characteristics of ground objects, hyperspectral images data can accurately distinguish ground objects, especially for ground objects with subtle spectral differences. Therefore, hyperspectral images data are widely used in some fields such as geology(Schneider et al., 2014), agriculture and forestry. Classification of hyperspectral images has always been the focus of remote sensing community. However, duo to the "Hughes" phenomenon caused by the high-dimensional characteristics of hyperspectral data and the non-linearity data caused by signal instability, this research has become challenging.

Among the traditional research methods, Principal Component Analysis (PCA) and other technologies are used to reduce the dimension of the data. Although the classification efficiency is improved, to a certain extent, rich feature information lost, resulting in low classification accuracy. On the other hand, compared with the traditional Maximum Likelihood Classification (MLC), Spectral Angle Mapper (SAM) and other algorithms (Kumar et al., 2010), although the performance of some new algorithms such as Support Vector Machine (SVM)(Mountrakis et al., 2011) and Random Forest (RF) is continuously improving, extracting limited spectral information with single classifier can not meet the needs of high precision classification. How to make full use of spectral information and spatial information of hyperspectral images data and combine with multi-model classification is the

research trend.

Neural network, a programming paradigm inspired by biology, enables computers to learn from observed data, and deep learning is a powerful set of technologies to promote efficient learning of neural network(Nielsen, 2015). The proposed neural network algorithm can be traced back to 1940s and 1950s. With times, the structure of the network and the algorithm mechanism of propagation have been evolving. Since AlexNet(Krizhevsky et al., 2012) achieved unprecedented success in ILSVRC competition in 2012, this kind of Convolutional Neural Network (CNN) and learning mechanism has received extensive attention and has undergone continuous development and evolution. Until now, CNN has been widely used in computer vision, pattern recognition and other fields. As deep learning technology is migrated to remote sensing community, neural networks are also gradually applied to hyperspectral images data classification. In the early stage, some simple neural networks such as Multilayer Perceptron (MLP)(Kumar et al., 2010) and Deep Auto Encoder (DAE)(Ma et al., 2016) are used to extract spectral features of images. On the other hand, CNN is generally applied to RGB images, which have only three channels and have huge difference with remote sensing data. Remote sensing data is characterized by its generation mode, acquisition method and application(Zhu et al., 2017), which bring some new challenges for deep learning and CNN.

In the application of CNN on hyperspectal images classification, the research on self-extraction of spatial and spectral information by using CNN is continuously developing. (Paoletti et al., 2018) proposed a three-dimensional CNN to extract and classify spectral and spatial features of hyperspectral images and got the higher accuracy than traditional neural network, but the essence of that network is to use two dimensional convolution operation, which belongs to 2D-CNN. (Makantasis et al., 2015) used 2D-CNN to extract spatial and spectral informa-

^{*}Corresponding author

tion of hyperspectral images data for classification after PCA dimension reduction, and achieved good classification effect. On the other hand, in bid to achieve more efficient extraction of hyperspectral images features, the method, which focus on that spectral and spatial features are extracted step by step and then features fusion, is also being studied. Features fusion is an effective measure to improve classification progress in multimodel systems(Mangai et al., 2010). (Zhao, Du, 2016) used Balanced Local Discriminant Embedding (BLDE) algorithm to extract spectral features, 2D-CNN for mining spatial feature, and classified the fused features using logical regression algorithm. (Yue et al., 2016) used spectral features obtained by Auto Stack Encoder (ASE) and spatial features obtained by 2D-CNN for classification after fusion operation. both of which have achieve greater accuracy improvement. 3D-CNN, which is originally used in video processing and action capture, is also gradually applied to hyperspectal images classification(Chen et al., 2016), and achieved better classification results, however, the problem of increasing the difficulty of training and prediction due to the large number of parameters arises obviously. Migrating pre-trained CNN is an effective method for extracting hypersepctral images information. (Yang et al., 2017) used the concept of migration learning to use pre-trained CNN to improve the accuracy of hyperspectral images classification with fewer samples.

In this paper, 1D-CNN and 2D-CNN are used as feature extractors to extract spectral and spatial features respectively, and then, spectral and spatial features were fused for classification. In addition, the input form and basic structural characteristics of the features fusion model of MLP, 1D-CNN and 2D-CNN are described. At the same time, in order to compare the classification effects of MLP, 1D-CNN, 2D-CNN and features fusion model, on the basis of comparing the overall classification accuracy and Kappa coefficient, the evaluation and comparison of classification confidence is proposed. This is also a problem that can be easily ignored in neural network classification task.

