

# Hyperspectral Unmixing via Deep Convolutional Neural Networks

Xiangrong Zhang<sup>1</sup>, Senior Member, IEEE, Yujia Sun, Jingyan Zhang, Peng Wu, and Licheng Jiao, Fellow, IEEE

**Abstract**—Hyperspectral unmixing (HU) is a method used to estimate the fractional abundances corresponding to endmembers in each of the mixed pixels in the hyperspectral remote sensing image. In recent times, deep learning has been recognized as an effective technique for hyperspectral image classification. In this letter, an end-to-end HU method is proposed based on the convolutional neural network (CNN). The proposed method uses a CNN architecture that consists of two stages: the first stage extracts features and the second stage performs the mapping from the extracted features to obtain the abundance percentages. Furthermore, a pixel-based CNN and cube-based CNN, which can improve the accuracy of HU, are presented in this letter. More importantly, we also use dropout to avoid overfitting. The evaluation of the complete performance is carried out on two hyperspectral data sets: Jasper Ridge and Urban. Compared with that of the existing method, our results show significantly higher accuracy.

**Index Terms**—Convolutional neural networks (CNNs), end-to-end model, spectral unmixing, spectral-spatial information.

## I. INTRODUCTION

**H**YPERSPECTRAL remote sensing has been used in many applications and has bright prospects for use in the future applications. Hyperspectral image (HSI) data are characterized as multiband and have a high resolution in spectral space. Although hyperspectral data have high spectral resolution, they have relatively low spatial resolution, which results in the mixing phenomenon being inevitable in HSIs. The spectrum of a pixel is a combination of multiple spectra of various endmembers in accordance with a certain proportion, rather than a pure pixel. Therefore, hyperspectral unmixing (HU) is an important technique for hyperspectral data exploitation [1].

The HU model can be divided into two categories: the linear spectral mixture model (LSMM) and nonlinear spectral mixture model (NLSMM) [2]. The LSMM assumes that the pixel spectrum is a linear combination of the spectral components, and the NLSMM holds when the light suffers multiple scattering owing to different materials being involved. Keshava and Mustard [3] have studied and discussed the

mechanism and application range of linear mixed spectra and nonlinear mixed spectra.

Given that the LSMM is simpler and has some physical meaning, linear spectral unmixing (LSU) algorithms have been the focus of a significant amount of research. Traditional unmixing methods usually include two important procedures: endmember extraction and fraction estimation [4]. An endmember extraction algorithm mainly includes the pure pixel index [5], vertex component analysis [6], and orthogonal bases algorithm [7]. In recent years, Iordache *et al.* [8] introduced the idea of sparseness to HU algorithms. The advantage of this algorithm is that it does not have to assume the existence of pure pixels and avoids the potential errors produced by endmember extraction. However, the choice of spectral library leads to instability in the results.

In general, the LSMM is only applicable to scenes that are inherently or macroscopically considered to be linearly mixed, and for some special scenes that are more difficult to describe accurately, it is necessary to consider the more complex NLSMM. In recent times, two new techniques for nonlinear spectral unmixing have been attracting attention. In [9], the generalized bilinear model and hierarchical Bayesian algorithm for nonlinear unmixing of HSIs were proposed. In [10], the autoassociative neural network (AANN) for pixel abundances from hyperspectral data was presented, which expands over previous efforts in the literature focused on using neural networks (NNs) for nonlinear unmixing purposes. The NN structure consists of two stages: at the first stage features of the input vector are extracted and at the second stage a mapping is performed from the extracted feature space to obtain the abundance. However, this method does not take into account the joint information between spatial and spectral information, and the feature extraction and unmixing must be trained separately.

Recently, deep learning has attracted much attention and been applied to many domains such as object detection [12], [13] and image classification [14]. In particular, a convolutional NN (CNN) is one of the most popular networks owing to its capability to automatically discover relevant contextual features. Therefore, the CNN has been used for HSI classification [15] in which a 3-D CNN model was used to extract spectral-spatial features and classify the extracted features. However, there are a few works that consider HU by using NN, for example the AANN [10].

