

Convolutional Autoencoder for Spectral–Spatial Hyperspectral Unmixing

Burkní Palsson^{ID}, Member, IEEE, Magnus O. Ulfarsson^{ID}, Senior Member, IEEE,
and Johannes R. Sveinsson^{ID}, Senior Member, IEEE

Abstract—Blind hyperspectral unmixing is the process of expressing the measured spectrum of a pixel as a combination of a set of spectral signatures called endmembers and simultaneously determining their fractional abundances in the pixel. Most unmixing methods are strictly spectral and do not exploit the spatial structure of hyperspectral images (HSIs). In this article, we present a new spectral–spatial linear mixture model and an associated estimation method based on a convolutional neural network autoencoder unmixing (CNNAEU). The CNNAEU technique exploits the spatial and the spectral structure of HSIs both for endmember and abundance map estimation. As it works directly with patches of HSIs and does not use any pooling or upsampling layers, the spatial structure is preserved throughout and abundance maps are obtained as feature maps of a hidden convolutional layer. We compared the CNNAEU method to four conventional and three deep learning state-of-the-art unmixing methods using four real HSIs. Experimental results show that the proposed CNNAEU technique performs particularly well and consistently when it comes to endmembers’ extraction and outperforms all the comparison methods.

Index Terms—Hyperspectral data unmixing, deep neural network learning, spectral–spatial model, image processing.

I. INTRODUCTION

THE high spectral resolution of hyperspectral sensors allows for the identification of distinct materials based on their spectral signatures. This is, however, complicated by the limited spatial resolution of such sensors. Because of the low spatial resolution, there is usually more than one distinct material in every pixel of the acquired image. Blind hyperspectral unmixing (HU) is the process of determining the spectral signatures of different materials and also their fractional abundances in every pixel. At the end of the process, we want to be able to reconstruct the original data with as little error as possible, from the set of extracted endmembers and their associated abundance maps.

A. Motivation and Contributions

The motivation behind our method is to exploit the spatial structure of HSIs in addition to their spectral information.

Manuscript received December 1, 2019; revised February 18, 2020 and March 26, 2020; accepted May 2, 2020. Date of publication May 19, 2020; date of current version December 24, 2020. This work was supported in part by the Icelandic Research Fund under Grant 174075-05. (*Corresponding author: Johannes R. Sveinsson*.)

The authors are with the Faculty of Electrical and Computer Engineering, University of Iceland, IS-107 Reykjavik, Iceland (e-mail: sveinss@hi.is).

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

Digital Object Identifier 10.1109/TGRS.2020.2992743

The pixels in HSIs are, as in other natural images, strongly correlated with their neighbors. This spatial structure contains information that can be used along with the spectral information to constrain further the unmixing problem.

Unmixing methods that exploit the spatial structure are called spectral–spatial methods. The most common way to utilize the spatial structure of HSIs is by refining a spectral method using some spatial prior to regularize or control the sparsity and the smoothness of the abundance maps obtained [1] or by doing preprocessing on the data prior to using a spectral only method. Developing entirely new spatially oriented methods is not as common.

In this article, a new spectral–spatial linear mixture model is introduced along with an associated estimation method based on a convolutional neural network (CNN) autoencoder. This is, to our best knowledge, the first blind spectral–spatial unmixing method based on CNNs. Since the method uses CNN, it involves spatial filtering to extract features to aid it in the unmixing task, thus directly exploiting the spatial structure of the HSI. Furthermore, the technique preserves the spatial structure of HSIs throughout the method while abundance maps arise naturally as feature maps of a hidden convolutional layer. This makes it very simple to apply various spatial regularizations on the abundance maps, such as total variation (TV).

B. Publication Review

The demand for ever better spectral unmixing algorithms has resulted in a great number of methods over the last three decades. The vast majority of them are based on the linear mixing model (LMM), the most common model in unmixing. The problem of spectral unmixing can be interpreted in different ways. We can interpret it geometrically, statistically, or as a sparse regression problem, leading to three different categories of methods [2], [3]. Sparse regression methods seek to express the observed spectra as linear combinations of known spectral signatures from spectral libraries [4]–[7]. Such methods are among the earliest unmixing methods. Approaches based on compressed sensing, such as [8]–[10], are usually grouped with the sparse regression methods. The geometrical-based methods can, whether they rely on pure pixels or not, be further categorized into so-called pure pixel methods or minimum volume methods. The best-known pure pixel method is undoubtedly vertex component analysis (VCA) [11], while minimum volume simplex analysis (MVSA) [12] is a good example of a minimum volume technique. Examples of $\ell_{1/2}$

and ℓ_q sparsity constrained minimum volume methods are [13] and [14], respectively. Finally, statistical methods reformulate the unmixing problem as an inference problem [15]–[17]. An example of a statistical technique that implements spatially constrained unmixing is [18].

The number of unmixing methods based on deep learning has been increasing rapidly over the last few years. Most of these methods are based on autoencoders and are therefore unsupervised. The nonnegative sparse autoencoder (NNSAE) has proved to be successful for unmixing. Such autoencoders satisfy the linear mixing model by having a linear decoder with nonnegative weights and by enforcing the abundance sum-to-one constraint (ASC) for abundances [19]. Examples of recent autoencoder methods are [20]–[25]. All recent autoencoder approaches, except for [26], process a single spectrum at a time and hence do not include any spatial information.

Incorporating spatial information in unmixing, specifically for endmember extraction, has mainly been done in three different ways. First, by developing new spatially oriented methods that provide some way of assessing the purity of a pixel or a neighborhood without using a geometrical method [27]. Examples of such methods are [28] and [29]. Second, by the refinement of spectral only methods, for example, using spatial priors to regularize or penalize solutions obtained by existing methods. Good example of this is the SUNSAL-TV method [5] and also [30]. An example of methods using spectral–spatial weighted sparse regression is described in [31]. Examples of how a TV regularization term can be used to exploit spatial homogeneity in HSIs while keeping sharp edges are given in [30] and [33]–[35]. Some examples of spatial regularization in NMF methods are [33], [36], and [37]. An example of a hybrid linear and nonlinear NMF unmixing method based on a spatial prior is given in [37].

Lastly, incorporating spatial information can be done by preprocessing the data before applying a spectral only method. A good example of this is [38], where for each pixel, a scalar spatially derived factor relating to the similarity of pixels lying within a certain spatial neighborhood is computed and used to weigh the importance of the spectral information in a pixel in terms of its spatial context. Superpixel methods are also examples of methods that use a superpixel segmentation method as a preprocessing technique before unmixing. Examples of superpixel methods are given in [39]–[41] and [42] which applies a spatial group sparsity regularization derived from segmentation.

C. Notation

In this article, the following notation will be used. The letter P denotes the number of pixels, B the number of bands, and R the number of endmembers. We write CONV n to refer to convolutional layer number n . We will use the notation $\mathbf{W}_{ij}, i, j = 1, \dots, f$ to denote the $B \times R$ matrices that are obtained by fixing the first two indices of \mathbf{W} to i and j . Often we will use a single index to index matrices and will use the convention shown in Table I for labeling them.

The remainder of this article is organized as follows. Section II-A describes the proposed CNN autoencoder

TABLE I
INDEXING CONVENTION USED FOR THE MATRICES \mathbf{M}_m AND \mathbf{W}_m . SHOWN IS AN EXAMPLE FOR A 3×3 FILTER. WE WILL OFTEN USE THE LETTER c FOR THE CENTER LOCATION, I.E., $\mathbf{W}_c = \mathbf{W}_5$ IN THIS EXAMPLE

	col. 1	col. 2	col. 3
row 1	\mathbf{W}_1	\mathbf{W}_4	\mathbf{W}_7
row 2	\mathbf{W}_2	\mathbf{W}_5	\mathbf{W}_8
row 3	\mathbf{W}_3	\mathbf{W}_6	\mathbf{W}_9

unmixing (CNNAEU) method. In Section III, experimental results using four real HSI data sets are given along with a comparison with four conventional and three deep learning state-of-the-art HU methods. Finally, in Section IV conclusions will be drawn.

II. PROBLEM FORMULATION AND METHOD

We are given P observed spectra, each having B bands. Having estimated the number of endmembers in the HSI as R , which can be done for example using [44] and [45], we assume a new spectral–spatial model

$$\mathbf{x}_p = \mathbf{Ms}_p + \sum_{i \in \mathcal{N}_p \setminus p} \mathbf{M}_i \tilde{\mathbf{s}}_i + \epsilon_p, \quad p = 1, \dots, P \quad (1)$$

where $\mathbf{s}_p \in \mathbb{R}_+^{R \times 1}$ is an abundance vector that sums to one, $\mathbf{M} \in \mathbb{R}_+^{B \times R}$ contains the endmembers in its columns, $\mathcal{N}_p \setminus p$ is an $f \times f$ neighborhood around p , with f an odd number, while excluding p (the center pixel), $\mathbf{M}_i \in \mathbb{R}_+^{B \times R}$, $\tilde{\mathbf{s}}_i \in \mathbb{R}_+^{R \times 1}$, $\mathbf{M}_i \tilde{\mathbf{s}}_i$ is the spectral contribution to \mathbf{x}_p from location i in its neighborhood, and ϵ_p is noise. We index the pixels in the neighborhood \mathcal{N} as shown in Table I.

This new model differs from the widely used LMM by allowing all pixels in an $f \times f$ neighborhood centered on a pixel to contribute to the pixel's reconstruction. In this article, we investigate the problem of estimating the endmember matrix \mathbf{M} and the abundances \mathbf{s}_p for every pixel in the HSI by interpreting the problem as blind unmixing, and solving it using a CNN autoencoder.

An autoencoder is a neural network that learns a representation of the HSI data. This means that the training is unsupervised, even though it might seem supervised, as it does not require labeled data. By introducing a hidden layer that acts as a bottleneck, the network learns a low-dimensional code for the input, and learns to reconstruct the input data from the code. The encoder is the part that performs this encoding, and the decoder is the part that reconstructs the input from the code. If the decoder is linear, the low-dimensional code for an HSI are the abundances and the weights of the decoder are the endmembers.

A. Estimation Method

The method is a CNN autoencoder that is trained on N patches, $\mathcal{B}^i = \{\mathbf{x}_1, \dots, \mathbf{x}_{n \times n}\}, i = 1, \dots, N$, and $\mathbf{x}_j \in \mathbb{R}^{B \times 1}$, from an HSI, $\mathbf{Y} \in \mathbb{R}^{D_1 \times D_2 \times B}$. We do not use biases, pooling layers, or upsampling, only nonstrided CONV layers that do not reduce the spatial resolution or the size of the input.

