arXiv:1711.07141v1 [cs.CV] 20 Nov 2017

Spectral-Spatial Feature Extraction and Classification by ANN Supervised with Center Loss in Hyperspectral Imagery

Alan J.X. Guo and Fei Zhu *

August 15, 2018

Abstract

In this paper, we propose a spectral-spatial feature extraction and classification framework based on artificial neuron network (ANN) in the context of hyperspectral imagery. With limited labeled samples, only spectral information is exploited for training and spatial context is integrated posteriorly at the testing stage. Taking advantage of recent advances in face recognition, a joint supervision symbol that combines softmax loss and center loss is adopted to train the proposed network, by which intra-class features are gathered while inter-class variations are enlarged. Based on the learned architecture, the extracted spectrum-based features are classified by a center classifier. Moreover, to fuse the spectral and spatial information, an adaptive spectral-spatial center classifier is developed, where multiscale neighborhoods are considered simultaneously, and the final label is determined using an adaptive voting strategy. Finally, experimental results on three well-known datasets validate the effectiveness of the proposed methods compared with the state-of-the-art approaches.

1 Introduction

Recent advances in remote sensing technology lead to improved imaging quality in terms of both spectral and spatial resolutions for hyperspectral imagery [1]. Typically, a hyperspectral image is collected by remote sensors with hundreds of channels over a wavelength range. It is regarded as a data cube consisting of continuous images, each acquired by a certain channel over the same land-cover. Consequently, every pixel corresponds to a reflectance spectrum. In hyperspectral imagery analysis, an essential task is the differentiation of target materials by identifying the class of each pixel.

To address the hyperspectral classification problem, traditional feature extraction (FE) and classification strategies have been extensively investigated, seeking to extract shallow representations for input spectra. Extracting features with linear transformations permits representing the original data in a subspace with reduced-dimension. This category of methods includes principal component analysis (PCA) [2], independent component analysis (ICA) [3] and linear discriminant analysis (LDA) [4], to name a few. Since the nonlinearities often exist in hyperspectral images, various nonlinear methods have been proposed accordingly, such as support vector machine (SVM) [5], manifold learning [6], and the kernel-based algorithms [7, 8].

Most of the aforementioned methods are spectrum-based ones, *i.e.,* to extract the features directly from the pixel-wise spectra while ignoring the intrinsic geographical structure in specific data. In fact, the spatially adjacent pixels tend to be inherently correlated. A joint usage of the spectral and spatial information could improve the representation ability of the extracted features [1]. To this end, numbers of spectral-spacial methods have been derived and have outperformed the spectral ones in terms of classification accuracy [9–14]. In [15], the multiscale adaptive sparse representation (MASR) method exploits the spatial information derived from neighborhoods of varying sizes by an adaptive sparse strategy. Despite promising classification results achieved on several datasets, MASR produces high computational cost, since multiscale windows around a testing pixel are constructed, in which every pixel needs to be sparsely coded using a structural dictionary consisting of training samples. To effectively use the spectral-spatial information remains a challenging problem. Meanwhile, the greatly improved resolutions in both spectral and spatial domains may enlarge the intra-class variation (*e.g.*, roofs with shadows) and decrease the interclass variation (*e.g.*, roads and roofs have similar spectra), thus greatly affecting the classification accuracy [16].

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

Deep learning has emerged as a powerful feature extraction technology and gained great success in many machine learning tasks. Compared with traditional methods, deep models are capable to extract more abstract and complex features with better representation ability. The features are generated in a hierarchical manner, namely the high-level representations are successfully built from a cascade of lower layers consisting of nonlinear processing units. However, to train a deep network with large number of unknown parameters, a large-scale training set is required. Otherwise, the network is prone to overfitting. This becomes a major obstacle to exploiting deep learning in hyperspectral imagery, where the number of labeled samples are quite limited due to high expense of manually labeling and even the available labels are not always reliable [17]. To extract deep spectral-spatial representation, a conventional strategy is to train a network or a classifier based on patch-based samples, as did in [16–20]. However, this will further aggregate the sample shortage problem, if the overlap of training and testing samples is not allowed.

To alleviate this dilemma with deep models, small networks with respectively less parameters have been advocated, including the earlier stacked autoencoder (SAE) [21, 22], the deep belief network (DBN) [23], and a few models based on the convolutional neural networks (CNN). In [24], an unsupervised CNN has been proposed to obtain sparse representations, mainly by using the greedy layerwise unsupervised pre-training along with the efficient enforcing of population and lifetime sparsity. Three supervised CNN architectures, referred by 1-D, 2-D and 3-D CNNs, were investigated in [18]. More specifically, 3-D CNN could jointly extract the spectral-spatial integrated features by working on data cube composed by a neighborhood of spectra. The authors enlarged the training set by virtual sample enhancement, and avoided the overfitting issue by a few techniques, such as dropout and ℓ_2-norm regularization. Similar strategies were also adopted in a novel CNN architecture proposed in [25]. In [16], a spectral-spatial feature based classification framework was proposed. To address the increasing intra-class variation and interclass confusion issues in spectral domain, the spectral feature in low-dimensionality was firstly extracted by the so-called BLDE method, aiming to increase the interclass variation while keeping the intra-class samples stay close. The deep spatial-based feature was extracted by CNN. Finally, the fusion feature, stacked by extracted spectral and spatial features, was used to train the classifier.

Taking advantage of pre-training and fine-tuning techniques, relative deep networks have also been investigated in the literature. The pre-training on large-scale data sets other than target data is advisable, since the initial layers of deep network tend to be generic filters, such as edge or color blob detector, which are independent from specific data and transferable for different recognition tasks [17]. Studies in [26] compared three possible practices to exploit existing CNNs, namely full-training, fine-tuning, and using CNNs as feature extractor, finding out that fine-tuning tended to be the best strategy. In [17], a deep multiscale spatial-spectral feature extraction algorithm is proposed to learn effective discriminant features of hyperspectral images. The VGG-verydeep-16 net is firstly pre-trained on the natural image data sets, and then used to extract the spatial structural feature. The fusion vector stacked by the spatial and spectral features is fed to the classifier.

