CNN-based augmentation strategy for spectral unmixing datasets considering spectral variability

Anastasiadis, Johannes, Heizmann, Michael

Johannes Anastasiadis, Michael Heizmann, "CNN-based augmentation strategy for spectral unmixing datasets considering spectral variability," Proc. SPIE 11533, Image and Signal Processing for Remote Sensing XXVI, 115330V (20 September 2020); doi: 10.1117/12.2575875

Event: SPIE Remote Sensing, 2020, Online Only

CNN-based augmentation strategy for spectral unmixing datasets considering spectral variability

Johannes Anastasiadis and Michael Heizmann

Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstraße 16, 76187 Karlsruhe, Germany

Dedicated to Prof. Dr.-Ing. Fernando Puente León

ABSTRACT

Spectral unmixing is often relying on a mixing model that is only an approximation. Artificial neural networks have the advantage of not requiring model knowledge. Additional advantages in the domain of spectral unmixing are the easy handling of spectral variability and the possibility to force the sum-to-one and the non-negativity constraints. However, they need a lot of significant training data to achieve good results.

To overcome this problem, mainly for classification problems, augmentation strategies are widely used to increase the size of training datasets synthetically. Spectral unmixing can be considered as a regression problem, where data augmentation is also feasible. One intuitive strategy is to generate spectra based on abundances that do not occur in the training dataset, while taking spectral variability into account.

For the implementation of this approach, we use a convolutional neural network (CNN), where the input variables are extended by random values. This allows spectral variability to be taken into account. The random inputs are re-sampled for each data point in every epoch. During training the CNN learns the mixing model and the characteristic spectral variability of the training dataset. Additional spectra can be generated afterwards for any given abundances to extend the original training dataset.

Because the generative CNN minimizes the error between generated spectra and the corresponding ground truth for the whole dataset during training, the variance of the spectra based on the same abundances is lower than in the training data. We have investigated two approaches for improvement. One is to increase the variance of the random input variables when generating new spectra. For the second, the estimated covariance matrices are considered by the objective function.

The presented method is evaluated with real data, which were captured in our image processing laboratory. We found that the augmentation of the training dataset with the presented strategy leads to an improvement for spectral unmixing of the test dataset.

Keywords: Spectral unmixing, spectral variability, data augmentation, convolutional neural network

1. INTRODUCTION

In remote sensing, hyperspectral images (HSI) are an important source of information about the properties of Earth's surface. In recent years, they have also been used in industrial applications such as the food industry.[1] In all cases they have the advantage of providing a large amount of information, despite being contactless and non-destructive. This is because, unlike colour images, they have not only three colour channels (RGB), but many narrow-band wavelength channels. In remote sensing, the distance to the object under investigation is large and therefore individual pixels of a hyperspectral camera cover large areas. Therefore, there are often several components, also referred to as endmembers, in one pixel. This also applies to fine-grained materials at

Further author information: (Send correspondence to Johannes Anastasiadis)
Johannes Anastasiadis: E-mail: anastasiadis@kit.edu,
Michael Heizmann: E-mail: michael.heizmann@kit.edu

smaller distances. To determine the relative proportions (abundances) of the endmembers in those mixed pixels, spectral unmixing is used.[2]

Spectral unmixing is typically performed using mixing models, such as the linear mixing model (LMM).[2] These models do not reflect the accurate physical conditions, but are approximations that work well enough in many cases. The best-working mixing model depends on the particular application.[3] Artificial neural networks (ANNs) do not require the assumption of a model and have achieved significant improvements in many areas of signal processing. For spectral unmixing, ANNs have also achieved promising results.[4] They have the advantage compared to most mixing models that they consider spectral variability implicitly. Mixing models that take into account the spectral variability would be, for instance, the extended linear mixing model (ELMM)[5] and the generalized linear mixing model (GLMM).[6] In addition, there is an iterative algorithm that minimizes the violation of physical constraints by varying the endmember spectra.[7]

However, in order to achieve good results, artificial neural networks require large amounts of labelled and representative training data. Otherwise, especially with a regression problem, such as spectral unmixing, there is a high risk of overfitting. While large amounts of data exist for RGB images, there are only few labelled hyperspectral images. Various techniques are used to address this problem, such as unsupervised training[8] or the use of convolutional neural networks (CNNs).[4,9] Another approach is the augmentation of existing data, which is frequently done for classification tasks.[10,11] For spectral unmixing it is suitable to generate mixed spectra with abundances that do not occur in the training dataset. We already presented a mixing model based approach in the past, where we also considered spectral variability.[12] It uses various mixing models to create artificial mixed spectra from sets of endmember spectra. The advantage is that depending on the mixing model only endmember spectra are needed. In this paper we present an alternative that uses a generative CNN instead of a mixing model to generate additional mixed spectra considering spectral variability. In addition to the endmember spectra, a certain number of mixed spectra are required. There is a similar approach that uses a variational autoencoder to generate additional spectra, but only for the spectra of the endmembers.[13]

The further paper is organized as follows: In Sections 2 and 3 the basics of spectral unmixing and CNNs are summarized, which are necessary to understand the presented augmentation strategy. Section 4 then describes the generative CNN for data augmentation and the ideas behind it. In Section 5, the generated data and the unmixing results are evaluated using real data. Finally the paper is summarized and conclusions are drawn in Section 6.