Fig. 1 shows a schematic of the method. The first layer in the network is the input layer. The second layer, CONV 1, is a 2-D convolutional layer with 48 feature maps, filter of size 3×3 ,

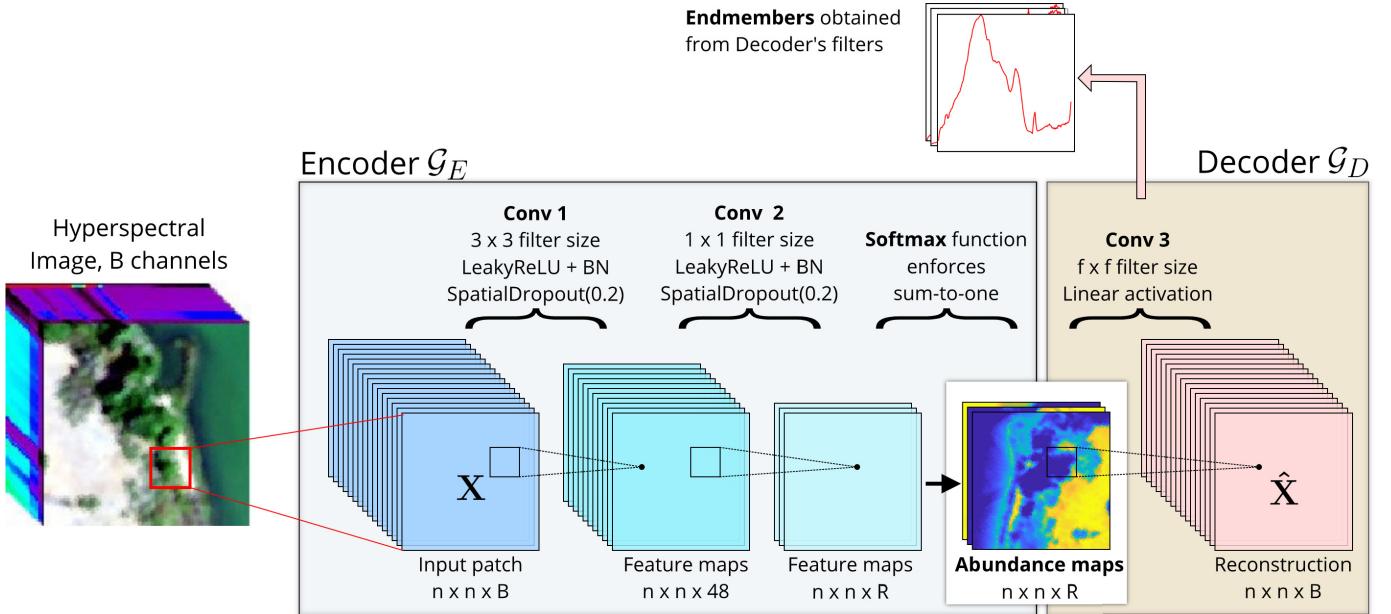


Fig. 1. Schematic of the method. The convolutional autoencoder is trained on patches from the HSI. It has three convolutional layers. The abundance maps of the input patches are the feature maps from the second convolutional layer. The endmembers are extracted from the filters of the last layer which has linear activation. BN stands for batch normalization.

and LeakyReLU activation. We apply batch normalization [45] after the convolutional layer to speed up the learning. Next, we apply spatial dropout, a version of dropout better suited to CNNs [46] to reduce overfitting and improve generalization, with dropout rate 0.2. This means that entire feature maps are randomly set to zero with a probability of 20%.

Next comes another convolutional layer, CONV 2, having R feature maps, filter size 1×1 , and LeakyReLU activation. Again we apply both batch normalization and spatial dropout with dropout rate 0.2 after the convolutional layer. This is the spectral bottleneck leading to the abundance maps. The next layer and the final layer in the encoding part of the autoencoder, is the layer that enforces the ASC at every pixel. It applies the softmax function pixelwise to the collection of output feature maps from the previous layer, but scales the values by a factor α first. We used $\alpha = 3.5$ in experiments. The output of this layer are feature maps that can be interpreted as the abundances that sum-to-one at every pixel.

The last layer in the network is the linear decoder layer. This is a convolutional layer with B feature maps, filter size $f \times f$ where f is an odd number, and linear activation. This layer reconstructs the input patch from the abundance patches resulting from the abundances-sum-to-one enforcing layer. If the filter size is 1×1 , it is easy to show that the weights, \mathbf{W} , of the filter is a $B \times R$ matrix which contains the endmembers as its columns. If the filter size is $f \times f$, we will get $f^2 B \times R$ matrices since the filter is an element of $\mathbb{R}^{f \times f \times R \times B}$. Furthermore, it is straightforward to rewrite the formula for the reconstruction of pixel p in a patch as

$$\hat{\mathbf{x}}_p = \sum_{m=1}^{f^2} \mathbf{W}_m \mathbf{h}_m,$$

where the single index m is indexing the locations in the $f \times f$ patch as shown in Table I. Thus, for every location in row i

and column j of the $f \times f$ neighborhood, there corresponds an element of $\mathbb{R}^{B \times R}$, that is, a matrix having the same dimensions as the endmember matrix. This expression can be rearranged to obtain

$$\hat{\mathbf{x}}_p = \left(\sum_{m=1}^{f^2} \mathbf{W}_m \right) \mathbf{h}_c + \sum_{\substack{m=1 \\ m \neq c}}^{f^2} \mathbf{W}_m (\mathbf{h}_m - \mathbf{h}_c).$$

By comparing to (1), we can identify

$$\begin{aligned} \mathbf{M} &= \sum_{m=1}^{f^2} \mathbf{W}_m \\ \mathbf{M}_i &= \mathbf{W}_i \\ s_p &= \mathbf{h}_c \\ \tilde{s}_i &= \mathbf{h}_i - \mathbf{h}_c. \end{aligned} \quad (2)$$

Thus, the abundance maps are obtained by encoding the HSI with the autoencoder. The endmembers are obtained directly from the weights of the decoder layer.

1) *Loss Function:* We use the spectral angle distance (SAD) for the fidelity term of the loss function for our autoencoder. The SAD between two vectors, \mathbf{x} and $\hat{\mathbf{x}}$, is given by

$$J_{\text{SAD}}(\mathbf{x}, \hat{\mathbf{x}}) = \arccos \left(\frac{\langle \mathbf{x}, \hat{\mathbf{x}} \rangle}{\|\mathbf{x}\|_2 \|\hat{\mathbf{x}}\|_2} \right). \quad (3)$$

The loss measures the discrepancy between the input patch and its reconstruction by the CNN. The loss for patch \mathcal{B}^i is calculated as

$$\mathcal{L}^{(i)} = \frac{1}{|\mathcal{B}^i|} \sum_{x_p \in \mathcal{B}^i} J_{\text{SAD}}(\mathbf{x}_p, \hat{\mathbf{x}}_p). \quad (4)$$

The loss for the training data set is then

$$\mathcal{L} = \sum_{i=1}^N \mathcal{L}^{(i)}. \quad (5)$$

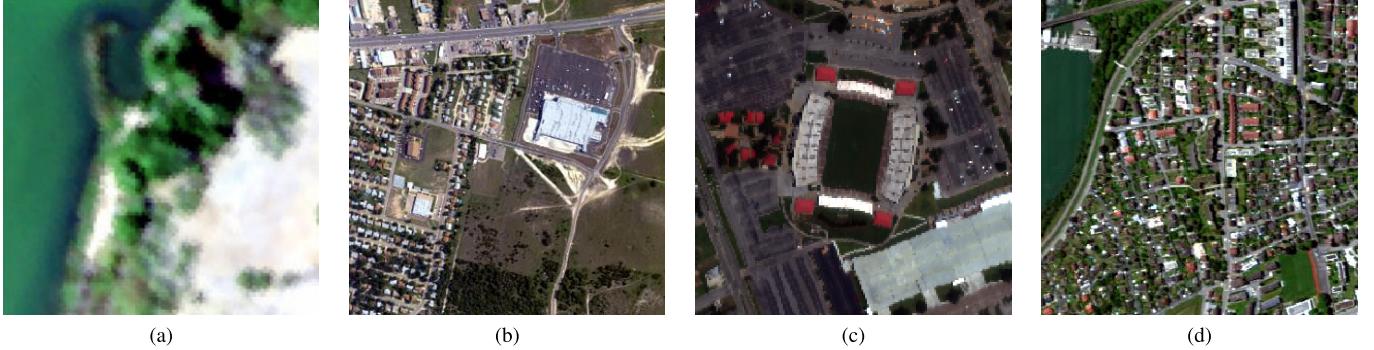


Fig. 2. Simulated and actual RGB images of the data sets used in the experiments (a) Samson—simulated. (b) Urban—simulated. (c) Houston—actual. (d) Apex—actual.

III. EXPERIMENTS

All experiments were performed using four real HSIs. The methodology for determining the reference endmembers is described in [47]. The data sets used in the experiments are as follows.

- 1) *Samson*: This hyperspectral data set is obtained by the SAMSON sensor and is a cropped 95×95 pixel image from a larger image and has 156 bands covering the 401–889 nm wavelength range. The image has the following endmembers: Water, Soil, and Tree.
- 2) *Urban*: This hyperspectral data set was obtained by the Hyperspectral Digital Image Collection Experiment (HYDICE) [48] sensor. The image is 307×307 pixel recorded at 210 bands covering the 400–2500 nm wavelength range. We use 162 bands in experiments after removing noisy and corrupted bands. Grass, Tree, Asphalt, and Roof were selected as the references for four endmembers.
- 3) *Houston*: This hyperspectral data set was used in the 2013 IEEE GRSS Data Fusion Contest. The image is cropped from a larger HSI acquired over the University of Houston campus, TX, USA, in June 2012. It has 144 bands covering the wavelengths of 380–1050 nm and the spatial resolution is 2.5 m. We use the $170 \times 170 \times 144$ subimage of the image centered on the Robertson stadium on the Houston Campus and surrounding area. We estimate four endmembers and use as reference endmembers the class averages of the 15 classification ground truth categories that come with the data set.
- 4) *Apex*: This hyperspectral data set obtained using APEX [49] sensor during an APEX flight campaign on a clear day in June 2011 and is a cropped image from a larger one. The flight altitude is 4600 m above sea level with a heading of 56° in the vicinity of Baden, Switzerland. We use the 300×300 subimage cropped from the original $1500 \times 1000 \times 285$ at location (70,650). There are 285 bands covering wavelengths from 413 to 2421 nm, all of which are usable. We estimate four endmembers and pick the reference endmembers manually from the data set.

Simulated and actual RGB images of the data sets used are shown in Fig. 2.

A. Methodology and Performance Metrics

To evaluate the proposed CNNAEU method, we quantify how well it performs in endmember extraction and how good the abundance maps are. We use the average SAD (mSAD) to measure how close the extracted endmembers are to the reference endmembers

$$\text{mSAD} = \frac{1}{R} \sum_{i=1}^R \arccos \left(\frac{\langle \hat{\mathbf{m}}_i, \mathbf{m}_i \rangle}{\|\hat{\mathbf{m}}_i\|_2 \|\mathbf{m}_i\|_2} \right) \quad (6)$$

where $\hat{\mathbf{m}}_i$ are the endmembers extracted by the method, and \mathbf{m}_i are the reference endmembers. The lower the mSAD, the higher the similarity. The metric we use to evaluate the obtained abundance maps is given by

$$\text{MSE} = \frac{1}{R} \sum_{i=1}^R \|\mathbf{S}_i - \hat{\mathbf{S}}_i\|^2 \quad (7)$$

where $\hat{\mathbf{S}}_i$ are the abundance fractions of all pixels for endmember i and \mathbf{S}_i are the reference abundance fractions. The quality of extracted endmembers weighs more heavily in our evaluation of the method than the quality of the abundance maps. This is because the abundance maps produced by our method tend to be very binary, that is abundances are either very low or very high, almost like classification maps. This happens because of batch normalization and the fact that we are using a ReLU like activation and the softmax function to enforce the ASC constraint.