To explore spectral-spatial information, the routine by training with patch-based pixels might easily cause the overlap between training and testing sets. An alternative is postprocessing of the extracted features at the testing stage. Authors in [27] proposed a deep pixel-pair feature (PPF) classification framework based on a well-designed CNN. Pixel-pairs are generated by taking Cartesian product and used to train a deep model. For a testing sample, paired samples are generated by combining the centering pixel with its neighbors and classified using the learned network. The final label is predicted by a voting strategy. The idea of postprocessing will also be adopted in this paper, namely spatial context is only exploited at the testing stage. Therefore, PPF and the aforementioned MASR are chosen for comparison in Section 4, since these two state-of-the-art methods are trained only based on spectral samples.

Raised in the field of face recognition, deep metric learning is becoming a hotspot in deep learning and has gained great success. The basic idea is to enhance the discriminative power of the deeply learned features by simultaneously enlarging the inter-class variations and decreasing the intra-class variations. To this end, several metric learning loss functions have been introduced to CNN as supervision signal, such as triplet loss [28], magnet loss [29], and newly introduced center loss [30] that will be investigated in this paper. To the best of our knowledge, to design a neuron network based on deep metric learning has not been approached so far in the context of hypersepctral FE and classification.

In this paper, we propose a deep spectral-spatial FE and classification framework that allows to extract discriminative spectral-spatial features with intra-class compactness and interclass dispersity. We apply a joint supervision symbol proposed in [30] to train the network, in which the conventional softmax loss and the so-called center loss is combined. The spectral pixels are directly fed to the network to estimate the network parameters and the class centers as well. Based on the learned network, the extracted features are classified by a center classifier. To further improve the classification performance, we propose to adaptively integrate the spatial information from multi-scale neighborhoods at the testing stage. For each scale, the label and its associated weight are estimated using center classifier, and the final label is determined by a voting strategy.

The proposed framework has following characteristics:

1. *Discriminative features*: Taking advantage of center loss, the proposed ANNC allows to extract spectral features that are not only separable but also discriminative. This helps to decrease the undesirable side effects brought by the high-resolutions to the classification task, namely the increasing intra-class variation and inter-class similarity. The extracted features fit the proposed center classifier well. Compared with raw spectra, the discriminative features are easier to handle, facilitating an effective integration of spatial information at the testing stage.

2. *Less overfitting issues*: Facing the limited number of training samples in hyperspectral images, we design a small network with less parameters in order to alleviate the overfitting issue. To make full use of training set, virtual samples are generated. Moreover, only spectral information is exploited at the training stage to avoid overlap of training and testing sets. As indicated in [30], the joint supervision which complicates the training task also helps to alleviate the overfitting problem to some degree.

3. *Training with less prior knowledge on the image*: In the proposed framework, the training procedure is strictly spectrum-based, without requiring any local information on the training pixel, such as the coordinate and the (unlabeled) neighboring pixels that may confront the testing set. This offers a more flexible model for real world applications with less conditions to satisfy.

The remainder of this paper is structured as follows. Section 2 succinctly describes the basis of artificial neural network. In Section 3, we propose a deep spectral-spatial FE and classification framework. Experimental results and analysis on three real hyperspectral images are presented in 4. Finally, Section 5 provides some conclusion remarks.

2 Basic concepts of ANN

Artificial neural network (ANN) is inspired by the organization and functioning of biological neurons [31, 32]. Typically, it is viewed as a trainable multilayer structure with multiple feature extraction steps, and is composed by an input layer of neurons, several hidden layers of neurons, and a final layer of output neurons, as shown in Figure 1.

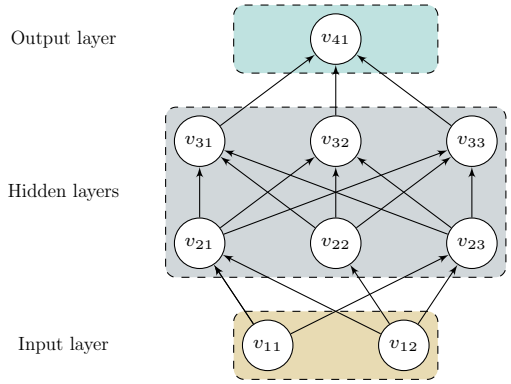


Figure 1: An example of ANN with 2 hidden layers.

Given an ANN architecture, let v_{ij} denote the j-th neuron in the i-th layer. This value is computed based on the neurons in previous layer, and takes the form

$$v_{ij} = \Phi\left(b_{ij} + \sum_{k=1}^{n_{i-1}} w^{ij}_{(i-1)k} v_{(i-1)k}\right), \tag{1}$$

where n_{i-1} denotes the number of neurons in the $(i-1)$-th layer, $w^{ij}_{(i-1)k}$ represents the weight connecting neurons v_{ij} and $v_{(i-1)k}$, and b_{ij} indexes the bias for the neuron v_{ij}. The pointwise activation function $\Phi(\cdot)$ is often applied to introduce nonlinearity to the neutron network. In this paper, the commonly-used ReLU function is chosen, which is defined by

$$\Phi(x) = \max(0, x).$$

The loss function measures the global error between the ANN output and the groundtruth. To train a network, namely to determine the parameters of weight and bias, one seeks to minimize the loss function defined on the given training set consisting of labeled samples in pair. The resulting optimization problem is addressed by the celebrated gradient-based, error back-propagation algorithm.

To avoid gradient computation on the whole training set, the stochastic gradient descent (SGD) with mini-batch strategy is often applied, which alleviates the computational burden by processing a mini-batch at each iteration of training.

The main objective of dropout is to regularize the neuron network in order to prevent overfitting [33]. During the training step, the weights associated to some units are excluded from being updated temporarily, and these units are randomly chosen with a fixed probability, namely dropout ratio.

3 Proposed spectral-spatial feature extraction and classification framework

This section firstly presents the proposed neuron network, followed by explaining the joint supervision symbol used for training. Based on the learned architecture, a spectral center classifier is proposed at the testing stage. Finally, a spectral-spatial FE and classification framework termed ANNC-ASSCC is presented.

3.1 Proposed ANN structure

We first briefly describe the proposed ANN architecture. As illustrated in Figure 2, the proposed network consists of an input layer, three hidden layers, and an output layer followed by a softmax classifier, and all the layers are fully connected. We call the i-th layer fully connected, if every neuron in it has connections to all activations in the previous layer, namely all the parameters $w^{ij}_{(i-1)k}$ in (1) are set learnable.

The sizes of the input and output layers are determined according to the hyperspectral image under study: the former is set as the size of each sample pixel L, and the later is chosen to be the number of classes K. The numbers of neurons in Layer $1, 2$ and 3 are empirically set to be $512, 256$ and 32.

