NMF-SAE: AN INTERPRETABLE SPARSE AUTOENCODER FOR HYPERSPECTRAL UNMIXING

Fengchao Xiong[1] *Jun Zhou*[2] *Minchao Ye*[3] *Jianfeng Lu*[1] *Yuntao Qian*[4]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2] School of Information and Communication Technology, Griffith University, Australia
[3] College of Information Engineering, China Jiliang University, China
[4] College of Computer Science, Zhejiang University, China

ABSTRACT

Hyperspectral unmixing is an important tool to learn the material constitution and distribution of a scene. Model-based unmixing methods depend on well-designed iterative optimization algorithms, which is usually time consuming. Learning-based methods perform unmixing in a data-driven manner but heavily rely on the quality and quantity of the training samples due to the lack of physical interpretability. In this paper, we combine the advantages of both model-based and learning-based methods and propose a nonnegative matrix factorization (NMF) inspired sparse autoencoder (NMF-SAE) for hyperspectral unmixing. NMF-SAE consists of an encoder and a decoder, both of which are constructed by unrolling the iterative optimization rules of L_1 sparsity-constrained NMF for the linear spectral mixture model. All parameters in our method are obtained by end-to-end training in a data-driven manner. Our network is not only physically interpretable and flexible but also has higher learning capacity with fewer parameters. Experimental results on both synthetic and real-world data demonstrate that our method is capable of producing desirable unmixing results when compared against several alternative approaches.

***Index Terms*—** Hyperspectral unmixing, model-based neural network, autoencoder, sparse coding

1. INTRODUCTION

Hyperspectral imagery (HSI) not only captures two-dimensional spatial information but also one-dimensional spectrum indexed by a large number of narrow spectral bands. These spectral bands enable HSI material identification ability, making HSI widely used in many remote sensing and computer vision applications [1, 2]. Because of limited spatial resolution of HSI sensors, a pixel possibly covers several materials, i.e., “mixed” pixels. The presence of these pixels dramatically hinders the practical utility of HSIs. Hyperspectral unmixing decomposes a pixel into the combination of a number of constituent materials, i.e., endmembers, and their corresponding fractions, i.e., abundances, offering an attractive way to tackle this problem.

Model-based methods make explicit assumptions on the physical interaction between materials for unmixing. Under the framework of linear mixture model (LMM), many model-based methods have been developed, ranging from geometrical to statistical ones [3]. Nonnegative matrix factorization (NMF) and its extensions [4–6] are widely used thanks to better interpretability, i.e., the factor matrices can be respectively linked with endmembers and abundances. In order to tackle the information loss when converting the three-dimensional HSI into a two-dimensional matrix, nonnegative tensor factorization (NTF) was also adopted for unmixing task [7]. Unfortunately, these model based approaches rely on well-designed optimization algorithms, which usually require high number of numerical iterations and tedious hyperparameter tuning [8].

Driven by the powerful learning ability of deep neural networks (DNNs), some attempts have been made on unsupervised deep learning for hyperspectral unmixing, i.e., learning-based unmixing [9–13]. Autoencoders are commonly adopted methods, in which the encoder module transforms the input data to hidden concepts, i.e., abundances, and decoder module uses their bases, i.e., endmembers, to reconstruct the data. Su *et.al* [14] proposed a deep autoencoder network (DAEN), in which variational autoencoder (VAE) is employed to simultaneously obtain endmember signatures and abundance fractions. Instead of tied weights structure between encoder and decoder, Qu *et.al* [15] developed an untied sparse denoising autoencoder (uDAS) to make them independent, producing more favorable endmembers and less reconstruction error. Additionally, deep convolutional autoencoders are also employed for this task [16, 17], where a series of convolutional layers act as the encoder module.

