Auswahl geeigneter Mikrofonarrayverfahren mithilfe von Convolutional Neural Networks

Simon Jekosch, Adam Kujawski, Ennes Sarradj und Gert Herold

Technische Universität Berlin, Einsteinufer 25, 10587 Berlin, Deutschland,

Email: s.jekosch@tu-berlin.de, adam.kujawski@tu-berlin.de, ennes.sarradj@tu-berlin.de und gert.herold@tu-berlin.de

Einleitung

Die Berechnung von aeroakustischer Quellverteilung mit Hilfe von Mikrofonarrays ist ein beliebtes Mittel für die Charakterisierung von Schallquellen. Für die Berechnung der Quellverteilungen besteht, neben klassischem Beamforming, eine große Auswahl an Entfaltungs- und inversen Verfahren, die zum Teil jedoch hohe Rechenleistung benötigen. Da die Verfahren zu Teil stark von einander abweichende Ergebnisse liefern, ist es für praktische Anwendung hilfreich, bei einer Auswertung der Messung eines unbekannten Schallfeldes abschätzen zu können, wie geeignet ein beliebiges Verfahren aller Voraussicht nach sein wird. Das Ziel dieses Beitrages ist es Vorhersage über die Tauglichkeit verschiedener Mikrofonarrayverfahren für unbekannte Messdaten zu geben. Dazu wurde untersucht, in wie weit die Verwendung von Convolutional Neural Networks zur Identifizierung von geeigneten Mikrofonarrayverfahren möglich ist.

Als Grundlage des Trainings dient ein Datensatz aus Synthetischen Messdaten einer Monte-Carlo-Simulation. Dieser umfasst 12600 bekannte Quellanordnungen mit den zugehörigen Pegelabweichungen der Quellen für die verschiedenen Mikrofonarrayverfahren. Eingangsdaten für die Berechnung sind Quellkartierung, die mit schnellem konventionellem Beamforming berechnet wurden. Das dazugehörige Target ist das Mikrofonarrayverfahren mit den kleinsten Pegelabweichungen der Quellen.

Datensatz

Der Datensatz basiert auf der Arbeit von Herold und Sarradj [1]. Als Grundlage für die Auswertung wurden 12600 synthetische Datensätze mit bekannter Schallquellverteilung erzeugt. Die Datensätze unterscheiden sich in der Anzahl an unkorrelierten Quellen, sowie Position und Quellstärke der einzelnen Quellen. Die Verteilungen wurden per Zufallsprinzip nach festgelegten Verteilungsfunktionen bestimmt. Anhand der simulierten Daten wurden Quellkartierung mit verschiedenen Entfaltungsmethoden berechnet und statistisch ausgewertet.

Für jede Quellkarte wurde die Gesamtpegeldifferenz aller Quellen, Pegelabweichung der Einzelquellen und inverse Pegeldifferenzen - des Auftretens von Quellen abseits der eigentlichen Positionen - berechnet. Abbildung 1 zeigt beispielhaft die Schallkartierung für eine der 12600 berechneten Quellverteilungen für das Terzband bei Helmholtzzahl 2 mit den drei berechneten Pegeldifferenzen.

Die statistische Auswertung über die 12600 Quellkarten

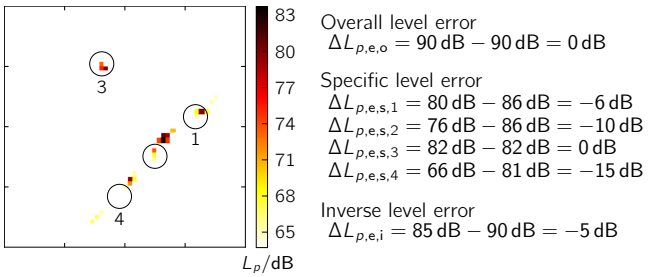

Abbildung 1: Beispiel einer Schallkartierung für den Clean-SC Algorithmus mit 4 simulierten Quellen und Helmholtzzahl 2. Die Kreise geben die korrekten Quellpositionen an. Die berechneten Pegeldifferenzen für diesen Fall sind beispielhaft aufgeführt.

