Title: A Deep Network Architecture for Super-resolution aided Hyperspectral Image Classification with Class-wise Loss

This paper appears in: IEEE Transactions on Geoscience and Remote Sensing

Date of Publication: 2018

Author(s): Siyuan Hao; Wei Wang; Yuanxin Ye; Enyu Li; Lorenzo Bruzzone

Volume:

Page(s):

DOI:

A Deep Network Architecture for Super-resolution aided Hyperspectral Image Classification with Class-wise Loss

Siyuan Hao, Wei Wang, Yuanxin Ye, Enyu Li, and Lorenzo Bruzzone, *Fellow, IEEE*

***Abstract*—The supervised deep networks have shown great potential in improving the classification performance. However, training these supervised deep networks is very challenging for hyperspectral image given the fact that usually only a small amount of labeled samples is available. In order to overcome this problem and enhance the discriminative ability of the network, in this paper, we propose a deep network architecture for a super-resolution aided hyperspectral image classification with class-wise loss (SRCL). First, a three-layer super-resolution convolutional neural network (SRCNN) is employed to reconstruct a high-resolution image from a low-resolution image. Second, an unsupervised triplet-pipeline convolutional neural network (TCNN) with an improved class-wise loss is built to encourage intra-class similarity and inter-class dissimilarity. Finally, SRCNN, TCNN and a classification module are integrated to define the SRCL, which can be fine-tuned in an end-to-end manner with a small amount of training data. Experimental results on real hyperspectral images demonstrate that the proposed SRCL approach outperforms other state-of-the-art classification methods, especially for the task in which only a small amount of training data is available.**

***Index Terms*—Remote Sensing, Hyperspectral Image Classification, Deep Learning, Super-resolution, Class-wise Loss, Convolutional Neural Networks.**

I. Introduction

With the development of hyperspectral image sensors, the continuously increasing spatial resolution leads to blurry image, especially for the boundary between classes. These blurry images bring a great challenge to hyperspectral image classification task whose performance is mainly determined by image quality [1]. Therefore, it is very important to obtain an image with improved spatial detail before implementing classification [2]. Image super-resolution (SR) is the most widely used technique to recover a high-resolution image from a low-resolution image, and makes the blurry images more clear and sharp. Interpolation-based SR [3], reconstruction-based SR [4] and learning-based SR [5], [6] are the three most common types of super-resolution methods. Recently, some deep learning networks have been proposed to reconstruct the high-resolution image. For example, Cui *et al.* proposed a deep network cascade for image super-resolution, in which the non-local self-similarity search was integrated with a collaborative local auto-encoder [7]. However, the efficiency of this network is not very high because of its iterative strategy. In order to improve the efficiency, a deep network combined with sparse prior was presented in [8], where the domain information can be represented by a conventional sparse coding model. The sparse representation results in the acceleration of the model training speed. Moreover, the SR model based on deep spectral difference convolutional neural networks has been designed without causing spectral information distortion [9]. SR methods implemented by convolutional neural networks [10], generative adversarial networks [11] and coupled deep autoencoders [12] have also been proposed. However, in the domain of hyperspectral image, training these SR networks is very difficult given the fact that only a small amount of training samples are available.

Deep learning methods, e.g., stacked denoising auto-encoders (SdAE), convolutional neural networks (CNNs), deep belief network (DBN) and dense convolutional network (DenseNet), have also been proven to be effective tools for classification tasks [13]–[16]. The features extracted by deep networks can greatly contribute to improve classification performance, and they are more robust and invariant to most local changes [17], [18]. Accordingly, Chen *et al.* introduced the stacked auto-encoders to extract deep features for hyperspectral image classification [19]. Ma *et al.* optimized the traditional stacked auto-encoder by introducing constraints on spectral and spatial information into the reconstruction loss function [20]. To extract the spectral and spatial features simultaneously, Chen *et al.* proposed 3D convolutional neural networks (3D-CNNs) and used the l_2 regularizer and the dropout to alleviate the curse of dimensionality [21]. Then, Li *et al.* further improved the 3-D CNNs for the hyperspectral image classification task [22]. However, these models were always built based on shallow networks (2-6 layers). The latest researches show that features extracted by a deeper network are more abstract and thus have better performances in the classification task. Many deeper convolutional networks have been constructed, such as AlexNet (8 layers) [23], VGG-Net (16-19 layers) [24], GoogLeNet (22 layers) [25] and

Manuscript received April 19, 2017; revised August 26, 2017. This work was supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2017PF004 and Grant ZR2018MF002, in part by the National Natural Science Foundation of China under Grant 61701272, and in part by the Fundamental Research Funds for the Central Universities under Grant 2682016CX083. *(Corresponding author: Yuanxin Ye.)*

Siyuan Hao and Enyu Li were with the College of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China. E-mail: lemonbananan@163.com, lienyu0123@163.com.

Wei Wang and Lorenzo Bruzzone were with the Department of Information Engineering and Computer Science, University of Trento, Italy. E-mail: wei.wang@unitn.it, lorenzo.bruzzone@ing.unitn.it.

Yuanxin Ye was with the Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 610031, China. E-mail: yeyuanxin@home.swjtu.edu.cn.

residual net (152 layers) [26]. However, the application of these networks are greatly restricted in the hyperspectral image tasks, because they require lots of training samples to learn the parameters.

Considering the limitation of small training set, more investigations have been devoted to train deep networks without relying on large amounts of labeled samples. For example, Hinton *et al.* presented the deep belief networks (DBN), which can be trained in a purely unsupervised manner [27]. Then, Lee *et al.* scaled DBN to high-dimensional realistic images, and proposed the convolutional deep belief network to learn hierarchical representations using unlabeled images [28]. Meanwhile, Romero *et al.* introduced an unsupervised deep convolutional networks to seek sparse features [29] for the problem of hyperspectral image classification. Moreover, transfer learning can be also helpful to improve the classification performance with few training samples, if we can transfer the knowledge learned from other domains (e.g., ImageNet [30]) to the hyperspectral image classification task [31], [32]. Yang *et al.* presented a two-channel deep convolutional neural network (Two-CNN). They used samples from the source domain to pretrain the bottom and middle layers of the whole network, and transferred the parameters to the target domain [33]. Yuan *et al.* exploited the knowlege learned from natural images to train an improved SR network [34]. In summary, the unsupervised training and transfer learning allow deeper networks to be applied to hyperspectral image classification even with only a limited amount of training data. However, the networks above do not take the correlation among samples into account. In contrast, the Siamese Architecture can not only work in an unsupervised manner but also learn the intrinsic structure of the samples [35], [36].

Inspired by the Siamese Architecture, we propose a deep network architecture for a super-resolution aided hyperspectral image classification with class-wise loss (SRCL). First, a spatially enhanced image is obtained by super-resolution convolutional neural network, which is implemented by a three-layer CNNs. Then, the class-wise loss function is designed for TCNN, which can encourage intra-class similarity and inter-class dissimilarity. After that, the classification layers are added on the top of the TCNN to increase the discriminative ability of the network. Finally, SRCNN, TCNN and the classification layers are integrated to define SRCL, which can be fine-tuned in an end-to-end manner with a small amount of training data. The main contributions of this study consist in:

(1) The proposed SRCL is composed of three modules, with one module to construct a spatially enhanced image, one module to learn the correlation among samples, and the last one to implement classification.
(2) The super-resolution convolutional neural network (SRCNN) is used and pretrained on an auxiliary domain to solve the problem of data limitation. After that, we transfer the parameters learned on the auxiliary domain into the target hyperspectral image domain.
(3) We design a novel class-wise loss function for the triplet-pipeline convolutional neural network (TCNN) which can be trained in an unsupervised manner. This loss function is designed to encourage intra-class similarity and inter-class dissimilarity.

The rest of the paper is organized as follows. Section II briefly reviews the related works. Section III introduces the proposed method. Section IV evaluates the effectiveness of the proposed method using hyperspectral datasets. Section V analyzes the experimental results and draws the conclusion.

II. Related Works

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) include several convolutional layers, which are often combined with the pooling layers, and then followed by one or more fully-connected layers. A representative structure of CNNs is shown on the top line in Fig.3, where C_i-layer, P_i-layer and F_i-layer denote the *i*-th convolutional layer, pooling layer and fully-connected layer, respectively.

Different from the fully-connected networks whose inputs are flattened vectors, the input for CNNs should be an image patch. Let $\mathbf{x} \in \mathbb{R}^d$ represent a pixel with d-dimension in the hyperspectral image, and $\mathbf{S} \in \mathbb{R}^{d\times m\times m}$ denote the image patch centered at $\mathbf{x}$. When passing through l-th convoulutional layer and l-th pooling layer, the output feature map $\mathbf{H}_l(\mathbf{S})$ for the l-th layer can be calculated as:

$$\mathbf{H}_l(\mathbf{S}) = Pool(g(\mathbf{H}_{l-1}(\mathbf{S}) * \mathbf{W}_l + \mathbf{B}_l)) \tag{1}$$

where $Pool(\cdot)$ is the pooling operation and '$*$' denotes the convolutional operation. $\mathbf{W}_l$ and $\mathbf{B}_l$ represent the filters and the biases of the *l*-th layer, respectively. $g(\cdot)$ is the activation function, such as the $sigmoid$ function and the $relu$ function. Besides, before $\mathbf{H}_l(\mathbf{S})$ is fed into the fully-connected layer, it should be flattened to a vector in advance.

With the help of the local connections and tied weights in convolutional layers, the CNNs can take fully advantage of the spatial structure of an image. The pooling operation will further result in translation invariant features. Therefore, CNNs have been widely used to extract the spatial features. Another benefit of CNNs is that they have much fewer parameters than fully-connected networks with the same number of hidden units, which make the CNNs training easier.

B. Siamese Architecture

Siamese Architecture comprises two sub-networks and one cost model [35], whose architecture is given in Fig.1. We take as input an image pair that is represented as $\{\mathbf{S}_1, \mathbf{S}_2\}$. Let Y be a binary label for this input image pair, where $Y = 0$ if $\mathbf{S}_1$ and $\mathbf{S}_2$ belong to the same class, and $Y = 1$ if they are from the different classes. $G_w(\mathbf{S}_1)$ and $G_w(\mathbf{S}_2)$ could be yielded after the input image pair passes through the two sub-networks, and then they will be fed into the cost model. w denotes the shared parameters of two sub-networks. The output energy function $\mathbf{E}_w$ is used to measure the correlation between $\mathbf{S}_1$ and $\mathbf{S}_2$.