A problem of the above learning-based methods is that they ignore an important prior knowledge on the unmixing

This work was supported in part by “111” Program (No. B13022), Jiangsu Provincial Natural Science Foundation of China under Grant BK20200466, the National Key Research and Development Program of China 2018YFB0505000 and the National Natural Science Foundation of China under Grant 62002169, 61701468 and 62071421.

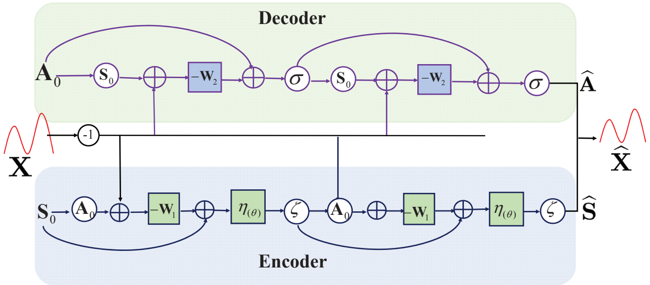


Fig. 1. Network architecture of NMF-SAE which consists of encoder and decoder modules respectively inspired by the optimization of endmember ($\mathbf{A}$) and abundances ($\mathbf{S}$).

problem, i.e., spectral mixture model. Instead, they perform unmixing in a "black box" manner and lack physical interpretability. Their complex topological network structure also hinders the theoretical analysis of their unmixing mechanism [18]. Recently, Qian *et.al* [19] blended the model-based approaches with learning-based approaches and introduced a spectral mixture model inspired neural network for blind unmixing (MNN-BU) by unrolling the optimization of abundances. Though better results were obtained, the decoder module is interpreted as an additional endmember matrix directly connected with the encoder (abundances), which fails to consider unrolling the optimization of endmembers.

In this paper, we propose a nonnegative matrix factorization inspired sparse autoencoder (NMF-SAE) for hyperspectral unmixing, which combines the learnable DNN together with prior knowledge formulated by L_1-norm regularized NMF (L_1-NMF). Fig. 1 presents the framework of NMF-SAE. Specifically, an L_1-NMF unmixing model is first constructed under the framework of LMM. Second, the iterative update rules are induced for solving endmembers and abundances. Third, by introducing some learnable weights and uniting them across the layers as parameters, the update rules are unrolled to yield two DNNs, respectively for endmembers and abundances, i.e., the decoder and encoder in Fig. 1. Finally, all the desired parameters including hyperparameters and introduced parameters are learned by discriminatively training NMF-SAE. NMF-SAE inherits strong representation learning ability of learning-based methods while simultaneously maintaining the physical interpretability of model-based methods. Compared with MNN-BU, NMF-SAE not only better considers the optimization of endmembers but also has more residual connections with less parameters, significantly enhancing the learning ability and accelerating the training process. Experimental results on synthetic dataset and real-world datasets confirm its effectiveness in unmixing.

2. PROPOSED NMF-SAE METHOD

2.1. L_1-NMF Unmixing Model

LMM considers the acquired spectrum as a linear combination of R endmembers. Letting $\mathbf{X}$ be HSI with N pixels and L bands, LMM can be mathematically formulated as

$$\mathbf{X} = \mathbf{AS} + \mathbf{E} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{L\times R}$ is the endmember matrix, $\mathbf{S} \in \mathbb{R}^{R\times N}$ denotes abundances, and $\mathbf{E}$ accounts for measurement errors.

Abundance nonnegative constraint (ANC) and abundance sum-to-one constraint (ASC) are two constraints usually added to Eq. (1) to satisfy the physical interpretation. ANC means the fractions of each endmembers should never be negative and ASC requires the total contributions of all the endmembers should equal one. Besides ANC and ASC, sparsity constraint is also widely applied based on the fact that most pixels only consist of a subset of endmembers. Adding these three constraints into Eq. (1), the sparse unmixing problem can be modeled as the following optimization problem

$$\min_{\mathbf{A},\mathbf{S}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2+\lambda\|\mathbf{S}\|_1 \quad \text{s.t. } \mathbf{1}_R^{\mathrm{T}}\mathbf{S} = \mathbf{1}_N^{\mathrm{T}}, \mathbf{A}, \mathbf{S} \succeq 0 \tag{2}$$

Here, L_1 norm of $\mathbf{S}$ is used to achieve sparse solutions and $\mathbf{1}$ is a vector of all ones. The L_1-NMF can be interpreted as one-layer sparse autoencoder, in which $\mathbf{S}$ denotes the hidden layer representation in the encoder component and $\mathbf{A}$ acts as a decoder component mapping the representation to the original feature space.