Another potential issue regarding the abundance estimation is our use of SAD loss. This loss, although very good for endmember extraction, is not an optimal metric for data reconstruction since it is scale invariant. It is possible that SAD loss leads to higher variance in the abundance estimation than is explainable by considering only the variance in the quality of the extracted endmembers. Still, methods using SAD loss can achieve fairly good results for abundance estimation as the results of the MTAEU method demonstrate.

We can always get fairly good abundance maps from good endmembers using the method of fully constrained least squares (FCLS) [50], or using an autoencoder that has a fixed decoder, such as, endmembers, and does not use batch normalization. We did in fact additionally use a fully con-

TABLE II

BENCHMARK METHODS USED FOR COMPARISON IN EXPERIMENTS

Method	Description
$\ell_{1/2}$ -NMF	Sparsity constrained nonnegative matrix factorization [13]. A spectral only method.
SGSRNMF	Spatial group sparsity regularized nonnegative matrix factorization [43]. A spectral-spatial method.
SHDP	Sticky hierarchical Dirichlet process [18]. A spectral-spatial method.
uDAS	Untied denoising autoencoder with sparsity [24]. A spectral only method.
SNSA	Stacked nonnegative sparse autoencoder [52]. A spectral only method.
DAEU	Deep autoencoder unmixing [23]. A spectral only method.
MTLAEU	Multitask learning autoencoder unmixing [26]. A spectral-spatial method.

nected autoencoder to produce improved abundance maps for fixed endmembers extracted by CNNAEU.

To evaluate our method, we compare the average scores for our extracted endmembers and abundance maps to the results from seven state-of-the-art methods. We use the mSAD and MSE of 25 runs for all experiments and report the average values along with the standard deviation. We also report the average SAD from reference along with standard deviation for each individual endmember. The comparison methods are listed in Table II.

Methods in rows 2, 3, and 7 in the table are spectral–spatial methods, and the methods in rows 4–7 are based on autoencoders. The DAEU, MTLAEU, and CNNAEU methods were all initialized randomly, while all other methods are initialized or partially initialized using VCA.

B. Hyperparameter Settings

Table III lists the CNNAEU’s hyperparameters and their settings. Values of most hyperparameters were chosen using grid search and a single data set (Urban) with reference endmembers. The size of the filters in layer CONV 1 was chosen to be 3×3 as larger sizes make the abundance maps look a bit fussy. The size of filters in layer CONV 2 was chosen to be 1×1 because of similar considerations. We performed a sensitivity analysis of selected hyperparameters with respect to the average SAD. One hyperparameter at a time was increased and decreased by 5%, 10%, 15%, and 20%, respectively. The experiment was performed using the Urban data set, and ten runs were used to calculate the average SAD and MSE for each setting. The results, given in Table IV, show that the maximum deviation in average SAD is only 14% and the maximum deviation in average MSE is only 17%. This means that the chosen settings of the hyperparameters do not correspond to some unusual and deep local minima. Values of the parameters that are close to the original values yield results that are close to the score obtained for the original values. Also, an increase of 14% in the average SAD from the reference score would

TABLE III

HYPERPARAMETER SETTINGS USED IN THE EXPERIMENTS

Hyperparameter	Value used in experiments
Input patch size	40×40
Number of filters in CONV 1	48
Size of filters in CONV 1	3×3
Size of filters in CONV 2	1×1
Size of filters in CONV 3	11×11
Softmax scaling parameter α	3.5
Optimizer	RMSprop
Learning rate	0.0003
Batch size	15
Training Epochs	320

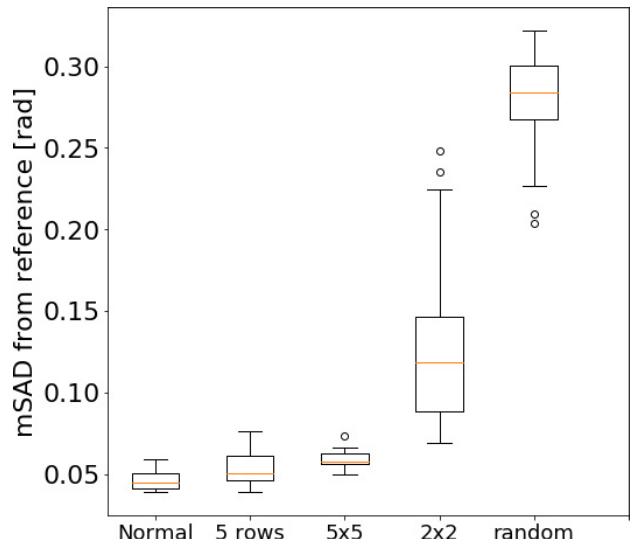


Fig. 3. Effect of gradual spatial degradation of the input HSI on the performance of the method.

not affect the ranking of our method for any data sets. Since the number of epochs is fixed, we allowed the number of patches used in training to vary between data sets as their sizes are different and we wish to avoid overtraining on smaller data sets. For our experiments, we determined the number of patches for the Urban data set using grid search and then for the other data sets we scaled this number by the ratio of the product of the number of columns, rows, and bands for the data set and the corresponding product for the Urban data set.

C. Spatiality

To confirm if the method is spatial in nature, we performed an experiment where we investigate the performance of the method with increasing spatial degradation of an HSI. If the technique is utilizing the spatial structure of the HSI to perform the unmixing, then gradually destroying the spatial structure by permuting areas in the HSI should result in the method performing increasingly worse when holding all hyperparameters constant.

Fig. 3 shows the result of this experiment for the Urban data set as a boxplot. There are 25 runs behind every setting. We used patch size of 40×40 with 200 training patches in

TABLE IV

SENSITIVITY TO HYPERPARAMETER VALUES. THE NUMBERS ARE THE CHANGE IN AVERAGE SAD SCORE WHEN A HYPERPARAMETER IS INCREASED OR DECREASED BY THE VALUE DISPLAYED IN THE TOP ROW OF THE TABLE. THE NUMBERS IN PARENTHESIS ARE THE CORRESPONDING CHANGE IN THE AVERAGE MSE SCORE OF THE ABUNDANCE MAPS. THE AVERAGE SAD SCORE IS THE AVERAGE SAD OF EXTRACTED ENDMEMBERS FROM REFERENCE ENDMEMBERS OF TEN RUNS FOR EACH HYPERPARAMETER CONFIGURATION. THE AVERAGE MSE IS THE AVERAGE MSE OF EXTRACTED ABUNDANCE MAPS FROM REFERENCE MAPS OF TEN RUNS FOR EACH HYPERPARAMETER CONFIGURATION

Hyperparameter	-5%	+5%	-10%	10%	-15%	+15%	-20%	+20%
Batch size	7.4% (5.5%)	2.6% (2.5%)	10.7% (3.5%)	6.2% (2.4%)	13.6% (12.4%)	5.8% (5.5%)	3.5% (10.2%)	7.5% (10.0%)
Learning rate	3.3% (7.2%)	8.9% (10.8%)	7.2% (3.9%)	12.2% (7.0%)	4.4% (2.0%)	11.7% (5.8%)	-1.4% (8.9%)	12.5% (11.7%)
Number of patches	7.7% (3.4%)	5.4% (4.5%)	6.2% (11.2%)	14.2% (12.2%)	7.7% (16.6%)	15.5% (12.2%)	10.8% (10.7%)	10.0% (6.9%)
#filters in CONV 1	6.4% (5.2%)	5.5% (1.3%)	9.0% (2.4%)	10.0% (-0.5%)	8.3% (4.7%)	10.1% (8.6%)	13.5% (12.5%)	13.7% (5.7%)
Patch size	6.6% (1.6%)	4.5% (-0.5%)	11.2% (1.1%)	9.4% (5.9%)	9.8% (11.2%)	17.1% (8.8%)	4.3% (11.9%)	4.9% (8.4%)

TABLE V

URBAN DATA SET. MEAN SAD AND STANDARD DEVIATION COMPARISON FOR ENDMEMBER EXTRACTION IN RADIAN. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Asphalt	0.1161	0.0233	0.2446	0.0204	0.2658	0.0751	0.2912	0.0003	0.1805	0.0321	0.0720	0.0235	0.0843	0.0046	0.0575	0.0058
Grass	0.5730	0.3226	1.3006	0.0444	0.5524	0.3172	1.2165	0.0042	1.1609	0.1515	0.0769	0.0285	0.0421	0.0036	0.0366	0.0047
Tree	0.1571	0.0921	0.0967	0.0113	0.0777	0.0171	0.0777	0.0001	0.1496	0.0385	0.0707	0.0291	0.0539	0.0039	0.0321	0.0039
Roof	0.4891	0.2065	0.1916	0.0862	0.4117	0.1720	0.0923	0.0005	0.1700	0.0011	0.2155	0.0845	0.0415	0.0045	0.0332	0.0066
Average	0.3338	0.0644	0.4584	0.0148	0.3269	0.0555	0.4194	0.0011	0.4153	0.0403	0.1088	0.0234	0.0555	0.0019	0.0398	0.0030

TABLE VI

SAMSON DATA SET. MEAN SAD AND STANDARD DEVIATION COMPARISON FOR ENDMEMBER EXTRACTION IN RADIAN. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Soil	0.4013	0.2097	0.0086	0.0001	0.2147	0.3299	0.2493	0.3122	0.0312	0.0014	0.0497	0.0216	0.0225	0.0060	0.0373	0.0210
Tree	0.0636	0.0379	0.0395	0.0019	0.0375	0.0004	0.0750	0.0146	0.0547	0.0049	0.0553	0.0121	0.0371	0.0028	0.0397	0.0038
Water	0.1319	0.1005	0.0923	0.0024	0.2064	0.0916	0.2844	0.0677	0.1405	0.0117	0.0530	0.0208	0.0338	0.0031	0.0430	0.0092
Average	0.1989	0.1034	0.0468	0.0003	0.1527	0.1390	0.2029	0.1218	0.0755	0.0051	0.0527	0.0117	0.0311	0.0017	0.0400	0.0067

the training data set and trained the network for 100 epochs. The categorical labels have the following meaning.

- 1) *Normal*: This is using the original data set unchanged.
- 2) *Five Rows*: This data set is obtained by permutation of the rows of the image in blocks of five.
- 3) 5×5 : This data set is obtained by dividing the original image into 5×5 patches and randomizing their locations.
- 4) 2×2 : This data set is obtained by dividing the original image into 2×2 patches and randomizing their locations.
- 5) *Random*: This data set is obtained by totally randomizing every pixel location by a random permutation of the array of spectra before reshaping it into a 2-D image.

As can be seen clearly from the figure, the performance, as measured by the mSAD in radians, decreases with increasingly fine grained permutations of the pixels of the original image. When all spatial structure has been completely destroyed, the mSAD is between 0.25 and 0.30 radians after 100 epochs when compared with around 0.05 radians for the original image. This shows that the CNNAEU method is indeed spatial in nature.

D. Matrices W_{ij}

In this section, we will investigate the matrices W_{ij} , $i, j = 1, \dots, f$, that are obtained from the decoder's filter by fixing the first two indices to i, j . Fig. 4 shows plots of

the contents of the filter for the Urban data set for all four endmembers and all 25 locations for a filter having size 5×5 .

Looking at Fig. 4, there are two things that stand out. First, the relative scale of the plots when compared with the center plot, and second the variations in the shape of the plots; the filter for certain endmembers seems to contain plots of more than one different endmembers. Based on our experimentation, we can summarize the behavior of the filter regarding the two above-mentioned outstanding features with regard to the spatial structure of the HSI as follows.

