IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received September 11, 2019, accepted September 22, 2019, date of publication September 26, 2019, date of current version October 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944072

Spectral-Spatial Hyperspectral Unmixing Using Multitask Learning

BURKNI PALSSON, (Student Member, IEEE),
JOHANNES R. SVEINSSON, (Senior Member, IEEE),
AND MAGNUS O. ULFARSSON, (Senior Member, IEEE)
Faculty of Electrical and Computer Engineering, University of Iceland, 107 Reykjavik, Iceland

Corresponding author: Johannes R. Sveinsson (sveinsso@hi.is)

This work was supported in part by the Icelandic Research Fund under Grant 174075-05.

ABSTRACT Hyperspectral unmixing is an important and challenging task in the field of remote sensing which arises when the spatial resolution of sensors is insufficient for the separation of spectrally distinct materials. Hyperspectral images, like other natural images, have highly correlated pixels and it is very desirable to make use of this spatial information. In this paper, a deep learning based method for blind hyperspectral unmixing is presented. The method uses multitask learning through multiple parallel autoencoders to unmix a neighborhood of pixels simultaneously. Operating on image patches instead of single pixels enables the method to take advantage of spatial information in the hyperspectral image. The method is the first in its class to directly utilize the spatial structure of hyperspectral images (HSIs) for the estimation of the spectral signatures of endmembers in the data cube. We evaluate the proposed method using two real HSIs and compare it to seven state-of-the-art methods that either rely only on spectral or both on spectral and spatial information in the HSIs. The proposed method outperforms all the baseline unmixing methods in experiments.

INDEX TERMS Hyperspectral, unmixing, autoencoder, multitask learning, deep learning.

I. INTRODUCTION

Due to physical limitations of the sensors used in the acquisition of hyperspectral imagery, their spatial resolution is insufficient to separate spectrally distinct materials in the scene, resulting in mixed pixels. The task of hyperspectral unmixing (HSU) [1] is the simultaneous estimation of a fixed number of distinct spectral signatures (endmembers) in a hyperspectral image (HSI) along with their fractional abundances for every pixel under the linear mixing model (LMM), which is the model most spectral unmixing methods assume.

A. MOTIVATION AND CONTRIBUTIONS

As in most natural images, there is a strong spatial correlation between pixels in an HSI. The majority of HSU methods developed so far do not exploit the spatial structure of HSIs but methods that do so are called spectral-spatial methods. What such spectral-spatial methods have in common, is the use of natural assumptions about the spatial correlation of pixels in an HSI as priors to control sparsity and smoothness of the obtained abundance maps.

In this paper, a novel autoencoder based method which attempts to make direct use of spatial information from neighboring pixels in an HSI is proposed. The method directly unmixes a whole neighborhood of pixels at a time using an architecture that is inspired by multitask learning (MTL) [2], [3]. It consists of multiple branches of unmixing autoencoders, each tasked with unmixing a single pixel in a neighborhood, that share features between them. Even though the tasks are identical, MTL is still beneficial. The main benefits of MTL are the following:

- **Faster learning**. Correlated tasks contribute to the aggregate gradient during backpropagation and thus increase the effective learning rate on the input to hidden layer weights [2]. Useful features form faster in the shared hidden layer of the network.
- **Reduced risk of overfitting**. Having shared hidden layers can greatly reduce the risk of overfitting. It has been shown that the risk of overfitting shared parameters can be up to an order N smaller, where N is the number of tasks, than overfitting task-specific parameters [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.

- **Improved stability.** For some HSI datasets with correlated and underrepresented endmembers, a nonnegative sparse autoencoder (NNSAE) often finds two (or more) very different solutions that correspond to different local minima, with one of them preferred. It can be difficult to ensure consistency in such cases, i.e., that the network always chooses the preferred solution. It has been demonstrated that MTL tasks prefer hidden layer representations that other tasks prefer [2]. In our case, this along with the fact that all autoencoders share a decoder, means that the preferred solution becomes even more preferred in an MTL setting, leading to increased stability.
- **Incorporation of spatial information**. Through the sharing of the first hidden layer between autoencoders, each autoencoder or task has access to all features from all the pixels input to the network. By selecting these pixels from a neighborhood in the HSI, we are exploiting the spatial correlation in the HSI, i.e., the assumption that all the pixels from a small neighborhood should have similar abundances. This reinforces the network's learning of shared features between pixels and enhances the effectiveness of the multitask learning.

In more traditional methods, i.e., non-neural network methods, MTL is often implemented by capturing the relationship of multiple related tasks using a low-rank structure through a nuclear norm regularization, and identifying the outlier tasks using a group-sparse structure [5].

To summarize, the main contributions of the proposed method and improvements over our previous method described in [6] are the following:

- The method makes direct use of the spatial information in an HSI by unmixing a whole patch at a time using multiple parallel autoencoders. Hence, the major difference is that the proposed method is a spectral-spatial unmixing method, while the previous method is a spectral unmixing method.
- The method uses the softmax function to enforce the abundances sum-to-one constraint.
- The method makes better use of batch normalization and dropout than our previous method.

The experimental part aims to confirm that the incorporation of spatial information using multiple tasks gives better results than a single task autoencoder. The proposed method is a substantial improvement over our previous autoencoder method [6] as can be seen in the experimental section. It significantly outperforms all the comparison methods, especially regarding consistency and has much lower variance.

B. PUBLICATION REVIEW

Most spectral unmixing methods, blind or not, can be broadly classified into three major categories: geometrical methods, statistical methods, and sparse regression methods [7], [8]. The geometric methods can further be categorized as pure pixel methods or minimum volume methods [7], [9], [10]. Vertex component analysis (VCA) [11] is a well known example of a pure pixel method, while minimum volume simplex analysis (MVSA) [12] is an example of a minimum volume method. An example of $\ell_{1/2}$ sparsity constrained method is [13] and of a ℓ_q sparsity constrained method is [14]. Statistical methods formulate the unmixing problem as a statistical inference problem [15]–[17]. Example of such a method that implements spatially constrained unmixing is [18]. Sparse regression based unmixing methods are based on the assumption that the observed spectra can be expressed as linear combinations of known spectral signatures which can be identified in spectral libraries, makes them semi-supervised [19]–[22]. Methods based on compressed sensing also belong to this category of methods [7], [23]–[25].

The last few years have seen a steadily growing number of deep learning based methods for unsupervised unmixing. Recent applications of neural network methods for unmixing are given in [6], [26]–[30]. So far, most deep learning unmixing methods are based on the NNSAE having linear decoder which effectively perform nonnegative matrix factorization (NMF) according to the LMM, enabling the endmembers to be extracted as the weights of the decoder layer and the abundances as the activities of the units in the hidden layer before it. To the authors' best knowledge, all recent autoencoder based approaches do not directly incorporate any spatial information to assist in the endmember extraction. All the methods take a single pixel spectrum as input.

Spectral-spatial methods are unmixing methods that seek to exploit the spatial relationship in HSIs. Such methods most often introduce weighing factors to penalize non-zero coefficients on the sparse solution or regularization terms based on the abundance fractions to enhance sparsity and sharpness of abundance maps. A good example of such a method and one of the first methods to exploit the rich spatial structure of HSIs was the SUnSAL-TV method [20]. Another similar method is [31]. Other examples are methods using spectral–spatial weighted sparse regression described in [32]. In [33]–[35], a TV regularization term is used as a way of exploiting spatial homogeneity in the image while still retaining sharp edges. Examples of NMF methods using spatial regularization are [36], [37]. An example of a hybrid linear and nonlinear NMF unmixing method based on a spatial prior is given in [38]. Another class of spatial-spectral methods are superpixel segmentation methods. These methods use a superpixel segmentation method as preprocessing technique to hyperspectral images. Following this, the unmixing is applied directly on the mean spectra of the various image segments. Such processing significantly improves the signal quality of the data and reduces the ill effects of noise. Examples of this are [39]–[41] and [42] which applies a spatial group sparsity regularization derived from segmentation. What all these methods have in common, is the use of natural assumptions about the spatial correlation of pixels in an HSI as priors to control sparsity and smoothness of the obtained abundance maps.