2.2. NMF-SAE

The optimization problem in Eq. (2) can be decomposed into two subproblems, respectively for estimating $\mathbf{A}$ and $\mathbf{S}$, i.e.,

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X}-\mathbf{A}_0\mathbf{S}\|_F^2+\lambda\|\mathbf{S}\|_1 \quad \text{s.t. } \mathbf{1}_R^{\mathrm{T}}\mathbf{S} = \mathbf{1}_N^{\mathrm{T}}, \mathbf{S} \succeq 0 \tag{3}$$

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{X}-\mathbf{A}\mathbf{S}_0\|_F^2 \quad \text{s.t. } \mathbf{A} \succeq 0 \tag{4}$$

where $\mathbf{A}_0$ and $\mathbf{S}_0$ are the initial values of $\mathbf{A}$ and $\mathbf{S}$, respectively. Proximal gradient descent (PGD) algorithm is a common practice to solve above two problems. Taking $\mathbf{S}$ for example, under the framework of PGD, $\mathbf{S}$ can be obtained by recursively executing the following equation until convergence:

$$\mathbf{S}_k = \mathbf{prox}_\theta\left(\mathbf{S}_{k-1} - t_s\mathbf{A}_0^{\mathrm{T}}(\mathbf{A}_0\mathbf{S}_{k-1} - \mathbf{X})\right) \tag{5}$$

where $t_s > 0$ is the step size usually set as $\frac{1}{\|\mathbf{A}_0^{\mathrm{T}}\mathbf{A}_0\|_2}$. $\mathbf{prox}_\theta$ is the proximal operator guarantying the sparsity and nonnegativity of $\mathbf{S}$. It is defined as an element-wise vector one-sided shrinkage function with a vector of thresholds $\theta = \lambda t_s$, i.e., $\mathbf{prox}_\theta(\mathrm{s}) = \max(\mathrm{s} - \theta, 0)$. Afterwards, each column of $\mathbf{S}$ is normalized to satisfy ASC by the following equation:

$$S_{rn} = \frac{S_{rn}}{\sum_{r=1}^{R} S_{rn}} \tag{6}$$

The iteration of $\mathbf{S}$ collectively resembles a typical local block of DNN, consisting of a linear operation, a proximal operation and a normalization operation. The proximal operation mimics a nonlinear activation function, e.g., $\mathbf{prox}_\theta(\mathbf{s}) = \mathrm{Relu}(\mathbf{s} - \theta)$. By introducing a learnable parameter $\mathbf{W}_1 = t_s\mathbf{A}_0^{\mathrm{T}}$, Eq. (5) can be unrolled into a network structure with a fixed number of layers (K), whose consecutive layers are connected by

$$\mathbf{S}_{k+1} = \zeta\left(\eta_{(\theta)}\left(\mathbf{S}_k - \mathbf{W}_1\left(\mathbf{A}_0\mathbf{S}_k - \mathbf{X}\right)\right)\right) \tag{7}$$

where $\zeta(\cdot)$ is a normalization layer satisfying ASC in Eq. (6). Parameterized by a vector of thresholds θ, $\eta_{(\theta)}(\cdot)$ acts as $\mathbf{prox}_\theta(\cdot)$ in Eq. (5). Likewise, we can obtain the subnetwork for $\mathbf{A}$ whose consecutive layers are given by

