# Spatially resolved ingredient detection in spice mixes using 3D convolutional neural networks

Johannes Anastasiadis, Wolfgang Krippner,
and Fernando Puente León

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

**Abstract** A method using spectral information to detect substances in mixtures is given. The presented convolutional neural network is using three-dimensional convolutions to process hyperspectral images. Reflectance values can be fed directly into the network and are not preprocessed. Due to the architecture, the neural network performs a spatially invariant operation. Detection performance is demonstrated by a dataset containing spice mixtures.

**Keywords:** Hyperspectral image, optical measurement, convolutional neural network, three dimensional.

## 1 Introduction

Optical measuring methods play a major role in food investigation as non-contact and non-destructive methods. They can be used for quality assessment, e.g., by detection of undesired substances. Hyperspectral images (HSIs) are often used if normal colour images do not provide enough information. While the latter only comprise three colour channels (red, green, and blue), the former contain up to several hundred wavelength channels [1]. By this additional information, conclusions about material properties can be drawn [2–4].

Artificial neural networks have been very successful in recent years. Convolutional neural networks (CNNs) are particularly successful in image processing [5]. They are also used to process HSIs, but many approaches only perform a convolution along either the spectral dimension [6] or the spatial dimensions [7, 8]. To merge information of

DOI: 10.58895/ksp/1000087509-4 erschienen in:

**OCM 2019 - 4th International Conference on Optical Characterization of Materials, March 13th – 14th, 2019, Karlsruhe, Germany**

DOI: 10.58895/ksp/1000087509 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000087509/

the two domains, several approaches exist in literature. The mayor amount uses fully connected layers [9, 10]. In [11] three-dimensional (3D) convolutions are used, however, fully connected layers are needed in later process steps to get an output value for each pixel.

In our approach, 3D convolutions are used in the first layers to process information of spatial and spectral domain simultaneously. The following layers use $1 \times 1$ two-dimensional (2D) convolutions along the spatial dimensions, which can be regarded as a full connection along the spectral dimension. Because of this, the approach is spatially invariant. Furthermore, we do not require any preprocessing such as principle component analysis, used in [7], for instance. Therefore, the CNN uses the complete information provided by HSIs. In this work, the approach is applied to detect substances in mixtures. The aim is to decide whether a substance is comprised in a pixel.

The rest of the paper is organized as follows: Basics of neural networks and CNNs are given in Section 2. In Section 3, the structure of the proposed CNN is described. The results attained by this CNN are shown in Section 4. A brief summary is given in Section 5.

## 2 Convolutional neural networks

Neural networks are black box modelling approaches in which data is used to learn non-linear functions. The basic modules of neural networks are called neurons, which are inspired by biological neurons. Each neuron consists of several inputs and one output. The neurons are connected with each other, and those connections can end up in loops. Feedforward neural networks, as used in this work, have no loops, and neurons are arranged in layers. Every neuron of each layer is connected to every neuron of the previous layer and to every neuron of the following layer. There are no connections within a layer. Such layers are called fully connected layers. A single layer is described mathematically as

$$\mathbf{h} = \boldsymbol{\varphi}(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{4.1}$$

where $\mathbf{h} \in \mathbb{R}^K$ is the output and $\mathbf{x} \in \mathbb{R}^J$ the input of the layer. The matrix $\mathbf{W} \in \mathbb{R}^{K \times J}$ contains the weights the input is multiplied by, and $\mathbf{b} \in \mathbb{R}^K$ is the bias vector. There is one scalar bias value for every

neuron. The non-linear activation function $\varphi$ is applied elementwise. It is necessary to enable approximations of functions different from linear functions. The neural network is trained by adjusting weights and biases for every layer. This is done by optimizing an objective function. In most cases, a gradient-based method is used. The gradient can be backpropagated through the neural network to update all parameters [12].

In CNNs, layers are not fully connected. Instead, the input is convolved with a filter kernel which is much smaller than the input. Afterwards, a scalar bias value is added. This leads to some advantages in image processing. First of all, the same kernels are used for every region of the image, and therefore, the operation is spatially invariant. The spatially resolved approach proposed in this work exploits this fact. Furthermore, spatial relations of data are taken into account. Last but not least, significantly less parameters are required compared to fully connected layers. Only the parameters describing the kernels and the biases have to be trained.

An important aspect to understand CNNs in image processing is the treatment of channels or feature maps. A convolution is performed along spatial dimensions (two dimensions for colour images). This is done with a different filter kernel for every channel (or feature map). The output of this operation is added up to a new feature map. This is done with several filter kernel sets to produce more feature maps and, thereby, extract more features [5]. It can be interpreted as the network is convolutional along spatial dimensions and fully connected in spectral direction (see Fig. 4.1). A bias value is added to every pixel of the output feature maps afterwards. In Fig. 4.1, a 2D convolution is shown as an example.

