Improved Deep Spectral Convolution Network For Hyperspectral Unmixing With Multinomial Mixture Kernel and Endmember Uncertainty

Savas Ozkan, Member, IEEE, and Gozde Bozdagi Akar, Senior Member, IEEE

Abstract—In this study, we propose a novel framework for hyperspectral unmixing by using an improved deep spectral convolution network (DSCN++) combined with endmember uncertainty. DSCN++ is used to compute high-level representations which are further modeled with Multinomial Mixture Model to estimate abundance maps. In the reconstruction step, a new trainable uncertainty term based on a nonlinear neural network model is introduced to provide robustness to endmember uncertainty. For the optimization of the coefficients of the multinomial model and the uncertainty term, Wasserstein Generative Adversarial Network (WGAN) is exploited to improve stability. Experiments are performed on both real and synthetic datasets. The results validate that the proposed method obtains state-of-the-art hyperspectral unmixing performance particularly on the real datasets compared to the baseline techniques.

Index Terms—Hyperspectral Unmixing, Deep Spectral Convolution Network, Endmember Uncertainty, Multinomial Mixture Model

I. INTRODUCTION

A hyperspectral imagery can provide substantial knowledge about the Earth surface in the form of narrow-band spectra that has been used in the variety of remote sensing applications. Indeed, the separation of constituent materials (i.e., endmembers) and their fractions (i.e., abundances) which contribute to the measured hyperspectral data is a fundamental step since the spectra of the materials can be subjected to complex interactions [1], [2]. As a consequence, they can be mixed in different fractions due to the limitations of hyperspectral sensors and aggravate the problem to proceed with under these scattering effects. Spectral unmixing aims to determine the fractions of the materials blindly from the mixture data that enables to analyze the data more easily [1], [3], [4], [5].

Even if the mixture data can be simplified with a linear formulation (i.e., linear mixing model) [6], in real applications, the model remains weak and underestimates the solution especially for the cases such as atmospheric/illumination conditions and intrinsic material spectra variability. To straighten the model, nonlinear models reformulate the solution by employing nonlinearity and sparsity in their assumptions [6], [7], [8], [9], [10], [11]. In the literature, the bilinear model is a commonly used technique to handle the non-linear scenarios where the bilinear interactions of linear abundances are taken into account in the model. As stated in [6], the methods based on the bilinear model mainly differ with the additional parameter terms and constraints exploited in their formulations.

However, both models assume that a hyperspectral scene is composed of a limited number of fixed endmember spectra. Hence, the performance of the methods for endmembers and their abundances might be limited due to not accounting the possible spectral variability / endmember uncertainty exhibited from the data. Fig. 1 illustrates the spectra of pixels from the Jasper and Urban datasets [12], [13] that are labeled as pure materials. It can be seen that each material can form differently even for the same scene under the assumption of spectral variability. To this end, the conventional methods suffer from the shortcomings of the models (i.e., linear and non-linear models) and the representations are deteriorated by the usage of incorrect endmembers and their combinations. Recently, the methods with angle-based loss functions obtain state-of-the-art performance by which they can partially overcome these limitations due to their robustness to the illumination conditions [2]. Similarly, the idea of stack autoencoders is adapted to learn blindly spectral signatures. This provides robustness to outliers and noise-prone data due to the large set of trainable parameters [14]. Moreover, the methods [2], [15] explain the importance of sparsity for hyperspectral data. [16] proposes a deep convolution network to estimate abundances while using both spectral and spatial neighborhoods. However, due to the lack of examples, the solution is overfitted.

Beside the linear and non-linear models, a large number of techniques incorporates spectral variability in their formulations. In general, the definition of spectral variability assumption is simply a linear/non-linear combination of true endmembers $e_k \in \mathbb{R}^D$ and abundances $y_k \in \mathbb{R}^1, k = 1, ..., K$ for each pixel $x \in \mathbb{R}^D$ such that there is a random uncertainty term $\gamma_k \in \mathbb{R}^D$ which also contributes to the material response [17]:

\[
x = \sum_{k=1}^{K} (e_k + \gamma_k)y_k + \eta, \ s.t. \ y_k \geq 0, \ \sum_{k=1}^{K} y_k = 1 \tag{1}
\]

where $K$ is the number of endmembers exhibited in the scene and $D$ is the spectral bands of a pixel. Also, $\eta \sim \mathcal{N}(0, 1)$ is the additive Gaussian noise to account the possible noise sources. Note that $\gamma_k$ can vary for each pixel and abundances regardless of the scene that practically defines the uncertainty exhibited from data.
In the literature, the studies that intend to be robust to the endmember uncertainty are based on two different sets of assumptions. One of them exploits a library of on-hand material spectra (i.e., the set-based assumption) so as to determine abundances [18], [19], [20], [21], [22] while the other uses the statistical distributions of the materials (i.e., the distribution-based assumption) [23], [24], [25], [17]. For the set-based assumption, the straightforward technique to represent the spectral variability is the multiple endmember spectral mixture analysis (MESMA) [18]. This algorithm intuitively unmixes data iteratively by trying all possible material spectra presented in the library until the error/conditions are completely satisfied or reached to the desired values. The main drawback is that this method is computationally demanding especially when a large body of a spectra library is used. Inevitably, performing a brute-force search for every possible models is intractable. Its variants have an objective to reduce the computation complexity and makes the assumption more tractable. Most of the variants in the literature focus to reduce the library size by selecting the representative spectra from the library [20], [21], [22]. The rest [19] modifies the search algorithm which incrementally determines the possible solutions from the subsets of endmembers so that it moderately reduces the complexity exhausted in the search step.

Similarly, the distribution-based models represent the endmembers as if they are drawn from some distributions so that the flexibility for the spectral variability can be improved. For this purpose, various distribution types can be exploited in the literature to represent the endmembers (i.e., normal composition model (NCM) [23] or beta composition model (BCM) [25]). However, Zhou et. al. [17] states that the true representation of endmembers might not be approximated to an unimodal distribution and the mixture of sub-distributions (i.e., Gaussian Mixture Model (GMM)) can be more convenient to model real data in particular. However, note that the method needs high memory requirements which make the algorithm impractical for any real scene even with the medium sized spatial resolutions. Furthermore, even if the expectation maximization is one of the prominent techniques that has been used to optimize the coefficients of the GMM [26], it suffers from to converge to global solutions and it is not reliable to learn the ideal multinomial distributions as explained in [26], [27], [28], [29], [30].