2. THE PROPOSED ALGORITHM

2.1 MLP

Multilayer perceptron (MLP) is a fully connected neural network in which the input and output layer are transmitted in a direct direction. The network is mainly divided into an input layer, a hidden layer and an output layer. Each layer consists of a plurality of neurons which are connected with the neurons in the previous layer and the neurons in the next layer. After the input value of the previous layer is calculated and output through an activation function, the input value of the previous layer is used as the input value of the next layer and continues to be transmitted until output.

$$\alpha^{(l+1)} = \sigma(w^l * \alpha^l + b^l) \tag{1}$$

In the equation 1, l it refers to the current layer, w and b is the weight value and bias value of the current layer. For the activation function, there are many choices of activation functions, such as sigmoid function and rectified linear units. It is worth noting that the current MLP neurons, i.e. activation functions, select artificial neurons instead of perceptrons, but we still call this network as MLP. $\alpha^{(l+1)}$ is the output value of the current layer.

2.2 1D CNN

In image classification task, since MLP is fully connected, specifically for a certain pixel, the pixel close to it and the pixel far away from it have no different influence on it. This type of network structure does not consider the spatial characteristics of the image. CNN is a neural network different from MLP. Its unique convolution and pooling operations can fully extract the spatial structure features of images. Unlike MLP, which spreads out all pixel values of an image, CNN uses normal planar images as input to determine local receptive fields, and uses convolution kernel to perform convolution operation on the image to obtain feature layers, which is more in line with the characteristics of human visual observation, and the parameters of convolution kernel are shared in this process, which also greatly reduces the amount of network parameters.

$$y_{i,j}^{l} = \sigma^{l} \left(\sum_{n=1}^{k} \sum_{m=1}^{k} W_{n,m}^{l} * x_{i+n,j+m}^{l-1} + b^{l}\right)$$
(2)

In equation 2, $y_{i,j}^l$ is the output value of the previous layer after convolution operation, σ^l is the activation function, $W_{n,m}^l$ is the convolution kernel of k * ksize, b^l and is the bias value of the feature layer. Generally, convolution operations are followed by a layer of pooling operations to refine the abstract features. Maximizing pooling operation is the process of finding the maximum value of the feature layer after convolution operation.

$$I_{1dcnn} = [[B_1], [B_2], [B_3], ..., [B_N]]$$
(3)

Figure 1 shows the structural framework of 1D-CNN. Reflectance values of all bands of a single pixel are extracted from the original hyperspectral images data to form a two-dimensional array as the input layer of the model (see equation 3).

In Figure 1, there are two 1D convolution operations and two 1D pooling operations in each 1D convolution layer. after the first 1D convolution operation, the number of feature layers in the input layer changes to 3. after the 1D pooling operation, the number of feature layers remains unchanged, but the dimension of feature layers is halved. Similarly, after the second 1D convolution, the number of feature layers becomes 6, and after the second 1D pooling, the feature dimension is halved. In the 1D convolution layer, 1D convolution and 1D pooling operations are used to extract the band feature information of the input layer, the outputs of 1D convolution layer are stacked and input to the full connection layer, and the neurons in the full connection layer are connected to propagate the feature information correspondingly. Finally, Softmax function is used to classify the labels of the output layer.



Figure 1. The architecture of 1D-CNN

2.3 2D CNN

The structure of 2D-CNN shows as shown in Figure 2. A patch with a radius of n around a certain pixel as the center is extracted from the original hyperspectral images data in the form of a three-dimensional array with a shape as the input layer (see equation 4).

In the 2D convolution layer, there are two 2D convolution and 2D pooling operations. As shown in Figure 2, after the first 2D convolution operation, the number of feature layers in the input layer is 5. After the first 2D pooling operation, the number of feature layers is unchanged, and the feature layers are halved in both dimensions. Similarly, after the second 2D convolution and 2D pooling operation, the number of feature layers is 10, and the dimension of feature layers is halved again. Carry out 2D convolution and 2D pooling operations on the input image patches, extract the spatial features of the patches, expand the output of the 2D convolution layer horizontally, correspondingly connect neurons in the full connection layer, further spread the feature information, and finally classify the output layer category labels by using Softmax function.



