Gaussian Process Inspired Neural Networks for Spectral Unmixing Dataset Augmentation

Johannes Anastasiadis and Michael Heizmann

Institute of Industrial Information Technology (IIIT),
Karlsruhe Institute of Technology (KIT),
Hertzstr. 16, 76187 Karlsruhe, Germany
anastasiadis@kit.edu

Abstract. Hyperspectral imaging is increasingly used for product monitoring in industrial processes. Spectral unmixing is an important task in this context. As in many other areas of signal processing, neural networks also provide promising results for spectral unmixing. Unfortunately, it is very time-consuming to prepare labelled training data for the neural networks. To address this problem, this paper presents a method where small training datasets are augmented to improve spectral unmixing performance. Inspired by Gaussian processes, simple neural networks are trained which are capable of generating additional training data. These are similar to the original training data but cover areas in the continuous label space that are not covered by the original data.

Keywords: Spectral unmixing, spectral variability, data augmentation, neural network, Gaussian process

1 Introduction

Since they are non-contact and non-destructive, optical measurement methods are often used for monitoring industrial processes. This also includes checking for the correct product composition. For this task hyperspectral images are often used because they have a finely sampled spectrum in each pixel characterizing the materials involved [1]. In contrast, conventional colour images are usually not able to solve this problem sufficiently because these only contain three colour channels and different spectra can result in the same colour channel values. Spectral unmixing is needed if more than one material is contained in a pixel and therefore only a mixed material spectrum is available. The aim is to get the relative proportions, the abundances, of the pure materials covered by the pixel [2]. This is often done using mixing models, such as the linear mixing model (LMM), which has proven to be a good approximation. However, depending on the problem, more complex mixing models can provide better results but are also more difficult to use [3]. In addition, there is spectral variability, which can be taken into account by the models using additional parameters.

Instead of a model-based, a data-based approach is also feasible. Artificial neural networks in particular have achieved great success in recent years. This is also true for spectral unmixing and comes with additional advantages [4]. One of them is that the non-negativity and the sum-to-one constraints can be enforced by output layer design. The other advantage is that spectral variability can be taken into account if it is contained in the training data [5]. Ideally, the trained neural networks are robust to spectral variability. To achieve this and a good spectral unmixing performance, lots of significant training data are needed, which are often not available in this domain.

Mainly for classification problems, augmentation strategies are widely used to increase the size of training datasets synthetically and improve performance [6, 7]. Data augmentation can also be used with spectral unmixing, which can be considered as a regression problem. Here it appears useful to generate spectra based on abundances that do not occur in the original training dataset. Ideally, spectral variability is also taken into account by generating many spectra for each abundance set. We have shown in a previous paper that this improves spectral unmixing performance of convolutional neural networks (CNNs) [8]. There we used a generative convolutional neural network with additional random inputs for spectral variability to learn the relationship between abundances and mixed spectra.

In this paper we model the spectra as Gaussian processes with the wavelength as the index and the abundances as parameters. Gaussian processes are defined by the mean and covariance functions. Here we are dealing with functions which depend on the wavelength and are parametrised by the abundances. To learn these functions, we use simple neural networks. To generate the additional training data, further abundances are given to the neural networks. Spectral variability is taken into account by the generation of multiple spectra.

In a previous paper we have already modelled the spectra as Gaussian random vectors [9]. However, that paper was not about dataset augmentation, but about model-based data generation taking spectral variability into account. For the model-based approach, only a set of spectra of each pure material is needed, whereas here we need additional sets of spectra of material mixtures. The additional information should lead to better spectral unmixing performance. Another approach exists where spectral unmixing is achieved by direct application of Gaussian process regression [10], however, not for the augmentation of training data.

The rest of the paper is organized as follows: Section 2 summarises the necessary basics regarding spectral unmixing. Afterwards in Section 3 the proposed approach is described in detail. The evaluation of the approach is given in Section 4. The paper is summarized and a conclusion is drawn in Section 5.