In this study, we extend the deep spectral convolution network (DSCN) framework which is combined with the endmember uncertainty and the multinomial mixture kernel for hyperspectral unmixing. For this purpose, first, the architecture is modified to reduce the chance of overfitting. Later, the prediction error caused by the spectral variability is alleviated with the multinomial mixture kernel since each abundance map for a material is overly expressed with multinomial distributions correlated to the assumption of spectral variability [17]). Moreover, we redefine the formulation in Eq. 1 with two additional neural network (NN) models (i.e., for reconstruction and uncertainty terms). To this end, hyperspectral data is unmixed and the abundances are estimated from a set of pre-computed endmember spectra in an end-to-end learning scheme by a modular neural network (NN) model.

Fig. 1. Spectra of pixels from the Jasper (first row) and Urban (second row) datasets that correspondence to two different pure materials. It can be seen that hyperspectral data can be highly influenced by the assumption of spectral variability.

In summary, the overall model takes hyperspectral data as input and computes high-level low-dimension representations by using Improved DSCN (DSCN++). Then, these representations are modeled with Multinomial Mixture Model to estimate abundance maps. In the reconstruction step, abundance maps are fed to two NN modules along with the multiplication of endmember signatures. To this end, individual outputs are summed and the input is reconstructed.

The contributions presented in the manuscript can be summarized as follows:

- First, we improve the deep spectral convolution neural network (DSCN) model [31] by introducing critical modifications to the architecture which practically reduce the chance of vanishing gradient problem and enhance the selectivity of the filters. To this end, the trainable parameters tend to converge to a more stable solution which is one of the drawbacks highlighted in the baseline algorithm [31].

- Second, we introduce a multinomial mixture kernel based on a neural network (NN) architecture that estimates the abundances per-pixel by mimicking the mixture of Gaussian distributions. To improve the effectiveness, the latent features obtained from the DSCN are used as the input for the multinomial mixture kernel. In particular, the parameter set used for the kernel is optimized with the Wasserstein GAN [32], [33] which leads to more accurate and stable solutions than the expectation maximization, the Kullback-Leibler (KL) [34] divergence or the mean square/absolute error (ME) [35]. WGAN model mitigates the limitations of EM-like methods such as sensitivity of parameter initialization and continuity that reduce the performance of GMM models. To this end, a flexibility to the model is defined with the robustness to spectral variability.

- Third, we define a new trainable uncertainty term based on a nonlinear NN model which simply provides robustness to the assumption of endmember uncertainty for the model. To do so, the estimated abundances are conditioned with an additional noise distribution as in [32], [33] to synthesize possible uncertainty. Furthermore, the similar architecture is exploited to minimize the residue
Fig. 2. Flow of the proposed method. This model takes hyperspectral data \( x \) as input and computes high-level representations \( z \) by using Improved DSCN. Then, these representations are modeled with Multinomial Mixture Model to estimate abundance maps \( \hat{y} \). In the reconstruction step, abundance maps are fed to two NN modules with the multiplication of endmember signatures. To this end, individual outputs are summed and the input is reconstructed.

estimated from precomputed endmember spectra in the reconstruction step. To this end, this auxiliary model practically helps to converge to a better solution for abundance estimation.

- Note that all these modifications are formulated as a full hyperspectral unmixing pipeline with NN modules and it is optimized by a stochastic gradient-based solver in an end-to-end learning scheme.

The rest of the manuscript is structured as follows. First, we define the concept of the DSCN and explain the modifications presented in the manuscript. Then, the details of the multinomial mixture kernel and endmember uncertainty are described. Later, the optimization step for learning trainable parameters are summarized. Lastly, the experimental results obtained on several real and synthetic datasets are presented and we conclude the manuscript.

II. IMPROVED DEEP SPECTRAL CONVOLUTION NETWORK

In this section, we first provide a basic definition related to the deep spectral convolution networks (DSCNs). Then, the modifications for the architecture as well as the multinomial mixture kernel and endmember uncertainty are explained in detail. Lastly, the optimization step is described. To ease the understandability, the flow of the proposed method is illustrated in Fig. 2. In this figure, \( U_u \) and \( U_r \) correspond to NN modules to capture the random uncertainty and auxiliary terms in the Equation 1 respectively. The further information about these modules will be provided in this section.

A. Baseline Deep Spectral Convolution Network

The proposed method in [31] is equivalent to the standard autoencoder formulation where it initially encodes the input spectra to a hidden latent feature \( \hat{y} = Enc(x; \theta_e) \) and then decodes this latent feature to recompute the reconstructed version of the original spectra \( \hat{x} = Dec(\hat{y}; \theta_d) \). Ultimately, iterative transformations are performed and the trainable parameter sets for both encoder \( \theta_e \) and decoder \( \theta_d \) parts are updated based on the reconstruction error calculated by a loss function(s). To this end, \( \hat{y} \) corresponds to the abundances per-pixel.

In particular, the main objective of the deep spectral convolution networks (DSCNs) is that on the contrary to linear layers [2], it extracts discriminative latent features about data blindly by abstracting the local spectral characteristics. This is essential since the dimensionality of data limits the model to reach an ideal solution by the fact that affine transformations (i.e., corresponds to matrix operations as explained in [21]) learned from unsupervised data might lead to irregularities/holes in the course of partitioning of the feature space. To overcome this limitation, either supervised data [50], [57] or splitting the data into several overlapping/non-overlapping parts [38] is needed to be utilized (the detailed information about this discussion can be found in [20], [31]). Note that the assumption of splitting the data into several overlapping/non-overlapping parts directly shares similar objectives in the DSCNs with an unsupervised setup.

Furthermore, the DSCN architecture is mainly composed of 1D stacked convolutions (i.e., with normalization, non-linear activation and pooling layers) per-pixel to reveal the latent spectral characteristics of data. However, even if state-of-the-art performance is achieved, the ideal parameter solution can be highly influenced by the initialization, thus unstable results and performance variations can be observed in the solution [31]. Furthermore, since deep neural networks can be easily fooled with a random noise as described in [39], the baseline DSCN can be quite sensitive to noise.

