Spatial-spectral feature classification of hyperspectral image using a pretrained deep convolutional neural network

Bing Liu, Anzhu Yu, Xibing Zuo, Zhixiang Xue, Kuiliang Gao and Wenyue Guo
PLA Strategic Support Force Information Engineering University, Zhengzhou, China

ABSTRACT
Deep learning based methods have recently been successfully explored in hyperspectral image classification field. However, training a deep learning model still requires a large number of labeled samples, which is usually impractical in hyperspectral images. In this paper, a simple but effective feature extraction method is proposed for hyperspectral image classification. Specifically, a pretrained deep convolutional neural network based on the ImageNet dataset is used to extract spatial features of a hyperspectral image. Recently, it is easy to obtain a pretrained convolutional neural network on the Internet. Note that the pretrained models are trained by using the ImageNet dataset. This means that the proposed method does not need labeled hyperspectral samples to train the deep model. Therefore, the proposed method alleviates the problem of lacking labeled samples and avoids the artificial design of feature extraction rules. Finally, the extracted features are stacked with spectral features as the input of a support vector machine classifier. The proposed method is conducted on three widely used hyperspectral image datasets. The experimental results demonstrate that the proposed method could outperform the conventional feature extraction methods and deep learning based methods.

Introduction
Hyperspectral image (HSI) classification has become a hot topic in the field of remote sensing and has also been widely used in many applications. (Bing Liu, Guo et al., 2020) In general, the complex characteristics of hyperspectral data make the accurate classification of such data challenging for traditional machine learning methods. (S. Li et al., 2019) Recently, deep learning based methods have been successfully explored for HSI classification and demonstrate good performance. (Bing Liu, Yu, Zhang, Yu et al., 2018) (B. Liu et al., 2019) (Bing Liu, Gao et al., 2020)

Deep learning based pixelwise classifiers including one-dimensional convolution neural network (1D-CNN), (Hu et al., 2015) deep belief network (DBN) (Chen et al., 2015) and recurrent neural network (RNN) (Mou et al., 2017) are first used for supervised classification of HSI. A disadvantage of the pixelwise classifiers is that they do not consider spatial information in the classification procedure. In this context, 2D-CNN (Yue et al., 2015) (Bing Liu et al., 2017) (Y. Chen et al., 2017), 3D-CNN (Y. Chen et al., 2016) (Bing Liu, Yu, Zhang, Tan et al., 2018) (H. Zhang et al., 2019) and 2D-RNN (Bing Liu, Yu, Yu et al., 2018) (Hang et al., 2019) models are used to mine the spatial information of HSIs, which greatly improves the classification performance of HSI. Meanwhile, to further improve the classification accuracy, deep residual network (Zilong et al., 2017) (Haut et al., 2019) and deep dense network (C. Zhang et al., 2019) are also used for HSI classification.

The supervised deep learning classifiers could improve the classification performance of HSI. However, the aforementioned methods require a certain number of labeled samples to ensure ideal classification results. To deal with the lack of labeled samples, researchers have explored unsupervised deep learning of HSIs and obtained many meaningful research results. For example, an autoencoder network (Koda et al., 2019) is designed to extract spectral-spatial features from HSI. Furthermore, an unsupervised spatial-spectral feature learning strategy (Mei et al., 2019) is proposed for HSIs using 3-Dimensional (3D) convolutional autoencoder (3D-CAE). A wasserstein generative adversarial network (WGAN) (M. Zhang et al., 2018) is designed to train a deep learning based feature extractor without supervision.

The above unsupervised deep learning methods could improve the classification performance. However, they usually require complex training strategies and long training time. With the rapid development and application of deep learning, it is easy to obtain a pretrained deep convolution neural network on the Internet. More importantly, a lot of research and practice show that the pretrained network on large datasets such as ImageNet can effectively improve the performance of the model in other tasks. In
particular, some studies (Zeiler & Fergus, 2014) (Yu et al., 2014) have also shown that the underlying layers of a deep convolution neural network can learn to extract general features for different tasks. Motivated by this, a simple but effective feature extraction method is proposed to improve the classification accuracy of HSIs. Specifically, a pretrained VGG19 model (Simonyan & Zisserman, 2014) trained by the ImageNet dataset is used as a spatial feature extractor of a HSI. This means that it does not need the process of training deep learning models and the manual design of feature extraction rules. The proposed method can not only alleviate the lack of labeled training samples, but also effectively improve the classification accuracy of HSIs. The main contribution of this paper is that a simple but effective feature extraction method is proposed for hyperspectral image classification.

**Proposed method**

In this section, we explain the architecture of VGG19 and the proposed feature extraction method in detail.