The parameters of Siamese Architecture can be learnt by minimizing the following contrastive loss function:

$$\begin{aligned} L(w) &= \sum_{i=1}^{P} L(w, (\mathbf{S}_1, \mathbf{S}_2, Y)^i) \\ L(w, (\mathbf{S}_1, \mathbf{S}_2, Y)^i) &= (1-Y) L_S(\mathbf{E}_w(\mathbf{S}_1, \mathbf{S}_2)^i) + Y L_D(\mathbf{E}_w(\mathbf{S}_1, \mathbf{S}_2)^i) \end{aligned} \tag{2}$$

where P is the number of image pairs in training set, L_S is the partial loss function for a similar image pair, and L_D denotes the partial loss function for a dissimilar image pair. In order to minimize the contrastive loss function, L_S and L_D should be designed to decrease the energy of similar image pairs and increase the energy of dissimilar image pairs [37]. This Siamese Architecture has the intrinsic capability to take into account properly the correlation among samples.

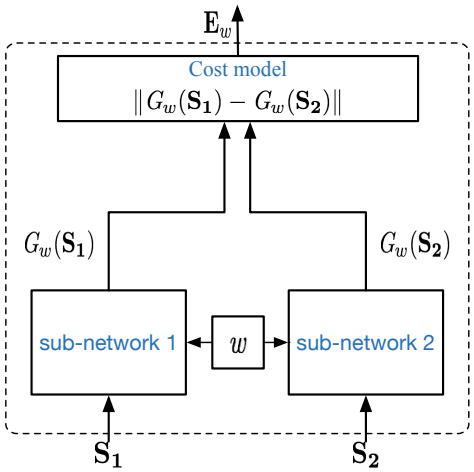


Fig. 1. **Siamese Architecture. The input pair images are $\mathbf{S}_1$ and $\mathbf{S}_2$. The output is $\mathbf{E}_w$ which represents the correlation between the input pair images $\mathbf{S}_1$ and $\mathbf{S}_2$. $G_w(\mathbf{S}_1)$ and $G_w(\mathbf{S}_2)$ are deep features which are obtained throught the two sub-networks, and the two sub-networks share the same parameter w.**

III. Proposed Method

The proposed SRCL consists of three modules, which are the SRCNN module that is in charge of reconstructing an image having enhanced resolution, the TCNN module that is in charge of learning the informative class-wise features and the classification module which performs hyperspectral image classification. The details of these three modules are introduced in the following subsections.

A. Super-resolution Convolutional Neural Network (SRCNN)

At present, the sparse-coding-based SR method and its improvements are very mature and widely used [38]–[40]. The process to reconstruct the high-resolution images can be decomposed into three operations, i.e., patch extraction and representation, non-linear mapping, and reconstruction [10]. These operations can be achieved by three different convolutional layers (see Fig.2).

The operation of patch extraction and representation is equivalent to convolving $\mathbf{S}$ by a set of filters. Formally, the first layer can be expressed as F_1:

$$F_1(\mathbf{S}) = \max(0, \mathbf{W}_1 * \mathbf{S} + \mathbf{B}_1) \tag{3}$$

where $\mathbf{W}_1$ and $\mathbf{B}_1$ denote the filters and biases, respectively, and '$*$' denotes the convolute operation. $\mathbf{W}_1$ is a 4-dimensional tensor, which corresponds to n_1 filters with size of $d \times f_1 \times f_1$. $\max(0, x)$ is the *relu* activation function.

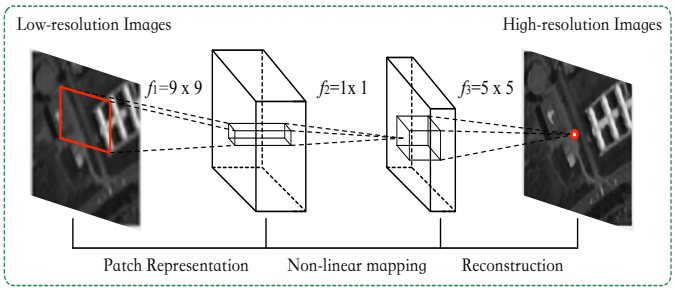


Fig. 2. **Super-resolution CNN consists of three convolutional layers. ($f_i, i \in 1, 2, 3$ is the filter size for the i-th layer.)**

The non-linear mapping in sparse-coding-based SR method is aimed to map the n_1-dimensional vector into n_2-dimensional vector. This can be achieved through n_2 filters, whose size ($f_2 \times f_2$) is 1×1. The operation of non-linear mappling using the second layer is descripted as:

$$F_2(\mathbf{S}) = \max(0, \mathbf{W}_2 * F_1(\mathbf{S}) + \mathbf{B}_2) \tag{4}$$

where $\mathbf{W}_2$ contains n_2 filters with size of $n_1 \times 1 \times 1$, and $\mathbf{B}_2$ has the dimension of n_2. It should be noted that the filter size of 1×1 can be generalized to 3×3 or 5×5, which means that the non-linear mapping is implemented on the 3×3 or 5×5 image patches instead of 1×1.

In order to reconstruct the final high-resolution image, the overlapping high-resolution patches are often averaged. It can be considered as the convolution operation of a predefined filter and feature maps. We use the the following expression to implement the reconstruction operation.

$$F_3(\mathbf{S}) = \mathbf{W}_3 F_2(\mathbf{S}) + \mathbf{B}_3 \tag{5}$$

where $\mathbf{W}_3$ represents d filters with size of $n_2 \times f_3 \times f_3$, and $\mathbf{B}_3$ is a d-dimensional vector.

In summary, the sparse-coding-based SR method can be implemented by a three-layer CNNs, and we name this network as SRCNN. However, training SRCNN is very tricky because of the limitation of the training samples in hyperspectral images. In order to overcome this problem, we apply transfer learning which is based on the assumption that the parameters could be shared among different learning models for the related tasks. Here we define the domain of ImageNet as the source domain, and the domain of hyperspectral image corresponds to the target domain. For the same classification task, we can transfer the parameters of **ImageNet** to the domain of **hyperspectral image**. After transferring the parameters, the network just need to be fine-tuned with a small amount of training images in hyperspectral image domain.

B. Triplet-pipeline Convolutional Neural Network (TCNN)

Inspired by Siamese Architecture, we introduce a novel contrastive loss function that takes into account the intrinsic structure of the samples. The structure of TCNN is illustrated in Fig.3. The TCNN consists of three spaces, i.e., the image space, the learning space and the loss space. In the image space, an image triplet is taken as input for the three parallel CNN pipelines [41]–[43]. For the learning space, the main

target is to learn the high-level mapping relationship. It is known that the nonlinear mapping can be easily achieved by a deep learning network, and we choose the convolutional neural networks in our implementation. In the loss space, the intra-class and inter-class constraints are considered when constructing *the class-wise loss function*, where the intra-class constraint serves to pull the samples of the same class closeby, whereas the inter-class constraint pushes the samples belonging to different classes far away from each other in the high-level mapping space [44], [45].

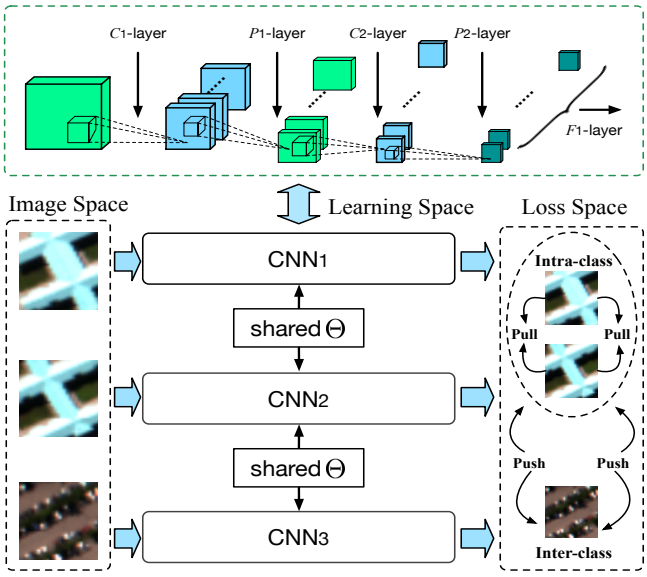


Fig. 3. **Structure of Triplet-pipeline Convolutional Neural Network. (The construction of each CNN pipeline is shown on the top line, where C_i-layer, P_i-layer and F_i-layer denote the ith convolutional layer, pooling layer and fully-connected layer, respectively.)**

Let $I_i = \{I_i^o, I_i^+, I_i^-\}$ denote the i-th image triplet in training set formed by three cropped image patches, where I_i^o, I_i^+ come from the same class, and I_i^o, I_i^- belong to the different classes. It is noted that if the center pixels of two image patches are from the same class, we consider these two image patches share the similarity, otherwise they are dissimilar with each other. $I_i = \{I_i^o, I_i^+, I_i^-\}$ is fed into three parallel CNNs, which share the same network parameters Θ, and the output of the fully-connected layer is represented by $f_\Theta(I_i) = \{f_\Theta(I_i^o), f_\Theta(I_i^+), f_\Theta(I_i^-)\}$. $f_\Theta(*)$ is the high-level mapping function learned by CNNs. In order to learn and optimize the network parameters with the help of similarity and dissimilarity between images, the class-wise loss function is constructed with the intra-class and inter-class constraints, which drive the intra-class samples to lie closeby, and inter-class samples to lie far away from each other. The designed class-wise loss function is formulated as follows:

$$L(I,\Theta) = Avr\_Pool(relu(d_\Theta(I_i^o, I_i^+, I_i^-))) = \frac{1}{N}\sum_{i=1}^{N}(relu\left(\underbrace{d(f_\Theta(I_i^o), f_\Theta(I_i^+))}_{\text{intra-class constraint}} - \underbrace{d(f_\Theta(I_i^o), f_\Theta(I_i^-))}_{\text{inter-class constraint}}\right)) \tag{6}$$

where N is the number of the image triplets in training set, and the distance function $d(\cdot,\cdot)$ is formulated using the L_2-norm function. The activation $relu$ only activates the neurons whose values are larger than 0. This means that if the distance $d_\Theta(\cdot)$ is larger than 0, the neurons will be activated and their corresponding loss will be backpropagated. Otherwise, the neurons will be deactivated and they will not be considered to update the parameters of the network.

