Estimation of strength and position of individual sound sources with Deep Learning using microphone array data

Adam Kujawski, Gert Herold, Ennes Sarradj, Simon Jekosch

FG Technische Akustik, Technische Universität Berlin, 10587 Berlin, Deutschland, adam.kujawski@tu-berlin.de

Introduction

Microphone array techniques have proven their worth for the localization and characterization of acoustic sources in various application areas such as acoustic measurement technology, medical ultrasound or speech signal processing. When microphone arrays are used in acoustic measurement technology, the primary goal is to identify and quantify the causes of sound. However, a low dynamic range often makes it difficult to assign the illustrated contributions in the conventional beamforming map to their causes and to identify the source mechanisms. Therefore, different methods have been developed in the past which differ greatly in terms of accuracy, computational effort and robustness against disturbance influences.

At the same time, a large number of research projects in the field of supervised machine learning for image processing have shown that it is possible to recognize various types of objects and its properties on the basis of images. In this contribution it was therefore examined whether the use of convolutional neural networks (CNN) is suitable for the identification of point sources using the conventional beamforming map as a visual representation. Synthetically generated data from single sound sources were used for this purpose.

State of the art

Convolutional neural networks (CNN) according to LeCun et al. [6] have already been investigated as an alternative to existing beamforming algorithms in other research projects, especially in the field of ultrasonic research.

Reiter et al. [7] showed that a reliable estimation of different point source locations is possible on the basis of photoacoustic images. In addition, the authors discovered that multiple sources can be identified within an image, even if it was trained exclusively with single sources. This approach was further developed by Allman et al. [1]. By using methods from object recognition and object classification, it could be shown that pseudo-sources which result from reflections on structures and several real sources can be reliably differentiated.

The results from the literature mentioned in this section are first approaches showing the promising potential of neural networks as methods for sound source localization. At the same time there are strong restrictions. So far, in all the approaches mentioned, only conclusions have been drawn about the position of possible sources, but not about their strength.

Methodology

Since extracting spatial features from images is an advantageous ability of CNNs, it is conceivable to find the true source distribution outgoing from the conventional beamforming map $\mathbf{B} \in \mathbb{R}^{m \times n}$, calculated with delay-and-sum beamforming in the frequency domain. Therefore, a CNN represented as a function $\mathcal{F}_{\text{cnn}}$ is trained, which maps from the conventional beamforming map to a vector which holds the predicted source position and strength $\mathcal{F}_{\text{cnn}} : \mathbb{R}^{m \times n} \to \mathbb{R}^3$

$$\mathcal{F}_{\text{cnn}}(\mathbf{B}) \coloneqq \mathbf{y}, \tag{1}$$

with $\mathbf{y} \coloneqq [x_1, x_2, p^2] \in \mathbb{R}^3$. Due to the unknown quantity of sources in a map, an iterative application of the function with an intermediate step of removing the components belonging to the approximated source would be necessary if multiple sources are present. But since the feasibility of the approach is investigated, only single sources are considered here.

For the optimization of the trainable variables of the network in training mode, an objective error function $\mathcal{L}$ is necessary, which measures the error of the network on a specific input sample. The mean square error was selected and calculated for strength and position as follows:

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{2}(x_i - \hat{x}_i)^2 + (p^2 - \hat{p}^2)^2. \tag{2}$$

The hat is denoting the true values. However, this type of error is not suitable to estimate the performance of the network as an inverse beamforming method. For example the loss value for a given input does not explain if it is caused by a wrong positioning or by a wrong estimation of the amplitude or both. Therefor in the evaluation of the model the distance error and the level error are used for monitoring the performance of the model.

$$\mathcal{E}_{\text{dist}} = \|\mathbf{x} - \hat{\mathbf{x}}\| \tag{3}$$

$$\mathcal{E}_{\text{level}} = |L_{\text{p}} - \hat{L}_{\text{p}}| \tag{4}$$

Model Architecture

The choice of the architecture has some restrictions compared to the application of neural networks in usual image processing tasks. Namely the input-image size is different for beamformed data in comparison to normal images, which are normally 256 × 256 pixels. As an example, Herold and Sarradj [4] used for comparison of different beamforming-methods a grid of the size 51×51.