**Architecture of VGG19**

VGG19 (Simonyan & Zisserman, 2014) is first designed for large-scale image recognition task on the ImageNet dataset. As shown in (Figure 1), VGG19 contains 16 convolutional layers, 3 fully connected layers with learning parameters and 5 pooling layers. In (Figure 1), Conv 3-n represents that this convolutional layer uses n convolutional kernels with a size of 3 x 3, Block n represents repeating this convolutional layer n times, Max-pool represents the max pooling layer, FC-n represents the fully connected layer with n neurons. The input of VGG19 is a fixed-size 224 x 224 RGB image. In the ith convolutional layer, the input size is w i x h i x c i, where w i and h i are the row and column in spatial dimensions, respectively, and c i is the feature dimension. Given inputs x ij of the lth convolutional layer, the output x ij+1 is defined as below:

\[ x_{ij}^{l+1} = f(\sum_{i \in M'} x_{ij} * K_{ij}^{l+1} + b_{ij}^{l+1}) \]  

where \( M' \) is the set of input maps, \( K_{ij}^{l+1} \) is the convolutional kernel of layer \( l + 1 \) that connects the ith map in layer \( l \) and the jth map in layer \( l + 1 \), \( b_i \) is the bias of the jth map of layer \( l + 1 \), \( f(\bullet) \) is a nonlinear activation function. Each output map of a convolutional layer is the combination of convolution of input maps.

Compared with conventional CNN, VGG19 has a deeper architecture, which makes the network hard to train. Rectified Linear Units function is selected as the nonlinear activation function in Eq (1) to avoid the vanishing gradient problem. To prevent the overfitting, dropout is also introduced. Note that dropout is a technique that some weights of hidden nodes are randomly discarded. This can be seen as deleting part of the network structure, but retaining their weights (only not update temporarily) for next training use.

**Spatial-spectral feature extraction**

As for HSI classification task, one of the greatest challenges is determining what types of features should be extracted from the pixels. In this paper, a pretrained VGG19 model is selected as a spatial feature extractor. The extracted spatial features are then concentrated with the original spectral features to generate the spatial-spectral features.

The proposed spatial-spectral feature extraction method is shown in (Figure 2). Note that the input size of the pretrained VGG19 is 224 x 224 x 3. The dimension of HSI is generally different from that of VGG19 input. First, we use PCA to reduce the HSI to 3 dimensions. Then we resample the reduced image to 224 x 224. By this way, the processed HSI can be input into VGG19 model to extract features. Each

![Figure 1. Illustration of the architecture of VGG19.](image-url)
convolutional layer of VGG19 can generate features. The feature dimension is related to the number of convolution kernels in convolution layer. VGG19 contains 16 convolutional layers. The number of convolution kernels of 16 convolutional layers is listed in (Table 1). Some studies have shown that the underlying features of deep convolution neural network are easier to be transferred to other tasks. Therefore, the outputs of the first 5 convolutional layers are stacked as the spatial features. Note that the output size of the convolutional layer is different from the HSI size. Therefore, we need to up sample the output of each convolution layer to ensure the same size as the HSI. We use the first 5 convolutional layers to extract features, thus the feature dimension is \(640(64 + 64 + 128 + 128 + 256)\). Finally, the extracted spatial features and the original spectral features are stacked to form the spatial-spectral features. After feature extraction, we can use traditional classifier (e.g., SVM) to complete HSI classification.

**Experimental results**

The experimental results are generated on a PC equipped with an Intel 2.59 GHz Core i7-9750 H and an Nvidia GeForce RTX 2070 M. The PC uses 16 GB of memory. The VGG19 model is implemented by Keras.

**Experimental data**

To evaluate the efficacy of the proposed feature extraction method, classification experiments were conducted on three well-known data sets. The first one is the University of Pavia data set acquired by the Reflective Optics System Imaging Spectrometer sensor over the campus at the University of Pavia, Northern Italy. This data set mainly contains an urban environment with multiple solid structures (asphalt, gravel, metal sheets, bitumen, and bricks), natural objects (trees, meadows, and soil), and shadows. After discarding the noisy bands, the considered scene contains 103 spectral bands, with a size of 610 × 340 pixels in the spectral range from 0.43 to 0.86 µm and with spatial resolution of 1.3 m.

The second dataset is the Salinas dataset collected by the 224-band AVIRIS sensor over the Salinas Valley, USA. This dataset is characterized by a spatial resolution of 3.7 m. The image size is 512 × 217 pixels. 20 water absorption bands are discarded. This dataset includes a total of 16 ground-truth classes, such as vegetables, bare soils, and vineyard fields.

The third data set is the Indian Pines data set. This data set is gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana and consists of 145 × 145 pixels and 224 spectral reflectance bands in the wavelength range 0.4–2.5 µm. 24 bands covering the region of water absorption are removed, resulting in 200 bands for classification.

Class name, number of labeled training samples, number of testing samples are listed in (Tables 2–4). For a fair comparison, 200 labeled samples per class are randomly selected as the supervised samples for all methods. We repeat the experiments of different methods 20 times. Note that the training samples for different methods are exactly the same in each

**Table 1. Number of convolution kernels in different layers.**

| Layer | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|       | 64 | 64 | 128| 128| 256| 256| 256| 256| 512| 512| 512| 512| 512| 512| 512| 512|
Table 2. Labels, number of labelled training samples, number of testing samples for the University of Pavia data set.