The remainder of this paper is organized as follows. Section II describes the proposed method. In Section III,

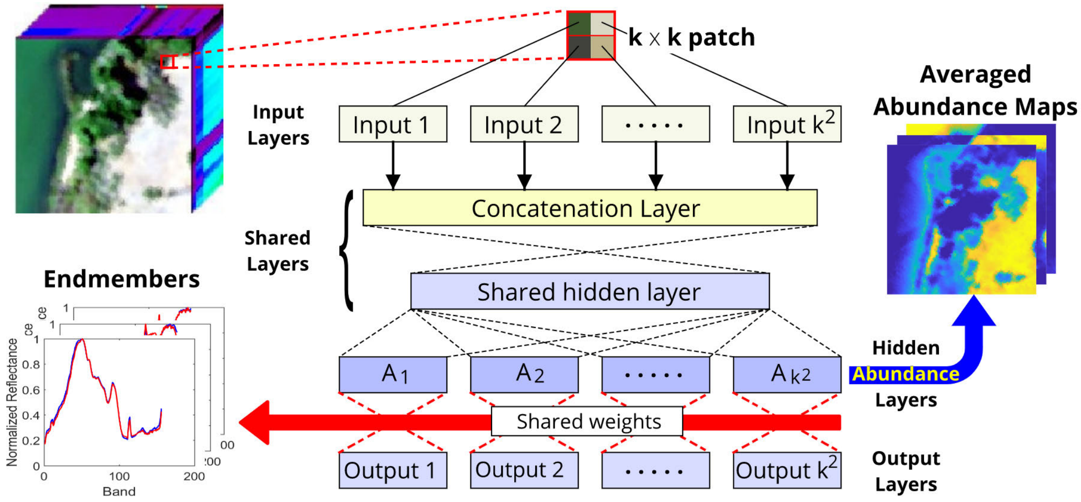


FIGURE 1. **A schematic of the proposed method.**

real HSI datasets are used in experiments for evaluation and comparison of the method to seven state-of-the-art methods. And finally, in Section IV conclusions will be drawn.

C. NOTATION

In the paper, the notation shown below will be used.

Number of pixels in HSI	P
$k \times k$ neighborhood centered on pixel p	$\mathcal{N}_k^p$
Activation function	g
Activations of layer/transform l, branch i	$\boldsymbol{a}_i^{(l)}$
Activation of unit j in $\boldsymbol{a}_i^{(l)}$	$\boldsymbol{a}_{i,j}^{(l)}$
Weights of layer l, branch i	$\boldsymbol{W}_i^{(l)}$

For layers shared between branches, the subscript i will be dropped for activations and weights.

II. PROBLEM FORMULATION

According to the LMM, an HSI $\boldsymbol{X} \in \mathbb{R}^{B \times P}$ having P pixels, each with B bands, can be written as

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{N}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{B \times R}$ contains the R endmembers as columns and $\boldsymbol{S} \in \mathbb{R}^{R \times P}$ contains the P abundances fractions as columns and $\mathbf{N}$ is noise. The LMM requires that the entries in both $\boldsymbol{A}$ and $\boldsymbol{S}$ are nonnegative and also that the columns in $\boldsymbol{S}$ sum to one. The problem investigated in this paper is the estimation of the endmember matrix $\boldsymbol{A}$ and the abundance matrix $\boldsymbol{S}$ in (1), by interpreting the problem as blind unmixing, and solving it in an unsupervised manner using an autoencoder.

An autoencoder is a feedfoward neural network that is trained to reproduce its input by learning the identity function. The autoencoder can be considered to consist of two parts: An encoder, $\mathcal{G}_E : \mathbb{R}^{B \times 1} \to \mathbb{R}^{R \times 1}$, which encodes the input spectrum $\boldsymbol{x}_p$ to a hidden representation, $\mathcal{G}(\boldsymbol{x}_p) = \boldsymbol{h}_p \in \mathbb{R}^{R \times 1}$, and a decoder, $\mathcal{G}_D : \mathbb{R}^{R \times 1} \to \mathbb{R}^{B \times 1}$, which decodes the hidden representation $\boldsymbol{h}_p$ to an approximation of the input, $\mathcal{G}_D(\boldsymbol{h}_p) = \hat{\boldsymbol{x}}_p$. The network is thus trained using backpropagation to minimize the loss function,

$$\mathcal{L}(\boldsymbol{x}_p, \mathcal{G}_D(\mathcal{G}_E(\boldsymbol{x}_p))), \tag{2}$$

where $\mathcal{L}(\cdot, \cdot)$ is some measure of the discrepancy between the input and the output. By having a hidden layer with only a few units as the final layer of the encoder, it is mapping spectral vectors of length B into a R dimensional latent space. The decoder then uses the latent vectors to reconstruct the input spectrum as faithfully as possible.

If we require the decoder to have a linear activation function and its weights to be nonnegative, we can interpret the weight matrix as the endmember matrix in accordance with the LMM. At the end of training, the weight matrix of the decoder is the endmember matrix and the abundances can be obtained by prediction on a network consisting only of the encoder part.

A. THE METHOD

Fig. 1 shows a schematic of the proposed method. There are k^2 tasks in the form of autoencoders, one for each pixel in a $k \times k$ neighborhood $\mathcal{N}_k$, input to the method. The inputs are concatenated and connected into a large hidden layer shared by all the autoencoders. This shared hidden layer enables all the unmixing tasks to access features from all the neighboring pixels. Such sharing of representations between tasks can improve both learning and generalization [2], and is called hard parameter sharing in the language of MTL. The architecture branches again after the shared layer. Each autoencoder performing further dimensional reduction through an additional hidden layer, called A_i in Fig. 1. The activations for each autoencoder become the abundance fractions after applying the abundance sum-to-one constraint (ASC). However, since endmembers are a property of an HSI as a whole, all autoencoders are required to use the same decoder having weights $\boldsymbol{W}_D$, placing a constraint on each of them to

TABLE 1. The layers/transformations in each encoder and their expressions listed sequentially.

	Layer/Transform	Expression
1	Input i	$\boldsymbol{a}_i^{(0)} = \boldsymbol{x}_i^{\mathcal{N}_k^{(p)}}$
2	Concatenation	$\boldsymbol{a}^{(1)} = [\boldsymbol{a}_1^{(0)}, \cdots, \boldsymbol{a}_i^{(1)}, \cdots, \boldsymbol{a}_{k^2}^{(1)}] = \boldsymbol{C}\boldsymbol{a}_i^{(0)}$
3	Hidden layer shared	$\boldsymbol{a}^{(2)} = g(\boldsymbol{W}^{(2)}\boldsymbol{a}^{(1)})$
4	Batch normalization	$\boldsymbol{a}_i^{(3)} = \mathrm{BN}\,\boldsymbol{a}_i^{(2)}$
5	Hidden layer not shared	$\boldsymbol{a}_i^{(4)} = g(\boldsymbol{W}_i^{(4)}\boldsymbol{a}^{(3)})$
6	Batch Normalization	$\boldsymbol{a}_i^{(5)} = \mathrm{BN}\,\boldsymbol{a}_i^{(4)}$
7	ASC constraint	$a_{i,j}^{(7)} = \frac{e^{\alpha a_{i,j}^{(6)}}}{\sum_{j=1}^{R} e^{\alpha a_{i,j}^{(6)}}} = \boldsymbol{\sigma_\alpha}\boldsymbol{a}_i^{(6)}$

discover the same endmembers. This is what connects all the tasks. If one tasks modifies the endmembers, the endmembers are modified for all the tasks.