In this letter, an end-to-end pixel-based CNN and cube-based CNN for HU are proposed. Owing to the good feature learning performance of CNNs, a CNN is used to explore contextual features of HSIs, and then obtain the abundance of a pixel by using the multilayer perceptron (MLP) architecture

Manuscript received April 2, 2018; revised June 21, 2018; accepted July 15, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61772400, Grant 61501353, Grant 61772399, Grant 91438201, and Grant 61573267. (Corresponding author: Xiangrong Zhang.)

The authors are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: xrzhang@mail.xidian.edu.cn).

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LGRS.2018.2857804

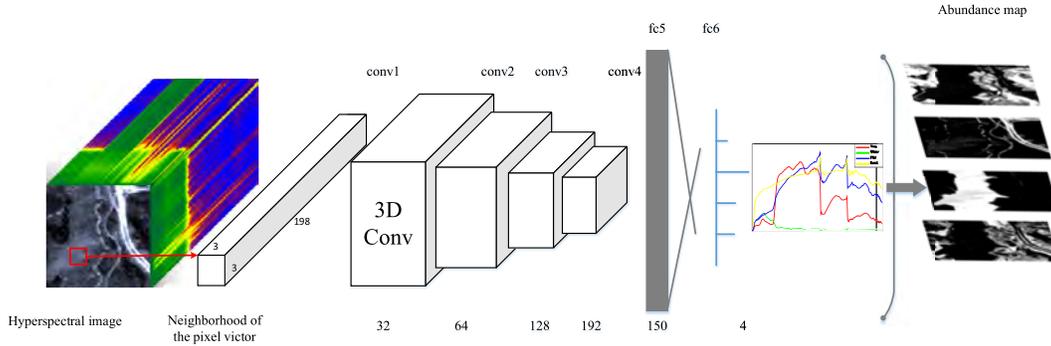


Fig. 1. Architecture of cube-based CNN with spectral-spatial feature extraction of HSI.

in the last layer. The main characteristics of our framework are as follows.

- 1) This work represents the first attempt to address the spectral unmixing using a deep CNN. In this way, discriminative feature representations can be automatically learned and interpreted.
- 2) The framework is an end-to-end model, which is straightforward and fast and avoids manual preprocessing and subsequent processing.
- 3) The spectral-spatial information is considered in the unmixing by the HSI patch, which helps in for unmixing performance improvement.

## II. PROPOSED FRAMEWORK

### A. Pixel-Based Convolution Spectral Hyperspectral Unmixing

CNNs have been successfully used for image classification [15]. The main characteristic of a CNN is the weight sharing, which can significantly reduce the number of NN parameters, and thus prevent the emergence of over fitting, while reducing the complexity of the NN model. CNNs usually consist of one or more pairs of convolution and down-sampling layers, and finally end with several fully connected layers. Owing to the good capability of high-level feature learning, we propose to integrate features learning and unmixing of HSI into a network by using a CNN. In the method, discriminative features are extracted by a CNN, and then, the features are used as the input to a new MLP architecture at the last layer of the network for a pixel-based fuzzy classification procedure that can obtain the abundance of a pixel by normalizing the results [16].

As depicted in Fig. 1, the network that we use contains eleven layers for HU. The input of the network is a spectral vector of a pixel in the HSI, and the output is the corresponding label that represents the abundance of land covers for the pixel. Furthermore, the network consists of four convolution layers, four down-sampling layers, and two fully connected layers. After several layers of convolution and down sampling, we can extract the high-level information of the input vector. Meanwhile, the fully connected layer is used to fuzzily classify the extracted features and obtain the abundance corresponding to the endmembers for each pixel.

