A New Deep Convolutional Network for Effective Hyperspectral Unmixing

Xuanwen Tao ⓘ, *Student Member, IEEE*, Mercedes E. Paoletti ⓘ, *Senior Member, IEEE*, Lirong Han ⓘ,
Zhaoyue Wu ⓘ, Peng Ren ⓘ, *Senior Member, IEEE*, Javier Plaza ⓘ, *Senior Member, IEEE*,
Antonio Plaza ⓘ, *Fellow, IEEE*, and Juan M. Haut ⓘ, *Senior Member, IEEE*

Abstract—Hyperspectral unmixing extracts pure spectral constituents (endmembers) and their corresponding abundance fractions from remotely sensed scenes. Most traditional hyperspectral unmixing methods require the results of other endmember extraction algorithms to complete the abundance estimation step. Due to the impressive learning and data fitting capabilities of convolutional neural networks (CNNs), deep learning (DL)-based hyperspectral unmixing technologies have rapidly developed in the literature. According to the procedure used to combine different layers (i.e., fully connected layers, convolution layers, and activation layers), these techniques are mainly divided into three main categories, i.e., those based on autoencoder networks, convolutional neural networks, and convolutional autoencoder networks. They usually extract the weight and output of a specific activation layer as endmember signatures and abundance maps, respectively. Moreover, most existing DL-based unmixing approaches usually use 2-D CNNs to learn the features contained in hyperspectral images, and very few approaches employ 3-D CNNs to extract spectral and spatial information. However, 2-D CNN-based techniques cannot capture good discriminative feature maps from the spectral viewpoint, and 3-D CNN-based techniques usually have high computational overload. In this work, to further exploit the feature extraction capability of CNNs, we combine 3- and 2-D convolutions to propose a cross-convolution unmixing network (CrossCUN) for hyperspectral unmixing. Simultaneously, to better illustrate the improvements of our proposed CrossCUN, we also build the corresponding 2-D convolution unmixing network (2-DCUN) and 3-D convolution unmixing network (3-DCUN). We evaluate the performance of our newly developed networks on two types of synthetic datasets and three real hyperspectral images. Experimental results show that the proposed networks not only obtain better results than other DL-based unmixing methods but also do not require any prior knowledge (e.g., the results of other endmember extraction algorithms) to estimate the abundance maps.

Index Terms—Convolutional neural networks (CNNs), cross convolution, deep learning, hyperspectral unmixing.

I. INTRODUCTION

HYPERSPECTRAL images (HSIs) have been widely studied and applied in different tasks with the aim of processing and analyzing their sheer amount of information. This has been done through various techniques, such as image classification [1], [2], [3], [4], [5], data fusion [6], [7], [8], target detection [9], [10], [11], anomaly detection [12], [13], [14], [15], data denoising [16], [17], [18], and so on. In this context, hyperspectral unmixing [19], [20], [21], [22], [23] is one of the most important applications of HSI data processing. It aims at addressing the problem of low spatial resolution HSIs, which usually contain many mixed pixels. These pixels pose a significant challenge for identifying the materials that compose the considered scenes. To deal with this problem, hyperspectral unmixing techniques decompose mixed pixels into pure materials (endmembers) and their corresponding abundance maps at each pixel. The linear (LMM) [24], [25], [26], [27], [28], [29], [30], [31] and nonlinear mixing model (NLMM) [32], [33], [34] are the two most widely used models for hyperspectral unmixing. The LMM assumes that the observed pixel spectrum is a linear combination of endmember signatures, where the weights conform the corresponding abundance maps that need to be estimated. Due to the simplicity and effectiveness of the LMM, significant efforts to develop hyperspectral unmixing approaches based on the LMM have been conducted in the literature. These hyperspectral unmixing approaches are mainly used to complete two significant tasks, i.e., endmember extraction and abundance estimation.

Traditional endmember extraction methods include vertex component analysis (VCA), automatic target generation process (ATGP), negative abundance-oriented (NABO), and alternating decoupled volume max-min (ADVMM), among others. VCA [26] iteratively projects the HSI data to one direction, which is orthogonal to the subspace formed by the determined endmembers. As a result, the pixel with the extreme projection is extracted as the new endmember. ATGP [29] uses the notion of orthogonal subspace projection to extract endmember signatures. NABO

Xuanwen Tao, Lirong Han, Zhaoyue Wu, Javier Plaza, Antonio Plaza, and Juan M. Haut are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, 10003 Cáceres, Spain (e-mail: taoxuanwenupc@gmail.com; lironghan_upc@163.com; zhaoyue_wu@163.com; jplaza@unex.es; aplaza@unex.es; juanmariohaut@unex.es).

Mercedes E. Paoletti is with the Department of Computer Architecture and Automatics, Complutense University of Madrid, 28040 Madrid, Spain (e-mail: mpaoletti@unex.es).

Peng Ren is with the College of Information and Control Engineering, China University of Petroleum (East China), Qingdao 266580, China (e-mail: pengren@upc.edu.cn).

[30] considers the pixels outside the hull as alternative candidate endmembers to complete the unmixing task. ADVMM [31] addresses the worst case simplex volume maximization problem by alternating optimization. In addition to these approaches, within the current literature, there are also other interesting probabilistic techniques, developed as an attempt to address some of the shortcomings faced by the previous methods [35], [36]. Nevertheless, these methods are computationally very expensive. In contrast, the maximum distance analysis (MDA) [37], [38] has proven to be a simple but effective unmixing method that does not require any prior knowledge about the number of endmembers. However, the above methods both focus on endmember identification and need to be combined with other abundance estimation methods to perform the abundance estimation task.

Traditional abundance estimation methods include fully constrained least squares (FCLS) [39], sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) [40], spatial group sparsity regularized nonnegative matrix factorization (SGSNMF) [41], minimum volume-constrained nonnegative matrix factorization (MVCNMF) [27], minimum-volume enclosing simplex (MVES) [24], and robust collaborative nonnegative matrix factorization (R-CoNMF) [42], among others. SUnSAL utilizes the alternating direction method of multipliers to decompose a difficult problem into a sequence of simpler ones. SGSNMF incorporates the group-structured prior information of hyperspectral images into the nonnegative matrix factorization optimization to complete abundance estimation, where the data are organized into spatial groups. MVCNMF includes a volume constraint into the nonnegative matrix factorization formulation to integrate least-squares analysis and the convex geometry model to perform the unmixing task. MVES incorporates convex analysis and Craig's criterion to enhance the performance of spectral unmixing. It is worth noting that FCLS, SUnSAL, and SGSNMF need the results of endmember extraction from other methods to complete abundance estimation, and MVCNMF and MVES can simultaneously complete endmember extraction and abundance estimation. Compared with the above methods, R-CoNMF can simultaneously complete the three tasks, i.e., determining the number of endmembers, extracting endmember signatures, and estimating the abundance maps. Especially, a different advantage of R-CoNMF compared to other abundance estimation methods is that it completes abundance estimation without any prior knowledge of the number of endmembers.

A. Deep Learning-Based Methods for Hyperspectral Unmixing

Recently, based on their impressive learning and data fitting capabilities, some DL-based unmixing methods have been proposed in the literature [43], [44], [45], [46], [47], [48], [49], [50]. Depending on how different layers are combined, these DL-based unmixing methods are mainly divided into three groups, i.e., autoencoder networks, convolutional neural networks, and convolutional autoencoder networks. These three types of unmixing methods usually extract the endmember signatures and the corresponding abundance maps from the weights and outputs of a specific activation layer, respectively.

Regarding the first group, unmixing methods based on autoencoder networks mainly use fully connected layers and activation layers. In [43], Guo et al. proposed an autoencoder cascade framework that integrates a denoising autoencoder and a nonnegative sparse autoencoder in order to conduct data denoising and endmember estimation, respectively. In this sense, the autoencoder cascade framework is the first attempt to apply DL-based methods to spectral unmixing, improving the performance of unmixing by considering the observation noise and a sparsity prior. Nevertheless, its performance may be affected due to the fact that the same encoder and decoder is used. To overcome this limitation, in [44], Qu et al. proposed a so-called untied denoising autoencoder with sparsity (uDAS) method to address the unsupervised unmixing problem for HSIs with high noise levels. uDAS incorporates denoising capabilities in the form of a constraint to reduce the reconstruction errors and introduces a novel backpropagation method that uses simple matrix operations to effectively force the endmembers to be nonnegative (while the abundance vector is constrained to sum to one). However, both the autoencoder cascade framework and uDAS ignore spatial information and aim at addressing LMM-based problems.