B. LOSS FUNCTION

Each NNSAE consists of an encoder part, $\mathcal{G}_E$, and a decoder part, $\mathcal{G}_D$. The encoder encodes the input $\boldsymbol{x}_p$ into a hidden representation $\boldsymbol{h}_p = \mathcal{G}_E(\boldsymbol{x}_p)$. The encoder of autoencoder i consists of the layers or transformations listed sequentially in Table 1. We have used operator notation to denote the operations given in rows 2, 4, 6, and 7. From the table we can construct an expression for the hidden representation $\boldsymbol{h}_p$, obtained by a feed-forward pass, as

$$\mathcal{G}_E(\boldsymbol{x}_i^{\mathcal{N}_k}) = \boldsymbol{\sigma_\alpha}\,\mathrm{BN}\,g(\boldsymbol{W}_i^{(3)}\mathrm{BN}\,g(\boldsymbol{W}^{(2)}\boldsymbol{C}\boldsymbol{a}_i^{(0)})). \tag{3}$$

Here g is the activation function which is the LeakyReLU [43] function, and $\boldsymbol{\sigma_\alpha}$ is the softmax function with scaling parameter α which enforces the ASC constraint according to row 7 in Table 1. We apply dropout [44] after the shared hidden layer using a dropout rate of 0.5. We also apply batch normalization [45], denoted BN, after both hidden layers. The decoder then reconstructs the input as $\hat{\boldsymbol{x}}_p$ from the hidden representation as

$$\hat{\boldsymbol{x}}_p = \mathcal{G}_D(\boldsymbol{h}_p). \tag{4}$$

The final layer in the network is the decoder having weights $\boldsymbol{W}_D$ which are constrained to be nonnegative, and which has linear activation as required by the LMM. The action of the decoder is a simple linear transformation which can be written as

$$\mathcal{G}_D(\boldsymbol{h}_i^{N_k^{(p)}}) = \boldsymbol{W}_D\boldsymbol{h}_i^{N_k^{(p)}}, \tag{5}$$

where $\boldsymbol{h}_i^{N_k^{(p)}}$ is the hidden representation belonging to pixel i in the neighborhood $\mathcal{N}_k^{(p)}$ which is the input to the method. The loss function we use for each autoencoder i is the spectral angle distance (SAD) and is given by

$$\mathcal{L}^{(i)} = \cos^{-1}\left(\frac{\langle \boldsymbol{x}_i, \hat{\boldsymbol{x}}_i \rangle}{\|\boldsymbol{x}_i\|_2 \, \|\hat{\boldsymbol{x}}_i\|_2}\right).$$

The total loss of the neural network is the sum of each individual autoencoder loss. The method divides the HSI into a collection of neighborhoods $\mathcal{N}_k$ and unmixes one neighborhood at a time. The neighborhood loss is the sum of the individual pixel losses for each pixel in the neighborhood

$$\mathcal{L}^{\mathcal{N}_k(\boldsymbol{x}_p)} = \sum_{i=1}^{k^2} \mathcal{L}^{(i)}, \tag{6}$$

and the total loss is the sum of the neighborhood losses for all $k \times k$ neighborhoods in the training dataset consisting of N randomly chosen patches from the HSI

$$\mathcal{L} = \sum_{p=1}^{N} \mathcal{L}^{\mathcal{N}_k(\boldsymbol{x}_p)}. \tag{7}$$

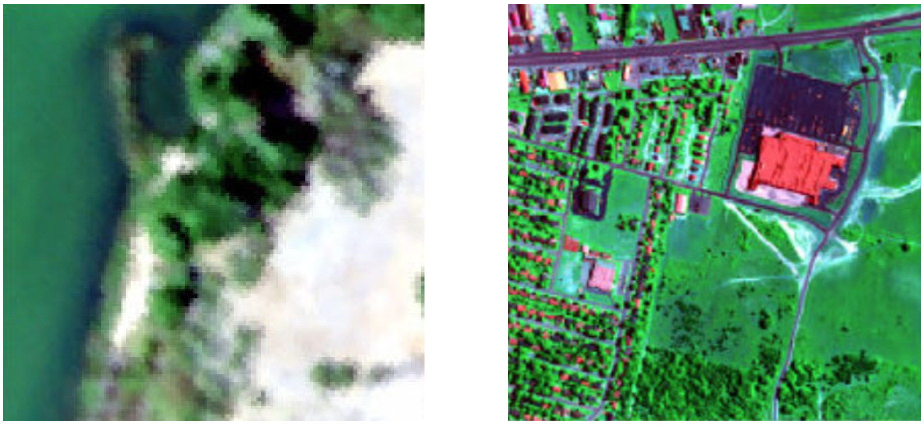
At the end of training, the decoder weights matrix, $\boldsymbol{W}_D$, contains the endmembers and the abundances of each pixel $\boldsymbol{x}_p$ are the activations $\boldsymbol{h}_p$ in the last hidden layer before the decoder. These can be extracted by doing prediction for the whole HSI using a network consisting only of the already trained encoder.

III. EXPERIMENTS

Experiments were conducted on two real datasets having different characteristics. The datasets were selected because they have been widely used in hyperspectral unmixing research and accepted references for both endmembers and abundance maps is available. The methodology for determining the references for these datasets is described in [46]. The datasets are the following:

1) **Samson**. This dataset is a 95 by 95 pixel sub-image of a large hyperspectral dataset acquired by the SAMSON instrument, a push-broom, visible to near IR, hyperspectral sensor. There are 156 bands covering the wavelength range 401-889 nm, all of which are usable. It has the following three endmembers: Soil (#1), Tree (#2), and Water (#3).
2) **Urban**. This dataset is 307 by 307 pixels and was acquired by the HYDICE (Hyperspectral Digital Image Collection Experiment) sensor. There are originally 210 bands covering a wavelength range of 400-2500 nm, of which 162 remain after removing noisy and corrupted bands. Ground truth is available for 4, 5 and 6 endmembers. We use the references for four endmembers which are the following: Asphalt (#1), Grass (#2), Tree (#3), and Roof (#4), and six endmembers which additionally has (#5) Metal, and (#3) Dirt.

Fig. 2 shows simulated RGB images of the datasets. We quantitatively evaluate the endmembers extracted for each dataset



(a) Samson

(b) Urban

FIGURE 2. Simulated RGB images of the HSIs used in experiments.

by calculating the average SAD from the reference endmembers using

$$\mathrm{mSAD} = \frac{1}{R}\sum_{i=1}^{R} \cos^{-1}\left(\frac{\langle \hat{\boldsymbol{m}}_i, \boldsymbol{m}_i\rangle}{\|\hat{\boldsymbol{m}}_i\|_2 \, \|\boldsymbol{m}_i\|_2}\right), \tag{8}$$

where $\hat{\boldsymbol{m}}_i$ are the endmembers extracted by the method, and $\boldsymbol{m}_i$ are the reference endmembers. The lower the SAD, the better the estimated endmembers resemble the reference endmembers. We also quantitatively evaluate the extracted abundance maps for each dataset by calculating the average mean squared error between them and the reference abundance maps using

$$\mathrm{MSE} = \frac{1}{P}\sum_{i=1}^{P} \|\mathbf{S}_i - \hat{\mathbf{S}}_i\|^2, \tag{9}$$

where $\hat{\mathbf{S}}_i$ are the abundance values for a particular endmember for pixel i and $\mathbf{S}_i$ are the reference abundance values.

We evaluate the proposed multitask autoencoder unxmixing (MTAEU) method using $k \times k$ neighborhoods for $k = 2, 3, 4, 5$. Also included in the experiments are the results of a single task autoencoder (STAEU) having the same architecture as the MTAEU network. We compare our results for extracted endmembers and abundance maps to seven other methods, six of which perform blind unmixing, one method that only extracts endmembers. Three of the comparison methods are based on deep learning. The comparison methods are vertex component analysis (VCA) [11], sparsity constrained nonnegative matrix factorization ($\ell_{1/2}$-NMF) [13], sticky hierarchical Dirichlet process (SHDP) [18] which is a spatial-spectral blind unmixing method, spatial group sparsity regularized nonnegative matrix factorization (SGSRNMF) which is a spatial-spectral blind unmixing method [42], deep autoencoder unmixing (DAEU), an autoencoder based method described in [6], stacked nonnegative sparse autoencoder (SNSA) unmixing method described in [47], and an untied denoising autoencoder with sparsity (uDAS) unmixing method described in [48].

