Hyperspectral Unmixing Using Deep Convolutional Autoencoders in a Supervised Scenario

Farshid Khajehrayeni, *Student Member, IEEE*, and Hassan Ghassemian, *Senior Member, IEEE*

***Abstract*—Hyperspectral unmixing (HSU) is an essential technique that aims to address the mixed pixels problem in hyperspectral imagery via estimating the abundance of each endmember at every pixel given the endmembers. This article introduces two approaches intending to solve the challenge of the mixed pixels using deep convolutional autoencoders (DCAEs), namely pixel-based DCAE, and cube-based DCAE. The former estimates abundances with the help of only spectral information, while the latter utilizes both spectral and spatial information which results in better unmixing performance. In the proposed frameworks, the weights of the decoder are set equal to the endmembers in order to address the issue in a supervised scenario. The proposed frameworks are also adapted to the VGG-Net that proved increasing depth with small convolution filters (3×3) leads to a considerable improvement. In other words, inspired by this idea, we utilize small and fixed kernels of size 3 in all layers of both proposed frameworks. The network is trained via the spectral information divergence objective function, and the dropout and regularization techniques are utilized to prevent overfitting. The superiority of the proposed frameworks is proven via conducting some experiments on both synthetic and real hyperspectral datasets and drawing a comparison with state-of-the-art methods. Moreover, the quantitative and visual evaluation of the proposed frameworks indicate the necessity of integrating spatial information into the HSU.**

***Index Terms*—Deep convolutional autoencoders (DCAEs), hyperspectral unmixing (HSU), spectral-spatial information, VGG-Net.**

I. Introduction

REMOTE sensing (RS) aims to acquire information about phenomena or objects on the Earth without any physical contact [1]. Hyperspectral data processing is a hot topic in this field that deals with hyperspectral images (HSIs), gathered by hyperspectral sensors in hundreds or thousands of contiguous spectral bands. It contributes to a wide range of applications, such as environmental monitoring, food analysis, biotechnology, and precision agriculture [2]. Hyperspectral unmixing (HSU) deals with one of the challenging problems in HSIs, *mixed pixels*, in which each pixel contains more than one distinct substance. Generally, HSIs suffer from having a low spatial resolution in spite of high resolution in the spectral domain, and that is why the mixed pixels issue occurs. HSU technique is to parse each pixel spectrum into a set of pure spectra (i.e., *endmembers*) and their corresponding proportions (i.e., *abundance*). Generally speaking, HSU comprises three main parts, namely, estimating the number of endmembers, extracting endmembers, and estimating the corresponding abundances. In this article, the term unmixing depicts the abundance estimation step as the number of endmembers and their signatures are assumed to be known.

Mixed pixels issue can be addressed in three different scenarios, namely, supervised, semisupervised, and unsupervised. In a supervised scenario, which is of concern in this article, endmembers are known *a priori*. They can be extracted from the data empirically [3] or via endmember extraction algorithms, such as vertex component analysis (VCA) [4] and N-FINDR [5], or captured from spectral libraries, e.g., Advanced Spaceborne Thermal Emission Reflection Radiometer (ASTER) and United States Geological Survey (USGS). The aim of HSU in this scenario is to estimate the abundances of the endmembers in each pixel. HSU approaches in an unsupervised scenario, such as [6]–[8], propose to estimate both endmembers and the corresponding abundances simultaneously from HSIs given the number of endmembers. HSU approaches in a semi-supervised scenario like [9] try to determine the optimal subset of the endmembers that suits the data from a given spectral library in advance.

There are two types of mixing models for the HSU: The linear mixing model (LMM) and the families of the nonlinear mixing models (NMMs) [10]. The LMM works on the assumption that the incident light interacts with a single component, and consequently each pixel spectrum is formulated as a linear combination of the endmembers and the corresponding abundances. NMMs are more committed to the actual mechanism of the hyperspectral sensors, albeit with computational cost and need for prior knowledge about the scene. On the contrary, the LMM has been widely utilized thanks to its simplicity and effectiveness in HSU algorithms [11], as adopted in this article.

In recent years, a number of algorithms have been suggested to deal with the linear mixing problem in a supervised scenario. It can be tackled by solving an optimization problem subjected to the physical constraints, referred to as fully constrained least square (FCLS). This algorithm has been modified by introducing a variable splitting and solving the optimization problem with the alternating direction method of multipliers (ADMM) in [12], denoted as SUnSAL and its developed version that employs spatial information via total variation (TV) regularizer in [13], expressed as SUnSAL-TV. Perturbed linear mixing model (PLMM) [14] and augmented linear mixing model

Manuscript received September 8, 2019; revised November 12, 2019 and December 21, 2019; accepted January 1, 2020. Date of publication February 5, 2020; date of current version February 13, 2020. *(Corresponding author: Hassan Ghassemian.)*

The authors are with the Laboratory of Image Processing and Information Analysis, Faculty of Electrical and Computer Engineering, Tarbiat Modares University 14155-4843, Tehran, Iran (e-mail: f.khajehrayeni@modares.ac.ir; ghassemi@modares.ac.ir).

Digital Object Identifier 10.1109/JSTARS.2020.2966512

(ALMM) [15] approaches try to estimate abundances when the hyperspectral imagery endures spectral variability. In the former, the problem is addressed by presenting a novel LMM that possesses an additive perturbation term in each endmember. In the latter, a spectral variability dictionary is introduced in the first step, and then a data-driven learning strategy is adopted to estimate abundances. After modeling the spectral variability, both of these approaches solve their designed optimization algorithm by the ADMM.

With the advent of deep learning, it appears to gain great popularity in state-of-the-art algorithms. Deep learning architectures have been frequently employed in pattern recognition and computer vision domains to enhance the performance compared with the state-of-the-art algorithms [16]. Currently, they are in focus of interest in RS applications [17]. However, by examining the literature, it is proved that they have not been applied in unmixing as other applications, like classification and pansharpening. In order to fill this gap, a new link between deep learning architectures and the HSU in the supervised scenario is established in this article in which a deep convolutional autoencoder (DCAE) network is employed to overcome the mixed pixel issue.

The recent attempts at utilizing neural networks for unmixing purpose are [11], [18]–[25] where the papers [18]–[21] follow the supervised learning approach and the rest follow the unsupervised learning approach. In other words, [18]–[21] address the problem by performing a pixel-based fuzzy classifier and map each pixel vector or its dimensionally reduced version to the corresponding abundances without requiring the endmembers. These papers follow the supervised learning approach, which means both input samples and the corresponding output samples are available, and the network learns the mapping function from the input to the output, which is not true in practical scenarios due to the absence of a so-called groundtruth of abundance maps [20], [26]. Among the remainder, the paper [22] deals with the issue in the supervised scenario, while [11], [23]–[25] tackle it in the unsupervised scenario. In [22], a Hopfield neural network (HNN) is employed whose novelty lies in utilizing a HNN for solving the seminonnegative matrix factorization problem, where both the abundances and the nonlinear coefficients are obtained after the training process. The papers [11], [23]–[25] present autoencoder (AE) networks that sort out the problem in the unsupervised scenario.

