# The Effect of Data Augmentation on Singing Voice Separation in Musical Arrangements Using a Deep U-net Convolutional Neural Network

**Lukas Frösslund**
lukasfro@kth.se

**Valdemar Gezelius**
vgez@kth.se

## Abstract

The task of Musical Source Separation, the isolation of one or more individual tracks in a polyphonic musical arrangement, is a popular undertaking in the field of Music Information Retrieval (MIR). In recent years, deep neural network architectures have dominated the field. A requirement for deep architectures to generalize well is the existence of large datasets, which has been a concern for the source separation task. In this study, we investigate the effect of data augmentation to alleviate the need for large datasets. Related works have shown promising initial results for vocal source separation, and in this study we build on this by comparing different methods of data augmentation and their effect on the quality of separation of vocal sources from the remaining arrangement. We use the U-net architecture, a network constructed to perform accurately despite few available training samples. Experimental results demonstrates a small, although not very significant, increase in performance when using data augmentation.

## 1 Introduction

The field of Music Information Retrieval (MIR) studies the many dimensions, or facets, of music [1]. One such dimension is the melodic dimension, typically conveyed by a main melodic line assigned to the singing voice and an accompaniment consisting of various monophonic instruments, combining to produce polyphonic musical works [2]. The process of separating a polyphonic musical mixture into two or more isolated tracks is known as audio source separation, a task that has been a popular topic in signal processing for several decades [3]. The problem can also act as an intermediary task in other domains, such as automatic speech recognition [4]. The case of separating the lead vocal source from the accompaniment in polyphonic music is of particular interest because of related tasks in MIR such as singer identification and lyric transcription, as well as a large commercial interest by way of the karaoke industry [5][6].

Many techniques have been proposed and implemented in pursuit of successful singing voice separation (or singing voice isolation) in polyphonic music [7]. Up until the last decade or so, techniques utilizing non-negative matrix factorizations were the most prominent and successful [8][9]. In recent past, deep neural network (DNN) approaches have emanated as a promising option to the traditional techniques. The architecture of the deep networks have varied, including recurrent neural networks (RNN) [10], multi-layer perceptrons (MLP) [11] and convolutional neural networks (CNN) [4][12]. While the exact performance comparison between these architectures is beyond the scope of this study, CNNs has the advantage of exploiting small scale, local-time features in the data, and it requires less memory than the fully connected network alternatives.

CNNs have proven formidable success in the task of pixel-wise semantic segmentation of image data [13][14]. In it, the network learns local features in the image represented as a two-dimensional vector of pixel intensities. One such network is the U-net architecture, originally introduced to perform semantic segmentation on biomedical image data [15]. In [5], this network architecture was proposed for the task of singing voice separation, utilizing a frequency-time, spectrogram representation of the audio by way of the short-time fourier transform (STFT) when training the model.

DNN approaches to singing voice separation generally requires a substaintial amount of training data to perform accurately and achieve a high level of robustness in the model. Moreover, musical datasets are often of small size [16]. One of the most proven and effective ways of alleviating the problem of limited data is the use of data augmentation, the various techniques to artificially expand a given dataset by methods of modifying existing data. Data augmentation has improved the level of generalization of DNN models in both image- and speech recognition tasks [17] [18]. In audio source separation, data augmentation of the audio has been used to achieve better generalization and robustness of various DNN architectures [16][19] [20] [21].

This study aims to further investigate the effects of data augmentation by exploring different audio deformation methods and their influence on the performance for the task of singing voice separation. An implementation of the U-net network architecture, similar to that of [5], will be adapted for our task. A pre-study will be conducted on an already trained model, evaluating if mel-scaled spectrograms improves the performance. The main experiment consists of an ablation study, measuring contrasting methods of data augmentation, including a set of popular, basic augmentations typically used for the task of source separation [19][20]. A more advanced approach to audio deformation, presented in [22], will also be evaluated.

## 2 Method

### 2.1 Data Description

The study required a dataset where each track in the dataset was paired with, at the very least, its corresponding vocal stem, since we needed clean voice stems to represent the ground-truth for our segmenting task. We considered a few different datasets (the mir-1k[1] dataset and the DSD100[2] dataset) but decided on using the MUSDB18 [23] dataset containing 150 full-length tracks (100 tracks which are taken from the DSD100 dataset), approximately 10 hours of music from a wide variety of genres.