| No. | Class                     | Training | Testing   |
|-----|---------------------------|----------|-----------|
| 1   | Asphalt                   | 200      | 6631      |
| 2   | Meadows                   | 200      | 18,649    |
| 3   | Gravel                    | 200      | 2099      |
| 4   | Trees                     | 200      | 3064      |
| 5   | Sheets                    | 200      | 1345      |
| 6   | Bare Soil                 | 200      | 5029      |
| 7   | Bitumen                   | 200      | 1330      |
| 8   | Bricks                    | 200      | 3682      |
| 9   | Shadows                   | 200      | 947       |
| Total|                           | 1800     | 42,776    |

Table 3. Labels, number of labelled training samples, number of testing samples for the Salinas data set.

| No. | Class                      | Training | Testing |
|-----|----------------------------|----------|---------|
| 1   | Brocoli_green_weeds_1      | 200      | 2009    |
| 2   | Brocoli_green_weeds_2      | 200      | 3726    |
| 3   | Fallow                     | 200      | 1976    |
| 4   | Fallow rough_plow          | 200      | 1394    |
| 5   | Fallow smooth              | 200      | 2678    |
| 6   | Stubble                    | 200      | 3959    |
| 7   | Celery                     | 200      | 3579    |
| 8   | Grapes_untrained           | 200      | 11,271  |
| 9   | Soil_vinyard_develop       | 200      | 6203    |
| 10  | Corn_senesced_green_weeds  | 200      | 3278    |
| 11  | Lettuce_romaine_4_wk       | 200      | 1068    |
| 12  | Lettuce_romaine_5_wk       | 200      | 1927    |
| 13  | Lettuce_romaine_6_wk       | 200      | 916     |
| 14  | Lettuce_romaine_7_wk       | 200      | 1070    |
| 15  | Vinyard_untrained          | 200      | 7268    |
| 16  | Vinyard_vertical_trellis   | 200      | 1807    |
| Total|                            | 1800     | 5,129   |

Table 4. Labels, number of labelled training samples, number of testing samples for the Indian Pines data set.

| No. | Class                  | Training | Testing |
|-----|------------------------|----------|---------|
| 1   | Corn-notill            | 200      | 1428    |
| 2   | Corn-mintill           | 200      | 830     |
| 3   | Grass-pasture          | 200      | 483     |
| 4   | Grass-trees            | 200      | 730     |
| 5   | Hay-windrowed          | 200      | 478     |
| 6   | Soybean-notill         | 200      | 972     |
| 7   | Soybean-mintill        | 200      | 2455    |
| 8   | Soybean-clean          | 200      | 593     |
| 9   | Woods                  | 200      | 1265    |
| Total|                        | 1800     | 9243    |

eperiment. There are 16 different land-cover classes in the original ground truth of the Indian Pines data set. However, only 9 classes are used in this paper so as to avoid a few classes that have very few training samples. (B. Liu et al., 2019) (Bing Liu, Yu, Zhang, Tan et al., 2018) (W. Li et al., 2017)

Parameter analysis

Training VGG19 is carried out by optimising the multinomial logistic regression objective using mini-batch gradient descent with momentum. According to the original paper, the batch size is set to 256, the learning rate is initially set to 0.01, and then decreased by a factor of 10 when the validation set accuracy stopped improving. The training is regularised by weight decay (the L2 penalty multiplier set to 0.0005) and dropout regularisation for the first two fully-connected layers (dropout ratio set to 0.5).

At present, all deep learning open source frameworks provide the classical deep models (e.g., VGG19, ResNet50) trained by the ImageNet dataset. Therefore, the VGG19 model is easy to obtain. Once obtaining the pretrained model, VGG19 could be considered as a spatial feature extractor. The FLOPs of the spatial feature extractor (VGG19) is 11,276.35. The number of trainable parameters of VGG19 is 555,328.

The VGG19 model does not need training, thus the proposed feature extraction method is very simple. We only need to set the convolution layer number using
for feature extraction. To analyze the influence of convolution layer number on the classification accuracy, the convolution layer number is set to be 1, 2, 3, 4, 5, 6, 7, 8 respectively. Note that we realize spatial feature classification and spatial-spectral feature classification to demonstrate the effectiveness of spatial-spectral features. The classification results on three HSI data sets are shown in (Figure 3). First, it is found that the classification accuracy increases with the number of convolution layers. However, the classification accuracy tends to be stable when the number of layers is greater than 5. This means that using more convolutional layers to extract spatial features is not helpful to improve the classification accuracy. In addition, more convolutional layers would increase feature dimension greatly, which lead to a sharp increase in training time. Considering the above factors comprehensively, we finally use the first five convolutional layers of the VGG19 to extract spatial features. Second, it is found that the classification accuracy using spatial-spectral features is better than using only spectral feature. This demonstrates the necessity of using spatial-spectral features.

In this paper, support vector machine (SVM) and random forest (RF) are selected as the classifiers to classify HSIs. SVM uses RBF kernel function. The optimal hyperplane parameters C (parameter that controls the amount of penalty during the SVM optimization) and K (spread of the RBF kernel) have been traced in the range of $C = 2^{-2}, 2^{-1}, \ldots, 2^7$ and $K = 2^{-2}, 2^{-1}, \ldots, 2^5$ using cross validation. (Ghamisi et al., 2017) As for RF classifier, the number of decision trees is set to be 500.