Layer 1 and 2 consist of ReLU as the activation function, while Layer 3 does not transformed with any activation function. The output layer is activated by the softmax function in order to perform the classification task based on the learned features, namely the outputs of Layer 3.

It is noteworthy that the pixel-wise spectra are directly fed into the proposed network at the training stage, differing from most existing spatial-spectral deep learning models which are trained using patch-based samples, each being a neighborhood area centering at every labeled pixel [18–20].

3.2 Joint supervision of softmax loss and center loss

Inspired by a discriminative feature learning strategy proposed for deep face recognition in [30], we propose to train the aforementioned network by simultaneously supervising the conventional softmax loss and the newly-introduced center loss.

Suppose that the proposed network is optimized by the SGD with mini-batch strategy at the training stage, namely the network parameters are updated based on a small set of samples at each iteration. Let $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_M, y_M)\}$ be a given mini-batch composed by M labeled samples, where $\boldsymbol{x}_i \in \mathbb{R}^{L\times 1}$ denotes the i-th input spectra in the batch, and $y_i \in \{1, ..., K\}$ is the corresponding ground-truth label.

The softmax loss combines softmax function with multinomial logistic loss. Softmax function is commonly used in multi-class classification, as its output could be regarded as a probability distribution over predefined classes. Let $\sigma(\boldsymbol{z}) = [\sigma_1(\boldsymbol{z}), \sigma_2(\boldsymbol{z}), \ldots, \sigma_K(\boldsymbol{z})]^\top$ represent the softmax function over the K-dimensional output feature $\boldsymbol{z} = [z_1, z_2, \ldots, z_K]^\top$. The predicted probability that sample $\boldsymbol{x}_i$ belongs to class j is given by

$$p(\hat{y}_i = j|\boldsymbol{x}_i) = \sigma_j(\boldsymbol{z}) = \frac{\exp(z_j)}{\sum_{l=1}^{K}\exp(z_l)}, \tag{2}$$

where $\hat{y}_i$ is the predicted label of sample $\boldsymbol{x}_i$. Multinomial logistic loss generalizes the logistic loss from binary classification to multi-class case, and is formulated as

$$\mathcal{L}_S = -\frac{1}{M}\left[\sum_{i=1}^{M}\sum_{j=1}^{K}\mathbb{I}_{\{j\}}(y_i)\log p(\hat{y}_i = j|\boldsymbol{x}_i)\right], \tag{3}$$

where $\mathbb{I}_{\mathcal{A}}(x)$ is the indicator function of the set $\mathcal{A}$ defined by

$$\mathbb{I}_{\mathcal{A}}(x) = \begin{cases} 1 & \forall x \in \mathcal{A}; \\ 0 & \text{otherwise.} \end{cases}$$

By integrating the expression (2) to (3), we get the expression of the softmax loss.

In [30], the center loss was firstly introduced to address the face recognition task, and has achieved state-of-the-art performance on several benchmarks. Intuitively, it is designed to enhance the separability of the deeply learned features, by minimizing the intra-class distances while encouraging the inter-class variations. To this end, the center loss is defined by averaging the distances between every extracted feature and the corresponding class center, namely

$$\mathcal{L}_C = \frac{1}{2M} \sum_{i=1}^{M} ||\hat{\boldsymbol{x}}_i - \boldsymbol{c}_{y_i}||_2^2. \tag{4}$$

Here, the learned feature $\hat{\boldsymbol{x}}_i$ corresponds to the i-th input spectra in the batch, for $i = 1, ..., M$, and $\boldsymbol{c}_k$ denotes the k-th class center defined by the averaging over the features in the k-th class, for $k = 1, ..., K$. In practice, the class centers are updated with respect to mini-batch instead of being updated based on the entire training set. At the t-th training round, let $\boldsymbol{c}_k^t$ denote the k-th estimated class center, and let $\bar{\boldsymbol{c}}_k^t$ denote the averaged features of the i-th class based on mini-batch. Following [30], the class centers are iteratively updated at every training round, and are formulated as

$$\boldsymbol{c}_k^{t+1} = \boldsymbol{c}_k^t + \alpha(\bar{\boldsymbol{c}}_k^{t+1} - \boldsymbol{c}_k^t), \tag{5}$$

with $\boldsymbol{c}_k^1 = \bar{\boldsymbol{c}}_k^1$, for $k = 1, ..., K$. Here, a hyperparameter α is introduced to control the learning rate of the centers, thus avoiding large perturbations. In this paper, we set $\alpha = 0.5$.

The loss function of the proposed architecture is then defined by combining the softmax loss $\mathcal{L}_S$ and the center loss $\mathcal{L}_C$, with

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C, \tag{6}$$

where hyperparameter λ balances the importance between two loss functions, and is fixed to be 0.01 in this paper.

We apply the algorithm given in [30] to train the proposed network. See Algorithm 1 and find more details in the related paper.

Algorithm 1 Training algorithm [30]

Input: The number of iteration $t \leftarrow 0$. Mini-batches $\{(\boldsymbol{x}_1^t, y_1^t), \ldots, (\boldsymbol{x}_M^t, y_M^t)\}$ for training. Initialized parameters θ in the network . Hyperparameters λ, α and learning rate μ^t.

Output: The trained network parameters θ.

1: Initialize the parameters θ by Gaussian distribution $N(0, 0.01^2)$ (weight parameters) and constant number 0 (bias parameters).
2: Compute the joint loss by $\mathcal{L}^0 = \mathcal{L}_S^0 + \lambda \mathcal{L}_C^0$.
3: Compute the backpropagation error $\frac{\partial \mathcal{L}^0}{\partial \boldsymbol{x}_i^0}$ for each i by $\frac{\partial \mathcal{L}^0}{\partial \boldsymbol{x}_i^0} = \frac{\partial \mathcal{L}_S^0}{\partial \boldsymbol{x}_i^0} + \lambda \frac{\partial \mathcal{L}_C^0}{\partial \boldsymbol{x}_i^0}$.
4: Update the parameters θ by $\theta^0 = \theta - \mu^0 \sum_i \frac{\partial \mathcal{L}^0}{\partial \boldsymbol{x}_i^0} \frac{\partial \boldsymbol{x}_i^0}{\partial \theta}$.
5: Update the estimated centers by $\boldsymbol{c}_k^0 = \boldsymbol{c}_k'^0$, for $1 \le k \le K$.
6: **while** not converge **do**
7: $t \leftarrow t + 1$.
8: Compute the joint loss $\mathcal{L}^t = \mathcal{L}_S^t + \lambda \mathcal{L}_C^t$
9: Compute the backpropagation error $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t}$ for each i, by $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} = \frac{\partial \mathcal{L}_S^t}{\partial \boldsymbol{x}_i^t} + \lambda \frac{\partial \mathcal{L}_C^t}{\partial \boldsymbol{x}_i^t}$.
10: Update the parameters θ by $\theta^t = \theta^{t-1} - \mu^t \sum_i \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} \frac{\partial \boldsymbol{x}_i^t}{\partial \theta^{t-1}}$.
11: Update the estimated centers $\boldsymbol{c}_k^t = \boldsymbol{c}_k^{t-1} + \alpha(\boldsymbol{c}_k'^t - \boldsymbol{c}_k^{t-1})$, for $1 \le k \le K$.
12: **end while**

