Johannes Anastasiadis* and Michael Heizmann

GAN-regularized augmentation strategy for spectral datasets

GAN-regularisierte Augmentierungsstrategie für spektrale Datensätze

https://doi.org/10.1515/teme-2021-0109
Received October 13, 2021; accepted January 26, 2022

Abstract: Artificial neural networks are used in various fields including spectral unmixing, which is used to determine the proportions of substances involved in a mixture, and achieve promising results. This is especially true if there is a non-linear relationship between the spectra of mixtures and the spectra of the substances involved (pure spectra). To achieve sufficient results, neural networks need lots of representative training data. We present a method that extends existing training data for spectral unmixing consisting of spectra of mixtures by learning the mixing characteristic using an artificial neural network. Spectral variability is considered by random inputs. The network structure used is a generative adversarial net that takes the dependence on the abundances of pure substances into account by an additional term in its objective function, which is minimized during training. After training further data for abundance vectors for which there is no real measurement data in the original training dataset can be generated. A neural network trained with the augmented training dataset shows better performance in spectral unmixing compared to being trained with the original dataset. The presented network structure improves already existing results obtained with a generative convolutional neural network, which is superior to model-based approaches.

Keywords: Spectral unmixing, data augmentation, generative adversarial nets, spectral variability.

Zusammenfassung: Künstliche neuronale Netze werden vielseitig eingesetzt. Auch bei der spektralen Entmischung zur Ermittlung der Anteile der beteiligten Stoffe in einem Stoffgemisch liefern sie vielversprechende Ergebnisse. Das gilt vor allem dann, wenn ein nichtlinearer Zusammenhang zwischen den Spektren der Stoffgemische und den Spektren der beteiligten Stoffe (Reinspektren) besteht. Dazu benötigen sie viele und repräsentative Trainingsdaten. Ein Verfahren wird vorgestellt, welches vorhandene Trainingsdaten für die spektrale Entmischung bestehend aus Spektren von Stoffgemischen erweitert, indem der Mischzusammenhang mit einem künstlichen neuronalen Netz gelernt wird. Dabei wird auch die Spektrenvariabilität durch zufällige Eingangsgrößen berücksichtigt. Als Netzstruktur wird ein *Generative Adversarial Net* verwendet, welches durch einen zusätzlichen Term in seiner Kostenfunktion, die beim Training minimiert wird, die Abhängigkeit von den Anteilen der Reinstoffe berücksichtigt. Nach dem Training können damit weitere Daten für Mischverhältnisse erzeugt werden, für die es keine echten Messdaten im ursprünglichen Trainingsdatensatz gibt. Ein neuronales Netz, das mit dem erweiterten Trainingsdatensatz trainiert wird, zeigt bessere Ergebnisse bei der spektralen Entmischung als wenn es mit dem usprünglichen Datensatz trainiert wurde. Die vorgestellte Netzstruktur verbessert bereits vorhandene Ergebnisse, die mit einem generativen Faltungsnetz, das modellbasierte Ansätze übertrifft, erzielt wurden.

Schlagwörter: Spektrale Entmischung, Augmentierung, *Generative Adversarial Nets*, Spektrenvariabilität.

1 Introduction

Hyperspectral Images (HSIs) were previously used primarily in remote sensing, but are gaining popularity in automatic visual inspection. There are a lot of applications such as in food industry or the processing of bulks [13]. An important task is to monitor the correct composition of substances, preferably without contact. Normal colour images with three colour channels are usually not sufficient to solve this task satisfactorily. In contrast, hyperspectral images are sampled in many narrowband wavelength channels. With fine materials or large distances between camera and object, several materials are usually contained in one pixel, which means that only a mixed spectrum is available. Spectral unmixing (SU) is used to examine the material composition of such pixels [20].

***Corresponding author: Johannes Anastasiadis,** Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstr. 16, 76187 Karlsruhe, Germany, e-mail: anastasiadis@kit.edu
Michael Heizmann, Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstr. 16, 76187 Karlsruhe, Germany

To solve SU problems, mixing models are typically used. The mixing models are approximations that show a certain model inaccuracy. The accuracy of the models depends on the application [10]. Data based methods can also be used for SU [32]. Especially artificial neural networks have performed very well in many domains, including SU [3, 28, 33], in recent years. The advantages of neural networks that apply especially to SU are as follows: Firstly, spectral variability is taken into account when it is included in the training data. Secondly, constraints on the estimated proportions can be enforced by network architecture [2, 4]. However, neural networks only succeed without a mixing model because they learn the relationships from data. In order to achieve good results, sufficient labelled and representative training data must therefore be available. Otherwise there is a risk of overfitting, where only the training dataset is memorised but no generalisation happens. Unfortunately, large datasets are often not available. Especially in the industrial environment, where a certain task has to be achieved, there is a lack of data, which must first be effortfully acquired.

One way to address this problem is to use data augmentation. This is mainly used for classification problems, where data with the same label are slightly altered to achieve desired invariances [25, 29]. For SU, it is desirable to estimate the proportions, also called abundances, of the pure substances involved as accurately as possible and to achieve invariance to the spectral variability. Therefore, it makes sense to generate additional training data during augmentation that feature spectral variability and correspond as closely as possible to the true mixing characteristics. We have already presented an approach in which augmentation is performed using a generative convolutional neural network (CNN) [1]. There, additional random input variables are used to generate data that feature spectral variability and correspond to abundances that are not contained in the original training dataset. Data generation involving spectral variability has already been performed in [6, 7] for pure spectra using variational autoencoders. In this paper, our approach is extended by a regularisation with a generative adversarial net (GAN) [11], which aims at a more detailed parametric-analytical modelling of the characteristic mixing behaviour and spectral variability. For unsupervised SU, regularisation with a GAN has already been implemented [17]. To the best of our knowledge, it has not been used yet for modelling of the mixing behaviour and augmentation of labelled training datasets. For this approach, a couple of spectra are needed for several abundance combinations. This requirement is fulfilled in an industrial environment, for example by preparing a calibration dataset.