2. SPECTRAL UNMIXING

Spectral unmixing is the process of estimating material fractions (abundances) based on the spectrum of a mixture of endmembers.[2] If the spectra of the endmembers are known, this is referred to as supervised spectral unmixing. Most unmixing approaches are based on a mixing model, with the LMM being the most commonly used as it is a valid approximation for many applications:[2,14–17]

$$\mathbf{y} = \sum_{p=1}^{P} \mathbf{m}_p \, a_p + \boldsymbol{\Omega} = \mathbf{M}\,\mathbf{a} + \boldsymbol{\Omega}\,. \tag{1}$$

Here $\mathbf{y} \in \mathbb{R}^{\Lambda}$ is a pixel of a hyperspectral image sampled at Λ wavelength channels, $\mathbf{M} = [\mathbf{m}_1, ..., \mathbf{m}_P] \in \mathbb{R}^{\Lambda \times P}$ are the spectra of the up to P involved endmembers, and $\mathbf{a} = [a_1, ..., a_P]^{\mathrm{T}} \in \mathbb{R}^{P}$ are their abundances. Differences between $\mathbf{y}$ and the weighted sum of the endmember spectra are covered by $\boldsymbol{\Omega} \in \mathbb{R}^{\Lambda}$. In addition, non-linear mixing models exist,[3] which will not be discussed in detail here. The goal is an estimation of the abundances $\hat{\mathbf{a}} \in \mathbb{R}^{P}$, which minimizes the squared error:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{M}\,\mathbf{a}\|_2^2\,. \tag{2}$$

Equation (2) can be optimized using the pseudo-inverse $\mathbf{M}^{\dagger}$ if $\Lambda > P$ and $\operatorname{rank}(\mathbf{M}) = P$, which is the case with HSI:

$$\hat{\mathbf{a}} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{y} = \mathbf{M}^{\dagger}\mathbf{y}\,. \tag{3}$$

This estimation is not optimal for a general $\boldsymbol{\Omega}$. Regardless of the selected mixing model, there are two constraints for $\mathbf{a}$, which ensure physical validity: The non-negativity constraint (4) and the sum-to-one constraint (5).

$$a_p \geq 0 \quad \forall p \tag{4}$$

$$\sum_{p=1}^{P} a_p = 1 \tag{5}$$

These constraints can be used to improve the estimation (2). An approach considering (4) and (5) is the Fully Constrained Least Squares (FCLS) algorithm.[18] Here, the Lagrangian $L : \mathbb{R}^{P+1} \to \mathbb{R}$ is used as the objective function:

$$L(\mathbf{a}, l) = \|\mathbf{y} - \mathbf{M}\,\mathbf{a}\|_2^2 - l\left(\sum_{p=1}^{P} a_p - 1\right). \tag{6}$$

The additional variable $l \in \mathbb{R}$ is called Lagrange multiplier. By using this objective function, constraint (5) is ensured. In addition, negative $\hat{a}_p$ and the corresponding endmember spectra in $\mathbf{M}$ are removed in an iterative process so as to fulfil constraint (4).

It is assumed in the mixing models that each endmember can be represented by a single spectrum. In reality, however, endmembers may have varying spectra.[19] These variations are referred to as spectral variability (see Fig. 1). This can be caused by many factors, such as the surface structure and the lighting. There are also

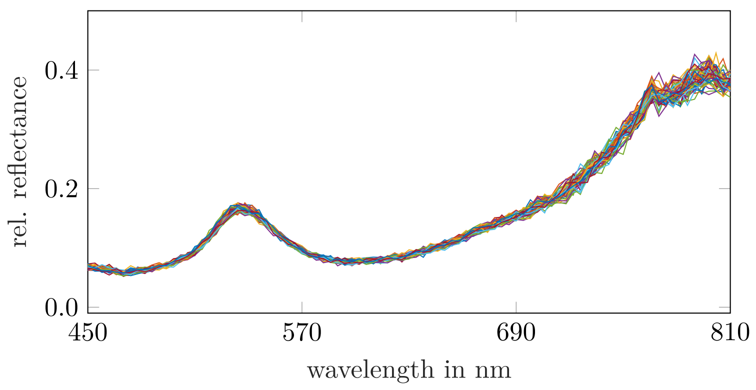


Figure 1. Examples of spectra of an endmember contained in the colour-4 dataset presented in Sec. 5.

unmixing algorithms considering the spectral variability such as the ELMM[5] and the GLMM.[6] The ELMM extends the LMM to

$$\mathbf{y} = \mathbf{M}\,\boldsymbol{\Phi}\,\mathbf{a} + \boldsymbol{\Omega}\,, \tag{7}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{P \times P}$ is a diagonal matrix. The elements of $\boldsymbol{\Phi}$ correspond to scaling factors of the individual endmember spectra in $\mathbf{M}$. This increases the number of parameters to be optimized (in addition to the P parameters of vector $\hat{\mathbf{a}}$) to $2 \cdot P$. The GLMM adds much more parameters and extends the LMM to

$$\mathbf{y} = (\mathbf{M} \odot \boldsymbol{\Psi})\,\mathbf{a} + \boldsymbol{\Omega}\,. \tag{8}$$

The matrix $\boldsymbol{\Psi} \in \mathbb{R}^{\Lambda \times P}$ now contains a scaling factor for each wavelength channel of every endmember spectrum. Here, the number of parameters to be optimized is increased to $(\Lambda + 1)P$. The estimation strategies for both the ELMM and the GLMM satisfy the constraints (4) and (5).[5,6]

The presented conventional methods will later be compared to the CNN-based unmixing, which is introduced in the next section.

3. CONVOLUTIONAL NEURAL NETWORKS

Artificial neural networks can learn correlations on the basis of data, if correctly dimensioned. With a dataset of many mixed spectra as data points, an ANN can be trained for spectral unmixing.[4] If the spectra in the dataset show spectral variability, this is also taken into account. This section first summarizes the basics of CNNs. Readers who are familiar with these basics can skip the next subsection. Afterwards, the CNN we use for spectral unmixing is presented.