3.3 Discriminative spectral feature extraction and center classifier

At the testing stage, the spectral features are firstly generated as the outputs of Layer 3, using the learned network. Taking advantage of center loss, the obtained features have good discriminative ability for subsequent classification task, namely the features are distributed compactly within class while different classes stay separable.

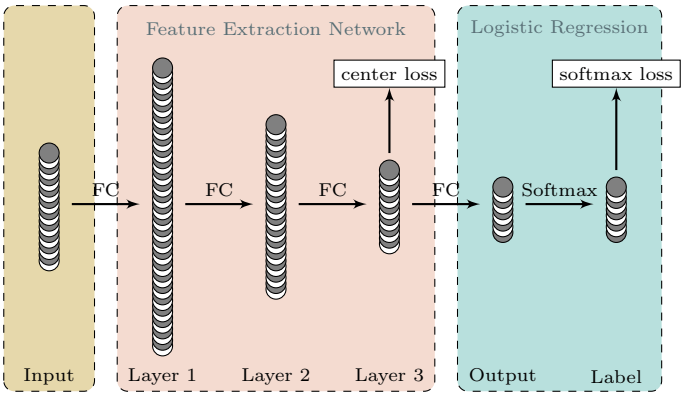


Figure 2: Structure of the proposed neural network.

To perform classification based on the extracted features, we introduce the so-called center classifier, which assigns label to a given feature according to its nearest class center. As the real class centers are not accessible, they are approximated by averaging the learned features for training samples from each class. Let $\widehat{\boldsymbol{c}}_k$ denote the k-th approximate class center, it is expressed as

$$\hat{\boldsymbol{c}}_k = \frac{\sum_{\boldsymbol{x}\in\{\boldsymbol{x}_i|y_i=k\}} f(\boldsymbol{x})}{\text{card}\{\boldsymbol{x}_i|y_i=k\}}, \tag{7}$$

for $1 \leq k \leq K$, where $\{(\boldsymbol{x}_i, y_i)\}$ represents the entire training set composed by pairs of spectrum and associated groundtruth label, and $f(\cdot)$ returns to the learned feature (the outputs of Layer 3) for any entry. For a testing sample $\boldsymbol{x}$, it is classified to have label k_0 such that

$$k_0 = \arg\min_k \{d(f(\boldsymbol{x}), \hat{\boldsymbol{c}}_k)\}, \tag{8}$$

where $d(\cdot,\cdot)$ measures the Euclidean distance between two vectors.

The above testing stage is combined with the training stage presented in Section 3.2, and the whole framework is termed artificial neuron network with center loss and spectral center classifier (ANNC-SCC). For comparison, the case without center loss is abbreviated by ANN-SCC.

3.4 Spectral-spatial feature generation and classification

Existing works in [16–20] seek the spectral-spatial representation by training a model with patch-based samples. Although effective, this may further aggregate the sample shortage problem in hyperspectral classification. Thanks to center loss, the spectral features extracted using the proposed framework have good enough discriminative ability that enables an effective integration of spatial information posteriorly at the testing stage.

1) *Spectral-Spatial Center Classifier*: We propose a spectral-spatial center classifier by applying center classifier to the averaged spectral feature, which is calculated within a neighborhood around a testing sample. Firstly, the spectral features, *i.e.*, the outputs of Layer 3, of the whole image are extracted and stored. Given a testing sample $\boldsymbol{x}$, let $N_{i\times i}(\boldsymbol{x})$ denote a local neighborhood of samples centering at $\boldsymbol{x}$, where $i \times i$ represents the pre-defined neighborhood scale with i being an odd number. The average feature of the neighborhood, represented by $\tilde{f}(\boldsymbol{x})$ is computed and taken as the spectral-spatial feature for the testing sample $\boldsymbol{x}$, as summarized in Algorithm 2. Here, all the training samples are excluded and not taken into account. The new feature is fed to the aforementioned center classifier. The method is termed artificial neuron network with center-loss and spectral-spatial center classifier (ANNC-SSCC).

2) *Adaptive Spectral-Spatial Center Classifier*: In ANNC-SSCC, the neighborhood scale greatly influences the classification performance and should be appropriately set according to the feature distribution and local contextual property. However, to obtain such information is not practical, thus posing an obstacle to the estimation of neighborhood scale. To overcome this difficulty, we improve ANNC-SSCC by adopting an adaptive-weighted multiscale voting strategy, where neighborhoods of varying scales are considered simultaneously.

For a testing sample $\boldsymbol{x}$, let $\{\tilde{f}_{i\times i}(\boldsymbol{x}), \ell_{i\times i}(\boldsymbol{x})\}$ denote the pair of spectral-spatial feature and predicted label computed by ANNC-SSCC, and let $P = \text{card}(\{\ell_{i\times i}(\boldsymbol{x})\})$ be the number of candidate labels. Eight scales of neighborhoods, *i.e.*, $3\times 3, 5\times 5, ..., 17\times 17$ are taken into account. Considering that smaller distance between

Algorithm 2 Spatial-spectral feature generation

Input: The testing sample x. The neighborhood of samples $N_{i\times i}(\boldsymbol{x})$ centering at $\boldsymbol{x}$ (with $\boldsymbol{x}$ included). The training set T.