2 Spectral Unmixing

This paper deals with supervised spectral unmixing, which assumes that the spectra of the pure substances involved are known [2]. Common spectral unmixing methods are model-based, with the LMM, representing a good approximation in many cases, being the most commonly used [2, 11–14]. There also exist non-linear mixing models [3], which are not considered in this paper. The objective, the estimation of abundances $\hat{\mathbf{a}} \in \mathbb{R}^P$, is achieved using the LMM by

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{M}\,\mathbf{a}\|_2^2 \,. \tag{1}$$

Here $\mathbf{y} \in \mathbb{R}^\Lambda$ is a measured spectrum, i.e. 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 pure materials, and $\mathbf{a} = [a_1, ..., a_P]^\mathrm{T} \in \mathbb{R}^P$ are the corresponding abundances. The optimisation can be done by calculating the pseudo-inverse. However, constraints must be fulfilled for the abundances in order to remain physically plausible. Those constraints are the non-negativity constraint (2) and the sum-to-one constraint (3).

$$a_p \geq 0 \quad \forall p \tag{2}$$

$$\sum_{p=1}^{P} a_p = 1 \tag{3}$$

The consideration of these constraints counteracts model errors caused by the assumption of a linear mixing behaviour. A well established approach that optimises the LMM considering (2) and (3) is the Fully Constrained Least Squares (FCLS) algorithm [15]. Instead of (1), the Lagrangian $L : \mathbb{R}^{P+1} \to \mathbb{R}$ with the Lagrange multiplier $l \in \mathbb{R}$ is optimized:

$$L(\mathbf{a}, l) = \|\mathbf{y} - \mathbf{M}\,\mathbf{a}\|_2^2 - l \left(\sum_{p=1}^{P} a_p - 1 \right) . \tag{4}$$

The second part of (4) forces constraint (3). Additionally, negative $\hat{a}_p$ and the corresponding spectra in $\mathbf{M}$ are removed in an iterative procedure to ensure (2) as well.

Until now, the assumption has been made that the pure substances involved can be represented by a single spectrum. However, there is so-called spectral variability. It is caused, among other things, by changing surface conditions and the resulting variation in the angle of illumination [5]. Extended mixing models are available that take spectral variability into account by using additional parameters, such as the extended linear mixing model (ELMM) [16] or the generalized linear mixing model [17]. The ELMM uses the diagonal matrix $\mathbf{B} \in \mathbb{R}^{P \times P}$ to extend the LMM optimization problem to

$$\hat{\mathbf{a}} = \arg \min_{\mathbf{a}, \mathbf{B}} \|\mathbf{y} - \mathbf{M}\,\mathbf{B}\,\mathbf{a}\|_2^2 \,. \tag{5}$$

After presenting the basics, the next section describes the approach used to augment training datasets. The aim is to improve the performance of spectral unmixing for data-based methods.

3 Proposed Approach

The prerequisites for this approach are a set of available spectra for different abundance vectors $\mathbf{a}$. This is quite reasonable in an industrial environment, e.g. in a calibration dataset. The measured spectra are available as vectors $\mathbf{y}_\mathbf{a} \in \mathbb{R}^{\Lambda}$ in which each entry corresponds to the reflectance of light for a specific wavelength. For each abundance vector $\mathbf{a}$ there are different measured spectra, which differ due to spectral variability. These are now to be modelled as one Gaussian process $Y(\lambda|\mathbf{a})$ with the wavelength index $\lambda \in \mathbb{N}$ as the index and parametrised with the abundance vector $\mathbf{a}$.