- 1) The relative scale of surrounding locations is entirely dependent upon the amount of spatial correlations in an HSI. If there are no spatial correlations, only the center location will be nonzero and it will contain the estimated endmember.
- 2) If an endmember is often considerably mixed with another, this will be reflected in the filter plots of the endmember. The plots in locations other than the central location can look similar to the other endmember.

Fig. 5 illustrates observation number 1 here above. When the spatial structure of the Urban HSI has been destroyed by randomly permuting the pixels, the scale of the plots in noncentral locations goes to zero as can be seen in Fig. 5(b). However, by selecting a subimage of Urban that is more homogeneous with respect to the Tree and Grass endmembers than the original image, the plots in Fig. 5(c) are obtained.

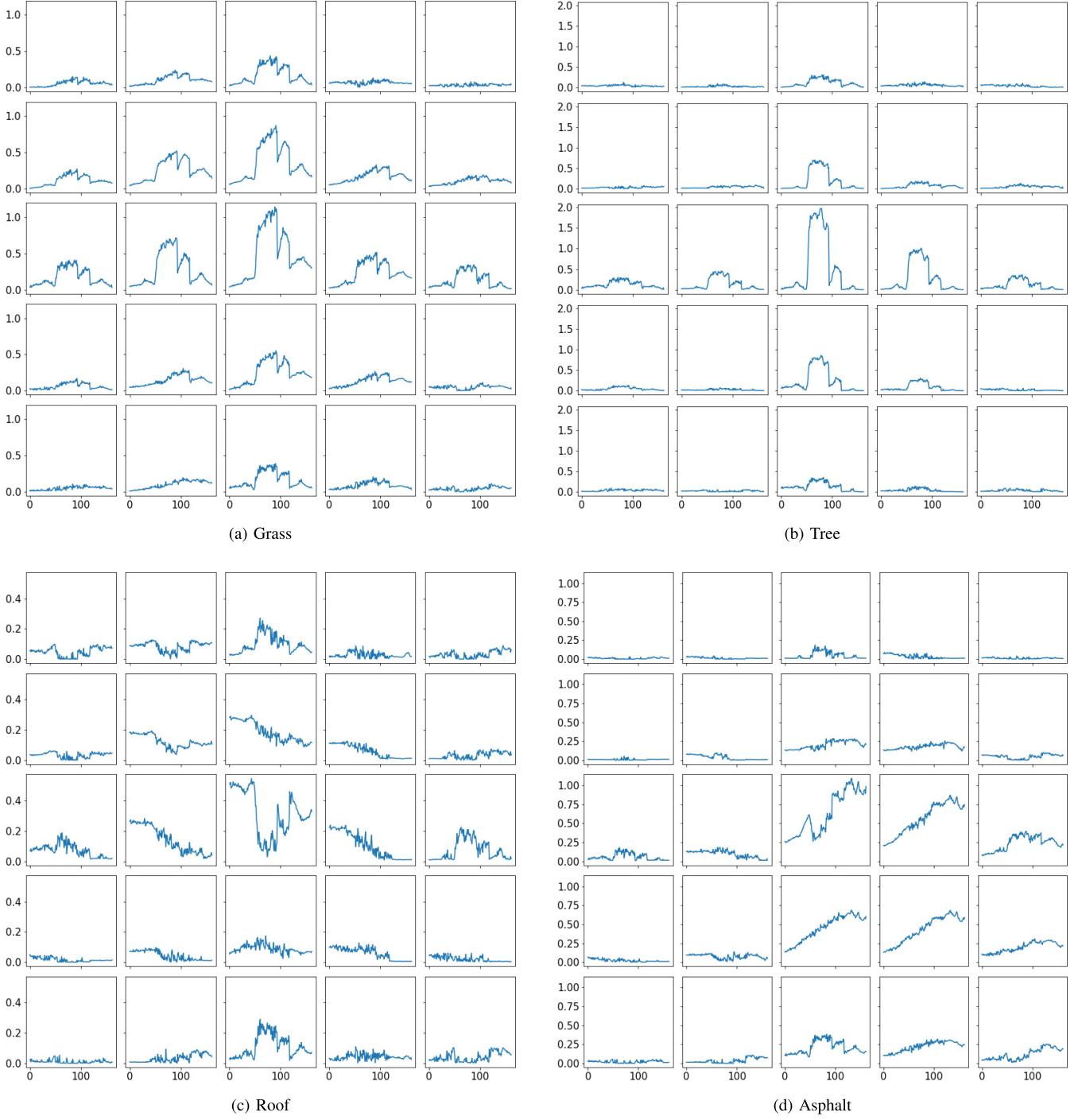


Fig. 4. Plots of the endmember matrices of the 5×5 filter for the Urban data set. (a) Grass. (b) Tree. (c) Roof. (d) Asphalt.

The relative scale of noncentral locations has increased with the increased spatial correlations in the scene.

Fig. 4(c) illustrates observation number 2 above well. It is the filter plot for the Roof endmember which, in the Urban HSI, is very often mixed with the Trees and Grass endmembers. In locations (1, 3), (3, 1), (5, 3), and (3, 5), the plots contain nothing that resembles the Roof endmember, but looks very similar to the Tree/Grass endmembers. The same can also be observed in the plot for the Asphalt endmember in Fig. 4(d), which is also often mixed with vegetation

endmembers in the Urban HSI. The same locations show something that looks similar to the vegetation endmembers.

The spatial nature of the reconstruction of a pixel's spectrum by the CNN autoencoder as described by (1) makes the two observations above very reasonable. For the case with no spatial structure, then generally \mathbf{h}_m is very different from \mathbf{h}_c , while at the same time the network cannot rely on the neighboring abundances of a pixel for its reconstruction, forcing it to make the \mathbf{W}_m close to zero for locations other than the center.

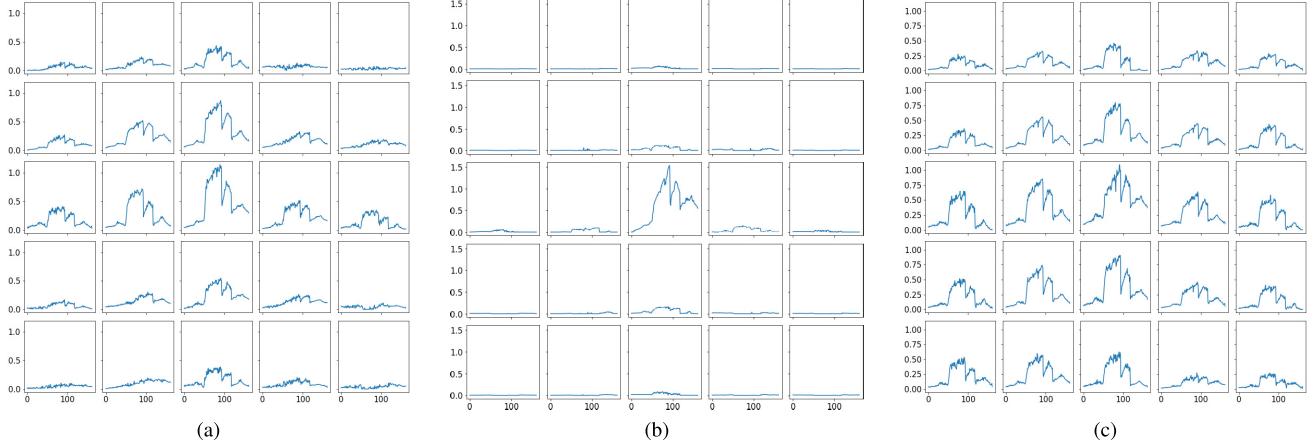


Fig. 5. “Grass” endmember filter for different degrees of spatial correlations in the Urban data set. (a) Original. (b) Randomized. No spatial correlations. (c) More homogenous subimage.

TABLE VII

HOUSTON DATA SET. MEAN SAD AND STANDARD DEVIATION COMPARISON FOR ENDMEMBER EXTRACTION IN RADIANS. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
parking_lot1	0.0665	0.0029	0.0939	0.0040	0.0751	0.0024	0.0971	0.0953	0.1130	0.1100	0.1263	0.0352	0.0427	0.0090	0.0291	0.0095
parking_lot2	0.7426	0.0910	0.1347	0.0018	0.6988	0.0255	0.8269	0.0366	0.3039	0.1492	0.2798	0.1588	0.0786	0.0128	0.0775	0.0070
running_track	0.2678	0.0696	0.4289	0.0034	0.2323	0.0013	0.4958	0.1987	0.1550	0.0530	0.1140	0.0389	0.2894	0.0886	0.0671	0.0767
grass_healthy	0.1366	0.0252	0.0931	0.0010	0.1585	0.0283	0.2485	0.0174	0.1080	0.0560	0.0937	0.0825	0.0642	0.0104	0.0272	0.0062
Average	0.3034	0.0305	0.1877	0.0010	0.2912	0.0082	0.4171	0.0568	0.1700	0.0388	0.1534	0.0353	0.1187	0.0226	0.0502	0.0193

Now, let us consider the case of an HSI that is heavily correlated. In this case, the network can rely on neighboring pixels for the reconstruction of a pixel. However, we can generally expect \mathbf{h}_m to be close to \mathbf{h}_c so the their difference will be small in magnitude, forcing the network to increase the norms of \mathbf{W}_m for locations other than the center. This means that the relative size of the plots of \mathbf{W}_{ij} will become similar and large when compared to the center location, as Fig. 5(c) is showing.

It is clear that a decoder filter of size $f \times f$ can give some qualitative information about the HSI being unmixed. Also, having f larger than 1 does indeed give slightly better estimation of endmembers.

E. Endmembers

In this section, we will focus on the extracted endmembers from the experiments. Table V shows the mSAD of the extracted endmembers along with standard deviation from 25 runs, along with the average SAD from references for individual endmembers for the Urban data set for all the methods. Fig. 6 shows the extracted endmembers for all the methods for the Urban data set with four reference endmembers. The blue curves are the extracted endmembers from 25 runs while the red curves are the reference endmembers.

As can be seen from the figure, the MTAEU and CNNAEU methods are in a class of their own for this data set. They are the only methods that have both good performance and excellent consistency. The proposed method actually has better SAD score and less variance of the two.

The SNSA method has a very low variance also, but its solution is not good. All the methods except for the DAEU, MTAEU, and CNNAEU have trouble with estimating the Grass endmember correctly, which is a bit surprising since it is a very well represented endmember in the data set. The NMF- $\ell_{1/2}$ does the best job aside from the aforementioned methods.

Table VI shows the mSAD of the endmembers extracted by all the methods for the Samson data set along with the average SAD from references for individual endmembers. Fig. 7 shows the extracted endmembers along with the reference. It can be seen from the figure that most of the methods do not have much trouble with this data set. The method that has the best SAD score is the MTAEU method and it also has the second least variance. This is the only data set where we had to use small patch size in order to get good and stable results. The data set itself is also very small, only 95×95 pixels, which could be the reason why a higher number of small patches gives better results than having fewer and much larger patches in the training data set.