In this article, an end-to-end 1-D DCAE and 3-D DCAE are proposed for addressing the mixed pixels problem in a supervised scenario, referred to as pixel-based DCAE and cube-based DCAE, respectively. The proposed DCAEs are able to estimate the abundances given the endmembers with the help of hierarchical features of different depth that are extracted via convolutional neural networks (CNNs) from HSI. In more details, the main contribution is threefold.

(1) To the authors' best knowledge, this is the first attempt at HSU using a DCAE. With this aim, two DCAEs are proposed to extract spectral and spectral–spatial information of HSI. The presented cube-based DCAE is able to learn the spectral–spatial information at high level, which produces better unmixing performance.
(2) Empirically and inspired by the VGG-Net, using very small receptive fields of size (1×3) in the pixel-based DCAE, and $(3 \times 3 \times 3)$ and $(1 \times 1 \times 3)$ in the cube-based DCAE, gives an improved performance among the other possible architectures.
(3) In order to train the network, the spectral information divergence (SID) [27] objective function is employed. Moreover, the dropout technique is utilized to prevent the overfitting that exists in RS domain due to the limited number of training samples.

The rest of this article is organized as follows. Section II presents the problem formulation and the proposed pixel-based and cube-based DCAEs. In Section III, a series of experiments are carried out on a synthetic and two real HSI, and the suggested DCAEs are compared with contemporary methods. Finally, Section IV concludes the article.

II. Proposed Method

In this section, we present the two proposed methods that estimate abundances given the endmembers in HSIs using DCAE networks.

A. Problem Formulation

This section defines the used mixture model and notation. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \Re^{L \times N}$ be the HSI possessing $N = r \times c$ pixels with ith column denoting ith pixel ($\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,L}]^T \in \Re^{L \times 1}$) with L spectral bands, where r and c are, respectively, the number of rows and columns of the HSI in the spatial domain and $i = 1, \ldots, N$; $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p] \in \Re^{L \times p}$ represents the endmember matrix with each column denoting one of the p pure spectra; and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N] \in \Re^{p \times N}$ is the corresponding abundance maps with ith column indicating the abundance vector for ith pixel ($\mathbf{a}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,p}]^T \in \Re^{p \times 1}$). According to the LMM, the data model in the ith pixel can be expressed as

$$\mathbf{x}_i = \sum_{j=1}^{p} a_{i,j}\mathbf{m}_j + \mathbf{n}_i = \mathbf{M}\mathbf{a}_i + \mathbf{n}_i \tag{1}$$

in which $\mathbf{n}_i$ is the additive white Gaussian noise vector. This formula can be expanded for the HSI, as

$$\mathbf{X} = \mathbf{M}\mathbf{A} + \mathbf{N}. \tag{2}$$

The abundance maps in this model are required to be subjected to abundance nonnegativity constraint (ANC), i.e., $a_{i,j} \geq 0$, and abundance sum-to-one constraint (ASC), $\sum_{j=1}^{p} a_{i,j} = 1$, which are two physical constraints. The former demonstrates all elements of abundance maps must be nonnegative, and the latter requires that the sum of abundances in each pixel equals one. In this article, given the pure spectra ($\mathbf{M}$) and the HSI ($\mathbf{X}$), estimating the corresponding abundance maps ($\mathbf{A}$) with a DCAE is of concern. It should be mentioned that in order to overcome the spectral variability, the pure spectra are manually selected form the HSI by visual judgment and by examining the spectral signatures.

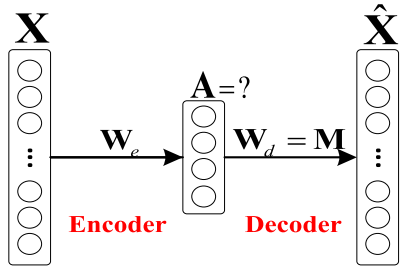

Fig. 1. Graphical representation of HSU using AEs in a supervised scenario.

B. DCAEs

In order to estimate abundances in such a scenario, two DCAEs are constructed. An autoencoder (AE) network is a typical neural network whose aim is to learn compress representation of input data in an unsupervised manner. As shown in Fig. 1, it is usually composed of an *encoder* that extracts the representation and represents it via a *code*, i.e., $\mathbf{A} = f(\mathbf{X})$, and a *decoder* that reconstructs the data from the code, i.e., $\hat{\mathbf{X}} = g(\mathbf{A})$. It simply learns to minimize the loss function, $L(\mathbf{X}, g(f(\mathbf{X})))$. For fully connected autoencoders (FCAEs),

$$f(\mathbf{X}) = \sigma_e(\mathbf{W}_e\mathbf{X}), \qquad g(\mathbf{A}) = \sigma_d(\mathbf{W}_d\mathbf{A}), \tag{3}$$

and for convolutional AEs (CAEs),

$$f(\mathbf{X}) = \sigma_e(\mathbf{X} * \mathbf{W}_e), \qquad g(\mathbf{A}) = \sigma_d(\mathbf{A} * \mathbf{W}_d), \tag{4}$$

in which σ_e and σ_d are, respectively, the element-wise activation functions of the encoder and the decoder, such as rectified linear unit (ReLU) and softmax. Furthermore, $*$ denotes the convolution operator, and $\mathbf{W}_e$ and $\mathbf{W}_d$ are the weights of the encoder and that of the decoder, respectively. Remark that to facilitate reading the bias term were removed in the above equations.

Regarding the convolutional layers, the convolutional operation of a 1-D convolutional layer can be expressed as

$$v_{lf}^{z} = \sigma\left(\sum_{m}\sum_{d=0}^{D_k-1} w_{lfm}^{d} v_{(l-1)m}^{z+d} + b_{lf}\right) \tag{5}$$

where v_{lf}^{z} is the value of a neuron at position z on the fth feature map on the lth layer; m indexes the sets of the feature map in the preceding $(l-1)$ layer; D_k denotes the depth of the kernel; w_{lfm}^{d} stands for the weight at position d connected to the fth feature map; and b and σ are the bias and the activation function, respectively. Moreover, the convolutional operation of a 3-D convolutional layer is

$$v_{lf}^{xyz} = \sigma\left(\sum_{m}\sum_{h=0}^{H_k-1}\sum_{w=0}^{W_k-1}\sum_{d=0}^{D_k-1} w_{lfm}^{hwd} v_{(l-1)m}^{(x+h)(y+w)(z+d)}\right) + b_{lf} \tag{6}$$

in which v_{lf}^{xyz} is the value of a neuron at position (x, y, z) on the fth feature map in the lth layer; H_k, W_k, and D_k denote the height, the width, and the depth of the kernel, respectively.