On the other hand, unmixing methods based on CNNs mainly contain convolutional and activation layers. For instance, in [45], Palsson et al. proposed a novel spectral and spatial linear mixture model by developing an associated estimation method based on a convolutional neural network autoencoder unmixing (CNNAEU) to perform the overall unmixing task, i.e., endmember extraction and abundance estimation. CNNAEU extends the LMM to consider the situation where neighboring pixels take part in the pixel reconstruction. As a result, it represents the first attempt to use a CNN to directly exploit the spatial correlation in HSIs for spectral unmixing. Nevertheless, CNNAEU requires a significant number of training samples to adequately fit a large number of learnable parameters. In this sense, the network quickly tends to overfit. In addition, CNNAEU is designed for LMM-based problems. In [46], Gao et al. proposed a cycle-consistency unmixing network (Cycu-Net) which learns two cascaded autoencoders (in an end-to-end manner) to more effectively improve unmixing performance. Cycu-Net designs a cycle-consistency strategy and introduces a new self-perception loss containing two spectral reconstruction terms and an abundance reconstruction term to further refine the unmixing process. However, it ignores the challenges introduced by the high spectral dimensionality of the data.

Finally, unmixing methods based on convolutional autoencoder networks mainly use fully connected layers, convolutional layers, and activation layers. For instance, in [47], Yasiru et al. developed a convolutional autoencoder (CAE) architecture for spectral unmixing. CAE captures the spatial distribution in the original data through convolution filters, which are used to parameterize the spectral features. The encoder and decoder parts of CAE are used to translate the feature space into a latent space representation and to reconstruct the input from the latent space, respectively. However, CAE is based on the LMM and is generally unsuitable for nonlinear unmixing problems. In [48], Qi et al. proposed a deep spectral convolution network with spectral library (SCSL) support to conduct spectral unmixing.

SCNL uses convolutional layers and fully connected layers to capture features and estimate abundance maps, respectively. Moreover, it exploits another fully connected layer to reconstruct the original data. Although SCNL reduces complexity and improves the utilization of the network, it heavily relies on the quality of the spectral library.

B. Contributions of This Work

As mentioned, existing unmixing methods based on DL architectures usually adopt autoencoder networks, CNNs, and convolutional autoencoder networks to perform spectral unmixing. The methods based on autoencoder networks are the most commonly used, whilst those involving CNNs and convolutional autoencoder networks are quite rare. Moreover, most existing unmixing methods usually employ 2-D convolution kernels to extract features, but 3-D CNNs have the ability to capture spectral and spatial features. Nevertheless, only using 2-D-pure CNNs or 3-D-pure CNNs has some disadvantages, such as the inability to capture information on the relationship between the channels or the need to design a very complex model. For instance, by only using 2-D kernels, the model cannot learn good discriminative feature maps from the spectral information. Similarly, a deep 3-D CNN is significantly more complex and expensive in computational terms. At the same time, it can obtain worse results for classes with similar textures on many spectral bands.

To address the aforementioned drawbacks and limitations faced by existing DL-based methods, in this article we combine 3- and 2-D convolution to propose a new cross convolution unmixing network (CrossCUN) for hyperspectral unmixing. The proposed CrossCUN can not only directly complete the abundance estimation task without any prior knowledge about endmember extraction, but also get more accurate results than traditional unmixing methods. Notably, most existing DL-based methods usually use autoencoder networks or 2-D convolutional networks to complete abundance estimation, and our proposed CrossCUN is the first one to exploit cross convolution—that combines 2- and 3-D convolutions—to perform abundance estimation. Our newly proposed CrossCUN shows the potential of cross convolution compared to using 2- or 3-D convolution alone in terms of abundance estimation, and obtains better results than other traditional and DL-based methods.

The rest of this article is organized as follows. Section II introduces the newly proposed CrossCUN and the corresponding 2-D and 3-D convolution networks, i.e., 2-DCUN and 3-DCUN. In Section III, we test the performance of different unmixing methods in abundance estimation on synthetic data and three real HSIs. Finally, Section IV, concludes this article.

II. PROPOSED FRAMEWORK

Our proposed CrossCUN combines 2- and 3-D convolutions and aims at estimating the abundance fraction of each endmember at each pixel in the scene. Let $\mathbf{Y} \in \mathbb{R}^{Nx \times Ny \times Nb}$ denote HSI data, where Nx is the width, Ny is the height, and Nb is the number of spectral bands. Since the 3-D CNN is more computationally complex and hyperspectral pixels contain mixed land cover

classes (with the subsequent inter-class similarity and intra-class variability), we employ principal component analysis (PCA) to remove the spectral redundancy on the proposed CrossCUN. PCA reduces the number of bands from Nb to D, maintaining the same spatial dimensions whilst keeping the most relevant spectral information. In this sense, let $\mathbf{X} \in \mathbb{R}^{Nx \times Ny \times D}$ denote the reduced data by PCA (in our work, D is empirically set to 15).

We build 3-D patches to further improve the performance of 2-DCUN, 3-DCUN, and CrossCUN when dealing with the unmixing task. In this context, let $\mathbf{P} \in \mathbb{R}^{S \times S \times D}$ denote the 3-D patches, whilst $S \times S$ denotes the window size. The number of created 3-D patches by these three networks is $(Nx - S + 1) \times (Ny - S + 1)$. We assume that $P_{a,b}$ is the 3-D patch and (a, b) is its center pixel. In this sense, the width and the height of the 3-D patch $P_{a,b}$ are from $a - (S - 1)/2$ to $a + (S - 1)/2$ and $b - (S - 1)/2$ to $b + (S - 1)/2$, respectively. In our work, S is empirically set to 9.

On the one hand, our proposed 2-DCUN and CrossCUN involve 2-D convolutions, and their input patches are convolved with 2-D kernels. 2-D convolution computes the sum of the dot products between the input patch and the 2-D kernel to obtain the corresponding results, and the kernel spans the input data to cover the entire spatial dimension. The convolved features use an activation function to introduce nonlinearity in the model. In 2-D convolution, the activation value $v_{i,j}^{x,y}$ at spatial position (x, y) in the jth feature map of the ith layer is computed by

$$v_{i,j}^{x,y} = \varphi \left(b_{i,j} + \sum_{\tau=1}^{d_{i-1}} \sum_{\beta=-\rho}^{\rho} \sum_{\alpha=-\gamma}^{\gamma} w_{i,j,\tau}^{\alpha,\beta} \times v_{i-1,\tau}^{x+\alpha,y+\beta} \right) \quad (1)$$

where φ denotes the activation function, $w_{i,j}$ is the value of the weight parameter for the jth feature map of the ith layer, $b_{i,j}$ is the bias for the jth feature map of the ith layer, d_{i-1} is the number of feature maps in the $(i - 1)$th layer, and $2\rho + 1$ and $2\gamma + 1$ are the width and height of kernel, respectively.

On the other hand, 3-DCUN and the proposed CrossCUN first employ 3-D convolution to convolve the input patches with 3-D kernels. Similarly, 3-D convolution also computes the sum of the dot products between the input patch and the 3-D kernel to obtain the corresponding results. We then use an activation function to learn the nonlinear features. In this regard, the activation value $v_{i,j}^{x,y,z}$ at spatial position (x, y, z) in the jth feature map of the ith layer is obtained by

$$v_{i,j}^{x,y,z} = \varphi \left(b_{i,j} + \sum_{\tau=1}^{d_{i-1}} \sum_{\lambda=-\eta}^{\eta} \sum_{\beta=-\rho}^{\rho} \sum_{\alpha=-\gamma}^{\gamma} w_{i,j,\tau}^{\alpha,\beta,\lambda} \times v_{i-1,\tau}^{x+\alpha,y+\beta,z+\lambda} \right)$$
$$(2)$$

where $2\eta + 1$ is the depth of the kernel.