Each track is represented as a dictionary containing isolated drums, bass, other instruments and vocals, as well as a mixture of said stems. All signals are stereophonic and encoded at 44.1kHz.

### 2.2 Data Preprocessing and Preparation

To be able to use the U-Net architecture [15], we had to convert the wave signal from the time domain into a frequency domain representation. To convert the data we performed the following steps:

1. divided each track in the dataset into frames of with window length 1024 and window shift 512
2. applied a Hann window to each frame to reduce spectral leakage and prepare each frame for the Fourier transform
3. calculated the frequency spectrum using a 513-point fast Fourier transform on each frame
4. squared the modulus of result from the fast Fourier transform to get the power spectrum
5. applied 128 triangular filters on the Mel-scale to the power spectrum to extract frequencies of each frame and and converted each frame to its logarithmic values

The short-time Fourier transform (STFT) representation of our dataset was obtained at step 3 and the log-Mel-spectrum representation of our dataset was obtained at step 5. The reason we did not fully convert our dataset into Mel frequency cepstral coefficients (MFCCs) is due to the fact that deep CNN networks has been proven to perform good with correlated features [24]. The added decorrelation process of the discrete cosine transform (DCT) is not a necessary step and will only add extra processing on our data.

[1] https://sites.google.com/site/unvoicedsoundseparation/mir-1k

[2] https://sigsep.github.io/datasets/dsd100.html

Preprocessing steps 1-5 was performed on the mixture tracks containing all stems, as well as their corresponding masks containing the vocal information. The vocal masks then had to be converted to binary masks to be able to function as ground-truth in our loss calculations. This was done by pixels-wise comparing the vocal spectrogram with a spectrogram containing all instrumental stems. If the pixel information in the vocal spectrogram was higher than in the instrumental spectrogram, the ground-truth spectrogram was set to 1, otherwise 0.

The conversion from the time domain into frequency domain representations was done using the python library libROSA[3]. This library was also used in reverting back to the time domain using the Griffin-Lim algorithm, see Figure 4.

STFTs and log-Mel-spectrum are widely used time-frequency representations [5][10][12][25] and in Table 1 we summarized the results of a pre-study conducted in 3.3, with the purpose of deciding which of the two that was most suited for this study.

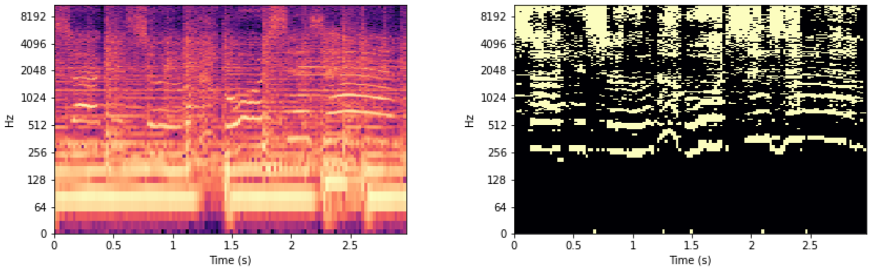

Figure 1: STFT time-frequency representation. **Left:** Mixture magnitude spectrogram, our training sample. **Right:** Binary vocal mask, our ground-truth label.

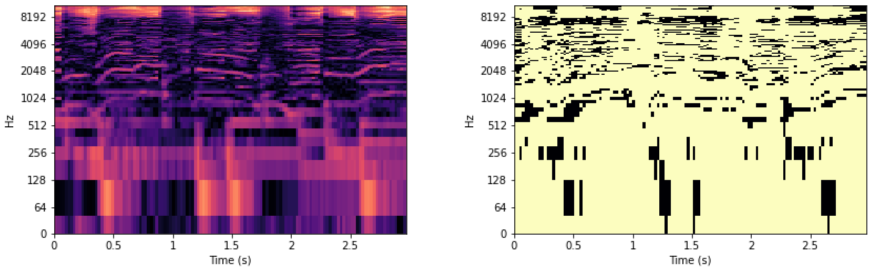

Figure 2: Log-Mel time-frequency representation. **Left:** Mixture magnitude spectrogram, our training sample. **Right:** Binary vocal mask, our ground-truth label.