All autoencoder methods except for SNSA and uDAS are initialized randomly, while the rest of the methods are all initialized or partially initialized using VCA. The activation function used for the autoencoder methods is the LeakyReLU function and the number of randomly selected patches used in the training for the proposed method was $N = 2000$ for the Urban dataset and $N = 300$ for the Samson dataset. The optimizer used was the RMSprop [49] optimizer having learning rate 0.02 and learning rate decay of 0.02. The networks were trained for 100 epochs.

Fig. 3 shows the average SAD from 25 runs in radians from the reference for both datasets as a function of k for $k = 1, \ldots, 5$. The error bars are the standard deviation. It can be seen from the figure that MTL is beneficial in both cases, especially for the Urban dataset and also that $k = 3$ achieves the lowest SAD score for both datasets and also the least variance.

Fig. 4 shows the result of an experiment designed to evaluate the effect of selecting the pixels in a patch in their original spatial arrangement versus selecting the input pixels randomly. Such an experiment can confirm whether the method is using the spatial information in the scene. The experiment was performed using the Urban dataset for four different sizes of $k \times k$ neighborhoods. Fig. 4 shows a boxplot of the results of 25 runs for each value of k. The figure shows clearly that keeping the spatial arrangement of the pixels gives much better average SAD from the reference and less variance than selecting the pixels randomly. The results are in line with the benefits of MTL discussed earlier. It must be noted that if the number of neighborhoods used in training is increased, the difference becomes somewhat less, which shows also that spatial arrangement speeds up the convergence of the method greatly. Another trend which the figure depicts well, is the more spatial the method becomes, i.e., as more neighboring pixels are used, the greater the difference between spatial

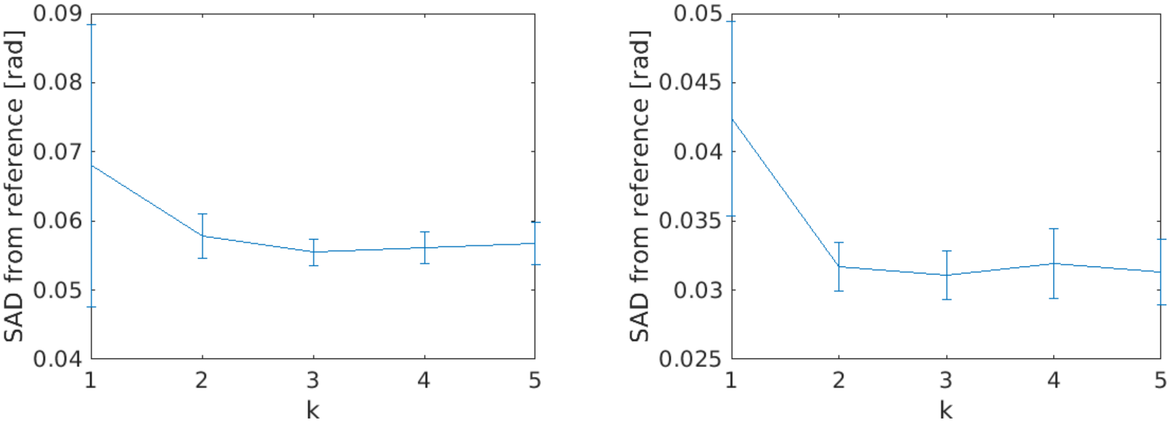


FIGURE 3. Mean SAD from reference for both datasets versus the width of patch, *k*. The plot for the Urban (4 endmembers) dataset is on the left and the plot for the Samson dataset is on the right.

TABLE 2. SAD in radians from reference for the Urban dataset with 4 reference endmember. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

Endm. #	VCA	SHDP	SGSRNMF	NMF-$\ell_{1/2}$	SNSA	uDAS	DAEU	STAEU 1 pixel	MTAEU 9 pixels
1 (Asphalt)	0.2191(0.031)	0.3288(0.14)	0.4819(0.35)	0.1515(0.10)	0.2912(0.0004)	0.1805(0.0327)	0.0720(0.024)	0.0975(0.028)	**0.0843(0.0047)**
2 (Grass)	0.4011(0.042)	0.4184(0.27)	0.6345(0.28)	0.5368(0.39)	1.2165(0.0042)	1.1609(0.1546)	0.0769(0.029)	0.0459(0.015)	**0.0421(0.0037)**
3 (Tree)	0.2019(0.086)	0.088(0.041)	0.0955(0.0070)	0.1571(0.094)	0.0777(0.0001)	0.1496(0.0393)	0.0705(0.030)	**0.0367(0.0080)**	0.0539(0.004)
4 (Soil)	0.824(0.0018)	0.4077(0.16)	0.2071(0.11)	0.4891(0.21)	0.0923(0.0005)	0.1700(0.0012)	0.2155(0.21)	0.0926(0.054)	**0.0415(0.0046)**
Average	0.4115(0.019)	0.3107(0.039)	0.3547(0.058)	0.2955(0.059)	0.4194(0.0011)	0.4153(0.0412)	0.1088(0.024)	0.0680(0.020)	**0.0555(0.0020)**

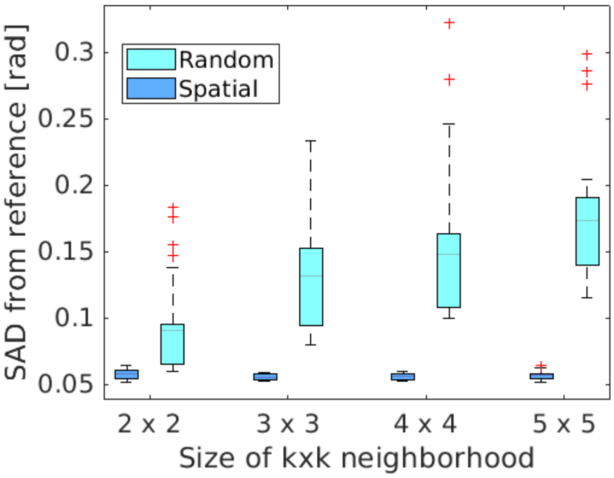


FIGURE 4. Boxplot of SAD from ground truth for the Urban dataset (4 endmembers) showing the effect of selecting pixels into patches randomly vs. spatially for three different patch sizes. Number of neighborhoods used in training is 2000 and number of epochs is 100.

and random arrangement becomes for a fixed number of neighborhoods used in training.

Table 2 shows the SAD in radians of extracted endmembers from the reference, both for individual endmembers and the average, for the proposed method and the comparison methods for the Urban dataset using four reference endmembers. Fig. 5 shows the endmembers extracted by all the methods except VCA. The blue curves are the extracted endmembers (from 25 runs) and the red curves are the reference endmembers. When calculating SAD scores and displaying endmembers in plots, the endmember solutions are matched to the most similar reference endmembers as measured by the SAD metric. This is needed because the order of endmembers returned by the autoencoder methods is essentially random between runs.

It can be seen from both Table 2 and Fig. 5 that the autoencoder methods are in a class of their own compared to the other methods used in the experiments. The MTAEU method achieves the best score for all endmembers except the Tree endmember and also the lowest average SAD score, whereas STAEU achieves the lowest SAD score for the Tree endmember. This should not be over-interpreted, the scores which are averages, are similar and also the reference should not be considered some final truth. What is more important, is the fact that the proposed method has a lower variance. A consistent method having low variance is preferable over method which might achieve a slightly better SAD score w.r.t. some reference endmembers but has high variance. Also, the individual endmember scores are not entirely independent but are linked through the ASC constraint. A bad solution for one endmember can affect the solutions for the other endmembers if its abundance is significant. In the end, the MTAEU method does achieve better MSE score for the Tree abundance map than STAEU.