The spatial feature extractor (VGG19) takes 3-band image as input. In this paper, PCA is used to reduce the dimension of HSI. We also test other band selection strategies, such as randomly selecting three bands, band selection via adaptive subspace partition strategy. (Wang et al., 2019) The mean classification accuracy of 20 runs are shown in (Figure 4). It could be found that “Band Selection” obtains a higher classification accuracy than that of “Random” and “PCA” obtains the best classification result. This is because band selection method can select more representative bands, so its classification accuracy is higher than randomly selecting three bands. Moreover, PCA can concentrate the information to the first three bands, thus retaining more information, so it obtains the best classification result.

In order to better observe the extracted spatial features, we take the University of Pavia data set as an example to visualize the features. The spatial feature visualization results are shown in (Figure 5). To help observe the features, we also show the HSI after dimensionality reduction. We can find that the first convolution layer could extract the edge information of the image. The higher the number of layers, the more abstract the features extracted. These extracted abundant features are helpful for further processing.

**Classification performance**

To demonstrate the effectiveness of the proposed method, VGG19+ SVM is compared with two traditional methods and several deep learning based methods. Two traditional methods for comparison are support vector machine (SVM) and spectral-spatial classification (EMPs). (Fauvel et al., 2008) In addition, we also apply PCA to SVM and EMPs. Deep learning based methods include 3D-CNN, (Mei et al., 2019) 3D-CNN, (Y. Chen et al., 2016) CNN-PPF. (W. Li et al., 2017) Note that 3D-CAE is an unsupervised feature learning based on deep learning. 3D-CNN and CNN-PPF are two supervised classifiers achieving the state-of-art results. To further demonstrate the effectiveness of the proposed method, we also test VGG19 with a random forest classifier (VGG19+ RF). 200 samples per class are randomly selected as the training data set. Note that the training data set for different methods are exactly the same.

In order to compare different classification methods more comprehensively, class-specific accuracy, overall Accuracy (OA), average accuracy (AA), $\kappa$ are used as evaluation criteria, in this paper. The compared results of class-specific accuracy, OA, AA and $\kappa$ are listed in (Tables 5–7). PCA+SVM and EMPs+PCA achieves the lowest overall classification accuracy. This shows that PCA can reduce the feature dimension and

![Figure 3](image-url). Classification accuracy with different convolution layer number. (a) the University of Pavia data set, (b) the Salinas data set, (c) the Indian Pines data set.
Figure 4. Classification accuracy with different band selection strategies. "PCA" denotes using PCA to reduce the dimension to 3, "Random" denotes randomly selecting three bands, "Band Selection" denotes selecting three bands via adaptive subspace partition strategy. (a) the University of Pavia data set, (b) the Salinas data set, (c) the Indian Pines data set.

Figure 5. Spatial feature visualization of the University of Pavia data set. (a) Image after dimension reduction using PCA (b) Randomly selecting a feature map of the first convolutional layer, (c) Randomly selecting a feature map of the second convolutional layer, (d) Randomly selecting a feature map of the third convolutional layer (e) Randomly selecting a feature map of the fourth convolutional layer (f) Randomly selecting a feature map of the fifth convolutional layer.

Table 5. Class-specific accuracy, OA(%), AA(%), and κ (%) of different techniques for the University of Pavia data set.

| Class no. | SVM | PCA +SVM | EMPS | EMPS +PCA | Method | 3D-CAE | 3D-CNN | CNN-PPF | VGG19 +SVM | VGG19 +RF |
|-----------|-----|----------|------|----------|--------|--------|--------|---------|------------|----------|
| 1         | 88.24 | 70.89 | 93.27 | 92.01 | 92.87 | 99.03 | 97.23 | 98.82 | 95.55 |
| 2         | 92.61 | 66.38 | 95.79 | 79.08 | 97.46 | 98.11 | 95.27 | 98.20 | 96.57 |
| 3         | 85.66 | 74.51 | 91.14 | 71.22 | 91.90 | 88.56 | 95.13 | 99.09 | 95.28 |
| 4         | 97.68 | 93.96 | 99.22 | 98.43 | 97.68 | 93.51 | 96.89 | 99.80 | 99.71 |
| 5         | 99.33 | 99.33 | 99.41 | 98.59 | 99.85 | 99.49 | 99.99 | 99.85 | 99.93 |
| 6         | 94.33 | 68.40 | 95.63 | 76.70 | 98.65 | 93.33 | 98.55 | 99.98 | 99.28 |
| 7         | 94.89 | 91.65 | 97.74 | 93.46 | 97.74 | 96.31 | 96.56 | 99.85 | 96.84 |
| 8         | 87.07 | 74.36 | 89.63 | 82.92 | 86.01 | 97.58 | 94.43 | 98.07 | 96.74 |
| 9         | 100.0 | 100.0 | 100.0 | 100.0 | 99.26 | 96.25 | 99.39 | 100.0 | 100.0 |
| OA        | 92.12 | 72.95 | 95.14 | 84.01 | 95.77 | 96.37 | 97.63 | 98.80 | 97.10 |
| AA        | 93.31 | 82.17 | 95.76 | 88.38 | 95.71 | 94.82 | 97.04 | 99.30 | 97.77 |
| κ         | 89.70 | 65.98 | 93.60 | 79.43 | 94.42 | 95.02 | 96.90 | 98.41 | 96.18 |
Table 6. Class-specific accuracy, OA(%), AA(%) and k (%) of different techniques for the Salinas data set.