For the clarity and understandability of the terminology, we divide the encoder part \( Enc(x; \theta_e) \) into two consecutive steps. First, the proposed DSCN module is applied to extract high-level latent features \( z = H(x; \theta_e) \) (i.e. DSCN++) where \( z \in \mathbb{R}^M \) is the latent feature for each input spectra \( x \). Here, \( M \) denotes the dimension of the latent feature and it can be tuned by the users (It is set to 10 throughout the experiments). Then, these latent features \( z \) are used to estimate the abundances \( \hat{y} = G(z; \theta_e) \) (i.e. Multinomial Mixture Kernel) by exploiting the proposed multinomial mixture kernel for the second step. In the following section, we first describe the architecture details of the improved DSCN.

B. Modifications In the Architecture

As previously explained, the lack of parameter convergence is an issue in order to obtain stable results for the DSCNs. Thus, the architecture should be restructured such that the error calculated from the loss function should be effectively propagated to the earlier layers in the model otherwise the overfitting risk can be multiplied for the architecture. Note that this problem is renowned as the vanishing gradient problem in the literature [37], [40].

For this purpose, initially, the parameter-free ReLU activation (called ReLU) is replaced by leaky rectified units (LReLU) [41], [42] in the model. Precisely, this unit allows small yet non-zero gradients for negative responses of an input even if it is not completely active. Moreover, the influ-
ence/resolution of negative responses is adjusted by a constant value determined as [41].

Second, the normalization layers (i.e. spectral normalization [31] or batch normalization [43]) and spectral convolutions are distinctly reversed to reduce the chance of getting small/no activation(s) calculated from the previous layers. Spectral normalization rescales the input by merely leveraging spectral statistics (i.e., mean and variance of spectral responses) of data than batch characteristics (As explained [44], for noise-prone data, use of batch normalization can saturate the responses due to the extra noise introduced in the learning step). Therefore, even if the normalized spectral values improve the selectivity of the representation particularly for the problem, it has also chance to increase the sparsity with the combination of ReLU [31] (in other words, the chance of vanishing gradient problem). To this end, some of the representative responses can be eliminated intentionally. By this modification, the resolution of the inputs is improved while still preserving the filter selectivity. Furthermore, we observe that after the initial convolution layer, particularly applying batch normalization than spectral normalization leads to more accurate/stable representations since batch normalization can regularize the network as intended in the Dropout layer [40] that reduces the overfitting of the trainable parameters. Lastly, trainable parameters in convolution operators are normalized with L2-norm to eliminate small values.

Moreover, if we consider the first convolution layer of the model as a low-level feature extractor, the consecutive layers target to compute higher/distinct representations about data. To enrich the capacity of the representation, the inception module [45] is adapted. Therefore, multiple filters with different kernel sizes are performed for the same input and the filters can select the best filters by reweighing the responses according to their importance for the data.

In addition, average pooling layers are leveraged to reduce the dimensionality of data at each layer to obtain a low-dimension latent representation for the input spectra. Note that the purpose of using an average operation instead of a max operation is to decrease the chance of overfitting to the constantly responding filters which is a well-known issue in the neural network (NN) community [40]. At the end, the modified architecture for H(x; θ_e) is illustrated in Table I. Note that in the experiments, we observe that performance does not significantly change by parameter sizes in the model.

Moreover, WaveNet model applied for the synthesis of audio signals [46] has a shared notion as in the proposed model and proves the success for 1D signals.

To show the effectiveness of the improved DSCNs, Figure 3 visualizes the data distributions projected to 2D spaces before and after applying the improved DSCN (colors are discriminative based on their abundance values and the PCA is used to reduce the dimensionality). It can be seen that the improved DSCN transforms the representation of data such that the latent features become more representative and the materials can be easily separable from the others.

In the following section, we will explain the multinomial mixture kernel that calculates the abundances per-pixel by exploiting the hidden latent features obtained by the improved DSCN and their distribution characteristics.

### TABLE I

**Architecture of the Improved DSCN. The notations of Conv1D(W, K, S) indicates 1D convolution whose output dimension, kernel size and stride are W, K and S respectively. Furthermore, SN denotes Spectral Normalization, BN indicates Batch Normalization and AvgP(Κ) corresponds to Average Pooling layer with kernel size and stride of Κ.**

| Index | Index of Inputs | Operation(s) | Output Shape |
|-------|-----------------|--------------|--------------|
| (1)   | -               | Input Spectra | D            |
| (2)   | (1)             | Conv1D(10, 21, 1) | D × 10       |
| (3)   | (2)             | LReLU + AvgP(5) + BN | D/5 × 10    |
| (4)   | (3)             | Conv1D(10, 3, 1) | D/5 × 10     |
| (5)   | (3)             | Conv1D(10, 5, 1) | D/5 × 10     |
| (6)   | (3)             | Conv1D(10, 7, 1) | D/5 × 10     |
| (7)   | (4),(5),(6)     | Concat        | D/5 × 30     |
| (8)   | (7)             | LReLU + AvgP(2) + SN | D/10 × 10   |
| (9)   | (8)             | Conv1D(10, 3, 1) | D/10 × 10    |
| (10)  | (9)             | LReLU + AvgP(2) + SN | D/20 × 10   |
| (11)  | (10)            | Linear(M) + LReLU | M            |

**Fig. 3.** The data distributions before (first column) and after (second column) applying the improved DSCN to the input spectra for the Jasper (first row), Urban (second row) and Synthetic (third row) datasets. As seen, the latent feature space can be more separable than the original material spectra.

#### C. Multinomial Mixture Kernel For Abundance Maps

For the assumption of endmember uncertainty, compositional models sampled from an unimodal distribution estimate the distributions as follows:

\[
p(z|θ_k) = \mathcal{N}(z|μ_k, Σ_k)
\]

(2)

Here, \( θ_k \) denotes the parameter set of a Gaussian distribution that is composed of mean \( μ_k \in \mathbb{R}^M \) and covariance matrix \( Σ_k \in \mathbb{R}^{M \times M} \) where \( k = 1, ..., K \).