In summary, the following innovations are presented in this paper:

- The modelling of mixed spectra including spectral variability using a GAN-based neural network.
- The use of a hybrid objective function to model the dependence on the abundances.
- The use of the discriminator during data generation to reject bad samples that can occur when unknown abundances are specified.
- The investigation of the effect of the augmentation using data generated in this way on the performance of spectral unmixing.

The rest of the paper is structured as follows. The next section briefly introduces the basics of SU. Then, in Section 3, the basic concept of GANs is presented, which will later serve as the network structure for the augmentation. Thereafter, Section 4 presents the previous work on which the approach presented in this paper is based. In Section 5, the procedure for regularisation with GANs is described. An evaluation and comparison of the methods presented is carried out in Section 6. Both the augmented data itself and its effect on SU are discussed. Finally, the paper is summarised in Section 7.

2 Spectral unmixing

This section briefly introduces the basics of spectral unmixing. Readers who are familiar with these may skip this section.

Spectral unmixing is about estimating the relative proportions (abundances) and spectra of the pure substances involved from a spectrum of a mixture of substances [20]. The term supervised SU is used if the spectra of the pure substances involved are known, otherwise it is called unsupervised SU. In this paper, a data-based approach is used in which the spectra of the pure substances are also included in the training datasets. Therefore, this paper only considers supervised SU from here on. Thus, the goal of supervised SU is to estimate the abundances $\mathbf{a} = [a_1,...,a_P]^{\mathrm{T}} \in \mathbb{R}^P$, given a recorded discrete spectrum $\mathbf{y} \in \mathbb{R}^\Lambda$ and the involved discrete pure spectra $\mathbf{M} = [\mathbf{m}_1,...,\mathbf{m}_P] \in \mathbb{R}^{\Lambda\times P}$. In general, the non-negativity constraint (1) and the sum-to-one constraint (2) must be fulfilled for the abundances to ensure physical plausibility.

$$a_p \geq 0 \quad p = 1,...,P \tag{1}$$

$$\sum_{p=1}^{P} a_p = 1 \tag{2}$$

A model-based approach, in which a mixing model describes the relationship between the variables, is often used to solve SU problems. These mixing models approximately describe the relationship between $\mathbf{y}$, $\mathbf{a}$, and $\mathbf{M}$. The most commonly used mixing model is the linear mixing model, which is a valid approximation for many applications [5, 20, 22–24]:

$$\mathbf{y} = \sum_{p=1}^{P} \mathbf{m}_p \, a_p = \mathbf{M}\,\mathbf{a}\,. \tag{3}$$

The linear mixing model can be solved for $\mathbf{a}$ under consideration of the constraints (1) and (2) using methods like the Fully Constrained Least Squares algorithm [15]. Furthermore, there are also non-linear mixing models, which can be superior to the linear mixing model depending on the application [10]. For example, there are the generalized bilinear model [14] and the linear quadratic model [26]. Especially when the substances involved have small grain sizes, the linear mixing model is not sufficient [20]. This is the case with the datasets used for evaluation in Section 6, where homogeneous mixtures of coloured quartz sands and colour powders are investigated. However, it is also much more difficult to determine parameter $\mathbf{a}$ using a non-linear mixing model.

The approaches presented so far assume that the pure substances involved can each be described by one spectrum. However, the pure substances have varying spectra [30]. This is called spectral variability and has various causes [8]. An important reason is the surface topography, which causes a variation in the angle of the light scattered by the substances under investigation. An example that illustrates spectral variability is shown in Fig. 1. There are also mixing models that take spectral variability into account. This is done through additional parameters. Based on the linear mixing model, the extended linear mixing model [31] and the generalized linear mixing model [18] are to be mentioned. While the former introduces a scaling factor per pure substance involved, the latter uses one for each element in $\mathbf{M}$.

In contrast to the methods presented in this section, there are also data-based methods that do not require any mixing model at all. Those are especially useful if the mixing behaviour is non-linear, because there the model based methods are hard to optimize. In this paper, the training dataset for such methods is augmented using GANs, which are introduced in the following section.

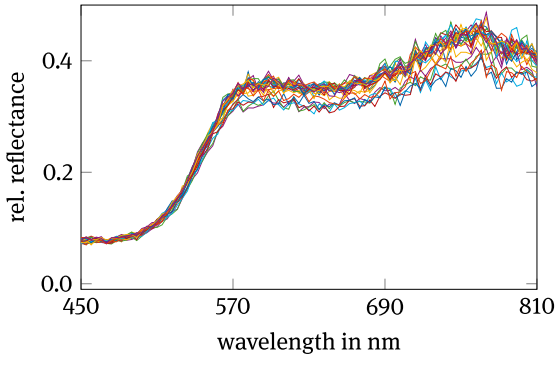

Figure 1: Spectral variability example: 25 spectra of a pure substance contained in the colour-4 dataset presented in Sec. 6. Each line corresponds to a spectrum. The colours serve to improve distinguishability.

3 Generative adversarial nets

For the approach presented in this paper, we use a GAN structure to extend our generative CNN (see Section 4 and [1]). This section briefly introduces the basics of GANs. Readers who are familiar with these may skip this section.

Generative adversarial nets [11] are neural networks that are trained in an unsupervised manner and learn the distribution of the training data, which enables them to produce further samples from the distribution. They consist of two subnetworks: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The input variables of the discriminator are the output variables of the generator, as well as the real training data $\mathbf{x}_{\text{tr}}$. The output of the discriminator is a scalar value and the inputs of the generator are random values $\mathbf{x}_{\text{rnd}}$. This setup is illustrated in Fig. 2. The aim of the generator is to produce data that are as similar as possible to the real training data. The discriminator tries to distinguish whether the respective input data originates from the generator or is real training data.