Table VII shows the mSAD of the endmembers extracted by all the methods for the Houston data set along with the average SAD from references for individual endmembers. Fig. 8 shows all the extracted endmembers along with the reference. Looking at the figure, one can see that the proposed method has again both very good endmembers and very good consistency and low variance. Only in one run does it fail to extract the “running_track” endmember correctly and instead produces a variant of the “grass_healthy” endmember. The MTAEU method has some trouble with the consistency of

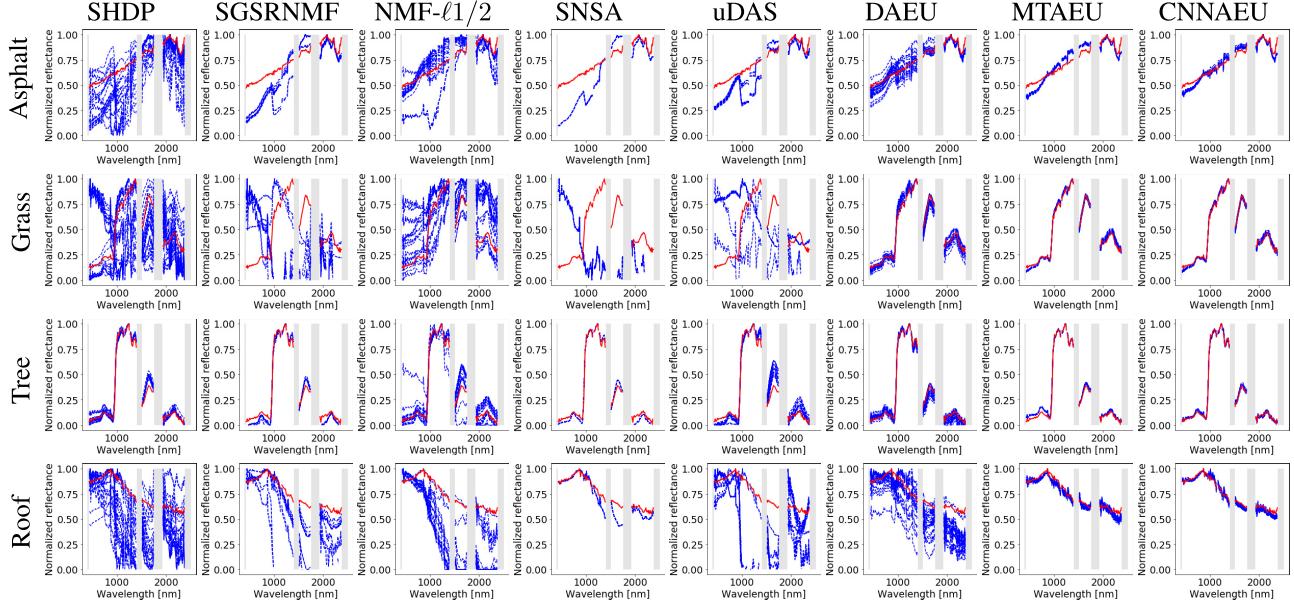


Fig. 6. Urban data set. Endmembers extraction for comparison. Reference endmembers in red.

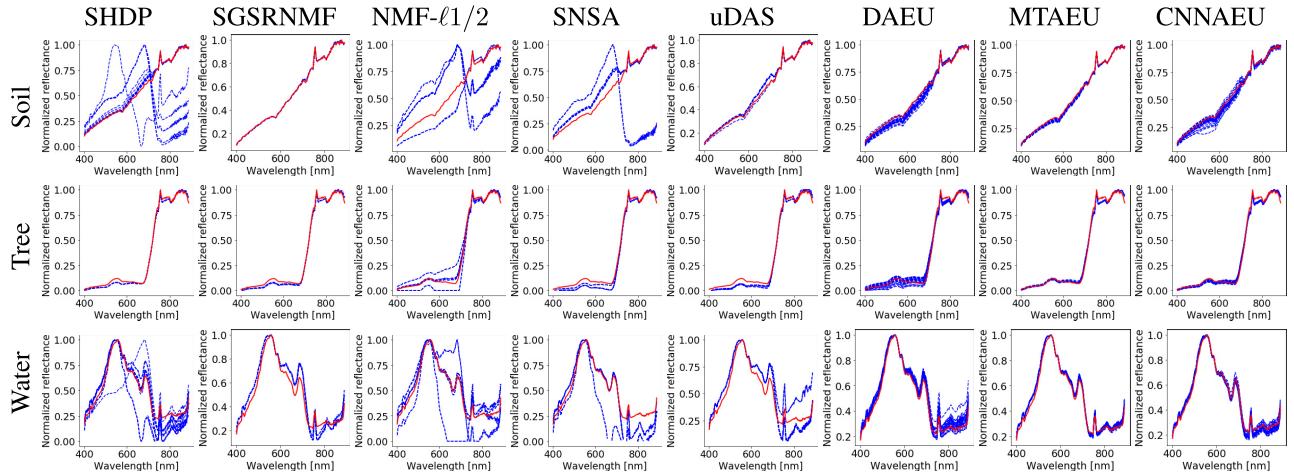


Fig. 7. Samson data set. Endmembers extraction for comparison. Reference endmembers in red.

TABLE VIII

APEX DATA SET. MEAN SAD AND STANDARD DEVIATION COMPARISON FOR ENDMEMBER EXTRACTION IN RADIANS. BEST RESULTS IN BOLD

	NMF-ℓ1/2		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Road	0.5265	0.0484	0.1779	0.1707	0.4123	0.1293	0.6324	0.1790	0.3109	0.1021	0.3637	0.1113	0.1126	0.0403	0.0587	0.0196
Water	0.4725	0.0717	0.3619	0.0917	0.6068	0.1956	0.5669	0.0909	0.5195	0.1595	0.1061	0.0245	0.1365	0.0163	0.0417	0.0037
Roof	0.1301	0.0175	0.1327	0.0794	0.1960	0.0631	0.0946	0.0604	0.1289	0.0483	0.2320	0.1731	0.5460	0.2461	0.1233	0.0060
Grass	0.1036	0.0147	0.0574	0.0082	0.0947	0.0270	0.0607	0.0024	0.0984	0.0469	0.1193	0.0369	0.0992	0.0076	0.0621	0.0038
Average	0.3081	0.0270	0.1825	0.0788	0.3274	0.0690	0.3387	0.0525	0.2644	0.0623	0.2053	0.0343	0.2236	0.0711	0.0714	0.0045

the endmember “running_track” and often finds a variant of the “grass_healthy” endmember instead, most likely it is the “grass_stressed” endmember. The DAEU method has high variance for this data set and often gets a bad solution for the “parking_lot2” endmember. It is very correlated with the “parking_lot2” endmember which could be an explanation for this behavior. The NMF methods generally have trouble with the “running_track” endmember.

Table VIII shows the mSAD of the extracted endmembers for all the methods for the Apex data set along with the average SAD from references for individual endmembers. Fig. 9 shows all the extracted endmembers along with the reference. Herein the proposed method achieves the best SAD score and has the best consistency and lowest variance by far of all the methods. The MTAEU method often incorrectly estimates the roof endmember and extracts an endmember corresponding to

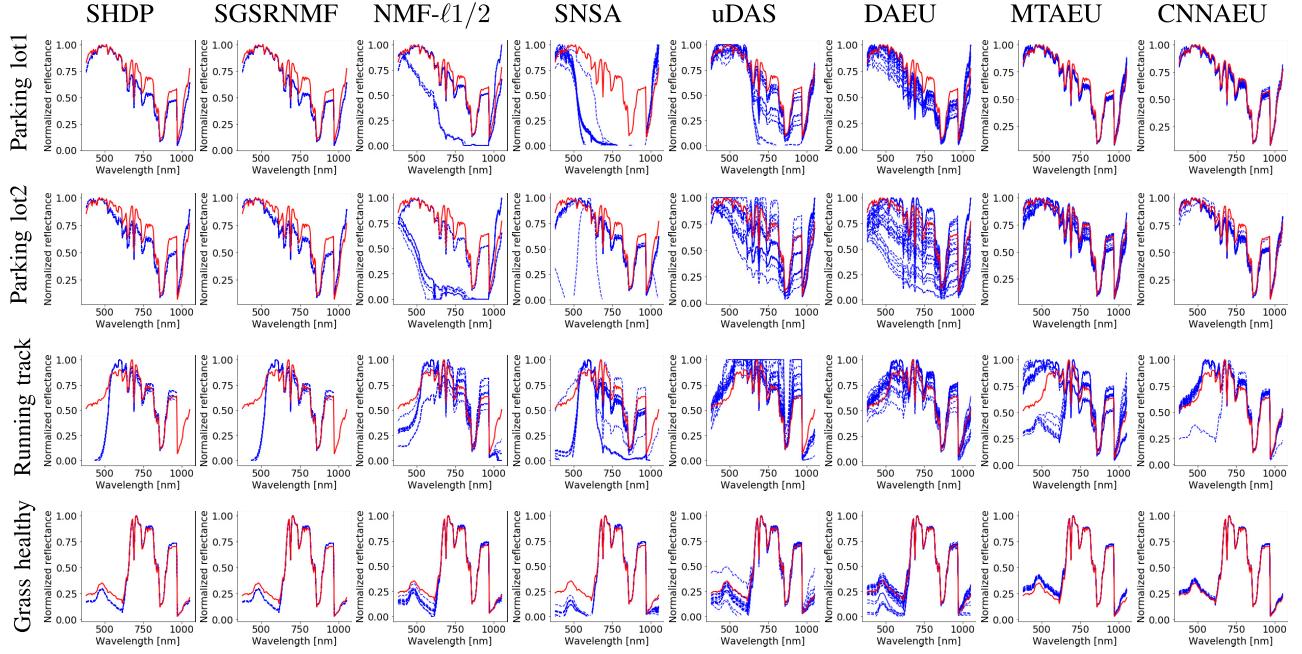


Fig. 8. Houston data set. Endmembers extraction for comparison. Reference endmembers in red.

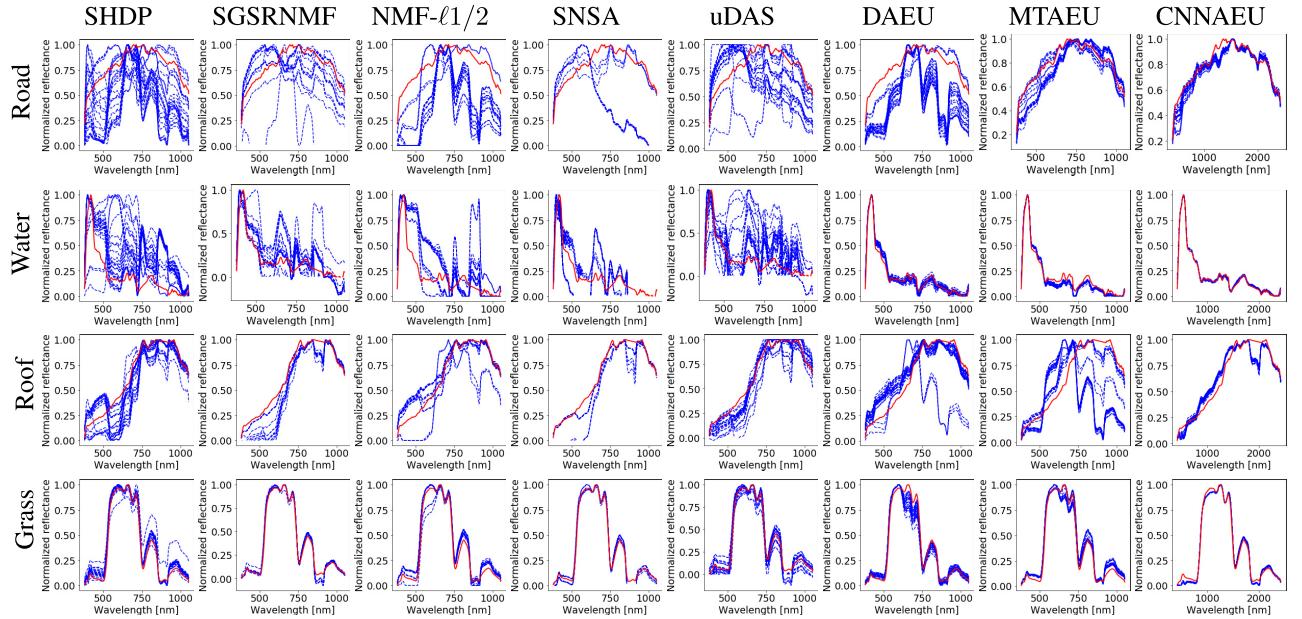


Fig. 9. Apex data set. Endmembers extraction for comparison. Reference endmembers in red.

trees. There are a lot of trees in the image and often obscuring roofs partially so there are a lot of mixed pixels containing both the tree and the roof endmember.