Gaussian processes are completely defined by a mean function and an (auto-)covariance function [18, p. 13]. In this case, the mean value function is

$$m_Y(\lambda|\mathbf{a}) \tag{6}$$

and the covariance function with the second wavelength index $\lambda^* \in \mathbb{N}$

$$k_Y(\lambda, \lambda^*|\mathbf{a}) \,. \tag{7}$$

3.1 Data Preparation

In order to be able to represent this model with neural networks, the data are prepared. First, for all abundance vectors $\mathbf{a}$, the mean vector (8) and auto-covariance matrix (9) are calculated.

$$\mathbf{m}_\mathbf{a} = \frac{1}{N_\mathbf{a}} \sum_{n=1}^{N_\mathbf{a}} \mathbf{y}_{\mathbf{a}n} \tag{8}$$

$$\mathbf{K}_{\mathbf{a}} = \frac{1}{N_{\mathbf{a}} - 1} \sum_{n=1}^{N_{\mathbf{a}}} (\mathbf{y}_{\mathbf{a}n} - \mathbf{m}_{\mathbf{a}}) (\mathbf{y}_{\mathbf{a}n} - \mathbf{m}_{\mathbf{a}})^{\mathrm{T}} \tag{9}$$

Here $N_{\mathbf{a}} \in \mathbb{N}$ denotes the number of measured spectra for a given abundance vector $\mathbf{a}$. The elements of $\mathbf{m}_{\mathbf{a}}$ and $\mathbf{K}_{\mathbf{a}}$ for all available $\mathbf{a}$ can now be used as training data for the neural networks $\mathcal{N}_m$ and $\mathcal{N}_k$ that are supposed to learn (6) and (7).

The neural network $\mathcal{N}_m$ has the abundance vector $\mathbf{a}$ and the wavelength index λ as input variables and as the output variable the corresponding value of $\mathbf{m}_{\mathbf{a}}$. The neural network $\mathcal{N}_k$ has the abundance vector $\mathbf{a}$ and the transformed indices $\lambda_1' = \lambda + \lambda^*$ and $\lambda_2' = \max(\lambda) - |\lambda - \lambda^*|$ as input variables and as the output variable the corresponding value of $\mathbf{K}_{\mathbf{a}}$. The indices λ_1' and λ_2' are used because of two properties that the covariance matrices have. Firstly, there are higher values on the main diagonal, and secondly, they are symmetrical. This results in the neural network being kept quite simple later on, as it has to learn fewer changes in monotonicity (see Fig. 1).

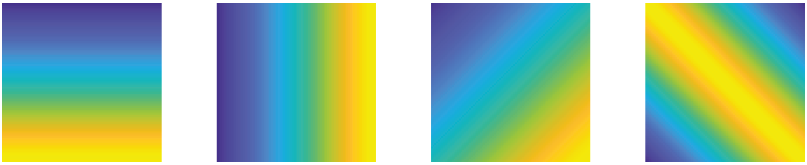

Fig. 1. Illustration of the values of the indices in the auto-covariance matrix (from left to right): λ, λ^*, λ_1' and λ_2'. Dark blue denotes a low value, yellow a high value

3.2 Neural Networks for Data Augmentation

The neural networks can now be trained with the prepared training data as described above. The neural networks each have $P+1$ ($\mathbf{a}$ and λ) or $P+2$ ($\mathbf{a}$, λ_1', and λ_2') inputs and only one output. To learn the desired relation a quite simple neural network is sufficient.

The networks consist of fully connected layers, i.e. layers in which all neurons of one layer are connected to all neurons of the neighbouring layers. The rectified linear unit (ReLU) $f_{\text{relu}}(z) = \max(0, z)$ is used as the activation function in all layers, with the exception of the last layer, where the logistic function

$$f_{\log}(z) = \frac{1}{1 + \mathrm{e}^{-z}} \tag{10}$$

is used. Batch normalisation is carried out prior to the rectified linear units [19]. The networks $\mathcal{N}_m$ and $\mathcal{N}_k$ have the same structure, which is shown in Fig. 2. The logistic loss function is used as objective function:

$$-\frac{1}{B} \sum_{b=1}^{B} o_b \cdot \log(\hat{o}_b) + (1 - o_b) \cdot \log(1 - \hat{o}_b)\,. \tag{11}$$

It is evaluated for each output value $\hat{o}_b \in (0, 1)$ and corresponding label $o_b \in (0, 1)$ of a training batch of size $B \in \mathbb{N}$. The logistic loss function is often used for a two-class

