A CNN-based Spatial Feature Fusion Algorithm for Hyperspectral Imagery Classification

Alan J.X. Guo *Member, IEEE*, and Fei Zhu *Member, IEEE* *†

April 30, 2019

Abstract

The shortage of training samples remains one of the main obstacles in applying the neural networks to the hyperspectral images classification. To fuse the spatial and spectral information, pixel patches are often utilized to train a model, which may further aggregate this problem. In the existing works, an ANN model supervised by center-loss (ANNC) was introduced. Training merely with spectral information, the ANNC yields discriminative spectral features suitable for the subsequent classification tasks. In this paper, we propose a novel CNN-based spatial feature fusion (CSFF) algorithm which allows a smart integration of spatial information to the spectral features extracted by ANNC. As a critical part of CSFF, a CNN-based discriminant model is introduced to estimate whether two pixels belong to the same class. At the testing stage, by applying the discriminant model to the pixel pairs generated by a test pixel and each of its neighbors, the local structure is estimated and represented as a customized convolutional kernel. The spectral-spatial feature is generated by a convolutional operation between the estimated kernel and the corresponding spectral features within a local region. The final label is determined by classifying the resulting spectral-spatial feature. Without increasing the number of training samples or involving pixel patches at the training stage, the CSFF framework achieves the state-of-the-art by declining $20\% - 50\%$ classification failures in experiments on three well-known hyperspectral images.

1 Introduction

A hyperspectral image is taken as a collection of spectral pixels. Each of them records a radiance vector acquired over a same land-cover, with hundreds of spectral bands across a certain wavelength range [1]. The classification of these spectral pixels into a set of land-cover materials is a crucial task in hyperspectral images analysis [2]. To address this issue, an amount of spectrum-based works are proposed mainly by exploiting the abundant spectral information containing in the data. Among, linear algorithms in dimension reduction and feature classification are the most investigated, for example, principal component analysis (PCA) [3, 4], independent component analysis (ICA) [5, 6] and linear discriminant analysis (LDA) [7]. Nonlinear models and their extensions are also introduced to achieve better representations of the spectra, including but not limited to support vector machine (SVM) [8], manifold learning [9], random forest [10] and kernel-based strategies [11, 12].

Benefiting the increasing imaging qualities of both spectral and spatial resolutions [13], numerous spectral-spatial algorithms have been developed in order to obtain more accurate classification performance [14–20]. Specifically, a multi-scale adaptive sparse representation (MASR) method is proposed in [21], where the spatial information at different scales is explored simultaneously. In MASR, each neighboring pixel is represented via an adaptive sparse combination of training samples, and the label of the centering test pixel is determined by the recovered sparse coefficients. It is noteworthy that although MASR utilizes the spatial information at the testing stage, only spectral information is engaged at the training stage. This helps to mitigate the shortage of training data to some degree.

Deep learning frameworks, including artificial neural networks (ANN), convolutional neural networks (CNN), and recurrent neural network (RNN) have been successfully applied in many fields related to machine learning

*The work was supported in part by the National Natural Science Foundation of China under Grant 61701337 and the Natural Science Foundation of Tianjin under Grand 18JCQNJC01600. *(Corresponding author: Fei Zhu)*

†A. Guo and F. Zhu are with the Center for Applied Mathematics, Tianjin University, China. (jiaxiang.guo@tju.edu.cn; fei.zhu@tju.edu.cn)

and signal processing [22–26]. Recently, the neural network-based models have been utilized in hyperspectral images classification, achieving remarkable improvements over the traditional methods in terms of classification performance. Earlier works include the stacked autoencoder (SAE) [27–29], the deep belief network (DBN) [30] and *etc.*. More recently, many researches are dedicated to the varieties of CNN and RNN-based models, as studied in [31–41].

Training with pixel patches is a natural idea to take advantage of both spectral and spatial information, and is adopted by most of the aforementioned neural network-based studies [31–38, 41, 42]. Representative works include the so-called 3D-CNN in [32], where pixel patches are directly fed to the deep model, and the integrated spectral-spatial features can be extracted from the hyperspectral data. However, it should be noticed that this strategy may further aggravate the shortage of training data. Different from the numerous accessible RGB images, the labeled samples are limited in hyperspectral imagery, which are usually insufficient for network training [32]. Compared with the spectrum-based models, the pixel-patch-based ones aggregate this contradiction from following two aspects: i) using pixel patches often complicates the model by introducing additional undetermined parameters, thus requiring more training data, and ii) a hyperspectral image usually contains fewer mutually non-overlapped patches with specified size than pixel-wise samples.

Several attempts have been made to alleviate the shortage of training samples, including the virtual sample strategy [32] and the pre-training and fine-tuning techniques [37, 38]. Moreover, to avoid training with pixel patches, frameworks have been built based on concatenating a spectrum-based model and a post spatial information integration scheme. The pixel-pair feature algorithm (PPF) [43] is based on a CNN model trained with pixel pairs. For testing, pixel pairs are firstly generated by the centering testing pixel and its neighbors, and then classified by the trained model. A voting strategy is adopted to determine the final classification result. More recently, the ANNC with adaptive spatial-spectral center classifier (ANNC-ASSCC) is proposed in [39], where the spectral features are extracted by a well-designed ANN jointly supervised by the softmax loss and the center loss, and the spectral-spatial fused feature is generated at the testing stage for classification. These two state-of-the-art methods are briefly revisited in Section 2.

In this paper, we choose the ANNC network in [39] as the spectral feature extraction model, and design a CNN-based Spatial Feature Fusion (CSFF) algorithm to posteriorly integrate the spatial information [1]. The main characters of the proposed classification framework are listed as below.

1. Rather than implicit spatial information utilized in [39], explicit spatial structures are produced by our novel CNN-based model. Therefore, the CSFF works more reasonably than the averaging fusion strategy in the previous work [39].

2. The CSFF algorithm shares the same training data with the spectral feature extraction model, which is chosen as ANNC in this paper. Without increasing the size of the training set, this settlement keeps the whole framework working on a small amount of training data.

3. To enhance the representative ability of the framework, the spatial structure extraction algorithm in CSFF is designed to have a totally different network structure from ANNC, such that the two models are distinguished at feature expression and abstraction levels. To be precise, the ANNC expresses the shallow features of spectra, while the spatial structure extraction algorithm produces more abstract features with a deeper and more complex model structure. This helps to alleviate the correlation between these two models, thus increasing the performance of the whole framework.

Taking advantage of the above characters, without requiring more training data, the proposed CSFF algorithm allows to achieve the state-of-the-art by declining $20\% - 50\%$ classification failures in experiments. In addition, the spectral-spatial features by the proposed framework are shown to be effective and suitable for different classifiers.

The reminder of this paper is organized as follows. We firstly review the related works in Section 2. The whole framework outline and the CSFF algorithm are discussed in Section 3. Section 4 reports the experimental results and analysis on three datasets. In Section 5, some concluding remarks are presented.

[1] The code of this work is available at: https://github.com/aalennku/CSFF.