3.1 Basics

Artificial neural networks are derived from biological neurons and have the ability to learn correlations from a dataset. It is important that the dataset contains many representative examples. The aim is to achieve the best possible generalization in order to correctly process previously unseen data. If the ANN is too large or the training dataset too small, there is a potential risk of overfitting, i.e. learning the training dataset by heart. If the net is too small, this can result in underfitting, which means the ANN is not able to correctly map the relationships represented in the training dataset. Especially with regression problems, such as spectral unmixing, it has been found that overfitting can easily become a problem.

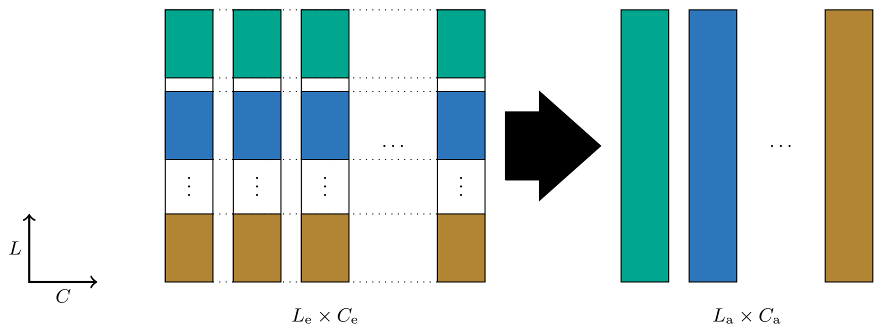
Artificial neural networks are composed of artificial neurons. In general the neurons can have arbitrary connections. However, most networks are connected in such a way that signal flow is only possible in forward direction (feed forward networks) and the neurons are arranged in layers. If each neuron of a layer is connected to each neuron of the subsequent layer, the layer is called fully connected layer. One layer can be described by

$$\mathbf{h} = \boldsymbol{\varphi}(\mathbf{W}\mathbf{x} + \mathbf{b})\,, \tag{9}$$

where $\mathbf{h} \in \mathbb{R}^K$ and $\mathbf{x} \in \mathbb{R}^J$ are the K-dimensional output and the J-dimensional input of the layer, respectively. The trainable parameters are the weights $\mathbf{W} \in \mathbb{R}^{K\times J}$ and the bias $\mathbf{b} \in \mathbb{R}^K$. The training is done by minimizing an objective function using gradient descent. The activation function $\boldsymbol{\varphi} : \mathbb{R}^K \to \mathbb{R}^K$ is applied element-wise and must be non-linear, otherwise the ANN could only model linear relationships.

In CNNs convolutional layers are used, which are not fully connected. The neurons are only locally connected and connections share weights. These layers can also be described by equation (9) but the matrix $\mathbf{W}$ of a convolutional layer is sparse. The parameters in $\mathbf{W}$ consist of the convolution kernel, which is repeated. This has advantages, especially in signal processing. The operation is spatially invariant because the convolution kernel is shifted. In addition, the local relationships are preserved. Moreover, considerably fewer parameters need to be trained, which helps to prevent overfitting.

In reality, several convolution kernels are used in each layer. This results in multiple outputs called feature maps. In the next layer these are convolved with several convolution kernels (see Fig. 2).



L
C
$L_\mathrm{e} \times C_\mathrm{e}$
$L_\mathrm{a} \times C_\mathrm{a}$

Figure 2. 1D example of a convolutional layer: The input data consists of C_e feature maps each containing L_e values. They are convolved with C_a sets of filters, where each of the sets consists of C_e filters. The C_a new feature maps are the accumulated convolutions of a set and contain L_a values each.[12]

For CNNs, pooling layers are also used, which summarize a neighbourhood of values in a single values.[11] Usually, max-pooling is used, which propagates the maximum of each neighbourhood to the next layer. Pooling has the advantage of reducing the size of the network in the subsequent layers, which reduces the computing and memory requirements. In addition, better generalization performance can be achieved.

Furthermore, batch normalization is often used with ANNs.[20] On the one hand, batch normalization compensates the different distributions of the input data mini-batches. On the other hand, it compensates the different distributions of the input data of deeper layers, which are caused by the fact that the parameters of the previous layers are adjusted during training. Batch normalization is performed before the activation function and executes the following transformation of $o_{\mathrm{e}} \in \mathbb{R}$:

$$o_{\mathrm{a}} = \delta \, \frac{o_{\mathrm{e}} - \mu_{\mathrm{B}}}{\sqrt{\sigma_{\mathrm{B}}^2 + \epsilon}} + \beta \, . \tag{10}$$

Here, μ_{B} and σ_{B}^2 are the sample mean and variance of o_{e} with respect to a mini-batch, the set of data points considered in the current iteration. The parameters $\beta \in \mathbb{R}$ and $\delta \in \mathbb{R}$ are trainable and $\epsilon \in \mathbb{R}$ is a small number that prevents a division by 0. In convolutional layers all values in a feature map share a pair of parameters β and δ, while in fully connected layers each output value gets its own pair. With the initial choice of β and δ the values can be shifted to a range of the activation function that has a high derivative, allowing for faster training. Other advantages are higher possible learning rates and a regularizing effect.[20] The next subsection presents a CNN for spectral unmixing, which consists of these elementary components.

3.2 CNN for spectral unmixing

The CNN used in this paper for spectral unmixing is similar to the one we previously used for spatially resolved spectral unmixing.[4] However, because this paper focuses only on spectra and does not consider local relations, only a one-dimensional version is used.