Typically, in a deep CNN, after each convolutional layer, a pooling layer is used. It helps to decrease the number of network parameters and computation, leading to having a less tendency toward overfitting. In this article, we utilize the max-pooling, being the most popular pooling layer, in designing the pixel-based DCAE that operates independently over each feature map and yields the maximum value in the specific neighborhood.

To prohibit overfitting, it is essential to employ the dropout technique [28] in layers with too many parameters, as utilized in designing the cube-based DCAE. The dropout technique refers to randomly selecting and removing a portion with rate α of neurons during the training process. As a result, the network tends to be less sensitive to particular weights and its generalization capability increases while having a less probability of overfitting.

C. Proposed DCAE Networks

1) Pixel-Based DCAE for HSU: The proposed pixel-based DCAE framework is an end-to-end model that follows the unsupervised learning approach. In other words, the network takes each pixel vector as an input and tries to reconstruct the original input from the abstracted code, which is its abundance vector. The encoder has the task of extracting hierarchical features of different depth of the data through 1-D convolution operators and estimating the corresponding abundance vector (the code) via these features. The abstracted code is then decoded in the original input in the decoder part with the LMM represented in (1). Thus, it is allowed for the encoder part to possess one or more either FC or convolutional layers. The decoder part, however, has to perform as a single FC layer with p input neurons and L output neurons. It acts as the mixture model in (1) where the decoder weights are set equal to the endmember matrix, and consequently, there are no parameters to be adjusted in this layer.

The presented 1-D network, as shown in Fig. 2, is composed of twelve layers in the encoder part and a single layer in the decoder part. Concerning the encoder, it consists of one input layer ($I1$), five 1-D convolutional layers ($C2, C4, C6, C8, C10$), four pooling layers ($P3, P5, P7, P9$), and two FC layers ($F11, F12$). The input of this encoder is a spectrum of each pixels, and it outputs its corresponding abstracted code. Consequently, each pixel vector of size $(1 \times 1 \times L)$ is sent to the encoder and after passing the designed architecture, the corresponding abundance vector is obtained. In more detail, the deep spectral features are extracted via the pixel vector with several very small convolutional kernels of size 1×3. The pooling layers after each convolutional layer have the benefits of reducing the resolution of feature maps which often prevents overfitting. In order to set the number of kernels in each convolutional layer, we adopt the frequent ratio used in the literature ([21], [29]–[31]), which is the number of kernels in each convolutional layer is twice that of the previous convolutional layers.

With regard to the decoder, it possesses a single FC layer ($O11$) with no trainable parameter. The code is entered as its input and according to (1) the original input is constructed. By doing so, the network learns to generate the best corresponding abundance vector for each pixel spectrum and expresses it in the code. Table I tabulates the detailed architecture of the proposed pixel-based DCAE.

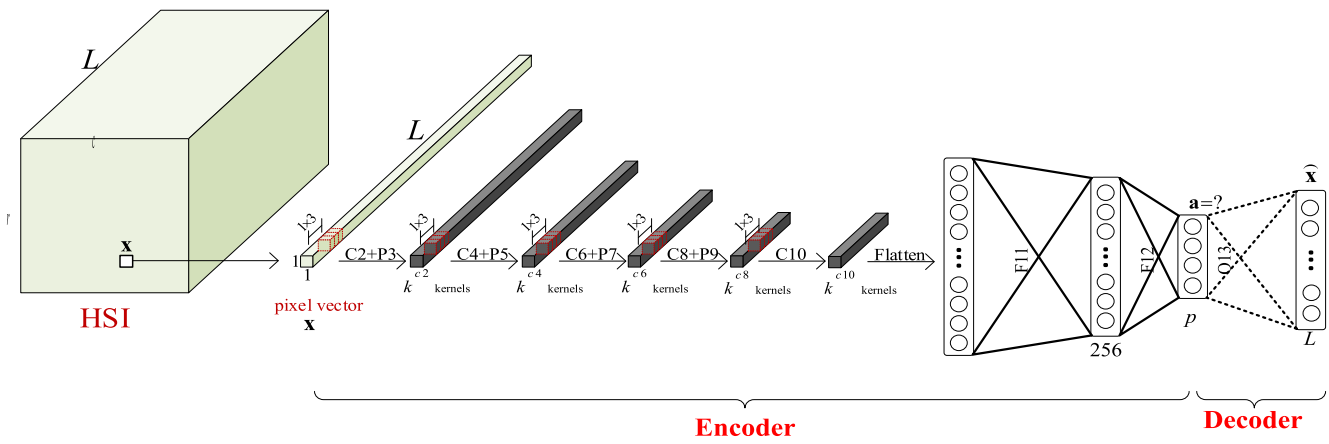

Fig. 2. Architecture of the proposed pixel-based DCAE.

TABLE I
THE ARCHITECTURE OF THE PROPOSED PIXEL-BASED DCAE AND CUBE-BASED DCAE

Proposed algorithm	Layer	Kernel size $(k^c@l^c \times l^c \times q^c)$	Activation	Pooling	Dropout
Pixel-based DCAE $(I_1 : 1 \times 1 \times L)$	$C2$	$2@1 \times 3$	ReLU	1×2	No
	$C4$	$4@1 \times 3$	ReLU	1×2	No
	$C6$	$8@1 \times 3$	ReLU	1×2	No
	$C8$	$16@1 \times 3$	ReLU	1×2	No
	$C10$	$32@1 \times 3$	ReLU	No	No
	$F11$	256	ReLU	-	No
	$F12$	p	ReLU+Softmax	-	No
	$O13$	L	Eq. (5)	-	No
Cube-based DCAE $(I_1 : 5 \times 5 \times L)$	$C2$	$16@3 \times 3 \times 3$	ReLU	No	No
	$C3$	$32@3 \times 3 \times 3$	ReLU	No	No
	$C4$	$64@1 \times 1 \times 3$	ReLU	No	No
	$C5$	$128@1 \times 1 \times 3$	ReLU	No	No
	$F6$	256	ReLU	No	20%
	$F7$	p	ReLU+Softmax	-	No
	$O8$	L	Eq. (5)	-	No

2) Cube-Based DCAE for HSU: The proposed pixel-based DCAE employs the spectral features of each pixel spectrum and estimates the corresponding abundance vector. HSIs possess local spatial information in addition to spectral information, and in this section, we develop the cube-based DCAE that utilizes a 3-D CNN in order to incorporate spectral-spatial information and achieve better unmixing performance. In this way, we adopt a hyperspectral cube of size $(S \times S \times L)$, where S denotes the spatial window size and L is the number of spectral bands, around each pixel in order to estimate its corresponding abundance vector. Therefore, the HSI is, first, split into overlapping 3-D patches. Then, each patch is entered into the encoder of the cube-based DCAE and its spectral–spatial information is extracted via 3-D convolution operators, and its code is obtained. In the further step, the code must be used to reconstruct the original central pixel spectrum in the decoder and consequently the same decoder as the pixel-based DCAE, (1), is employed. It should be mentioned that we replicate the pixels near the borders to be able to form the patches for the border pixels.