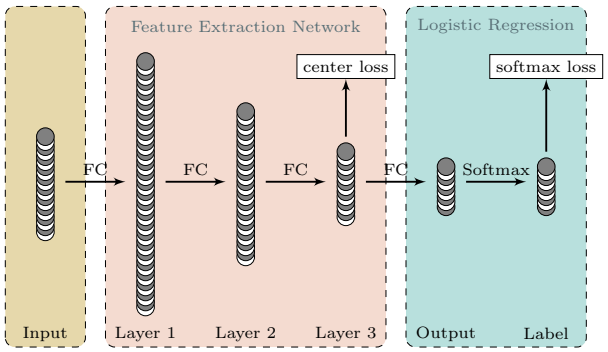


Figure 1: Structure of the spectral feature extracting network [39].

2 Related works

2.1 Pixel-pair feature algorithm (PPF)

In [43], the PPF framework investigates a pixel-pair-based classification model based on CNN, in order to address the multi-classification task. Let x_1 and x_2 be two pixels with label $\ell(x_1)$ and $\ell(x_2)$, respectively. The pixel-pair for training is generated as (x_1, x_2), with the label determined by

$$\ell((x_1, x_2)) = \begin{cases} \ell(x_1) & \text{if } \ell(x_1) = \ell(x_2), \\ 0 & \text{otherwise.} \end{cases}$$

The pixel-pair strategy is applied for alleviating the shortage of training data. It is observed that the number of pixel-pairs computes $\binom{N}{2}$, where N is the size of the training set. This enlarged size of the training set enables the training of a deep CNN model. The model classifies the pixel-pairs into $K+1$ classes with labels 0 and $1, 2, \ldots, K$, by

- class i, if two pairing pixels are from the same class i, for $1 \leq i \leq K$;
- class 0, otherwise.

Here, K is the number of pixel classes. At the testing stage, pixel-pairs are firstly generated by the testing pixel and its neighbors, and then classified into $K+1$ classes by the learned CNN model. The final label of a test pixel is determined by a voting strategy among the non-zero classification results.

2.2 ANNC with adaptive spatial-spectral center classifier (ANNC-ASSCC)

In [39], an ANN-based feature extraction and classification framework is proposed. The so-called ANNC-ASSCC algorithm consists of two successive steps, namely i) the spectral feature extraction with ANN, and ii) the adaptive spatial information fusion and classification.

In the first step, a simple but efficient ANN structure is designed to extract spectral features. As illustrated in Fig. 1, the network contains 4 fully connected layers, and the numbers of neurons in Layer 1,2 and 3 are set to 512,256 and 32. A joint supervision of softmax loss and center loss is applied for classification task. During the training stage, the k class centers, *i.e.* $c_1, c_2, \ldots c_k$, are updated by averaging the 3-rd layer's outputs within respective classes. For a training pixel, the introduced center loss encourages the output of the 3-rd layer to gather around its corresponding class center in Euclidean space. At the testing stage, the first 3 layers of the learned network are utilized and the outputs of the 3-rd layer are regarded as the extracted spectral features. In the second step, the spatial information is merged by averaging the extracted spectral features within neighboring regions of different sizes. After the resulting features are classified, a voting strategy is applied to generate the final prediction.

3 Framework outline and the CNN-based spatial feature fusion algorithm

In this section, we present an end-to-end spectral-spatial feature extraction and classification framework. The whole structure is firstly outlined, with detailed explanations on each of the three components. As a key ingredient of the whole framework, the CSFF algorithm is introduced with emphasis placed on the newly proposed CNN-based discriminant model, which outputs the predicted probability of two pixels belonging to the same class.

3.1 Framework outline

As discussed previously, training with pixel patches may further aggregate the lack of labeled samples. Based on this concern, the training stage of the proposed framework is performed merely using the spectral data, while the spatial information is involved posteriorly at the testing stage. The proposed framework is designed to consist of three parts, namely

- spectral feature extraction;
- spatial structure extraction and fusion of spectral feature and spatial structure;
- classification of the spectral-spatial feature.

The flowchart of the whole framework is given in Fig. 2.

In the first part, we extract the spectral features of the whole hyperspectral image by directly applying the ANNC model proposed in [39]. This spectrum-based model is supervised with a joint loss of center loss and softmax loss, producing discriminative spectral features with inner-class compactness and inter-class variations. Taking advantage of this, the spectral features obtained by the learned ANNC model is suitable for the successive spatial information fusion step. In the following, we use the notations x and $f_{\text{ANNC}}(x)$ to denote a given pixel and its corresponding spectral feature extracted by the ANNC model.

In the second part, a spatial structure extraction algorithm is designed to explore the local structure for a given pixel within some pre-defined neighborhood. More precisely, the local structure is characterized by a customized "convolutional kernel". The fusion of the spectral-spatial information is realized by a "convolutional" operation between the kernel and the extracted spectral features within the neighborhood. Let $N(x)$ be a neighborhood centering at pixel x and $W(N(x))$ be the customized "convolution kernel". Use the notation $f_{\text{ANNC}}(N(x))$ to represent the data cube formed by the extracted spectral features within the neighborhood $N(x)$, where the i, j-th entry is defined by

$$f_{\text{ANNC}}(N(x))_{i,j} = f_{\text{ANNC}}(N(x)_{i,j}). \tag{1}$$

Using $f_{N(x)}(x)$ to denote the spectral-spatial feature of x within the neighborhood $N(x)$, it is defined by

$$f_{N(x)}(x) = W(N(x)) \bigotimes f_{\text{ANNC}}(N(x)), \tag{2}$$

where $\bigotimes$ indicates the convolutional operation. A more clear interpretation of this model will be given in Section 3.2.

The last part of the framework aims to classify the resulting spectral-spatial feature $f_{N(x)}(x)$. In [39], a center classifier is introduced, which assigns the label to each sample according to its nearest class center. Let $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k$ be the class centers estimated from the extracted spectral features of the training set. The label of the spectral-spatial feature $f_{N(x)}(x)$ is predicted by

$$\ell(f_{N(x)}(x)) = \arg\min_i \{||f_{N(x)}(x) - \hat{c}_i||_2\}. \tag{3}$$

Experiments in Section 4.2 will show that the spectral-spatial features are not sensitive to the applied classifiers, that similar classification accuracies are obtained by using the center classifier, the support vector machine (SVM) algorithm, and the k-nearest neighbors (kNN) algorithm. However, the center classifier is still the simple and straightforward one, and is engaged as the default classifier in the proposed framework.

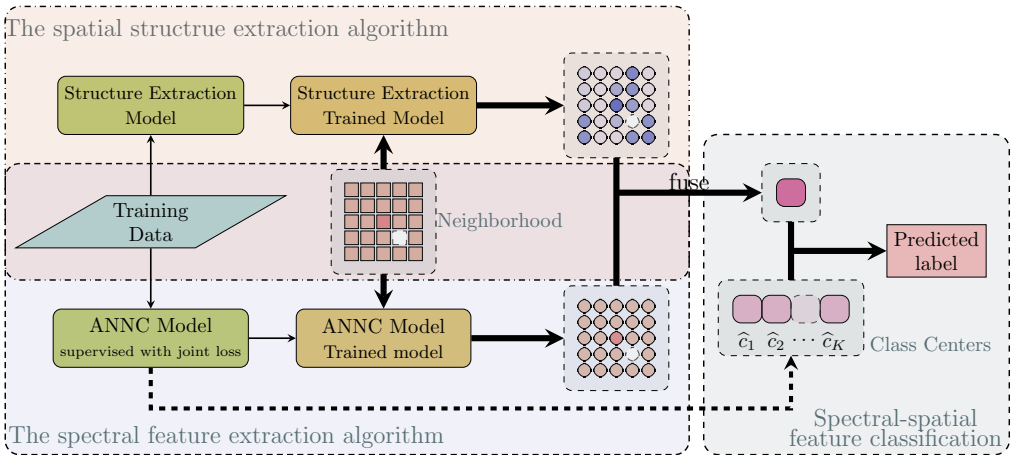


Figure 2: Flowchart of the proposed framework. The three components are represented by three dashed rectangles. The training and testing stages are distinguished by thin and thick solid arrows, respectively. The dashed arrows represent the estimation of class centers of features. The intersection of neighborhood and training samples are represented by the white pixel, and are excluded at the testing stage.