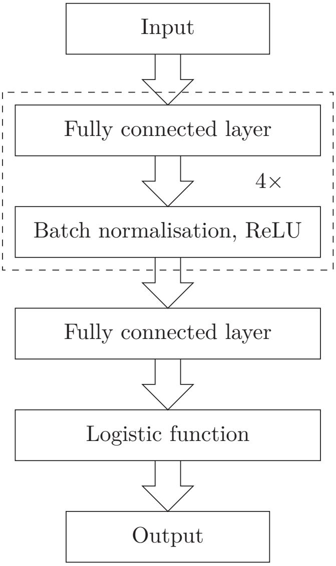

Fig. 2. Schematic representation of the neural network: There are four blocks consisting of a fully connected layer, batch normalisation and a ReLU activation function. This is followed by a fully connected layer with the logistic activation function

classification problem (cross-entropy loss). However, it also works with continuous labels and is suitable here because the values of the spectra range between 0 and 1.

Using the trained neural networks $\mathcal{N}_m$ and $\mathcal{N}_k$, an augmentation of the original training dataset can now be performed.

3.3 Augmentation Strategy

For the augmentation, additional mean value vectors and covariance matrices can now be generated by specifying abundance vectors for $\mathcal{N}_m$ and $\mathcal{N}_k$ that do not occur in the original training dataset. This allows pseudo-random generators to be used to produce spectra that complement the original training dataset. The spectra generated in this way also show spectral variability.

In order to augment the datasets at a lower effort a second strategy is used, where the original training datasets are only augmented by the mean value spectra. The neural network, which is later used for spectral unmixing (see Section 4), then has to learn the spectral variability on basis of the already existing training data.

The spectral unmixing performance of the augmented datasets is compared with that of the non-augmented datasets.

4 Experimental Results

Preceding the evaluation, the parameters used for $\mathcal{N}_m$ and $\mathcal{N}_k$ and the evaluation datasets are presented. The number of neurons was determined to be 32 for all layers and in both cases ($\mathcal{N}_m$ and $\mathcal{N}_k$). The neural networks were trained with the Adam optimizer [20]. The parameters from [20] were used, except for the learning rate, which was set to 0.01. The number of epochs was set to 2000 (both networks) for the datasets containing mixtures of quartz sand (see below) and to 3000 ($\mathcal{N}_m$) and 4000 ($\mathcal{N}_k$) for the colour powder dataset.

4.1 Datasets

Three datasets are used, which were recorded in our image processing laboratory. This ensures that we know the abundances as accurately as possible. All datasets consist of fine powders. These were mixed according to the specified abundances until the mixtures were homogeneous. A white balance with a reflectance standard was carried out after the recordings of the hyperspectral images, which compensates both spatial and spectral inhomogeneities of illumination and measurement setup. All datasets were acquired in 91 wavelength channels, ranging from 450 nm to 810 nm. For each mixture, 400 samples were acquired.

Two of the datasets contain mixtures of coloured quartz sand. The first of them (quartz-3) contains 45 mixtures of at maximum 3 components varying in abundance steps of 0.125 . The other one (quartz-4) includes 56 mixtures of at most 4 components, varying in abundance steps of 0.2 . The quartz sand datasets have a lower spectral variability and the non-linearity in the mixing behaviour is less significant compared to the following dataset. The third dataset consists of 56 mixtures of colour powders (colour-4), which also have up to 4 components. Again, the components are varied in abundance steps of 0.2 . The colour-4 dataset shows a high non-linearity between mixed spectra and the spectra of the pure substances and a high spectral variability. Hence, its spectra are more difficult to unmix.

All three datasets are divided into a test and a training dataset according to the abundances. For the datasets with four components, the samples with no abundance of value 0.2 or 0.8 are included in the training dataset. All other samples are included in the test dataset. This yields 16 abundance vectors in the training and 40 in the test dataset. The quartz-3 test dataset includes those where at least one abundance has the value 0.125, 0.375, 0.625 or 0.875. In consequence, there are 30 abundance vectors in the test dataset and 15 in the training dataset.