Figure 2. The architecture of 2D-CNN

2.4 Features Fusion Model

Different models have different data feature mining due to their structural characteristics. In this paper, in order to fully mine the spectral and spatial features of hyperspectral images for classification, a features fusion model is designed that combines the features from 1D-CNN and 2D-CNN. The specific model diagram is shown in Figure 3. The training and prediction steps of the feature fusion model are as follows:

(1) Training 1D-CNN and 2D-CNN respectively by using different data input forms, and saving model parameters.

(2) Extracting the network structures of 1D-CNN and 2D-CNN from the input layer to the convolution layer as 1D-CNN feature extractors and 2D-CNN feature extractors (and in Figure 3).

(3) The 1D-CNN features extractor and 2D-CNN features extractor are used to extract sample features corresponding to input forms, and the extracted features are stacked and fused as input layers of the new model.

(4) Building a full connection layer and an output layer for the new features fusion model, interfacing with the input layer, retraining the model, and fixing parameters for prediction and classification.



Figure 3. The architecture of Features Fusion Model

2.5 Uncertainty Evaluation

Apart from the indexes of classification accuracy (OA, Kappa and F1-socre), in order to measure the Uncertain of classification, the uncertainty index and classification confidence graph are proposed as the basis for generalization of the comprehensive evaluation model. Before introducing uncertainty, it should be mentioned that MLP, 1D-CNN, 2D-CNN and feature fusion model all have one thing in common, that is, in the final classification, Softmax function is adopted for decision classification. After Softmax function, the output layer takes the form of:

$$O = [P_{C_1}, P_{C_2}, P_{C_3} \dots P_{C_n}]$$
(5)

$$U = \frac{Count(n|max((O) < conf)}{Len(n)}$$
(6)

In equation 5, the output probability array representing the output layer represents the probability that the output result is a category, and the sum of all category probabilities is 1. In almost all neural network models, Softmax function is used for prediction classification, because it can output probability range, which is conducive to learning and decision analysis. In the actual prediction of the model, the work is only to select the one with the highest probability as the final prediction value, even if the probabilities between the categories are not different. The confidence level of model classification is difficult to reflect.

In this paper, firstly, the output probability array of the model is used to obtain the maximum probability value in each sample classification probability array, and visual analysis is carried out to obtain the classification confidence level map. Then, the classification confidence level map is used to count the Uncertain value to measure the generalization confidence level accuracy of the model. In Equation 6, U represents Uncertainty, Len(n)is the total number of samples, and the Count() in the numerator is to count the number of samples that satisfy the maximum probability value in the output probability array to be less than conf. In this paper, conf is set to 0.5, which means that the maximum probability value in the output probability array is less than 0.5, and the classification is low in confidence or uncertain. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-3/W10, 2020 International Conference on Geomatics in the Big Data Era (ICGBD), 15–17 November 2019, Guilin, Guangxi, China

Number	Color	Class	Train	Test
1		Asphalt	200	6431
2		Meadows	200	18449
3		Gravel	200	1899
4		Trees	200	2864
5		Painted metal sheets	200	1145
6		Bare Soil	200	4829
7		Bitumen	200	1130
8		Self-Blocking Bricks	200	3482
9		Shadows	200	747

Table 1. Pavia University dataset with train and test sample size per class

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

3.1 Hyperspectral Images Dataset

The Pavia University image was acquried by ROSIS sensor during a flight campaign over Pavia, northern Italy. The number of spectral bands is 103 and spatial size is 610 * 340 pixels, and the spatial resolution is 1.3 meters. There are 9 different ground objects, in experiments, 200 pixels per class for training and the remaining for test (see Table 1) In order to speed up the data training and model fitting, band values of the data sets are normalized. Since the range of pixel values of each band is very different, according to the maximum and minimum values of each band and scale the data into (-1,1).

3.2 Model Architecture and Configuration

In this experiment, all models were built, trained and predicted using the TensorFlow, compiled using Python3. Since the neural network model has a large amount of parameters and involves matrix operations, it is very time-consuming to use CPU to train and adjust the appropriate super parameters. Therefore, GPU acceleration is essential and Nvidia Tesla T4 GPU was used in this experiment.

In the MLP, the input layer is a pixel value of n bands in the form of a one-dimensional array (also called tensor in Tensor-Flow) with a length of n, i.e. there are n neurons in the input layer, and two hidden layers are set, each layer is provided with 256 neurons, the neurons between layers are connected with each other, the neurons in the input layer are multiplied by the initialized weight and transferred to the hidden layer through the RELU activation function, the neurons in the hidden layer are transferred to the output layer, and the number of neurons set in the output layer is 9, corresponding to the ground object category. In order to prevent the overfitting, Dropout layer (Srivastava et al., 2014) is added after each hidden layer. Cross entropy function is selected as the loss function, Adam is selected as the optimizer, and the learning rate is 0.001.