As the pixel-based DCAE, the framework of the proposed cube-based DCAE is an end-to-end model that adopts the unsupervised learning approach. It is composed of one input layer (I_1), four 3-D convolutional layers $(C2, C3, C4, C5)$, and two FC layer $(F6, F7)$ regarding the encoder and an single FC layer $(O8)$ in the decoder, which is identical to the pixel-based DCAE decoder. As depicted in Fig. 3, once the patches are extracted, they are entered the first convolutional layer of the encoder that includes k^{c2} kernels of $l^{c2} \times l^{c2} \times q^{c2}$ and a stride of 1 and no padding. This layer produces k^{c2} data cube of size $(S - l^{c2} + 1) \times (S - l^{c2} + 1) \times (L - q^{c2} + 1)$, which are sent to the second convolutional layer with k^{c3} kernels of $l^{c3} \times l^{c3} \times q^{c3}$ and the same stride and padding as the first convolutional layer. The resulting output volume is k^{c3} data cubes with size $(S - l^{c2} - l^{c3} + 2) \times (S - l^{c2} - l^{c3} + 2) \times (L - q^{c2} - q^{c3} + 2)$ and they are sent to the third convolutional layers possessing k^{c4} kernels of $l^{c4} \times l^{c4} \times q^{c4}$. Again, the generated output volumes are sent to the last convolutional layer with k^{c5} kernels of size $l^{c5} \times l^{c5} \times q^{c5}$. These extracted spectral–spatial features are sent to two FC layers $(F6, F7)$ after being flattened. The abstracted code, which is the abundance vector of the central pixel, is obtained from the output of the second FC layer and it is entered into the decoder to restore the original central pixel spectrum. In order to impose the ANC and ASC on the abundance vector of each pixel spectrum, the ReLU and softmax functions are, respectively, applied on the output of $F7$ layer. In the proposed cube-based DCAE, since the input hyperspectral cube empirically is of size $5 \times 5 \times L$, we perform two convolutional kernels with a spatial size of 3×3 which decreases the spatial size of features to 1×1. After that, two other convolutional kernels of size $1 \times 1 \times 3$ are performed in order to achieve deeper spectral-spatial features. Using the ReLU function in CNNs gives better performances in numerous applications. By using both ReLU and dropout in the cube-based DCAE, the outputs of many neurons are 0, which helps us to design a deeper network based on sparse regularization and meets the overfitting issue in hyperspectral images. The more detailed configuration of this network is reported in Table I.

D. Kernel Size

It has been proven in VGG team's paper [29] that using small and fixed convolution kernels (3×3) in all layers for

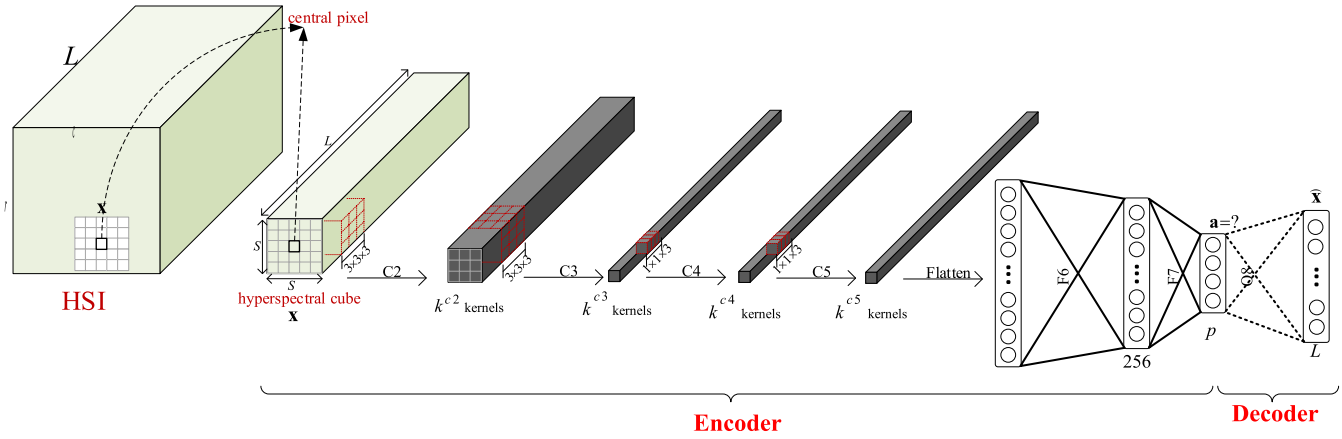

Fig. 3. Architecture of the proposed cube-based DCAE.

2-D CNNs while increasing the CNN depth produces substantial improvements in image recognition. Moreover, the authors in [30] have empirically found that $(3 \times 3 \times 3)$ convolution kernels are more convenient for 3-D CNNs in video analysis task. Inspired by these papers, we utilize only the convolution kernels of (1×3) in the case of the pixel-based DCAE and $(3 \times 3 \times 3)$ or $(1 \times 1 \times 3)$ in the case of the cube-based DCAE.

E. Network Parameters

Having established the network, mini-batch gradient descent strategy by backpropagation algorithm with Adam optimizer [32] is employed to minimize the loss function. Adam is prevalent among deep learning researchers thanks to being based on classical stochastic gradient descent and having both advantages of AdaGrad and RMSProp optimizers. Moreover, we utilized spectral information divergence (SID), introduced in [27], as the loss function, which is

$$L(\mathbf{x}, \hat{\mathbf{x}}) = D(\mathbf{x}||\hat{\mathbf{x}}) + D(\hat{\mathbf{x}}||\mathbf{x})$$

$$D(\mathbf{x}||\hat{\mathbf{x}}) = \frac{1}{m}\sum_{k=1}^{m}\sum_{j=1}^{p} p_{j,k}\log\left(\frac{p_{j,k}}{q_{j,k}}\right)$$

$$D(\hat{\mathbf{x}}||\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m}\sum_{j=1}^{p} q_{j,k}\log\left(\frac{q_{j,k}}{p_{j,k}}\right)$$

$$p_{j,k} = \frac{x_j^{(k)}}{\sum_{k=1}^{m} x_j^{(k)}}, \quad q_{j,k} = \frac{\hat{x}_j^{(k)}}{\sum_{k=1}^{m} \hat{x}_j^{(k)}} \tag{7}$$

in which m indexes the mini-batch size and superscript (k) indicates k*th* mini-batch. This measure, which has been derived from the concept of divergence in information theory, expresses each pixel spectrum as a random variable and enforces minimum dissimilarity between the predicted and true spectra.