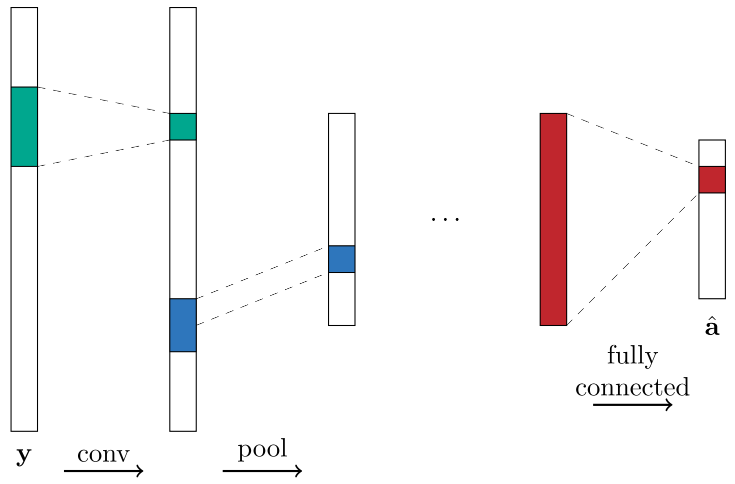


Figure 3. First layers and last layer of the presented CNN: The first layers are a convolutional layer (conv) and a pooling layer (pool). The last one is a fully connected layer. In between there are first more alternating convolutional and pooling layers and then more fully connected layers. For the purpose of a better overview, only one feature map and one convolution kernel is displayed in each layer.

The input data of the CNN are the pixels of the hyperspectral images, i.e. spectra, and the labels are the abundances of the endmembers. The CNN can be divided into two parts, with the first part extracting features and the second part using them for spectral unmixing. The first part consists of one-dimensional convolutional layers that alternate with max-pooling layers. The second part consists of fully connected layers (see Fig. 3). Batch normalization is used in every layer except the last. All layers use the Sigmoid function $\sigma : \mathbb{R} \to \mathbb{R}$ as activation function:

$$\sigma(z) = \frac{1}{1 + \mathrm{e}^{-z}} \, . \tag{11}$$

At the output a normalization with

$$\hat{a}_p = \frac{a_p^*}{\sum_{\tilde{p}=1}^{P} a_{\tilde{p}}^*} \quad \text{for} \quad p = 1, 2, \ldots, P \tag{12}$$

is performed. This normalization and the last sigmoid function force the constraints (4) and (5). It has been found that this leads to a better result than using a softmax layer at the output.[4] The CNN is trained by minimizing the mean squared error between $\mathbf{a}$ and $\hat{\mathbf{a}}$.

The presented CNN will be later used for evaluation, which is done by training the CNN with the non-augmented and the augmented training datasets. The used strategy for augmentation is presented in the next section.

4. AUGMENTATION USING A GENERATIVE CNN

A CNN like the one in Section 3 has to be trained in a supervised manner. This means that the training data consist of spectra and the corresponding abundances of the endmembers. The difficulty here is that the output variables are continuous. However, only discrete output values are possible in a limited training dataset, which is why these should be sampled in as small steps as possible. This is usually not the case. For this reason, there is a risk that the CNN, beyond the values available in the training dataset, will generalise poorly. In addition, the spectral variability (see Sec. 2) leads to a non-negligible variance in the input values for the same output values. The CNN is thus trained to determine the same output value for many input values, which further increases the difficulty of a successful regression. Ideally, CNN learns the relationship between the abundances and the mixed spectra and at the same time acquires an invariance to spectral variability.

The approach we use in this paper to address this problem is to artificially generate additional training data. For this purpose, a generative CNN is used, where the abundances $\mathbf{a}$ are the input values and the generated spectra $\hat{\mathbf{y}} \in \mathbb{R}^{\Lambda}$ are the output values (see Fig. 4). As a result, much more different output data are available in the training dataset of the generative CNN. In reality, the spectra depend not only on the abundances but also on the environmental conditions, which result in spectral variability. Because those conditions are not represented as labels in the training dataset, additional statistically independent random input variables $\boldsymbol{X} \in \mathbb{R}^{R}$ are added. These can be interpreted as the environmental information that has already been processed by some ANN. In order to address as many scenarios as possible, the random input values are redrawn in every epoch.

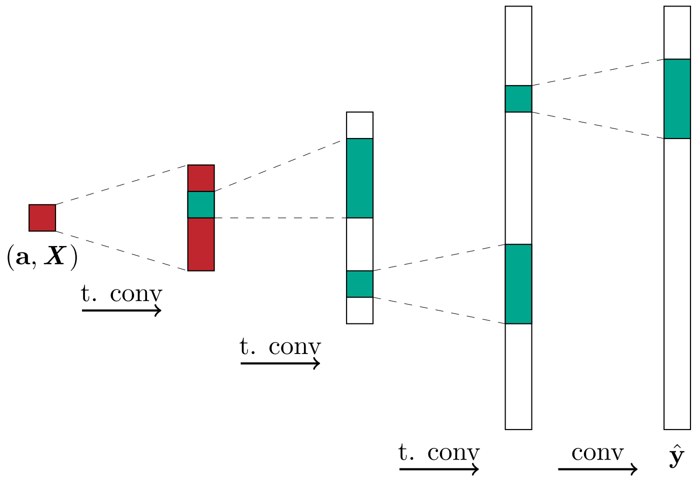


Figure 4. The generative CNN as used for the results of this paper. Three transposed convolutions (t. conv) are used to up-sample from abundances and random values to spectra. Depending on the dataset, more or less layers may be suitable. For the last layer a convolution is used to merge the feature maps, which are not shown in this figure due to clarity.

The generative CNN consists basically of convolutional layers, which apply zero-padding to create larger structures from smaller ones. This procedure is also called transposed convolution. Every layer in this network uses the sigmoid function (11) as its activation function. Input variables are the abundances $\mathbf{a}$ and the random values $\boldsymbol{X}$, which are interpreted as 1×1 sized feature maps. The output variables are mixed spectra ($\Lambda \times 1$ feature maps). The generative CNN is trained by minimizing the cross-entropy loss function J_1 between the generated and the real spectra:

$$J_1(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{\Lambda} \sum_{\lambda=1}^{\Lambda} \left[y_\lambda \log \hat{y}_\lambda + (1 - y_\lambda) \log(1 - \hat{y}_\lambda) \right]. \tag{13}$$

After training the CNN is able to generate spectra for any given abundances. By drawing $\boldsymbol{X}$ several times, multiple spectra can be generated for the same abundances, so the synthetic spectra also show spectral variability.