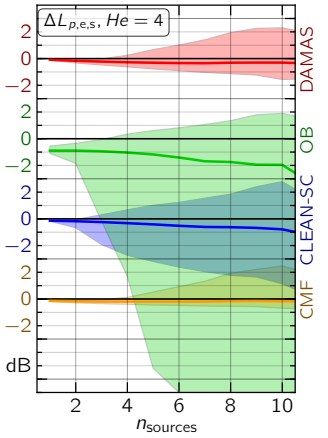

Abbildung 2: Abhängigkeit der Pegelabweichung der Einzelquellen von Quellanzahl.

zeigt die Abhängigkeit der Pegelabweichungen von der Frequenz, Quellenanzahl, örtlichen Verteilung und Dynamik der Quellen. Die Abbildung 2 zeigt beispielhaft den Median und die Standardabweichung der spezifischen Pegelabweichung für vier verschiedene Mikrofonarraymethoden. Für die Klassifizierung wurden in dieser Arbeit die Fehlermaße von vier Entfaltungsmethoden auf Basis der Kreuzspektralmatrix verglichen:

DAMAS [2] berechnet die Entfaltung aus der theoretischen räumlichen Filterantwort (point spread function) über ein modifizertes Gauss-Seidel-Einzelschrittverfahren.

CMF [3] Covariance Matrix Fitting ist ein inverses Verfahren, dass die Differenz zwischen einer gemessenen und einer aus unbekannten Quellen modellierten Kreuzspek-

tralmatrix minimiert.

OrthBF [4] Orthogonal Beamforming basiert auf der Eigenwertzerlegung der Kreuzspektralmatrix. Dabei wird der konventionelle Beamforming-Algorithmus auf verschiedene, aus den orthogonalen Eigenvektoren des Signal-Unterraums zusammengesetzte CSM angewendet.

CleanSC [5] zerlegt die Kreuzspektralmatrix in kohärente Anteile und entfaltet diese ophne vorherige Berechnung der Filterantworten.

Für die Entscheidung, welcher Algorithmus für die Berechnung einer Quellkarte am besten geeignet ist, wurde ein Gesamtfehlermaß aus spezifischer Pegelabweichung $L_{p,e,s,N}$ und inverser Pegelabweichung $L_{p,e,i}$ gebildet:

$$L_{p,g} = \sum_{n=0}^{N} \sqrt{(\Delta L_{p,e,s,N})^2} + \alpha \cdot \Delta L_{p,e,i} \qquad (1)$$

Dieser wird noch mit einem Regularisierungsparameter α gewichtet, um eine gleichmäßigere Verteilung der Algorithmen zu gewährleisten. Für den Datensatz wurde $\alpha = 0.01$ gewählt. Die Abbildung 3 zeigt die Häufigkeitsverteilung der Algorithmen nach ihrem kleinsten Gesamtfehlermaß.

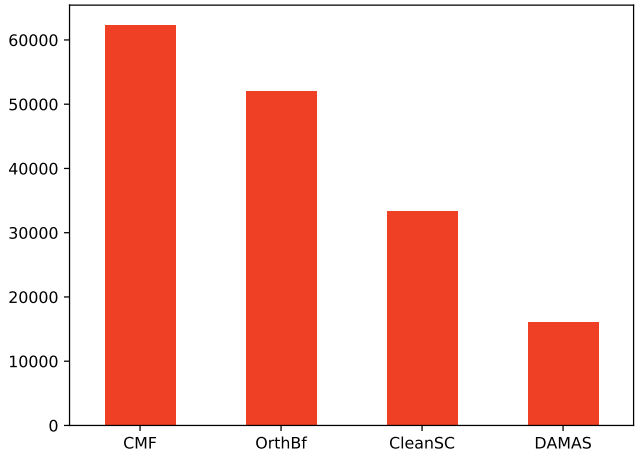

Abbildung 3: Häufigkeitsverteilung der Mikrofonarrayverfahren nach kleinstem Gesamtfehlermaß $L_{p,g}$