Output: The spectral-spatial feature $\tilde{f}(\boldsymbol{x})$

Initialize the summation variable featureSum = 0 and counter cnt = 0.
for all $\boldsymbol{z}$ in neighborhood $N_{i\times i}(x)$ **do**
  if z not in training set T **then**
    featureSum = featureSum + $f(\boldsymbol{z})$; cnt = cnt + 1.
  end if
end for
return $\tilde{f}(\boldsymbol{x}) = \frac{\text{featureSum}}{\text{cnt}}$

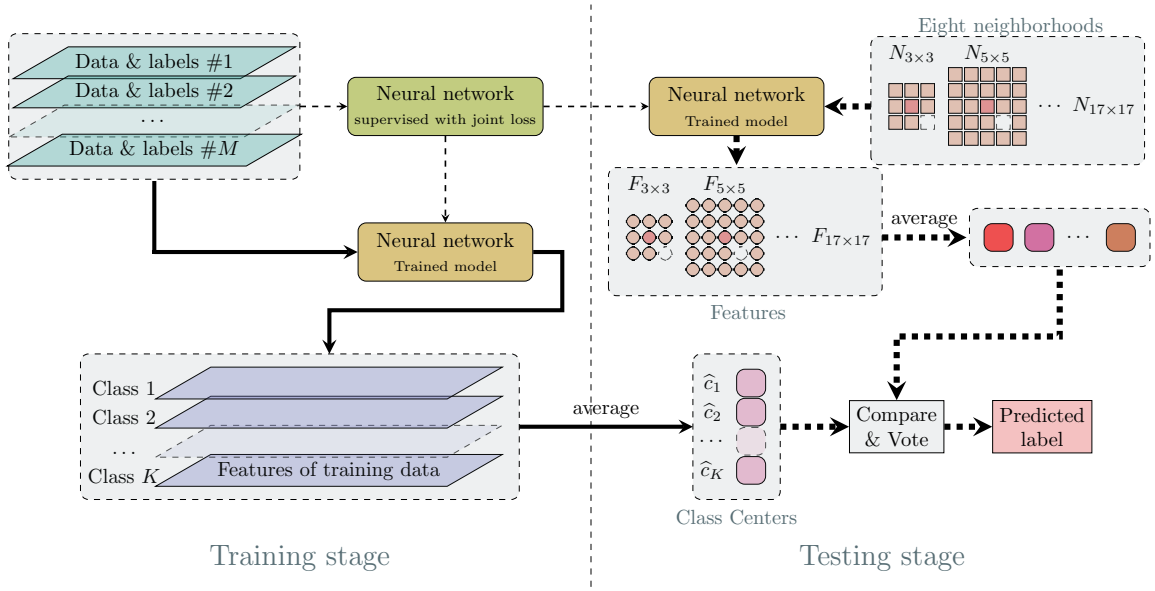


Figure 3: Flowchart of the proposed ANNC-ASSCC framework. Training and testing stages are distinguished by thin and thick dashed arrows, respectively. The estimation of class centers with trained model is drawn with solid thick arrows. At the testing stage, the dashed rectangle in white represents the excluded training sample, appearing within the neighborhoods of a testing sample in red.

style.where three arrow types are used to distinguish different procedures.

a feature to its predicted class center means more reliable label prediction, the weight of neighborhood $N_{i\times i}$ is estimated by

$$w_{i\times i(\boldsymbol{x})} = \frac{1}{d(\tilde{f}_{i\times i}(\boldsymbol{x}), \hat{\mathbf{c}}_{\ell_{i\times i}(\boldsymbol{x})})}, \tag{9}$$

where $d(\cdot,\cdot)$ is Euclidean distance between the feature and its predicted class center, for $i = 3, 5, .., 17$. After gathering the neighborhoods with the same label, the weight for candidate label p is expressed by

$$W_p(\boldsymbol{x}) = \sum_{\{i|\ell_{i\times i}(\boldsymbol{x})=p\}} w_{i\times i}, \tag{10}$$

for $p = 1, ..., P$. The sample $\boldsymbol{x}$ is predicted to have label p_0, such that

$$p_0 = \arg\max_p \{W_p(\boldsymbol{x})\}. \tag{11}$$

The whole training and testing procedure is referred as artificial neuron network with center-loss and adaptive spatial-spectral center classifier (ANNC-ASSCC), and the flowchart is depicted in Figure 3.

4 Experimental results and analysis

4.1 Data description and quantitative metrics

Three well known hyperspectral images are used in experiments[1]. The first one is Pavia Centre scene acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over Pavia, northern Italy. After removing a strip without information, a sub-image of size 1096×715 is taken, with a spatial resolution of 1.3 meters. After discarding the noisy bands, $L = 102$ out of 115 spectral bands are utilized for analysis, covering a wavelength range from 0.43 to $0.86\mu m$. The groundtruth information consists of $K = 9$ classes, as given in Table 1. The false color composite and the groundtruth map of Pavia Center are shown in Figure 4.

Table 1: 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

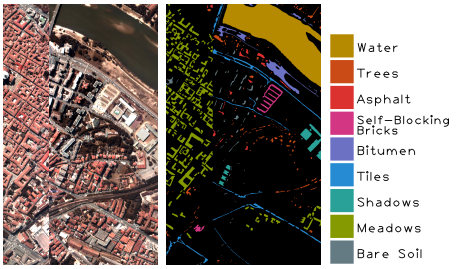


Figure 4: The false color composite and groundtruth map of Pavia Centre

The second image is Pavia University scene that is also returned by the ROSIS sensor. This data comprises 610×340 pixels with $L = 103$ relatively clean spectral bands of interest. As detailed in Table 2, the available groundtruth information consists of $K = 9$ classes. The false color composite and the groundtruth map are illustrated in Figure 5.

The last image under study is the Salinas scene, collected by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) over Salinas Valley, California. This image contains 512×217 pixels, characterized by a spatial resolution of 3.7 meters. After removing the water absorption bands, $L = 204$ out of 224 bands are retained for analysis, with a wavelength range of $0.4 - 2.5\mu m$. There are $K = 16$ classes of land-cover in the groundtruth information, as reported in Table 3. The false color composite and the groundtruth map are shown in Figure 6.

Three metrics, *i.e.*, overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ) are adopted to quantitatively evaluate the classification performance. OA represents the general percentage of the testing samples that are correctly classified, AA computes the average percentage of correctly classified testing samples for each class, and κ measures the percentage of classified samples adjusted by the number of agreements that would be expected by chance alone.