III. Experimental Results

In this section, we prove the performance of the presented architectures on a synthetic dataset (presented in [33]) and two widely used real hyperspectral datasets, namely Jasper Ridge, and Urban. In order to have a fair comparison, we make a comparison between the proposed frameworks and some classical and state-of-the-art methods that deal with the mixed pixels in the supervised scenario. They are FCLS, SUnSAL-TV [13], K-Hype [34], Spatial K-Hype [35], ALMM [15], GBM-HNN [22], and FCAE. Besides, in order to highlight the effect of our convolutional approach over deep unmixing approaches in the literature, the results are also compared with DAEN [11].

The K-Hype proposes a novel nonlinear kernel-based mixture model, called K-Hype, and corresponding algorithms to estimate abundances considering only spectral information. Spatial K-Hype is an extension of the K-Hype algorithm that tries to utilize both spectral and spatial information. It integrates spatial information into the nonlinear HSU issue using $L1$ regularization technique. FCAE is an AE network we designed whose encoder part is composed of FC layers and its decoder part is identical to the proposed frameworks. In more detail, each pixel spectrum is encoded to an abstract code via two identical FC layers with p neurons followed by ReLU and softmax layers, respectively.

A. Data Description

1) Synthetic Data: In order to simulate a synthetic data of size $64 \times 64 \times 224$ we followed the procedure in [33]. Thus, five pure spectra were randomly selected from USGS (splib06) spectral library [36]. The data was split into 8×8 blocks, and each block was filled up with one of the pure spectra at random. Then, a spatial low-pass filter with size 9×9 was applied to the data to mix the spectra with LMM. In order to simulate a data with more mixed pixels, the pixels possessing abundances greater than 0.8 were removed and replaced with a mixture of all endmembers with equal abundance fractions $1/p$. Lastly, zero-mean white Gaussian noise was added to each pixel. The synthetic HSI and the selected spectra are shown in Fig. 4.

2) First Real Data (Jasper Ridge): The Jasper Ridge dataset originally possess 512×614 pixels and 224 electromagnetic bands in the range of 380 to 2500 nm. Some of these bands are removed $(1-3, 108-112, 154-166, 220-224)$ due to

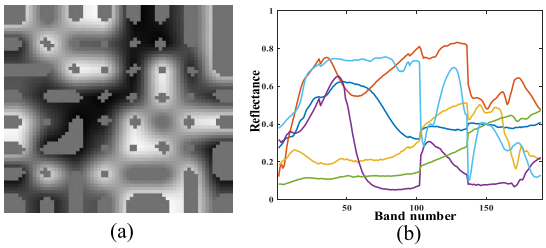

Fig. 4. Synthetic data image (band 100) and the selected endmembers taken from USGS spectral library. (a) Synthetic data. (b) Selected endmembers.

atmospheric effects and being noisy, and 198 bands remain. We utilized a subimage involving 100×100 pixels from (\text{195,209})th to (\text{294,108})th pixel in the original. There are four endmembers in this scene, namely, "#1 Tree," "#2 Water," "#3 Dirt," and "#4 Road".

3) Second Real Data (Urban): The Urban data set is a widely used HSI for HSU research. It contains 307×307 pixels and 210 spectral bands ranging from 400 to 2500 nm. This data also possesses badly degraded bands $(1\text{–}4, 78, 87, 101\text{–}111, 136\text{–}153, 198\text{–}210)$ and after removing them, 162 bands remain. There are four endmembers for this data: "#1 Asphalt," "#2 Grass," "#3 Tree," and "#4 Roof". The more detailed description of these real datasets is found in [37].

B. Parameter Setting

The hyperparameters in the proposed architectures that are required to be set are as follows: The number of layers and the number of neurons that are determined empirically; the Adam optimizer with default parameters, except for the learning rate, produces the best results; the learning rate was set 0.001 for the pixel-based DCAE and 0.0005 for the cube-based DCAE; the batch size was tuned to 100; the dropout rate was chosen 20%; the maximum number of epochs was selected 100; and the spatial window size $(S \times S)$ for the cube-based DCAE was set (5×5). Regarding the FCAE, the Adam optimizer with a learning rate of 0.001 was utilized to optimize the problem; the mini-batch size was set as 100; and the training part lasted 100 epochs.

Considering the fact that the performance of the state-of-the-art approaches depend on their regularization parameters, we empirically found out that the following parameters give the best results: The regularization parameter λ of SUnSAL-TV was set to 0.001 and the regularization parameter of the isotropic TV was 0.003; the parameter μ of K-Hype and spatial correlation η of Spatial K-Hype were set to be 0.002 and 0.5, respectively; K-Hype was run with polynomial kernel; and the parameters α, β, γ, η, and the number of basis vectors L in ALMM were set to be 0.002, 0.002, 0.005, 0.005, and 100.

C. Quantitative and Visual Assessment

The estimated abundances are compared quantitatively and visually with several state-of-the-art methods in the literature. For quantitative assessment, we used abundance overall SID (aSID) as a metric and three criteria (introduced in [15], namely abundance overall root mean square error (aRMSE), reconstruction overall root mean square error (rRMSE), and average spectral angle mapper (aSAM). Once the groundtruth of abundance maps is given, the aSID and aRMSE can be utilized. The rest can be used in case there is no groundtruth of abundance maps and quantifying the performance from data reconstruction perspective is of concern. They are defined by

$$a\text{RMSE} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{p}\sum\nolimits_{j=1}^{p}(a_{i,j}-\hat{a}_{i,j})^2} \tag{8}$$

$$r\text{RMSE} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{L}\sum\nolimits_{l=1}^{L}(x_{i,l}-\hat{x}_{i,l})^2} \tag{9}$$

$$a\text{SAM} = \frac{1}{N}\sum_{i=1}^{N}\arccos\left(\frac{\mathbf{x}_i^T\hat{\mathbf{x}}_i}{\|\mathbf{x}_i\|\,\|\hat{\mathbf{x}}_i\|}\right). \tag{10}$$

1) Results on Synthetic Dataset: In order to validate the robustness of the proposed frameworks, the performances (aRMSE, rRMSE, aSAM, aSID) of each method on synthetic data with different signal-to-noise ratio (SNR) values from 10 to 40 dB were investigated. As illustrated in Fig. 5, the proposed frameworks are more robust to noise with the various SNRs in comparison with the other methods. With regards to aRMSE, the cube-based DCAE achieves by far the better performance, and the pixel-based DCAE obtains the sufficiently lower error. According to rRMSE there is not a significant difference between the proposed architectures and the competitors, but the results of our methods are more promising in terms of aSAM criterion. In aSAM, all of the state-of-the-art methods excluding ALMM DAEN perform approximately the same, and the proposed architectures are superior to the others especially in a scene with a high SNR value. Besides, according to aRMSE and aSID the estimation error of abundances in DAEN is high, but it has an acceptable performance in reconstructing the pixel spectrum.