Fig. 5 shows very clearly that the MTAEU method has both a far less variance than the STAEU method and has far better consistency, i.e., no outlier solutions. In contrast, the comparison methods, aside from STAEU and SNSA, have bad consistency and large variance, especially the SHDP and NMF-$\ell_{1/2}$ methods. The low variance and the excellent

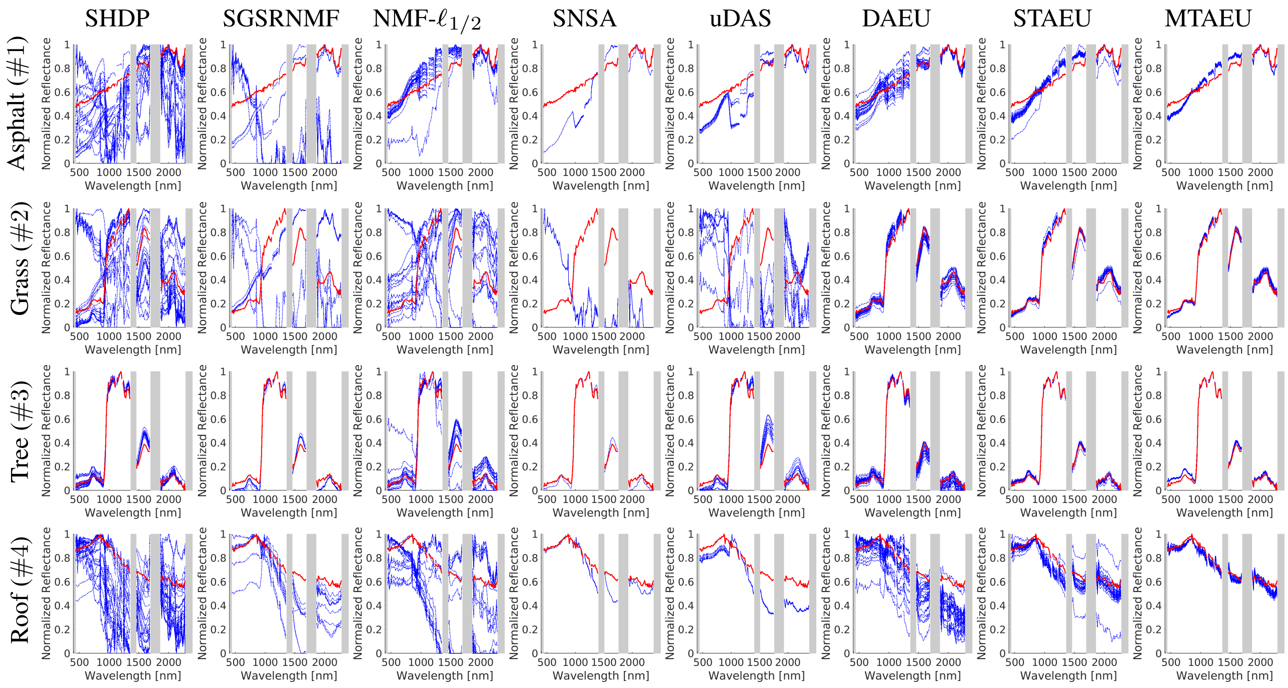


FIGURE 5. The endmember spectra extracted by the proposed method using 3 × 3 pixel neighborhood and the comparison methods for the Urban data set with 4 endmembers. The red curves are the reference endmembers and the blue curves are extracted endmembers.

TABLE 3. SAD in radians from reference for the Urban dataset with 6 reference endmembers. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

Endm. #	VCA	SHDP	SGSRNMF	NMF-$\ell_{1/2}$	SNSA	uDAS	DAEU	STAEU 1 pixel	MTAEU 9 pixels
1 (Asphalt)	0.3204(0.29)	0.4424(0.20)	0.28(0.069)	0.4369(0.21)	0.7172(0.2457)	0.2850(0.1417)	0.1928(0.11)	0.0500(0.012)	**0.0387(0.0083)**
2 (Grass)	0.5502(0.42)	0.2850(0.12)	0.5500(0.17)	0.7347(0.58)	0.6437(0.0906)	0.8915(0.4456)	0.1911(0.039)	**0.0845(0.022)**	0.0896(0.025)
3 (Tree)	0.3035(0.067)	0.1461(0.077)	0.0956(0.014)	0.4560(0.56)	0.1693(0.0102)	0.1438(0.0526)	0.1302(0.047)	0.1026(0.029)	**0.0828(0.045)**
4 (Roof)	0.2810(0.11)	0.3040(0.17)	0.0579(0.031)	0.0858(0.049)	0.3605(0.0865)	0.2122(0.0892)	0.5064(0.062)	**0.0622(0.028)**	0.0753(0.021)
5 (Metal)	0.6883(0.17)	0.6424(0.22)	0.5410(0.12)	0.5886(0.23)	0.8141(0.1955)	0.4257(0.2269)	**0.3367(0.29)**	0.5619(0.21)	0.7158(0.18)
6 (Dirt)	1.0615(0.32)	0.3908(0.18)	0.1759(0.20)	0.1337(0.14)	0.4157(0.1250)	0.5465(0.4119)	0.3983(0.27)	**0.0974(0.052)**	0.1085(0.038)
Average	0.5342(0.067)	0.3684(0.036)	0.2836(0.052)	0.4059(0.030)	0.5201(0.0236)	0.4174(0.0495)	0.2926(0.035)	**0.1598(0.035)**	0.1851(0.023)

consistency of the MTAEU method is directly in accordance with the benefits of MTL listed above in the introduction.

The extracted endmembers by MTAEU need to simultaneously give optimal reconstruction for k^2 autoencoders in parallel. As discussed earlier, multiple tasks increase stability. The endmembers are the weights of the last layer in the network which acts as a decoder for all the different encoder branches and are influenced by them all. Even if one branch seeks a solution in a different local minima, it will not have too much effect on the extracted endmembers of the whole network. The consistency of the method can thus be expected to increase with increasing k. This increased consistency is a clear benefit over ordinary single task deep learning based methods.

Table 3 shows the SAD in radians of extracted endmembers from the reference, both for individual endmembers and the average, for the proposed method and the comparison methods for the Urban dataset using six reference endmembers. Fig. 7 shows the endmembers extracted by all the methods except VCA. For six endmembers the table shows that the STAEU method achieves the lowest average SAD score for the Urban dataset. Fig. 5 shows the reason for this as it can be seen that both STAEU and MTAEU have trouble estimating the "Metal" endmember. In the Urban dataset, the Metal endmember is underrepresented and we have seen that in such situations the autoencoders can achieve lower global reconstruction error or loss by estimating the underrepresented endmember as a variant of a heavily represented endmember. A better estimate could be obtained by the autoencoder methods by increasing the number of patches used in training and/or applying regularization on abundances, but there is a trade-off between this number and the general quality of estimated endmembers.

As can be seen in Fig. 7, both the STAEU and MTAEU methods tend to estimate it as a variant of either the Tree or the Grass endmember. The SAD between the reference Grass endmember and the reference Metal endmember is lower than it is between the Tree endmember and Metal. For STAEU, the Metal endmember is more often similar to the reference Grass endmember than it is to the reference Tree endmember,

TABLE 4. SAD in radians from reference for the Samson dataset. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

Endm. #	VCA	NMF-$\ell_{1/2}$	SHDP	SGSRNMF	SNSA	uDAS	DAEU	STAEU	MTAEU
								1 pixel	9 pixels
1 (Soil)	0.1174(0.24)	0.0471(0.18)	0.02147(0.34)	**0.0086(0.0001)**	0.2493(0.3187)	0.0312(0.0015)	0.027(0.046)	0.020(0.0045)	0.0225(0.0061)
2 (Tree)	0.0487(0.0025)	0.0359(0.0060)	0.0370(0.0004)	0.0395(0.0020)	0.0750(0.0149)	0.0547(0.0050)	**0.033(0.0033)**	0.056(0.0099)	0.0371(0.0028)
3 (Water)	0.1297(0.0010)	0.0425(0.0017)	0.2064(0.094)	0.0923(0.0025)	0.2844(0.0691)	0.1405(0.0119)	0.039(0.0041)	0.036(0.0071)	**0.0338(0.0031)**
Average	0.0986(0.073)	0.0418(0.062)	0.1527(0.14)	0.0468(0.0003)	0.2029(0.1243)	0.0755(0.0052)	0.033(0.0162)	0.037(0.0040)	**0.0311(0.0018)**