The DAEU method has less trouble with the roof endmember but is often estimating the tree endmember instead of the road endmember. There are also a lot of mixed pixels containing both these endmember in the scene. It is somewhat surprising how much trouble all the other methods, uDAS in particular, have with the water endmember. This endmember is abundant in the scene and its variability is not great.

Regarding the quality of extracted endmembers and method consistency it is clear that the proposed method ranks first

among all the methods in the experiments. It seems that the spatial aspect of the method greatly reduces the variance of the method and improves its consistency. It outperforms the MTAEU method, which is also a spectral–spatial exhibiting low variance and good consistency, in all the data sets except for Samson.

F. Abundance Maps

As was mentioned at the start of this section, the abundance maps produced by CNNAEU are very intense looking, that is both very sparse and binary. It is possible to extend the method so it refines the abundance maps after extracting the

TABLE IX

URBAN DATA SET. MEAN MSE AND STANDARD DEVIATION COMPARISON FOR ABUNDANCE MAPS EXTRACTION. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU		CNNAEU2	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Asphalt	0.1733	0.0013	0.0816	0.0058	0.0909	0.0144	0.1129	0.0005	0.1013	0.0036	0.0307	0.0172	0.0230	0.0010	0.0660	0.0041	0.0156	0.0016
Grass	0.2035	0.0039	0.1995	0.0103	0.1480	0.0724	0.1681	0.0005	0.1872	0.0381	0.0556	0.0258	0.0225	0.0016	0.0868	0.0071	0.0158	0.0016
Tree	0.1886	0.0032	0.0715	0.0171	0.0833	0.0235	0.1260	0.0007	0.0916	0.0103	0.0391	0.0330	0.0068	0.0009	0.0437	0.0073	0.0073	0.0015
Roof	0.0572	0.0006	0.0358	0.0018	0.0745	0.0569	0.0433	0.0002	0.0385	0.0012	0.0189	0.0131	0.0079	0.0008	0.0284	0.0067	0.0073	0.0015
Average	0.1557	0.0004	0.0971	0.0085	0.0992	0.0204	0.1126	0.0002	0.1046	0.0071	0.0360	0.0147	0.0150	0.0008	0.0562	0.0049	0.0115	0.0010

TABLE X

SAMSON DATA SET. MEAN MSE AND STANDARD DEVIATION COMPARISON FOR ABUNDANCE MAPS EXTRACTION. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU		CNNAEU2	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Soil	0.1400	0.0622	0.0316	0.0004	0.0814	0.0288	0.1251	0.0200	0.0651	0.0079	0.0439	0.0450	0.0076	0.0014	0.1068	0.0600	0.0765	0.0676
Tree	0.1045	0.0192	0.0574	0.0031	0.0623	0.0145	0.1347	0.0084	0.0666	0.0053	0.0198	0.0103	0.0037	0.0007	0.0899	0.0558	0.0631	0.0707
Water	0.0658	0.0417	0.1227	0.0034	0.1559	0.0090	0.1691	0.0086	0.1674	0.0067	0.0239	0.0471	0.0029	0.0010	0.0376	0.0982	0.0166	0.0133
Average	0.1034	0.0323	0.0706	0.0021	0.0999	0.0120	0.1430	0.0066	0.0997	0.0037	0.0292	0.0311	0.0048	0.0008	0.0781	0.0576	0.0521	0.0449

TABLE XI

HOUSTON DATA SET. MEAN MSE AND STANDARD DEVIATION COMPARISON FOR ABUNDANCE MAPS EXTRACTION. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU		CNNAEU2	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
parking_lot1	0.1366	0.0163	0.2799	0.0003	0.1881	0.0079	0.2455	0.0038	0.2428	0.0459	0.1903	0.0461	0.1558	0.0821	0.2178	0.0685	0.1196	0.0926
parking_lot2	0.0997	0.0055	0.2253	0.0004	0.1362	0.0049	0.1554	0.0125	0.2218	0.0394	0.1665	0.0692	0.2431	0.0968	0.2034	0.0701	0.1187	0.0810
running_track	0.0183	0.0039	0.0359	0.0009	0.0214	0.0012	0.0629	0.0172	0.0319	0.0051	0.0834	0.0795	0.1610	0.1094	0.0669	0.0153	0.0090	0.0064
grass_healthy	0.0059	0.0027	0.0040	0.0001	0.0085	0.0007	0.1235	0.0117	0.0452	0.0794	0.0266	0.0195	0.0151	0.0101	0.0314	0.0111	0.0067	0.0011
Average	0.0651	0.0050	0.1363	0.0002	0.0885	0.0027	0.1468	0.0025	0.1354	0.0245	0.1167	0.0335	0.1437	0.0537	0.1299	0.0348	0.0635	0.0441

TABLE XII

APEX DATA SET. MEAN MSE AND STANDARD DEVIATION COMPARISON FOR ABUNDANCE MAPS EXTRACTION. BEST RESULTS IN BOLD

	NMF- $\ell_{1/2}$		SGSRNMF		SHDP		SNSA		uDAS		DAEU		MTAEU		CNNAEU		CNNAEU2	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Road	0.2926	0.0385	0.1027	0.0674	0.1720	0.0455	0.0973	0.0016	0.1082	0.0801	0.1830	0.0226	0.1457	0.0523	0.1458	0.0196	0.0575	0.0339
Water	0.1578	0.0121	0.1646	0.0231	0.1270	0.0302	0.1862	0.0072	0.2319	0.0401	0.1074	0.0054	0.0790	0.0048	0.1058	0.0181	0.0808	0.0098
Roof	0.0510	0.0152	0.0461	0.0086	0.0363	0.0106	0.0809	0.0003	0.0567	0.0056	0.0809	0.0486	0.1267	0.0543	0.0959	0.0185	0.0448	0.0142
Grass	0.0199	0.0226	0.0084	0.0224	0.0204	0.0214	0.1091	0.0052	0.0284	0.0233	0.0443	0.0052	0.0432	0.0140	0.0647	0.0070	0.0278	0.0055
Average	0.1303	0.0096	0.0805	0.0255	0.0889	0.0180	0.1184	0.0026	0.1063	0.0254	0.1039	0.0094	0.0987	0.0242	0.1031	0.0111	0.0527	0.0142

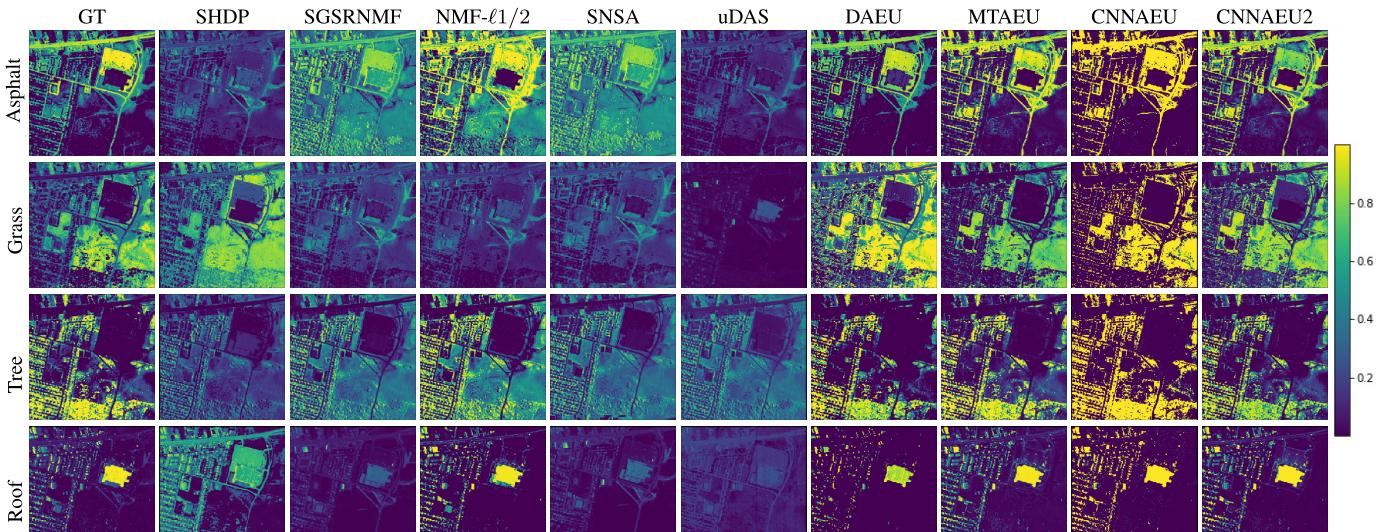


Fig. 10. Urban data set. Abundance maps extraction comparison and reference maps in the first column.

endmembers. This could be done using an autoencoder in serial with the unmixing one, which has its decoder's weights set to be the extracted endmember matrix and made nontrainable so they are fixed, and does not use batch normalization layers.

By training it for a few epochs, it will produce abundance maps which are not nearly as binary looking.

We did this to improve the abundance map results, using a fully connected autoencoder with a single hidden layer and

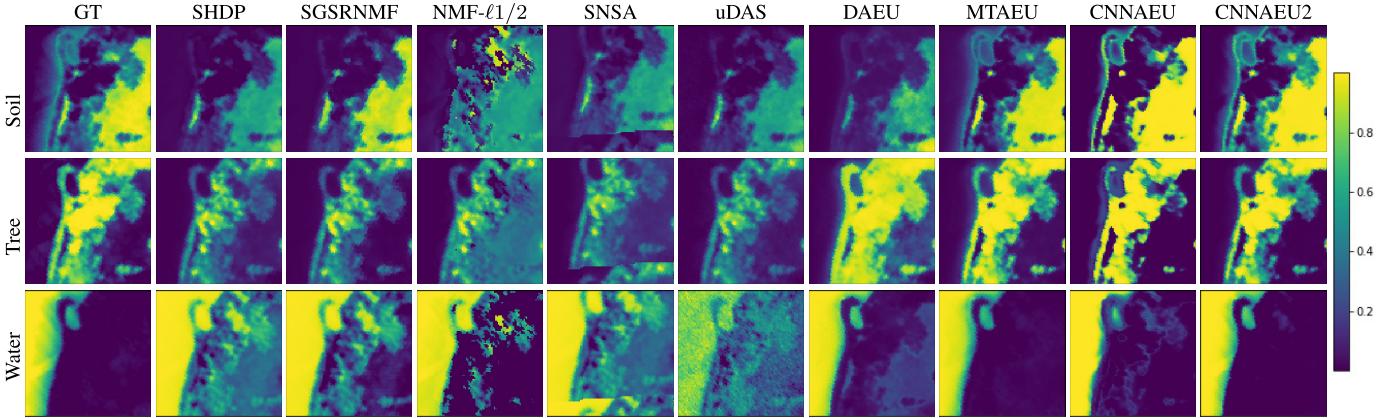


Fig. 11. Samson data set. Abundance maps extraction comparison and reference maps in the first column.