The 2-DCUN, 3-DCUN frameworks and our proposed CrossCUN are graphically illustrated in Figs. 1–3. From Fig. 1, we see that 2-DCUN contains three 2-D convolutional layers, one flatten layer, and one dense layer. The kernel sizes of the three 2-D convolutional layers are 3×3 ($f_1^1 = f_1^2 = 3, f_2^1 = f_2^2 = 3, f_3^1 = f_3^2 = 3$, where f_j^i denotes the kernel size of the ith dimension in the jth convolution), and the number of kernels
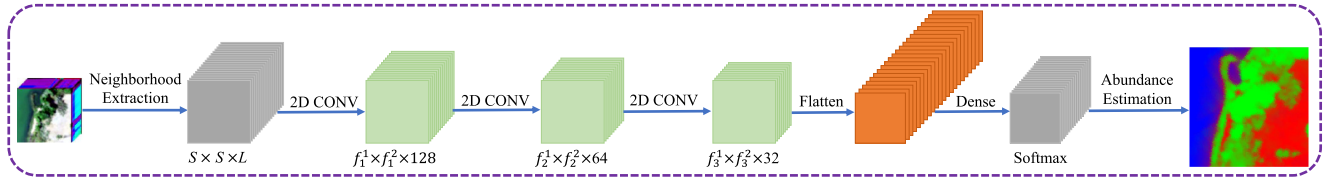
Fig. 1.   Two-dimensional convolution network for spectral unmixing.
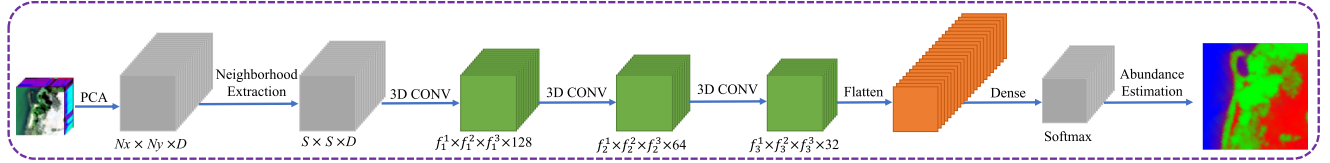

Fig. 2.   Three-dimensional convolution network for spectral unmixing.
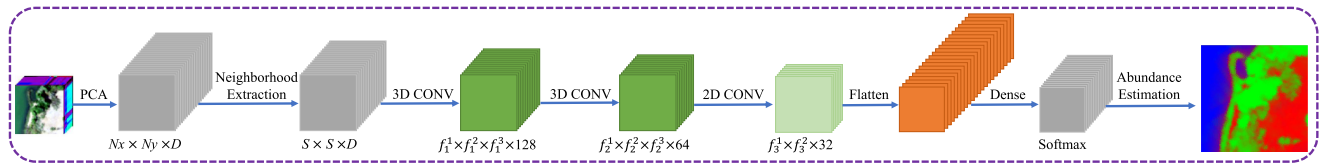

Fig. 3.   Graphical overview of our proposed CrossCUN for spectral unmixing.

(i.e., the depth) of the three 2-D convolutional layers are 128, 64, and 32, respectively. Furthermore, from Fig. 2, we observe that 3-DCUN includes three 3-D convolutional layers, one flatten layer, and one dense layer. The kernel sizes of three 3-D convolutional layers are $3 \times 3 \times 7$ (in particular $f_1^1 = f_1^2 = 3$ and $f_1^3 = 7$), $3 \times 3 \times 5$ (where $f_2^1 = f_2^2 = 3$ and $f_2^3 = 5$), and $3 \times 3 \times 3$ (setting $f_3^1 = f_3^2 = f_3^3 = 3$), respectively. Moreover, the number of kernels of the three 3-D convolutional layers is 128, 64, and 32, respectively. Finally, from Fig. 3, we see that our proposed CrossCUN has two 3-D convolutional layers, one 2-D convolutional layer, one flatten layer, and one dense layer. The kernel sizes of the two 3-D convolutional layers are $3 \times 3 \times 7$ (with $f_1^1 = f_1^2 = 3$ and $f_1^3 = 7$), $3 \times 3 \times 5$ (where $f_2^1 = f_2^2 = 3$ and $f_2^3 = 5$). The kernel size of the 2-D convolutional layer is $f_3^1 = f_3^2 = 3$. Especially, applying a 2-D convolutional layer after the 3-D convolutional layer and before the flatten layer can better distinguish spatial information in different spectral bands without losing a large amount of spectral information, which is very significant for HSI data analysis. Similar to existing unmixing methods, the estimated abundance maps need to satisfy two constraints, i.e., abundance sum-to-one constraint (ASC) and abundance nonnegativity constraint (ANC). We use SoftMax as the activation function of the last dense layer in 2-DCUN, 3-DCUN, and CrossCUN, which is computed by

$$\hat{a}_{ij} = \frac{e^{z_j}}{\sum_{j=1}^{C} e^{z_j}} \qquad (3)$$

where $\hat{a}_{ij}$ is the estimated abundance of the jth endmember on the ith pixel, z_j is the output value of the dense layer, and C is the total number of endmembers. We use cross-entropy as the loss function to train the 2-DCUN, 3-DCUN, and CrossCUN,

TABLE I
CONFIGURATION OF 2-DCUN ON SAMSON DATA

Layer type	Output shape	Parameter
input_1 (InputLayer)	(9,9,156)	0
conv2d_1 (CONV 2-D)	(7,7,128)	179 840
dropout_1 (Dropout)	(7,7,128)	0
conv2d_2 (CONV 2-D)	(5,5,64)	73 792
dropout_2 (Dropout)	(5,5,64)	0
conv2d_3 (CONV 2-D)	(3,3,32)	18 464
dropout_3 (Dropout)	(3,3,32)	0
flatten_1 (Flatten)	288	0
dense_1 (Dense)	3	867

Total trainable parameter: 272,963

which is computed as follows:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [\mathbf{a}_i \log \hat{\mathbf{a}}_i] \qquad (4)$$

where $\hat{\mathbf{a}}_i$ denotes the abundance estimated for the ith pixel, $\mathbf{a}_i$ is the real abundance vector of the ith pixel, and N is the number of pixels. Tables I– III show a detailed description of 2-DCUN, 3-DCUN, and our proposed CrossCUN in terms of layer types, output types, and the number of parameters on a specific HSI dataset (Samson data, described in the following section). From Tables I–III, we can see that 2-DCUN requires the fewest number of parameters, and 3-DCUN and CrossCUN have the same number of parameters. Although CrossCUN does not

TABLE II
CONFIGURATION OF 3-DCUN ON SAMSON DATA

Layer type	Output shape	Parameter
input_1 (InputLayer)	(9,9,13,1)	0
conv3d_1 (CONV 3-D)	(7,7,7,128)	8192
dropout_1 (Dropout)	(7,7,7,128)	0
conv3d_2 (CONV 3-D)	(5,5,3,64)	368 704
dropout_2 (Dropout)	(5,5,3,64)	0
conv3d_3 (CONV 3-D)	(3,3,1,32)	55 328
dropout_3 (Dropout)	(3,3,1,32)	0
flatten_1 (Flatten)	288	0
dense_1 (Dense)	3	867

Total trainable parameter: 433,091

TABLE III
CONFIGURATION OF OUR PROPOSED CROSSCUN ON SAMSON DATA

Layer type	Output shape	Parameter
input_1 (InputLayer)	(9,9,13,1)	0
conv3d_1 (CONV 3-D)	(7,7,7,128)	8192
dropout_1 (Dropout)	(7,7,7,128)	0
conv3d_2 (CONV 3-D)	(5,5,3,64)	368 704
dropout_2 (Dropout)	(5,5,3,64)	0
reshape_1 (Reshape)	(5,5,192)	0
conv2d_1 (CONV 2-D)	(3,3,32)	55 328
dropout_3 (Dropout)	(3,3,32)	0
flatten_1 (Flatten)	288	0
dense_1 (Dense)	3	867

Total trainable parameter: 433,091

need the fewest number of parameters, it obtains more accurate results on experimental data. Moreover, it is worth noting that the number of nodes of the last dense layer in the three networks is 3, which is equal to the number of endmembers in Samson data.

III. EXPERIMENTS

To assess the performance of 2-DCUN, 3-DCUN, and our proposed CrossCUN in terms of abundance estimation, two big families of synthetic data, i.e., with and without pure pixels, and three real HSIs (Samson, Jasper, and Urban scenes) are adopted in our experiments. We use the following:

1) *minimum-volume enclosing simplex* (MVES) [24];
2) *spatial group sparsity regularized nonnegative matrix factorization* (SGSNMF) [41];
3) *robust collaborative nonnegative matrix factorization* (R-CoNMF) [42];
4) *convolutional autoencoder* (CAE) [47];

5) *untied denoising autoencoder with sparsity* (uDAS) as comparison methods to test the performance of different unmixing methods in abundance estimation.