Table 1: Network architecture of the used residual network based on He et al. [2, 3]

Block	Dimension input	Dimension output	No. kernels	Size	No. weights
ConvLayer	$51 \times 51 \times 1$	$51 \times 51 \times 26$	26	3×3	234
Residual Layer 1	$51 \times 51 \times 26$	$26 \times 26 \times 26$	$\begin{bmatrix} 26 \\ 26 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 3$	37180
Residual Layer 2	$26 \times 26 \times 26$	$13 \times 13 \times 52$	$\begin{bmatrix} 52 \\ 52 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 3$	135200
Residual Layer 3	$13 \times 13 \times 52$	$7 \times 7 \times 104$	$\begin{bmatrix} 104 \\ 104 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix} \times 3$	540800
AvgPoolLayer	$7 \times 7 \times 104$	104×1	1	7×7	-
Regression Layer	$104 \times 1, 104 \times 1$	$334 \times 1, 334 \times 1$	-	334 nodes	35070, 35070
Output Layer	$334 \times 1, 334 \times 1$	$2 \times 1(\mathbf{x}), 1 \times 1(p^2)$	-	3 nodes	670, 335

Since many of the developed convolutional neural network models for image recognition tasks perform a spatial subsampling starting from the input data, a too small input image can not be processed by these networks. As a requirement the model architecture should be applicable also to smaller images. For this reason, a residual neural network invented by He et al. [2, 3] is used. Beside the successful application for image recognition tasks, it has already been used for processing of tiny images [2], e.g. the CIFAR-10 data set with a dimension of 32×32 pixels.

Table 1 shows the general architecture of the network. This initially consists of a convolutional layer. Then, the input feature map is processed by different residual layers. Every residual layer reduces the dimension of the input feature map by half. After the feature maps have been reduced in size, an average pooling layer follows, which averages the output feature maps of the last residual layer. A single vector remains, which is linked to the output of the network by a corresponding regression layer. This layer processes the values for the position $\mathbf{x} := [x_1, x_2]$ and strength p^2 separately in two parallel fully-connected hidden layers with 334 nodes. The output values of the hidden layer have not passed through a nonlinear function. Thus, the regression layer corresponds to a simple linear transformation. The structure of the regression layer was determined by a random search experiment.

Apart from the appended regression layer the architecture is similar to the one introduced by He et al. [2] on the CIFAR-10 dataset. The optimized building block in the residual layers investigated by He et al. in 2016 [3] with batch normalization and nonlinear ReLU function before each weight layer was used.

Data set

The synthetic data sets for this contribution are calculated with the Acoular package [9] and are based on the work of Herold and Sarradj [4]. The properties are shortly explained in the following.

A virtual microphone array consisting of 64 sensors is used. This array is focusing an area as regularly-spaced rectangular grid laying in a resting, homogeneous fluid. Herold and Sarradj also used a slight jitter on the position of the microphones to simulate realistic measurement conditions. The same was only applied to the data generated for the test set but not to the data used for training. In the area of interest, single monopole sources emitting uncorrelated white noise. The source positions of the sources in the test data set following a bivariate normal distribution. In contrast, the positions of the sources in the training data set are sampled from a bivariate uniform distribution for the following reason. It is desired that the trained algorithm does not behave differently in its accuracy depending on the position of the source. Since the PSF is shift variant, it can not be assumed that the algorithm can transfer local mapping properties within one area to others without explicit training.

Note that the positions of the simulated sources are mostly laying in between the grid points, as it is under realistic conditions. The resulting time data at the microphones are simulated, following the parameters in Table 4. The CSM is calculated using Welch's method with the main diagonal removed. The sound maps are generated for various third-octave bands covering Helmholtz numbers from 1 to 16. Furthermore, steering vector formulation III according to Sarradj [8] was used as the transfer function from the individual grid points to the sensor positions. For the test data set there are 613 different positions of single sources. For the training data 10000 positions were considered. Table 5 summarizes the data sets with its properties. All sound maps were normalized to a maximum value of 1.

Experimental Settings

During training, 32 random samples were randomly taken from the training data set and presented to the model in

Table 2: Environment parameters according to [4]

Environment	resting, homogeneous fluid
Array	7 logarithmic spirals, 64 sensors
Focus grid	$x, y \in [-0.5, 0.5], z = 0.5, \Delta x = 0.02$

Table 4: Processing parameters according to [4]

Sampling rate	20 kHz
No. of time samples	512000
Block size	1024 samples
Block overlap	50 %
Windowing	von Hann / Hanning
CSM main diagonal	removed
Steering vector	fromulation 3, see [8]
Evaluation basis	third-octave bands
Frequency range	$He_{min} = 1, He_{max} = 16$

Table 3: Sound source parameters according to [4].