For the fully connected layer, it is assumed that the output unit is “1” when it is associated with the actual land cover and the others are “0.” Although the network is trained using binary values in the output vector, the activation function of the processing layer is a sigmoid function that results

in the output value being in the range [0, 1]. Thus, such values can be considered to represent the abundance, which is correlated with fuzzy membership values. The abundance  $a_i$  corresponding to the  $i$ th class is given by the following equation:

$$a_i = o_i / \sum_{k=1}^M o_k \quad (1)$$

where  $o_k$  denotes the CNN output associated with the  $k$ th endmember and  $M$  is the number of endmembers.

After establishing the model, the next step is training the network. The parameters of the CNN are randomly initialized and trained by an error back-propagation algorithm. In our implementation, we use the mini-batch strategy to update. Finding the optimal NN for unmixing accurately to minimize a loss function  $L$  between the predicted values and the target values in a training set

$$L = -\frac{1}{m} \sum_{i=1}^m [y'_i \log y_i] \quad (2)$$

where  $y'_i$  is the output vector of the  $i$ th layer,  $y_i$  is the true abundance of the input pixel vector, and  $m$  is the number of mini batch, which is set to 100. The objective is to optimize (2) using the AdaGrad optimizer with a batch size of 100.

### B. Cube-Based Convolution Spectral-Spatial Hyperspectral Unmixing

In addition to spectral information, spatial information is also very important for HSI analysis. The model integrating spectral and spatial information will improve the performance further. CNNs are good at 2-D image analysis. Integrating spectral and spatial information of HSI, the cube-based convolution is adopted for HSI unmixing.

Cube-based convolution is described as follows. Formally, the value at position  $(x, y, z)$  of the  $j$ th feature map in the  $i$ th layer is given by

$$v_{ij}^{xyz} = \text{ReLU} \left( b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \right) \quad (3)$$

where  $\text{ReLU}(\cdot)$  is the activate function,  $b_{ij}$  is the bias of the  $j$ th feature map in the  $i$ th layer,  $m$  indexes the feature map in the  $(i-1)$ th layer connected to the current feature map, and  $P_i$  and  $Q_i$  are the height and width of the spatial

TABLE I  
ARCHITECTURE OF NN

| Layer Name      | Input       | C2<br>S3   | C4<br>S5   | C6<br>S7         | C8<br>S9         | F10 | F11 |
|-----------------|-------------|------------|------------|------------------|------------------|-----|-----|
| Pixel-based CNN | 1×1×198     | 1×5<br>1×2 | 1×4<br>1×2 | 1×5<br>1×2       | 1×4<br>1×2       | FC  | FC  |
| Kernel Size     | Feature map | 1          | 3          | 6                | 12               | 24  |     |
| Cube-based CNN  | 3×3×198     | 5×1×1      | 4×1×1      | 5×1×1<br>dropout | 4×1×1<br>dropout | FC  | FC  |
| Kernel Size     | Feature map | 1          | 16         | 32               | 64               | 128 |     |

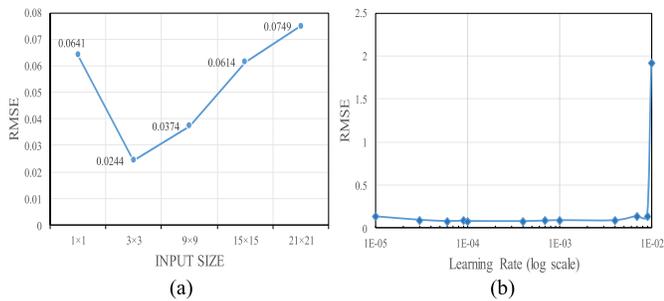


Fig. 2. Quantitative analysis of (a) different input size and (b) learning rate for the cube-based CNN method in the Jasper Ridge data set.

convolution kernel, respectively.  $R_i$  is the size of the 3-D kernel along the spectral dimension and  $w_{ijm}^{pqr}$  is the  $(p, q, r)$ th value of the kernel connected to the  $m$ th feature map in the previous layer [17].