Especially, the initial endmember results required by SGSNMF are extracted by VCA. We employ the root-mean-square error (rmse) and the mean rmse as metrics to test the performance of 2-DCUN, 3-DCUN, and CrossCUN. RMSE is computed by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{a}_{ij} - a_{ij})^2} \tag{5}$$

where N is the number of pixels, and $\hat{a}_{ij}$ and a_{ij} are the estimated and real abundance fractions of the jth endmember on the ith pixel. The mean rmse is computed by

$$\text{Mean RMSE} = \frac{1}{C}\sum_{j=1}^{C}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{a}_{ij} - a_{ij})^2} \tag{6}$$

where C is the number of endmembers.

A. Data Descriptions

1) Synthetic Dataset: To evaluate the impact of pixel purity, the number of endmembers, and the level of noise on abundance estimation, we generate two big families of synthetic data, i.e., with and without pure pixels. Each kind of synthetic data comprises two types: 1) synthetic data with 2500 pixels and 20dB noise with different numbers of endmembers, i.e., 5, 10, 15, and 20; and 2) synthetic data with 2500 pixels and 5 endmembers under different levels of noise, i.e., 10, 20, 30, and 40 dB.

2) Samson Dataset: The Samson dataset was captured by the SAMSON sensor, and it is one of the most widely used datasets for evaluating hyperspectral unmixing algorithms. The scene contains three endmembers, i.e., soil, tree, and water. The original Samson data has 952×952 pixels and 156 bands covering the wavelengths from 401 to 889 nm. In our experiments, to reduce the computational burden, we start from the (252,332)-th pixel and utilize a region of 95×95 pixels to validate the performance of different unmixing methods in estimating abundance maps. The false color composition and the corresponding spectral of Samson are shown in Fig. 4(a) and (d), respectively.

3) Jasper Dataset: The Jasper dataset was obtained by the airborne visible/infrared imaging spectrometer (AVIRIS) of NASA's Jet Propulsion Laboratory (JPL). The original Jasper data contains 512×614 pixels recorded in 224 bands, ranging from 380 to 2500 nm. It has four endmembers, i.e., tree, water, soil, and road. Since the original Jasper data are too complex and will bring high computational overload, we start from the (105 269)th pixel and keep 100×100 pixels in our experiments. In addition, due to dense water vapor and atmospheric artifacts, we remove bands 1–3, 108–112, 154–166, and 220–224 and retain 198 bands in our experiments. The false color composition and the corresponding spectral of Jasper are shown in Fig. 4(b) and (e), respectively.
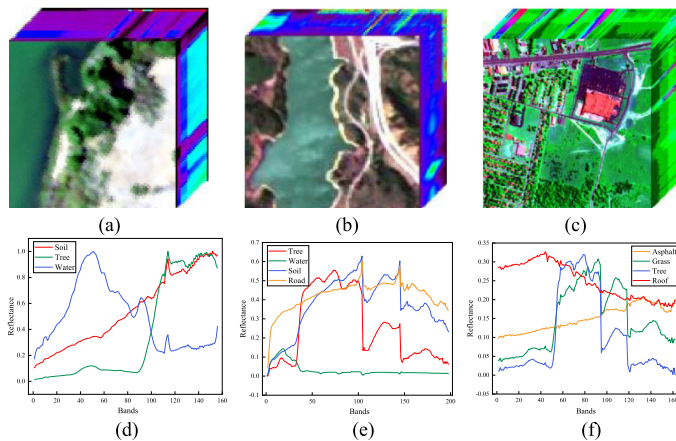
Fig. 4.   Proposed models have been evaluated on three different real hyperspectral scenes. (a) Samson, (b) Jasper, (c) Urban, (d) endmember spectrum of Samson, (e) eEndmember spectrum of Jasper, and (f) endmember spectrum of Urban.

4) Urban Dataset: The Urban dataset contains four different endmembers, i.e., asphalt, grass, tree, and roof. It comprises 307×307 pixels ranging from 400 to 2500 nm. Due to dense water vapor and atmospheric effects, we remove the bands 1–4, 78, 87, 101–111, 136–153, 198–210 and retain 162 bands for our experiments. The false color composition and the corresponding spectral signatures of Urban are shown in Fig. 4(c) and (f), respectively.

B. Implementation Details and Performance Metrics

In our work, we empirically set the number of layers and randomly select 80% pixels and run 50 epochs for training 2-DCUN, 3-DCUN, and our proposed CrossCUN. Especially, for randomly selecting 80% pixels, we first determine the number of pixels (e.g., N) and then randomly generate $N \times 80\%$ non-repeating numbers between 1 and N. Next, we use the $N \times 80\%$ numbers to extract the corresponding data and abundances from the original experimental data and abundances to ensure randomness. Finally, we use the extracted data and abundances to train our proposed network. The parameters that need to be adjusted in our proposed CrossCUN are the learning rate, input size, input dimension, dropout rate, and weight decay. We perform a parameter sensitivity analysis on Samson data and report their results in Fig. 5. From Fig. 5(a), we see that the mean rmse increases as the learning rate grows, and the mean rmse reaches the lowest value when the learning rate is 0.0001. Therefore, we choose 0.0001 as the final training rate in our proposed CrossCUN. From Fig. 5(b), the mean rmse achieves the lowest value when the input size is 9×9. In this sense, we choose 9×9 as the final input size to train our proposed CrossCUN. From Fig. 5(c), we can see that the value of the mean rmse fluctuates as the input dimension increases, and the mean rmse reaches an optimal value when the input dimension is 13. Therefore, we consider 13 as the final input dimension in our proposed CrossCUN. From Fig. 5(d), we also can observe that the mean rmse fluctuates as the dropout rate grows, and reaches the lowest value when the dropout rate is 0.03. Therefore, we choose 0.03 as the final dropout rate to train our proposed CrossCUN. Fig. 5(e)

shows that our method obtains the best results in terms of mean rmse when the weight decay is 0.0001. As a result, 0.0001 is considered the final weight decay in our experiments.

C. Experiments With Synthetic Data

We generate two big families of synthetic data, i.e., with and without pure pixels, to evaluate the performance of different unmixing methods in estimating abundance maps, and retrain the network when synthetic data have different numbers of endmembers or different levels of noise. Especially, considering that synthetic data include many endmembers, we only report the mean rmse results of different unmixing methods on the two big families of synthetic data. Table IV presents the mean rmse results obtained by different unmixing methods on synthetic data with pure pixels under different numbers of endmembers. From Table IV, we see that 2-DCUN, 3-DCUN, and CrossCUN both get better results than other methods and CrossCUN obtains the most accurate results on synthetic data with any number of endmembers. The results shown in Table IV reflect that the number of endmembers almost has no effect on the performance of our proposed CrossCUN in abundance estimation. Table V displays the mean rmse results of different unmixing methods on synthetic data with pure pixels under different levels of noise, i.e., 10, 20, 30, and 40 dB. From Table V, we observe that our proposed CrossCUN obtains the best results on synthetic data with 10 and 20 dB, and R-CoNMF achieves the best results on synthetic data with 30 and 40dB. From Table V, we also see that although our proposed CrossCUN does not achieve the best results on all synthetic data (the gap between it and R-CoNMF is very small). Therefore, our proposed CrossCUN can be used as an effective abundance estimation method.

Table VI shows the mean rmse results obtained by different unmixing methods on synthetic data without pure pixels using different numbers of endmembers. From Table VI, we observe that MVES and R-CoNMF exhibit better results than other methods. Moreover, 2-DCUN, 3-DCUN, and CrossCUN achieve more accurate results than MVES and R-CoNMF, and CrossCUN obtains the best results in terms of abundance estimation on synthetic data with any number of endmembers. Table VII presents the mean rmse results obtained by different abundance estimation methods on the synthetic data without pure pixels under different levels of noise, i.e., 10, 20, 30, and 40 dB. From Table VII, we observe that our proposed CrossCUN is better than other methods on synthetic data with 10 and 20 dB, and R-CoNMF obtains the most accurate results compared with other methods on synthetic data with 30 and 40dB (but the gap between R-CoNMF and our proposed CrossCUN is small). Therefore, our proposed CrossCUN is also very effective in terms of abundance estimation.