The parameters of the two subnetworks are updated alternately. When training with M samples, the objective

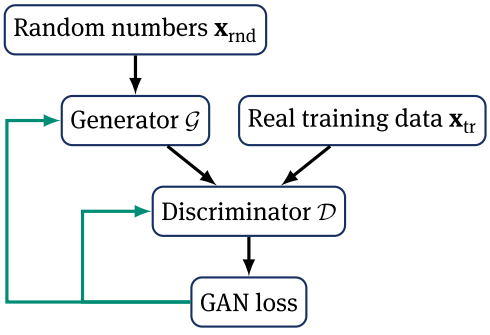

Figure 2: Illustration of a GAN setup.

functions are:

$$J_{\mathcal{D}} = -\sum_{m=1}^{M} \left[\log\left(\mathcal{D}(\mathbf{x}_{\mathrm{tr},m})\right) + \log\left(1 - \mathcal{D}(\mathcal{G}(\mathbf{x}_{\mathrm{rnd},m}))\right)\right], \quad (4)$$

$$J_{\mathcal{G}} = -\sum_{m=1}^{M} \log\left(\mathcal{D}(\mathcal{G}(\mathbf{x}_{\mathrm{rnd},m}))\right). \quad (5)$$

The objective function $J_{\mathcal{D}}$ is used for training the discriminator and $J_{\mathcal{G}}$ for training the generator. Consequently, the value 1 at the output of the discriminator represents the classification as real training data, while the value 0 corresponds to generated data. By training alternately, the generator and discriminator improve until, ideally, the generator's data are that accurate that the discriminator is no longer able to distinguish between real and generated data.

Now the basics of GANs have been provided, the following section presents the preceding work, which is later enhanced by a GAN structure.

4 Preceding work

This section summarises previous work on which the approach presented in this paper is based. Firstly, the generative CNN is presented, which is used for augmentation and is integrated into a GAN structure in this paper. Secondly, the CNN for SU is presented, which is used to evaluate the augmentation methods. Basics on artificial neural networks are required in this section and can for example be looked up in [12].

4.1 Generative CNN

The generative CNN is described in detail in [1]. The main idea of this approach is not to use the training data directly for training a CNN for SU. Instead, a generative CNN is trained first in order to augment the training dataset. The generative CNN uses the abundance vectors $\mathbf{a}$ as input data and the generated spectra $\hat{\mathbf{y}} \in \mathbb{R}^{\Lambda}$ as output data. Independent random values $\boldsymbol{A} \sim \mathcal{N}(\mathbf{0}; \mathbf{1})$ are added as additional input data. An illustration of the generative CNN can be found in Fig. 3.

Each input variable of the generative CNN corresponds to an 1×1 feature map. These input variables are first processed with three transposed convolutional layers. These are convolutional layers with preceded upsampling by zero padding. Note that in Fig. 3, for the sake of clarity, only one feature map per layer is shown. A final convolutional layer is used to merge the feature maps into a discrete spectrum $\hat{\mathbf{y}}$. The logistic loss function

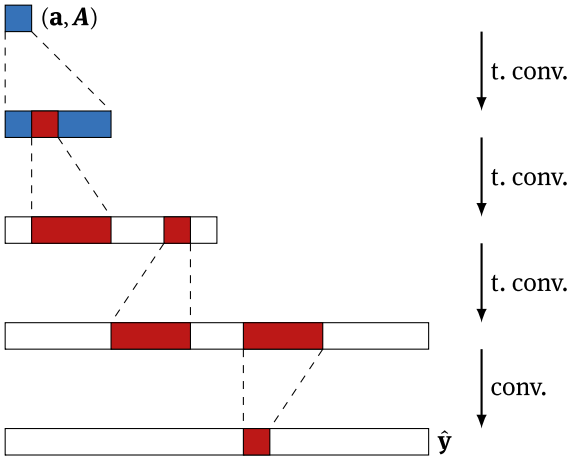

Figure 3: Layout of the generative CNN. It consists of transposed convolutional layers (t. conv.) and a convolutional layer (conv.). Only one feature map is shown in each layer.

$$J_{\log} = \sum_{\lambda=1}^{\Lambda} y_\lambda \cdot \log(\hat{y}_\lambda) + (1 - y_\lambda) \cdot \log(1 - \hat{y}_\lambda) \quad (6)$$

is used for training, which is suitable because the spectra consist of values between zero and one. Note that for sake of clarity, the objective function is only shown for one sample. In the case of several samples, the individual objective functions are added up like in Eq. (5). Adam is used as optimizer [21].

Due to spectral variability, there are many more different output values for the generative CNN than there are different abundance vectors (the output values of the original CNN). This allows the generative CNN to be trained for many epochs without overfitting. It can then be used to obtain spectra for additional abundance vectors in order to augment the original training dataset.

In order to be able to model the spectral variability more accurately, additional modifications were made [1]. One that performed well will be used for comparison with the method presented later. There, the covariance matrix is calculated for all spectra that share an abundance vector. During training, the mean squared error between the elements of these covariance matrices and those of the generated spectra was used for regularisation.

4.2 CNN for spectral unmixing

The CNN for SU was first introduced in [3] as a version with three-dimensional convolutional kernels. Since only

individual spectra are considered here, a version with one-dimensional convolutional kernels is used, which is described in detail in [1].

The input data of the CNN are discrete spectra and the output data are the estimated abundance vectors $\hat{\mathbf{a}} \in \mathbb{R}^P$. The layout of the CNN for SU can be found in Fig. 4.