As indicated, these kernels can be misleading since the data cannot be directly approximated to an unimodal distribution [17] and multinomial mixture models need to be utilized to model the data properly.
For this purpose, Eq. [2] is generalized with Gaussian Mixture Model (GMM) by computing the distribution \( p(z|\Theta_k) \) from the statistical perspective:

\[
p(z|\Theta_k) = \sum_{n \in K} p(n)p(z|\Theta_k, n) \tag{3}
\]

Here, \( n \) is a random variable whose sample space is \( K \). Moreover, the probabilistic values correspond to \( p(n) = \pi_{n,k} \) and \( p(z|\Theta_k, n) = \mathcal{N}(z|\mu_n, \Sigma_n) \), \( n = 1, \ldots, N \). Hence, the model formulates the estimation of abundances such that each latent feature \( z \) of a pixel for the \( k^{th} \) endmember \( e_k \) can be represented with the mixture of multivariate Gaussian distributions as follows:

\[
p(z|\Theta_k) = \sum_{n=1}^{N} \pi_{k,n} \mathcal{N}(z|\mu_n, \Sigma_n), \pi_{k,n} \geq 0, \sum_{n=1}^{N} \pi_{k,n} = 1. \tag{4}
\]

where \( N \) is the number of components in the GMM and \( N \geq K \) assures that endmembers are overly represented with a mixture of Gaussian components that practically provides robustness to the endmember uncertainty. Moreover, \( \pi_{k,n} \) is the weight term which determines the influence of the \( n^{th} \) GMM component to the \( k^{th} \) abundance value.

Inspired by the multinomial distribution-based methods, we claim that the distributions (i.e. components) can be imposed on the abundances such that each abundance for a material can be overly expressed with the distributions which is ultimately related to the idea of spectral variability. For this purpose, a novel NN kernel is structured such that it especially computes the material abundances from the latent features \( z \) (i.e. Multinomial Mixture Kernel).

For our model, we mimic the GMM with a neural network (NN) module and a novel multinomial mixture kernel is proposed. Note that this model is trainable in an end-to-end manner and necessary parameters in the multinomial mixture kernel can be learned by error backpropagation.

In particular, the distance between the latent representation \( z \) and \( n^{th} \) distribution in Eq. [3] can be simply calculated by the multivariate mahalanobis distance:

\[
d_n = (\mu_n - z)^T \Sigma_n^{-1} (\mu_n - z). \tag{5}
\]

By expanding the formulation in Eq. [5] it can be roughly rewritten as \( d_n = z^T \Sigma_n^{-1} z - 2\mu_n^T \Sigma_n^{-1} z + \mu_n^T \Sigma_n^{-1} \mu_n \). Note that the formulation can be simplified with a linear neural network layer by losing the form of quadratic function (i.e., with trainable weights \( w_n \) and bias \( b_n \)) as \( d_n = w_n^T z + b_n \) where \( w_n^T z = z^T \Sigma_n^{-1} z - 2\mu_n^T \Sigma_n^{-1} z + \mu_n^T \Sigma_n^{-1} \mu_n \). Furthermore, the distances for all endmembers is normalized with a sigmoid activation to produce a probabilistic relation (Please note that softmax distribution also works seamlessly/similarly for this model). To this end, the normal distribution is summarized by a cumulative distribution function as follows:

\[
\mathcal{N}(z|\mu_n, \sigma_n) = \frac{1}{1 + e^{-w_n^T z + b_n}}. \tag{6}
\]

We should note that in the implementation of the kernel, we tested to utilize a model that directly uses \( \mu_n \) and \( \Sigma_n \) coefficients (i.e. as a quadratic form) rather than parameterizing them with a linear NN layer as described in the previous paragraph. However, we observed that even if this model yields stable results (i.e., small variations), the overall accuracy becomes low. As explained in [47], this issue can be summarized by the fact that parameterizing the formulation with a linear NN layer makes the operations differentiable which enables to obtain better performance. However, the capacity of representing true relations (i.e., covariance matrix) is also underestimated. Therefore, parameterized coefficients are used.

Similarly, the weight term \( \pi_{k,n} \) is calculated by a NN model and the sum-to-one constraint is supplied with a softmax activation:

\[
\pi_{k,n} = \frac{e^{w_{k,n}^T z + b_{k,n}^T}}{\sum_{n=1}^{N} e^{w_{k,n}^T z + b_{k,n}^T}}. \tag{7}
\]

To this end, an abundance per-material is modeled with Eq. [5] by replacing the original components with the differentiable NN layers formulated as in Eq. [6] and Eq. [7]. To assure the sum-to-one constraint for abundances of each pixel, L1-norm is applied at the end [48].

Last but not least, note that the proposed kernel decreases the drawback of NNs to noise, since multinomial mixture models can effectively capture the uncertainty in the data so that the trainable parameters yield more stable results as stated in [49], [50].

D. Spectral Variability Assumption

In Eq. [1] the formulation encapsulating spectral variability has an extra term \( \gamma_k \) compared to the linear/nonlinear models which defines the uncertainty per-pixel subjected to the abundances. In particular, this term should be also modeled with an additional parameter set in the model in order to obtain state-of-the-art performance, since it captures the possible deformations of spectra due to the spectral variability which cannot be incorporated by the multinomial mixture kernel.

In the formulation, by simply expanding Eq. [1] it is easy to see that there are two sets of equations where \( e_k \hat{y}_k \) denotes the projection of endmember spectra \( e_k \) (i.e., precomputed material spectra) with the estimated abundances \( \hat{y}_k \) while the other term corresponds to \( \gamma_k \hat{y}_k \) that defines the influence of the uncertainty depending on the abundance. For this purpose, we introduce a trainable NN module \( U_u(\tilde{y}, \eta; \theta_u) \) which takes the estimated abundances \( \tilde{y} \) and Gaussian noise \( \eta \sim \mathcal{N}(0,1) \) as input and generates an additive nonlinear spectra that degrades the reconstructed spectra based on the uncertainty assumption. More precisely, the uncertainty module \( U_u(.) \) is composed of two consecutive nonlinear NN layers that incrementally increases the dimension of spectra (i.e., \( K + L \to 20 \to D \)) by exploiting the abundance values and Gaussian noise where \( L \) is the dimension of the noise. In the last layer, the responses are rescaled with TanH activation as well as a trainable coefficient \( \alpha_u \in [0, 0.1] \) to limit the influence.