D. Experiments With Samson Data

Table VIII quantitatively compares the performance of different unmixing methods in the task of estimating abundance maps on Samson data, reporting the rmse and mean rmse scores obtained by them. From Table VIII, we see that SGSNMF obtains the best results in the task of estimating the abundance

TABLE IV
MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON SYNTHETIC DATA (CONTAINING PURE PIXELS) WITH 2500 PIXELS AND 20 dB NOISE USING DIFFERENT NUMBERS OF ENDMEMBERS

Number of endmembers	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
5	0.0947	0.2239	0.1041	0.2474	0.1471	0.0686	0.0632	**0.0617**
10	0.0652	0.1495	0.0666	0.1268	0.0919	0.0452	0.0403	**0.0400**
15	0.0633	0.0976	0.0900	0.0860	0.1124	0.0436	0.0380	**0.0380**
20	0.0500	0.0878	0.0841	0.0644	0.0894	0.0372	0.0329	**0.0328**

Best results are highlighted in bold.

TABLE V
MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON SYNTHETIC DATA (CONTAINING PURE PIXELS) WITH 2500 PIXELS AND FIVE ENDMEMBERS USING DIFFERENT LEVELS OF NOISE

SNR (dB)	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
10	0.1473	0.1770	0.1337	0.2467	0.1692	0.0675	0.0649	**0.0647**
20	0.0882	0.2453	0.0600	0.2454	0.0676	0.0530	0.0517	**0.0504**
30	0.0499	0.2169	**0.0362**	0.2436	0.1414	0.0648	0.0455	0.0432
40	0.0287	0.2855	**0.0099**	0.2485	0.1268	0.0512	0.0408	0.0391

Best results are highlighted in bold.

TABLE VI
MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON SYNTHETIC DATA (MAXIMUM PURITY OF 0.8) WITH 2500 PIXELS AND 20dB NOISE UNDER DIFFERENT NUMBERS OF ENDMEMBERS

Number of endmembers	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
5	0.0696	0.1953	0.0506	0.2455	0.0601	0.0483	0.0461	**0.0457**
10	0.0774	0.1615	0.0821	0.1285	0.1276	0.0525	0.0468	**0.0464**
15	0.0626	0.1114	0.0853	0.0859	0.0970	0.0427	0.0377	**0.0374**
20	0.0472	0.0732	0.0801	0.0647	0.0895	0.0365	0.0316	**0.0315**

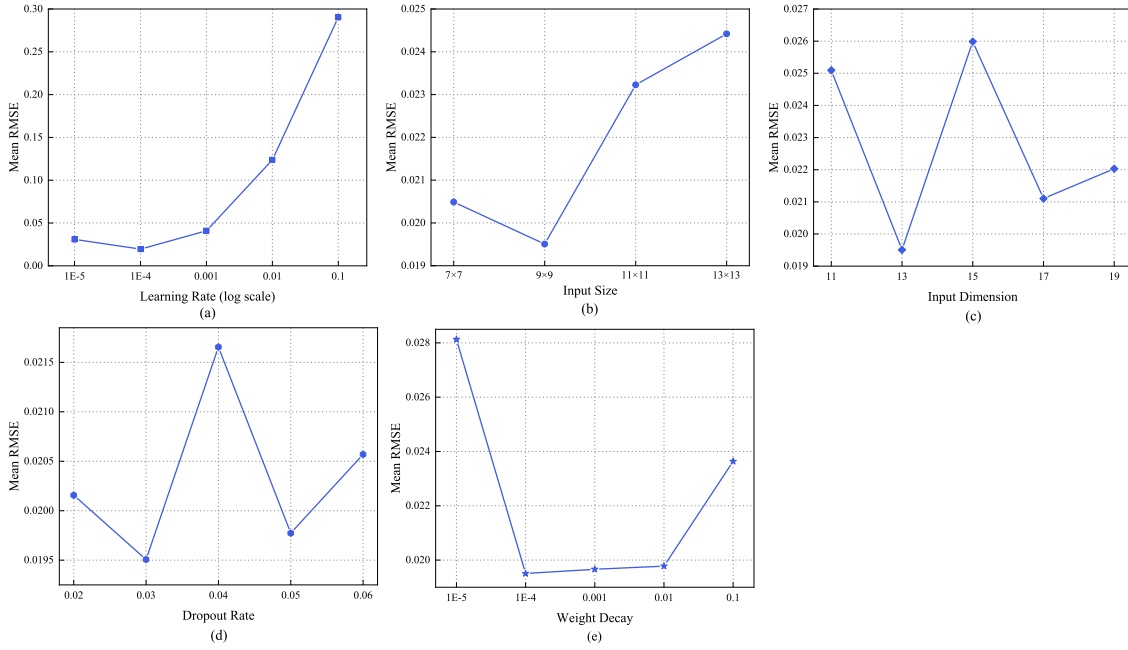Best results are highlighted in bold.

Fig. 5. Quantitative analysis of different (a) learning rates, (b) input sizes, (c) input dimensions, (d) dropout rates, and (e) weight decays for our proposed CrossCUN on Samson dataset.

TABLE VII
MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON SYNTHETIC DATA (MAXIMUM PURITY OF 0.8) WITH 2500 PIXELS AND FIVE ENDMEMBERS
UNDER DIFFERENT LEVELS OF NOISE

SNR (dB)	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
10	0.1335	0.1732	0.1575	0.2480	0.1571	0.0895	0.0783	**0.0761**
20	0.0701	0.2106	0.0547	0.2442	0.0753	0.0498	0.0468	**0.0455**
30	0.0355	0.2179	**0.0208**	0.2481	0.0501	0.0484	0.0418	0.0403
40	0.0063	0.2206	**0.0059**	0.2484	0.0394	0.0447	0.0386	0.0380

Best results are highlighted in bold.

maps of soil, tree, and water compared with the other two traditional unmixing methods, i.e., MVES and R-CoNMF. Moreover, CAE is the best abundance estimation method compared with uDAS. From Table VIII, we also find that 2-DCU, 3-DCUN, and CrossCUN are better than the best traditional and DL-based methods, i.e., SGSNMF and CAE, and our proposed CrossCUN achieves better abundance estimation results compared with 2-DCUN and 3-DCUN. This fact reveals that our proposed CrossCUN is not only very effective for abundance estimation purposes but also improves the results obtained by the corresponding 2- and 3-D convolution networks, i.e., 2- and 3-DCUN.

Fig. 6 presents the abundance maps obtained by different unmixing methods from the Samson data. From Fig. 6, we observe that SGSNMF and CAE provide results that are more similar to the ground-truth compared with other traditional and DL-based unmixing methods. We also find that 2-DCUN, 3-DCUN, and our proposed CrossCUN are closer to the ground-truth compared with all methods, and CrossCUN is slightly better than 2- and 3-DCUN.

E. Experiments With Jasper Data

Table IX shows the results of rmse and mean rmse obtained by different unmixing methods and reports their performance evaluation in abundance estimation on Jasper data. From Table IX, we observe that SGSNMF is the best method in the task of estimating the abundance maps of three, water, soil, and road compared with another two traditional unmixing methods, i.e., MVES and R-CoNMF. CAE obtains the best results in

TABLE VIII
RMSE AND MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON SAMSON DATA