Here we would like to explain the intuition of using the activation function $relu$. As shown in Fig.3, for any input image triplets, our target is to make the intra-class distance smaller than the inter-class distance. Therefore, if the intra-class distance is smaller than the inter-class distance, i.e., $d(f_\Theta(I_i^o), f_\Theta(I_i^+)) - d(f_\Theta(I_i^o), f_\Theta(I_i^-)) \leq 0$, the network parameters do not need to be updated since they already satisfy the target. Otherwise, if the intra-class distance is larger than the inter-class distance, i.e., $d(f_\Theta(I_i^o), f_\Theta(I_i^+)) - d(f_\Theta(I_i^o), f_\Theta(I_i^-)) > 0$, it means that the network do not satisfy the target and the parameters should be updated with respect to this image triplet. The $relu$ activation function perfectly handles the task mentioned above. Its mathematical expression can be reflected via:

$$\begin{cases} d(f_\Theta(I_i^o), f_\Theta(I_i^+)) - d(f_\Theta(I_i^o), f_\Theta(I_i^-)) \leq 0, \ fix\ \Theta \\ d(f_\Theta(I_i^o), f_\Theta(I_i^+)) - d(f_\Theta(I_i^o), f_\Theta(I_i^-)) > 0, \ update\ \Theta \end{cases} \tag{7}$$

During the phase of training, the network parameters are updated and optimized by the following stochastic gradient descent method:

$$\begin{aligned} \frac{\partial d_\Theta}{\partial \Theta} = {} & 2(f_\Theta(I_i^o) - f_\Theta(I_i^+))\frac{\partial f_\Theta(I_i^o) - \partial f_\Theta(I_i^+)}{\partial \Theta} \\ & -2(f_\Theta(I_i^o) - f_\Theta(I_i^-))\frac{\partial f_\Theta(I_i^o) - \partial f_\Theta(I_i^-)}{\partial \Theta} \end{aligned} \tag{8}$$

where $f_\Theta(I_i^o)$, $f_\Theta(I_i^+)$, $f_\Theta(I_i^-)$, and $\frac{\partial f_\Theta(I_i^o)}{\partial \Theta}$, $\frac{\partial f_\Theta(I_i^+)}{\partial \Theta}$, $\frac{\partial f_\Theta(I_i^-)}{\partial \Theta}$ can be computed through the forward and backward propagations.

It should be emphasized that TCNN can be trained in an unsupervised manner, thus it can effectively alleviate the difficulty of training the deep networks with a small amount of labeled data. Moreover, three input images are no longer required during the test phase, that is, we just need to feed the test image patches into the first CNN pipeline to obtain the corresponding high-level representations, which are the output of the last fully-connected layer of TCNN.

C. Classification Module

Through the analysis above, the network parameters of TCNN are learnt by minimizing the class-wise loss function, whilst the informative feature representation can also be extracted. In order to apply these features for the subsequent classification, a classification module should be added on the top of TCNN. Referring to the construction scheme of AlexNet [23], its performance is good enough for the classification task. Thus, we also integrate two fully-connected layers and one softmax layer into our classification module.

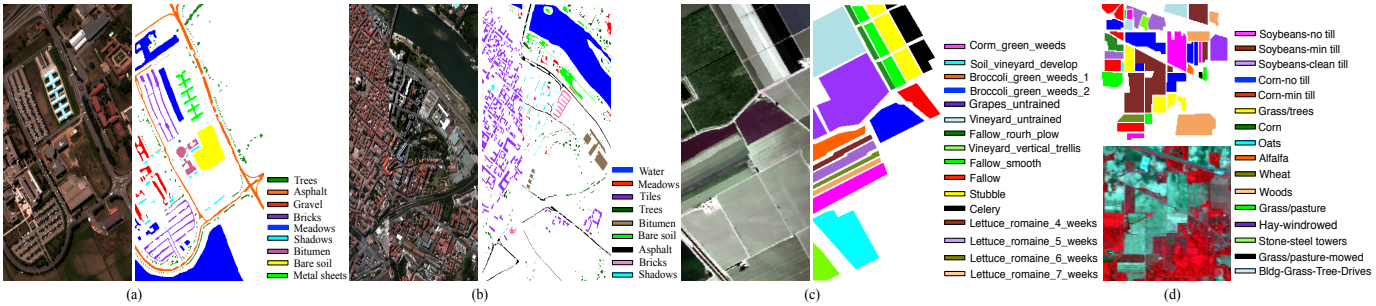


Fig. 4. **False-color composition and reference land-cover for different hyperspectral datasets (a) Pavia University dataset, (b) Pavia Center dataset, (c) Salinas Valley dataset, (d) Indian Pines dataset.**

In this way, the integrated network can not only extract the informative features but also increases the discriminative ability of TCNN.

In summary, the proposed SRCL is a cascade of SRCNN, TCNN and a classification module, in which the unsupervised learning and transfer learning are employed to overcome the difficult training problem of the deep networks. The main procedure for the definition of the proposed SRCL is depicted as follows:

Step 1: Construct the SRCNN model and transfer the knowledge of parameters, which is obtained using the ImageNet dataset, to the domain of hyperspectral image.

Step 2: Build the TCNN model and deploy the loss function including the class-wise constraints. Then initialize the shared parameter for three parallel pipelines, and update the parameters by minimizing the class-wise loss.

Step 3: Integrate the SRCNN, TCNN and the classification module together into a whole network, and fine-tune the final network with a small amount of training samples.

Step 4: Feed the image patches of test set into the proposed SRCL, and perform image classification.

D. Training Strategy

At first, we need to initialize the parameters of the three modules. In detail, the SRCNN module is pre-trained with the ImageNet data, and then the learnt parameters are transferred to the domain of hyperspectral images. In contrast, when training the TCNN, we use the hyperspectral image directly. This is because the TCNN works in an unsupervised manner, and it updates the network parameters via minimizing the class-wise loss function. For the classification module, the initial parameters of two fully-connected layers and the softmax layer are randomly initialized within the range of $[0, 1]$.

Finally, we integrate these modules into a whole network (SRCL), and fine-tune the parameters. Let γ_1, γ_2 and γ_3 represent the learning rate for each module, respectively. The process of fine-tuning can be devided into two rounds. In the first round, we only need to train the parameters of the classification module and freeze the pre-trained parameters of the SRCNN and TCNN, which is implemented by setting $\gamma_1 = \gamma_2 = 0$ and $\gamma_3 = 0.1$. In order to make the network task specific, the parameters of the final network need to be fine-tuned in the domain of hyperspectral image. Thus, in the second round, we set all the learning rates of each network to 0.1 ($\gamma_1 = \gamma_2 = \gamma_3 = 0.1$), which is equivalent to fine-tuning all the parameters of the whole network at the same time. From the analysis of the process above, we can find that the proposed network works in an end-to-end manner, and it does not need a large amount of training data to learn the parameters.

IV. Experimental Results

A. Dataset Description and Parameter Setting

In order to demonstrate the effectiveness of the proposed method, we implemented the following experiments on four benchmark hyperspectral datasets: the Pavia University, Pavia Center, Salinas Valley and Indian Pines. The related false-color compositions of the images and reference land-cover maps are shown in Fig. 4. The details of the four datasets are described as follows:

1) *Pavia University* was obtained by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the Engineering School at University of Pavia, Italy. The spatial size of the image is 610×340 pixels, and 103 spectral bands are remained after discarding the effect of noise and water absorption. It includes nine classes, *i.e.*, asphalt, meadows, trees, metal sheets, bare soil, bitumen, bricks, shadows and gravel.

2) *Pavia Center* was collected by ROSIS sensor with a spectral range from 0.43μm to 0.86μm. This image covers two dense residential areas, one of which is on a side of the river Ticino, and the other one is an open area on the other side. It has a spatial size of 1096×715 pixels and 102 spectral bands. Nine mutually exclusive classes (*i.e.*, water, tiles, meadows, trees, bitumen, bare soil, asphalt, bricks and shadows) are included in this image.

3) *Salinas Valley* was acquired by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over Salinas Valley, Southern California in 1998. It has a high spatial resolution of 3.7m. This area contains a spatial size of 512×217 pixels and 206 spectral bands from 0.4 to 2.5μm, and sixteen mutually exclusive classes are included in this image.

4) *Indian Pines* was collected by AVIRIS sensor over Northwestern Indiana in June 1992. This image contains 220 spectral bands and has a spatial size of 145×145 pixels, with each pixel measuring approximately 20m by 20m on the ground. 20 spectral bands are removed due to the noise

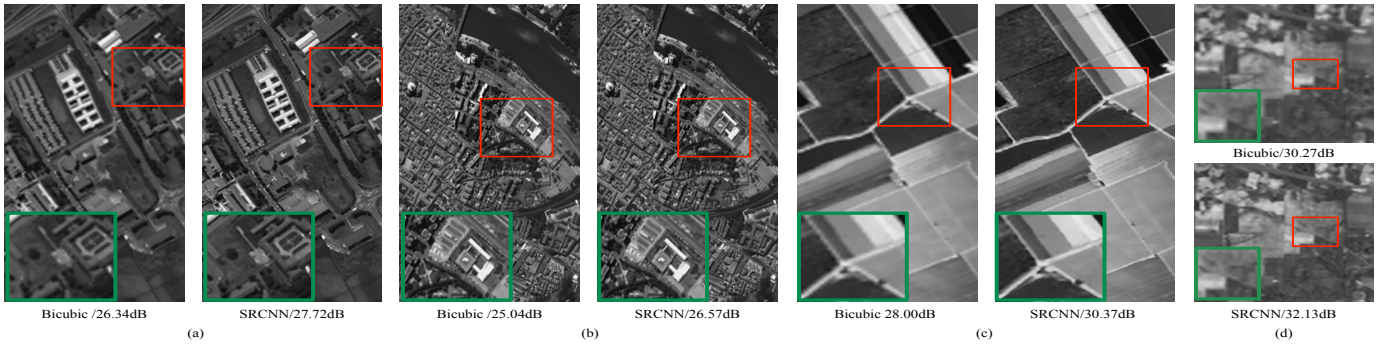


Fig. 5. **Visual super-resolution maps for different datasets (a) Pavia University dataset, (b) Pavia Center dataset, (c) Salinas Valley dataset, (d) Indian Pines dataset. The area in green box is the magnified image of the one in red box.**

TABLE I
ACCURACY COMPARISON (IN PERCENT) BETWEEN RAW FEATURES AND SR FEATURES FOR DIFFERENT DATASETS.

	Pavia University		Pavia Center		Salinas Valley		Indian Pines	
	Raw feature	SR feature	Raw feature	SR feature	Raw feature	SR feature	Raw feature	SR feature
OA	91.55±2.38	94.47±0.45	98.05±0.03	98.92±0.02	88.77±0.90	94.11±1.34	82.93±0.71	88.40±0.27
κ	88.70±2.54	92.64±1.11	97.23±0.04	98.46±0.04	87.41±1.21	93.44±1.33	80.43±0.83	86.74±0.30
Recall	91.55±2.38	94.47±0.45	98.05±0.03	98.92±0.02	88.77±0.90	94.11±1.34	82.93±0.71	88.40±0.27
F1-score	91.49±1.67	94.45±2.11	98.03±0.05	98.91±0.03	88.95±1.20	94.12±0.56	82.53±0.80	88.28±0.28

and water absorption phenomena. Sixteen mutually exclusive classes are included in this image.