Source type	monopole
Source positions	Training: normal distributed, Test: uniform distributed
Signals	uncorrelated white noise

Table 5: Properties of data sets

Properties	Training Data	Test Data
Sensor disturbance	False	True
position distribution	uniform	normal
No. source positions	10000	613
No. third-octave bands	13	13
No. sound maps	130000	7969

each iteration steps. For cross validation, all 7969 sound maps of the test data set were used to mark the best state of the model every 500 iteration steps. A training state was being stored when the loss has improved over the cross-validation data set compared to the previously saved training state. A total of 100000 iteration steps have been performed. For the optimization of the trainable variables of the network, the Adam optimization algorithm [5] with the parameters in Table 6 was used. The optimization was done on a node of a CPU cluster consisting of four Intel Xeon CPU E5-2620 v4 (32 CPUs).

Table 6: Training settings

properties	values
loss-function $\mathcal{L}$	MSE, see eq. 2
No. iteration steps	100000
cross-validation interval	500
cross-validation metrics	$\mathcal{L}$, $\mathcal{E}_{\text{dist}}$, $\mathcal{E}_{\text{level}}$
batch size	32
learning rate η	0.0049
β_1	0.905
β_2	0.772

Results

Training the model took about 27 hours for all iteration steps. Figure 1 shows the development of the individual error metrics over the training and cross-validation process. The loss function itself provides only little information about the success of the optimization in terms of source characterization. Therefore, in the following only the development of the level and distance error via the optimization process will be discussed. The grey curve shows the mean error over a training batch in each iteration step. The black curve displays the mean error over the test data set. The vertical dotted line indicates the training state with the lowest loss value occurred during cross-validation.

In general, one can see a decaying convergence behaviour. Since both errors decrease constantly and the cross-validation error does not increase, it can be assumed that no overfitting occurred and the global minimum was achieved. Regarding the distance error on the left of Figure 1, both curves show the same tendency of decreasing error magnitude. This shows that the trained localization ability can be very well generalized to the data from the test data set. With a mean error of $\mathcal{E}_{\text{dist}} = 0.004$, a precision better than the grid-resolution of $\Delta x = 0.02$ has been achieved. Regarding the level error on the right in Figure 1, it is noticeable that the error over the individual training batches indeed decrease, but this has less influence on the test data set with ongoing training. However, the average level error of $\mathcal{E}_{\text{level}} = 0.014\,\text{dB}$ is still remarkably low.

The training results are confirmed if one considers the estimates of the model for sound maps from the cross-validation data set with the lowest and highest occurring Helmholtz number as examples in Figure 2. The images 2a and 2c show the level representation of the input map at $He = 1$ and $He = 16$. The sound maps 2b and 2d show the corresponding point sources estimated by the method as a black dot. By using the trained model, it is possible to precisely characterize the point source with a level error significantly less than 1 dB. Moreover, the estimated location of the source corresponds to the true position.

Discussion

The proposed method for sound source characterisation with methods of image recognition shows a promising potential. The results shown are surprisingly positive for the following reasons. First, it should be mentioned that the conventional beamforming map only consists of one channel representing the squared sound pressure, while in contrast normal images are consisting of three channels representing the color values. This means, that the information content is reduced compared to normal images. Furthermore an additional challenge is the similarity of the objects with regard to their geometric shape and the shift variance of the point spread function. However, this challenge has not turned out to be a problem.

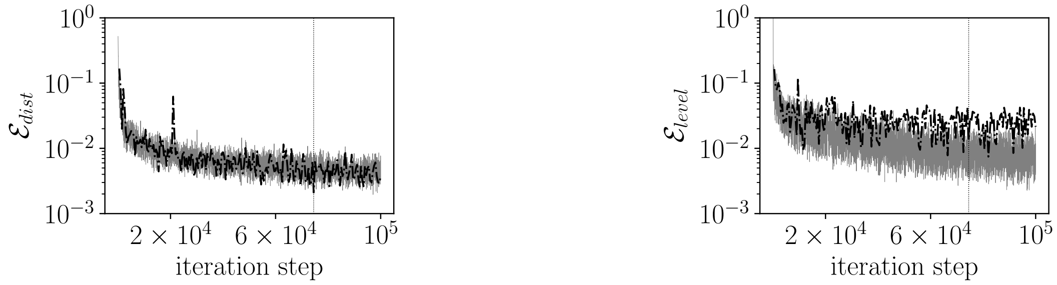

Figure 1: Development of the individual error metrics about the training process (grey: training, black: cross-validation). The vertical dotted line shows the iteration step for which the lowest error occurred during cross-validation.