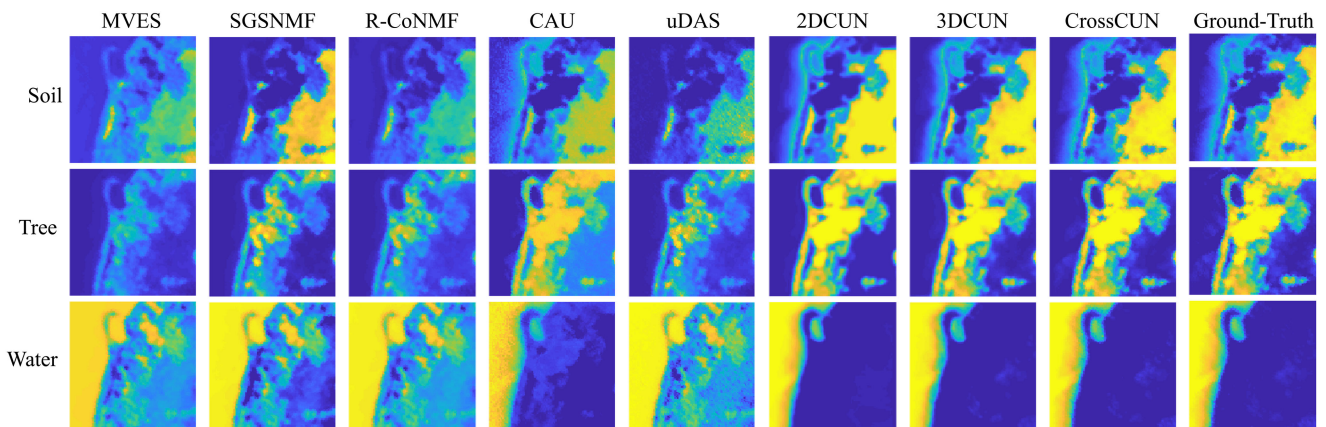| Mineral		Methods							
		MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
RMSE	Soil	0.2326	0.1784	0.2584	0.1156	0.2598	0.0491	0.0354	**0.0234**
	Tree	0.3400	0.2448	0.2729	0.1186	0.2509	0.0489	0.0366	**0.0235**
	Water	0.3916	0.3579	0.4125	0.0667	0.4178	0.0238	0.0254	**0.0116**
Mean RMSE		0.3214	0.2604	0.3146	0.1003	0.3095	0.0406	0.0325	**0.0195**

Best results are highlighted in bold.

Fig. 6. Ground-truth and estimated abundances obtained by different unmixing methods for each endmember material on Samson dataset.

TABLE IX
RMSE AND MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON JASPER DATA

Mineral		Methods							
		MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
RMSE	Tree	0.2156	0.1937	0.1790	0.1311	0.1587	0.0672	0.0385	**0.0292**
	Water	0.2650	0.1264	0.2181	0.1285	0.1918	0.0288	0.0267	**0.0213**
	Soil	0.2194	0.1697	0.1508	0.2225	0.1315	0.0851	0.0437	**0.0320**
	Road	0.1252	0.1719	0.1281	0.2276	0.1150	0.0685	0.0361	**0.0252**
Mean RMSE		0.2063	0.1654	0.1690	0.1774	0.1492	0.0624	0.0363	**0.0269**

Best results are highlighted in bold.

estimating the abundance maps of tree and water, and uDAS obtains the best abundance estimations for soil and road. In fact, uDAS is the best among DL-based methods in terms of the value of the mean rmse. In addition, from Table IX we also see that 2-DCUN, 3-DCUN, and CrossCUN are not only better than the best traditional and DL-based methods, i.e., SGSNMF and uDAS, but also can get quite accurate results when estimating the abundance maps of all minerals. Simultaneously, CrossCUN is better than 2- and 3-DCUN. Therefore, CrossCUN is not only very effective in abundance estimation but also has
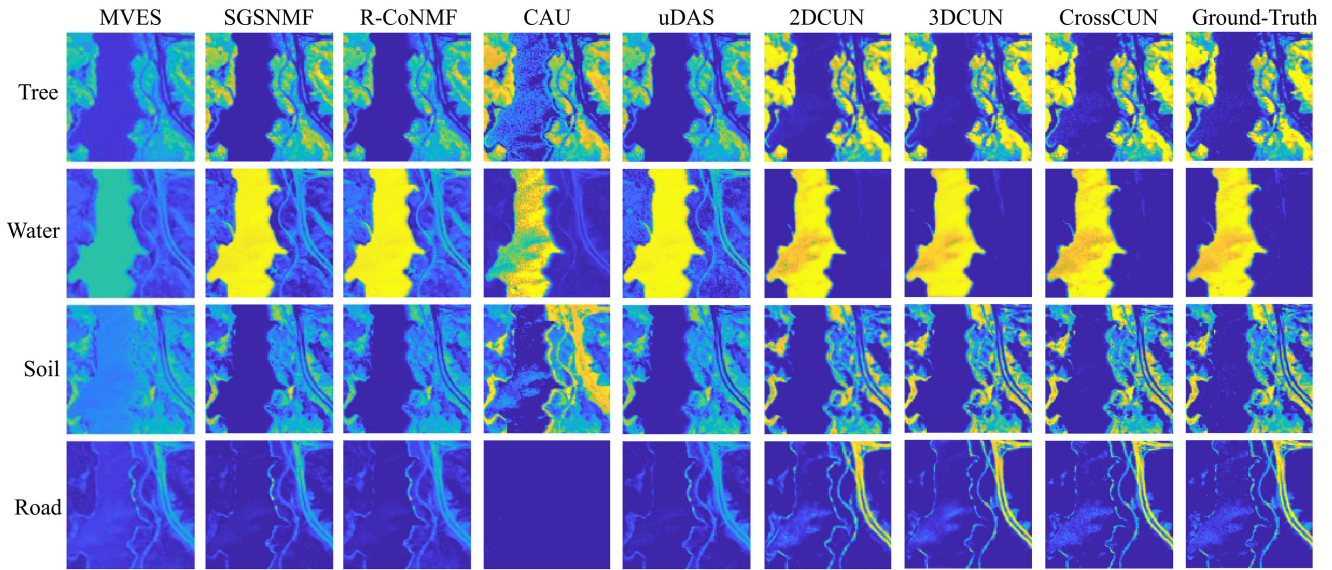
Fig. 7. Ground-truth and estimated abundances obtained by different unmixing methods for each endmember material on Jasper dataset.

TABLE X
RMSE AND MEAN RMSE RESULTS OF DIFFERENT UNMIXING METHODS ON URBAN DATA

Mineral		Methods							
		MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
RMSE	Asphalt	0.2886	0.2870	0.3307	0.1745	0.3280	0.0844	0.0327	**0.0247**
	Grass	0.3341	0.4452	0.4024	0.4409	0.4241	0.0956	0.0390	**0.0293**
	Tree	0.2991	0.2558	0.3045	0.2884	0.3204	0.0725	0.0349	**0.0231**
	Roof	0.1849	0.1899	0.1974	0.2254	0.2089	0.0602	0.0275	**0.0185**
Mean RMSE		0.2767	0.2945	0.3088	0.2823	0.3203	0.0782	0.0335	**0.0239**

Best results are highlighted in bold.

improvements compared with traditional and DL-based methods, and the corresponding 2- and 3-D convolution networks, i.e., 2- and 3-DCUN.

The results obtained by different unmixing methods on Jasper data are presented in Fig. 7. From Fig. 7, we see that MVES and CAE provide the worst results compared with the other traditional and DL-based methods, while 2-DCUN, 3-DCUN, and our proposed CrossCUN are more similar to ground-truth than all other methods. This fact, together with the results in Table IX, leads us to conclude that our proposed CrossCUN is not only an effective abundance estimation method but also outperforms 2- and 3-DCUN.

F. Experiments With Urban Data

Table X shows the rmse and mean rmse values obtained by different unmixing methods and reports their performance evaluation in abundance estimation on Urban data. From Table X, we can conclude that the three traditional unmixing methods, i.e., MVES, SGSNMF, and R-CONMF, and the two DL-based methods i.e., CAE and uDAS cannot obtain ideal results in abundance estimation. Clearly, 2-DCUN, 3-DCUN, and CrossCUN obtain better results in terms of rmse and mean rmse compared with all traditional and DL-based unmixing methods. As in previous experiments, CrossCUN is better than 2- and 3-DCUN.