For parameter setting, the filter size of each layer in SRCNN is set to $f_1 = 9 \times 9, f_2 = 1 \times 1, f_3 = 5 \times 5$ respectively. In the TCNN module, the structure of each CNN pipeline is shown on the top of Fig.3, where the spatial size of input is 7×7, the filter size of two convolutional layers is 3×3, and the filter numbers correspond to 32 and 64 for the bottom two convolutional layers. Moreover, the filter size of the pooling layer in each CNN pipeline is fixed to 2×2. For the classification module, the numbers of units in two fully-connected layers will be explored in the following section, and the class number in the final softmax layer is set according to the considered datasets. *relu* function is adopted in all the activation functions involved in the proposed architecture. When training the whole network, the initial learning rate is set to 10^{-1} and the weight decay is 0.5. To compare with other methods, we select dictionary based classification methods (OMP [46] and LC-KSVD2 [47]) and deep learning based classification methods (SdAE [48], LeNet [49], 3D-CNNs [21] and SRCNN) as the reference methods.

The super-resolution performance evaluation metric is Peak Signal-to-Noise Ratio (PSNR). The overall accuracy (OA), the kappa coefficient (κ), the average accuracy (AA), the Recall and the F1-score are used to evaluate the classification performance. To avoid biased estimation, ten independent tests were carried out using Theano and TensorFlow on a computer equipped with an Intel Core i5 Processor at 2.70-GHz. The evaluation indexes are given in the form of mean±standard deviation.

B. Investigation of the Architecture of SRCL

The proposed SRCL is a cascade of SRCNN, TCNN and a classification module. In this section, we would like to explore the performance of the three modules of the introduced SRCL architecture.

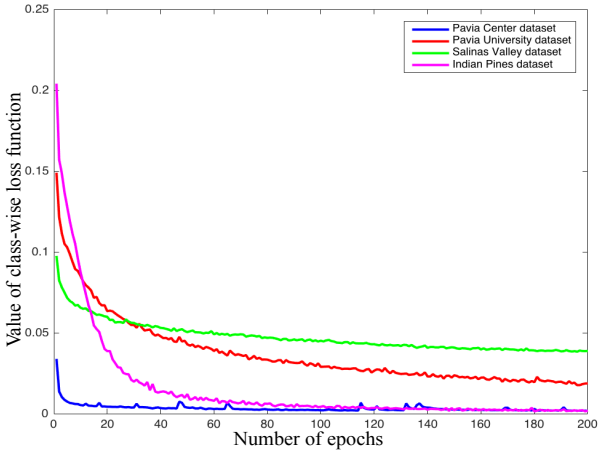


Fig. 6. **Behaviour of the class-wise loss function over the training epochs for different hyperspectral datasets.**

For the **SRCNN** module, the experimental results are summarized in Fig.5 and Tab.I. From Fig.5, we can observe that SRCNN outperforms the Bicubic baseline with gains on PSNR of 1.38dB, 1.53dB, 2.37dB and 1.86dB for the four different hyperspectral images. To visually show this advantage, the areas in the red box in Fig.5 are enlarged and placed at the left bottom of the images, which suggests the image quality is improved a lot by the SRCNN method. We can observe that the obtained super resolution images are much clearer. In addition, we compare the classification performance obtained by the super-resolution feature (SR feature) and the raw feature (Raw feature). To this purpose, the extracted features have been given as input to a reference classifier. Here we use a Support Vector Machine (SVM) to this purpose. As shown in Tab.I, the

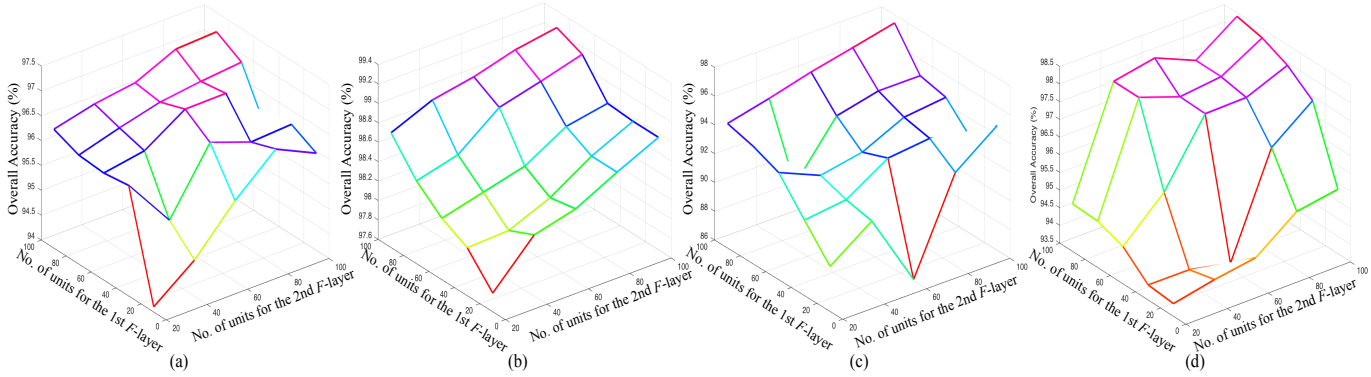


Fig. 7. **Overall accuracy versus the numbers of units in the two fully-connected layers for different hyperspectral image datasets: (a) Pavia University dataset, (b) Pavia Center dataset, (c) Salinas Valley dataset, (d) Indian Pines dataset.**

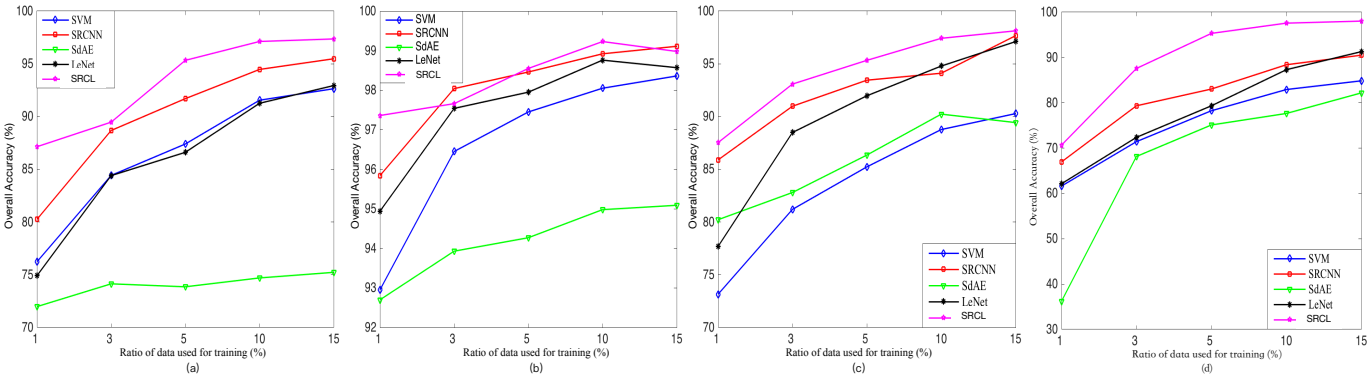


Fig. 8. **Overall classification accuracies of the models versus different ratios of training samples: (a) Pavia University dataset, (b) Pavia Center dataset, (c) Salinas Valley dataset, (d) Indian Pines dataset.**

TABLE II
COMPUTATIONAL COST FOR DIFFERENT METHODS ON DIFFERENT DATASETS.

	Pavia University		Pavia Center		Salinas Valley		Indian Pines	
Time (s)	Training	Testing	Training	Testing	Training	Testing	Training	Testing
3D-CNNs	2807	69	4187	106	2987	74	1639	54
SRCL	3156	61	4780	167	2453	62	2006	40

SR features yield higher scores in all the evaluation matrices than those provided by the Raw features. The improvement is most evident for the Indian Pines dataset, where the OA, κ, Recall and F1-score of the SR features are 88.40%, 86.74%, 88.40% and 88.28% respectively, which are higher than those obtained by the raw features. Similar observations can be done for the other three datasets. This confirms that the SR features are superior to the Raw features, and that SRCNN can help to improve the classification accuracy.

Moreover, if the parameters of SRCNN are randomly initialized, the number of parameters to learn with a small amount of training samples will increase significantly. When transfer learning is not applied, the OA, κ, Recall and F1-score of SRCL for the Pavia University dataset are 94.34%, 93.67%, 94.34% and 94.32%, respectively. These classification results are inferior compared with the case in which applying the transferred parameters (97.12% of OA, 96.18% of κ, 97.12% of Recall and 97.11% of F1-score). Therefore, transfer learning is crucial for the SRCNN module to address the problem of limited training samples.

For the **TCNN** module, we selected 10% samples from each dataset as the training set to observe the evolution of the class-wise loss in the training phase. The behaviour is shown in Fig.6. The loss function curves in Fig.6 have similar trends, i.e., as the number of epochs increases, the value of the class-wise loss decreases sharply at the beginning, and then the curves become smooth and the value decreases slowly. Finally, the loss function converges and its value becomes stable. Among these four curves, the loss function curves of the Indian Pines and Pavia University datasets decrease from relatively large values (0.2040 and 0.1490) to small stable values (0.0014 and 0.0176). On the contrary, the change of the Pavia Center dataset loss curve is relatively small, and the loss values are smaller than 0.0150 during the training phase. The loss function of the Pavia Center dataset converges after the first 20 epochs, whereas the convergence speed of other three datasets is relatively slow. The loss functions of the Pavia University dataset, the Indian Pines dataset and

TABLE III
COMPARISON OF CLASSIFICATION ACCURACIES (IN PERCENT) PROVIDED BY DIFFERENT METHODS USING 10% TRAINING SAMPLES (PAVIA UNIVERSITY).