As depicted in Fig. 1, we choose a local spatial region with a certain pixel as the center and form a new data cube with its spectral dimension data. The data cube is used as the input of the CNN model to perform spectral unmixing. The size of the input cube is  $K \times K \times B$ , where  $K \times K$  is the spatial window size and  $B$  indexes the number of spectra. Specifically, each layer of the CNN contains 3-D convolution and cube-based down sampling. We use the dropout strategy to avoid over fitting, which is also helpful for improving unmixing performance.

### III. EXPERIMENTS

To test the performance of the proposed method, we apply our method to different public data sets, including Jasper Ridge and Urban data sets. The ground truth for the data sets dates from [18].

#### A. Jasper Ridge

Jasper Ridge is a popular hyperspectral data set. It has  $100 \times 100$  pixels. Each pixel is recorded on 224 channels ranging from 380 to 2500 nm. After removing the channels affected by water vapor and the atmosphere, we obtained 198 channels. There are four endmembers in the data set: “road,” “soil,” “water,” and “tree” [19]. On the Jasper Ridge data set, we randomly select about 7500 pixels for learning weights and biases of each neuron, and the remaining 2500 pixels are used for testing.

The concrete model of the network is shown in Table I. C refers to the convolution layers, S refers to the pooling layers, and F refers to the fully connected layer. We perform sensitivity analysis of the learning rate parameter on the Jasper Ridge data set. From Fig. 2(b), we can see that,

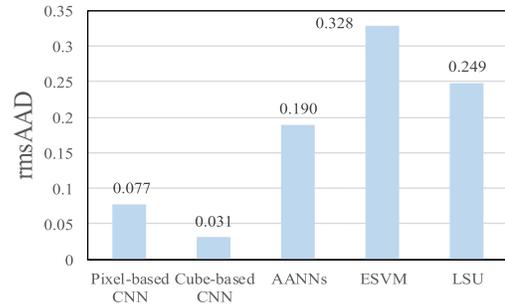


Fig. 3. rmsAAD values of the Jasper Ridge data set by different methods.

TABLE II  
ABUNDANCE STATISTICS AND RMSE VALUES OF JASPER RIDGE DATA SET BY DIFFERENT METHODS

| End-member | LSU   | AANN  | eSVM  | Pixel-based CNN | Cube-based CNN |
|------------|-------|-------|-------|-----------------|----------------|
| Tree       | 0.184 | 0.065 | 0.093 | 0.031           | 0.017          |
| Water      | 0.202 | 0.052 | 0.084 | 0.022           | 0.011          |
| Soil       | 0.135 | 0.091 | 0.128 | 0.037           | 0.027          |
| Road       | 0.115 | 0.071 | 0.091 | 0.035           | 0.022          |
| Sum        | 0.636 | 0.279 | 0.396 | 0.125           | 0.077          |

TABLE III  
ABUNDANCE STATISTICS AND RMSE VALUES OF URBAN DATA SET BY DIFFERENT METHODS

| End-member | LSU   | AANN  | eSVM  | Pixel-based CNN | Cube-based CNN |
|------------|-------|-------|-------|-----------------|----------------|
| Asphalt    | 0.256 | 0.153 | 0.110 | 0.048           | 0.031          |
| Grass      | 0.296 | 0.167 | 0.156 | 0.047           | 0.027          |
| Tree       | 0.300 | 0.160 | 0.106 | 0.037           | 0.022          |
| Roof       | 0.156 | 0.089 | 0.078 | 0.026           | 0.021          |
| Metal      | 0.165 | 0.125 | 0.057 | 0.034           | 0.019          |
| Dirt       | 0.166 | 0.120 | 0.133 | 0.044           | 0.025          |
| Sum        | 1.339 | 0.814 | 0.640 | 0.236           | 0.145          |

for the cube-based CNN when the learning rate is between 0.0001 and 0.001, the root-mean-square error (RMSE) has the lowest value. Therefore, we set the learning rate  $\lambda$  as 0.0005. Similarly, for the pixel-based CNN, we set the learning rate  $\lambda$  as 0.01. We test different sizes of fields to choose an optimal size for the cube-based CNN framework. We compare several sizes of input, namely,  $1 \times 1$ ,  $3 \times 3$ ,  $9 \times 9$ ,  $15 \times 15$  and  $21 \times 21$ ; the corresponding RMSE is shown in Fig. 2(a). It is clear that the best output is obtained when the input size is  $3 \times 3$ . Therefore, we adopt  $3 \times 3$  as the input size in the two public data sets.