4.2 Experimental setting

1) *Training and Testing Sets:* Before subsequent training and testing procedures, each raw image is firstly normalized to have zero mean and unit variance. To alleviate overfitting caused by limited training samples,

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

Table 2: 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

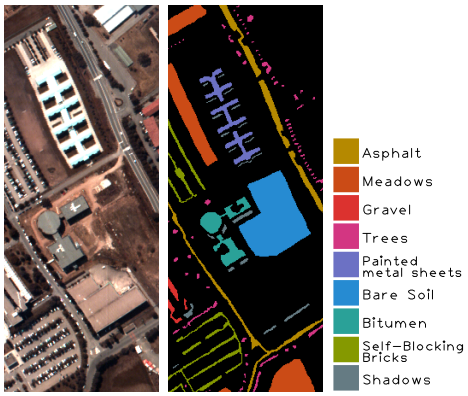


Figure 5: The false color composite and groundtruth representation of Pavia University

Table 3: 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

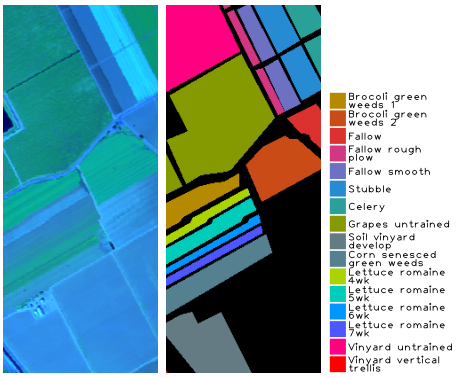
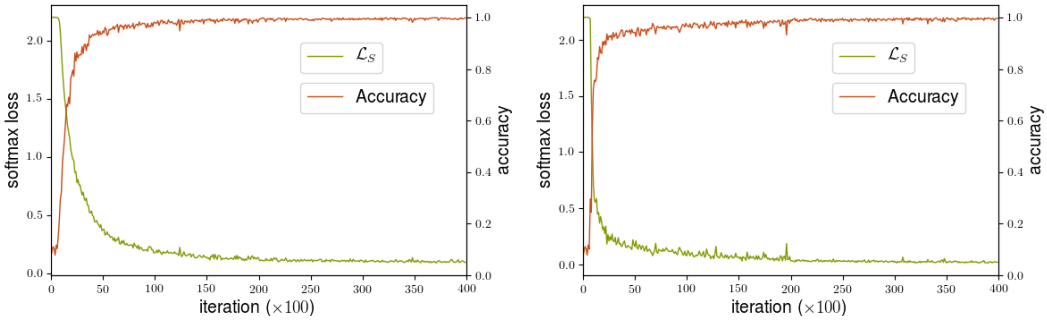


Figure 6: The false color composite and groundtruth representation of Salinas



(a) ANNC (b) ANN

Figure 7: Comparison of (a) ANNC and (b) ANN in terms of softmax loss $\mathcal{L}_S$ and classification accuracy along with iterations (×100).

we enlarge the training set by virtual samples generation in a similar way as did in [18]. Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two samples from the same class, a virtual sample y with the same label is generated by $\boldsymbol{y} = q\boldsymbol{x}_1 + (1-q)\boldsymbol{x}_2$, where the range of q is empirically set as $[-1, 2]$. In the following, the training set of each class consists of 200 real samples enhanced by 80000 virtual ones. All the remaining samples with groundtruth information are used for testing.

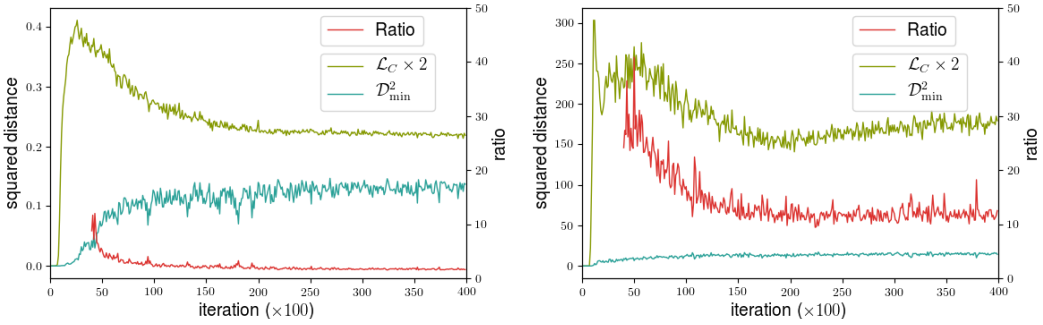
2) *Network Configuration:* The same network structure, as illustrated in Figure 2, is utilized on each of the datasets, except for various sizes of input and output layers that are determined case by case. The mini-batch size is 512, and the learning rate in SGD is initialized as 0.01, and decays every 20000 mini-batches with a multiplier $\sqrt{0.1} \approx 0.3162$. Dropout strategy is adopted in Layer 3 with the dropout ratio to be 0.3. The proposed framework is built on the open source deep learning framework Caffe [34], and each experiment is repeated 5 times.

4.3 Investigation of spectrum-based FE and classification

1) *Enhancing Discriminative Ability with Center Loss:* We analyze how does center loss enhance the discriminative ability on spectral feature extraction. To this end, experiments are performed on Pavia Center, using two neuron networks supervised by conventional softmax loss (ANN) and by a joint loss combining softmax loss and center loss (ANNC), respectively. For fair comparison, all the other conditions are identically set in both cases, such as network structure, training set, hyperparameters, *etc.*

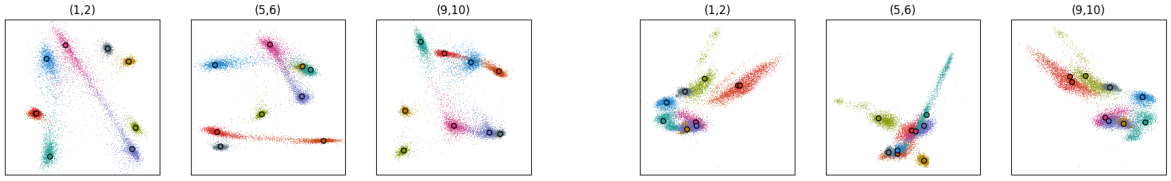
The behaviours of softmax loss, *i.e.* $\mathcal{L}_S$ in (3), and classification accuracy of models ANN and ANNC in training stage are illustrated in Figure 7. We observe that training with center loss leads to more stable changes in terms of both softmax loss and classification accuracy along with iterations, when compared to its counterpart without center loss.