Class	No.	SVM	Dictionary learning		Deep learning				
			OMP	LC-KSVD2	SdAE	LeNet	SRCNN	3D-CNNs	SRCL
Asphalt	6631	91.32±3.23	77.51±7.74	94.10±3.44	87.32±2.45	92.00±1.34	93.08±2.32	93.19±0.94	**97.37±0.14**
Meadows	18649	97.61±2.45	95.25±3.89	99.13±0.34	91.48±2.34	97.69±0.23	98.76±1.01	98.26±0.22	**99.90±0.04**
Gravel	2099	74.38±9.78	60.72±6.67	60.61±10.12	2.56±20.12	74.33±8.78	79.72±4.56	74.31±7.00	**86.24±4.56**
Trees	3064	86.43±0.67	82.95±2.12	89.19±5.78	83.87±1.23	93.84±2.31	91.51±2.45	96.52±1.16	**98.15±0.98**
Metal sheets	1345	87.44±3.56	99.50±0.56	99.92±0.12	97.99±0.23	**100.00±0.00**	95.12±0.98	**100.00±0.00**	**100.00±0.00**
Bare soil	5029	84.14±2.77	56.98±7.56	66.20±11.23	24.33±18.12	83.09±2.12	94.17±1.02	88.50±1.96	**98.34±0.78**
Bitumen	1330	83.71±3.12	76.52±3.78	41.19±20.13	1.99±16.45	78.45±5.43	**86.22±2.89**	84.51±2.07	81.29±3.34
Bricks	3682	88.38±2.11	76.49±5.13	86.18±5.67	81.48±2.56	76.17±1.23	88.29±2.78	85.46±1.39	**90.40±1.45**
Shadows	947	96.95±0.56	89.91±1.26	96.83±2.21	98.66±1.34	95.99±3.22	98.24±0.98	97.85±0.40	**99.30±0.12**
AA	-	87.82±3.44	79.54±5.35	81.48±5.23	63.01±2.23	87.95±2.89	91.68±2.22	90.96±1.31	**94.55±1.11**
OA	-	91.55±2.38	83.24±3.49	88.93±4.54	74.70±4.55	91.26±1.56	94.47±0.45	93.55±0.86	**97.12±0.34**
κ	-	88.70±2.54	77.45±6.27	84.98±5.56	65.27±4.78	88.34±2.56	92.64±1.11	91.41±1.14	**96.18±1.03**

TABLE IV
COMPARISON OF CLASSIFICATION ACCURACIES (IN PERCENT) PROVIDED BY DIFFERENT METHODS USING 10% TRAINING SAMPLES(PAVIA CENTER).

Class	No.	SVM	Dictionary learning		Deep learning				
			OMP	LC-KSVD2	SdAE	LeNet	SRCNN	3D-CNNs	SRCL
Water	65553	99.95±0.01	99.99±0.02	**100.00±0.00**	99.89±0.02	**100.00±0.00**	**100.00±0.00**	99.99±0.01	**100.00±0.00**
Trees	7583	95.49±1.45	89.70±3.32	96.36±1.45	86.45±3.34	95.00±1.22	**96.85±1.13**	95.04±2.82	94.97±2.56
Meadows	2940	84.77±5.34	91.12±1.67	88.78±6.76	88.87±5.66	**96.48±2.11**	93.88±3.07	93.14±6.36	92.29±1.45
Bricks	2685	82.70±3.31	74.26±5.00	65.44±7.88	75.34±3.06	92.79±2.21	94.12±0.23	93.05±0.69	**95.16±0.09**
Bare soil	6570	94.40±0.23	95.53±0.03	93.82±1.34	77.81±3.56	96.18±0.18	97.23±0.08	93.76±0.52	**98.80±0.03**
Asphalt	9230	95.75±0.15	85.52±1.76	98.13±0.11	88.82±2.45	**98.60±0.12**	96.03±0.34	98.12±0.53	98.24±0.19
Bitumen	7287	91.31±2.34	91.75±2.11	92.24±2.34	81.58±3.67	95.21±0.23	95.06±0.12	95.03±0.25	**98.66±0.12**
Tiles	42826	99.59±0.02	99.15±0.01	99.65±0.01	96.47±0.34	99.02±0.03	99.79±0.01	99.33±0.05	**99.94±0.02**
Shadows	2863	99.22±0.07	94.49±1.78	99.15±0.23	**99.95±0.24**	99.80±0.05	99.22±0.78	99.65±0.31	98.91±0.44
AA	-	93.69±1.11	91.28±1.23	92.62±1.89	88.35±2.45	97.01±1.02	96.91±0.05	96.35±0.46	**97.44±0.23**
OA	-	98.05±0.03	96.96±0.07	98.06±0.12	94.98±0.87	98.76±1.09	98.92±0.02	98.63±0.01	**99.23±0.06**
κ	-	97.23±0.04	95.69±0.12	97.25±0.23	92.89±1.34	98.25±0.23	98.46±0.04	98.06±0.02	**98.91±0.03**

the Salinas Valley dataset tend to be stable at about 120, 100 and 60 epochs. In the steady stage, there are still some small fluctuations, which are caused by the stochastic nature of stochastic gradient descend optimization method [50].

For the **classification** module, the numbers of units in two fully-connected layers are the important hyper-parameters. Fig.7 shows their effects on the OA performance. In this test, the number of units in the first fully-connected layer is represented by u_1, whilst u_2 corresponds to the number of units in the second fully-connected layer. We make u_1 and u_2 vary in the two set of $\{20, 40, 60, 80, 100\}$ and $\{10, 30, 50, 70, 90\}$, respectively. Fig.7 shows that the classification OA is generally enhanced by increasing the number of units in both fully-connected layers. For the Pavia Center dataset, the OA surface is almost stretched into a plane, which demonstrates that the OA performance is proportional to u_1 and u_2. There are some fluctuations in the surfaces of other datasets, however, the proposed method can produce the best OA results at ($u_1 = 100$, $u_2 = 90$) for these three datasets. Nevertheless, it should be noticed that when u_1 and u_2 increase, the computational cost sharply raises. Therefore, the choice of the unit number for the two fully-connected layers is very important for the performance improvement.

C. Effect of Different Ratios of Training Samples

In order to demonstrate the superiority of the proposed method in the case of a small training set, we conducted an experiment by analyzing the ratio of training samples (represented as $r \in \{1\%, 3\%, 5\%, 10\%, 15\%\}$) with different classification methods. The optimal parameters of SVM with a Gaussian kernel are tuned by a 10-fold cross-validation, while the parameter setting for SdAE and LeNet can be found in the corresponding references [48], [49]. By analyzing Fig.8(a) related to the Pavia University dataset, we can find that all the OA curves show an upward trend when increasing r. Compared with SVM, SdAE and LeNet show lower classification performance when r varies between 1% and 10%. When $r = 15\%$, LeNet outperforms SVM, however, this improvement (only 0.3% in OA) is almost insignificant. In contrast, the OA results of SRCNN and of the proposed SRCL are significantly better than those of SVM, SdAE and LeNet. The proposed SRCL also outperforms SRCNN, and the superiority is especially evident in the case of $r = 1\%$. When r increases from 1% to 15%, the OA of the proposed SRCL improves from 87.14% to 97.36%. From Figs.8(b) to (d) (which correspond to the other three considered datasets), we can observe that the performace ranking of different classification methods have changed. However, the proposed SRCL and SRCNN are more competitive than SVM, SdAE and LeNet. Moreover, at $r = 1\%$, the OAs of SRCL can reach 97.36%, 87.53% and 70.65% for these three datasets, and are both greater than those of SRCNN. This confirms that our SRCL can be trained with a small amount of training samples, and still can obtain satisfying classification performance.

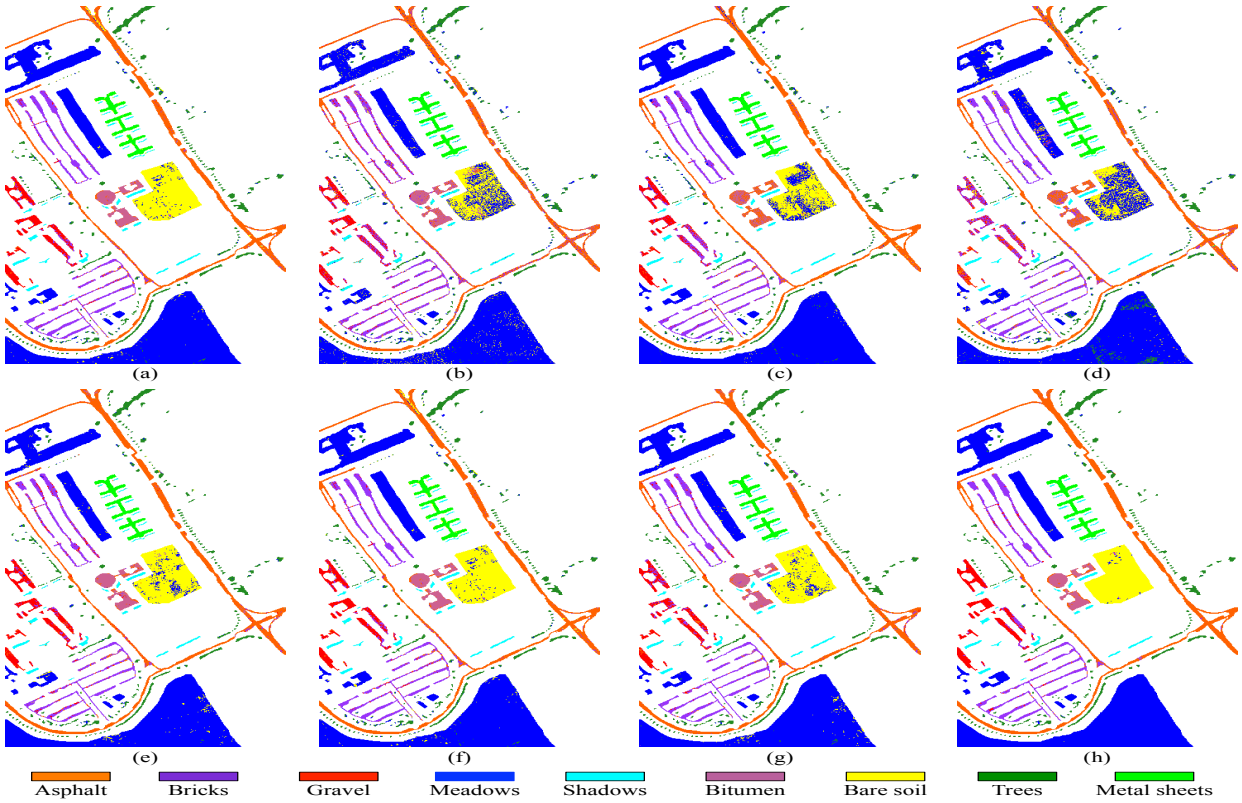


Fig. 9. **Visual comparison of the classification maps obtained by different methods for the Pavia University dataset: (a) SVM, (b) OMP, (c) LC-KSVD2, (d) SdAE, (e) LeNet, (f) CNN-SR, (g) 3D-CNNs, (h) SRCL.**