After this CNN has been trained for many epochs, the spectra generated will be close to the mean spectrum, i.e. have a lower spectral variability than the training data. This is because $\boldsymbol{X}$ is redrawn for each spectrum in every epoch, and in the training data, the spectra tend to be concentrated around the mean spectrum. However, it is necessary to redraw $\boldsymbol{X}$ in every epoch in order to get as many input values of $\boldsymbol{X}$ as possible and be later able to generate new spectra using any realization of $\boldsymbol{X}$. To address this problem we present two different approaches. The first one is to simply increase the variance of $\boldsymbol{X}$ when generating new spectra compared to the training.

The second approach uses regularisation to increase the spectral variability. For this purpose, the objective function (13) is extended by a term which minimizes the squared Frobenius norm between the estimated covariance matrices of all I real $\mathbf{Y}^{\mathbf{a}} = [\mathbf{y}_1^{\mathbf{a}}, \ldots, \mathbf{y}_I^{\mathbf{a}}]$ and all I generated $\hat{\mathbf{Y}}^{\mathbf{a}} = [\hat{\mathbf{y}}_1^{\mathbf{a}}, \ldots, \hat{\mathbf{y}}_I^{\mathbf{a}}]$ spectra resulting from identical abundances. The extended objective function (in contrast to (13) presented for all I spectra) is:

$$J_2(\mathbf{Y}^{\mathbf{a}}, \hat{\mathbf{Y}}^{\mathbf{a}}) = -\frac{1}{\Lambda I} \sum_{i=1}^{I} \sum_{\lambda=1}^{\Lambda} \left[y_{\lambda i}^{\mathbf{a}} \log \hat{y}_{\lambda i}^{\mathbf{a}} + (1 - y_{\lambda i}^{\mathbf{a}}) \log(1 - \hat{y}_{\lambda i}^{\mathbf{a}}) \right] + \mathrm{k}_1 \left\| \mathrm{K}_{\mathbf{Y}^{\mathbf{a}}\mathbf{Y}^{\mathbf{a}}}, \mathrm{K}_{\hat{\mathbf{Y}}^{\mathbf{a}}\hat{\mathbf{Y}}^{\mathbf{a}}} \right\|_{\mathrm{F}}^2. \tag{14}$$

The presented generative CNN is evaluated in the next section by generating additional data to improve the performance of a CNN for spectral unmixing. In the next section, the different strategies presented here are compared to each other and to spectral unmixing methods that do not use neural networks.

5. EXPERIMENTAL RESULTS

To evaluate our generative CNN we use real data acquired in our image processing lab. All datasets have 91 wavelength channels with an average width of 4 nm from 450 nm to 810 nm and 400 spectra per mixture. In order to be able to capture HSIs of mixtures of the endmembers, we have filled them into small boxes (see Fig. 5).

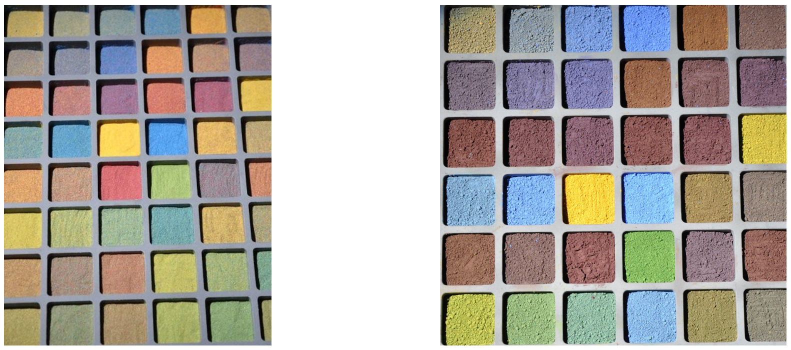


Figure 5. Images of quartz sands (left) and colour powders (right) in boxes to be captured by our hyperspectral camera.

Two of the datasets consist of mixtures of coloured quartz sand. The first of them (quartz-3) contains 45 mixtures of maximum 3 components, which vary in abundance steps of 0.125. The second one (quartz-4) contains 56 mixtures of maximum 4 components, which vary in abundance steps of 0.2. The datasets quartz-3 and quartz-4 have a low spectral variability and show an almost linear mixing relation. The third dataset consists of 56 mixtures of colour powders (colour-4), which also have a maximum of 4 components. Here, the components are varied in steps of 0.2, too. The colour-4 dataset shows a high non-linearity between mixed spectrum and endmember spectra and a high spectral variability. Therefore, its spectra are more difficult to unmix than those of the quartz sand datasets.

All datasets are divided into a training dataset and a test dataset. The division is based on the abundances. For the quartz-3 dataset, all mixtures in which no abundance has one of the values 0.125, 0.375, 0.625 or 0.875 are included in the training dataset, the remaining mixtures in the test dataset. This results in a total of 15 mixtures in the training dataset and 30 in the test dataset. For the quartz-4 and colour-4 datasets, all mixtures in which no abundance has one of the values 0.2 or 0.8 are part of the training dataset, the remaining mixtures of the test dataset. This makes a total of 16 mixtures in the training dataset and 40 in the test dataset. The training datasets are used to train the generative CNN in order to be able to generate additional data. The CNN for spectral unmixing is trained afterwards with the extended training datasets. The test datasets are used to evaluate all methods in a comparison.