$$\mathbf{A}_{k+1} = \sigma\left(\mathbf{A}_k - \left(\mathbf{A}_k\mathbf{S}_0 - \mathbf{X}\right)\mathbf{W}_2\right) \tag{8}$$

where $\sigma(\cdot)$ indicates the ReLu activate function ensuring the nonnegativity of $\mathbf{A}$ and $\mathbf{W}_2 = t_a\mathbf{S}_0^{\mathrm{T}}$ with $t_a = \frac{1}{\|\mathbf{S}_0\mathbf{S}_0^{\mathrm{T}}\|_2}$

In summary, we unroll the iterative optimization of L_1-NMF to formulate two subnetworks corresponding to the decoder and encoder respectively. Gathering their output as in Fig. (1) yields an end-to-end sparse autoencoder (NMF-SAE) with parameter set of $\Theta = \{\mathbf{W}_1, \mathbf{W}_2, \theta\}$. All the parameters can be learned by the training data through backpropagation. Forward feeding the network resembles executing the traditional L_1-NMF for a finite number of times, dramatically boosting computational efficiency. Unlike traditional deep unmixing network, the network architecture is induced under the framework of LMM and the optimization of L_1-NMF, therefore the physical interpretation is naturally preserved. Thanks to such interpretation, the domain prior knowledge and sophisticated learned knowledge about the HSI data can be more flexibly transferred into the network, e.g., the parameters can be easily initialized by existing unmixing method.

2.3. Training and Inference

In general, the ground truth $\mathbf{A}$ and $\mathbf{S}$ are both unknown in blind unmixing. Instead of discriminatively learning $\mathbf{A}$ and $\mathbf{S}$, we take reconstruction error of $\mathbf{X}$ as the loss function, i.e.,

$$\mathcal{L}_\Theta = \frac{1}{2}\|\mathbf{A}_K\mathbf{S}_K - \mathbf{X}\|_F^2 \tag{9}$$

where K indexes the last layer. For simplicity, the numbers of layers in encoder and decoder are set to the same. Once the network is trained, $\mathbf{A}$ can be obtained by the decoder subnetwork. For new samples, their abundance $\mathbf{S}$ can be obtained by the forward process of encoder sub-network.

3. EXPERIMENTAL RESULTS

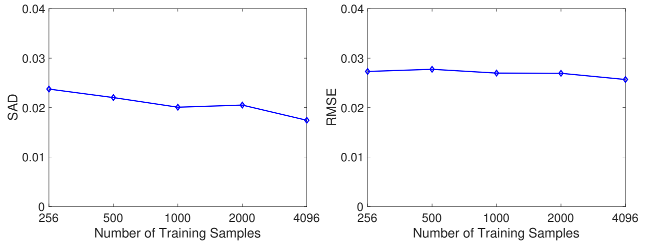
We have conducted a range of experiments to demonstrate the superiority of the proposed method. We selected two model-based methods TV-RSNMF [20], MV-NTF [7], two learning-based methods, uDAS [15] and DAEN [14] and one model-based neural network method, MNN-BU [19] as alternative methods to be compared with our method. All relevant parameters are set as suggested in the original implementation. The spectral angle distance (SAD) and root square error (RMSE) defined in [21] are used to measure the unmixing performance.

3.1. Network Setup

We first run vertex component analysis (VCA) [22] and fully constrained least squares (FCLS) [23] algorithm to respectively yield $\mathbf{A}_0$ and $\mathbf{S}_0$. Afterwards, the network is initialized by setting $\mathbf{W}_1 = t_s\mathbf{A}_0^{\mathrm{T}}$, $\mathbf{W}_2 = t_a\mathbf{S}_0^{\mathrm{T}}$. All the values in θ are set as $0.01t_s$. The network is implemented on the PyTorch platform and trained using Adam optimizer. We use full batch for training. For synthetic data, the learning rate of encoder and decoder respectively are set as $1e-4$ and $5e-4$. In terms of real-world data, their learning rates are both set as $1e-6$. The number of training iterations and layers are set as 1000 and 2, respectively.