We investigate how the distribution of features changes in Euclidean space at the training stage. To this end, three statistics are monitored, as shown in Figure 8. The center loss, *i.e.* $\mathcal{L}_C$ in (4), measures the averaged intra-class distance of features among classes, with a smaller value reflecting better intra-class compactness. As a measurement of inter-class variation, the minimum squared distance between any two centers are traced,



(a) ANNC (b) ANNC

Figure 8: Comparison of (a) ANNC and (b) ANN in terms of center loss $2 \times \mathcal{L}_C$, $\mathcal{D}^2_{min}$ and the ratio between them, along with iterations (×100).



(a) Features under ANNC (b) Features under ANN

Figure 9: The distributions of spectral features on training samples, using networks ANNC and ANN. Features from various classes are denoted in different colors.

denoted by $\mathcal{D}^2_{\min}$. The ratio between these two statistics are also considered, with a smaller magnitude signifying better discriminative ability.

We observe that training with center loss yields smaller $\mathcal{L}_C$ and greater $\mathcal{D}^2_{\min}$, with the ratio between them several times smaller, when compared to its ANN counterpart. This demonstrates that the learned features under ANNC are more gathered within class while different classes are more distant. Moreover, the curves of ANNC are smoother, showing more steady convergence property over iterations.

Figure 9 visualises the spatial distribution of spectral features extracted by ANN and ANNC, on 2000 random testing samples. For illustration purpose, the 32-dimension features are projected to certain 2 coordinates (i, j). As observed, the features extracted by ANNC are gathered more closely within class while the distances between different classes are more enlarged, compared to the case ANN. The above discussions validate that training with center loss helps to improve the discriminative property of the network that is otherwise trained by trivial softmax loss.

2) *Classification with Spectral Features:* To demonstrate the effectiveness of the discriminative spectral features for subsequent classification, comparative experiments are performed on spectral features that are extracted with ANN and ANNC, respectively. Two classifiers, namely the aforementioned center classifier (CC) and the logistic regression (LR) are considered. The results on datasets Pavia Center, Pavia University and Salinas are presented in Tables 4-6.

Compared with ANN, the spectral features extracted by ANNC generally leads to better classification accuracy in terms of AA, OA and κ, using either classifier on all the datasets. It demonstrates that the discriminative ability induced by center loss has a positive influence on classification performance. To be precise, ANNC combined with CC yields best classification accuracy on Pavia University and Salinas. Regarding Pavia Center, it is ANNC combined with LR that archives the best performance, second by ANNC classified with CC. The combination of ANN and CC leads to the worst results on all the datasets, since features trained with ANN are lack of intra class compactness and inter class dispersion, thus not suitable for CC.

It is noteworthy that ANNC is effective in addressing the confusion between similar classes. Concerning Salinas, while most classes are well recognized with high classification accuracy achieved by both ANN and ANNC, there are two classes, namely Grapes untrained and Vinyard untrained, that are locally adjacent and hard to be distinguished one from the other. By examining the predictions and groundtruth labels, one could find that most of failed predictions are caused by the mislabeling between these two classes located at the up-left region of the image, see Figure 10. Table 6 illustrates that ANNC improves the classification result on such

Table 4: The results on Pavia Centre scene

Classifier	LR		CC	
FE method	ANN	ANNC	ANN	ANNC
Water	99.92	99.96	99.94	99.96
Trees	96.13	97.10	90.61	97.24
Asphalt	97.38	97.21	92.33	97.11
Self-Blocking Bricks	97.15	96.61	94.13	96.60
Bitumen	95.65	95.67	93.50	95.60
Tiles	97.94	98.10	99.32	98.22
Shadows	94.05	94.34	88.84	94.29
Meadows	99.37	99.44	94.93	99.37
Bare Soil	99.92	99.95	99.99	99.94
OA(%)	98.88	①98.98	96.90	②98.96
AA(%)	97.50	①97.60	94.84	②97.59
κ	0.9840	①0.9855	0.9561	②0.9852

Table 5: The results on Pavia University scene

Classifier	LR		CC	
FE method	ANN	ANNC	ANN	ANNC
Asphalt	88.41	90.06	88.46	90.47
Meadows	94.12	95.27	58.21	94.94
Gravel	85.76	87.71	84.09	85.52
Trees	97.43	97.35	90.77	97.41
Painted metal sheets	99.96	99.98	99.05	99.98
Bare Soil	95.68	95.71	90.01	96.09
Bitumen	95.41	95.07	95.61	94.96
Self-Blocking Bricks	88.57	85.30	88.85	88.06
Shadows	99.70	99.81	99.86	99.81
OA(%)	93.08	②93.66	75.69	①93.76
AA(%)	93.89	②94.03	88.32	①94.14
κ	0.9078	②0.9153	0.6976	①0.9166

difficult task, compared with its ANN counterpart, using either classifier.

4.4 Performance of Adaptive Spectral-Spatial Center Classifier (ANNC-ASSCC)

The performance of ANNC-ASSCC is compared with two state-of-the-art methods, namely PPF [27] and MASR [15]. To examine the proposed spatial integration strategy, ANNC-SCC is also considered. As shown in Tables 7-9, ANNC-ASSCC achieves classification improvements over ANNC-SCC in terms of all the metrics on three datasets. It demonstrates that the discriminative features facilitate the postprocessing, and the proposed spatial integration strategy is effective in classifying these features.

We also observe that ANNC-ASSCC outperforms the comparative methods on Pavia Center and Pavia University, but is inferior to MASR on Salinas. One possible reason is that MASR exploits multiscale neighborhoods at the testing stage without removing the training samples. To guarantee a fair comparison and clarify how does train-test overlap influence the classification performance, supplementary experiments are performed with slight modifications on MASR. At the testing stage, we design MASR-t that exclusively deteriorates training information by replacing all the training pixels with a L-dimensional vector of identical entries 0.01. As a contrast, MASR-r introduces an even disturb by randomly replacing the same number of pixels with the previous vector. The results are appended to Table 9. Compared with MASR, MASR-r barely deteriorates the classification performance with slight drops less than 0.1% in terms of OA, AA and κ. However, MASR-t yields over 2% decreases in terms of OA and κ, and only achieves comparable results as the proposed ANNC-ASSCC. To conclude, the proposed method provides competitive classification results over the state-of-the-art approaches on three datasets.