3.2 Spatial structure extraction algorithm

The most crucial aspect of CSFF is the spatial structure extraction algorithm. This algorithm relies on a CNN-based discriminant network, which is designed to predict the probability that an input pair of pixels have the same label. By applying this model to the pixel-pairs generated by the test pixel and its neighbors, the corresponding local spatial structure can be extracted and represented as a matrix that maps the neighborhood. The procedure is illustrated in Fig. 3. To be more precise, use $D((x, x'))$ to denote the predicted probability that the entries of pixel-pair (x, x') have the same class label. Let $\{x\} \times N(x) = \{(x, x') | \forall x' \in N(x)\}$ be a set of pixel pairs generated by the centering pixel x and its neighbors within $N(x)$, we use $D(\{x\} \times N(x))$ to denote the corresponding spatial matrix, whose i, j-th entry is defined by

$$D(\{x\} \times N(x))_{i,j} = D((x, N(x)_{i,j})). \tag{4}$$

By applying an element-wise threshold function f_t, the real-valued $D(\{x\} \times N(x))$ is calibrated to a binary matrix

$$D_t(\{x\} \times N(x)) = f_t(D(\{x\} \times N(x))), \tag{5}$$

where t is the pre-determined threshold value. The convolutional kernel $W(N(x))$ in(2) is calculated by normalizing $D_t(\{x\} \times N(x))$ such that all the elements add up to a unit. In essence, the convolutional kernel $W(N(x))$ records the positions where the spectra are supposed to have the same class label as the centering test pixel. Thus, only the neighboring pixels which correspond to the non-zero positions in $W(N(x))$ will contribute to the estimation of spectral-spatial feature in(2), in a equal manner. In practice, both the size of neighborhood $N(x)$ and the threshold value t should be appropriately chosen, as to be analysed in Section 4.2.

To enhance the hierarchical representative capability of the whole framework, the aforementioned discriminant model is designed to extract the deep features, while the spectral feature extraction model (ANNC) focuses on exploring the shallow features with a simple network structure. From this point of view, a diverse structure from ANNC should be considered, in order to make the proposed discriminant model different from ANNC, namely

1. While ANNC uses a structure of fully connected layers, the CNN architecture will be employed in the discriminant model. A CNN-based model is not only built differently from ANNC model, but is also expected to be more powerful in expressing hierarchical features.

2. While ANNC uses a shallow structure of 3 layers, a deep model with more layers will be engaged in the discriminant model. In most cases, deep networks enable the extraction of more abstract features than the respectively shallow ones.

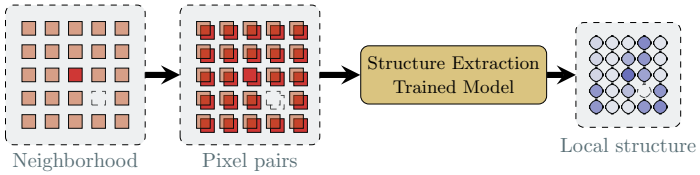


Figure 3: Flowchart of the spatial structure extraction algorithm. Pixel pairs are firstly generated by the center pixel and its neighbors and then fed into the model. For each pixel-pair, the model predicts the probability that the two elements share the same class label. The spatial structure is extracted as the output of this discriminant model.

It is noteworthy that training a complex discriminant model with a limited number of labeled pixels is possible. To address the task of distinguishing whether two spectra belong to the same class, the discriminant model is designed to be fed with pixel-pairs at both training and testing stages. Without considering the geometric information, the pixel-pairs for training are generated by the pixels chosen from training data. According to this settlement, the training set can be enlarged to roughly have a squared quantity of training samples. In practice, the enlarged training set is sufficient for training our discriminant model. More details on the training set is given in Section 4.2.1.

Based on above discussions, we choose to modify the pixel-pair features (PPF) model proposed in [43] and use the resulting network as the discriminant model in our spatial structure extraction algorithm.

Fortunately, the PPF model is not only CNN-based, but also has a relatively deep structure with 9 convolutional or fully connected layers and 3 pooling layers. This meets the directions discussed above for structure design of our discriminant model. Moreover, the binary classification task in our problem is relatively easier than the original multi-classification task in PPF, that should be well-tackled by a similar model with PPF. By modifying the first data layer and the last two fully connected layers, the structure is able to address different datasets and is suitable for the binary classification task. The modified structure of PPF model is applied as our discriminant model, which is a part of the spatial feature extraction algorithm.

The structure details of the discriminant model are illustrated in Fig. 4. Considering the slim shape of the input data, namely $(2, L)$, with L being the number of spectral channels, the convolution kernels and the pooling regions are chosen as narrow rectangular ones instead of the traditional squared ones. All the strides of the convolutional layers are set to be 1 and the pooling layers are defined to use the max-pooling function. In order to introduce nonlinearity to the CNN-based model, the commonly-used rectified linear unit (ReLU) function, defined by $\phi(x) = \max\{0, x\}$, is applied after every convolutional and fully connected layer.

4 Experimental settings and result analysis

4.1 Datasets description

Experiments are performed on three real hyperspectral images, namely the Pavia University scene, the Salinas scene, and the Pavia Centre scene[2]. The first data is the Pavia University scene, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. After removing the noisy bands and a blank strip, a sub-image of 610×340 pixels with 103 spectral bands are retained for analysis. The image is characterized by a spatial resolution of about 1.3 meters. As summarized in TABLE 1, this area is known to be mainly composed by $K = 9$ classes of materials, denoted by labels from 1 to 9. The background pixels are represented by label 0, and will not be taken into account for classification. Fig. 5 presents the false color composite and groundtruth map.

The second one is the Salinas scene collected by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS). This dataset contains 512×217 pixels, and is characterized by a resolution of 3.7 meters. After removing the water absorption bands, the remaining 204 (out of 224) bands are utilized. According to the available groundtruth information in TABLE 2, there are $K = 16$ composition categories of interest, with the background pixels represented by label 0. False color composite and groundtruth map of the Salinas scene are shown in Fig. 6.