4.2 Evaluation of Generated Data

Using the neural networks from Section 3, new data are generated. The abundance vectors used as inputs are exactly the same as those in the test dataset. For each given abundance vector 400 spectra are generated, which correspond to the number of spectra per abundance vector in the test dataset.

As a measure of performance, the average minimum norm Δ_{AMN} is used between I measured spectra $\mathbf{y}_i$ and H generated spectra $\hat{\mathbf{y}}_h$ corresponding to an abundance vector:

$$\Delta_{\mathrm{AMN}} = \frac{1}{I} \sum_{i=1}^{I} \min_h \|\mathbf{y}_i, \hat{\mathbf{y}}_h\|_2 \,. \tag{12}$$

This performance measure was chosen because it tests whether a spectrum was generated as similar as possible to each spectrum in the test dataset. The calculation is done separately for each abundance vector and the corresponding spectra. The mean value of all Δ_{AMN} over all abundance vectors in the test dataset is called global average minimum norm Δ_{GAMN} (see Table 1). The results of both proposed augmentation strategies are compared with the performance of the generative convolutional neural network (Gen. CNN) with and without covariance matrix regularisation (-CovR) we presented in [8].

Table 1. Comparison of Δ_{GAMN} for all test datasets. In the first two columns the results from [8] are listed for comparison. The third column presents the values for the proposed method and the last column uses only the generated mean vectors $\hat{\mathbf{m}}_{\mathbf{a}}$ as generated spectra

Δ_{GAMN}	Gen. CNN [8]	Gen. CNN-CovR [8]	Proposed (normal)	Proposed (mean only)
quartz-3	0.1219	0.0812	0.0993	0.1371
quartz-4	0.1113	0.0787	0.0903	0.1298
colour-4	0.1242	0.0967	0.1016	0.1470

Table 1 shows that the inclusion of the covariance matrices results in lower Δ_{GAMN} values for all datasets compared to only using the mean vectors. This is because spectral variability is taken into account. The results of the proposed method are better than those of the unmodified generative CNN. However, the best results were achieved with the generative CNN with covariance matrix regularisation. It is also noticeable that within a method, the order of the datasets regarding Δ_{GAMN} always remains the same, which is due to the difficulty of the datasets.

In the next subsection, it will be investigated whether these results are consistent with those of spectral unmixing using augmented training datasets.

4.3 Spectral Unmixing Performance

For evaluation of the spectral unmixing performance, we use the same CNN as in [8], of which we have already presented the three-dimensional version in [4]. The CNN is trained with the original training dataset as well as with different augmented training datasets. The performance with respect to the test dataset is compared below. The CNN for spectral unmixing consists of three convolutional layers with a convolutional kernel length of 3. Then two fully connected layers follow. The numbers of feature maps from the input layer to the output layer are 1, 16, 32, 64, 64 and 1. We use the root-mean-square error over all N samples of a test dataset

$$\Delta_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{P}\sum_{p=1}^{P}(\hat{a}_{pn} - a_{pn})^2} \tag{13}$$

as a performance measure. For the methods that are not based on neural networks (see Section 2), the results are shown in Table 2 for the sake of clarity. For the remaining methods Δ_{RMSE} is displayed in Figure 3.

To obtain the results below, the network was trained with different numbers of epochs depending on the dataset and method. The quartz-3 dataset was trained for 251, the quartz-4 dataset for 41 and the colour-4 dataset for 31 epochs for the proposed method. When only mean vectors are used for augmentation, the number of epochs reduces to 31 (quartz-4) and 21 (colour-4). As a reference, we use the non-augmented training dataset, that was trained for 81 (quartz-3), 21 (quartz-4) and 21 (colour-4) epochs. The different numbers of epochs are chosen to avoid overfitting.