In most CNNs, pooling layers, in which local clusters of values are combined to a single value (see e.g. [13]), are also used. Commonly, and also used in this work, is max pooling, which propagates only the largest value to the next layer. Using pooling leads to less parameters in the subsequent layers, which results in shorter training times and reduces the risk of overfitting.

All basics provided in this section are used in the next section to design a CNN consisting of convolutional and pooling layers.

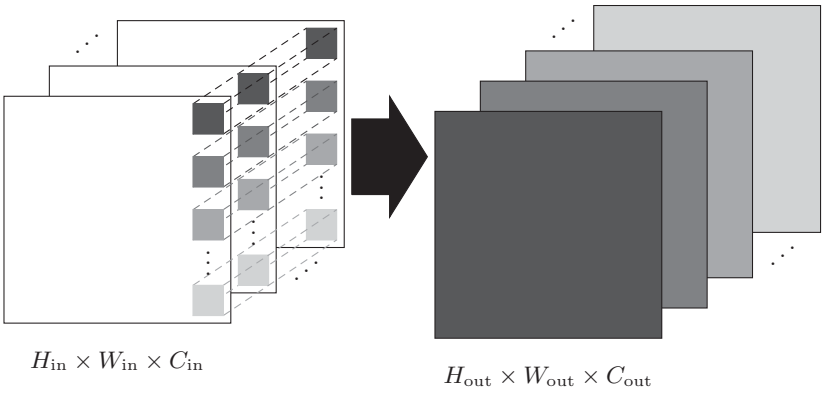


**Figure 4.1:** Principle of a convolutional layer (without bias): The input data has $C_{\text{in}}$ feature maps of size $H_{\text{in}} \times W_{\text{in}}$ pixels. The input is convolved with $C_{\text{out}}$ sets of filter kernels. Each set contains one filter kernel for each input feature map, respectively. The sum of the convolutions of each filter kernel set results in an output feature map. Therefore, the output has $C_{\text{out}}$ feature maps of size $H_{\text{out}} \times W_{\text{out}}$ pixels.

## 3 Neural network design

The input to the CNN are HSIs, which can be interpreted as 3D data cubes with two spatial and one spectral dimension. To each element of the cube a reflectance value is assigned.

The proposed CNN consists of two parts: The first part exploits 3D convolutions along the spatial and the spectral dimensions resulting in 3D feature maps. After each convolutional layer, a pooling layer along the spectral dimension is used. For this reason, the spatial resolution is preserved (see Fig. 4.2, first row). This design allows for getting the position of a detected ingredient. In the second part, the 3D feature maps are transformed into 2D feature maps by splitting them along the spectral dimension (see Fig. 4.2, second row). For example, $W$ feature maps of size $X \times Y \times \Lambda$ lead to $W \cdot \Lambda$ feature maps of size $X \times Y$ after splitting. It is an important step in our approach to combine 3D convolutions in the first few layers with 2D $1 \times 1$ convolutions along the spatial dimensions in the subsequent layers (see Fig. 4.2). The 3D con-

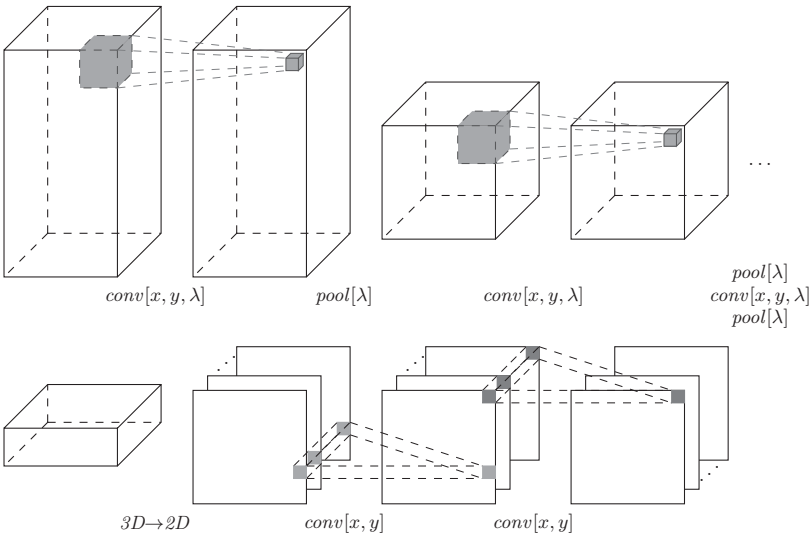


**Figure 4.2:** Proposed net architecture: The first three layers are 3D convolutional (*conv*) layers followed by a pooling layer (*pool*), respectively. The last two layers perform a 2D $1 \times 1$ convolution. The square brackets define along which dimensions the operation is performed (spatial dimensions: $x, y$, spectral dimension: $\lambda$). Note that only one 3D feature map is shown for each step.

volutional layers are used for feature extraction. The 2D convolutional layers operate as fully connected layers along the spectral dimension.