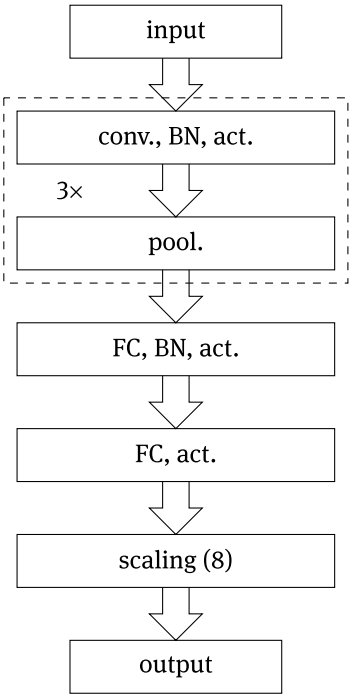

Figure 4: Layout of the CNN for SU. It consists of three one-dimensional convolutional (conv.) layers, a pooling layer (pool.) after each convolutional layer, two fully connected (FC) layers, batch-normalisation (BN) [19] before each activation function (except for the last), and the logistic function (7) as activation (act.) function.

The logistic activation function of the last layer enforces the non-negativity constraint (1):

$$f_{\log}(z) = \frac{1}{1+\mathrm{e}^{-z}}\,. \tag{7}$$

The sum-to-one constraint (2) is enforced by

$$\hat{a}_p = \frac{a_p^*}{\sum_{\tilde{p}=1}^{P} a_{\tilde{p}}^*} \quad \text{for} \quad p = 1, 2, \ldots, P\,, \tag{8}$$

where $\mathbf{a}^* = [a_1^*, ..., a_P^*]^{\mathrm{T}} \in \mathbb{R}^P$ denotes the output of the last activation function. This has lead to slightly better performance than using a softmax layer to enforce the constraints in prior experiments [3]. As objective function the mean squared error between $\mathbf{a}$ and $\hat{\mathbf{a}}$ is used. Adam is used as optimizer [21].

In the following section, the generative CNN (Fig. 3) is extended by integrating it into a GAN structure.

5 GAN-regularized generative CNN

For augmentation as described above, it is necessary to be able to specify abundance vectors for the neural network. However, with standard GANs there is no way to achieve this [11]. A conditional GAN can be used for this purpose [27]. The conditions, in this case the abundance vectors, are used as additional input variables for the generator as well as for the discriminator. The conditional GAN is intended for problems where all conditions occur during training that are used during data generation. This is the case, for example, when data samples for different classes are to be generated. Here, however, we intend to specify conditions during data sampling that lie between those of the training dataset. A variant of the conditional GAN exists for continuous input variables [9], but there was a lot of training data available and scalar conditions were used in the experiments. In our experiments it turned out that conditional GANs cannot achieve satisfactory results for SU dataset augmentation. This is due to the rather small training datasets, where the distances between included abundance vectors are large.

Instead, a hybrid approach is used in which the generative CNN, that already has the abundance vectors as input variables, is used as the generator of a GAN. In contrast to a standard GAN, the reconstruction error of the output variables of the generator is also taken into account in the objective function. In other words, the generative CNN is regularised by the GAN, for which reason the approach is referred to as rGAN below. The concept of the rGAN is illustrated in Fig. 5.

The generator loss ensures dependence on the abundances, while the GAN loss ensures that the generated spectra are as close as possible to the real ones.

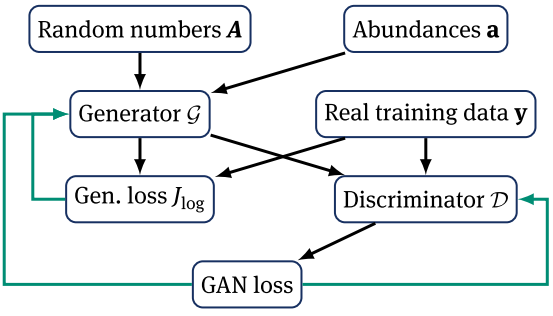

Figure 5: Illustration of the rGAN setup. In comparison to the standard GAN, the abundances enter the generator as input variables. In addition to the GAN loss, there is also a generator (gen.) loss that takes into account the error between generated and real spectra.

The generator we used is exactly the same as the generative CNN from Section 4. The discriminator consists of three fully connected layers that map the spectra to a scalar value. All layers use the logistic function (7) as activation function. For the discriminator training step, $J_{\mathcal{D}}$ according to Eq. (4) is used as objective function with $\mathbf{x}_{\text{tr}} = \mathbf{y}$ and $\mathbf{x}_{\text{rnd}} = \boldsymbol{A}$. For the generator training step, the objective function is

$$J_{\text{rGAN}} = J_{\mathcal{G}} + \mathrm{k} \cdot J_{\log}\,, \tag{9}$$

where $\mathrm{k} \in \mathbb{R}$ is used to weigh the parts differently. Adam is used as optimizer [21].

For augmentation, the generator of the trained rGAN can be used to generate further training data. This is done by giving the generator abundance vectors $\mathbf{a}$ that are not contained in the original training dataset. To ensure that the spectral variability is also taken into account in the generated data, several spectra are generated for each abundance vector by varying the random input variables $\boldsymbol{A}$.

Alternatively, the discriminator can be used in addition to the generator for data generation. In this case, only those of the generated spectra are used for augmentation that the discriminator classifies as real spectra. During its training, the discriminator does not see any data generated on the basis of the new abundance vectors. Therefore, it happens that for some abundance vectors all generated spectra are classified as generated. Because of this, after a fixed number of unsuccessful generations $\zeta \in \mathbb{N}$, the decision limit, which is originally 0.5, is lowered by $\epsilon \in \mathbb{R}$. This is repeated until the desired number of spectra is generated.