2) Results on Real Datasets: With respect to the real datasets, given the fact that there is no groundtruth of abundance maps rRMSE and aSAM were employed to assess the simulation results quantitatively. The results are tabulated in Tables II and III where the first and the second best results have been stressed respectively in bold and underline. In addition, in order to make a qualification assessment, the absolute error between the estimated abundance maps and the groundtruth established in [37] achieved by proposed DCAEs and the best of the others as determined visually, as are shown in Figs. 6 and 7. This groundtruth has been utilized in many papers such as [11], [23], [24], and [38] to evaluate the qualitative abundance estimation.

For Jasper Ridge dataset, the quantitative results that show the proposed cube-based DCAE achieves the best performance to the rest of the other methods. It attends the rRMSE value of 1.11×10^{-2} and aSAM value of 4.93×10^{-2}. For this dataset, SUnSAL-TV obtains the second best results based on all criteria and quantitative results of the proposed pixel-based DCAE are somewhat good. The visual inspection of the pixel-based DCAE,

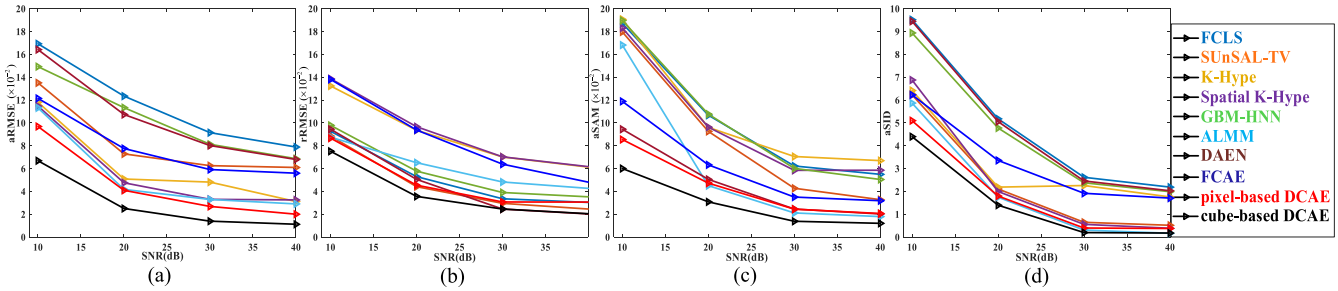

Fig. 5. Robustness evaluation of the synthetic dataset with different SNR values. (a) aRMSE. (b) rRMSE. (c) aSAM. (d) aSID.

TABLE II
SIMULATION RESULTS FOR JASPER DATASET WITH ENDMEMBERS MANUALLY SELECTED ($\times 10^{-2}$)

Algorithms	FCLS	SUnSAL-TV	K-Hype	Spatial K-Hype	ALMM	GBM-HNN	FCAE	Pixel-based DCAE	Cube-based DCAE
rRMSE	2.03	1.29	4.30	4.72	4.99	1.98	3.61	2.89	**1.11**
aSAM	8.03	7.06	19.29	21.85	33.67	7.90	8.60	7.51	**4.93**

TABLE III
SIMULATION RESULTS FOR URBAN DATASET WITH ENDMEMBERS MANUALLY SELECTED ($\times 10^{-2}$)

Algorithms	FCLS	SUnSAL-TV	K-Hype	Spatial K-Hype	ALMM	GBM-HNN	FCAE	Pixel-based DCAE	Cube-based DCAE
rRMSE	5.14	3.17	7.17	6.88	4.62	5.06	6.39	4.33	**2.23**
aSAM	14.99	11.54	16.70	12.84	25.75	14.86	8.84	6.91	**4.98**

however, indicates that it performs better than the competitors in estimating abundance maps. The major difference of proposed DCAEs that is noticed is the edge problem with Dirt material, where incorporating spatial information in the cube-based DCAE leads to a less edge error. It is worth pointing out that DAEN estimates the Water abundance accurately, but it has a disappointing performance in identifying the other materials. It has the rRMSE value of 11.76×10^{-2} and the aSAM value of 10.76×10^{-2}.

Regarding the Urban dataset, once again, the cube-based DCAE leads to the lower reconstruction error according to rRMSE and aSAM metrics with the values of 2.23×10^{-2} and 4.98×10^{-2}, respectively. It also provides the more akin abundance maps to the groundtruth. The aSAM value of the pixel-based DCAE makes it the second-best result with the value of 6.91×10^{-2}, and it attends the rRMSE value of 4.33×10^{-2} which is quite low enough. Among the state-of-the-art methods, the major differences occur in estimating the Asphalt and Roof abundances, where ALMM is the only method gives an acceptable performance. In addition, although the quantitative results of ALMM do not highlight its superior performance, its estimated abundance maps do. It is worth mentioning that DAEN does not have a good performance in terms of abundance estimation on Urban data, but it has the rRMSE value of 5.03×10^{-2} and the aSAM value of 10.89×10^{-2}. It is clearly observed that exploiting the enriched spatial information beside the spectral one results in more accurate abundance maps.

As described above, the results in Tables II and III were obtained using endmembers manually selected form the HSI. In order to provide a fair comparison, VCA was employed to extract the endmembers from each real data, and the evaluation was carried out with these new endmembers. Tables IV and V report the results, where the first and the second best results have been stressed in bold and underline, respectively. As illustrated, there has been a decline in the performance of each method. Nevertheless, the cube-based DCAE still outperforms the others, and the pixel-based DCAE gives the rather great performance. In this case, SUnSAL-TV and ALMM are the two state-of-the-art methods that obtain the best quantitative results.

D. Computation Time

Since having low computational complexity makes a method practically useful, we study the computation time of proposed DCAEs and the other methods on each dataset. As tabulated in Table VI, using the benefits of AEs for spectral unmixing requires too much computation time. However, owing to the neural network architecture of the proposed DCAEs, they have the capability to parallelize on graphical processing units which certainly leads to less computation time.