The original training datasets were augmented with a different number of spectra. Figure 3 shows the step size s $\in [0, 1]$ in which the additional abundance vectors were

Table 2. Comparison of Δ_{RMSE} for all test datasets for FCLS and ELMM based spectral unmixing

Δ_{RMSE}	quartz-3	quartz-4	colour-4
FCLS	0.1608	0.1115	0.2987
ELMM	0.1555	0.1056	0.2990

varied to generate the new data. All possible abundance vectors corresponding to the step size s are used in spectra generation, except for those already contained in the original training dataset.

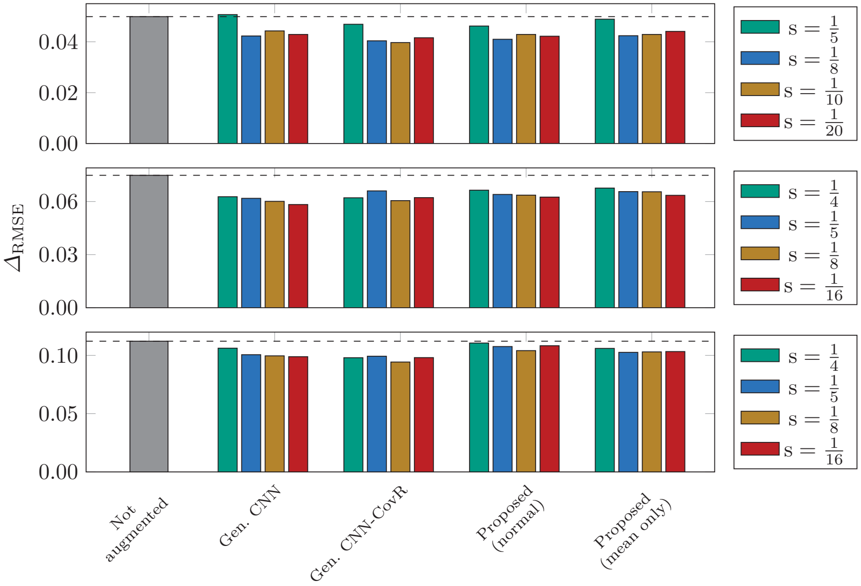

Fig. 3. Comparison of Δ_{RMSE} for the test datasets of the quartz-3, quartz-4, and colour-4 datasets (top to bottom) for CNN-based spectral unmixing using different augmentation strategies. The dashed lines are used for a better visual comparability with the non-augmented case

It is shown that the data-based spectral unmixing methods (Figure 3) perform better than the model-based methods (Table 2). Gaussian process based augmentation leads to an improvement compared to the non-augmented training dataset for all datasets and all step sizes s. The size of s does not have a major influence, unless it is chosen too large, in which case the performance deteriorates. If only the mean spectra are used for augmentation, the results are comparable. This is probably due to the fact that the spectral variability does not depend too much on the abundances and is already well represented by the spectra available in the original training dataset. For the colour-4 dataset, the result is worsened by adding the information from the covariance matrices. In this case, the assumption of a Gaussian process is likely to be an oversimplification.

The results from [8] cannot be reached with this approach. However, the training time of the neural networks for augmentation is approximately 9 times[1] shorter. On the one hand, this is due to the lower dimensional data points and therefore simpler neural networks. On the other hand, the size of the training dataset for augmentation is reduced if only the described moments are used as training data. The latter is especially true if only the mean spectra are used. In this case it is only one spectrum per abundance vector instead of $N_{\mathbf{a}}$. This leads to an approximately 120 times[1] shorter training time compared to [8].

5 Conclusion

In this work, an approach to augment training datasets for spectral unmixing was presented. For this purpose, inspired by Gaussian processes, a mean and a covariance function are learned by two neural networks. These networks are then used to generate additional training data.

It was shown that the performance of spectral unmixing with a CNN can be improved by the additional training data generated by these neural networks. It depends on the dataset how significant the improvement is. The improvement is slightly lower as with an existing method that uses a generative CNN for augmentation. However, the training time is an order of magnitude shorter. If only the neural network for the mean value function is used, where a similar increase in performance was observed depending on the dataset, the training time decreases by another order of magnitude.