To evaluate the performance of our methods, we compare our methods with three existing state-of-the-art approaches; AANN, LSU, and extended support vector machine (eSVM) [20]. For AANN, the neuron numbers of the three hidden layers are set as 30, 4, and 30. The training phase lasts 700 epochs, and the learning rate is set as 0.01. For LSU, we use fully constrained least-squares LSU, which satisfies the abundance sum-to-one and the abundance nonnegativity constraints. The eSVM uses an RBF kernel, and parameters are selected by five-fold cross validation.

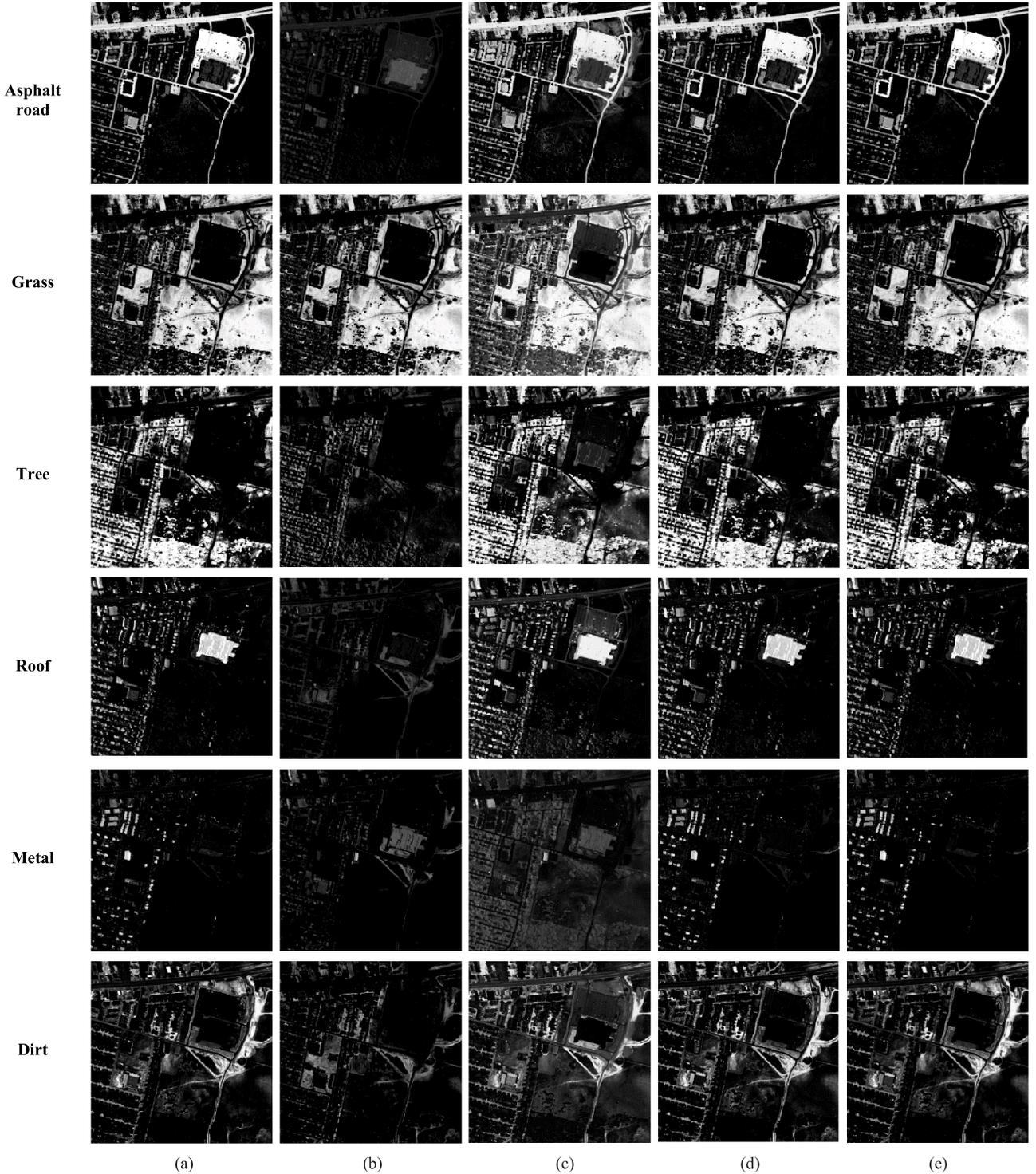


Fig. 4. Ground-truth and estimated abundances obtained for each endmember material in the Urban data set by different methods (a) Ground truth. (b) LSU (c) AANN (d) Pixel-based CNN (e) Cube-based CNN.

We use the root-mean square of the abundance angle distance (rmsAAD) for quantitative evaluation of performance. The abundance angle distance (AAD) measures the similarity between the original abundance fractions ( $a_{wi}$ ) and estimated ones ( $a_{ri}$ ) as formulated in

$$\text{AAD}_{a_{wi}} = \cos^{-1} \left( \frac{a_{wi}^T a_{ri}}{\|a_{wi}\| \|a_{ri}\|} \right) \quad (4)$$

$$\text{rmsAAD} = \left( \frac{1}{N} \sum_{i=1}^N (\text{AAD}_{a_{wi}})^2 \right)^{1/2}. \quad (5)$$

Fig. 3 depicts the rmsAAD values of different methods. We can see that the pixel-based CNN and cube-based CNN achieve better result than those of AANN, LSU, and eSVM on the Jasper Ridge data set in terms of rmsAAD.

Table II gives a quantitative assessment by calculating the RMSE of the entire testing set. For each land cover type,

the RMSE value is given by

$$\text{RMSE} = \sqrt{\frac{\sum_{i \in N} (a_{wi} - a_{ri})^2}{N}} \quad (6)$$