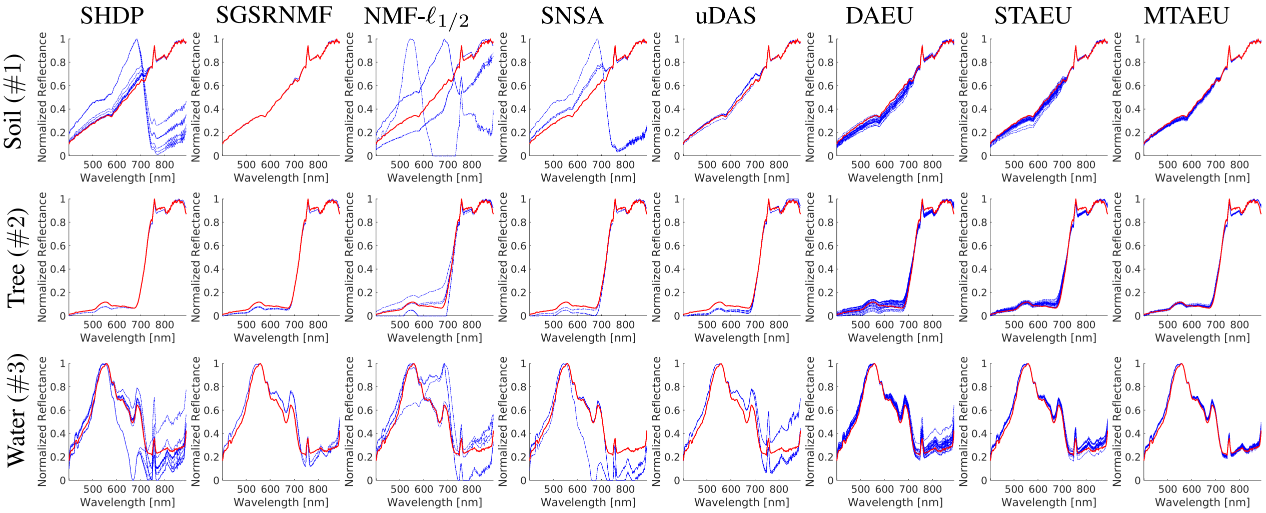


FIGURE 6. The endmember spectra extracted by the proposed method using 3 × 3 pixel neighborhood and the comparison methods for the Samson data set. The red curves are the reference endmembers and the blue curves are extracted endmembers.

TABLE 5. MSE between extracted abundance maps and the reference abundance maps for all methods for the Urban dataset with four reference endmembers. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

	SHDP	SGSRNMF	NMF-$\ell_{1/2}$	SNSA	uDAS	DAEU	STAEU	MTAEU
Endm. #							1 pixel	9 pixels
1 (Asphalt)	0.0949(0.020)	0.1397(0.025)	0.1719(0.0029)	0.1129(0.0005)	0.1013(0.0036)	0.0307(0.018)	0.0271(0.008)	**0.0230(0.0010)**
2 (Grass)	0.1348(0.071)	0.1301(0.043)	0.2047(0.0041)	0.1681(0.0005)	0.1872(0.0389)	0.0556(0.026)	0.0280(0.0038)	**0.0225(0.0016)**
3 (Tree)	0.0833(0.024)	0.0715(0.018)	0.1886(0.0033)	0.1260(0.0007)	0.0916(0.0106)	0.0391(0.034)	0.0099(0.0027)	**0.0068(0.0009)**
4 (Soil)	0.0700(0.055)	0.0358(0.018)	0.0572(0.0006)	0.0433(0.0002)	0.0385(0.0012)	0.0189(0.013)	0.0120(0.011)	**0.0079(0.0008)**
Average	0.0957(0.020)	0.0942(0.010)	0.1556(0.0009)	0.1126(0.0002)	0.1046(0.0072)	0.0360(0.015)	0.0193(0.0062)	**0.0150(0.0008)**

TABLE 6. MSE between extracted abundance maps and the reference abundance maps for all methods for the Urban dataset with six reference endmembers. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

	SHDP	SGSRNMF	NMF-$\ell_{1/2}$	SNSA	uDAS	DAEU	STAEU	MTAEU
Endm. #							1 pixel	9 pixels
1 (Asphalt)	0.0802(0.025)	0.0875(0.010)	0.0986(0.018)	0.0960(0.0072)	0.1138(0.0257)	0.0429(0.021)	0.0230(0.0069)	**0.0157(0.0049)**
2 (Grass)	0.1376(0.061)	0.2136(0.049)	0.2585(0.060)	0.1593(0.0385)	0.1919(0.0634)	0.0510(0.047)	0.0658(0.036)	**0.0475(0.031)**
3 (Tree)	0.0906(0.034)	0.0729(0.013)	0.1822(0.045)	0.1260(0.0044)	0.0890(0.0186)	**0.0411(0.031)**	0.0548(0.015)	0.0643(0.021)
4 (Roof)	0.035(0.013)	0.0287(0.0015)	0.0331(0.011)	0.0387(0.0011)	0.0360(0.0054)	0.0192(0.012)	**0.0057(0.0011)**	0.0058(0.00081)
5 (Metal)	0.0684(0.051)	0.0249(0.046)	**0.0246(0.007)**	0.0679(0.0144)	0.0258(0.0291)	0.0593(0.039)	0.0754(0.022)	0.0717(0.019)
6 (Dirt)	0.0499(0.038)	0.0256(0.035)	0.0394(0.019)	0.0520(0.0095)	0.1004(0.0877)	0.0798(0.034)	0.0207(0.013)	**0.0104(0.004)**
Average	0.0769(0.015)	0.0755(0.0098)	0.1061(0.014)	0.0900(0.0085)	0.0928(0.0183)	0.0489(0.014)	0.0409(0.0085)	**0.0359(0.0076)**

giving a better average score for the Metal endmember. MTAEU, however, is more consistent in estimating the Metal endmember similar to the Tree one.

Table 4 shows the SAD from the reference for the Samson dataset. As before, the values in the table are the average of 25 experiments and the results shown for the proposed method are for the optimal neighborhood size, which was for 3 × 3 neighborhoods. The best results are in bold typeface. The MTAEU method achieves the best average SAD score on this dataset and the best SAD score for the Water endmember. More significantly, MTAEU has substantially lower variance than the single pixel method, STAEU. The DAEU method achieves the second best average score followed by the SGSRNMF method which has the least variance of all the methods. Fig. 6 shows the extracted endmembers and again the same pattern can be observed with MTAEU having considerably less variance and more consistency than STAEU.

Tables 5, 7, and 6 show the MSE between the extracted abundance maps, both for individual endmembers and overall, for all blind unmixing methods for all the datasets. Again the numerical values are the mean value of 25 runs along with their standard deviation and the best results are in

TABLE 7. MSE between extracted abundance maps and the reference abundance maps for all methods for the Samson dataset. The figures are the mean of 25 runs along with the standard deviation. Best results are in bold typeface.