In the 1D-CNN, although the input layer is also pixel values of n bands, due to the unique 1D convolution operation, it takes the form of a two-dimensional array with the shape of (N, 1). In the Pavia University data set, the input layer is an array with the shape of (103, 1), the 1D convolution layer includes 1D convolution and 1D pooling operations (see Figure 1). In the first 1D convolution layer, The convolution kernel size is 5, the number of convolution kernels is 12. After 1D convolution operation, the array shape changes to (103,12). The result is subjected to 1D pooling operation. The pooled windows size is 2, the step size is 2, and the output result is (52,12). In the second and third 1D convolution layers, the above operation is repeated, and the output result form of the third 1D convolution layer is (13,48). The result is expanded to 13 * 48 neurons connected to the full connection layer (including 512 neurons), which has same operation mechanism as MLP's layer. In the 1D-CNN model, a batch normalization layer (Ioffe, Szegedy, 2015) (BN) is added after the 1D convolution operation of each 1D convolution layer in order to speed up the function convergence. At the same time, a Dropout layer is provided behind both 1D convolution layer and full connection layer to prevent overfitting. The selection of loss function, activation function and optimizer is consistent with MLP, and the learning rate is 0.0001.

In the 2D-CNN, different from the input layers of MLP and 1D-CNN, the 2D-CNN model takes the three-dimensional array of as input. Taking Pavia Center data set, M=21 as an example, in the first layer of 2D convolution layer (see Figure 2), the 2D convolution kernel size is (3,3), the number of convolution kernels is 12, and the convolution step size is 1. After the data passes through 2D convolution operation, its output form is (21,21, 12). In 2D pooling operation, the size of the pooling windows is (2,2) and the step size is 2. After 2D pooling operation, the output form is (11,11,12). Similarity, after the 2D convolution operation of the third layer is completed, the output result is in the form of (3,3,48), and the result is expanded horizontally to 3 * 3 * 48 neurons connected with the full connection layer. BN and Dropout layers are added to accelerate convergence and prevent over-fitting. The selection of loss function, activation function and optimizer are consistent with MLP and 1D-CNN, and the learning rate is 0.0001.

3.3 Classification Results and Comparison

3.3.1 MLP and 1D CNN It can be seen from Table 2 that the overall classification accuracy (OA) and Kappa of 1D-CNN reach 88.61% and 0.8497, which are 5.75% and 0.734 higher than MLP(82.86% and 0.7763), respectively. In terms of F1score of each category, 1D-CNN is higher than MLP in all categories of F1-score, especially in categories 6, 4, 3 and 2. 1D-CNN's F1-score is 10.78%, 10.70%, 6.96% and 5.73% higher than MLP respectively. Figure 4 clearly shows the influence of classification between categories, as can be seen from the left of Figure 4, based on MLP algorithm, a large number of categories in category 1 are wrongly classified into categories 7(0.14), and categories 2 and 6, 3 and 8 all have serious misclassification (0.11 and 0.15, 0.15 and 0.19), while compared with the right of Figure 4, based on 1D-CNN algorithm, Although category 1 is similar to MLP algorithm in that a large number of samples are misclassified into category 7(0.15), category 2 and category 6, category 3 and category 8, the misclassification of the two groups (0.07 and 0.10, 0.17 and 0.10) is significantly lower than MLP algorithm. In combination with Figure 5, we can see the specific locations of such misclassification, and by comparing MLP and 1D CNN, we can clearly see that the misclassification of categories 2 and 6, 3 and 8 has been weakened to different degrees. When comparing the full classification result, it can also be seen that the internal misclassification of various types of objects is weakened as a whole.