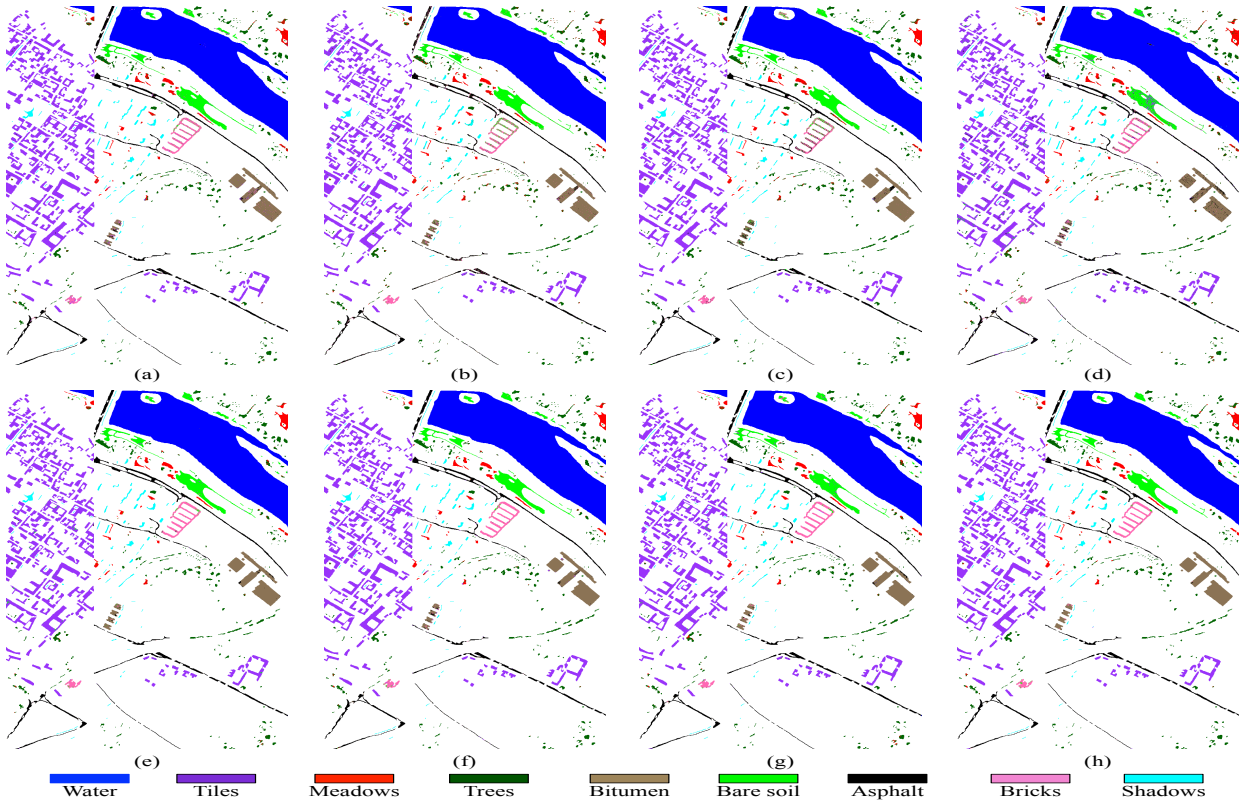


Fig. 10. **Visual comparison of the classification maps obtained by different methods for the Pavia Center dataset: (a) SVM, (b) OMP, (c) LC-KSVD2, (d) SdAE, (e) LeNet, (f) CNN-SR, (g) 3D-CNNs, (h) SRCL.**

TABLE V
COMPARISON OF CLASSIFICATION ACCURACIES (IN PERCENT) PROVIDED BY DIFFERENT METHODS USING 10% TRAINING SAMPLES (SALINAS VALLEY).

Class	No.	SVM	Dictionary learning		Deep learning				
			OMP	LC-KSVD2	SdAE	LeNet	SRCNN	3D-CNNs	SRCL
Brocoli-green-weeds-1	2009	97.57±1.99	99.50±0.12	**99.94±0.23**	97.58±0.56	97.64±0.12	85.84±2.45	98.58±1.53	98.67±0.78
Brocol-green-weeds-2	3726	97.58±1.00	**99.82±0.03**	99.76±0.02	98.10±0.24	98.83±0.12	**99.82±0.05**	99.04±0.17	98.30±0.23
Fallow	1976	95.50±1.67	98.54±1.09	95.11±2.09	91.90±2.12	95.32±0.78	93.98±2.34	**99.08±0.44**	97.53±0.56
Fallow-rough-plow	1394	95.85±2.56	99.68±0.05	99.36±0.02	96.78±0.78	**99.91±0.02**	98.41±0.67	99.50±0.30	98.09±0.28
Fallow-smooth	2678	95.85±3.45	97.43±0.56	99.00±0.09	96.47±1.56	98.74±0.67	**99.67±0.07**	99.04±0.16	97.55±0.15
Stubble	3959	96.91±1.40	99.75±0.09	99.92±0.20	99.30±0.12	99.52±0.08	**100.00±0.00**	99.82±0.20	98.93±0.98
Celery	3579	97.73±1.34	99.81±0.09	**99.88±0.15**	99.33±0.21	99.02±0.45	99.60±0.08	99.45±0.26	97.33±0.32
Grapes-untrained	11271	84.98±5.89	76.95±7.84	90.53±4.12	80.28±4.55	87.18±2.45	85.77±1.56	91.39±4.07	**97.16±0.47**
Soil-vineyard-develop	6203	98.50±0.12	99.30±0.34	**99.96±0.02**	97.22±1.03	98.83±0.76	99.86±0.12	98.61±0.37	99.00±0.07
Corn-green-weeds	3278	88.34±2.30	93.76±0.98	94.64±1.78	95.70±0.89	**98.96±0.12**	93.93±1.34	98.67±1.20	96.68±1.11
Lettuce-romaine-4wk	1068	82.31±1.34	94.07±0.12	95.42±0.78	92.76±1.34	97.62±0.04	96.36±0.90	**99.18±0.80**	96.98±2.34
Lettuce-romaine-5wk	1927	96.08±0.56	97.23±1.04	99.77±0.02	96.86±0.12	98.56±0.34	**99.83±0.02**	99.08±1.20	99.60±0.01
Lettuce-romaine-6wk	916	91.26±1.23	93.81±0.09	98.30±1.45	96.55±0.23	**99.86±0.23**	96.12±1.45	98.48±2.14	97.82±1.23
Lettuce-romaine-7wk	1070	83.28±2.33	94.70±2.12	93.98±0.78	97.86±1.07	98.34±1.67	**99.79±0.03**	97.12±2.81	97.51±0.23
Vineyard-untrained	7268	64.16±2.56	62.48±1.78	67.41±7.45	72.98±3.56	88.53±0.45	88.79±0.34	89.23±0.25	**94.48±2.56**
Vineyard-vertical-trellis	1807	89.85±1.20	98.83±0.78	**99.32±0.05**	93.90±2.11	95.56±0.90	98.40±0.67	99.06±0.70	97.17±2.10
AA	-	90.99±1.34	94.10±0.67	95.77±2.89	93.97±0.67	97.03±0.06	96.01±0.05	**97.83±0.98**	97.67±0.04
OA	-	88.77±0.90	88.99±1.78	92.78±0.50	90.23±1.22	94.79±1.09	94.11±1.34	96.37±1.22	**97.42±0.98**
κ	-	87.41±1.21	87.74±0.09	91.94±1.03	89.12±2.56	94.20±0.16	93.44±1.33	95.95±1.36	**97.12±1.45**

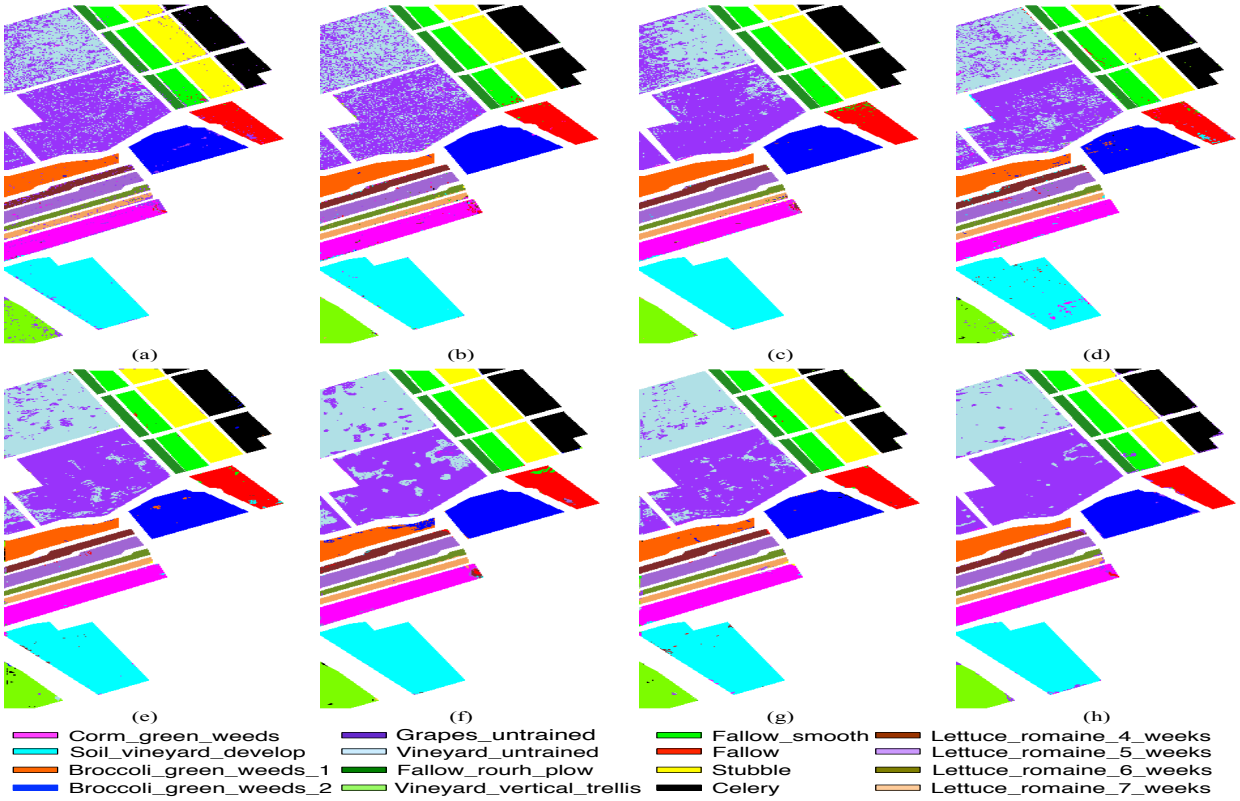


Fig. 11. **Visual comparison of the classification maps obtained by different methods for the Salinas Valley dataset: (a) SVM, (b) OMP, (c) LC-KSVD2, (d) SdAE, (e) LeNet, (f) CNN-SR, (g) 3D-CNNs, (h) SRCL.**

D. Analysis of Computational Cost

In the training phase, the overall computational cost is composed of two parts: i) the training of three modules (SRCNN (t_1), TCNN (t_2) and classification module (t_3)), and ii) the fine-tuning of the whole network (t_4). In this experiment, we set the ratio of training samples r to 10%. The parameters of SRCNN are transferred from the domain of the ImageNet. Thus t_1=0s. For the Pavia University dataset, the loss function of TCNN converges at 120 epochs, each of which takes about 19s. Accordingly, t_2=2280s. Furthermore, it consumes 798s to obtain the parameters of the classification module and takes 78s to fine-tune the whole network, i.e., t_3=798s and t_4=78s. To sum up, the overall training time is t_1+t_2+t_3+t_4=3156s.