Fig. 8 reports the abundance maps obtained by different unmixing methods on Urban data. From Fig. 8, we see that 2-DCUN, 3-DCUN, and our proposed CrossCUN are all close to the ground-truth. After analyzing the results in Table X and Fig. 8, we can conclude that CrossCUN not only gets the best results in abundance estimation compared with other traditional and DL-based unmixing methods, but is also more effective than 2- and 3-DCUN.
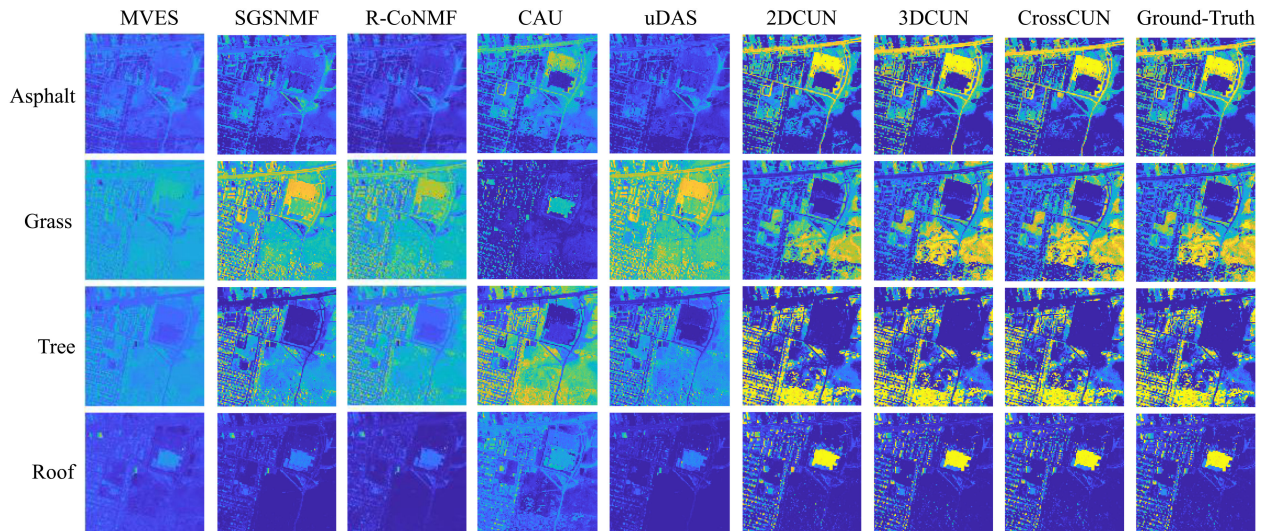
Fig. 8.   Ground-truth and estimated abundances obtained by different unmixing methods for each endmember material on Urban dataset.

TABLE XI

AVERAGE RUNNING TIME (INCLUDING TRAINING AND TESTING TIME) OF DIFFERENT UNMIXING METHODS FOR ALL PIXELS ON SYNTHETIC DATA WITH DIFFERENT NUMBERS OF ENDMEMBERS

Number of	Methods							
endmembers	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
5	**2.1387**	10.4748	4.3838	11.0584	6.2217	15.6695	22.6058	23.2870
10	56.6434	16.3271	**5.7390**	11.0626	27.8004	16.0360	23.0618	23.1512
15	340.6189	20.1648	**6.5037**	11.1522	50.3769	15.9557	23.2579	23.1756
20	676.8515	24.1815	**7.5456**	11.1313	53.6579	16.6114	23.1987	23.5684

Best results are highlighted in bold.

TABLE XII

AVERAGE RUNNING TIME (INCLUDING TRAINING AND TESTING TIME) OF DIFFERENT UNMIXING METHODS FOR ALL PIXELS ON SYNTHETIC DATA WITH DIFFERENT LEVELS OF NOISE

SNR (dB)	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
10	**1.7187**	7.8009	3.9688	10.7874	44.0712	16.6987	22.9547	22.4610
20	**1.7969**	7.2501	4.2816	10.9256	4.0456	15.6064	23.0554	22.7529
30	**1.9807**	9.9005	4.2605	10.7421	21.0370	15.8768	22.9798	22.1205
40	**2.4055**	8.0369	4.7387	10.8124	6.0168	15.5082	23.1463	22.7488

Best results are highlighted in bold.

TABLE XIII
RUNNING TIME (INCLUDING TRAINING AND TESTING TIME) OF DIFFERENT UNMIXING METHODS FOR ALL PIXELS ON DIFFERENT DATASETS

Datasets	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
Samson	**1.3309**	20.9016	3.7967	42.9406	8.4782	45.4194	74.2941	79.8372
Jasper	**1.3629**	26.3976	3.5857	59.8035	174.4122	57.2994	88.4914	83.2869
Urban	**1.5257**	189.6412	30.4376	454.7000	1222.8033	577.2054	791.3201	769.4406

Best results are highlighted in bold.

TABLE XIV
MEAN RMSE RESULTS OF 2-DCUN, 3-DCUN, AND CROSSCUN ON THE CONSIDERED DATASETS USING DIFFERENT PERCENTAGES OF THE AVAILABLE TRAINING SAMPLES

Training samples (%)	Samson			Jasper			Urban		
	2DCUN	3DCUN	CrossCUN	2-DCUN	3-DCUN	CrossCUN	2-DCUN	3-DCUN	CrossCUN
20	0.0558	0.0475	**0.0447**	0.1294	0.0529	**0.0480**	0.0925	0.0498	**0.0417**
40	0.0464	0.0405	**0.0327**	0.1153	0.0398	**0.0385**	0.0860	0.0424	**0.0352**
60	0.0437	0.0360	**0.0262**	0.1248	0.0374	**0.0320**	0.0798	0.0378	**0.0268**
80	0.0406	0.0325	**0.0195**	0.0624	0.0363	**0.0269**	0.0782	0.0335	**0.0239**

Best results are highlighted in bold.

TABLE XV
SUMMARY OF THE MEAN RMSE RESULTS OBTAINED BY DIFFERENT UNMIXING METHODS ON THE CONSIDERED DATASETS (ALL METHODS TRAINED USING 80% OF THE AVAILABLE TRAINING SAMPLES)

Datasets	Methods							
	MVES	SGSNMF	R-CoNMF	CAE	uDAS	2-DCUN	3-DCUN	CrossCUN
Samson	0.3214	0.2604	0.3146	0.1003	0.3095	0.0406	0.0325	**0.0195**
Jasper	0.2063	0.1654	0.1690	0.1774	0.1492	0.0624	0.0363	**0.0269**
Urban	0.2767	0.2945	0.3088	0.2823	0.3203	0.0782	0.0335	**0.0239**

Best results are highlighted in bold.

G. Computational Cost

We conducted all experiments on a computer with 2.6-GHz Intel Core i7 CPU and 16 GB of memory (NVIDIA GeForce RTX 2060 GPU) and evaluated the computational cost of different unmixing methods using different datasets. By analyzing the experimental results on synthetic data, we find that the existence of pure pixels does not significantly affect the efficiency of different unmixing methods. In this sense, we report the results of the average running time (including training and testing time) of different unmixing methods on synthetic data with and without pure pixels. Table XI displays the average running time of different unmixing methods on synthetic data under different numbers of endmembers. From Table XI, we see that MVES achieves the highest efficiency on synthetic data with five endmembers, and the number of endmembers has a greater impact on MVES. In addition, R-CoNMF is the fastest on synthetic data with 10, 15, and 20 endmembers, and it is stable on synthetic data with any number of endmembers. From Table XI, we also find

that 2-DCUN is faster than 3-DCUN and CrossCUN, and the efficiency of 3-DCUN and CrossCUN is similar. Although our proposed CrossCUN is not faster than MVES and R-CoNMF, the time gap between them is not significant. Moreover, the results in Tables IV and VI show that CrossCUN is more effective than MVES and R-CoNMF. Therefore, our proposed CrossCUN is very effective and efficient for abundance estimation. Table XII shows the results of different unmixing methods on synthetic data under different levels of noise, 10, 20, 30, and 40 dB. The experimental results in Table XII indicate that MVES is the fastest method on synthetic data with any level of noise. Moreover, we find that 2-DCUN achieves higher efficiency than 3-DCUN and CrossCUN, and the running time of 3-DCUN and CrossCUN is similar. Although our proposed CrossCUN does not have higher efficiency than MVES, the results in Tables V and VII show that CrossCUN is more effective than MVES. Therefore, our proposed CrossCUN is an effective and efficient abundance estimation method.

Table XIII shows the running time of different unmixing methods on the three real hyperspectral datasets, i.e., Samson, Jasper, and Urban. From Table XIII, we see that MVES is the most efficient among the considered methods. Although MVES is faster than CrossCUN, the results in Tables VIII–X show that CrossCUN is more effective than MVES. Moreover, the computational cost of CrossCUN is acceptable for abundance estimation.