3.2. Experimental Results on Synthetic Data

The synthetic data was produced following the method in [21]. Six pure signatures, i.e., Carnallite, Ammoniojarosite, Almandine, Brucite, Axinite and Chlonte were selected from USGS library to construct the endmembers. The abundances were produced by following procedures: 1) Divide a synthetic image size of $z^2 \times z^2$ into z^2 non-overlapping regions containing $z \times z$ pixels; 2) Fill each region with two randomly selected endmembers respectively of fraction β and $1 - \beta$; 3) Blur the image using a Gaussian filter with kernel size set as $(z + 1) \times (z + 1)$; 4) Rescale the fractions of all endmembers in a pixel to satisfy the ASC constraint. In our setting, the clean HSI was generated under the configuration of $z = 8$, $\beta = 0.8$, yielding HSI with 4096 pixels.



(a) SAD

(b) RMSE

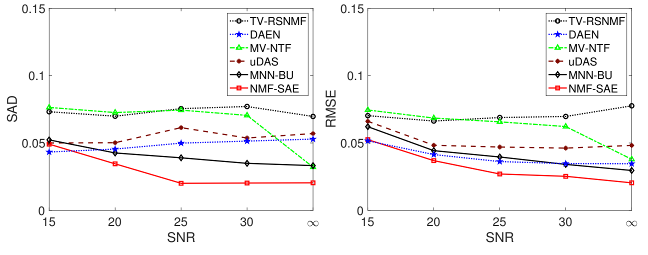
Fig. 2. SAD and RMSE with respect to the number of training samples.

Influence of the number of training samples: Setting SNR=25dB, Fig. 2 shows the impact of the number of training samples on SAD and RMSE. Even with only 256 pixels (6.25% of all the pixels), NMF-SAE is able to produce very comparable SAD and RMSE, which clearly demonstrates the strong learning ability of NMF-SAE. The more training samples there are, the better the data distribution can usually be described, and therefore the better unmixing performance accordingly is provided. On the other hand, with the increasing

Table 1. Means and standard deviations of the SAD on Jasper Ridge dataset.

Algorithm	TV-RSNMF [20]	MV-NTF [7]	DAEN [14]	uDAS [15]	MNN-BU [19]	NMF-SAE
Tree	0.0776±4.91%	0.2209±2.17%	0.1774±0.82%	0.1511±1.37%	**0.0468±0.32%**	0.0494±1.90%
Water	0.1337±1.31%	0.2352±1.65%	0.3237±1.42%	0.1742±11.39%	0.1179±0.23%	**0.0729±0.50%**
Soil	0.1047±10.11%	0.1752±4.67%	0.1123±2.03%	0.1639±6.09%	**0.0357±0.43%**	0.0527±4.17%
Road	0.6515±27.46%	0.1741±4.35%	**0.0588±1.59%**	0.0597±0.27%	0.0901±0.37%	0.0932±0.57%
Mean	0.2419±3.06%	0.2015±1.77%	0.1680± 0.43%	0.1373±3.27%	0.0726±0.11%	**0.0671±1.47%**

of training samples, the more training time is required. Considering the balance between training time and unmixing accuracy, we set the number of training samples as 1000 in the rest experiments.



(a) SAD

(b) RMSE

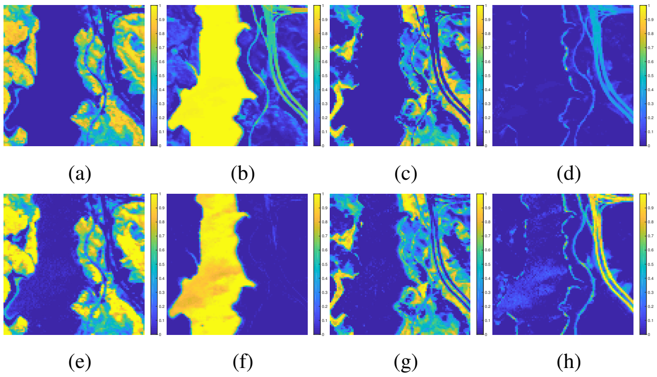
Fig. 3. Performance of all six unmixing methods with respect to different noise levels in terms of (a) SAD and (b) RMSE.