Modell

Für die Implementierung musste das Modell so gewählt werden, dass als Eingangsdaten die Quellkarten aus klassischem Beamforming mit einer Auflösung von $51x51$ Pixeln und als Targets die 4 Mikrofonarrayverfahren genutzt werden können. Diese Aufgabe entspricht im Grunde einer Bilderkennung, bei dem Muster in den Quellkarten erkannt und kategorisiert werden. Forschungen auf dem Gebiet der Bildverarbeitung haben gezeigt, dass Convolutional Neural Network in Kombination mit überwachtem Lernen die Möglichkeit bieten verschiedenste Eigenschaften zuzuordnen. In dieser Arbeit soll ein Modell verwendet werden, dass von He et al. [6],[7] entwickelt wurde. Das Residual Neural Network ist ein Convolutional Neural Network, welches aus gleichartigen Blöcken (Residual Layer) gebaut wird, die sich beliebig oft hintereinander schalten lassen. Das Netzwerk besteht insgesamt aus 22 Layern mit rund 725000 Variablen. Der Aufbau des gesamten Netzwerks ist in Tabelle 1 dargestellt. Für die Implementierung wurde das Software Framework Tensorflow[8] genutzt.

Processing Block	Dimension input	Dimension output	No. kernels	Size
ConvLayer	$51 \times 51 \times 1$	$51 \times 51 \times 26$	26	3×3
Residual Layer 1	$51 \times 51 \times 26$	$26 \times 26 \times 26$	$\begin{bmatrix}26\\26\end{bmatrix} \times 3$	$\begin{bmatrix}3\times3\\3\times3\end{bmatrix} \times 3$
Residual Layer 2	$26 \times 26 \times 26$	$13 \times 13 \times 52$	$\begin{bmatrix}52\\52\end{bmatrix} \times 3$	$\begin{bmatrix}3\times3\\3\times3\end{bmatrix} \times 3$
Residual Layer 3	$13 \times 13 \times 52$	$7 \times 7 \times 104$	$\begin{bmatrix}104\\104\end{bmatrix} \times 3$	$\begin{bmatrix}3\times3\\3\times3\end{bmatrix} \times 3$
AvgPoolLayer	$7 \times 7 \times 104$	104×1	1	7×7
Regression Layer	104×1	64×1	-	64 nodes
Regression Layer	64×1	64×1	-	64 nodes
Output Layer	64×1	$4 \times 1(a)$	-	4 nodes

Tabelle 1: Struktur des Residual Neural Network ResNet-v2 22 Layer

Ergebnisse

Für die Auswertung wurde der Datensatz in 3 Teile gespalten: einen Trainingsdatensatz mit 80% der Daten und damit 131040 Quellkarten, sowie einen Validierungs- und Evaluierungsdatensatz mit jeweils 80% der Daten und jeweils 16380 Quellkarten. Die Trainingsdaten wurden zum Anlernen der Residual Neural Networks benutzt, die Validierungsdaten zur Kreuzvalidierung während des Lernens und die Evaluierungsdaten zum Überprüfen des trainierten Netzes.

Training

Das Training der Modells wurde mit dem Adam-Optimizer Algorithmus [9] realisiert. Die Lernrate wurde auf 0.001 gesetzt, die Parameter für den Optimierer wurden gemäß der Literaturangaben gewählt ($\beta_1 = 0.9$,$\beta_2 = 0.999$ und $\epsilon = 10^{-8}$). Als Loss-Funktion wurde die Softmax-Cross- entropy Funktion gewählt:

$$L(y,p) = -\sum_i y_i \log(p_i), \quad p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a} \qquad (2)$$

Diese gibt den Abstand zwischen der Wahrscheinlichkeitsverteilung des Modellausgangs und der wirklichen Wahrscheinlichkeitsverteilung an. Als weitere Zielfunktion wurde außerdem die Genauigkeit (Accuracy) berechnet:

$$A(y,p) = \frac{\sum_i (TP(y,p))}{(TP(y,p) + TN(y,p) + FP(y,p) + FN(y,p))} \qquad (3)$$

Die Genauigkeit gibt an, wie oft die Targets mit dem Modellausgang übereinstimmen. Die Abbildung 4 zeigt den Verlauf der Loss-Funktion die Abbildung 5 den Verlauf

der Genauigkeitsfunktion jeweils über 6000 Trainingsepochen. Für die Trainingsdaten wird das Maximum an Genauigkeit mit 92% zum Ende des Trainings erreicht. Die Validierungsdaten zeigen jedoch eine Überanpassung des Netzwerks an die Trainingsdaten nach der 2000sten Trainingsepoche. Dieses äußert sich durch ein Ansteigen des Validierungsfehler, obwohl der Trainingsfehler weiter sinkt. Das Maximum der Genauigkeit für die Validierungsdaten liegt nach 1000 Epochen bei 58%.