| Class no. | SVM | PCA + SVM | EMPS | EMPS + PCA | 3D- CAE | 3D- CNN | CNN- PPF | VGG19 + SVM | VGG19 + RF |
|-----------|-----|-----------|------|------------|---------|---------|---------|-------------|------------|
| 1         | 99.05 | 97.41 | 99.40 | 98.61 | 99.85 | 99.94 | 99.84 | 100.0 | 99.95 |
| 2         | 99.81 | 97.70 | 99.60 | 100.0 | 85.45 | 99.77 | 98.11 | 100.0 | 99.75 |
| 3         | 99.60 | 97.64 | 99.89 | 99.93 | 99.96 | 99.87 | 99.75 | 99.95 | 99.92 |
| 4         | 99.71 | 99.57 | 99.64 | 99.93 | 99.96 | 99.87 | 99.75 | 99.95 | 99.92 |
| 5         | 97.76 | 93.35 | 98.25 | 91.78 | 99.59 | 98.54 | 99.70 | 99.75 | 99.77 |
| 6         | 99.82 | 99.75 | 99.90 | 99.87 | 100.0 | 99.92 | 99.60 | 99.64 | 99.25 |
| 7         | 99.22 | 98.94 | 99.39 | 96.17 | 99.92 | 99.60 | 99.96 | 99.96 | 99.25 |
| 8         | 81.33 | 72.79 | 85.31 | 69.54 | 86.52 | 99.31 | 89.11 | 100.0 | 89.28 |
| 9         | 99.69 | 96.60 | 99.61 | 96.49 | 99.76 | 99.97 | 99.69 | 99.48 | 99.50 |
| 10        | 95.91 | 89.17 | 97.35 | 89.26 | 99.76 | 99.41 | 97.78 | 100.0 | 96.43 |
| 11        | 99.53 | 98.13 | 99.63 | 95.04 | 99.81 | 100.0 | 99.33 | 100.0 | 99.63 |
| 12        | 99.90 | 99.69 | 100.0 | 99.90 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 13        | 99.13 | 97.27 | 99.67 | 98.80 | 100.0 | 100.0 | 99.67 | 99.78 | 99.89 |
| 14        | 98.32 | 95.89 | 98.13 | 97.57 | 100.0 | 98.75 | 98.99 | 98.73 | 98.73 |
| 15        | 78.87 | 67.46 | 88.50 | 78.73 | 94.48 | 90.36 | 89.99 | 97.12 | 94.94 |
| 16        | 99.06 | 96.40 | 99.28 | 88.71 | 99.94 | 85.93 | 99.07 | 99.94 | 98.73 |
| OA        | 92.65 | 88.05 | 94.92 | 88.26 | 96.34 | 97.28 | 94.87 | 98.43 | 96.59 |
| AA        | 96.67 | 93.83 | 97.72 | 93.42 | 98.67 | 97.48 | 98.07 | 99.39 | 98.39 |
| k         | 91.83 | 86.72 | 94.35 | 85.99 | 95.93 | 96.95 | 94.04 | 98.25 | 96.21 |
### Table 7. Class-specific accuracy, OA(%), AA(%), and $\kappa$ (%) of different techniques for the Indian Pines data set.

| Class no. | SVM | PCA + SVM | EMPS + PCA | 3D-CAE | 3D-CNN | CNN-PPF + SVM | VGG19 | VGG19 + RF |
|-----------|-----|-----------|------------|--------|--------|---------------|-------|------------|
| 1         | 77.24 | 62.71 | 88.94 | 68.38 | 88.31 | 92.99 | 95.31 | 89.29 |
| 2         | 84.10 | 46.11 | 96.87 | 45.88 | 92.65 | 93.25 | 96.66 | 97.59 | 97.95 |
| 3         | 96.48 | 58.29 | 98.76 | 62.96 | 99.17 | 96.69 | 98.58 | 98.96 | 97.72 |
| 4         | 99.45 | 93.03 | 99.86 | 95.22 | 98.49 | 97.26 | 100.0 | 99.86 | 99.18 |
| 5         | 99.58 | 99.17 | 100.0 | 97.53 | 100.0 | 100.0 | 100.0 | 100.0 | 99.58 |
| 6         | 85.70 | 64.03 | 93.12 | 58.90 | 90.23 | 91.05 | 96.24 | 98.15 | 94.55 |
| 7         | 73.65 | 75.56 | 88.64 | 81.86 | 79.23 | 85.74 | 87.80 | 87.01 | 82.16 |
| 8         | 88.53 | 40.59 | 96.29 | 40.54 | 94.44 | 96.29 | 98.98 | 99.33 | 96.63 |
| 9         | 97.94 | 98.66 | 99.53 | 96.48 | 96.84 | 99.92 | 99.81 | 99.60 | 98.58 |
| OA        | 85.27 | 69.13 | 93.95 | 69.68 | 90.03 | 91.23 | 94.34 | 95.25 | 92.22 |
| AA        | 89.19 | 70.90 | 95.78 | 71.97 | 93.26 | 93.57 | 96.78 | 97.31 | 95.07 |
| $\kappa$  | 82.89 | 64.41 | 92.92 | 65.28 | 88.38 | 89.76 | 93.97 | 94.45 | 90.94 |
the classification accuracy. EMPs, 3D-CAE, 3D-CNN, CNN-PPF can both improve the classification accuracy of HSIs. However, 3D-CAE, 3D-CNN and CNN-PPF require complex training procedure and the effect of EMPs on improving accuracy is limited. In contrast with the compared methods, our proposed method (VGG19+ SVM) achieves the state-of-art result and bypassings the complex training procedure of deep learning models. Meanwhile, VGG19+ RF could also provide a competitive classification result, which further demonstrates the effectiveness of the proposed method.