[2]The datasets are available online: `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`

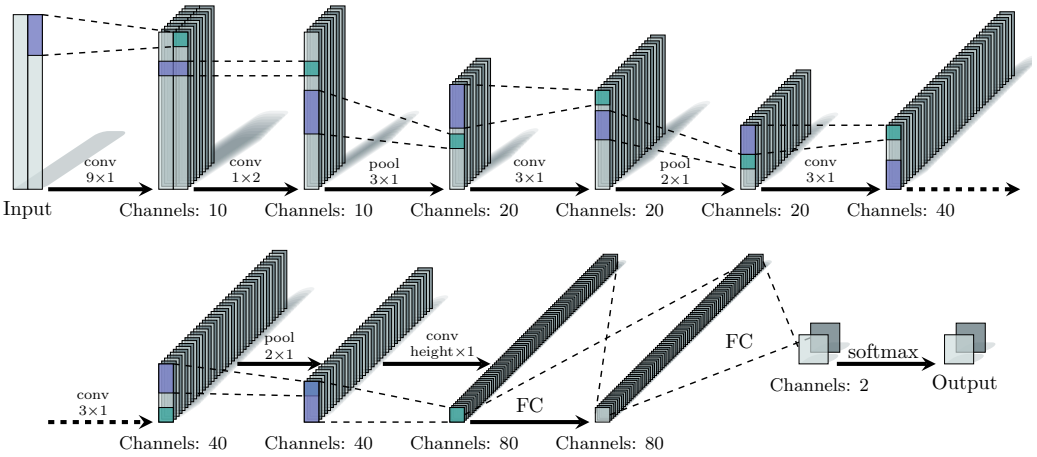


Figure 4: Structure of the discriminant model, modified from PPF [43]. The sizes of convolution kernels are marked under the layer type conv, while the cardinalities of channels are presented under the graphical data blobs. In fact, each channel of a preceding data blobs is connected to all the channels of latter data blobs in the convolutional layers, but only one connection is drawn for simplicity.

Table 1: Reference classes and sizes of training and testing sets of Pavia University image

No.	Class	Cardinality	Train	Test
1	Asphalt	6631	200	6431
2	Meadows	18649	200	18449
3	Gravel	2099	200	1899
4	Trees	3064	200	2864
5	Painted metal sheets	1345	200	1145
6	Bare Soil	5029	200	4829
7	Bitumen	1330	200	1130
8	Self-Blocking Bricks	3682	200	3482
9	Shadows	947	200	747
Total		42776	1800	40976

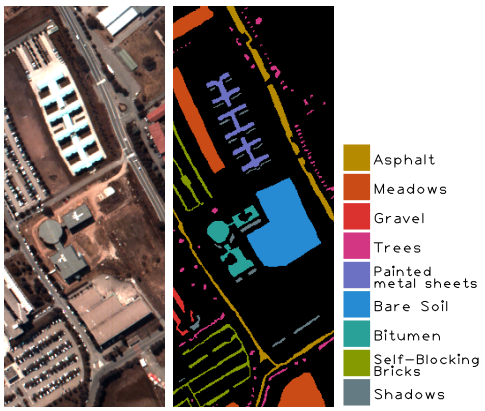


Figure 5: The false color composite (band $10, 20, 40$) and groundtruth representation of Pavia University

Table 2: Reference classes and sizes of training and testing sets of Salinas image

No.	Class	Cardinality	Train	Test
1	Brocoli green weeds 1	2009	200	1809
2	Brocoli green weeds 2	3726	200	3526
3	Fallow	1976	200	1776
4	Fallow rough plow	1394	200	1194
5	Fallow smooth	2678	200	2478
6	Stubble	3959	200	3759
7	Celery	3579	200	3379
8	Grapes untrained	11271	200	11071
9	Soil vinyard develop	6203	200	6003
10	Corn senesced green weeds	3278	200	3078
11	Lettuce romaine 4wk	1068	200	868
12	Lettuce romaine 5wk	1927	200	1727
13	Lettuce romaine 6wk	916	200	716
14	Lettuce romaine 7wk	1070	200	870
15	Vinyard untrained	7268	200	7068
16	Vinyard vertical trellis	1807	200	1607
Total		54129	3200	50929

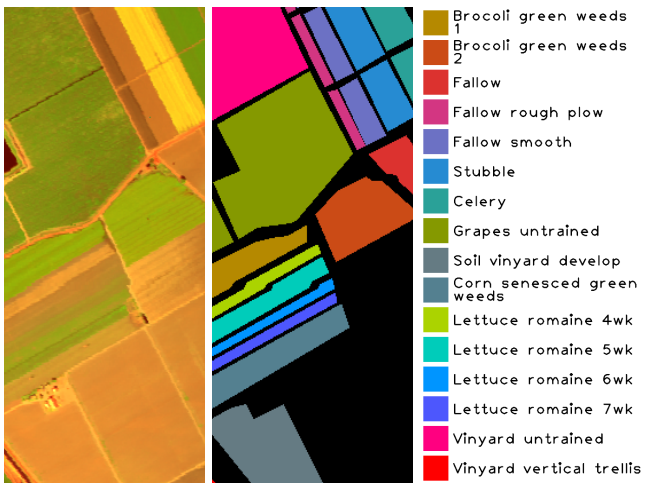


Figure 6: The false color composite (band $180, 100, 10$) and groundtruth representation of Salinas

The last image is the Pavia Centre scene, which is also returned by the ROSIS sensor over Pavia, northern Italy. A sub-image of 1096×715 pixels with 102 relative clean bands is taken into account, where the geometric resolution is 1.3 meters. Ignoring the background pixels, the groundtruth labels fall into $K = 9$ reference classes, as given in TABLE 3. The false color composite and the groundtruth map of Pavia Center are shown in Fig. 7.

Table 3: Reference classes and sizes of training and testing sets of Pavia Centre image

No.	Class	Cardinality	Train	Test
1	Water	65971	200	65771
2	Trees	7598	200	7398
3	Asphalt	3090	200	2890
4	Self-Blocking Bricks	2685	200	2485
5	Bitumen	6584	200	6384
6	Tiles	9248	200	9048
7	Shadows	7287	200	7087
8	Meadows	42826	200	42626
9	Bare Soil	2863	200	2663
Total		148152	1800	146352

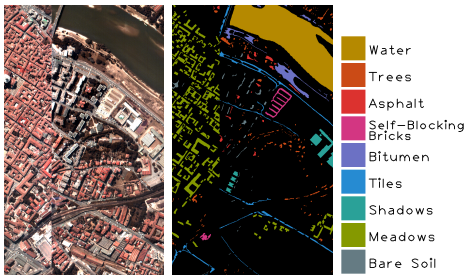


Figure 7: The false color composite (band $10, 20, 40$) and groundtruth map of Pavia Centre

4.2 Experimental settings

4.2.1 Training and testing sets

Before further processing, each dataset is firstly normalized to have zero mean and unit variance. To form the training set, 200 pixels are randomly chosen from each class. The two neuron networks, *i.e.*, the ANNC model and discriminant model, are trained by employing the same training set. TABLES 1–3 present the sizes of training and testing sets in three datasets.

To train the ANNC model, the original training set is enlarged by virtual samples [32,39], until the size of each class reaches 80000. Using x_1, x_2 to denote two training pixels chosen from the same class ℓ, a virtual sample $\tilde{x}$ with the same label is generated by $\tilde{x} = qx_1 + (1-q)x_2$, where q is a random number chosen from the uniform distribution on $[-1, 2]$. The testing set is composed by all the unused pixels, which are directly fed to the learned ANNC model at the testing stage.