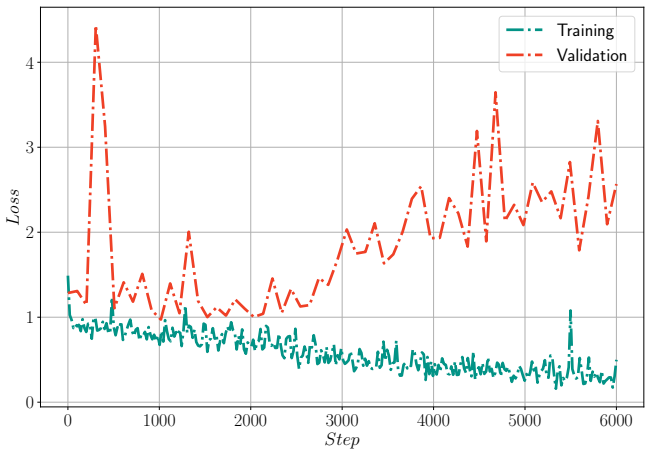

Abbildung 4: Funktion des Loss Wertes über die Trainingsepochen (Step) für den Trainings- und Validierungsdatensatz

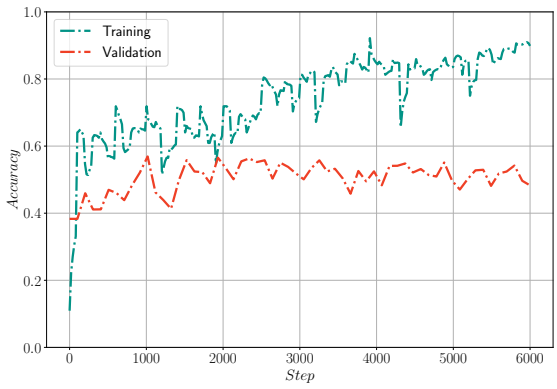

Abbildung 5: Funktion des Genauigkeit Wertes über die Trainingsepochen (Step) für den Trainings- und Validierungsdatensatz

Evaluierung

Für die Evaluierung wurde das trainierte Modell mit dem Testdatensatz und dem Evaluierungsdatensatz ausgewertet. Der Modellausgang wurde anschließend mit den Targets aus dem Datensatz vergleichen. Beim Vergleich von jedem der jeweils 4 Mikrofonarrayverfahren entsteht eine Verwechslungsmatrix zwischen dem Modellausgang und den Targets. Die Abbildung 6 bis 8 zeigen Verwechslungsmatrizen für die beiden Datensätze. Für den Trainingsdatensatz stimmen die Arrayverfahren sehr gut überein, nur bei orthogonalem Beamforming treten in 19% Verwechslungen mit CMF auf. Für den Evaluierungsdatensatz sind 2 Verwechslungsmatrizen dargestellt. Eine für die Trainingsepoche 6000, mit dem kleinsten Trainingsfehler, und eine für die Trainingsepoche 1000, mit dem kleinsten Validierungsfehler. Die Verwechslungsmatrix mit dem kleinsten Trainingsfehler in Abbildung 7 zeigt deutliche schwächen des Modells durch Überanpassung. Die Unterscheidung zwischen OrthBf und CMF durch das Modell ist fast nicht möglich und auch die Erkennungsrate bei DAMAS und CleanSC liegen unter 60%. Die Verwechslungsmatrix mit dem kleinsten Validierungsfehler in Abbildung 8 zeigt hingegen gute Ergebnisse für DAMAS und CMF mit Erkennungsraten von 89% bzw. 98%. Dafür wird Das Verfahren OrthBF nicht erkannt und dafür CMF zugeordnet. Die Erkennung für CleanSC ist ebenfalls nicht gegeben.