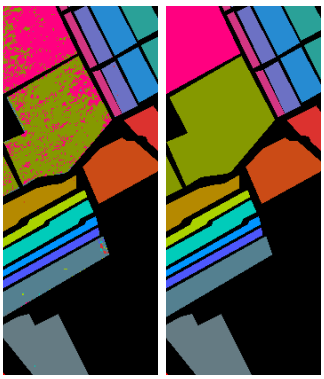


Figure 10: Comparison of predicted labels by ANN and groundtruth on Salinas. As observed, most failed predictions are caused by the mislabeling between classes Grapes untrained and Vinyard untrained located at up-left corner.

Table 6: The results on Salinas scene

Classifier	LR		CC	
FE method	ANN	ANNC	ANN	ANNC
Brocoli_green_weeds_1	99.91	99.81	99.96	99.80
Brocoli_green_weeds_2	99.91	99.97	99.54	99.96
Fallow	99.68	99.76	95.23	99.86
Fallow_rough_plow	99.66	99.66	99.71	99.68
Fallow_smooth	98.97	99.10	98.70	99.08
Stubble	99.92	99.98	98.58	99.99
Celery	99.84	99.85	99.90	99.85
Grapes_untrained	79.33	80.52	79.27	83.26
Soil_vinyard_develop	99.77	99.69	99.89	99.70
Corn_senesced_green_weeds	96.03	96.47	88.94	96.47
Lettuce_romaine_4wk	99.51	99.74	97.74	99.77
Lettuce_romaine_5wk	100.00	100.00	100.00	100.00
Lettuce_romaine_6wk	99.77	99.58	99.60	99.58
Lettuce_romaine_7wk	99.21	99.35	97.70	99.35
Vinyard_untrained	81.96	82.75	68.17	80.14
Vinyard_vertical_trellis	99.29	99.06	97.82	99.09
OA(%)	92.58	②92.98	89.84	①93.22
AA(%)	97.05	②97.21	95.05	①97.22
κ	0.9170	②0.9215	0.8863	①0.9241

Table 7: Classification accuracy (averaged over 5 runs) obtained by MASR [15], PPF [27], ANNC-SCC and ANNC-ASSCC on Pavia Centre

	MASR	PPF	ANNC-SCC	ANNC-ASSCC
Water	99.87	99.15	99.91	100.00
Trees	94.22	97.96	96.34	98.75
Asphalt	99.45	97.37	97.72	99.26
Self-Blocking Bricks	99.98	99.27	96.67	99.96
Bitumen	98.75	98.79	95.94	99.35
Tiles	80.23	98.95	98.28	99.73
Shadows	99.34	94.36	93.62	97.49
Meadows	99.90	99.90	99.41	99.99
Bare Soil	84.80	99.96	99.95	98.72
OA(%)	98.02	②99.03	98.91	①99.73
AA(%)	95.17	②98.41	97.54	①99.25
κ	0.9719	②0.9862	0.9845	①0.9961

Table 8: Classification accuracy (averaged over 5 runs) obtained by MASR [15], PPF [27], ANNC-SCC and ANNC-ASSCC on Pavia University

	MASR	PPF	ANNC-SCC	ANNC-ASSCC
Asphalt	89.97	97.25	90.06	98.69
Meadows	98.78	95.24	95.75	99.97
Gravel	99.78	94.17	86.76	93.85
Trees	97.47	97.20	97.54	96.68
Painted metal sheets	100.00	100.00	100.00	100.00
Bare Soil	99.87	99.37	94.88	100.00
Bitumen	100.00	96.16	94.81	96.88
Self-Blocking Bricks	98.76	93.83	83.68	93.11
Shadows	92.17	99.46	99.63	97.62
OA(%)	②97.42	96.25	93.60	①98.55
AA(%)	②97.42	96.97	93.68	①97.42
κ	②0.9654	0.9499	0.9145	①0.9805

5 Conclusion

This paper presented an ANN-based framework for spectral-spatial hyperspectral feature extraction and classification. The center loss was introduced to train the network in order to enhance the discriminative ability of the model. Based on the learned network, two classification strategies were proposed at the testing stage. In ANNC-SCC, the spectral features were classified using the center classifier. In ANNC-ASSCC, multi-scale spatial information was adaptively integrated to the spectral features, and the classification was performed using a voting strategy. The effectiveness of the proposed methods were demonstrated on three well-known hyperspectral images. As for future work, other loss functions proposed in deep metric learning, such as triplet loss and magnet loss, deserve investigation to address the hyperspectral feature extraction and classification problems.

Acknowledgment

This work was supported by the National Science Foundation for Young Scientists of China under grant 61701337.

References

[1] 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.

Table 9: Classification accuracy (averaged over 5 runs) obtained by MASR [15], PPF [27], ANNC-SCC and ANNC-ASSCC on Salinas

	MASR	PPF	ANNC-SCC	ANNC-ASSCC	MASR-r	MASR-t
Brocoli_green_weeds_1	100.00	100.00	99.82	100.00	100.00	100.00
Brocoli_green_weeds_2	99.97	99.89	99.91	100.00	99.89	99.86
Fallow	100.00	99.61	99.91	100.00	100.00	100.00
Fallow_rough_plow	99.89	99.50	99.63	99.53	99.92	99.50
Fallow_smooth	99.58	98.35	99.10	99.67	99.80	99.44
Stubble	100.00	99.97	99.96	100.00	99.95	100.00
Celery	99.95	100.00	99.85	100.00	99.97	99.94
Grapes_untrained	97.82	88.68	83.95	92.34	97.12	87.20
Soil_vinyard_develop	99.99	98.33	99.78	99.99	100.00	100.00
Corn_senesced_green_weeds	99.90	98.60	95.91	99.44	99.81	99.09
Lettuce_romaine_4wk	100.00	99.53	99.65	100.00	100.00	100.00
Lettuce_romaine_5wk	99.98	100.00	100.00	100.00	100.00	100.00
Lettuce_romaine_6wk	100.00	99.44	99.47	99.50	100.00	100.00
Lettuce_romaine_7wk	99.90	98.97	99.15	99.63	99.77	99.54
Vinyard_untrained	99.22	83.53	78.49	90.79	99.75	96.87
Vinyard_vertical_trellis	99.99	99.32	99.30	99.99	99.94	99.88
OA(%)	①99.38	94.80	93.12	②96.98	99.30	96.66
AA(%)	①99.76	97.73	97.12	②98.81	99.74	98.83
κ	①0.9930	0.9418	0.9230	②0.9662	0.9921	0.9627

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[29] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric learning with adaptive density discrimination," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[30] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[31] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100.

[32] 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.

[33] S. Nitish, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[34] 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.