(Figures 6–8) show the ground truth map and the classification maps obtained by different methods. The
classification maps produced by the proposed VGG19 + SVM and VGG19+ RF are consistent with the ground truth map and have less classification noise. This further proves the effectiveness of the proposed method.

To further validate the classification performance of VGG19+ SVM, we repeat the experiments of different methods 20 times. 200 samples per class are randomly selected as the training data set in each experiment. Note that the training data set for different methods are the same in each experiment. (Figure 9) shows the distribution of $\kappa$ of different algorithms in 20 experiments through box plots. In the box plots, the horizontal line inside the box represents the median, and the upper and lower edges of the box correspond to the upper quartile and the lower quartile. The two horizontal lines connected to the box represent the maximum and minimum values, respectively. The shape of rhombus represents the outlier points in the data. From the results of (Figure 9), we can see that VGG19+ SVM outperforms the other compared methods. We also use paired t-test (Pan et al., 2017) to show the statistical evaluation about the results, the paired t-test is a widely used statistical method to verify whether there is a significant difference between the two groups of related samples. We accept the assumption that the mean $\kappa$ of VGG19 + SVM is larger than a compared method only if Equation (2) is valid.

$$\frac{(\overline{m} - \overline{n})\sqrt{n_1 + n_2 - 2}}{\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}} > t_{1-\alpha} [n_1 + n_2 - 2]$$

Where $\overline{m}$ and $\overline{n}$ are the means of $\kappa$ of VGG19+ SVM and a compared method, $s_1$ and $s_2$ are the corresponding standard deviations, $n_1$ and $n_2$ are the number of realizations of experiments reported which is set as 20 in this paper. The t values of t-test larger than 3.57 mean that two results are statistically different at the 99.9% confidence level. As listed in (Table 8), all the values are much larger than 3.57, which indicates that the increases in $\kappa$ are statistically significant.

In order to further illustrate the effectiveness of the VGG19 network structure for spatial feature extraction of HSI, we compare VGG19 with other pre-trained networks under the condition that 200 samples per class are randomly selected as the training data set, including ResNet50 (He et al., 2015) with deeper network structure and InceptionV3 (Chollet, 2016) with wider network reception domain. ResNet50 uses the first two convolutional layers for spatial feature extraction, and InceptionV3 uses the first five convolutional layers for spatial feature extraction. The classification

---

**Figure 8.** Classification maps with different classifiers for the Indian Pines data set. (a) Ground-truth map, (b) SVM, (c) PCA+SV, (d) EMP, (e) PCA+EMPs, (f) 3DCAE, (g) 3D-CNN, (h) CNN-PPF, (i) VGG19 + SVM, (j) VGG19 + RF.

**Figure 9.** $\kappa$ of 20 experiments. (a) the University of Pavia data set, (b) the Salinas data set, (c) the Indian Pines data set.
Table 8. Paired t-test value of statistical significance between different methods.

| Methods          | SVM | PCA +SVM | EMPS | EMPS +PCA | 3D-CAE | 3D-CNN | CNN-PPF |
|------------------|-----|----------|------|-----------|--------|--------|---------|
| University of Pavia | 52.54 | 182.29 | 23.80 | 60.32     | 15.50  | 12.16  | 12.16   |
| Salinas          | 33.59 | 32.68   | 18.90 | 35.25     | 9.99   | 8.06   | 24.61   |
| Indian Pines     | 50.96 | 54.95   | 9.32  | 55.03     | 19.56  | 13.02  | 3.89    |

Table 9. The overall accuracy(%) of different pre-trained networks.

| Pre-trained Network | ResNet50 | InceptionV3 | VGG19 |
|---------------------|----------|-------------|-------|
| Classifier          | SVM      | RF          | SVM   |
| University of Pavia | 95.17    | 95.77       | 98.14 |
| Salinas             | 94.86    | 94.87       | 97.51 |
| Indian Pines        | 89.46    | 88.78       | 94.43 |

Table 10. The overall accuracy(%) of different ablation structures.

| Network          | VGG19-Spectral | VGG19-Spatial | VGG19 |
|------------------|----------------|---------------|-------|
| Classifier       | SVM            | RF            | SVM   |
| University of Pavia | 91.44         | 82.01         | 98.07 |
| Salinas          | 92.52          | 89.67         | 98.39 |
| Indian Pines     | 88.49          | 84.31         | 94.43 |