To train the discriminant model in the spatial feature extraction algorithm, the pixel-pairs should be firstly generated. Let Tr_i be the set of training pixels with label i, for $i = 1, 2, \ldots, K$. The positive training set is expressed by

$$\begin{aligned} Tr_+ &= \bigcup_i Tr_i \times Tr_i \\ &= \bigcup_i \{(x_1, x_2) | \, \forall x_1, x_2 \in Tr_i\}. \end{aligned} \tag{6}$$

The negative training set is generated by taking the Cartesian product between any two different classes of training pixels, namely

$$\begin{aligned} Tr_- &= \bigcup_{i \neq j} Tr_i \times Tr_j \\ &= \bigcup_{i \neq j} \{(x_1, x_2) | \, \forall x_1 \in Tr_i, \, x_2 \in Tr_j\}. \end{aligned} \tag{7}$$

Compared with the original training set with 200 pixels for each class, the enlarged training set composed by pixel pairs is sufficient to train the discriminant model without causing overfitting problem. Taking Pavia University for example, the original training set consists of $200 \times 9 = 1800$ samples. According to (6) and (7), the sizes of the positive training set Tr_+ and the negative training set Tr_- compute 360000 and 2880000, respectively. In practice, considering that the size of negative training set Tr_- is overwhelmingly greater than that of the positive set Tr_+, only half of the negative pixel pairs are randomly chosen for training. It is also noteworthy that, the training pixels are excluded from either the testing samples or their neighbors on all the three datasets.

4.2.2 Networks configurations

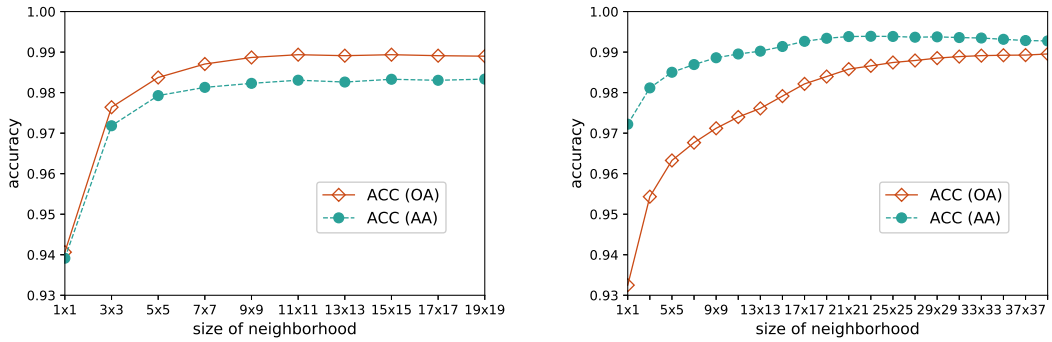
The parameters in the ANNC network are set as recommended in [39]. To be precise, the weight of center loss is set to be $\lambda = 0.01$. To train ANNC, the stochastic gradient descent (SGD) is applied with a mini-batch size 512, and the learning rate is initialized by 0.01 and decays by multiplying 0.3162 every 20000 steps. This model is implemented on the open source deep learning framework Caffe [44].

The discriminant model has a softmax layer at the top followed by the cross entropy loss function. At the training stage, the SGD algorithm is used with the mini-batch size set to be 512, where the learning rate is initialized by 0.01 and decays every 50 epochs by multiplying 0.1. For the convenience of using rectangular convolutional kernels, this model is implemented on the compatible machine learning framework Tensorflow [45], following the codes of [43].

4.2.3 Hyperparameter selection

We discuss how to select the two crucial hyperparameters introduced by the proposed CSFF framework, *i.e.*, the size of neighborhood $N(x)$, and the calibration threshold value t in (5). Firstly, to study the influence of neighborhood size on classification accuracies, we vary this parameter within some range, while fixing the threshold to a modest value with $t = 0.01$. Experiments are performed on datasets Pavia University and Salines, with the neighborhood size varying within the sets $\{1 \times 1, 3 \times 3, ..., 19 \times 19\}$ and $\{1 \times 1, 3 \times 3, ..., 39 \times 39\}$, respectively. As presented in Figure 8, an increasing neighborhood size generally leads to improvements on classification performance. This phenomenon is not surprising. Because of the good property of the convolutional kernel, only the useful spatial information from the neighboring spectra will contribute to the fusion of the spectral-spatial feature corresponding to the centering test pixel. A relatively larger neighborhood usually accounts for more useful spatial information, thus being favorable to boost the classification performance. However, it is also noticed that at the testing stage, a doubled neighborhood radius will quadruple the computational complexity. By balancing the computational cost and the classification accuracy, the neighborhood size is set to be 19×19 for Pavia datasets and 39×39 for Salinas dataset.

Secondly, we validate the selection of threshold value t in (5), which calibrates the probabilities in the spatial matrix into binary predictions. Experiments are performed on Pavia University and Salinas datasets by varying t within interval $[0, 1]$, where the neighborhood sizes are fixed as 19×19 and 39×39, respectively. As the learned discriminant model allows to estimate whether two paring pixels belong to the same class (probability near 1) or not (probability near 0), the elements presented in spatial matrices are generally distributed close to 0 and 1. Thus, the classification performance tends to be particularly sensitive to the values of t close to 0 and 1. Accounting for this fact, the threshold value t is set by $t = \frac{1}{1+e^{-0.7x}}$, where $x \in \mathbb{N}$ and $x \in [-9, 11]$. Two extreme cases with $t = 0$ and $t = 1$ are also examined. It is noticed that with $t = 0$, the proposed spectral-spatial feature fusion algorithm in (2) is reduced to a simple average-over-neighborhood algorithm, while with $t = 1$, the algorithm does not exploit any spatial information for feature fusion. The classification results using different threshold values are given in Figure 9. As observed, on both datasets, small threshold values close to $t = 0$ lead to promising classification accuracies, that are much more advantageous over the results obtained by $t = 0$ and $t = 1$. In this paper, we apply a unified threshold value with $t = 0.01$ in all the experiments on three datasets.



(a) Pavia University (b) Salinas

Figure 8: Comparisons of classification accuracies in terms of AA and OA, along with varying neighborhood sizes, on (a) Pavia University and (b) Salinas.

4.2.4 Classifiers

To evaluate the performance of the spectral-spatial features $f_{N(x)}(x)$, a comparative study is performed by classifying them using three classifiers, namely the center classifier mentioned in(3), the kNN algorithm [46], and the SVM algorithm [47]. Without involving extra training data, these three classifiers are trained based on the same training pixels as used in training the ANNC model and the discriminant model. Precisely, by applying a transfer learning strategy [48], the classifiers are trained with spectral features (with their respective labels) extracted by the learned ANNC model from the training pixels. Here, the spectral-spatial features of the training pixels are not directly used for training the classifiers, for the sake that no additional information from the pixels other than the training ones should be used before the testing stage. For a given training pixel x, its neighborhood $N(x)$ used for spectral-spatial feature generation may contain the pixels from the testing set. Hence, the resulting spectral-spatial feature $f_{N(x)}(x)$ may contain the information from the testing set, and is not proper for training the classifiers. In practice, the class centers in center classifier are estimated by averaging the spectral features of training pixels within each class. Concerning kNN, it assigns a test spectral-spatial feature to the class most common among its k nearest training spectral features, where two cases with $k = 5$ and $k = 10$ are considered in the experiments. Similar to the kNN algorithm, the SVM algorithm is also trained by the spectral features with their respective labels. The rbf kernel is applied and the multi-class classification is handled according to a one-vs-one scheme. The Python packages SciPy [49] and Scikit-learn [50] are applied directly for kNN and SVM algorithms, with the parameters set to be the default values.[3]