5.1 Evaluation of generated spectra

In this subsection we compare the generated spectra with the spectra in the corresponding test dataset. The generative CNN was chosen as in Fig. 4, with a length of the convolution kernels of 5 and an upsampling rate of 2. The exception is the first convolution kernel with a length of 23. The numbers of feature maps from input to output are $P + R$, 32, 32, 16, and 1. For $\boldsymbol{X}$ we use a three-dimensional standard normal distribution. Adam optimizer[21] is used with a learning rate of 0.01 (other parameters as suggested in the paper) to train the generative CNN over 4000 epochs. An example of generated spectra compared to the spectra of the test dataset is shown in Fig. 6. In order to be able to compare the sets of spectra quantitatively with each other, we introduce

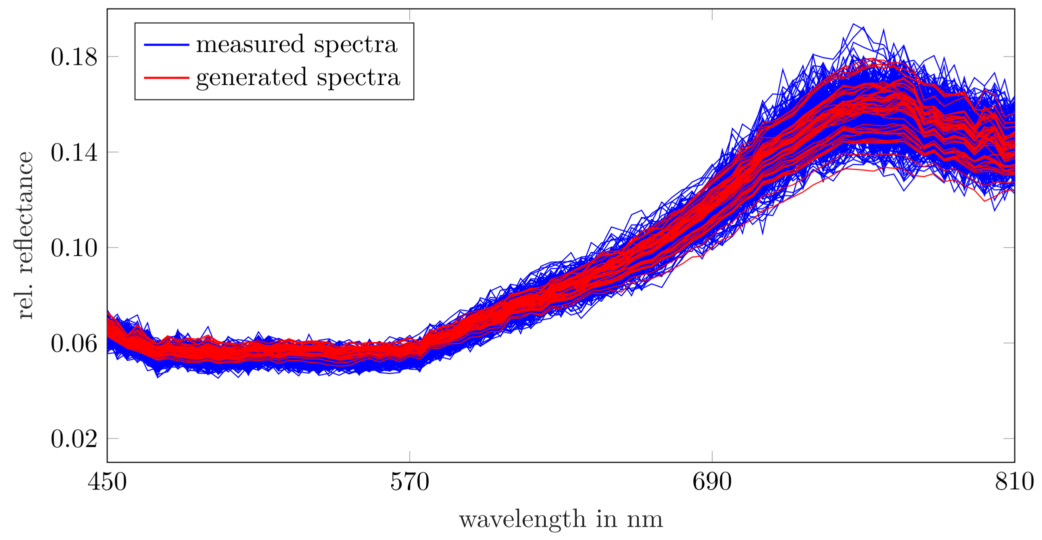


Figure 6. Example comparing measured spectra (test dataset, 400 samples) with generated spectra (50 samples). The shown spectra are from the colour-4 dataset with $\mathbf{a} = [0, 0.6, 0.2, 0.2]^{\mathrm{T}}$ and J_2 was used for the training of the generative CNN.

the average minimum (euclidean) norm between a set of I measured spectra and a set of H generated spectra:

$$\Delta_{\mathrm{AMN}} = \frac{1}{I} \sum_{i=1}^{I} \min_{h} \|\mathbf{y}_i, \hat{\mathbf{y}}_h\|_2 \,. \tag{15}$$

If this quality measure is averaged over all groups of spectra we call it global average minimum norm Δ_{GAMN}. Table 1 shows Δ_{GAMN} for all datasets and presented methods for generating synthetic mixed spectra. To obtain these results, we compared 400 generated spectra with the corresponding 400 spectra from the test dataset. It is

Table 1. Comparison of Δ_{GAMN} for all test datasets and the different variants of our generative CNN. The attributes "same" and "double" refer to the variance of the random inputs during spectra generation compared to training.

Δ_{GAMN}	trained with J_1 (same)	trained with J_1 (double)	trained with J_2
quartz-3	0.1219	0.1168	0.0812
quartz-4	0.1113	0.1070	0.0787
colour-4	0.1242	0.1167	0.0967

shown that the best results can be achieved by regularizing the covariance matrices. By increasing the variances of the elements of $\boldsymbol{X}$, only a slight improvement over the standard version can be obtained. The results differ only slightly between the different datasets. The quartz-4 dataset can achieve the best results. This is because the colour-4 dataset has a higher non-linearity and the quartz-3 dataset has been split differently into training and test dataset.

Whether these results apply to the results of the CNN for spectral unmixing, which is trained using the augmented training data, will be investigated in the next subsection.

5.2 Evaluation of spectral unmixing

The test data of the datasets are to be spectrally unmixed in this subsection. The CNN from Section 3 is trained with different training datasets, that are augmented by the generative CNN (see Section 4). For comparison, the same CNN is trained with the non-augmented training datasets. Additionally, the results are compared with methods that are not based on a neural network (see Section 2). The root-mean-square error

$$\Delta_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{P}\sum_{p=1}^{P}(\hat{a}_{pn} - a_{pn})^2}\,, \tag{16}$$

based on all N spectra of a test dataset, is used as a measure of performance. The constraints (4) and (5) are fulfilled by all the methods compared here and, therefore, do not need to be further investigated.

As a method based on the LMM, the FCLS algorithm is used here. For the unmixing methods based on the ELMM and GLMM, the solution found with the FCLS algorithm was used as initialization. In all these methods, the mean spectra of all acquired endmember spectra are used in the matrix $\mathbf{M}$.

For the generative CNN the same configuration is used as in the last subsection. The CNN used here for spectral unmixing consists of a first part with three layers as shown in Fig. 3 (left). This is followed by a second part consisting of two layers as shown in Fig. 3 (right). The length of the convolution kernels is 3 and the numbers of feature maps from input to output are 1, 16, 32, 64, 64, and 1. Adam optimizer[21] is used with a learning rate of 0.01 (other parameters as suggested in the paper) to train the CNN for spectral unmixing. The number of epochs depends on the dataset and how early overfitting problems occur. Therefore, the CNN is trained with the quartz-3 dataset for 81 (not augmented) or respectively 251 (augmented) epochs, the quartz-4 dataset for 21 (not augmented) or respectively 51 (augmented) epochs, and the colour-4 dataset for 21 (both) epochs.