where  $N$  is the number of pixels in the testing set,  $a_{wi}$  is the true abundance of pixel  $i$ , and  $a_{ri}$  is the estimated abundance by unmixing methods. We can see that the RMSE of the pixel-based CNN is 0.125, which is much lower than that of the eSVM 0.396. In addition, the pixel-based CNN RMSE is approximately half of the RMSE obtained by the AANN. The RMSE of the cube-based CNN is about half of the RMSE obtained by the pixel-based CNN.

### B. Urban

Urban is one of the most widely used hyperspectral data sets in hyperspectral hybrid studies. It has  $307 \times 307$  pixels, and there are 210 wavelengths from 400 to 2500 nm. After removing the channels affected by water vapor and the atmospheric environment, we obtained 162 channels. There are six endmembers in these data: “asphalt,” “grass,” “tree,” “roof,” “metal,” and “dirt.” On the Urban data set, we randomly select about 47 000 pixels for training and the remaining 47 249 pixels are used for testing.

The concrete model of the network is shown in Table I. For the pixel-based CNN, we set the learning rate  $\lambda$  as 0.01, and for the cube-based CNN, we set the learning rate  $\lambda$  as 0.001. For the AANN, the neuron numbers of the three hidden layers are set to be 40, 4 and 40. The training phase lasts 500 epochs, and the learning rate is set to be  $1e-6$ .

Table III reports the RMSE values of each endmember. For the pixel-based CNN the total RMSE value computed over all of the endmembers is 0.236, which is better than the value of 0.814 obtained with the AANN and the value of 0.640 obtained with eSVM. For the cube-based CNN, the RMSE value computed over all of the endmembers is 0.145, which is the best among all the methods. It can be seen that the CNN effectively improves the accuracy of the unmixing, and the cube-based CNN, which utilizes the spectral-spatial information achieves superior performance.