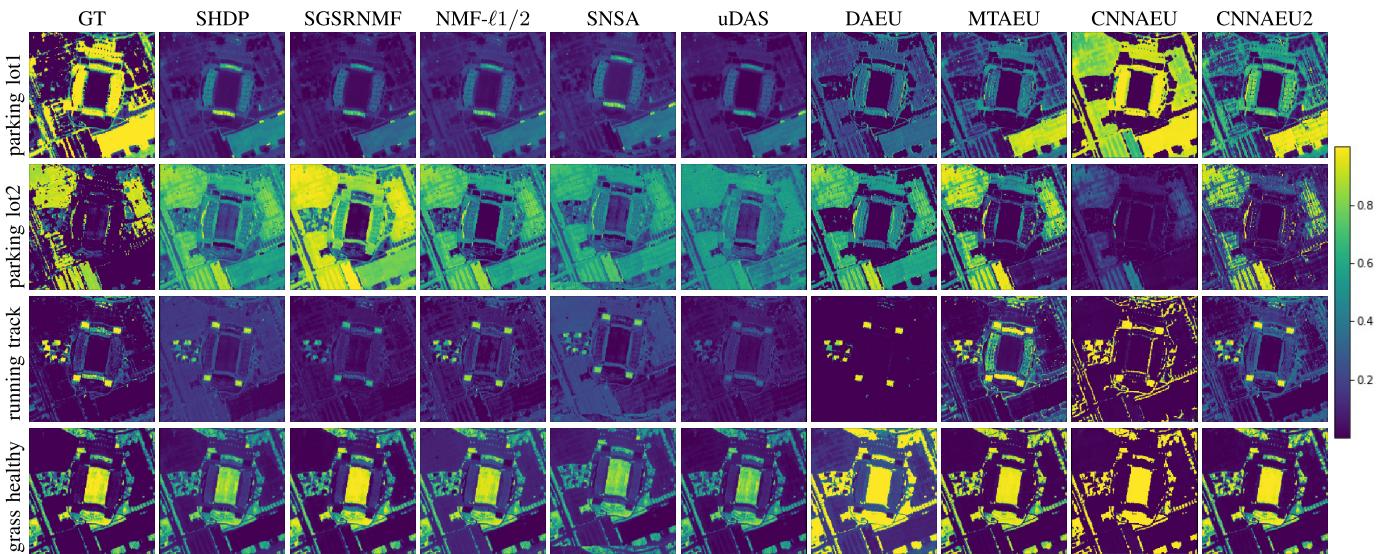


Fig. 12. Houston data set. Abundance maps extraction comparison and reference maps in the first column.

enforcing the ASC constraint using the softmax function. The resulting method is named CNNAEU2 and we will report MSE from the reference abundance maps for both CNNAEU and CNNAEU2.

In Table IX, the mean MSE and standard deviation comparison for abundance maps extraction for the Urban data set with four endmembers are given. Fig. 10 shows examples of the extracted abundance maps by all the methods for the same data set. The figure shows well the issue with the abundance maps produced by the proposed method. It also shows the refined maps by the CNNAEU2 method.

The MTAEU method achieves the lowest MSE score with a very low variance, and is closely followed by the CNNAEU2 method which also has very small variance. The maps produced by CNNAEU rank third (excluding CNNAEU2). As can be seen from the figure, even though they are intense, they are still good maps and could actually be used for classification.

Table X shows the MSE from reference abundance maps for the Samson data set for all the methods. Fig. 11 shows examples of the extracted abundance maps by all the methods

for the same data set. MTAEU scores best of all the methods again followed closely by CNNAEU2. It is interesting to see how high the variance is for the CNNAEU method. It does not reflect the low variance of the SAD score for the extracted endmembers. Of the other spectral-spatial methods, SGSRNMF achieves the best score.

Table XI shows the MSE of abundances from the reference abundance maps for the Houston data set for all the methods. Fig. 12 shows examples of the extracted abundance maps by all the methods for the Houston data set. On this data set, it is CNNAEU2 that achieves the best score. Second best is NMF- $\ell_{1/2}$ and third best is DAEU. The SHDP and SGSRNMF and the SNSA methods have all very low variance when compared with the other methods.

Table XII shows the MSE from the reference abundance maps for the Apex data set for all the methods. Fig. 13 shows examples of the extracted abundance maps by all the methods for the Apex data set. The maps produced by the CNNAEU2 method achieve the lowest MSE score for this data set. All the other methods have average MSE close to 0.1.

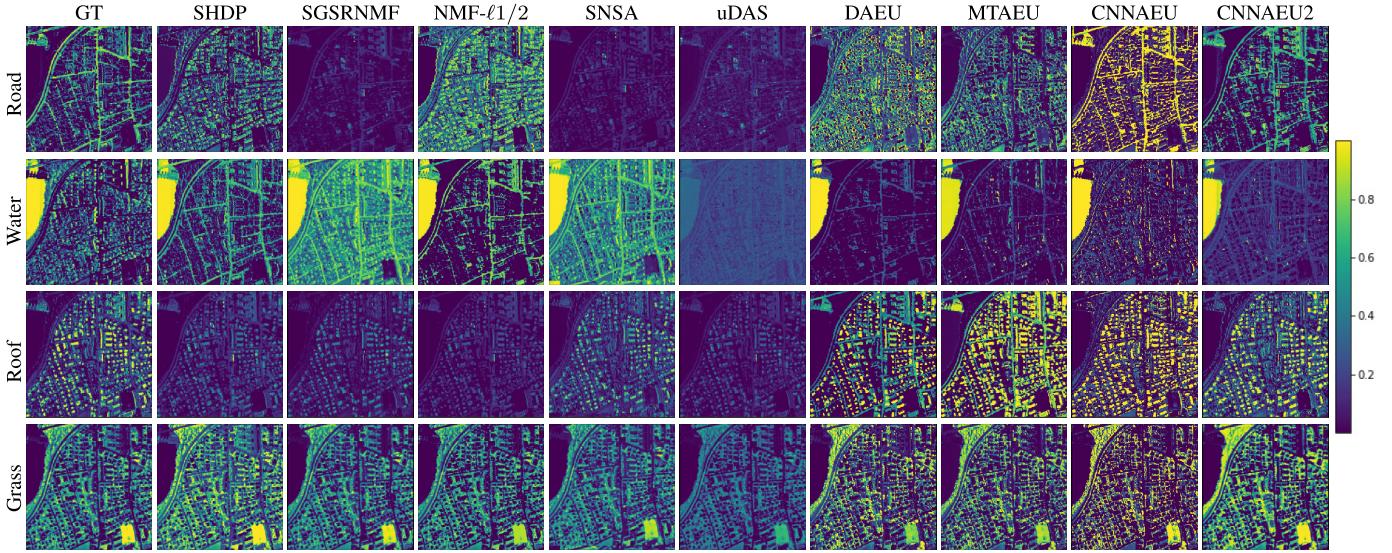


Fig. 13. Apex data set. Abundance maps extraction comparison and reference maps in the first column.

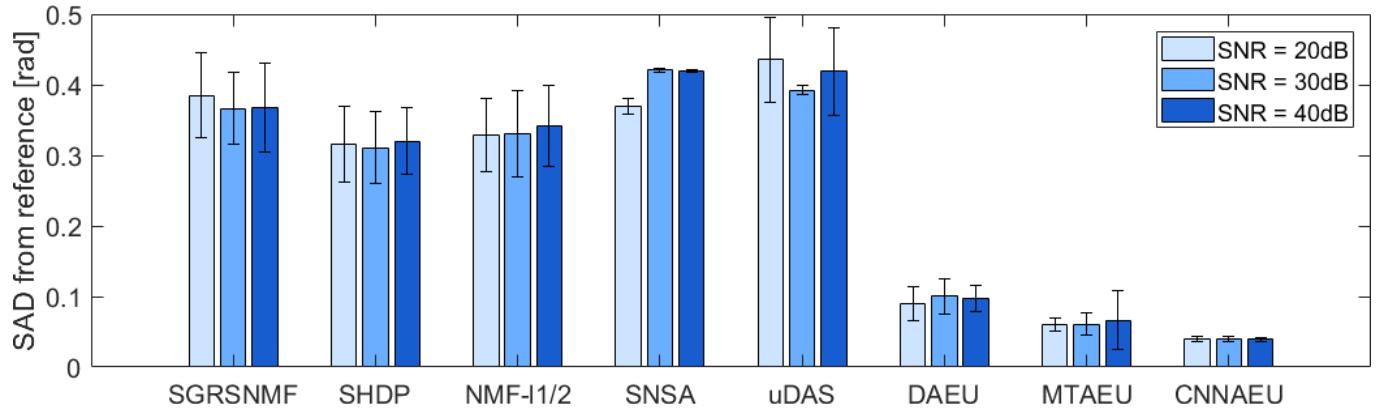


Fig. 14. Urban data set. Mean SAD and standard deviation comparison in radians for robustness test.

G. Robustness to Noise

To investigate the method's robustness to noise, we performed experiments where we added white noise with SNR = 20, 30, and 40 dB to the Urban data set prior to unmixing with all methods. Fig. 14 shows a bar chart of the mSAD score for all the methods versus SNR grouped by methods. The figure shows that all the methods are fairly robust to noise. Also it is surprising how well the methods, except for SGSRNMF and uDAS, performed with 20-dB SNR. The proposed method seems to be largely unaffected by the added noise. The low variance of the method for all noise levels stands out in the comparison.

H. Computational Complexity

Herein, we discuss the running time of the method and compare it to the running time of the other methods. Table XIII shows the average running time in seconds for all the methods for all the data sets and additionally one large data set,

a 900×900 pixel subset of the Apex HSI named “Apex big” in the table. It must be kept in mind that for the methods DAEU, MTAEU, and CNNAEU, the training set is only a small subset of all possible patches or pixels, thus the effect of the data set size is not very apparent for the smaller data sets for these methods. In general, these methods produce good results for running times around 2 min for the smaller data sets. The running time for the other methods is more closely related to the sizes of the data sets. As can be seen from the table, the running time for the methods DAEU, MTAEU, and CNNAEU are much lower than for the comparison methods for the large data set. The running time for the proposed method is around 2 min compared to 17 min for $\ell_{1/2}$ -NMF and around 4 h for the uDAS method. The running time for SHDP scales exponentially with data set size, and therefore it was omitted from the comparison for the large data set. The experiments were performed on a computer having an eight-core CPU and 64 GB of memory and a GPU having 11 GB of memory.