For the evaluation of the augmentation strategy, the existing training datasets are systematically augmented. The quartz-3 training dataset is extended in such a way that all possible combinations of mixtures are available in steps of $\frac{1}{5}$ (CNN-A5), $\frac{1}{8}$ (CNN-A8), $\frac{1}{10}$ (CNN-A10), or $\frac{1}{20}$ (CNN-A20). The datasets consisting of up to 4 components, quartz-4 and colour-4, are extended in a manner that all possible combinations of mixtures are available in steps of $\frac{1}{4}$ (CNN-A4), $\frac{1}{5}$ (CNN-A5), $\frac{1}{8}$ (CNN-A8), or $\frac{1}{16}$ (CNN-A16). The original training data are always contained in the training datasets. If a mixture is already contained in the training dataset, no artificial mixed spectra are generated for that mixture. Each augmented dataset is generated three times (compare with Table 1): First with the generative CNN that was trained with the objective function J_1 (no suffix), then with a generative CNN that has been trained with the objective function J_1 and the variance of $\boldsymbol{X}$ has been doubled while generating the spectra (-D), and finally with a generative CNN that was trained with J_2 (-J_2).

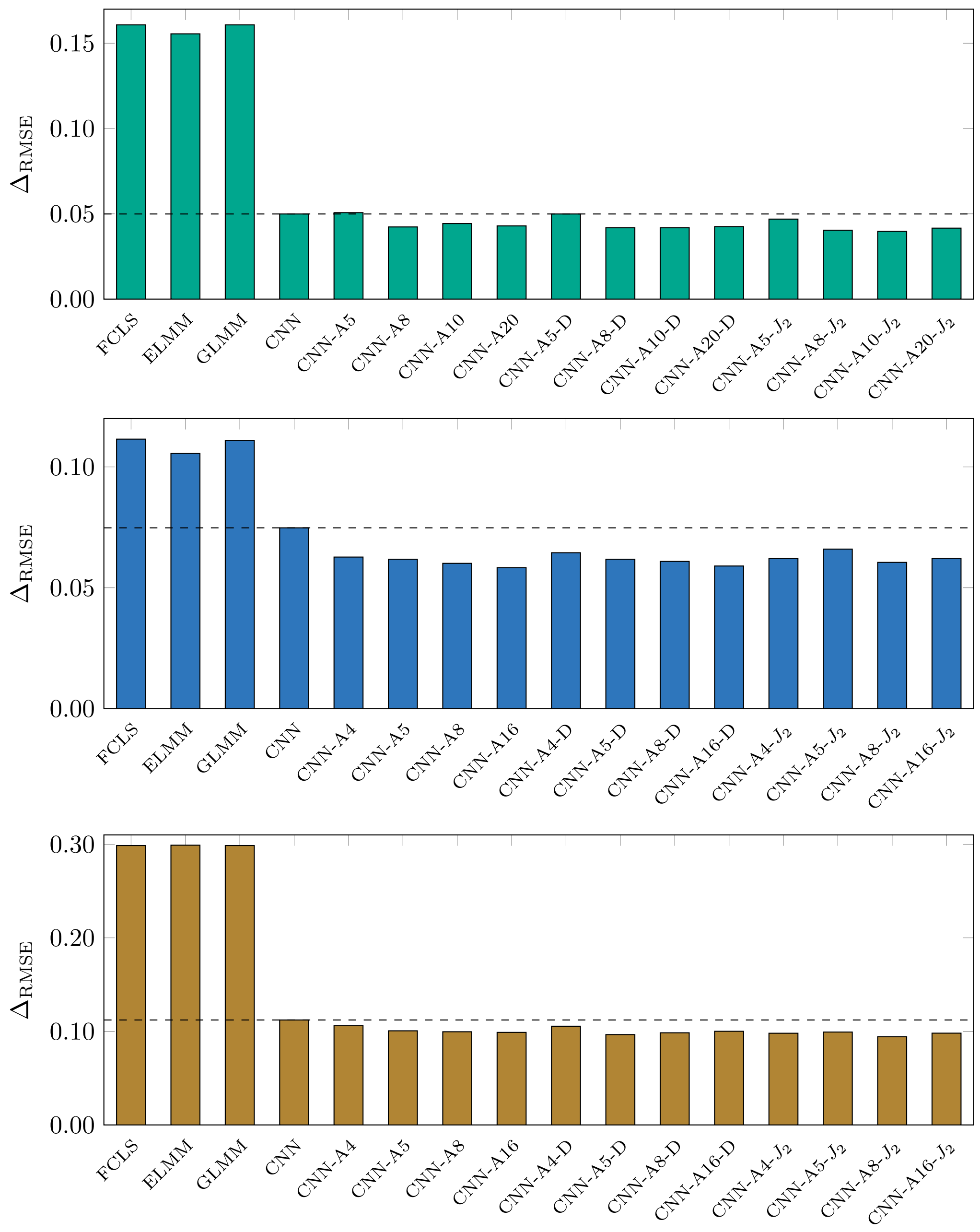


Figure 7. Root-mean-square error of the abundances for the different spectral unmixing methods applied on the quartz-3 (top), the quartz-4 (middle), and the colour-4 (bottom) dataset, respectively. The dashed line is used for better comparability and corresponds to Δ_{RMSE} using a CNN with the non-augmented training data.