Endm.	SHDP	SGRSNMF	SNSA	uDAS	NMF-$\ell_{1/2}$	STAEU 1 pixel	MTAEU 9 pixels
1 (Soil)	0.0814(0.029)	0.0316(0.0004)	0.1251(0.0205)	0.0651(0.0081)	0.14(0.063)	0.0204(0.0079)	**0.0076(0.0014)**
2 (Tree)	0.0623(0.015)	0.0574(0.0032)	0.1347(0.0086)	0.0666(0.0054)	0.1045(0.020)	0.0084(0.0051)	**0.0037(0.0007)**
3 (Water)	0.1559(0.0091)	0.1227(0.0035)	0.1691(0.0087)	0.1674(0.0069)	0.0658(0.043)	0.0097(0.002)	**0.0029(0.001)**
Average	0.1000(0.012)	0.0706(0.0022)	0.1430(0.0067)	0.0997(0.0038)	0.1034(0.033)	0.0128(0.0044)	**0.0048(0.0008)**

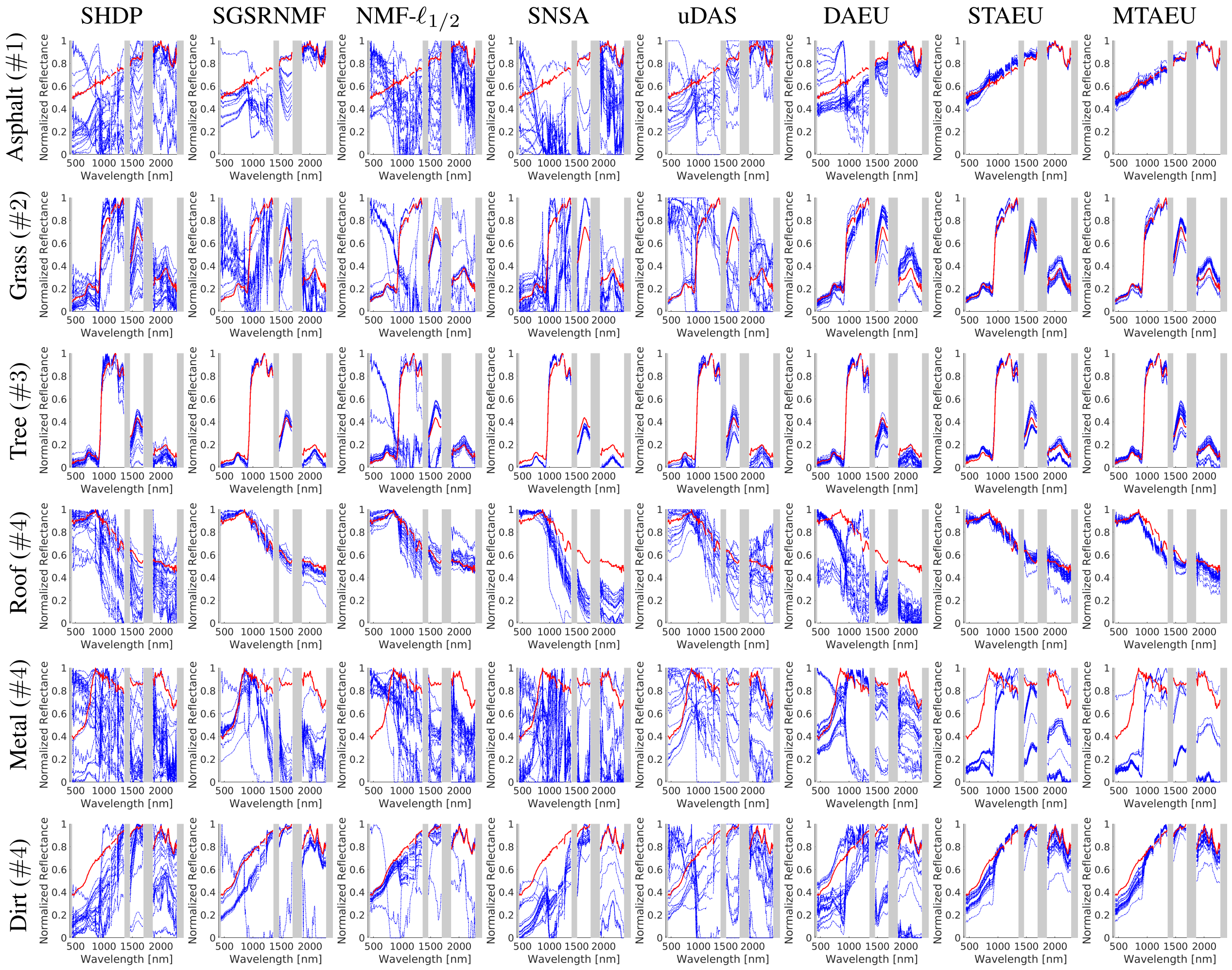


FIGURE 7. The endmember spectra extracted by the proposed method using 3 × 3 pixel neighborhood and the comparison methods for the Urban data set with 6 endmembers. The red curves are the reference endmembers and the blue curves are extracted endmembers.

bold typeface. In all cases, the MTAEU method achieves the lowest average MSE score. It also has the least variance of all the methods. This is not surprising, both given the benefits of MTL discussed earlier, and the fact that the abundance maps for MTAEU are the mean of all the abundance maps of the k^2 autoencoders in the network. There will generally be some variance in the k^2 decoders which produce the abundances even though the decoders are all identical.

Fig. 8 shows an example of the abundance maps extracted by the method for the Urban dataset (4 endmembers) along with the reference maps. It can be seen that the abundance maps produced by the method are somewhat more intense and the fraction of low pixel abundances is lower in the maps, i.e., the method is more confident. This can be seen, e.g., in the figure for the Asphalt endmember. The roads are more intense in color than in the reference map shown. This is a result of using the nonlinear softmax function to enforce the ASC constraint. As the scaling factor α is increased, relatively small values get pushed to smaller values, increasing the "confidence" of abundances as they can be interpreted as probabilities being a result of the softmax function.

Fig. 9 shows the SAD from reference endmembers for all methods for three different SNR values, 20 dB, 30 dB,

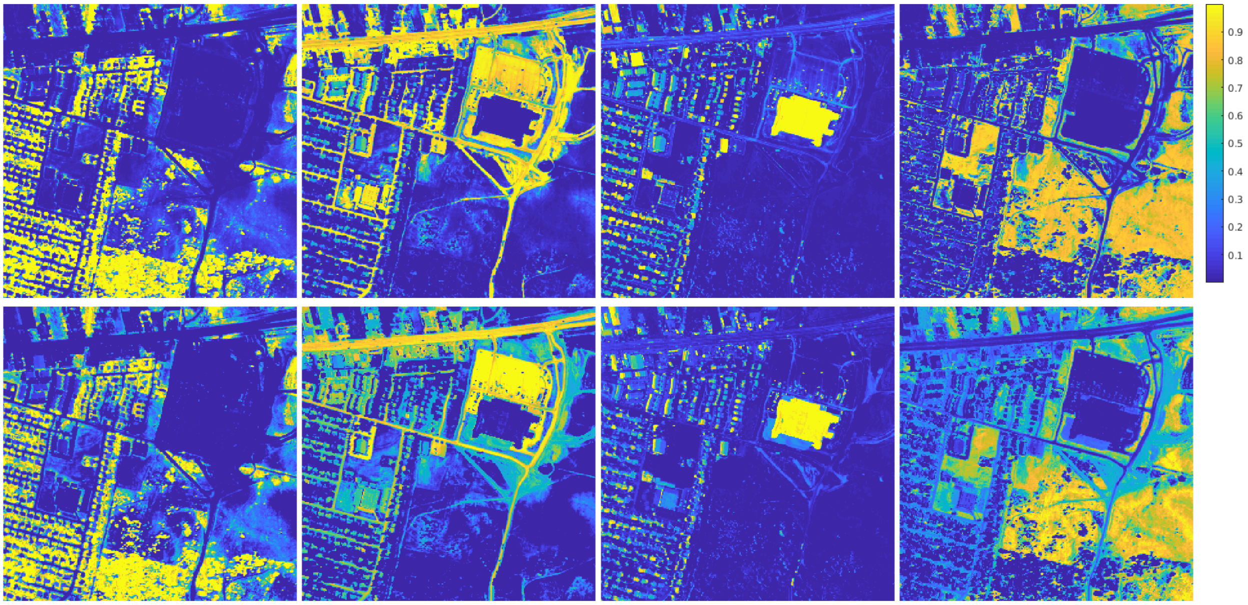


FIGURE 8. The abundance maps produced by MTAEU for the Urban dataset (top row) and the reference abundance maps (bottom row). The first column is the abundance for the "Tree" endmember, the second column the "Asphalt" endmember, the third column is the "Roof" endmember, and the last column the "Grass" endmember.