The CNN produces a map for each ingredient as an output. Having the same spatial resolution as the HSI, the maps may indicate where an ingredient is detected. The operation performed by the CNN is spatially invariant because only convolutions are performed along the spatial dimensions (see Section 2).

The CNN shown in Fig. 4.2 is evaluated in Section 4 with different filter sizes. In all experiments, batch normalisation is performed before activation [14], and in each layer, the sigmoid function $\sigma : \mathbb{R} \to \mathbb{R}$ is used as the activation function:

$$\sigma(z) = \frac{1}{1 + \mathrm{e}^{-z}} \,. \tag{4.2}$$

## 4 Experimental results

The dataset used for training and evaluation of the CNN was acquired in the image processing laboratory of the Institute of Industrial Information Technology. Eleven spices were mixed in 155 mixtures, each consisting of maximum four different spices. Hyperspectral images of the mixtures with a spatial size of $24 \times 24$ pixels were acquired. They consist of 91 wavelength channels from 450 nm to 810 nm. A white balance was applied in order to ensure reflectance values as data. The dataset is divided into a training and a test set with a ratio of 2:1. The input of the CNN are HSIs, each containing several mixtures (see Fig. 4.2).

To evaluate the result F-measure $F$ is used. It is the harmonic mean of precision $PRE$ and recall $REC$:

$$F = \frac{2 \cdot PRE \cdot REC}{PRE + REC}\,, \tag{4.3}$$

$$PRE = \frac{TP}{TP + FP}\,, \qquad REC = \frac{TP}{TP + FN}\,. \tag{4.4}$$

In Equation (4.4) $TP$ is the number of true positives, $FP$ the number of false positives, and $FN$ the number of false negatives.

In the following sections, several parameter sets are compared with both each other and with the method provided by *Makantasis et al.* [7]. This method only uses 2D convolutions along the spatial dimensions. The spectral dimension is treated as a channel or feature map, respectively (see Fig. 4.2, second row). To account for the spectral information, Randomized Principal Component Analysis (R-PCA) is used along the spectral dimension. For all experiments, the sizes of the filter kernels were chosen according to [7].

### 4.1 Comparison of filter sizes

In this section, an appropriate size for the filter kernels used in the 3D convolutional layers is determined. The size of the 2D filter kernels is restricted to $1 \times 1$. In Table 4.1, the number of output feature maps for all experiments is given. The number of input channels is one.

**Table 4.1:** Number of the output feature maps of each convolutional layer: The number of input feature maps of the next layer corresponds to the number of output feature maps of the current layer, except for layer 4. Here, the number of input feature maps is the number of output feature maps times the size of the spectral dimension of layer 3 (see Section 3).

| Convolutional layer | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Output feature maps | 16 | 32 | 64 | 66 | 11 |

In Figure 4.3, several filter kernel sizes are compared using F-measure. The proposed CNN performs better than the method by *Makantasis et al.* [7] for all filter sizes. This implies that using 3D convolutions works better than using R-PCA for feature extraction. The inclusion of spatial information ($w_{xy} > 1$) improves the results, but depends on the setting of the edge lengths $w_{xy}$ and $w_\lambda$. If the spatial edge lengths are chosen too large, the performance decreases. The best result is achieved for $w_\lambda = 7$ and $w_{xy} = 3$. In addition, for smaller $w_\lambda$ larger $w_{xy}$ are beneficial.

The detection performance depends not only on the parameters, but also on the input data. Therefore, all the spices contained in the mixtures are evaluated separately in the next section.