Comparison with the state-of-the-arts: Here, we show the unmixing performance of all the competing methods with respect to different SNRs of Gaussian noise. Except our method, all other learning-based methods use all the pixels for training. Fig. 3 presents the experimental results. Generally, learning-based methods outperform TV-RSNMF and MV-NTF in most cases, which implies that learning-based methods take more advantages of available training data to enhance representation ability. Compared with DAEN, uDAS and MNN-BU, NMF-SAE achieves even better in terms of both SAD and RMSE by using approximately 25% of the pixels for training. The main reason is that our method incorporates the linear spectral mixture model into deep neural network, combining the benefits of model-based methods in good interpretability and the learning-based methods in strong learning ability. Moreover, the number of parameters of NMF-SAE is $R + RN + RL$, much smaller than that of MNN-BU ($KR + KRN + KRL + KR^2$). Fewer parameters, additional consideration of unrolling the optimization of endmembers and more residual connections contribute to improved performance over MNN-BU. In summary, experimental results in this study illustrates the effectiveness in unmixing task and learning ability of NMF-SAE.

3.3. Experimental Results on Real-world Data

In order to test the unmixing ability on real-world dataset, we ran all the methods on the widely-used Jasper Ridge dataset. The dataset was collected by AVIRIS sensor and originally contains 512 × 614 pixels with 224 electromagnetic bands whose the electromagnetic spectrum ranges from 380 nm to 2, 500 nm. We removed bands corrupted by water vapor absorption and noises and cropped a sub-scene from the image, leaving 198 bands with 100 × 100 pixels for evaluation. In this experiment, we set four types of endmembers, i.e., tree, water, soil, and road.

Table 1 compares the unmixing accuracy in terms of SAD. Similar to the observation on the synthetic data, learning-based methods outperform TV-RSNMF and MV-NTF. In contrast to other three learning-based methods, NMF-SAE achieves comparatively more accurate endmembers by providing an average SAD of 0.0671. Fig. 4 gives a qualitative comparison between abundance produced by NMF-SAE and corresponding references. It can be seen that the abundance matches the references both perceptually and visually. Overall, the promising unmixing performance in real-world dataset evidently confirms the unmixing ability of NMF-SAE again.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Abundance maps comparison. Estimated abundance map of each endmember: (a)Tree; (b) Water; (c) Soil; (d) Road. (e)-(h) are associated reference abundance maps.

4. CONCLUSION

In this paper, an L_1-NMF inspired sparse autoencoder, NMF-SAE, is proposed for hyperspectral unmixing. NMF-SAE unrolls the iterative update scheme of L_1-NMF driven by LMM into a sparse autoencoder whose encoder and decoder respectively account for abundances and endmembers. In contrast to alternative autoencoder networks, the favorable interpretability makes it flexible to incorporate the prior knowledge into the network. The strong learning ability facilitating improving unmixing performance. Experimental results on synthetic data and real-world dataset demonstrate its advantage over alternatives. In our future research, we will incorporate spatial structure into the network to enhance unmixing performance.

5. REFERENCES

[1] S. Jia, Z. Lin, B. Deng, J. Zhu, and Q. Li, “Cascade superpixel regularized gabor feature fusion for hyperspectral image classification,” *IEEE TNNLS*, vol. 31, no. 5, pp. 1638–1652, 2020.

[2] F. Xiong, J. Zhou, and Y. Qian, “Material based object tracking in hyperspectral videos,” *IEEE TIP*, vol. 29, pp. 3719–3733, 2020.