In the next section, the methods presented here are evaluated using real data. They will be compared with the methods from Section 4.

6 Experimental results

The datasets used for the evaluation were captured in our image processing laboratory. This makes it possible to have real images where we know the abundances precisely. In total, three datasets are used, two of which consist of differently coloured quartz sands and one of colour powders. To produce mixed substances, the pure substances were filled into small bins in different proportions by volume and mixed until homogeneous mixtures were obtained. The mixtures and pure substances were then filled into small boxes (see Fig. 6). Those were captured by a hyperspectral imaging system. The ANDOR iXon3 897 camera used was together with the acousto-optic tunable filter HSi-300 from Gooch & Housego. With this setup, two-dimensional images are taken one after the other for the different wavelength channels. After taking the hyperspectral images, a white balance using a reflectance standard was performed to compensate both spatial and spectral inhomogeneities of the illumination and the measurement setup. The hyperspectral imaging system is able to record wavelengths from 450 nm to 810 nm. This range has been divided into 91 channels with a bandwidth of approximately 4 nm. For each substance mixture 400 spectra were captured.

Figure 6: RGB images of some of the boxes filled with coloured quartz sands (left) and colour powders (right).

Of the two datasets which consist of coloured quartz sands, one contains 45 mixtures of up to 3 pure substances (quartz-3) and the other 56 mixtures of up to 4 pure substances (quartz-4). In the quartz-3 dataset the abundances of the substances vary in steps of $\frac{1}{8}$, in the quartz-4 dataset in steps of $\frac{1}{5}$. The dataset consisting of colour powders also contains 56 mixtures of up to 4 pure substances (colour-4) that vary in abundance steps of $\frac{1}{5}$. The colour-4 dataset has a higher spectral variability and shows a higher nonlinearity in the mixing behaviour than the datasets containing coloured quartz sands. This is due to the smaller grain size and the fact that the colour powders tend to clump together.

All datasets are divided into a test and a training dataset depending on the abundance vectors. This ensures that no (different) spectra with the same abundance vector are included in both datasets. The split is performed systematically such that the data from the test dataset lies (in terms of abundance vectors) "between" the data from the training dataset. It is done in this way because the aim is to augment an existing training dataset by adding data with abundance vectors between the existing ones. The details are described by the following rules:

– For the quartz-4 and colour-4 datasets, the samples that do not have at least one abundance of $\frac{1}{5}$ or $\frac{4}{5}$ are

included in the training dataset. The rest of the samples forms the test dataset. There are in total 16 abundance vectors in the training dataset and 40 in the test dataset.
– The quartz-3 test dataset includes the samples where at least one abundance has a value of $\frac{1}{8}$, $\frac{3}{8}$, $\frac{5}{8}$ or $\frac{7}{8}$. This means that there are 15 abundance vectors in the training dataset and 30 in the test dataset.

6.1 Evaluation of generated spectra

For the evaluation of the artificial spectra generated by our GAN, which are later used for augmentation of the real spectra, they are compared with the real spectra in the corresponding test dataset. The generated spectra can therefore only be evaluated for those abundance vectors that also occur in the test datasets. It is to be expected that by using the GAN-based regularisation the generated spectra are more similar to the real ones than when using the generative CNN alone, because spectral variability should be modelled more accurately. Using the discriminator also for sampling might further improve the result, because bad samples can be sorted out.

The structure of the generator was chosen as in Fig. 3, with the number of feature maps equal to $P + 3$, 32, 32, 16, and 1 from front to back. The additional 3 feature maps at the input correspond to the size of the random input vector $\boldsymbol{A}$. The size of the convolution kernels was set to 5 and convolutions are performed along the wavelength dimension. To get 91 wavelength channels at the output, the size of the convolution kernel in the first layer is 23. Together with the upsampling rate 2 this results in 92 wavelength channels (the last channel is ignored to get the desired 91 channels). For the discriminator, the number of neurons was chosen to be 64, 16, and 1.

During training of the rGAN, the generator was trained every seventh iteration, the discriminator every remaining iteration. During generator training, $k = 10$ (see Eq. (9)) was used. Some training parameters were chosen differently for the datasets. The quartz-3 dataset was trained for 6000 epochs with a learning rate of 0.01. For the quartz-4 and colour-4 datasets, pre-training was performed first. Only the generator was trained with $J_{\log}$ for 450 epochs with a learning rate of 0.01. After the pre-training, the entire rGAN was then trained with the combined objective function for 4000 epochs with a learning rate of 10^{-3}. The difference in the training procedure is due to the fact that the abundance vectors in the quartz-4 and colour-4 datasets have a greater step size than in the quartz-3 dataset. Therefore the latter is easier to train.

For the quantitative evaluation of the generated spectra, we introduce the average minimum (euclidean) norm,

$$\Delta_{\mathrm{AMN}} = \frac{1}{I}\sum_{i=1}^{I} \min_{h} \|\mathbf{y}_i, \hat{\mathbf{y}}_h\|_2 \,, \tag{10}$$

between a subset of I measured spectra and a subset of H generated spectra. By using the minimum operation, the average distance of the closest generated spectrum $\hat{\mathbf{y}}$ to each spectrum $\mathbf{y}$ in the subset of the test dataset is evaluated. The subsets contain all spectra corresponding to one abundance vector. If Δ_{AMN} is averaged over a whole dataset we call it global average minimum norm Δ_{GAMN}. To obtain the results of this subsection, we generated 400 spectra for each abundance vector to be compared with the corresponding 400 spectra in the test dataset.