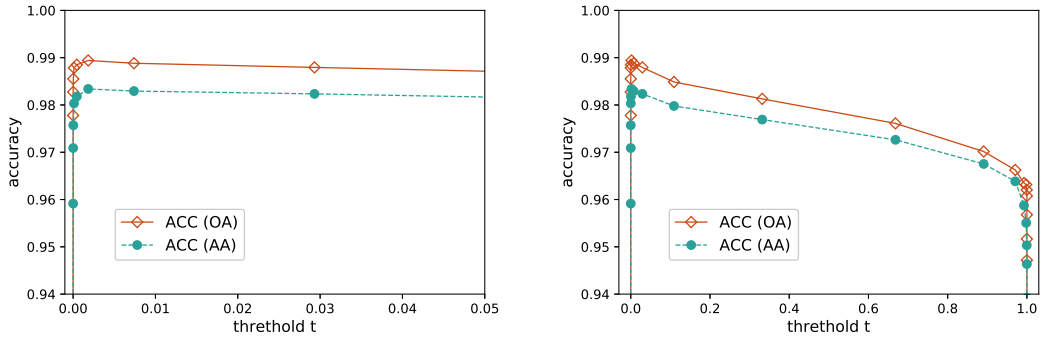
4.3 Results analysis

We evaluate the classification performance of the proposed framework on the aforementioned datasets. In this framework, the models are trained merely based on spectral pixels without using any spatial information, as the latter is likely to coincide with the testing set. From this aspect, three state-of-the-art methods are considered, which are the traditional method MASR [21], the CNN-based PPF [43], and the ANN-based ANNC-ASSCC [39]. As the state-of-the-art baseline of CNN models, the 3D-CNN [32] is also compared, which is trained based on pixel patches. For fair comparison, all the comparative methods are trained using the training sets of same size, namely 200 random pixels for each class.

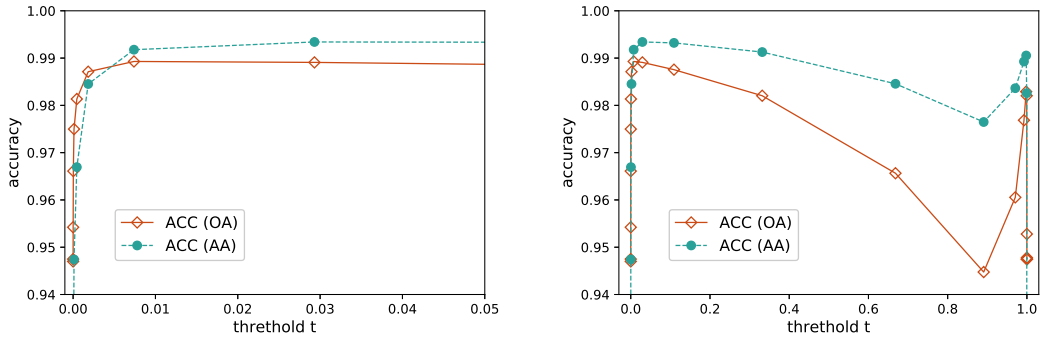
Three commonly-used metrics are adopted to evaluate the classification performance globally, which are the overall accuracy (OA), the average accuracy (AA), and the Kappa coefficient (κ). Briefly, OA represents the overall percentage of testing samples that are correctly classified, while AA calculates the average value of the accuracies of each class on testing samples. The κ coefficient measures the agreement between the predicted labels and groundtruth labels.

Figure 10 visualizes the spatial distribution of spectral features extracted by ANNC and ANN models in [39], on testing samples from Pavia Centre. As expected, the features extracted by ANNC have smaller within class

[3]The softwares are available at: https://www.scipy.org/ and https://scikit-learn.org/.

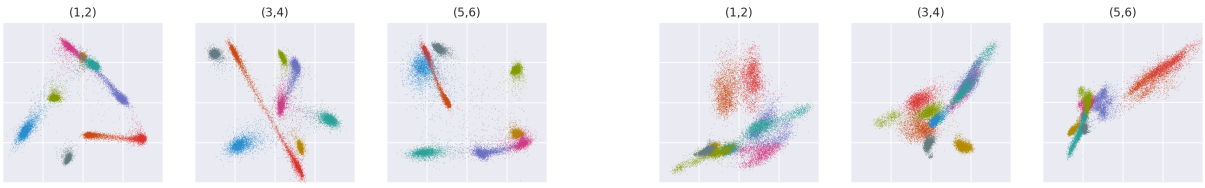


(a) Pavia University



(b) Salinas

Figure 9: Comparisons of classification accuracies in terms of AA and OA, along with varying threshold value t belonging to different intervals, namely $t \in [0, 0.05]$ for the left and $t \in [0, 1]$ for the right, on (a) Pavia University and (b) Salinas.



(a) Spectral features under ANNC

(b) Spectral features under ANN

Figure 10: Visualization of the spectral features extracted by (a) ANNC and (b) ANN, on Pavia Centre. 2 coordinates (i, j) out of 32 dimensions are visualized. For each of the $K = 9$ classes, 2000 randomly chosen testing samples are utilized. Features from various classes are marked with different colors.

Table 4: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of MASR, MASR-t, PPF, 3D-CNN, ANNC-ASSCC and CSFF on Pavia University scene

	MASR	MASR-t	PPF	3D-CNN	ANNC-ASSCC	
Asphalt	89.97 ± 1.81	45.21 ± 3.61	97.25 ± 0.35	95.18 ± 1.33	98.69 ± 0.78	98
Meadows	98.78 ± 0.36	94.75 ± 1.26	95.24 ± 0.42	98.90 ± 0.28	99.97 ± 0.03	99
Gravel	99.78 ± 0.47	81.15 ± 4.51	94.17 ± 0.49	95.55 ± 1.63	93.85 ± 2.15	96
Trees	97.47 ± 0.42	95.75 ± 0.89	97.20 ± 0.30	99.12 ± 0.50	96.68 ± 0.99	98
Painted metal sheets	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	99
Bare Soil	99.87 ± 0.23	78.49 ± 2.81	99.37 ± 0.19	98.23 ± 1.14	100.00 ± 0.00	10
Bitumen	100.00 ± 0.00	99.56 ± 0.14	96.16 ± 0.21	91.04 ± 5.60	96.88 ± 1.24	98
Self-Blocking Bricks	98.76 ± 0.63	86.79 ± 3.87	93.83 ± 0.59	98.17 ± 0.79	93.11 ± 2.50	95
Shadows	92.17 ± 1.78	47.98 ± 2.28	99.46 ± 0.15	99.68 ± 0.56	97.62 ± 0.37	10
OA(%)	97.42 ± 0.36	83.25 ± 0.93	96.25 ± 0.20	98.14 ± 0.10	②$\mathbf{98.55 \pm 0.16}$	①9
AA(%)	②$\mathbf{97.42 \pm 0.40}$	81.08 ± 0.77	96.97 ± 0.08	97.32 ± 0.68	②$\mathbf{97.42 \pm 0.22}$	①9
κ	0.9654 ± 0.0048	0.7764 ± 0.0119	0.9499 ± 0.0026	0.9687 ± 0.0015	②$\mathbf{0.9805 \pm 0.0021}$	①0.98

distances and bigger between class distances, compared to the case with ANN.