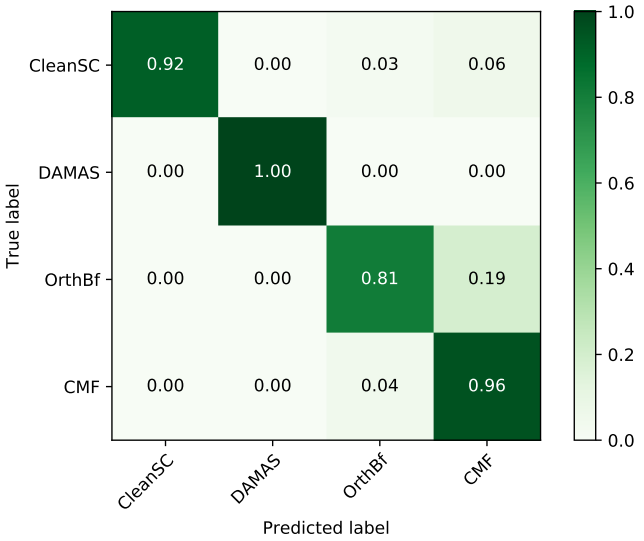

Abbildung 6: Verwechslungsmatrix zwischen Ausgang des Modells (Predicted Labels) und Targets des Datensatzes (True Labels) für Trainingsdaten und Trainingsepoche 6000.

Diskussion

Die Evaluation an Trainingsdaten konnte zeigen, dass das ResNet-Modell in der Lage ist, die Quellkarten aus konventionellem Beamforming über das Gesamtfehlermaß der Pegelabweichungen mit einem passendem Mikrofonarrayverfahren in Verbindung zu bringen. Damit konnte belegt werden, dass das Modell für diese Aufgabe brauchbar ist. Die Auswertung an dem Modell unbekannten Datensätzen hat jedoch hervorgebracht, dass die Generalisierbarkeit des Modells auf Daten außerhalb des Trainingsdatensatzes nur bedingt gegeben ist. In der aktuellen Fassung neigt das Modell im Training zum Overfitting. Dieses äußert sich durch starkes Ansteigen des Validierungsfehlers nach ca 2000 Trainingsepochen. Die Ursachen dafür könnten in der Beschaffenheit des Datensatzes liegen, was jedoch als unwahrscheinlich betrachtet wird da die Aufteilung zufällig erfolgte. Außerdem ist es möglich, dass zu wenig Regularisierung der Gewichte beim Training erfolgte. Dies könnte noch durch Verfahren wie weight decay, L2-Regularisierung oder Dropout, sowie benutzen anderer Optimieralgorithmen gesteuert werden.

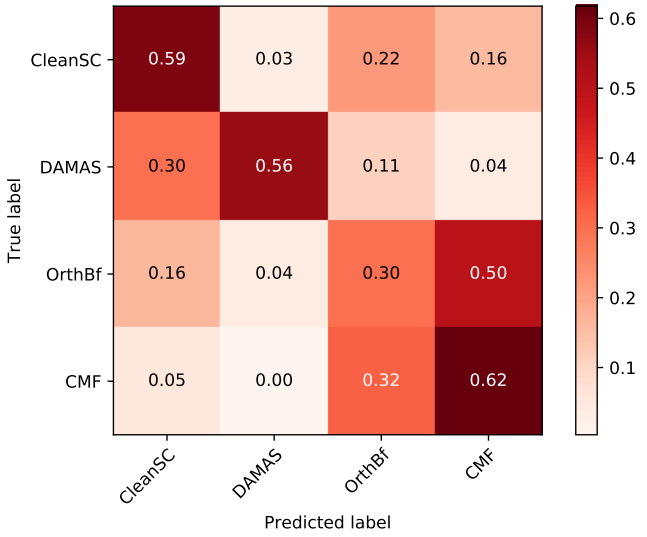

Abbildung 7: Verwechslungsmatrix zwischen Ausgang des Modells (Predicted Labels) und Targets des Datensatzes (True Labels) für Validierungsdaten und Trainingsepoche 6000.