Table 1 shows Δ_{GAMN} for all datasets, presented methods, and comparison methods to evaluate the synthetic mixed spectra. It can be seen that regularisation with a GAN leads to an improvement compared to the generative CNN alone. With covariance regularisation, the best value for Δ_{GAMN} is achieved. However, it must be noted that this method is only possible if many spectra per abundance vector are available, because the covariance matrices have to be calculated. Generating data using the rGAN with discriminator has only a minor impact here. This is because this method is used to sort out bad samples, but due to the minimum operator in Eq. (10) these spectra have no effect.

Within a method of generating spectra, quartz-4 always gives the best result and colour-4 always the worst. This is because colour-4 shows the most significant degree of non-linearity and the two quartz sand datasets have different abundances and are differently divided into a test and a training part. The quartz-3 dataset does not get any data with abundance step size of $\frac{1}{8}$ during training, but has to generate them for the test. The quartz-4 dataset gets data with abundance step size of $\frac{1}{5}$ for training and has only to generate spectra based on different abundance vectors for the test.

Table 1: Comparison of Δ_{GAMN} for all datasets. It is shown for the standard rGAN as presented above and for the rGAN where the discriminator is used for data generation, too (rGAN-DS). For comparison the values are also shown for the generative CNN (GCNN) described in Section 4 and the GCNN with covariance matrix regularisation (GCNN-C) [1].

Δ_{GAMN}	quartz-3	quartz-4	colour-4
GCNN [1]	0.1219	0.1113	0.1242
GCNN-C [1]	0.0812	0.0787	0.0967
rGAN	0.1087	0.1058	0.1188
rGAN-DS	0.1087	0.1051	0.1196

The next subsection investigates whether these results can also be transferred to the behaviour during SU using augmented training datasets.

6.2 Spectral unmixing results

In this section, the SU performance of the test datasets is compared after the CNN for SU from Section 4 has been trained with differently augmented training data. The test results for training with non-augmented training datasets are used as a baseline in each case (column *CNN* in Figs. 7–9). A comparison with methods that are not based on neural networks but on mixing models can be found in [1] for the datasets investigated here. The mixing model based methods did not reach the spectral unmixing performance of the CNN for SU trained without dataset augmentation.

It is to be expected that augmentation will improve the performance of spectral unmixing. GAN-based augmentation should yield better results than the generative CNN alone, because spectral variability should be modelled more accurately. Since bad samples can be sorted out, using the discriminator also for sampling might further improve the result. Because the real data used is the same in all cases and no new information is added, but only the existing data is better utilised, minor improvements are to be expected for all presented methods compared to directly using the CNN for SU on the real spectra.

The rGAN was configured in the same way as in the previous subsection to perform the augmentation tasks. Different numbers of abundance vectors were used for augmentation depending on the abundance step $s \in [0,1]$. This means that all possible abundance combinations at a step size of s were used for augmentation. If this scheme produces abundance vectors that already occur in the original training dataset, they are not used for augmentation. If the discriminator is also used for data sampling (rGAN-DS), then $\zeta = 100$ is used (see Section 5). The decision limit is decreased by $\epsilon = 0.01$.

The CNN for SU uses a convolution kernel size of 3 for all convolutional layers. The pooling layers are set to perform downsampling by factor 2 using max pooling. The numbers of feature maps from input to output are 1, 16, 32, 64, 64, and 1. The number of training epochs depends on how early overfitting occurs in a dataset. Therefore, the number of training epochs is different for the datasets. The CNN is trained using 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. As it can be seen, the augmentation also allows longer training times for datasets consisting of coloured quartz sands until overfitting becomes a problem. Adam is used as optimizer with a learning rate of 0.01 and the other parameters as suggested in [21].

To quantify SU performance the root-mean-square error

$$\Delta_{\mathrm{RMSE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{P}\sum_{p=1}^{P}(\hat{a}_{pn} - a_{pn})^2}\,, \tag{11}$$

over all N spectra of a test dataset is used. For comparison, the generative CNN (GCNN) is used, which corresponds to the generator of the rGAN. The variant that performs a regularisation with the covariance matrix is also added (GCNN-C, see Section 4).

In Figs. 7–9 the SU results are shown. Since the values of Δ_{RMSE} are sometimes very close to each other,

$$\Delta_{\mathrm{CNN}} = \Delta_{\mathrm{RMSE,CNN}} - \Delta_{\mathrm{RMSE}} \tag{12}$$

is used for better readability, which corresponds to the improvement of current Δ_{RMSE} compared to the baseline $\Delta_{\mathrm{RMSE,CNN}}$ (using the non-augmented training datasets).

For the quartz-3 dataset (Fig. 7) it can be seen that augmentation always improves SU performance compared to the baseline, unless s is chosen too large. GAN-based regularisation improves the results compared to the standard GCNN, most strongly when using the discriminator during data generation. Regularisation with the covariance matrix leads to slightly better results. However, many spectra per abundance vector have to be available for that method, because the covariance matrices have to be calculated.

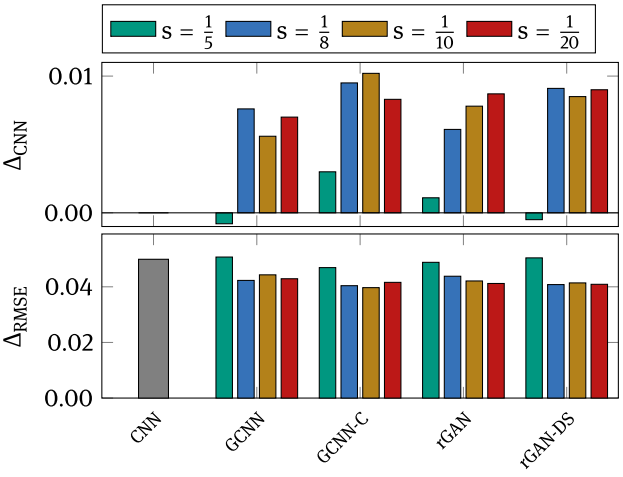

Figure 7: Spectral unmixing results of the quartz-3 dataset.