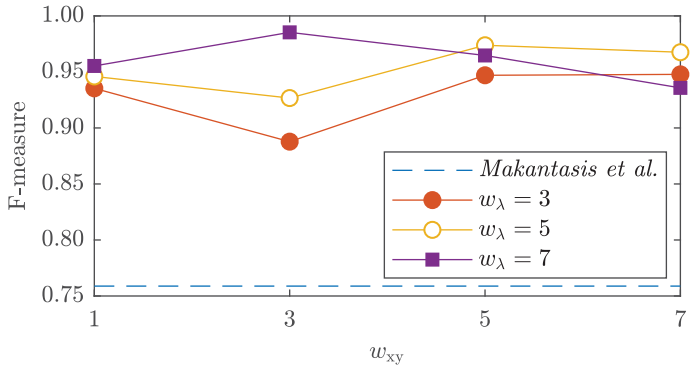


**Figure 4.3:** F-measure values for several filter sizes used in the first three layers (see Fig. 4.2, first row). Here, $w_{xy}$ is the spatial edge length and $w_\lambda$ the spectral edge length of the filters. The method by *Makantasis et al.* [7] is used for comparison with the recommended filter sizes.

### 4.2 Comparison of spices

Figure 4.4 shows F-measures calculated separately for all spices. In particular, the method in *Makantasis et al.* [7] and the proposed CNN using two different parameter sizes are investigated. The parameter size that leads to the overall best F-measure, and the one that leads to the best F-measure including no spatial information (see Fig. 4.3) are compared.

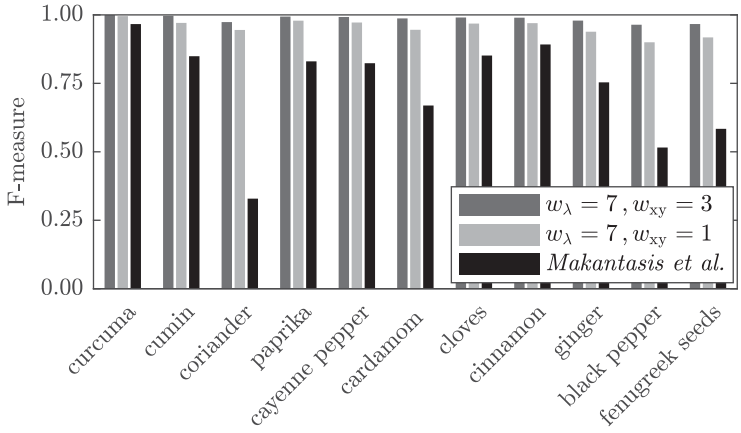


**Figure 4.4:** F-measure for all spices using the best filter size settings. *Makantasis et al.* [7] with recommended settings is displayed for comparison.

The accuracy of spice detection variates between the spices for all methods. The method by *Makantasis et al.* [7], using R-PCA and only 2D convolutions, shows much higher variance than our method, using 3D convolutions. Besides, spices, which bad detection performance using the method by *Makantasis et al.* [7], also lead to lower F-measure values using the proposed CNN. We conclude that the accuracy of the detection depends on the input data. This has much stronger implications for *Makantasis et al.* [7] than for our method.

## 5 Summary

A CNN architecture using only convolutional layers along the spatial dimensions has been presented. It was shown that the our CNN design performs a spatially invariant operation and maintains the spatial resolution of the input HSI. The CNN is performing 3D convolutions in the first few layers to extract features and is fed with non preprocessed reflectance values. In this work, the goal of the CNN is to detect ingredients in spice mixtures and was evaluated by a dataset created in our laboratory. Including spatial information in the 3D convolutional layers leads to the best results. Nevertheless, the size of the neighbourhood should not be chosen too large. A CNN, which uses only 2D convolutions along the spatial dimensions [7] and R-PCA for preprocessing, is outperformed by the proposed CNN. The accuracy of detection depends on the considered spice mixtures for all evaluated methods.

In the future, the CNN could be trained with data containing the quantitative amount of spices and used to determine the amount of ingredients.

## References

1. 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*, vol. 18, no. 12, pp. 590–598, 2007.
2. S. Bauer, J. Stefan, and F. Puente León, "Hyperspectral image unmixing involving spatial information by extending the alternating least-squares algorithm," *tm – Technisches Messen*, vol. 82, no. 4, pp. 174–186, 2015.
3. W. Krippner, S. Bauer, and F. Puente León, "Considering spectral variability for optical material abundance estimation," *tm – Technisches Messen*, vol. 85, no. 3, pp. 149–158, 2018.
4. W. Krippner, S. Bauer, and F. Puente León, "Optical measurement of material abundances in mixtures incorporating preprocessing to mitigate spectral variability," in *OCM 2017*. Karlsruhe: KIT Scientific Publishing, 2017, pp. 87–96.

5. Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.
6. W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.
7. K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 4959–4962.
8. S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.
9. Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.
10. C. Chen, F. Jiang, C. Yang, S. Rho, W. Shen, S. Liu, and Z. Liu, "Hyperspectral classification based on spectral-spatial convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 165–171, 2018.
11. Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sensing*, vol. 9, no. 1, 2017.
12. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
14. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.