In the testing phase, it consumes about 61s to predict the labels of testing samples for the considered dataset. The computational times taken from SRCL and 3D-CNNs for the three datasets are reported in Tab.II, from which we can observe that

TABLE VI
COMPARISON OF CLASSIFICATION ACCURACIES (IN PERCENT) PROVIDED BY DIFFERENT METHODS USING 10% TRAINING SAMPLES (INDIAN PINES).

Class	No.	SVM	Dictionary learning		Deep learning				
			OMP	LC-KSVD2	SdAE	LeNet	SRCNN	3D-CNNs	SRCL
Alfalfa	54	59.03±13.56	63.89±11.47	26.36±9.40	56.06±14.31	89.11±16.80	56.95±13.87	**92.86±10.10**	89.58±5.51
Corn-no till	1434	78.73±0.87	61.37±1.65	84.34±1.45	52.22±8.69	73.16±2.19	86.28±0.48	89.53±4.60	**97.62±0.68**
Corn-min till	834	64.31±2.08	43.07±1.97	54.42±4.29	66.95±5.52	82.04±2.87	72.89±1.068	91.74±4.92	**97.78±1.82**
Corn	234	72.07±11.43	49.84±3.61	46.17±5.36	57.11±10.89	84.01±2.43	85.88±1.98	**92.59±7.52**	81.91±9.07
Grass/trees	497	88.52±1.23	82.40±1.23	87.24±1.24	90.58±2.46	94.74±2.05	91.50±1.46	97.40±0.81	**98.51±1.01**
Grass/pasture	747	95.39±1.12	84.38±1.42	97.77±0.92	88.18±6.04	93.45±4.11	98.21±0.77	95.51±1.95	**98.76±0.09**
Grass/pasture-mowed	26	73.91±15.06	57.97±10.94	6.67±7.64	34.75±25.35	**77.58±15.30**	52.17±15.06	62.79±11.02	71.02±6.64
Hay-windrowed	489	98.48±0.35	96.59±1.20	99.66±0.30	97.85±1.38	99.57±0.74	99.02±0.47	**100.00±0.00**	**100.00±0.00**
Oats	20	38.89±20.03	24.07±12.83	2.08±3.61	47.82±40.21	**84.05±22.10**	51.85±3.21	74.73±24.87	3.70±6.41
Soybeans-notill	968	69.85±4.70	60.39±5.19	72.35±1.90	77.89±4.98	90.21±1.71	81.90±4.11	96.43±0.39	**98.55±0.37**
Soybeans-min till	2468	87.56±1.44	71.69±2.80	82.54±1.80	87.33±4.61	90.27±1.36	91.04±1.48	95.51±0.68	**98.54±0.56**
Soybeans-clean till	614	82.79±4.08	50.85±2.87	75.63±1.73	52.68±9.26	73.34±8.48	84.66±4.32	85.91±3.76	**94.57±3.18**
Wheat	212	98.07±1.52	98.60±0.61	98.82±0.00	91.70±6.09	90.36±4.20	95.26±1.06	99.59±0.58	**99.65±0.31**
Woods	1294	95.96±0.79	91.75±0.56	97.52±0.50	97.47±1.20	99.13±0.29	97.11±0.18	**99.88±0.01**	99.00±0.81
Bldg-grass-tree-drives	380	49.80±7.02	38.79±4.24	60.96±2.97	52.57±8.49	70.59±5.96	74.86±6.07	92.34±8.94	**98.74±1.44**
Stone-steel towers	95	89.41±5.13	84.31±1.80	90.79±5.74	90.34±2.29	**94.23±2.78**	92.16±4.13	88.33±11.79	94.12±2.36
AA	-	77.67±4.25	66.25±1.62	67.71±1.72	71.34±2.30	86.62±2.76	81.98±0.96	**90.94±1.01**	88.88±1.35
OA	-	82.93±0.71	69.58±0.87	81.33±0.63	77.65±1.11	87.31±0.68	88.40±0.27	94.44±1.74	**97.58±0.95**
κ	-	80.43±0.83	65.15±0.93	78.60±0.75	74.34±1.19	85.52±0.76	86.74±0.30	93.66±1.99	**97.24±1.08**

SRCL requires more time than the 3D-CNNs. This is due to the relatively complex architecture of the proposed method. However, our method achieves better performance compared with 3D-CNNs.

E. Analysis of Generalization Ability

In order to verify the generalization ability of the proposed method, we conducted training and testing on two separated regions. However, due to the spatial correlation in the hyperspectral images, it is difficult to crop a training region that is clearly separated and independent from the testing region (i.e., the pixels with the same class labels are usually located in the same region so that it is hard to partition them into training and testing sets without overlaps). Considering this factor, we selected the Pavia University dataset to do the experiment as it is relatively easier to partition this dataset into training and testing regions (i.e., the pixels with the same labels are located in different regions in the image) without any overlaps. We cropped the area with the shape of [200:350, 20:165] from the whole image as the training region and the remaining area as testing region. When the traditional SVM is used, we can obtain the classification results as OA=65.06%, AA=67.88% and κ=49.66%. We can observe that the generalization ability of the traditional SVM is limited and is seriously affected by the means of collecting training samples. The proposed SRCL method can achieve an OA=90.88%, an AA=81.47% and a κ=87.55%, which are obviously higher than those of SVM. This suggests that the generalization ability of the proposed method is better than that of SVM. From Fig. 12, we can observe that samples in the Bare soil are mostly misclassified by SVM, whereas they are better classified in the map obtained by the proposed method.

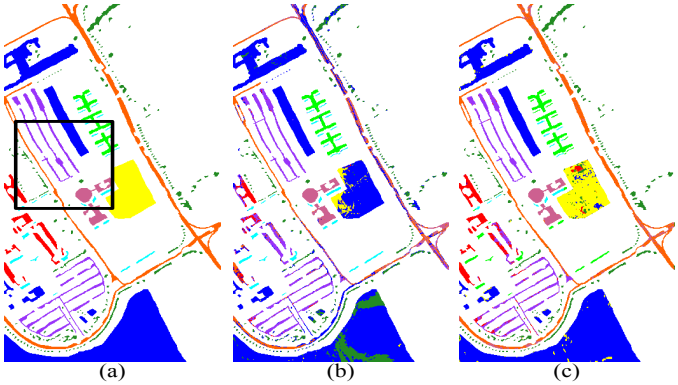


Fig. 12. **Classification maps to evaluate the generalization ability on the Pavia University dataset: (a) Regions cropped for training and testing, (b) SVM, (c) SRCL.**

F. Comparsion With Other Classification Methods

To evaluate the performance of the proposed method, we compare our SRCL with other state-of-the-art classification methods, including SVM, OMP, LC-KSVD2, SdAE, LeNet, 3D-CNNs and SRCNN. Considering the visual performance, we selected 10% of samples from each dataset for the training set. The classification accuracies of different datasets are given in Tabs. III-VI. We adopted the same experimental setting of the previous section for SVM, SdAE, LeNet, 3D-CNNs and SRCNN. The parameters of OMP and LC-KSVD2 have been selected according to [46] and [47], where the sparse level was set to 25 for the Pavia University and Pavia Center datasets and to 50 for the Salinas Valley and Indian Pines datasets.

Tab.III shows the statistical results of Pavia University dataset. We have the following observations: 1) The AA, OA and κ of the traditional SVM classifier (only based on the raw spectral features) can reach 87.82%, 91.55% and 88.70%, respectively. 2) The results of the dictionary based classification methods (OMP and LC-KSVD2) show lower performance than SVM, which suggests that these methods do not help to improve the performance of SVM for the considered dataset. 3) SdAE and CNN also can not obtain satisfying results. This is because SdAE is a fully-connected network which

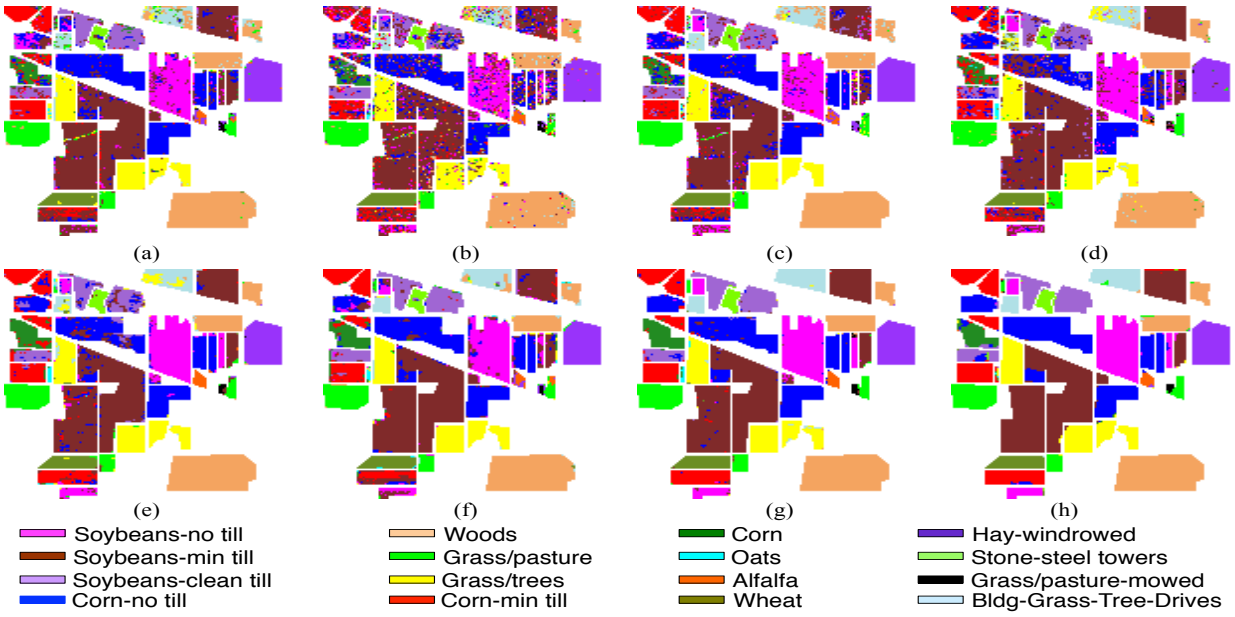


Fig. 13. **Visual comparison of the classification maps obtained by different methods for the Indian Pines dataset: (a) SVM, (b) OMP, (c) LC-KSVD2, (d) SdAE, (e) LeNet, (f) CNN-SR, (g) 3D-CNNs, (h) SRCL.**

requires more samples to learn the parameters. Nevertheless, the 3D-CNNs and SRCNN can compete with SVM with OA improvements of 2.00% and 2.92%, respectively. This observation is consistent with the conclusion that the SR features play a positive effect to improve the classification performance. 4) The proposed SRCL method can produce the best classification results (94.55% in AA, 97.12% in OA, 96.18% in κ), which are much higher than other classification methods. The class-by-class accuracy obtained by the SRCL also demonstrates a great advantage.