Table 11. Statistics of feature extraction time.

| Methods          | EMPS     | 3D-CAE   | 3D-CNN   | CNN-PPF  | VGG19    |
|------------------|----------|----------|----------|----------|----------|
| University of Pavia | 1.45s    | 19.47 min| 15.13 min| 4.0 h    | 3.90s    |
| Salinas          | 1.03s    | 19.32 min| 18.29 min| 6.0 h    | 3.19s    |
| Indian Pines     | 0.77s    | 19.27 min| 16.29 min| 5.2 h    | 1.96s    |

results are shown in (Table 9). From (Table 9), it can be found that among the three pre-training networks, whether it is classified using SVM or RF, the classification accuracy of ResNet50 is the lowest, and the classification accuracy of InceptionV3 is slightly lower than that of VGG19, VGG19 obtains the highest classification accuracy on three data sets. Therefore, the experimental data in (Table 10) illustrates the effectiveness of VGG19 for feature extraction.

In addition, we conduct an ablation study to demonstrate the effectiveness of the proposed method in combining spatial and spectral features. Concretely, we record the classification results produced using single spectral feature and single VGG feature respectively. And the simplified model is denoted as “VGG19-Spectral” and “VGG19-Spatial”. The comparative results are demonstrated in the (Table 10). As shown in the table, whether SVM or RF is selected as the classifier, the overall classification accuracy of VGG19 is higher than that of VGG19-Spectral and VGG19-Spatial, which fully demonstrates the effectiveness of combining spatial features and spectral features of the proposed method.

The feature extraction time of different methods are listed in (Table 11). Note that deep learning based methods learn to extract features from data. Thus 3D-CAE, 3D-CNN and CNN-PPF show the training time in (Table 11). Deep neural networks’ drawback of a long training time is becoming less and less decisive the, as rapid development of hardware technology, especially of GPU. However, training a deep learning model is still very time-consuming. From (Table 11), it can be found that the time of feature extraction of deep learning model is measured in minutes or hours. In contrast with deep learning based methods, the traditional spatial-spectral feature extraction method takes less time to extract features. The proposed method does not need to train the deep learning model, thus it takes less time to extract features.

Conclusion

In this paper, a simple but effective spatial-spectral feature extraction method is proposed for HSI classification. A pretrained VGG19 model is used to extract spatial features of a HSI. This method only needs to set
the number of convolution layers for feature extraction. It is found that using the first 5 convolutional layers are more reasonable by experiment and analysis. The proposed method does not need to train the deep learning model, so the feature extraction speed is fast. The extracted spatial features are stacked with the original spectral features to form the spatial-spectral features. Experiments on three HSI datasets show that using the extracted spatial-spectral features could greatly improve the classification accuracy.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was funded by National Natural Science Foundation of China [41801388].

ORCID

Xibing Zuo http://orcid.org/0000-0002-8120-8692

References

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P. J. I. T. O. G., & Sensing, R. (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. https://doi.org/10.1109/TGRS.2016.2584107

Chen, Y., Zhao, X., Observations, J. J. S. T. I. A. E., & Remote Sensing, I. J. O. (2015). Spectral–spatial classification of hyperspectral data based on deep belief network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(6), 2381–2392. https://doi.org/10.1109/JSTARS.2015.2388577

Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., Tang, L. J. J. G., & Letters, R. S. (2017). Hyperspectral images classification with gabor filtering and convolutional neural network. IEEE Geoscience and Remote Sensing Letters, 14 (12), 2355–2359. https://doi.org/10.1109/LGRS.2017.2764915

Chollet, F. (2016). Xception: deep learning with depthwise separable convolutions. CoRR, Abs/1610.02357. http://arxiv.org/abs/1610.02357

Fauvel, M., Chanussot, J., Benediktsson, J. A., & Sweeney, J. R. (2008). Spectral and spatial classification of hyperspectral data using svs and morphological profiles. (Ed.). Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International.

Ghamisi, P., Plaza, J., Chen, Y., Li, J., Plaza, A. J. J. G., & Sensing, R. (2017). Advanced spectral classifiers for hyperspectral images: A review. IEEE Geoscience and Remote Sensing Magazine, 5(1), 8–32. https://doi.org/10.1109/MGRS.2016.2616418

Hang, R., Liu, Q., Hong, D., Ghamisi, P. J. I. T. O. G., & Sensing, R. (2019). Cascaded recurrent neural networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 57(8), 5384-5394. https://doi.org/10.1109/TGRS.2019.2899129

Haut, M., Juan, F.-B., Ruben, P., Javier, A., Paolotti, J., Mercedes, E., Geoscience, P. J. I. T. O., & Sensing, R. (2019). Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 57(2), 740-754. https://doi.org/10.1109/TGRS.2018.2860125

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. CoRR, Abs/1512.03385. http://arxiv.org/abs/1512.03385

Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, (2015), 1–12. https://doi.org/10.1155/2015/258619

Koda, S., Melgani, F., Nishii, R. J. I. G., & Letters, R. S. (2019). Unsupervised spectral–spatial feature extraction with generalized autoencoder for hyperspectral imagery. IEEE Geoscience and Remote Sensing Letters, 17(3), 469-473. https://doi.org/10.1109/LGRS.2019.2921225