Furthermore, since the performance of hyperspectral unmixing is highly influenced by the precomputed material spectra \( \hat{e}_k \) (i.e., it can be computed from various endmember extraction techniques), the representation of endmember spectra can be also improved by the model for better abundance
results. Rather than tuning the precomputed material spectra by backpropagation, we observed that introducing a new trainable term which refines the material spectra can yield better performance. Mainly, this stems by the fact that the alteration of material spectra allows one of the crucial information about data to be lost even if the endmembers are blindly estimated. Therefore, this enables us to preserve endmember information by enhancing their weakness with an extra term. For this purpose, the true material spectra is decomposed into \( \mathbf{e}_k = \mathbf{e}_k + \mathbf{r}_k \) where \( \mathbf{e}_k \) is the precomputed material spectra and \( \mathbf{r}_k \) is the possible residue obtained by the endmember method. Similarly, this residue term can be modeled by exploiting the similar architecture as in the uncertainty with no shared parameters as \( \mathbf{r}_k = U_r(\mathbf{\hat{y}}; \mathbf{\theta}_r) \) and \( \alpha_r \in [0, 0.05] \). Hence, Eq. [1] in the manuscript that corresponds to \( \mathbf{\hat{x}} = Dec(\mathbf{\hat{y}}; \mathbf{\theta}_d) \) can be rewritten as follows:

\[
\mathbf{\hat{x}} = \sum_{k=1}^{K} \mathbf{\hat{e}}_k \mathbf{\hat{y}}_k + \alpha_r U_r(\mathbf{\hat{y}}; \mathbf{\theta}_r) + \alpha_u U_u(\mathbf{\hat{y}}, \mathbf{\eta}; \mathbf{\theta}_u) \quad (8)
\]

In particular, parameters that provide robustness to the endmember uncertainty need to be optimized with a novel scheme in order to obtain state-of-the-art performance.

### E. Parameter Learning

To minimize the reconstruction error, we exploit the optimization scheme proposed in [2]. For this purpose, the Spectral Angle Distance \( C(x, \mathbf{\hat{x}}) \) between the input spectra \( x \) and the reconstructed spectra \( \mathbf{\hat{x}} \) is dominantly maximized with the KL-divergence. Note that the similar notion (i.e. maximize the cross entropy) is used in GAN optimization:

\[
\mathcal{L}_{rec} = -E[KL(1.0||C(x, \mathbf{\hat{x}}))] + \lambda_0E[||x - \mathbf{\hat{x}}||_1]
+ \lambda_1||\mathbf{\hat{y}}||_1 + \lambda_2||\mathbf{\theta}_r||_2 \quad (9)
\]

where \( \lambda_1 \) determines the influence of the \( l_1 \) regularization term (i.e., sparsity) and \( \lambda_2 \) is used for regularizing the parameters of the encoder module (except \( \mathbf{w}^1 \), \( \mathbf{b}^1 \)) with \( l_2 \) norm. In this work, \( \lambda_1 \) and \( \lambda_2 \) are set to 0.4 and \( 10^{-5} \) respectively. Furthermore, \( \lambda_0 \) defines the ratio of the mean square/absolute error (ME) contributed to the reconstruction error. Note that ME is particularly critical for synthetic datasets since the angle distinctiveness is quite small compared to real data (we will explain this statement in the experiment section in detail). Hence, in the default setting, \( \lambda_0 \) is set to 0. Lastly, \( E[\cdot] \) is the expectation value of the possible values for the mini-batch.

In particular, to estimate the GMM parameters, the expectation maximization (EM) is the most popular technique in the literature and it theoretically guarantees to obtain a stationary solution by minimizing the negative log-likelihood function. However, the log-likelihood function highly suffers from to converge to a global solution by random parameter initialization and this assumption is highlighted in various studies in the literature. As explained in [26], [27], [28], [29], [30], this stems from the fact that solving negative log-likelihood function is similar to employ the Kullback-Leibler (KL)-divergence and it has weakness that can be summarized as twofold. First, since GMM is continuous and locally Lipschitz [27], the optimization method should be also continuous and differentiable where this cannot be completely supplied by the KL-divergence. Second, the KL-divergence suffers from the local minimum due to the discontinuity of the function which becomes highly sensitive to the choice of initial parameters. Further discussions related to the KL-divergence can be found in [27].

Similarly, for various studies regardless of their applications, the shortcomings of ME for the manifold learning are also stated [35]. In short, the ME-based solutions yield overly smoothed results due to the averaging of all possible solutions in the manifold space so that the resultant outputs can be underestimated. Possible solutions for an inverse problem can be exponentially multiplied if only pixel-wise relations are used. Note that this assumption is built for the assessment of image quality [51], [52] and due to its weakness, ME cannot preserve the structural similarity of data.

As stated, the loss function utilized in our method is mainly composed of the KL-divergence and the ME by minimizing the reconstruction error (The details of the loss functions will be provided in the following section.). Inevitably, the drawbacks of the techniques as we previously discussed affect the performance of the multinomial mixture kernel as well as the proposed method if these techniques are merely used in the formulation. For this purpose, we exploit the Wasserstein GAN [32], [33] to optimize the trainable parameters especially for the mixture model and the uncertainty term that implicitly correspond to \( \mathbf{w}^1 \), \( \mathbf{b}^1 \) and \( \mathbf{\theta}_u \).

Briefly, generative adversarial networks (GANs) have shown impressive performance for learning data distributions in various machine learning (ML) problems by their ability to transform a known distribution \( p \) to an unknown data distribution \( q \). However, the original GANs are potentially not continuous to minimize the error as explained in [32] and the similar limitations can be observed as if the KL-divergence is used (i.e., discontinuity). Therefore, the improved Wasserstein GAN is exploited in the proposed model which has several theoretical benefits over the original GAN architecture [32], [33], [34], [35].

Formally, Wasserstein distance \( W(p, q) \) is constructed to measure the distance between \( p \) and \( q \) by using the Kantorovich-Rubinstein duality [55]. Note that the Wasserstein distance is continuous and differentiable almost everywhere. Moreover, compared to the KL-divergence, the loss function is much smoother so that the method converges to the ideal so-

---

**TABLE II**

**ARCHITECTURE OF DISCRIMINATIVE MODULE. DIFFERENTLY, BATCH NORMALIZATION AND STRIDES ARE USED TO OBTAIN LATENT REPRESENTATIONS.**

| Index | Index of Inputs | Operation(s) | Output Shape |
|-------|-----------------|--------------|--------------|
| (1)   | -               | Input Spectra | D            |
| (2)   | (1)            | Conv1D(5, 21, 5) | D/5 × 5     |
| (3)   | (2)            | BN + PReLU   | D/5 × 5     |
| (4)   | (3)            | Conv1D(10, 5, 2) | D/10 × 10   |
| (5)   | (4)            | BN + PReLU   | D/10 × 10   |
| (6)   | (5)            | Conv1D(20, 5, 2) | D/20 × 20   |
| (7)   | (6)            | BN + PReLU   | D/20 × 20   |
| (8)   | (7)            | Linear(5)    | D/20 × 5    |
lution freely from the initialization. Additionally, as discussed in [35], more accurate results can be obtained, since it provides robustness to the averaging operations in the ME.