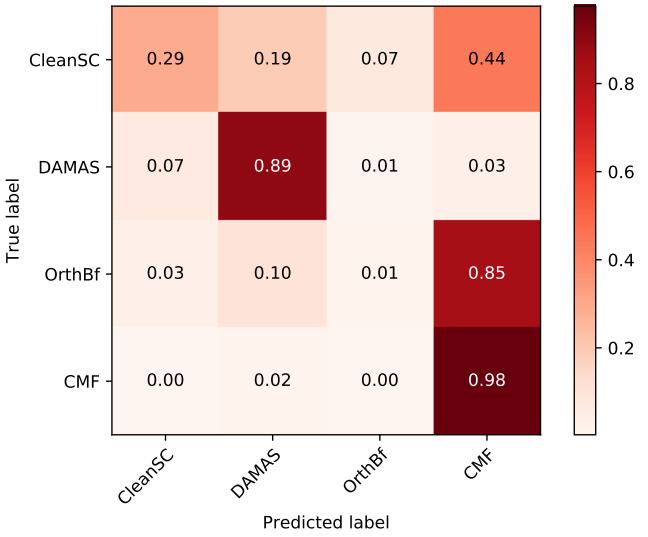

Abbildung 8: Verwechslungsmatrix zwischen Ausgang des Modells (Predicted Labels) und Targets des Datensatzes (True Labels) für Validierungsdaten und Trainingsepoche 1000.

Zusammenfassung

Die Arbeit beschäftigt sich mit der Aufgabe, bereits vor der Entfaltung einer Quellkarte einen passendes Mikrofonarrayverfahren auszuwählen. Dazu wurde ein Convolutional Neural Network implementiert und anhand von synthetischen Daten trainiert. Dieses bekommt als Eingangsdaten Quellkarten aus konventionellem Beamforming und liefert als Ausgabe eine Klassifikation der Mikrofonarrayverfahren. Die Genauigkeit die anhand der Trainingsdaten erzieht werden konnte liegt bei 92%. Die Genauigkeit beim Testen mit unbekannten Daten erreicht einen Wert von 60%.

Literatur

[1] Gert Herold und Ennes Sarradj. “Performance analysis of microphone array methods”. In: *Journal of Sound and Vibration* 401 (2017), S. 152–168. DOI: 10.1016/j.jsv.2017.04.030.

[2] Thomas F. Brooks und William M Humphreys. “A deconvolution approach for the mapping of acoustic sources (DAMAS) determined from phased microphone array”. In: *Journal of Sound and Vibration* 294.4-5 (2006), S. 856–879. DOI: 10.1016/j.jsv.2005.12.046.

[3] Tarik Yardibi, Jian Li, Petre Stoica und Louis N Cattafesta. “Sparsity constrained deconvolution approaches for acoustic source mapping.” In: *The Journal of the Acoustical Society of America* 123.5 (2008), S. 2631–2642. DOI: 10.1121/1.2896754.

[4] Ennes Sarradj. “A fast signal subspace approach for the determination of absolute levels from phased microphone array measurements”. In: *Journal of Sound and Vibration* 329.9 (2010), S. 1553–1569. DOI: 10.1016/j.jsv.2009.11.009.

[5] Peter Sijtsma. “CLEAN based on spatial source coherence”. In: *International Journal of Aeroacoustics* 6 (2007), S. 357–374. DOI: 10.1260/147547207783359459.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. “Identity mappings in deep residual networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9908 LNCS (2016), S. 630–645. ISSN: 16113349. DOI: 10.1007/978-3-319-46493-0_38. eprint: 1603.05027.

[8] Martín Abadi, Paul Barham, Jianmin Chen u. a. “TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16)* (2016), S. 265–284. DOI: 10.1038/nn.3331. eprint: 1605.08695.

[9] Jimmy Ba Diederik P. Kingma. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference for Learning Representations, San Diego, 2015*. 2015. ISBN: 9781509055449. DOI: 10.1109/ICCE.2017.7889386. eprint: 1412.6980v9.