As illustrated in TABLES 4–6, competitive classification results are obtained by the proposed CSFF method on all the three datasets. On the Pavia University scene, the CSFF provides the best classification accuracy in terms of OA, AA, and κ, which signifies a $20\% - 30\%$ drop in prediction failures over the second best ANNC-ASSCC. For the Pavia Centre scene, the CSFF still leads to slight improvements on all the metrics, considering the high classification accuracies achieved by the comparing counterparts. On the Salinas image, it is MASR that leads to the best classification results, second by CSFF. We observe that CSFF yields over 1.5% increase in OA when compared to the ANNC-ASSCC, the value corresponding to around 50% fewer prediction failures. Specifically, noteworthy improvements are achieved by CSFF over the methods PPF and ANNC-ASSCC on addressing two difficult classification tasks, *i.e.,* the Grapes untrained and the Vinyard untrained. Concerning the phenomenon that CSFF is inferior to MASR, one explanation is that the latter utilizes multiscale neighborhoods at the testing stage, without excluding the training pixels [39]. To keep a fair comparison, the influence brought by train-test overlap to classification result should be eliminated. Following [39], supplementary experiments are carried out using a modified version of MASR, termed MASR-t, on all the datasets. At the testing stage, the MASR-t is designed to exclusively deteriorate training information by replacing all the training pixels with a vector of identical elements 0.01. As observed from TABLE 5, MASR-t results to over 2% decreases in terms of both OA and κ, and is outperformed by the proposed CSFF on the Salinas image. To conclude, the proposed CSFF leads to promising classification results compared to the state-of-the-art methods on all the datasets.

The proposed CSFF generates explicit spectral-spatial features that can be classified using different classification algorithms. To examine the effectiveness of the features obtained by the CSFF under various classifiers, experiments are performed by using the default center classifier, kNN, and SVM, on all the datasets. The results are given in TABLES 7–9. We observe that on all the images, the three classifiers yield stable and similar classification accuracies in terms of all the quantitative metrics. It demonstrates that the spectral-spatial features generated by the proposed framework are effective and robust against different classifiers.

4.4 Computational Cost

The testing time of all the comparing methods are reported in Table 10. Experiments are performed on a machine equipped with CPU of Intel Xeon E5-2660@2.6GHz and GPU of NVIDIA TitanX. As for the proposed CSFF, both the total testing time and the time of applying discriminant model are presented. In fact, it is the testing data preparation in discriminant model, namely the generation of pixel-pairs using the centering pixel and each spectrum within the neighborhood, that is the most time-consuming. To alleviate computational burden, a slightly decreased neighborhood size can be adopted in practice. On one hand, this does not deteriorate the classification accuracies too much, as shown in Figure 8. On the other hand, a declined neighborhood size reduces the computational complexity of testing data preparation, which computes $O(n^2)$ with n being the neighborhood size.

Table 5: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of MASR, MASR-t, PPF, 3D-CNN, ANNC-ASSCC and CSFF on Salinas

	MASR	MASR-t	PPF	3D-CNN	ANNC-ASSCC	CSFF
Brocoli_green_weeds_1	100.00 ±0.00	100.00 ±0.00	99.98 ±0.03	100.00 ±0.00	100.00 ±0.00	99.97 ±0.03
Brocoli_green_weeds_2	99.97 ±0.05	99.95 ±0.07	99.58 ±0.09	99.64 ±0.39	100.00 ±0.00	100.00 ±0.00
Fallow	100.00 ±0.00	100.00 ±0.00	99.61 ±0.17	95.95 ±3.76	100.00 ±0.00	100.00 ±0.00
Fallow_rough_plow	99.89 ±0.08	99.66 ±0.18	99.73 ±0.08	100.00 ±0.00	99.53 ±0.41	95.54 ±3.64
Fallow_smooth	99.58 ±0.14	99.56 ±0.10	97.43 ±0.28	99.92 ±0.10	99.67 ±0.17	99.39 ±0.29
Stubble	100.00 ±0.01	99.98 ±0.03	99.66 ±0.16	100.00 ±0.00	100.00 ±0.00	99.93 ±0.05
Celery	99.95 ±0.09	99.97 ±0.02	99.93 ±0.04	99.18 ±0.87	100.00 ±0.00	99.94 ±0.05
Grapes_untrained	97.82 ±0.46	87.72 ±0.84	84.81 ±0.94	92.00 ±2.13	92.34 ±0.69	95.61 ±1.19
Soil_vinyard_develop	99.99 ±0.02	99.99 ±0.01	99.15 ±0.78	98.72 ±0.69	99.99 ±0.01	99.97 ±0.04
Corn_senesced_green_weeds	99.90 ±0.08	99.59 ±0.36	96.73 ±0.46	99.39 ±0.69	99.44 ±0.23	98.98 ±0.15
Lettuce_romaine_4wk	100.00 ±0.00	100.00 ±0.00	99.45 ±0.15	100.00 ±0.00	100.00 ±0.00	99.88 ±0.13
Lettuce_romaine_5wk	99.98 ±0.03	99.96 ±0.05	100.00 ±0.00	100.00 ±0.00	100.00 ±0.00	100.00 ±0.00
Lettuce_romaine_6wk	100.00 ±0.00	99.86 ±0.20	99.50 ±0.07	100.00 ±0.00	99.50 ±0.80	99.09 ±1.08
Lettuce_romaine_7wk	99.90 ±0.24	99.66 ±0.09	99.47 ±0.30	100.00 ±0.00	99.63 ±0.28	98.00 ±1.28
Vinyard_untrained	99.22 ±0.32	96.34 ±0.80	81.80 ±4.30	96.38 ±0.48	90.79 ±2.42	98.14 ±1.41
Vinyard_vertical_trellis	99.99 ±0.02	99.59 ±0.50	98.81 ±0.41	99.50 ±0.99	99.99 ±0.02	99.95 ±0.10
OA(%)	①**99.38 ±0.32**	96.74 ±0.80	93.61 ±0.64	95.91 ±0.87	96.98 ±0.39	②**98.53 ±0.34**
AA(%)	①**99.76 ±0.02**	98.86 ±0.50	97.23 ±0.32	98.79 ±0.29	98.81 ±0.13	②**99.02 ±0.20**
κ	①**0.9930 ±0.0009**	0.9635 ±0.0007	0.9285 ±0.0072	0.9480 ±0.0108	0.9662 ±0.0044	②**0.9835 ±0.0038**

Table 6: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of MASR, MASR-t, PPF, 3D-CNN, ANNC-ASSCC and CSFF on Pavia Centre