In our case, $p$ and $q$ correspond to the distributions of input spectra and reconstructed spectra respectively and the adversarial loss function can be defined as:

$$
\mathcal{L}_{adv} = \mathbb{E}[D(\hat{x}; \theta_{adv})] - \mathbb{E}[D(x; \theta_{adv})]
$$

where $D(\cdot)$ is the discriminative module and it is composed of a family of 1-Lipschitz functions [27]. Moreover, $\tilde{x}$ is sampled uniformly from $l1$ normalized input and reconstructed spectra (This is critical since the angle-based metric is used in the model). $\lambda_{pq}$ determines the influence of the gradient penalty [33] for stable parameter learning.

For the discriminative module $D(\cdot)$, we adapt the terminology of the PatchGAN which is extensively used in the image translation problem [56]. More precisely, instead of penalizing the overall structure of spectra with the discriminative module, the spectral patches of spectra are used for the purpose of capturing local spectral characteristics of real and reconstructed spectra. The architecture details are presented in Table II.

**F. Implementation Details**

For the parameter optimization, since the proposed model is composed of different modules, five individual parameter sets should be learned from data. These are $\theta_g$ (corresponds to $w^s, b^s$), $\theta_e$, $\theta_r$, $\theta_u$ and $\theta_{adv}$. Therefore, loss functions and corresponding scale values for each set can be summarized as follows:

$$
\begin{align}
\mathcal{L}_{\theta_g} &= 0.01 \mathcal{L}_{re} + 0.1 \mathcal{L}_{adv}, \\
\mathcal{L}_{\theta_e} &= \mathcal{L}_{re}, \\
\mathcal{L}_{\theta_r} &= 0.001 \mathcal{L}_{re}, \\
\mathcal{L}_{\theta_u} &= 0.001 \mathcal{L}_{adv}, \\
\mathcal{L}_{\theta_{adv}} &= \mathcal{L}_{adv}.
\end{align}
$$

From Eq. 11, it can be seen that the Wasserstein GAN is dominantly used to optimize the parameters of $\theta_g$ and $\theta_u$ by which these parameters are implicitly used to provide robustness to the assumption of spectral variability. GAN models are powerful to form the noise space based on the definition of a problem. This is critical since the uncertainty exhibited from hyperspectral data can be precisely captured by the Wasserstein distance and noise space, and Wasserstein distance has several theoretical benefits to obtain a stable and accurate solution as explained throughout the manuscript.

Moreover, to optimize the latent representations learned from data, the DSCN parameters are updated by exploiting only the reconstruction loss $\mathcal{L}_{re}$. In addition, the learning rates for multinomial mixture kernel and auxiliary functions are scaled with the constants to decrease the learning speed compared to the DSCN module, since the DSCN parameters should converge faster so as to obtain more reliable abundances as expected.

Lastly, Adam stochastic optimizer [57] is utilized to update the parameters for each mini-batch by empirically defining the first-order moment term $\beta_1$ as 0.7. Note that the mini-batch size is fixed to 64. Also, we set the learning rate and the number of training iterations to 0.002 and 10K respectively for all datasets. The proposed algorithm is implemented on Python by extensively leveraging Tensorflow framework [31]. Additional results are also provided.

**III. EXPERIMENTS**

Experiments to compare the performance of the proposed method (DSCN++) with the baseline techniques are conducted on one synthetic and two real datasets. For this purpose, Linear Mixture Model (LMM) [7], Generalized Bilinear Model (GBM) [58], Post-Nonlinear Mixing Model (PPNM) [59], Multilinear Mixing Model (MLM) [7], Normal Compositional Model (NCM) [60] are selected for the hyperspectral unmixing baselines that are extensively used methods in the literature (codes are available on the web.). Note that the recent method in [17] needs high memory requirement which is impossible to compute the algorithm on real datasets especially by maintaining their original spatial resolutions.

Furthermore, Spatial Compositional Model (SCM) [61], Distance-MaxD (DMaxD) [3] and Sparse AutoEncoder Network (EndNet) [2] are exploited to estimate the constituent endmember spectra from the scenes.

Although no additional experiment is conducted, by comparing the reported results in [31], improved DSCN model improves performance for real datasets compared to the base model.

To evaluate the unmixing performance and compare the estimated abundances with the ground truth, Root Mean Square Error (RMSE) is utilized:

$$
RMSE(y, \hat{y}) = \sqrt{\frac{1}{L} \|y - \hat{y}\|^2_2}.
$$

where $y$ denotes the true abundance per-pixel while $\hat{y}$ indicates the estimated one by the baselines. Also, $L$ corresponds to the total number of pixels in the scene.

In following sections, we first provide the quantitative performance for each dataset that clearly demonstrates the impact of the proposed method by highlighting each individual step as well as different parameter configurations in detail. Note

\footnote{Project Page: https://github.com/savasozkan/dscn}
that $N$ is the only coefficient needed to be tuned by the user. Therefore $N$ is determined for each step based on their overall results in the experiments (As expected, $N$ (the number of mixture components) can vary for each dataset as well as each endmember method based on its distinctiveness). Later, the comparisons with the baselines are explained. For reliability of assessments, tests are repeated 20 times for the proposed method, thus mean and standard deviation are reported.

To ease the understandability, some abbreviations used in the Tables can be summarized as follows: DSCN indicates the deep spectral convolution network as explained in the subsection II-B. EU stands for the usage of NN models for the uncertainty term and the auxiliary model described in the subsection II-D. Lastly, WGAN indicates the use of Wasserstein GAN in the parameter optimization as mentioned in the subsection II-E.

### A. Synthetic Dataset

The unmixing methods are tested on the synthetic dataset released for the experiments in [17]. The dataset consists of four constituent materials from the ASTER spectral library as limestone, basalt, concrete and asphalt. The spectral range of data is from 0.4μm to 14μm and the spectral dimension is resampled to 200. Furthermore, the spatial resolution of the scene is 60 × 60. For the scene, limestone is set as the background material, the rest is randomly placed to the scene by maintaining the Gaussian-shaped distributions. The endmember spectra and abundance maps for the scene are illustrated in Fig. 4.