Figure 7 displays Δ_{RMSE} for the different unmixing methods and datasets. It is evident that CNN-based approaches work much better than those based on the (E/G)LMM. In return, the latter, of which the ELMM based method works best here, have the advantage that they perform independently of any training data. It is also noticeable that for the quartz-3 dataset, the LMM-based methods perform worse than for the quartz-4 dataset, although it is exactly the opposite for the CNN-based methods. This is because the quartz-4 dataset is easier to unmix, but the available training data for CNN has gaps in the labels (see above). Augmentation with the generative CNN results in an improvement in almost all cases. The only exception is the quartz-3 dataset, where a minimal increase of Δ_{RMSE} occurs when only a few augmented spectra are used. In most cases it is helpful to have small steps in the labels of the augmented data. Improving the adaptation of the spectral variability of the original training dataset has a small beneficial effect. The regularization (use of J_2) works better than doubling the variance of the elements of $\boldsymbol{X}$. The quartz-4 dataset is the exception here, however, it also shows a rather low spectral variability.

6. CONCLUSION

In this paper we have presented an approach that generates additional mixed spectra based on endmember spectra and mixed spectra and that takes into account spectral variability. For this purpose, a generative CNN is used, which has random inputs in addition to the abundances, representing other unknown input variables that cause spectral variability. This can be used to augment existing training datasets, and these can in turn be used to train a CNN for spectral unmixing. In addition, we presented possibilities to improve the adaptation of spectral variability. Using real hyperspectral images, it has been shown that CNN-based methods are superior to methods based on the (G/E)LMM. With sufficient additional generated spectra, a decrease of the root-mean-square error of the abundances was achieved by the augmentation for all datasets. In many cases the regularization of the estimated covariance matrices resulted in a slight additional improvement.

The presented approach could also be suitable for further applications and is not limited to spectral unmixing. Whenever a regression problem is to be solved where the measured variables show a variance caused by unknown variables, it would be imaginable to use an adapted version of this approach to augment limited training data. For higher-dimensional output variables (in this case the abundances), online augmentation would also be feasible, allowing a great amount of artificial data to be generated directly during training, which would otherwise have far too high memory requirements.

REFERENCES

[1] Gowen, A., O'Donnell, C., Cullen, P., Downey, G., and Frias, J., "Hyperspectral imaging – an emerging process analytical tool for food quality and safety control," *Trends in Food Science & Technology* **18**(12), 590–598 (2007).

[2] Keshava, N. and Mustard, J. F., "Spectral unmixing," *IEEE signal processing magazine* **19**(1), 44–57 (2002).

[3] Dobigeon, N., Altmann, Y., Brun, N., and Moussaoui, S., "Linear and nonlinear unmixing in hyperspectral imaging," in [*Data Handling in Science and Technology*], Ruckebusch, C., ed., **30**, 185–224, Elsevier (2016).

[4] Anastasiadis, J. and Puente León, F., "Spatially resolved spectral unmixing using convolutional neural networks (German paper)," *tm – Technisches Messen* **86**(s1), 122–126 (2019).

[5] Veganzones, M. A., Drumetz, L., Tochon, G., Dalla Mura, M., Plaza, A., Bioucas-Dias, J., and Chanussot, J., "A new extended linear mixing model to address spectral variability," in [*2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*], 1–4, IEEE (2014).

[6] Imbiriba, T., Borsoi, R. A., and Bermudez, J. C. M., "Generalized linear mixing model accounting for endmember variability," in [*2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 1862–1866, IEEE (2018).

[7] Krippner, W., Anastasiadis, J., and Puente León, F., "Robust iterative estimation of material abundances based on spectral filters exploiting the svd," in [*Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*], **10986**, 109861T, International Society for Optics and Photonics (2019).

[8] Wang, M., Zhao, M., Chen, J., and Rahardja, S., "Nonlinear unmixing of hyperspectral data via deep autoencoder networks," *IEEE Geoscience and Remote Sensing Letters* **16**(9), 1–5 (2019).

[9] LeCun, Y. and Bengio, Y., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks* **3361**(10), 1995 (1995).

[10] Simard, P. Y., Steinkraus, D., Platt, J. C., et al., "Best practices for convolutional neural networks applied to visual document analysis.," in [*Icdar*], **3**(2003) (2003).

[11] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[12] Anastasiadis, J., Benzing, P., and Puente León, F., "Generation of artificial data sets to train convolutional neural networks for spectral unmixing (German paper)," *tm – Technisches Messen* **0**(ahead-of-print) (2020).

[13] Borsoi, R. A., Imbiriba, T., and Bermudez, J. C. M., "Deep generative endmember modeling: An application to unsupervised spectral unmixing," *IEEE Transactions on Computational Imaging* **6**, 374–384 (2020).

[14] Bauer, S., Stefan, J., and Puente León, F., "Hyperspectral image unmixing involving spatial information by extending the alternating least-squares algorithm," *tm – Technisches Messen* **82**(4), 174–186 (2015).

[15] Krippner, W., Bauer, S., and Puente León, F., "Optical determination of material abundances in mixtures (German paper)," *tm – Technisches Messen* **84**(3), 207–215 (2017).

[16] Krippner, W., Bauer, S., and Puente León, F., "Considering spectral variability for optical material abundance estimation," *tm – Technisches Messen* **85**(3), 149–158 (2018).

[17] Krippner, W. and Puente León, F., "Band selection and estimation of material abundances using spectral filters (German paper)," *tm – Technisches Messen* **85**(6), 454–467 (2018).

[18] Heinz, D., Chang, C.-I., and Althouse, M. L., "Fully constrained least-squares based linear unmixing," in [*IEEE 1999 International Geoscience and Remote Sensing Symposium.*], **2**, 1401–1403, IEEE (1999).

[19] Smith, M. O., Adams, J. B., and Sabol, D. E., "Spectral mixture analysis-new strategies for the analysis of multispectral data," in [*Imaging spectrometry – a tool for environmental observations*], Hill, J. and Mégier, J., eds., 125–143, Springer (1994).

[20] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR* **abs/1502.03167** (2015).

[21] Kingma, D. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* (2014).