[3] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE JSTARS*, vol. 5, no. 2, pp. 354–379, 2012.

[4] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, “Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization,” *IEEE TGRS*, vol. 49, no. 11, pp. 4282–4297, 2011.

[5] B. Yang, W. Luo, and B. Wang, “Constrained nonnegative matrix factorization based on particle swarm optimization for hyperspectral unmixing,” *IEEE JSTARS*, vol. 10, no. 8, pp. 3693–3710, 2017.

[6] X. Lu, L. Dong, and Y. Yuan, “Subspace clustering constrained sparse NMF for hyperspectral unmixing,” *IEEE TGRS*, vol. 58, no. 5, pp. 3007–3019, 2020.

[7] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, “Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery,” *IEEE TGRS*, vol. 55, no. 3, pp. 1776–1792, 2017.

[8] T. Tirer and R. Giryes, “Image restoration by iterative denoising and backward projections,” *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1220–1234, 2019.

[9] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, “Hyperspectral unmixing via deep convolutional neural networks,” *IEEE GRSL*, vol. 15, no. 11, pp. 1755–1759, 2018.

[10] S. Ozkan, B. Kaya, and G. B. Akar, “Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing,” *IEEE TGRS*, vol. 57, no. 1, pp. 482–496, 2019.

[11] G. A. Licciardi and F. Del Frate, “Pixel unmixing in hyperspectral data by means of neural networks,” *IEEE TGRS*, vol. 49, no. 11, pp. 4163–4172, 2011.

[12] Z. Dou, K. Gao, X. Zhang, H. Wang, and J. Wang, “Blind hyperspectral unmixing using dual branch deep autoencoder with orthogonal sparse prior,” in *Proc. ICASSP*, 2020, pp. 2428–2432.

[13] R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez, “Deep generative endmember modeling: An application to unsupervised spectral unmixing,” *IEEE TCI*, vol. 6, pp. 374–384, 2020.

[14] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, “DAEN: Deep autoencoder networks for hyperspectral unmixing,” *IEEE TGRS*, vol. 57, no. 7, pp. 4309–4321, 2019.

[15] Y. Qu and H. Qi, “uDAS: An untied denoising autoencoder with sparsity for spectral unmixing,” *IEEE TGRS*, vol. 57, no. 3, pp. 1698–1712, 2019.

[16] F. Khajehrayeni and H. Ghassemian, “Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario,” *IEEE JSTARS*, vol. 13, pp. 567–576, 2020.

[17] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, “Convolutional autoencoder for spectral-spatial hyperspectral unmixing,” *IEEE TGRS*, pp. 1–15, 2020 (In Press).

[18] R. Liu, S. Cheng, Y. He, X. Fan, Z. Lin, and Z. Luo, “On the convergence of learning-based iterative methods for nonconvex inverse problems,” *IEEE TPAMI.*, pp. 1–1, 2019.

[19] Y. Qian, F. Xiong, Q. Qian, and J. Zhou, “Spectral mixture model inspired network architectures for hyperspectral unmixing,” *IEEE TGRS*, vol. 58, no. 10, pp. 7418–7434, 2020.

[20] W. He, H. Zhang, and L. Zhang, “Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing,” *IEEE TGRS*, vol. 55, no. 7, pp. 3909–3921, 2017.

[21] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, “Hyperspectral unmixing via total variation regularized nonnegative tensor factorization,” *IEEE TGRS*, vol. 57, no. 4, pp. 2341–2357, 2019.

[22] J. M. P. Nascimento and J. M. B. Dias, “Vertex component analysis: a fast algorithm to unmix hyperspectral data,” *IEEE TGRS*, vol. 43, no. 4, pp. 898–910, 2005.

[23] D. C. Heinz and Chein-I-Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE TGRS*, vol. 39, no. 3, pp. 529–545, 2001.