![Fig. 5. Spectra of pixels from the Synthetic dataset. It can be seen that even if random noise is added in the construction of data, spectral variability cannot be exhibited truly and it can be misleading for learning-based methods since learning-based methods (generative-based models in particular) can learn this random noise easily.](image)

Note that even if synthetic datasets are conventional to test the hyperspectral unmixing performance, it is not quite possible to capture all adverse conditions observed in the real data (such as spectral variability). Hence, it can be hard to validate the performance truly under this strong bias. To degrade the bias, the only practical solution is to add random noise on the data which cannot reflect the true physical conditions (i.e., spectral variability) observed in the real data. More precisely, Fig. 1 and Fig. 5 show the severe difference between real and synthetic datasets for pure materials. It can be seen that the angler distinctiveness is quite small (i.e., spectra of pixels are nearly concentrated on a single response) and use of angler-based metric can be misleading for the dataset. Hence, only for the synthetic dataset, the KL-divergence with SAD in the reconstruction loss is replaced (for SCM) or jointly used (for DmaxD, Endnet) with the ME (i.e., $\lambda_0 = 1$).

The experimental results for each individual step and different parameter configuration settings are reported in Table III. From the results, the proposed method achieves significantly better performance without EU and WGAN steps. This is quite reasonable since the data is degraded by random noise to simulate the spectral variability and the proposed model ultimately learns the noise from data than the assumption of spectral variability. Hence, this leads to the corruption for the model. However, note that true impacts of these steps can be clearly observed for real datasets. Therefore, further discussions about these steps are reserved for the real datasets.

The RMSE performance for the baselines are shown in Table IV. From the results, the proposed method obtain slightly better performance compared to the baselines. Note that the proposed method is effective on the synthetic dataset even if it does not exploit all its features (i.e., EU and WGAN).

Table III: RMSE performance on the synthetic dataset for different configurations.

| $N$ | DMaxD  | SCM  | EndNet |
|-----|--------|------|--------|
| 4   | 8.32 ± 1.4 | 4.47 ± 0.7 | 9.20 ± 1.4 |
| 8   | 8.15 ± 1.4 | 4.36 ± 0.9 | 9.14 ± 1.0 |
| 16  | 8.18 ± 1.0 | 3.97 ± 0.5 | 9.06 ± 1.1 |
| 24  | 7.55 ± 1.1 | 4.42 ± 0.6 | 9.51 ± 1.2 |

Table IV: RMSE performance on the synthetic dataset for baselines. The best results are indicated as Bold-Red while the second best results are shown as Bold.

| Method | RMSE |
|--------|------|
| PPNM [59] | 7.66 ± 4.01 |
| GBM [58] | 5.62 ± 3.26 |
| PPNM [59] | 7.66 ± 4.01 |
| MLM [7] | 6.52 ± 3.85 |
| NCM [60] | 4.09 ± 3.85 |
| DSCN++ | 5.23 ± 3.22 |
| DMaxD [3] | 5.17 ± 3.36 |
| SCM [61] | 3.66 ± 17.62 |
| EndNet [2] | 10.78 ± 8.32 |

Last note that [17] obtains $1.53 \times 10^{-2}$ RMSE performance for this synthetic data. However, due to its computation requirement and difference for the endmember estimation step, it is not completely compatible with our method.

### B. Urban Dataset

The spatial resolution of the scene is 307 × 307. The spectral range covers from 10 nm to 2500 nm and it is composed of 210 spectral bands. Several spectral bands (1-4, 76, 87, 101-111, 136-153 and 198-210) are removed from data due to the water-vapor absorption and atmospheric effects, thus...
the Urban dataset.

Fig. 6. Endmember spectra (first row) and abundance maps (second row) for the scene. The endmember spectra and abundance maps for the scene are illustrated in Fig. 6. The endmember spectra and abundance maps for the scene are observed in the scene: asphalt, grass, tree and roof [13].

Four material types are observed in the scene: asphalt, grass, tree and roof [13]. Fig. 6 shows the endmember spectra and abundance maps for the scene. The Jasper dataset is captured by the AVIRIS sensor and the dimension is $100 \times 100$ pixels. Several channels (1-3, 108-112, 154-166 and 220-224) are discarded due to the atmospheric effects and water-vapor absorption. The spectral band is of 198. There are four constituent materials in the scene as tree, water, soil and road [12]. Fig. 6 shows the endmember spectra and abundance maps for the scene.

The performance is compared with the baselines in Table VI and the proposed method outperforms all hyperspectral unmixing techniques significantly. In particular, the usage of WGAN and EU steps improve the performance of DmaxD such that it leads to a highly accurate solution where the baselines stuck to stationary results.

C. Jasper Dataset

The Jasper dataset is captured by the AVIRIS sensor and the dimension is $100 \times 100$ pixels. Several channels (1-3, 108-112, 154-166 and 220-224) are discarded due to the atmospheric effects and water-vapor absorption. The spectral band is of 198. There are four constituent materials in the scene as tree, water, soil and road [13]. Fig. 6 shows the endmember spectra and abundance maps for the scene.

The detailed performance for different configuration settings and steps are reported in Table VII. It can be seen that similar results can be observed as stated for the Urban dataset. A single outlier is that SCM without the DSCN achieves slightly better performance compared to the baseline. This stems from the fact that the SCM representation space can be more discriminative than the DSCN for this dataset.

For the baselines comparisons, the proposed method obtains state-of-the-art performance as demonstrated in Tab VIII. Compared to [31] which corresponds to the baseline DSCN, the improved model significantly improves the performance while small variations in the performance can be observed.

IV. DISCUSSION AND CONCLUSION

In this manuscript, we introduce a novel neural network (NN) framework for hyperspectral unmixing. It is mainly composed of deep spectral convolution networks (DSCNs) by addressing the assumption of spectral variability that is one of the severe physical conditions frequently observed in real data. More precisely, in real applications, pixel responses for each material can be varied due to the atmospheric/illumination conditions instead of fixed material spectra. Therefore, these conditions should be addressed by an effective model in order to obtain state-of-the-art unmixing performance.