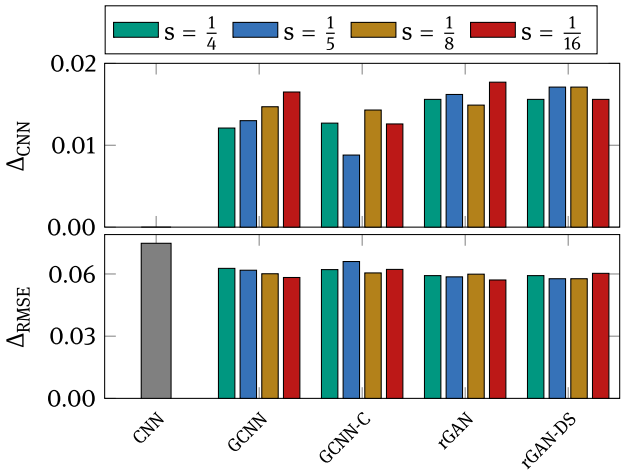

Figure 8: Spectral unmixing results of the quartz-4 dataset.

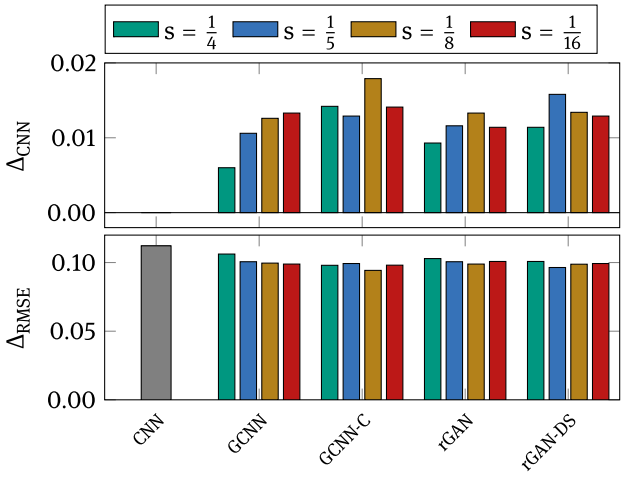

Figure 9: Spectral unmixing results of the colour-4 dataset.

For the datasets quartz-4 and colour-4, there is an improvement over the baseline for all methods presented and all s investigated. This is due to the fact that the distance between the abundance vectors in the original training dataset is greater and thus the baseline is worse than in the quartz-3 dataset. In the case of the quartz-4 dataset (Fig. 8), the regularisation with a GAN leads to an improvement compared to the existing methods (GCNN and GCNN-C). Again, using the discriminator for sampling is slightly better than using the generator alone.

For the colour-4 dataset (Fig. 9), an improvement compared to the GCNN can be achieved for some step sizes by the GAN-based regularisation presented here. Also for this dataset, the use of the discriminator while generating data is the superior alternative. As with the quartz-3 dataset, the GCNN with regularisation by the covariance matrices is slightly better. The impact of the step size s tends to be that better results can be obtained for smaller s. However, there are many irregularities in this observation. This could be because the spectra of different abundance vectors overlap due to spectral variability. If these overlaps in a dataset are disadvantageous, this leads to worse performance in SU. For the blue bars ($s = \frac{1}{8}$ and $s = \frac{1}{5}$, resp.), the abundance vectors of the spectra artificially generated during augmentation correspond exactly to the abundance vectors in the test data. However, this does not lead to a visible advantage of these step sizes. This confirms that overfitting is not a problem during training of the CNN for SU.

For all three datasets an improvement in the spectral mixing performance compared to the GCNN can be observed with the GAN-based regularisation. If the discriminator is also used while generating spectra, this improvement is even higher. For two datasets, the regularisation with the covariance matrices was slightly better. However, it must be noted that this procedure includes the calculation of the covariance matrices, which requires many spectra per abundance vector.

7 Summary

A GAN-based approach was presented to model the mixing behaviour of hyperspectral data. This approach is an extension of the GCNN we have presented earlier. By using the GAN structure, the distribution of the data including spectral variability is learned more accurately than with GCNN. It also resulted in better SU performance when it was used to augment the training datasets. This was further improved if the discriminator was used to generate data, allowing bad generated spectra to be rejected. This shows that the GAN's ability to learn the distribution of the data is advantageous for the application of modelling spectral data including spectral variability. For some of the tested datasets, regularisation with covariance matrices yields a slightly better result. However, the latter is only possible if there are enough spectra per abundance vector to be able to determine the covariance matrices.

In future work, this approach could also be used for other regression problems where a variance in the output variables is supposed to be modelled. Depending on the nature of the data, the hyperparameters would have to be adjusted. However, the presented approach would remain the same. In addition, other generative neural networks could be used to model mixing behaviour including spectral variability, such as denoising diffusion probabilistic models [16].