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., & Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: an overview. IEEE Transactions on Geoscience and Remote Sensing, 57(9), 6690–6709. https://doi.org/10.1109/TGRS.2019.2907932

Li, W., Wu, G., Zhang, F., Du, Q. J. I. T. O. G., & Sensing, R. (2017). Hyperspectral image classification using deep pixel-pair features. IEEE Transactions on Geoscience and Remote Sensing, 55(2), 844–853. https://doi.org/10.1109/TGRS.2016.2616355

Liu, B., Gao, K., Yu, A., Guo, W., & Zuo, X. J. J. O. A. R. S. (2020). Semisupervised graph convolutional network for hyperspectral image classification. Journal of Applied Remote Sensing, 14(2), 026516. https://doi.org/10.1.1109/JRS.14.026516

Liu, B., Guo, W., Chen, X., Gao, K., Zuo, X., Wang, R., & Yu, A. (2020). Morphological attribute profile cube and deep random forest for small sample classification of hyperspectral image. IEEE Access, 8, 117906–117910. https://doi.org/10.1109/ACCESS.2020.3004968

Liu, B., Yu, X., Yu, A., Zhang, P., Wan, G., & Wang, R. (2019). Deep few-shot learning for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 57(4), 2290–2304. https://doi.org/10.1109/TGRS.2018.2872830

Liu, B., Yu, X., Yu, A., Zhang, P., & Wan, G. J. R. S. L. (2018). Spectral-spatial classification of Hyperspectral Imagery Based on Recurrent Neural Networks. Remote Sensing Letters, 9(12), 1118-1127. https://doi.org/10.1080/2150704X.2018.1519133

Liu, B., Yu, X., Zhang, P., Tan, X., Wang, R., & Zhi, L. (2018). Spectral–spatial classification of hyperspectral image using three-dimensional convolution network. 12 %. Journal of Applied Remote Sensing, 1, 016005. https://doi.org/10.1111/JRS.12016005

Liu, B., Yu, X., Zhang, P., Tan, X., Yu, A., & Xue, Z. (2017). A semi-supervised convolutional neural network for hyperspectral image classification. Remote Sensing Letters, 8(9), 839–848. https://doi.org/10.1080/2150704X.2017.1331053

Liu, B., Yu, X., Zhang, P., Yu, A., Fu, Q., & Wei, X. (2018). Supervised deep feature extraction for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 56(4), 1909–1921. https://doi.org/10.1109/TGRS.2017.2769673

Mei, S., Ji, J., Geng, Y., Zhang, Z., Li, X., Du, Q. J. I. T. O. G., & Sensing, R. (2019). Unsupervised spatial–spectral
feature learning by 3d convolutional autoencoder for hyperspectral classification. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1820-1823). IEEE. https://doi.org/10.1109/IGARSS.2017.2908756

Mou, L., Ghamisi, P., & Zhu, X. X. (2017). Deep recurrent neural networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 55(7), 3639–3655. https://doi.org/10.1109/TGRS.2016.2636241

Pan, B., Shi, Z., & Xu, X. (2017). R-VCANet: A new deep-learning-based hyperspectral image classification method. IEEE Journal of selected topics in applied earth observations and remote sensing, 10(5), 1975–1986.

Simonyan, K., & Zisserman, A. J. C. S. (2014). Very deep convolutional networks for large-scale image recognition.

Wang, Q., Li, Q., & Li, X. (2019). Hyperspectral band selection via adaptive subspace partition strategy. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(12), 4940–4950. https://doi.org/10.1109/JSTARS.2019.2941454

Yu, W., Yang, K., Bai, Y., Yao, H., & Rui, Y. (2014). Visualizing and comparing convolutional neural networks.

Yue, J., Zhao, W., Mao, S., & Liu, H. J. R. S. L. (2015). Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. Remote Sensing Letters, 6(6), 468-477. https://doi.org/10.1080/2150704X.2015.1047045

Zeller, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. (Eds.). European Conference on Computer Vision.

Zhang, C., Li, G., Du, S. J. I. T. O. G., & Sensing, R. (2019). Multi-scale dense networks for hyperspectral remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing, 57(11), 9201-9222. https://doi.org/10.1109/TGRS.2019.2925615

Zhang, H., Li, Y., Jiang, Y., Wang, P., Shen, C. J. I. T. O. G., & Sensing, R. (2019). Hyperspectral classification based on lightweight 3-D-CNN with transfer learning. IEEE Transactions on Geoscience and Remote Sensing, 57(8), 5813–5828.

Zhang, M., Gong, M., Mao, Y., Li, J., Wu, Y. J. I. T. O. G., & Sensing, R. (2018). Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network. IEEE Transactions on Geoscience and Remote Sensing, 57(5), 2669-2688. https://doi.org/10.1109/TGRS.2018.2876123

Zilong, Z., Jonathan, L., Zhiming, L., Chapman, M., Geoscience, C. J. I. T. O., & Sensing, R. (2017). Spectral–spatial residual network for hyperspectral image classification: A 3-D DEEP LEARNING FRAMEWork. https://doi.org/10.1109/TGRS.2017.2755542