H. Sensitivity to the Number of Training Samples

Although the proposed CrossCUN is a supervised unmixing method, it obtains accurate abundance estimation results without requiring a lot of training samples. To illustrate this point, we choose different training set sizes (i.e., 20%, 40%, 60%, and 80% of the available labeled samples) to evaluate the performance of 2-DCUN, 3-DCUN, and CrossCUN on different datasets. The related mean rmse results are reported in Table XIV. From Table XIV, we see that: 1) more training pixels will lead to more accurate results (as expected); and 2) CrossCUN provides more accurate results than 2- and 3-DCUN with limited training samples. For clarity, we summarize the mean rmse results of different unmixing methods in Table XV (using 80% of the available samples). After analyzing the results of Tables XIV and XV, we can conclude that 2-DCUN, 3-DCUN, and CrossCUN outperform other compared methods even when the training pixels are very few, and CrossCUN obtains the best results compared to all other methods.

IV. CONCLUSION

In this work, a new cross convolution unmixing network (CrossCUN) has been introduced. The proposed CrossCUN can not only get more accurate results than DL-based methods but also directly complete the abundance estimation results without any prior results about endmember extraction, as opposed to traditional unmixing methods. We evaluated the performance of the newly proposed CrossCUN in terms of abundance estimation on synthetic data and three real HSIs, and our experiments reveal that CrossCUN outperforms the competitors in most cases.

As with any new approach, there are some unresolved issues that may present challenges over time. Similar to other DL-based methods, our proposed CrossCUN is a supervised method that requires labeled samples to train the network. Although the process of collecting training data for unmixing applications is costly, this can be done using image data at multiple resolutions or even field visits. In future work, we will work on designing a new unsupervised network to directly complete abundance estimation.

REFERENCES

[1] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.

[2] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 120–147, 2018.

[3] Q. Zhu et al., "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2021.3070577.

[4] K.-K. Huang, C.-X. Ren, H. Liu, Z.-R. Lai, Y.-F. Yu, and D.-Q. Dai, "Hyperspectral image classification via discriminant Gabor ensemble filter," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8352–8365, Aug. 2022.

[5] Z. Gong, W. Hu, X. Du, P. Zhong, and P. Hu, "Deep manifold embedding for hyperspectral image classification," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2021.3069790.

[6] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "CoSpace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.

[7] Y. Xu, Z. Wu, J. Chanussot, P. Comon, and Z. Wei, "Nonlocal coupled tensor CP decomposition for hyperspectral and multispectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 348–362, Jan. 2020.

[8] R. Dian, S. Li, L. Fang, T. Lu, and J. M. Bioucas-Dias, "Nonlocal sparse tensor factorization for semiblind hyperspectral and multispectral image fusion," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4469–4480, Oct. 2020.

[9] H. V. Nguyen, A. Banerjee, and R. Chellappa, "Tracking via object reflectance using a hyperspectral video camera," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 44–51.

[10] Y. Fu, Y. Zheng, I. Sato, and Y. Sato, "Exploiting spectral-spatial correlation for coded hyperspectral image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3727–3736.

[11] J. Zhang, M. Z. A. Bhuiyan, X. Yang, A. K. Singh, D. F. Hsu, and E. Luo, "Trustworthy target tracking with collaborative deep reinforcement learning in EdgeAi-Aided IoT," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1301–1309, Feb. 2022.

[12] H. Su, Z. Wu, Q. Du, and P. Du, "Hyperspectral anomaly detection using collaborative representation with outlier removal," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5029–5038, Dec. 2018.

[13] H. Su, Z. Wu, A.-X. Zhu, and Q. Du, "Low rank and collaborative representation for hyperspectral anomaly detection via robust dictionary construction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 169, pp. 195–211, 2020.

[14] Y. Yuan, D. Ma, and Q. Wang, "Hyperspectral anomaly detection by graph pixel selection," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3123–3134, Dec. 2016.

[15] W. Xie, X. Zhang, Y. Li, J. Lei, J. Li, and Q. Du, "Weakly supervised low-rank representation for hyperspectral anomaly detection," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3889–3900, Aug. 2021.

[16] W. He, Q. Yao, C. Li, N. Yokoya, and Q. Zhao, "Non-local meets global: An integrated paradigm for hyperspectral denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6861–6870.

[17] A. Maffei, J. M. Haut, M. E. Paoletti, J. Plaza, L. Bruzzone, and A. Plaza, "A single model CNN for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2516–2529, Apr. 2020.

[18] Y. Li, X. Luo, N. Wu, and X. Dong, "The application of semisupervised attentional generative adversarial networks in desert seismic data denoising," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[19] B. Pan, Z. Shi, Z. An, Z. Jiang, and Y. Ma, "A novel spectral-unmixing-based green algae area estimation method for GOCI data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 437–449, Feb. 2017.

[20] 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.

[21] X. Xu, B. Pan, Z. Chen, Z. Shi, and T. Li, "Simultaneously multiobjective sparse unmixing and library pruning for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3383–3395, Apr. 2021.

[22] F. Li, S. Zhang, B. Liang, C. Deng, C. Xu, and S. Wang, "Hyperspectral sparse unmixing with spectral-spatial low-rank constraint," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6119–6130, 2021.

[23] B. Rasti and B. Koirala, "SUnCNN: Sparse unmixing using unsupervised convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[24] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418–4432, Nov. 2009.

[25] J. Li, A. Agathos, D. Zaharie, J. M. Bioucas-Dias, A. Plaza, and X. Li, "Minimum volume simplex analysis: A fast algorithm for linear hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 5067–5082, Sep. 2015.

[26] J. M. P. Nascimento and J. M. B. 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.

[27] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765-777, Mar. 2007.

[28] W. Tang, Z. Shi, Y. Wu, and C. Zhang, "Sparse unmixing of hyperspectral data using spectral a priori information," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 770–783, Feb. 2015.

[29] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1232–1249, Oct. 2003.

[30] R. Marrero et al., "A novel negative abundance-oriented hyperspectral unmixing algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3772–3790, Jul. 2015.

[31] T.-H. Chan, J.-Y. Liou, A. Ambikapathi, W.-K. Ma, and C.-Y. Chi, "Fast algorithms for robust hyperspectral endmember extraction based on worst-case simplex volume maximization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 1237–1240.

[32] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4810–4819, Dec. 2015.

[33] 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.

[34] N. Yokoya, J. Chanussot, and A. Iwasaki, "Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1430–1437, Feb. 2014.

[35] 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.

[36] J. A. G. Jaramago, M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, A. Plaza, and J. Plaza, "GPU parallel implementation of dual-depth sparse probabilistic latent semantic analysis for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3156–3167, Sep. 2019.

[37] X. Tao, M. E. Paoletti, J. M. Haut, P. Ren, J. Plaza, and A. Plaza, "Endmember estimation with maximum distance analysis," *Remote Sens.*, vol. 13, no. 4, 2021, Art. no. 713.

[38] X. Tao et al., "Endmember estimation from hyperspectral images using geometric distances," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[39] D. C. Heinz and C. I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

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

[41] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, Nov. 2017.

[42] J. Li, J. M. Bioucas-Dias, A. Plaza, and L. Liu, "Robust collaborative nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6076–6090, Oct. 2016.

[43] R. Guo, W. Wang, and H. Qi, "Hyperspectral image unmixing using autoencoder cascade," in *Proc. 7th Workshop Hyperspectral Image Signal Process. Evol. Remote Sens.*, 2015, pp. 1–4.

[44] 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.

[45] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Convolutional autoencoder for spectral-spatial hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 535–549, Jan. 2021.

[46] L. Gao, Z. Han, D. Hong, B. Zhang, and J. Chanussot, "CyCU-Net: Cycle-consistency unmixing network by learning cascaded autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[47] Y. Ranasinghe et al., "Convolutional autoencoder for blind hyperspectral image unmixing," in *Proc. IEEE 15th Int. Conf. Ind. Inf. Syst.*, 2020, pp. 174–179, doi: 10.1109/ICIIS51140.2020.9342727.

[48] L. Qi, J. Li, Y. Wang, M. Lei, and X. Gao, "Deep spectral convolution network for hyperspectral image unmixing with spectral library," *Signal Process.*, vol. 176, 2020, Art. no. 107672.

[49] D. Hong, J. Chanussot, N. Yokoya, U. Heiden, W. Heldens, and X. X. Zhu, "WU-Net: A. weakly-supervised unmixing network for remotely sensed hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 373–376.

[50] Z. Han, D. Hong, L. Gao, B. Zhang, and J. Chanussot, "Deep half-siamese networks for hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1996–2000, Nov. 2021.