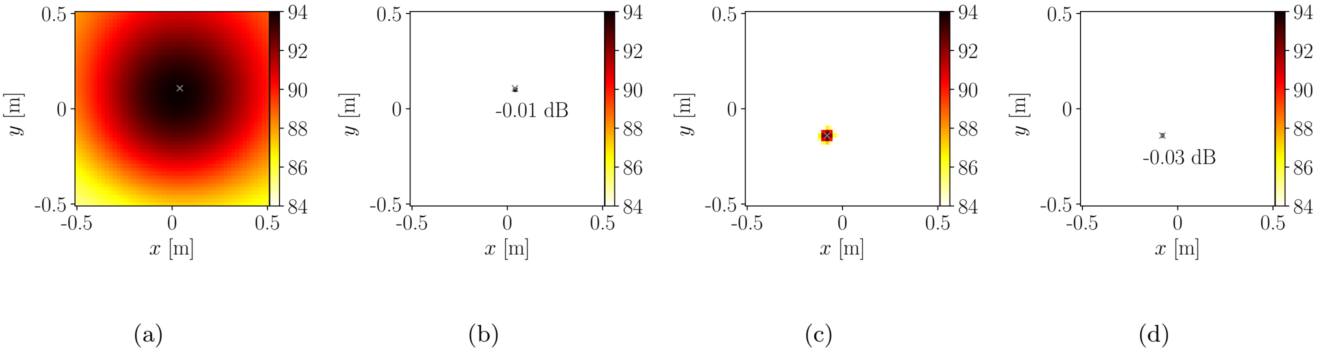

Figure 2: Examples of the conventional beamforming map ($He = 1$ (a) ,$He = 16$ (c)) and the sound map estimations ($He = 1$ (b) ,$He = 16$ (d)) by $\mathcal{F}_{\text{cnn}}$. The grey cross marks the true position of the simulated point source. The level error is far below 1 dB.

The low level error is not surprising, since the transfer function used is steering vector formulation III, which already provides a nearly correct source strength. However, according to Sarradj [8] the maximum value in the sound map does not match the correct position. The more surprising is the high accuracy of the source localization, which exceeds the spatial resolution of the input data. This fact underlines once again the suitability of the chosen approach.

The detection of individual point sources does not correspond to any task occurring in reality. Therefore, the next step is to investigate the method when several sources are present in the conventional beamforming map. Of particular interest here is how well, for example, weaker hidden sources can be detected.

References

[1] D. Allman, A. Reiter, and M. A. L. Bell. Photoacoustic Source Detection and Reflection Artifact Removal Enabled by Deep Learning. *IEEE Tran. on Med. Imaging*, 37(6):1464–1477, 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. and Pattern Recognit. (CVPR), Las Vegas*, pages 770–778, 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision, Amsterdam, The Netherlands*, volume 9908, pages 630–645, 2016.

[4] G. Herold and E. Sarradj. Performance analysis of microphone array methods. *J. Sound Vib.*, 401:152–168, 2017.

[5] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *In Proceedings of the International Conference for Learning Representations (ICLR), San Diego, 2014*, pages 1–15, 2014.

[6] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems*, 2:396–404, 1990.

[7] A. Reiter and M. A. L. Bell. A machine learning approach to identifying point source locations in photoacoustic data. *Proceedings of SPIE, Photons Plus Ultrasound: Imaging and Sensing*, 10064:100643J–1, 2017.

[8] E. Sarradj. Three-dimensional acoustic source mapping with different beamforming steering vector formulations. *Adv. Acoust. Vib.*, pages 1–12, 2012.

[9] E. Sarradj and G. Herold. A Python framework for microphone array data processing. *Appl. Acoust.*, 116:50–58, 2017.