3.3.2 2D CNN with Different M When comparing the classification accuracy and classification visualization effect of 2D-CNN with different M, this paper introduces a new indicator of generalization ability of the judgment algorithm in unknown regions (Uncertainty) (see equation 6). In Figure 6, it can be seen that with the increase of m (5 to 41), the OA of 2D-CNN models based on different M is increasing, and after M is greater than or equal to 21, its OA is not obviously increasing and tends to be flat. corresponding to Figure 7(A), it can be seen that

The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-3/W10, 2020 International Conference on Geomatics in the Big Data Era (ICGBD), 15–17 November 2019, Guilin, Guangxi, China



Figure 4. The Classification results Heat Map of MLP and $1D\mathchar`-CNN$



Figure 5. 1D-CNN and MLP classification results



Figure 6. Overall accuracy and Uncertainty as a function of M based on 2D-CNN



Figure 7. Classification results in different M based on 2D-CNN

with the increase of m, the classification effect of image labeled areas is getting better, specifically, the phenomenon of misclassification within categories is gradually decreasing and finally tends to be flat. However, the uncertainty of classification can be seen from Figure 6, as M increases, the value of uncertainly also generally increases, which indicates that the uncertainty of classification is increasing. Corresponding to the full classification confidence level map of Figure 7(C), it can be seen that with the increase of M, the gray areas in the map are deepening (the darker the color is, the lower the confidence level), and are mainly distributed on the boundaries between categories and unlabeled areas, indicating that the classification confidence level of the models in these areas is extremely low and the classification uncertainty is very high. As can be seen from Figure 6, when M is 21, it keeps high classification accuracy while the value of Uncertain is lower than that of M after it, so in the feature fusion stage, M selected by 2D-CNN model in Pavia University dataset is 21.

3.3.3 Feature Fusion Model From Table 2, we can see that on Pavia University data set, the overall classification accuracy and Kappa of 2D-CNN are 99.52% and 0.9935, which are 10.91% and 0.1438 higher than 1D-CNN(88.61% and 0.8497), respectively. When comparing the classification accuracy of various categories, it is found that the classification accuracy of 2D-CNN is higher than that of 1D-CNN, especially in categories 7, 6 and 3. F1-score of 2D-CNN is 30.23%, 24.29% and 15.45% higher than that of 1D-CNN respectively. However, in the uncertain comparison of classification, it is found that 2D-CNN(11.96%) is much higher than 1D-CNN(3.94%).

In combination with the comparison of Figure 8, salt and pepper phenomenon is more obvious on the full classification map on 1D-CNN than that of 2D-CNN, and misclassification phenomenon occurs in some categories of objects more than that of 2D-CNN, which also corresponds to the overall accuracy of 1D-CNN in Table 2 and F1-score is lower than that of 2D-CNN in some categories. However, when comparing the classification confidence maps of 1D-CNN and 2D-CNN, it can be found that the classification confidence distributions of 1D-CNN classification confidence maps are relatively average. From the color difference, it is known that the classification confidence differences are not large. However, it can be seen from the classification confidence map of 2D-CNN that the confidence regionalization is serious, the confidence is extremely high in the labeled sample area (pure white), but in the boundary of the category and the unknown area with few labeled samples, Then a large area with low confidence appears. Compared with gray scale, the confidence is much lower than that in 1D-CNN, which also corresponds to the Uncertain value counted in Table 2. It shows



Figure 8. Classification result and classification confidence level map based on different models

that 2D-CNN has worse generalization ability in the boundary and unknown areas than 1D-CNN and is more prone to misclassification.

As can be seen from Table 2, the overall accuracy and Kappa of the feature fusion model at Pavia University are 99.65% and 0.9953, which are slightly higher than those of 2D-CNN(99.52% and 0.9935) by 0.13% and 0.0018. Also on the classification accuracy comparison of various categories, the feature fusion model has slightly higher accuracy than 2D-CNN in most categories but has little difference. However, it is worth noting that on the comparison of classification uncertainty, the feature fusion models (3.43%) are much lower than 2D-CNN(11.96% and 12.31%). Referring to Figure 8, Comparing the full classification map and classification confidence map of 1D-CNN, 2D-CNN and feature fusion model, we can find that the feature fusion model performs better in full classification, without salt and pepper phenomenon and excessive smoothing problems. On classification confidence map, the feature fusion model has better processing ability in boundary and unknown areas than 2D-CNN model, and at the same time it also solves the problem that 1D-CNN model does not have high classification confidence in labeled areas, i.e. within category objects.