E. Effects of Parameters

In this section, the robustness of the proposed DCAEs with respect to the learning rate and the spatial window size that mainly influenced the performance are investigated. Regarding the learning rate, when it was between 0.0005 and 0.005, the pixel-based DCAE had a more accurate abundance maps, but its optimal value was found to be 0.001. In the same way, the optimal value of learning in the cube-based DCAE was tuned to 0.0005.

TABLE IV
SIMULATION RESULTS FOR JASPER DATASET WITH ENDMEMBERS GENERATED BY VCA ($\times 10^{-2}$)

Algorithms	FCLS	SUnSAL-TV	K-Hype	Spatial K-Hype	ALMM	GBM-HNN	FCAE	Pixel-based DCAE	Cube-based DCAE
rRMSE	9.32	7.14	17.43	15.86	25.93	9.20	13.31	9.79	**6.83**
aSAM	28.58	20.33	28.47	25.46	79.73	28.28	16.09	14.33	**12.62**

TABLE V
SIMULATION RESULTS FOR URBAN DATASET WITH ENDMEMBERS GENERATED BY VCA ($\times 10^{-2}$)

Algorithms	FCLS	SUnSAL-TV	K-Hype	Spatial K-Hype	ALMM	GBM-HNN	FCAE	Pixel-based DCAE	Cube-based DCAE
rRMSE	8.78	8.32	15.76	15.39	5.18	8.74	15.11	6.78	**4.17**
aSAM	42.05	40.40	35.25	33.19	28.54	41.78	17.00	14.32	**11.14**

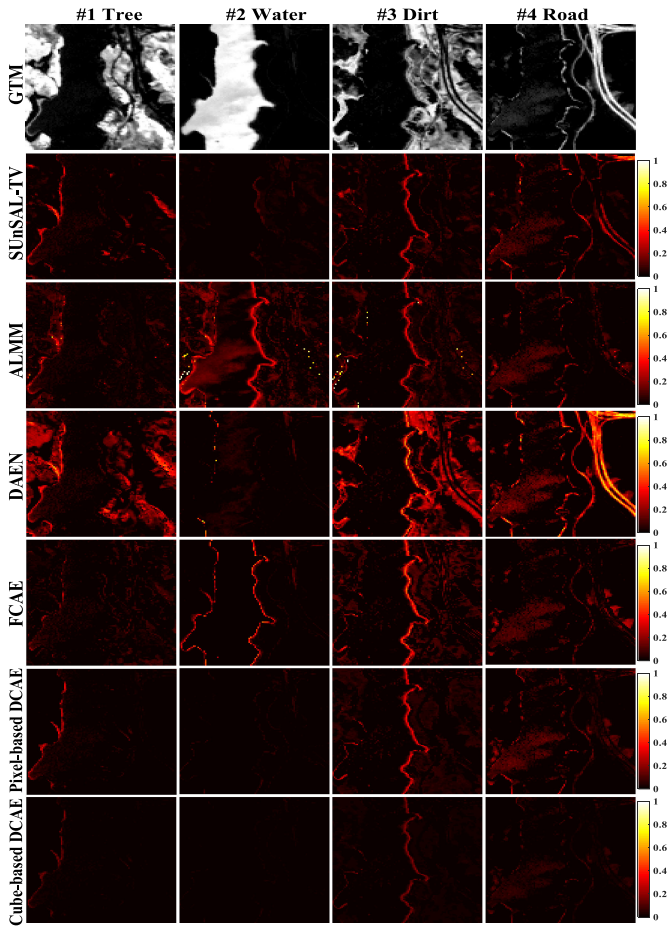

Fig. 6. Groundtruth map (GTM) and the absolute differences with the estimated abundances for Jasper Ridge dataset by different methods.

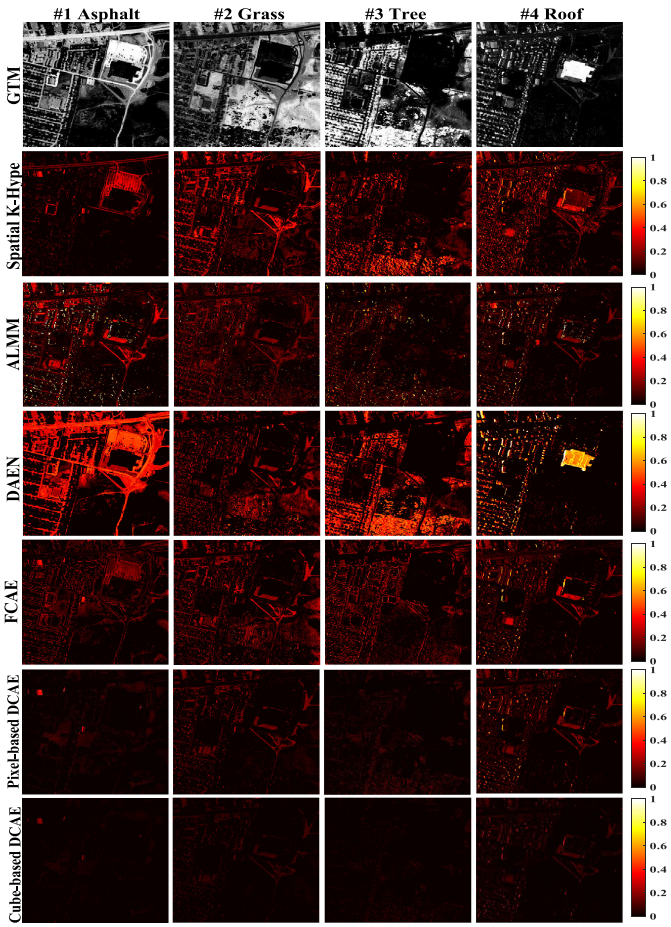

Fig. 7. Groundtruth map (GTM) and the absolute differences with the estimated abundances for Urban dataset by different methods.