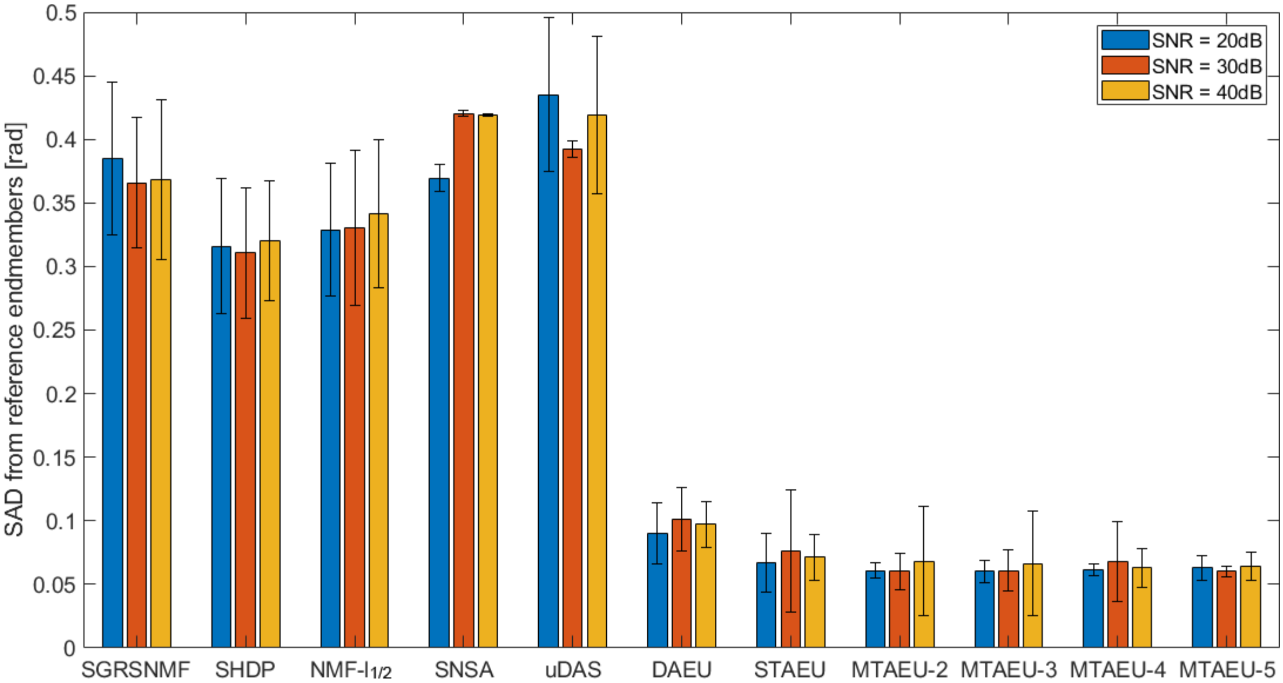


FIGURE 9. SAD from reference endmembers in radians for all methods and three different noise ratios for the Urban dataset (4 endmembers).

and 40 dB, for the Urban dataset with four endmembers estimated. As can be seen from the figure, all the methods are fairly robust to noise. Autoencoder methods are generally very robust to noise since they are denoising in nature as the network does not have the capacity to learn the added Gaussian noise. It is a bit surprising how well all the methods, except SGSRNMF and uDAS, perform with 20 dB SNR. Also, it is noticeable how small the variance for all noise levels is for the MTAEU method using 5×5 pixel neighborhood.

Our experiments confirm all of the main benefits of MTL that were listed in Sec. I. Fig. 4 shows that the method is indeed spatial in nature and the multiple tasks speed up learning considerably. Fig. 3 along with figures 5, 7, and 6 show that the MTAEU method has less variance and better

TABLE 8. Running times for all methods in minutes for the Urban dataset. The running time for the proposed method increases roughly linearly with increasing neighborhood size.

MTLAEU	STAEU	DAEU	SNSA	uDAS	SHDP	SGSRNMF	NMF-$\ell_{1/2}$
1-6	1	2	37	16	2	240	2

consistency than the DAEU and STAEU methods and that 3×3 neighborhoods are optimal, demonstrating that multitasking leads to better consistency and lower variance of endmember solutions. This is also true for the extracted abundance maps as our results also demonstrate. Lastly, the average running times for all methods for the Urban dataset can be seen in Table 8.

IV. CONCLUSION

This paper introduced a novel autoencoder based method for HSU that uses many autoencoders in parallel to benefit from multitask learning and exploit the spatial correlations in an HSI. The method is called MTAEU and was evaluated using two real HSIs and compared to some state-of-the-art methods and was found to perform significantly better than the comparison methods. The results for the two datasets used confirm that MTL

(a) makes the method spatial in nature.
(b) speeds up the convergence of the method.
(c) leads to better consistency.
(d) lowers the variance of estimated endmembers and abundance maps.

Our experiments also show that the method does indeed exploit the spatial correlations in an HSI.

ACKNOWLEDGMENT

The authors would like to thank professor Jun Li for providing us with the code for the method 'Stacked Nonnegative Sparse Autoencoders for Robust Hyperspectral Unmixing' (SNSA) described in [47] and Ying Qu for making the code for the method uDAS: An untied denoising autoencoder with sparsity for spectral unmixing described in [48] available on GitHub.

REFERENCES

[1] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.

[2] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[3] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: https://arxiv.org/abs/1706.05098

[4] J. Baxter, "A Bayesian/information theoretic model of learning to learn via multiple task sampling," *Mach. Learn.*, vol. 28, no. 1, pp. 7–39, Jul. 1997.

[5] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 42–50.

[6] 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.

[7] 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 J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

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

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

[10] S. Zhang, A. Agathos, and J. Li, "Robust minimum volume simplex analysis for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6431–6439, Nov. 2017.

[11] J. M. P. Nascimento and J. M. Bioucas-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] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.

[20] M.-D. Iordache, J. 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.

[21] 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.

[22] M.-D. Iordache, J. 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.

[23] 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.

[24] R. T. Albayrak, A. C. Gurbuz, and B. Gunyel, "Compressed sensing based hyperspectral unmixing," in *Proc. 22nd Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2014, pp. 1438–1441.

[25] Xusu, "Compressive sensing for endmember extraction," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 1345–1348.

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

[27] Z. Mitraka, F. D. Frate, and F. Carbone, "Spectral unmixing of urban landsat imagery by means of neural networks," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, Mar. 2015, pp. 1–4.

[28] G. A. Licciardi and F. D. Frate, "Pixel unmixing in hyperspectral data by means of neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4163–4172, Nov. 2011.

[29] Y. Su, A. Mariononi, 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.

[30] S. Ozkan, B. Kaya, E. Esen, and G. B. Akar, "EndNet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," 2017, *arXiv:1708.01894*. [Online]. Available: https://arxiv.org/abs/1708.01894

[31] X. Zhang, C. Li, J. Zhang, Q. Chen, J. Feng, L. Jiao, and H. Zhou, "Hyperspectral unmixing via low-rank representation with space consistency constraint and spectral library pruning," *Remote Sens.*, vol. 10, no. 2, p. 339, Feb. 2018.

[32] S. Zhang, J. Li, H.-C. 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[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*, vol. 10198, May 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] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," May 2015, *arXiv:1505.00853*. [Online]. Available: https://arxiv.org/abs/1505.00853

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

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167?context=cs

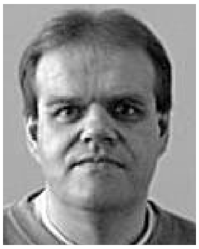
[46] F. Zhu, "Spectral unmixing datasets with ground truths," *CoRR*, to be published.

[47] 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.

[48] 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.

[49] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, Oct. 2012.

BURKNI PALSSON received the B.S. degrees in mathematics and 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 currently 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.



JOHANNES R. SVEINSSON (SM'02) received the B.S. degree from the University of Iceland, Reykjavík, 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 Department of Electrical and Computer Engineering 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 currently a Professor with the Department of Electrical and Computer Engineering, University of Iceland. His current research interests are in systems and signal theory. He received the Queen's Graduate Awards from Queen's University. He is also a co-recipient of the 2013 IEEE GRSS Highest Impact Paper Award.

MAGNUS O. ULFARSSON 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 currently 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. He has been an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, since 2018.