	MASR	MASR-t	PPF	3D-CNN	ANNC-ASSCC	
Water	99.87 ± 0.11	99.82 ± 0.14	99.15 ± 0.18	99.93 ± 0.11	100.00 ± 0.00	100.
Trees	94.22 ± 0.45	90.25 ± 0.86	97.96 ± 0.24	98.26 ± 0.31	98.75 ± 0.62	98.
Asphalt	99.45 ± 0.46	97.04 ± 0.65	97.37 ± 0.18	94.98 ± 1.97	99.26 ± 0.16	98.
Self-Blocking Bricks	99.98 ± 0.05	94.91 ± 2.87	99.27 ± 0.11	98.48 ± 2.11	99.96 ± 0.04	99.
Bitumen	98.75 ± 0.51	96.17 ± 1.03	98.79 ± 0.16	99.67 ± 0.18	99.35 ± 0.28	99.
Tiles	80.23 ± 1.78	46.58 ± 4.98	98.95 ± 0.08	99.29 ± 0.41	99.73 ± 0.06	99.
Shadows	99.34 ± 0.47	92.63 ± 1.06	94.36 ± 0.35	97.88 ± 0.97	97.49 ± 0.64	98.
Meadows	99.90 ± 0.04	99.86 ± 0.02	99.90 ± 0.03	99.99 ± 0.01	99.41 ± 0.06	99.
Bare Soil	84.80 ± 1.02	64.36 ± 1.16	99.96 ± 0.05	99.15 ± 0.75	98.72 ± 0.56	99.
OA(%)	98.02 ± 0.12	94.77 ± 0.31	99.03 ± 0.08	99.60 ± 0.07	**②$99.73 \pm 0.05$**	**①99**
AA(%)	95.17 ± 0.20	86.85 ± 0.56	98.41 ± 0.06	98.58 ± 0.47	**②$99.25 \pm 0.09$**	**①99**
κ	0.9719 ± 0.0018	0.9256 ± 0.0043	0.9862 ± 0.0011	0.9845 ± 0.0010	**②$0.9937 \pm 0.0011$**	**①0.99**

Table 7: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of kNN, SVM and center classifier on spectral-spatial feature of Pavia University scene

	kNN (5)	kNN (10)	SVM	C-Classifier
OA(%)	98.92 ± 0.19	98.93 ± 0.23	98.91 ± 0.12	98.90 ± 0.14
AA(%)	98.54 ± 0.23	98.53 ± 0.25	98.54 ± 0.10	98.49 ± 0.13
κ	0.9857 ± 0.0026	0.9858 ± 0.0031	0.9856 ± 0.0015	0.9852 ± 0.0018

5 Conclusion

In this paper, we investigated a novel ANN and CNN based classification framework that properly integrates the spatial information to the spectral-based features, and generates spectral-spatial features suitable for various classifiers. Based on a limited number of labeled pixels and without using any local information, both a spectral feature extraction model and a discriminant model were trained. Using the learned discriminant model, the local structure was extracted and represented as a customized convolutional kernel. The spectral-spatial feature was obtained by a convolutional operation between the kernel and the corresponding spectral features within a neighborhood. Experiments on three real hyperspectral images validated the performance of the proposed method in terms of classification accuracies, when compared to the state-of-the-art algorithms. We also studied the characteristics of the learning features, which showed robustness and stableness against various classifiers. Future works will focus on extending the proposed framework to multiple-model fusion.

References

[1] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

Table 8: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of kNN, SVM and center classifier on spectral-spatial feature of Salinas scene

	kNN (5)	kNN (10)	SVM	C-Classifier
OA(%)	96.80 ± 0.94	97.56 ± 0.77	98.42 ± 0.39	98.53 ± 0.34
AA(%)	98.36 ± 0.52	98.68 ± 0.32	98.92 ± 0.24	99.02 ± 0.20
κ	0.9642 ± 0.0105	0.9727 ± 0.0087	0.9823 ± 0.0044	0.9835 ± 0.0038

Table 9: Classification accuracies in percentage (mean ± standard deviation), averaged over 5 runs, of kNN, SVM and center classifier on spectral-spatial feature of Pavia Centre scene

	kNN (5)	kNN (10)	SVM	C-Classifier
OA(%)	99.77 ± 0.07	99.75 ± 0.08	99.74 ± 0.07	99.75 ± 0.07
AA(%)	99.45 ± 0.13	99.41 ± 0.16	99.40 ± 0.14	99.40 ± 0.13
κ	0.9967 ± 0.0010	0.9965 ± 0.0011	0.9963 ± 0.0010	0.9964 ± 0.0009

Table 10: Comparison of testing time (in seconds)

	MASR	PPF	3D-CNN	ANNC-ASSCC	CSFF (*Discriminant Model*)
Pavia University	2424.09	13.85	50.94	37.70	1779.45 (*1771.02*)
Salinas	10572.88	22.51	45.69	61.09	5453.36 (*5436.76*)
Pavia Centre	8480.59	92.66	164.06	135.87	5924.74 (*5897.63*)

[2] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[3] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 447–451, May 2012.

[4] J. A. Jablonski, T. J. Bihl, and K. W. Bauer, "Principal component reconstruction error for hyperspectral anomaly detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 8, pp. 1725–1729, 2015.

[5] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[6] R. J. Johnson, J. P. Williams, and K. W. Bauer, "Autogad: An improved ica-based hyperspectral anomaly detection algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3492–3503, 2013.

[7] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[8] 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, Aug. 2004.

[9] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 55–66, Jan. 2014.

[10] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[11] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1139–1155, Apr. 2009.

[12] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT press, 2002.

[13] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[14] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[15] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svms and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[16] L. Z. Huo and P. Tang, "Spectral and spatial classification of hyperspectral data using svms and gabor textures," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2011, pp. 1708–1711.

[17] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 844–856, Feb. 2013.

[18] A. Plaza, J. Plaza, and G. Martin, "Incorporation of spatial constraints into spectral mixture analysis of remotely sensed hyperspectral data," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, Sep. 2009, pp. 1–6.

[19] B. Song, J. Li, M. D. Mura, P. Li, A. Plaza, J. M. Bioucas-Dias, J. A. Benediktsson, and J. Chanussot, "Remotely sensed image classification using sparse representations of morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 5122–5136, Aug. 2014.

[20] R. Hang, Q. Liu, Y. Sun, X. Yuan, H. Pei, J. Plaza, and A. Plaza, "Robust matrix discriminative analysis for feature extraction from hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 2002–2011, 2017.

[21] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.

[22] T. Hill, L. Marquez, M. O'Connor, and W. Remus, "Artificial neural network models for forecasting and decision making," *International Journal of Forecasting*, vol. 10, no. 1, pp. 5–15, 1994.

[23] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, Nov 2000.

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

[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, no. Supplement C, pp. 85–117, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608014002135

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[28] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[29] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3660–3671, June 2016.

[30] Y. Chen, X. Zhao, and X. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[31] V. Slavkovikj, S. Verstockt, W. De Neve, S. Van Hoecke, and R. Van de Walle, "Hyperspectral image classification with convolutional neural networks," in *Proceedings of the ACM international conference on Multimedia*. ACM, 2015, pp. 1159–1162.

[32] Y. Chen, H. Jiang, C. Li, X. 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.

[33] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sensing*, vol. 8, no. 2, p. 99, 2016.

[34] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[35] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[36] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.

[37] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 10, pp. 5585–5599, Oct. 2017.

[38] K. Nogueira, O. A. B. Penatti, and J. A. D. Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[39] A. J. X. Guo and F. Zhu, "Spectral-spatial feature extraction and classification by ann supervised with center loss in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1755–1767, Mar. 2019.

[40] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, Apr. 2017.

[41] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification," *Remote Sensing*, 2017.

[42] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[43] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[46] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[47] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[48] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[49] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.