TABLE VI
THE AVERAGE COMPUTATION TIME FOR EACH DATASET

Computation Time (sec)			
	Synthetic	Jasper	Urban
ALMM	≈15	≈210	≈1580
GBM-HNN	≈45	≈70	≈390
Autoencoder in [23]	≈225	≈1425	≈7500
DAEN in [11]	≈80	≈165	≈870
FCAE	≈50	≈60	≈360
Pixel-based DCAE	≈90	≈110	≈500
Cube-based DCAE	≈170	≈380	≈1100

In order to find out the optimal size of the spatial window size ($S \times S$), the proposed cube-based DCAE was evaluated with different input sizes: 3×3, 5×5, 7×7, 9×9. Fig. 8 reports the RMSE results for each dataset in the different input sizes. As can be seen, the best estimation was obtained with a size of 5×5, and thus we set the input size to 5×5 for all datasets. Overall, using the spatial information of neighboring pixels could improve the performance, but increasing the spatial input size may have an adverse effect. In other words, the number

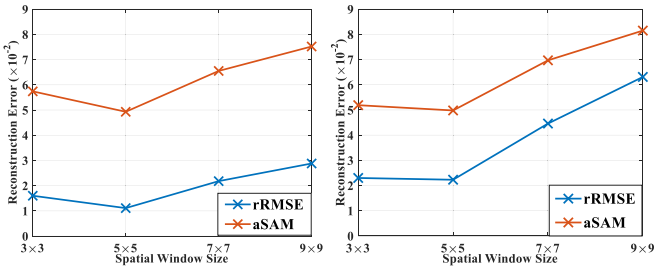

Fig. 8. Robustness evaluation of the proposed cube-based DCAE with the spatial window size. (a) Jasper. (b) Urban.

of the network parameters rises, inducing the overfitting and it may present extra noise especially for the pixels on the borders.

IV. Conclusion

In this article, a pixel-based DCAE and a cube-based DCAE with the aim of abundance estimation in hyperspectral imagery given the endmembers were presented. The proposed pixel-based DCAE utilized a 1-D CNN model to estimate abundances with the assistance of only spectral feature, while the cube-based DCAE employed a 3-D CNN model to integrate the spatial information into the HSU. In addition, inspired by VGG-Net, small and fixed kernels of size 3 were utilized in all layers of the proposed networks. Both quantitative and visual assessments validated the superiority of the proposed networks over several state-of-the-art methods on both synthetic and real datasets. Moreover, the quantitative evaluation and visual inspection proved that exploiting spectral–spatial information leads to a more accurate abundance estimation results.

References

[1] H. Ghassemian, "A review of remote sensing image fusion methods," *Inf. Fusion*, vol. 32, pp. 75–89, 2016.

[2] J. M. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[3] R. Heylen, M. Parente, and P. Gader, "A review of nonlinear hyperspectral unmixing methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, Jun. 2014.

[4] J. M. Nascimento and J. M. 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.

[5] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proc. SPIE*, vol. 3753. Bellingham, WA, USA: SPIE, 1999, pp. 266–276.

[6] F. Kowkabi, H. Ghassemian, and A. Keshavarz, "Hybrid preprocessing algorithm for endmember extraction using clustering, over-segmentation, and local entropy criterion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2940–2949, Jun. 2017.

[7] X.-R. Feng, H.-C. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6245–6257, Oct. 2018.

[8] F. Kowkabi, H. Ghassemian, and A. Keshavarz, "Enhancing hyperspectral endmember extraction using clustering and oversegmentation-based preprocessing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2400–2413, Jun. 2016.

[9] Y. E. Salehani, S. Gazor, and M. Cheriet, "Sparse hyperspectral unmixing via heuristic ℓ_{p}-norm approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1191–1202, Apr. 2018.

[10] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 82–94, Jan. 2014.

[11] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "DAEN: Deep autoencoder networks for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, Jul. 2019.

[12] J. M. Bioucas-Dias and M. A. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Proc. IEEE 2nd Workshop Hyperspectral Image Signal Process.: Evolution Remote Sens.*, 2010, pp. 1–4.

[13] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, Nov. 2012.

[14] P.-A. Thouvenin, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral unmixing with spectral variability using a perturbed linear mixing model," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 525–538, Jan. 2016.

[15] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[17] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[18] G. A. Licciardi and F. Del 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.

[19] Z. Mitraka, F. Del Frate, and F. Carbone, "Nonlinear spectral unmixing of landsat imagery for urban surface cover mapping," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 3340–3350, Jul. 2016.

[20] X. Xu, Z. Shi, and B. Pan, "A supervised abundance estimation method for hyperspectral unmixing," *Remote Sens. Lett.*, vol. 9, no. 4, pp. 383–392, 2018.

[21] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, "Hyperspectral unmixing via deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1755–1759, Nov. 2018.

[22] J. Li, X. Li, B. Huang, and L. Zhao, "Hopfield neural network approach for supervised nonlinear spectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 1002–1006, Jul. 2016.

[23] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25 646–25 656, 2018.

[24] S. Ozkan, B. Kaya, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 482–496, 2019.

[25] Y. Qu and H. Qi, "uDAS: An untied denoising autoencoder with sparsity for spectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1698–1712, Mar. 2019.

[26] A. Plaza, P. Martínez, R. Pérez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 650–663, Mar. 2004.

[27] C.-I. Chang, "Spectral information divergence for hyperspectral image analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 1, 1999, pp. 509–511.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 4489–4497.

[31] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, 2017, Art. no. 67.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, vol. 5, 2015.

[33] R. Rajabi and H. Ghassemian, "Spectral unmixing of hyperspectral imagery using multilayer NMF," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 38–42, Jan. 2015.

[34] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 480–492, Jan. 2013.

[35] J. Chen, C. Richard, and P. Honeine, "Nonlinear estimation of material abundances in hyperspectral images with l1-norm spatial regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2654–2665, Apr. 2014.
[36] R. N. Clark *et al.*, "USGS digital spectral library splib06a," United States Geological Surv., Reston, VA, USA, 2007, vol. 231.
[37] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5412–5427, Dec. 2014.
[38] R. Fernandez-Beltran, A. Plaza, J. Plaza, and F. Pla, "Hyperspectral unmixing based on dual-depth sparse probabilistic latent semantic analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6344–6360, Nov. 2018.

Farshid Khajehrayeni (Student Member, IEEE) received the B.S. degree in engineering science from the University of Tehran (UT), Tehran, Iran, and the M.S. degree in electrical engineering from Tarbiat Modares University (TMU), Tehran, Iran, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree in electrical engineering with TMU.

His research interests include deep learning, signal/image processing, and hyperspectral unmixing.

Hassan Ghassemian (Senior Member, IEEE) received the B.S. degree from Tehran College of Telecommunication, Tehran, Iran, in 1980, and the M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, USA in 1984 and 1988, respectively, all in electrical engineering.

Since 1988, he has been with the Faculty of Computer and Electrical Engineering at Tarbiat Modares University (TMU) in Tehran, Iran, where he is a Professor of Electrical and Computer Engineering. He has published more than 500 technical papers in peer-reviewed journals and conference proceedings. He has trained more than 130 M.S. and 30 Ph.D. students who have assumed key positions in software and computer system design applications related to signal and image processing in the past 30 years. He is the Director of Image Processing and Information Analysis (IPIA) Laboratory. His current research interests include signal/image processing and analysis, multisource information analysis and fusion in remote sensing, and biomedical engineering applications.