References

1. J. Anastasiadis and M. Heizmann. CNN-based augmentation strategy for spectral unmixing datasets considering spectral variability. In L. Bruzzone, editor, *SPIE Remote Sensing – Image and Signal Processing for Remote Sensing XXVI*, volume 11533 of *Proceedings of SPIE*, pages 188–199. SPIE, 2020. 10.1117/12.2575875.
2. J. Anastasiadis and M. Heizmann. Generation of artificial training data for spectral unmixing by modelling spectral variability using gaussian random variables. In J. Beyerer and T. Längle, editors, *OCM 2021 – Optical Characterization of Materials: Conference Proceedings*, pages 129–139. Karlsruher Institut für Technologie (KIT), 2021.
3. J. Anastasiadis and F. Puente León. Spatially resolved spectral unmixing using convolutional neural networks (German paper). *tm – Technisches Messen*, 86(s1):122–126, 2019.
4. J. Anastasiadis, P. Benzing, and F. Puente León. Generation of artificial data sets to train convolutional neural networks for spectral unmixing (German paper). *tm – Technisches Messen*, 87(9):542–552, 2020.
5. S. Bauer, J. Stefan, and F. Puente León. Hyperspectral image unmixing involving spatial information by extending the alternating least-squares algorithm. *tm – Technisches Messen*, 82(4):174–186, 2015.
6. R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez. Deep generative endmember modeling: An application to unsupervised spectral unmixing. *IEEE Transactions on Computational Imaging*, 6:374–384, 2020.
7. R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, and C. Richard. Deep generative models for library augmentation in multiple endmember spectral mixture analysis. *IEEE Geoscience and Remote Sensing Letters*, 2020.
8. R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, C. Richard, J. Chanussot, L. Drumetz, J.-Y. Tourneret, A. Zare, and C. Jutten. Spectral variability in hyperspectral data unmixing: A comprehensive review. *arXiv preprint arXiv:2001.07307*, 2020.
9. X. Ding, Y. Wang, Z. Xu, W. J. Welch, and Z. J. Wang. Ccgan: Continuous conditional generative adversarial networks for image generation. *arXiv preprint arXiv:2011.07466*, 2020.
10. N. Dobigeon, Y. Altmann, N. Brun, and S. Moussaoui. Linear and nonlinear unmixing in hyperspectral imaging. In C. Ruckebusch, editor, *Data Handling in Science and Technology*, volume 30, pages 185–224. Elsevier, 2016.
11. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 27, 2014.
12. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.
13. A. Gowen, C. O'Donnell, P. Cullen, G. Downey, and J. Frias. Hyperspectral imaging – an emerging process analytical tool for food quality and safety control. *Trends in Food Science & Technology*, 18(12):590–598, 2007.
14. A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret. Nonlinear unmixing of hyperspectral images using a generalized bilinear model. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4153–4162, 2011.
15. D. Heinz, C.-I. Chang, and M. L. Althouse. Fully constrained least-squares based linear unmixing. In *IEEE 1999 International Geoscience and Remote Sensing Symposium*, volume 2, pages 1401–1403. IEEE, 1999.
16. J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.
17. W. J. Holland and Q. Du. Adversarially regularized autoencoder for hyperspectral image unmixing. In *Image and Signal Processing for Remote Sensing XXVI*, volume 11533, page 115330U. International Society for Optics and Photonics, 2020.
18. T. Imbiriba, R. A. Borsoi, and J. C. M. Bermudez. Generalized linear mixing model accounting for endmember variability. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1862–1866. IEEE, 2018.
19. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.
20. N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE signal processing magazine*, 19(1):44–57, 2002.
21. D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
22. W. Krippner and F. Puente León. Band selection and estimation of material abundances using spectral filters (German paper). *tm – Technisches Messen*, 85(6):454–467, 2018.
23. W. Krippner, S. Bauer, and F. Puente León. Optical determination of material abundances in mixtures (German paper). *tm – Technisches Messen*, 84(3):207–215, 2017.
24. W. Krippner, S. Bauer, and F. Puente León. Considering spectral variability for optical material abundance estimation. *tm – Technisches Messen*, 85(3):149–158, 2018.
25. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
26. I. Meganem, P. Déliot, X. Briottet, Y. Deville, and S. Hosseini. Linear–quadratic mixing model for reflectances in urban environments. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):544–558, 2013.
27. M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
28. B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson. Hyperspectral unmixing using a neural network autoencoder. *IEEE Access*, 6:25646–25656, 2018.
29. P. Y. Simard, D. Steinkraus, J. C. Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition*, volume 3, 2003.
30. M. O. Smith, J. B. Adams, and D. E. Sabol. Spectral mixture analysis-new strategies for the analysis of multispectral data. In J. Hill and J. Mégier, editors, *Imaging spectrometry – a tool for environmental observations*, pages 125–143. Springer, 1994.
31. M. A. Veganzones, L. Drumetz, G. Tochon, M. Dalla Mura, A. Plaza, J. Bioucas-Dias, and J. Chanussot. 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)*, pages 1–4. IEEE, 2014.
32. X. Xu, Z. Shi, and B. Pan. A supervised abundance estimation method for hyperspectral unmixing. *Remote Sensing Letters*, 9(4):383–392, 2018.

33. X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao. Hyperspectral unmixing via deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1755–1759, 2018.

Bionotes

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

Johannes Anastasiadis studied electrical engineering and information technology at Karlsruhe Institute of Technology (KIT). Since 2017, he has been working as a research associate at the Institute of Industrial Information Technology (IIIT) at KIT. His current research focuses on processing hyperspectral images using artificial neural networks, with spectral unmixing as the main application.

Michael Heizmann
Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstr. 16, 76187 Karlsruhe, Germany
michael.heizmann@kit.edu

Michael Heizmann received the M. Sc. degree in mechanical engineering and the Ph. D. degree in automated visual inspection from the University of Karlsruhe, Germany, in 1998 and 2004, respectively. From 2004 to 2009, he was a Postdoctoral Research Assistant with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Karlsruhe, Germany, where he was the Head of the Department Systems for Measurement, Control and Diagnosis, from 2009 to 2016. From 2014 to 2016, he was a Professor of mechatronic systems with the Karlsruhe University of Applied Sciences. Since 2016, he has been a Full Professor of mechatronic measurement systems and has been the Director of the Institute of Industrial Information Technology, Karlsruhe Institute of Technology. His research interests include measurement and automation technology, machine vision and image processing, image and information fusion, and their applications.