TABLE XIII

CPU TIME IN SECONDS FOR ALL UNMIXING METHODS AND ALL DATA SETS USED IN THE EXPERIMENTS ALONG WITH ONE ADDITIONAL LARGE DATA SET NAMED “APEX BIG” IN THE TABLE

Method	Urban	Samson	Houston	Apex	Apex big
$\ell_{1/2}$ -NMF	83	7	31	152	1186
SGSNMF	276	27	85	312	1007
SHDP	14425	939	8745	16743	NA
SNSA	809	106	708	1813	5856
uDAS	940	5	158	1654	14424
DAEU	226	4	65	165	242
MTAEU	80	58	105	95	223
CNNAEU	143	37	59	111	150

IV. CONCLUSION

In this article, a new spectral–spatial linear mixing model for hyperspectral unmixing (HU) was introduced along with an associated convolutional autoencoder for doing blind unmixing. The new model extends the linear model to apply to the case where neighboring pixels take part in the reconstruction of a pixel. The unmixing technique is the first blind unmixing method to utilize a CNN and make direct use of the spatial correlations in an HSI for both endmember and abundance extraction. This new framework offers many advantages over older methods, such as making it very easy to implement spatial regularizations. We compared the CNNAEU method to four conventional and three deep learning state-of-the-art unmixing methods using four real HSIs. Three of the comparison methods are spectral–spatial methods. Experimental results show that the CNNAEU technique outperforms the comparison methods when it comes to endmember extraction. It achieved the best SAD score in all data sets except for Samson data set, which might be explained by its small size. Overall, the method had very low variance and the best consistency by far. Experiments also demonstrate the method’s robustness to noise. The abundance maps produced by CNNAEU are binary and therefore do not achieve very low MSE from reference. To circumvent this problem, the method was extended to determine the abundance maps directly from the endmembers using a fully connected autoencoder with a linear and fixed decoder. The maps produced by this new method, CNNAEU2, are not as binary looking and achieved the best MSE score for the Apex and the Houston data sets and the second-best score for the other two data sets. Overall, the proposed method significantly outperforms all the other methods and compares very favorably to our previous methods.

ACKNOWLEDGMENT

The authors would like to thank Jun Li for the code in [51] and Ying Qu for making the code in [24] available on GitHub.

REFERENCES

- [1] C. Shi and L. Wang, “Incorporating spatial information in spectral unmixing: A review,” *Remote Sens. Environ.*, vol. 149, pp. 70–87, Jun. 2014.
- [2] J. M. Bioucas-Dias *et al.*, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [3] R. Heylen, M. Parente, and P. Gader, “A review of nonlinear hyperspectral unmixing methods,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, Jun. 2014.
- [4] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Sparse unmixing of hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.
- [5] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Total variation spatial regularization for sparse hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, Nov. 2012.
- [6] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Collaborative sparse regression for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.
- [7] M.-D. Iordache, J. M. Bioucas-Dias, A. Plaza, and B. Somers, “MUSIC-CSR: Hyperspectral unmixing via multiple signal classification and collaborative sparse regression,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4364–4382, Jul. 2014.
- [8] L. Zhang, W. Wei, Y. Zhang, F. Li, and H. Yan, “Structured sparse Bayesian hyperspectral compressive sensing using spectral unmixing,” in *Proc. 6th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2014, pp. 1–4.
- [9] R. T. Albayrak, A. C. Gurbuz, and B. Gunzel, “Compressed sensing based hyperspectral unmixing,” in *Proc. 22nd Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2014, pp. 1438–1441.
- [10] V. Xusu, “Compressive sensing for endmember extraction,” in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 1345–1348.
- [11] 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.
- [12] 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.
- [13] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, “Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4282–4297, Nov. 2011.
- [14] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, “Hyperspectral unmixing with l_q regularization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6793–6806, Nov. 2014.
- [15] J. M. P. Nascimento and J. M. Bioucas-Dias, “Hyperspectral unmixing based on mixtures of Dirichlet components,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, Mar. 2012.
- [16] N. Dobigeon, S. Moussaoui, M. Coulon, J.-Y. Tourneret, and A. O. Hero, “Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4355–4368, Nov. 2009.
- [17] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret, “Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery,” *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1403–1413, Jun. 2010.
- [18] R. Mittelman, N. Dobigeon, and A. O. Hero, “Hyperspectral image unmixing using a multiresolution sticky HDP,” *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1656–1671, Apr. 2012.
- [19] Y. Su, A. Marinoni, J. Li, A. Plaza, and P. Gamba, “Nonnegative sparse autoencoder for robust endmember extraction from remotely sensed hyperspectral images,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 205–208.
- [20] R. Guo, W. Wang, and H. Qi, “Hyperspectral image unmixing using autoencoder cascade,” in *Proc. 7th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2015, pp. 1–4.
- [21] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, “DAEN: Deep autoencoder networks for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, Jul. 2019.
- [22] S. Ozkan, B. Kaya, and G. B. Akar, “EndNet: Sparse AutoEncoder network for endmember extraction and hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 482–496, Jan. 2019.
- [23] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, “Hyperspectral unmixing using a neural network autoencoder,” *IEEE Access*, vol. 6, pp. 25646–25656, 2018.
- [24] 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.
- [25] M. Wang, M. Zhao, J. Chen, and S. Rahardja, “Nonlinear unmixing of hyperspectral data via deep autoencoder networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1467–1471, Sep. 2019.

- [26] B. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, “Spectral–spatial hyperspectral unmixing using multitask learning,” *IEEE Access*, vol. 7, pp. 148861–148872, 2019.
- [27] L. Wang, C. Shi, C. Diao, W. Ji, and D. Yin, “A survey of methods incorporating spatial information in image classification and spectral unmixing,” *Int. J. Remote Sens.*, vol. 37, no. 16, pp. 3870–3910, Aug. 2016.
- [28] M. C. Torres-Madronero and M. Velez-Reyes, “Integrating spatial information in unsupervised unmixing of hyperspectral imagery using multiscale representation,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1985–1993, Jun. 2014.
- [29] S. Mei, M. He, Y. Zhang, Z. Wang, and D. Feng, “Improving spatial–spectral endmember extraction in the presence of anomalous ground objects,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4210–4222, Nov. 2011.
- [30] X. Zhang *et al.*, “Hyperspectral unmixing via low-rank representation with space consistency constraint and spectral library pruning,” *Remote Sens.*, vol. 10, no. 2, p. 339, 2018.
- [31] S. Zhang, J. Li, H. Li, C. Deng, and A. Plaza, “Spectral–spatial weighted sparse regression for hyperspectral image unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3265–3276, Jun. 2018.
- [32] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, “Blind hyperspectral unmixing using total variation and ℓ_q sparse regularization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 11, pp. 6371–6384, Nov. 2016.
- [33] J. Sigurdsson, M. O. Ulfarsson, J. R. Sveinsson, and J. A. Benediktsson, “Smooth spectral unmixing using total variation regularization and a first order roughness penalty,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2013, pp. 2160–2163.
- [34] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, “Total variation and ℓ_q based hyperspectral unmixing for feature extraction and classification,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 437–440.
- [35] A. Huck and M. Guillaume, “Robust hyperspectral data unmixing with spatial and spectral regularized NMF,” in *Proc. 2nd Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, Jun. 2010, pp. 1–4.
- [36] J. Liu, J. Zhang, Y. Gao, C. Zhang, and Z. Li, “Enhancing spectral unmixing by local neighborhood weights,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1545–1552, Oct. 2012.
- [37] X. Zhang, J. Zhang, C. Li, C. Cheng, L. Jiao, and H. Zhou, “Hybrid unmixing based on adaptive region segmentation for hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3861–3875, Jul. 2018.
- [38] M. Zortea and A. Plaza, “Spatial preprocessing for endmember extraction,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2679–2693, Aug. 2009.
- [39] X. Sun, F. Zhang, L. Yang, B. Zhang, and L. Gao, “A hyperspectral image spectral unmixing method integrating slic superpixel segmentation,” in *Proc. 7th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2015, pp. 1–4.
- [40] J. Yi and M. Velez-Reyes, “Dimensionality reduction using superpixel segmentation for hyperspectral unmixing using the cNMF,” *Proc. SPIE Algorithms Technol. Multispectral, Hyperspectral, Ultraspectral Imag.*, vol. 10198, Feb. 2017, Art. no. 101981H.
- [41] X. Xu, J. Li, C. Wu, and A. Plaza, “Regional clustering-based spatial preprocessing for hyperspectral unmixing,” *Remote Sens. Environ.*, vol. 204, pp. 333–346, Jan. 2018.
- [42] 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.
- [43] J. M. Bioucas-Dias and J. M. P. Nascimento, “Hyperspectral subspace identification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.
- [44] B. Rasti, M. O. Ulfarsson, and J. R. Sveinsson, “Hyperspectral subspace identification using SURE,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2481–2485, Dec. 2015.
- [45] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, *arXiv:1502.03167*. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [46] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 648–656.
- [47] F. Zhu, “Spectral unmixing datasets with ground truths,” Oct. 2017, *arXiv:1708.05125*. [Online]. Available: <http://arxiv.org/abs/1708.05125>
- [48] L. J. Rickard, R. W. Basedow, E. F. Zalewski, P. R. Silvergate, and M. Landers, “HYDICE: An airborne system for hyperspectral imaging,” in *Proc. Imag. Spectrometry Terr. Environ.*, Sep. 1993, pp. 173–179.
- [49] M. E. Schaeppman *et al.*, “Advanced radiometry measurements and Earth science applications with the airborne prism experiment (APEX),” *Remote Sens. Environ.*, vol. 158, pp. 207–219, Mar. 2015.
- [50] D. C. Heinz, “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.
- [51] Y. Su, A. Marinoni, J. Li, J. Plaza, and P. Gamba, “Stacked nonnegative sparse autoencoders for robust hyperspectral unmixing,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1427–1431, Sep. 2018.



Burkni Palsson (Member, IEEE) received the B.S. degree in mathematics and the B.S. degree in physics from the University of Iceland, Reykjavik, Iceland, in 1999, and the B.S. degree in energy and environmental technology from the University of Iceland, in 2012, where he is pursuing the M.S. degree in electrical engineering.

His research interests include hyperspectral unmixing and classification in remote sensing and the development and applications of deep learning-based methods in that field.



Magnus O. Ulfarsson (Senior Member, IEEE) received the B.S. and M.S. degrees from the University of Iceland, Reykjavik, Iceland, in 2002, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2007.

In 2007, he joined the University of Iceland, where he is a Professor and the Chair of the Faculty of Electrical and Computer Engineering. Since 2013, he has been with deCODE Genetics, Reykjavik. His research interests include statistical signal processing, image processing, machine learning, remote sensing, medical imaging, and genomics.

Dr. Ulfarsson has been an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING since 2018.



Johannes R. Sveinsson (Senior Member, IEEE) received the B.S. degree from the University of Iceland, Reykjavík, Iceland, and the M.S. and Ph.D. degrees from Queen’s University, Kingston, ON, Canada, all in electrical engineering.

He was with the Laboratory of Information Technology and Signal Processing from 1981 to 1982 and the Engineering Research Institute and the Faculty of Electrical and Computer Engineering, University of Iceland, as a Senior Member of Research Staff and a Lecturer, respectively, from 1991 to 1998.

He was a Visiting Research Student with the Imperial College of Science and Technology, London, U.K., from 1985 to 1986. At Queen’s University, he held teaching and research assistantships. He is a Professor with the Faculty of Electrical and Computer Engineering, University of Iceland, and was the Chair of the Faculty of Electrical and Computer Engineering from 2000 to 2006 and from 2010 to 2014 and a member of the University Council, University of Iceland, from 2002 to 2006. His research interest is in systems and signal theory.

Dr. Sveinsson received the Queen’s Graduate Awards from Queen’s University. He was a recipient of the Best Paper Award Novel Engineering Application at ANNIE 1995 and a co-recipient of the 2013 IEEE GRSS Highest Impact Paper Award. He has been an Associate Editor of *Remote Sensing* since 2018.