Fig. 4 shows the ground-truth and estimated abundances for each endmember material in the Urban data set. We can see that the distribution of each endmember is relatively discrete and complex. Nevertheless, the unmixing performance of our proposed methods is improved significantly. We can see from the distribution maps of asphalt road that the unmixing performance achieved by the CNN is more accurate than that by the other methods. The pixel-based CNN works better than LSU and the AANN. The cube-based CNN further enhances the unmixing performance, and the estimated abundances for each endmember are close to the true distribution image.

## IV. CONCLUSION

In this letter, we have presented a pixel-based CNN for HSI unmixing. To combine the spatial correlation between the image features in the unmixing process, we have proposed the cube-based CNN for spectral-spatial HU. It uses a 3-D CNN model to extract spectral-spatial features, and then obtain the abundance of a pixel by using MLP architecture in the last layer. Experimental results on the data sets of Jasper Ridge

and Urban have demonstrated that the proposed methods outperform the LSU and AANN. In addition, compared to the pixel-based CNN, the cube-based CNN, which includes more enriched features, achieves better performance. Deep learning achieves better results owing to a large number of labeling samples, but in reality, the labeling samples are difficult to obtain. In the future work, we can use other learning methods, such as few-shot learning, to overcome this drawback of HU.

## REFERENCES

- [1] S. Bernabé *et al.*, “Performance-power evaluation of an OpenCL implementation of the simplex growing algorithm for hyperspectral unmixing,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 304–308, Mar. 2017.
- [2] X. Zhang, J. Zhang, C. Li, C. Cheng, L. Jiao, and H. Zhou, “Hybrid unmixing based on adaptive region segmentation for hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3861–3875, Jul. 2018.
- [3] N. Keshava and J. F. Mustard, “Spectral unmixing,” *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.
- [4] V. Menon, Q. Du, and J. E. Fowler, “Random Hadamard projections for hyperspectral unmixing,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 419–423, Mar. 2017.
- [5] J. W. Boardman, F. A. Kruse, and R. O. Green, “Mapping target signatures via partial unmixing of AVIRIS data,” in *Proc. Summaries JPL Airborne Earth Sci. Workshop*, 1995, pp. 23–26.
- [6] J. M. P. Nascimento and J. M. Bioucas-Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.
- [7] X. Tao, B. Wang, and L. Zhang, “Orthogonal bases approach for the decomposition of mixed pixels in hyperspectral imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 219–223, Apr. 2009.
- [8] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, “Sparse unmixing of hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.
- [9] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, “Nonlinear unmixing of hyperspectral images using a generalized bilinear model,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4153–4162, Nov. 2011.
- [10] G. A. Licciardi and F. D. Frate, “Pixel unmixing in hyperspectral data by means of neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4163–4172, Nov. 2011.
- [11] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [12] J. Gao, Q. Wang, and Y. Yuan, “Embedding structured contour and location prior in siamese fully convolutional networks for road detection,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May/June. 2017, pp. 219–224.
- [13] Q. Wang, J. Wan, and Y. Yuan, “Locality constraint distance metric learning for traffic congestion detection,” *Pattern Recognit.*, vol. 75, pp. 272–281, Mar. 2018.
- [14] W. Shao and S. Du, “Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Oct. 2016.
- [15] 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 Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [16] G. M. Foody, “Approaches for the production and evaluation of fuzzy land cover classifications from remotely-sensed data,” *Int. J. Remote Sens.*, vol. 17, no. 7, pp. 1317–1340, Apr. 1996.
- [17] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [18] S. Jia and Y. Qian, “Spectral and spatial complexity-based hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3867–3879, Dec. 2007.
- [19] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, “Structured sparse method for hyperspectral unmixing,” *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 101–118, Feb. 2014.
- [20] L. Wang and X. Jia, “Integration of soft and hard classifications using extended support vector machines,” *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 3, pp. 543–547, Jul. 2009.