![Fig. 6. Endmember spectra (first row) and abundance maps (second row) for the Urban dataset.](image)

![Fig. 7. Endmember spectra (first row) and abundance maps (second row) for the Jasper dataset.](image)
For this purpose, we present critical contributions throughout the manuscript as follows: First, we make several modifications in the architecture of the DSCNs to obtain stationary yet more accurate results. Second, a novel NN layer is proposed to estimate the abundances per-pixel by leveraging the multinomial mixture kernel. This kernel particularly provides robustness for the assumption of spectral variability since the distribution-based assumptions can be more convenient to handle to these adverse conditions as explained in the literature. Moreover, an extra NN model is utilized to capture the uncertainty term in the formulation of the spectral variability. Third, the Wasserstein GAN is exploited in the parameter optimization which leads to more accurate and stable performance than the KL-divergence and the ME. Lastly, the model is formulated as a full pipeline based on a NN method that can be learned by backpropagation and a stochastic gradient solver.

The experimental results validate that the proposed method outperforms well-known hyperspectral unmixing baselines by a large margin. In particular, the methods obtains state-of-the-art performances on the real datasets where severe spectral variability can be observed.

Complexity: As analyzed in [2], [62], a NN model with the stochastic gradient algorithm significantly decreases the complexity and memory requirement needed by the model. More precisely, these parameters are independent from the data size due to the batch-based learning. Hence, the method is more applicable for large-scale data.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research.

### TABLE VII

RMSE PERFORMANCE ON THE JASPER DATASET FOR DIFFERENT CONFIGURATIONS.

| $\times 10^{-2}$ | DMaxD [3] | SCM [61] | EndNet [2] |
|-----------------|----------|----------|------------|
| $N = 4$ | 12.03 ±0.7 | 18.10 ±0.9 | 6.38 ±0.7 |
| $N = 8$ | 12.29 ±0.9 | 18.33 ±1.1 | 6.61 ±0.7 |
| $N = 16$ | 12.12 ±0.7 | 17.92 ±0.8 | 6.31 ±0.7 |
| $N = 24$ | 12.27 ±0.6 | 18.03 ±1.1 | 6.62 ±1.1 |
| w/o DSCN | 12.47 ±0.6 | 17.49 ±1.2 | 6.87 ±2.2 |
| w/o EU | 14.78 ±0.4 | 20.26 ±0.4 | 8.36 ±0.5 |
| w/o WGAN | 15.63 ±0.3 | 19.03 ±0.4 | 10.15 ±0.5 |

### TABLE VIII

RMSE PERFORMANCE ON THE JASPER DATASET FOR BASELINES. THE BEST RESULTS ARE INDICATED AS BOLD-RED WHILE THE SECOND BEST RESULTS ARE SHOWN AS BOLD.

| $\times 10^{-2}$ | DMaxD [3] | SCM [61] | EndNet [2] |
|-----------------|----------|----------|------------|
| LMM [7] | 15.31 | 19.78 | 22.60 |
| GBM [53] | 15.98 | 19.95 | 21.91 |
| PPNM [59] | 12.99 | 19.78 | 14.72 |
| MLM [7] | 16.59 | 20.73 | 8.13 |
| NCM [50] | 15.29 | 19.56 | 22.42 |
| DSCN++ | 12.03 | 17.92 | 6.31 |

### REFERENCES

[1] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," IEEE Transactions on Image Processing, vol. 20, no. 4, pp. 1112–1125, 2011.
[2] S. Ozkan, B. Kaya, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," arXiv preprint arXiv:1708.01894, 2017.
[3] R. Heylen, D. Burazinovic, and P. Scheunders, "Non-linear spectral unmixing by geodesic simplex volume maximization," IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 3, pp. 534–542, 2011.
[4] J. J. Settle, "On the relationship between spectral unmixing and subspace projection," IEEE Transactions on Geoscience and Remote Sensing, vol. 34, no. 4, pp. 1045–1046, 1996.
[5] N. Keshava and J. F. Mustard, "Spectral unmixing," IEEE signal processing magazine, vol. 19, no. 1, pp. 44–57, 2002.
[6] N. Dobigeon, Y. Altmann, N. Brun, and S. Moussaoui, "Linear and nonlinear unmixing in hyperspectral imaging," in Data Handling in Science and Technology. Elsevier, 2016, vol. 30, pp. 185–224.
[7] R. Heylen and P. Scheunders, "A multilinear mixing model for non-linear spectral unmixing," IEEE transactions on geoscience and remote sensing, vol. 54, no. 1, pp. 240–251, 2016.
[8] B. Somers, K. Cools, S. Delalleux, J. Stuckens, D. Van der Zande, W. W. Verstraeten, and P. Coppin, "Nonlinear hyperspectral mixture analysis for tree cover estimates in orchards," Remote Sensing of Environment, vol. 113, no. 6, pp. 1183–1193, 2009.
[9] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Bilinear models for nonlinear unmixing of hyperspectral images," in Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on. IEEE, 2011, pp. 1–4.
[10] J. M. Nascimento and J. M. Bioucas-Dias, "Nonlinear mixture model for hyperspectral unmixing," in Image and Signal Processing for Remote Sensing XV, vol. 7477. International Society for Optics and Photonics, 2009, p. 74770J.
[11] N. Raksutom and Q. Du, “Nonlinear spectral mixture analysis for hyperspectral imagery in an unknown environment,” IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 4, pp. 836–840, 2010.
[12] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 88, pp. 101–118, 2014.
[13] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, “Spectral unmixing via data-guided sparsity,” IEEE Transactions on Image Processing, vol. 23, no. 12, pp. 5412–5427, 2014.
[14] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravorty, “Daen: Deep autoencoder networks for hyperspectral unmixing,” IEEE Transactions on Geoscience and Remote Sensing, 2019.
[15] 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 Transactions on Geoscience and Remote Sensing, 2018.
[16] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, “Hyperspectral unmixing via deep convolutional neural networks,” IEEE Geoscience and Remote Sensing Letters, no. 99, pp. 1–5, 2018.
[17] Y. Zhou, A. Rangarajan, and P. D. Gader, “A gaussian mixture model representation of endmember variability in hyperspectral unmixing,” IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2242–2256, 2018.
[18] D. A. Roberts, M. Gardner, R. Church, S. Ustin, G. Scheer, and R. Green, “Mapping chaparral in the santa monica mountains using multiple endmember spectral mixture models,” Remote Sensing of Environment, vol. 54, no. 1, pp. 240–251, 2003.
[19] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, “Blind spectral unmixing based on sparse nonnegative matrix factorization,” IEEE Transactions on Image Processing, vol. 20, no. 4, pp. 1112–1125, 2011.