In the future, something in between the two approaches presented would also be feasible. There, only the more relevant parts of the covariance functions would be used.

References

1. Gowen, A., O'Donnell, C., Cullen, P., Downey, G., Frias, J.: Hyperspectral imaging – an emerging process analytical tool for food quality and safety control. Trends in Food Science & Technology **18**(12) (2007) 590–598
2. Keshava, N., Mustard, J.F.: Spectral unmixing. IEEE signal processing magazine **19**(1) (2002) 44–57
3. Dobigeon, N., Altmann, Y., Brun, N., Moussaoui, S.: Linear and nonlinear unmixing in hyperspectral imaging. In Ruckebusch, C., ed.: Data Handling in Science and Technology. Volume 30. Elsevier (2016) 185–224
4. Anastasiadis, J., Puente León, F.: Spatially resolved spectral unmixing using convolutional neural networks (German paper). tm – Technisches Messen **86**(s1) (2019) 122–126
5. Borsoi, R.A., Imbiriba, T., Bermudez, J.C.M., Richard, C., Chanussot, J., Drumetz, L., Tourneret, J.Y., Zare, A., Jutten, C.: Spectral variability in hyperspectral data unmixing: A comprehensive review. arXiv preprint arXiv:2001.07307 (2020)
6. Simard, P.Y., Steinkraus, D., Platt, J.C., et al.: Best practices for convolutional neural networks applied to visual document analysis. In: Icdar. Volume 3. (2003)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
8. Anastasiadis, J., Heizmann, M.: CNN-based augmentation strategy for spectral unmixing datasets considering spectral variability. In Bruzzone, L., ed.: SPIE Remote Sensing – Image and Signal Processing for Remote Sensing XXVI. Volume 11533 of Proceedings of SPIE., SPIE (2020) 188–199

[1] Training performed on NVIDIA Quadro P5000.

9. Anastasiadis, J., Heizmann, M.: Generation of artificial training data for spectral unmixing by modelling spectral variability using gaussian random variables. In: OCM 2021 – Optical Characterization of Materials : Conference Proceedings. Ed.: Beyerer, J., Längle, T., Karlsruher Institut für Technologie (KIT) (2021) 129–139
10. Altmann, Y., Dobigeon, N., McLaughlin, S., Tourneret, J.: Nonlinear spectral unmixing of hyperspectral images using Gaussian processes. IEEE Transactions on Signal Processing **61**(10) (2013) 2442–2453
11. Bauer, S., Stefan, J., Puente León, F.: Hyperspectral image unmixing involving spatial information by extending the alternating least-squares algorithm. tm – Technisches Messen **82**(4) (2015) 174–186
12. Krippner, W., Bauer, S., Puente León, F.: Optical determination of material abundances in mixtures (German paper). tm – Technisches Messen **84**(3) (2017) 207–215
13. Krippner, W., Bauer, S., Puente León, F.: Considering spectral variability for optical material abundance estimation. tm – Technisches Messen **85**(3) (2018) 149–158
14. Krippner, W., Puente León, F.: Band selection and estimation of material abundances using spectral filters (German paper). tm – Technisches Messen **85**(6) (2018) 454–467
15. Heinz, D., Chang, C.I., Althouse, M.L.: Fully constrained least-squares based linear unmixing. In: IEEE 1999 International Geoscience and Remote Sensing Symposium. Volume 2., IEEE (1999) 1401–1403
16. Veganzones, M.A., Drumetz, L., Tochon, G., Dalla Mura, M., Plaza, A., Bioucas-Dias, J., 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), IEEE (2014) 1–4
17. Imbiriba, T., Borsoi, R.A., Bermudez, J.C.M.: Generalized linear mixing model accounting for endmember variability. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2018) 1862–1866
18. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. [u.a.] (2006)
19. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR **abs/1502.03167** (2015)
20. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)