Class	MLP	1D-CNN	2D-CNN	FFM
Asphalt	84.38%	85.41%	99.48%	99.81%
Meadows	88.02%	93.75%	99.75%	99.70 %
Gravel	68.17%	75.13%	99.42%	99.79%
Trees	79.48%	90.18%	98.61%	98.92%
Painted metal sheets	98.28%	99.30%	100.00%	100.00%
Bare Soil	73.38%	84.16%	99.61%	99.51%
Bitumen	67.53%	68.59%	98.82%	99.91%
Self-Blocking Bricks	80.36%	84.44%	99.08%	99.55%
Shadows	99.20%	99.73%	99.93%	100.00%
Overall Accuracy(OA)	82.86%	88.61%	99.52%	99.65%
Kappa	0.7763	0.8497	0.9935	0.9953
Uncertain(U)	6.92%	3.94%	11.96%	3.43%

Table 2. Classification accuracy and uncertainty comparison among different models

4. CONCLUSION

Compared the experimental results, although MLP has same input as 1D-CNN, the classification accuracy and classification visualization results of 1D-CNN are better than those of MLP. The main reason is the different structure of two types of neural networks. MLP is a simple neural network in which neurons are directly connected with each other, while 1D-CNN has unique one-dimensional convolution and one-dimensional pooling operations, the size of convolution kernels, the number of convolution kernels and the adjustment mechanism of other super parameters can enable 1D-CNN to mine spectral characteristics of images more fully than MLP, thus achieving better classification effect.

In comparison with the classification results of 1D-CNN and 2D-CNN, it is found that the classification effect of 2D-CNN is much higher than that of 1D-CNN. What can be found is that unlike 1D-CNN, which takes the band value of a single pixel as input, in 2D-CNN, a pixel and its adjacent M*M pixels are taken as input, which provides favorable guarantee for 2D-CNN to extract the spatial features of images. Furthermore, 2D-CNN model has unique two-dimensional convolution and twodimensional pooling operations, which can fully excavate the spatial features of image patches, thus playing a more efficient classification performance. On the other hand, when comparing the classification results of 2D-CNN with different M, with the increase of M, the overall accuracy can only be improved to a limited extent, but at the same time, it faces a more serious problem, the classification uncertainty is gradually increasing, which is undoubtedly fatal to the generalization of unknown regions of the model. The reason for this kind of problem should be that with the increase of M, the reuse rate of samples becomes higher and higher, and the model becomes over-fitted, resulting in weak generalization ability.

The features fusion model proposed in this paper can solve the above problems well. Since the feature fusion model combines the structural features of 1D-CNN and 2D-CNN, it obtains the ability of 2D-CNN to extract spatial features, thus avoiding the occurrence of salt and pepper phenomenon and making the classification smoother. On the other hand, it inherits 1D-CNN's ability to mine spectral information, which enhances the generalization in category boundaries and unknown regions. Therefore, the accuracy and confidence of classification have been improved unprecedentedly. At present, with the technology of deep learning technology gradually migrating to the field of remote sensing community, hyperspectral image classifica-- tion based on deep learning technology is developing at a high = speed. Fully mining spectral and spatial characteristics of hyperspectral images is the key to classification. In future research work, it is very critical to reduce the cost of designing a network structure that conforms to hyperspectral data and shorten the time to adjust appropriate super parameters in combination with cloud computing technology.

REFERENCES

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10), 6232–6251.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Kumar, U., Kerle, N., Punia, M., Ramachandra, T., 2010. Mining land cover information using multilayer perceptron and decision tree from MODIS data. *Journal of the Indian Society of Remote Sensing*, 38(4), 592–603.

Ma, X., Wang, H., Geng, J., 2016. Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9), 4073–4085.

Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks. 2015 IEEE International Geoscience and Remote Sensing Symposium (IG-ARSS), IEEE, 4959–4962.

Mangai, U. G., Samanta, S., Das, S., Chowdhury, P. R., 2010. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical review*, 27(4), 293–307.

Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), 247–259.

Nielsen, M. A., 2015. *Neural networks and deep learning*. 25, Determination press San Francisco, CA, USA:.

Paoletti, M., Haut, J., Plaza, J., Plaza, A., 2018. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS journal of photogrammetry and remote sensing*, 145, 120–147.

Schneider, S., Murphy, R. J., Melkumyan, A., 2014. Evaluating the performance of a new classifier-the GP-OAD: A comparison with existing methods for classifying rock type and mineralogy from hyperspectral imagery. *ISPRS journal of photogrammetry and remote sensing*, 98, 145–156.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.

Yang, J., Zhao, Y.-Q., Chan, J. C.-W., 2017. Learning and transferring deep joint spectral-spatial features for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8), 4729–4742.

Yue, J., Mao, S., Li, M., 2016. A deep learning framework for hyperspectral image classification using spatial pyramid pooling. *Remote Sensing Letters*, 7(9), 875–884.

Zhao, W., Du, S., 2016. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8), 4544–4554.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8–36.