For illustrative purposes, the classification maps of the Pavia University dataset are presented in Fig.9. From an analysis of the figure, we can observe that the class of Bare Soil is easily misclassified as the class of Meadows, which results in the noise within the region of Bare Soil. However, this region in the map produced by the proposed SRCL is much less noisy than those of other classification methods.

As shown in Tab.IV for the Pavia Center dataset, we can observe that LC-KSVD2, LeNet, 3D-CNNs and SRCNN have better classification performance than the traditional SVM classifier. However, the improvement of the proposed SRCL is larger, and its AA, OA and κ reach to 97.44%, 99.23% and 98.91%, respectively. The corresponding classification maps can be found in Fig.10. The map of the proposed SRCL is very similar to that of the ground truth. Finally, the results of the Salinas Valley and Indian Pines datasets are shown in Tab.V-VI. In these datasets, the proposed SRCL can also produce the best classification results compared with other methods, and the same general conclusions stated for the previous datasets can be derived.

V. Conclusion

In this paper, we have proposed a novel deep network architecture for a super-resolution aided hyperspectral image classification method with class-wise loss (SRCL), which is composed of SRCNN, TCNN and a classification module. The proposed method has the following characteristics: 1) With the help of transfer learning and unsupervised training, it is effective to solve the problem of training data limitation. 2) It can learn the correlation among samples through TCNN with the novel class-wise loss function that encourages intra-class similarity and inter-class dissimilarity. 3) It can work in an end-to-end manner, therefore, it can be adapted to learn the task-specific features to better serve the hyperspectral image classification task.

To evaluate the effectiveness of the proposed method, experiments were conducted to compare its performance with those of the SVM, OMP, LC-KSVD2, SdAE, LeNet, 3D-CNNs and SRCNN on four hyperspectral datasets. Experimental results demonstrate that the proposed method outperforms other classification methods, especially when a relatively small amount of training samples is available.

The parameters of the proposed network come from three modules, i.e., SRCNN, TCNN and a classification module. Therefore, we need to train each module independently, and then the stacked network need to be fine-tuned again. This training process is time-consuming. Therefore, more efficient computational schemes for the training of the proposed architecture will be explored in our further research.

References

[1] 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.

[2] Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1279–1289, 2017.

[3] D. Rajan and S. Chaudhuri, "Generalized interpolation and its application in super-resolution imaging," *Image and Vision Computing*, vol. 19, no. 13, pp. 957–969, 2001.

[4] "Multiple frame image restoration and registration," in *Advances in Computer Vision and Image Processing Greenwich, Ct: Jai Press Inc*, 1984.

[5] J. Li, Q. Yuan, H. Shen, X. Meng, and L. Zhang, "Hyperspectral image super-resolution by spectral mixture analysis and spatialspectral group sparsity," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 9, pp. 1–5, 2016.

[6] H. Shen, L. Peng, L. Yue, Q. Yuan, and L. Zhang, "Adaptive norm selection for regularized image restoration and super-resolution," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, p. 1388, 2016.

[7] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, *Deep Network Cascade for Image Super-resolution*. Springer International Publishing, 2014.

[8] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[9] Y. Li, J. Hu, X. Zhao, W. Xie, and J. J. Li, "Hyperspectral image super-resolution using deep convolutional neural network," *Neurocomputing*, 2017.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[11] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, "Photo-realistic single image super-resolution using a generative adversarial network," 2016.

[12] K. Zeng, J. Yu, R. Wang, and C. Li, "Coupled deep autoencoder for single image super-resolution," *Cybernetics IEEE Transactions on*, pp. 1–11, 2015.

[13] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, no. 2, pp. 1–12, 2015.

[14] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, 2015.

[15] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[16] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[17] D. P. Yoshua Bengio, Pascal Lamblin and H. Larochelle, *Greedy Layer-Wise Training of Deep Networks*. MIT Press, 2007, pp. 153–160.

[18] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[19] 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, 2014.

[20] X. Ma, H. Wang, and J. Geng, "Spectral-spatial classification of hyperspectral image based on deep auto-encoder," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, no. 99, pp. 1–13, 2016.

[21] 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, 2016.

[22] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sensing*, vol. 9, no. 1, p. 67, 2017.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[27] G. E. Hinton, S. Osindero, and Y. W. Teh, *A fast learning algorithm for deep belief nets*. MIT Press, 2006.

[28] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *International Conference on Machine Learning*, 2009, pp. 609–616.

[29] 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, March 2016.

[30] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[31] 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, 2010.

[32] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 5, pp. 928–41, 2014.

[33] J. Yang, Y. Q. Zhao, and J. C. W. Chan, "Learning and transferring deep joint spectral-spatial features for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1–14, 2017.

[34] Y. Yuan, X. Zheng, and X. Lu, "Hyperspectral image superresolution by transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, no. 99, pp. 1–12, 2017.

[35] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 539–546.

[36] R. Hadsell, S. Chopra, and Y. Lecun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1735–1742.

[37] Y. Lecun and F. J. Huang, "Loss functions for discriminative training of energy-based models," in *Proc. of the 10-Th International Workshop on Artificial Intelligence and Statistics*, 2005.

[38] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[39] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*, 2014, pp. 111–126.

[40] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution." *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 21, no. 8, pp. 3467–78, 2012.

[41] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1386–1393.

[42] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1335–1344.

[43] M. Kim, S. Alletto, and L. Rigazio, "Similarity mapping with enhanced siamese network for multi-object tracking," *CoRR*, vol. abs/1609.09156, 2016. [Online]. Available: http://arxiv.org/abs/1609.09156

[44] S. Hao, W. Wang, Y. Yan, and L. Bruzzone, "Class-wise dictionary learning for hyperspectral image classification," *Neurocomputing*, 2016.

[45] Winn and Minka, "Object categorization by learned universal visual dictionary," vol. 2, pp. 1800–1807, 2005.

[46] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *Cs Technion*, vol. 40, 2009.

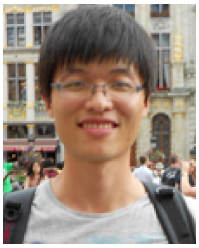
[47] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: learning a discriminative dictionary for recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[48] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, 2010.

[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[50] J. Yang, Y. Q. Zhao, and C. W. Chan, "Learning and transferring deep joint spectral-spatial features for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1–14, 2017.

Siyuan Hao (M'17) received her pH.D. degree from the College of Information and Communications Engineering in Harbin Engineering University, Harbin, China, in 2015. She is currently a Researcher at Qingdao University of Technology, China, where she teaches remote sensing and electrical communication. Her research interests focus on hyperspectral imagery processing and machine learning.



Wei Wang received the master degree from the University of Southern Denmark. He is currently a PhD student in Multimedia and Human Understanding Group with the Department of Information Engineering and Computer Science (DISI) in the University of Trento, Italy. His research interests include computer vision and deep learning. In particular, he is interested in face and human pose analysis.

Yuanxin Ye (M'17) received the B.S. degree in Remote Sensing Science and Technology from Southwest Jiaotong University, Chengdu, China, in 2008, and the Ph.D. degree in Photogrammetry and Remote Sensing from Wuhan University, Wuhan, China, in 2013. Since Dec. 2017, he has been an Associate Professor with the Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu, China. His research interests include remote sensing image processing, image registration, feature extraction, and change detection. He received The ISPRS Prizes for Best Papers by Young Authors of 23th International Society for Photogrammetry and Remote Sensing Congress (Prague, July 2016).

Enyu Li (M'18) received a BEng degree in electronic information engineering from Shandong Normal University in 2005, received a Master degree in pattern recognition and intelligent systems from Sichuan University of Science and Engineering in 2009, and received his Ph.D. degree in communication and information system in 2013 from Chongqing University. He is currently an associate professor at Qingdao University of Technology. His current research interests include cooperative communications, cognitive radio and physical-layer security.

Lorenzo Bruzzone (S'95-M'98-SM'03-F'10) received the Laurea (M.S.) degree in electronic engineering (summa cum laude) and the Ph.D. degree in telecommunications from the University of Genoa, Italy, in 1993 and 1998, respectively. He is currently a Full Professor of telecommunications at the University of Trento, Italy, where he teaches remote sensing, radar, and digital communications. Dr. Bruzzone is the founder and the director of the Remote Sensing Laboratory in the Department of Information Engineering and Computer Science, University of Trento. His current research interests are in the areas of remote sensing, radar and SAR, signal processing, machine learning and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects. He is the Principal Investigator of many research projects. Among the others, he is the Principal Investigator of the Radar for icy Moon exploration (RIME) instrument in the framework of the JUpiter ICy moons Explorer (JUICE) mission of the European Space Agency. He is the author (or coauthor) of 215 scientific publications in referred international journals (154 in IEEE journals), more than 290 papers in conference proceedings, and 21 book chapters. He is editor/co-editor of 18 books/conference proceedings and 1 scientific book. His papers are highly cited, as proven form the total number of citations (more than 22000) and the value of the h-index (70) (source: Google Scholar). He was invited as keynote speaker in more than 30 international conferences and workshops. Since 2009 he is a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS). Dr. Bruzzone ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Seattle, July 1998. Since that he was recipient of many international and national honors and awards, including the recent IEEE GRSS 2015 Outstanding Service Award and the 2017 IEEE IGARSS Symposium Prize Paper Award. Dr. Bruzzone was a Guest Co-Editor of many Special Issues of international journals. He is the co-founder of the IEEE International Workshop on the Analysis of Multi-Temporal Remote-Sensing Images (MultiTemp) series and is currently a member of the Permanent Steering Committee of this series of workshops. Since 2003 he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He has been the founder of the IEEE Geoscience and Remote Sensing Magazine for which he has been Editor-in-Chief between 2013-2017.Currently he is an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing. He has been Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society between 2012-2016.