### 2.3 Data Augmentation

With only few available training samples, data augmentation can act as an essential component in achieving a robust network model that can generalize effectively. In the sphere of MIR and more specifically audio source separation, several augmentation methods have shown promising results on small datasets [19][20]. Pitch Shifting and Time Stretching, consistently shown to be two of the most encouraging and high performing deformations, constitutes our set of basic augmentations.

The multi-track format of the chosen MUSDB18 dataset allowed for more advanced methods of data augmentation. One such method, presented in [22], performed random amplitude scaling on each individual stem of the accompaniment, from a uniform distribution [0.01, 1]. This study aims to further investigate the usefulness of random amplitude scaling on individual stems.

[3] https://librosa.github.io/librosa/index.html

## 2.4 Network Architecture

We decided on using the U-Net architecture [15] for this study since it is proven in the task of pixel-wise semantic segmentation of image data [13][14]. The U-Net was designed to aid in segmentation of medical images, but as has been shown in [5] it could be used in the field of separating a polyphonic musical mixture, in context of this study, the task of singing voice separation from a polyphonic mix of several musical stems.

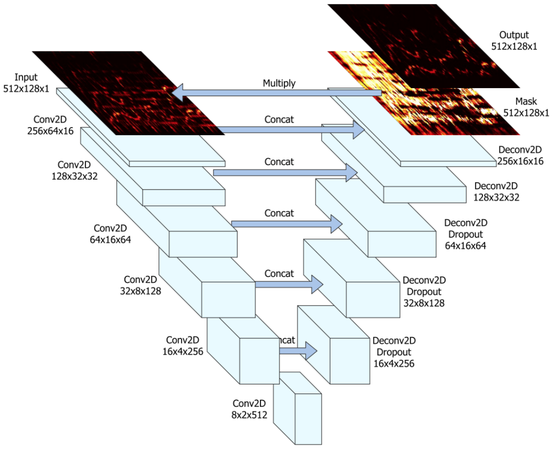

Figure 3: Modified U-Net architecture, (*image from* [5])

We decided to modify the original U-Net architecture to better suit our study, inspiration was drawn from [5] and the structure is shown in Figure 1. The architecture is built up of a contracting path which works as the encoder, and an expansive path which works as the decoder.

The contracting path is made up of blocks a convolutional layer with a 5x5 kernel and a stride of 2. Each convolutional layer is followed by batch normalization and a leaky rectified linear unit (ReLU) as activation function with leakiness 0.2. The feature maps used in the convolution layer are doubled for each time we down-sample in model (we start with 16 feature maps) [5].

The expansive path has similarities with the encoder path, but uses deconvolution with stride 2 and kernel size 5x5, batch normalization and a regular ReLU as activation function. The number of feature maps is halved for each layer and corresponding layers from the contracting path is concatenated for each deconvolution. A 50% dropout is applied for the first 3 layers of the decoder path [5].

The final layer a 1x1 convolution is used to map each feature vector to the desired number of classes and in doing so, generate the semantic segmentation which is then multiplied with the spectrogram of the mixture (containing all stems in the current song) to generate our prediction output [5].

In the U-Net architecture, regularization is implemented by inserting dropout-layers between convolution-layers. Regularization methods are an important part of any deep learning architecture to make sure that the model does not over-fit the learned parameters to the training data and in doing so, does not generalize well to unseen data. Dropout in CNNs implies that we choose ignore randomly selected units of our layer so that this unit has no impact on the next step of the network. The number of added dropout layers varies between implementations. We implemented dropout layers in the expansive path.

## 2.5 Loss function

The task of segmenting an image means that every single pixel is given a class. In our ground-truth mask, a pixel with the value 1 is classified to be part of the vocal stem, and pixels with the value 0 is classified to not be part of the vocal stem (hence belong to the musical background). To calculate the loss on how well our model predicted the values, we used binary cross-entropy loss. Below follows the formula for binary cross-entropy:

$$\mathcal{L}\,(y,\hat{y}) = -\frac{1}{N}\sum_{i=0}^{N}(y_i * log(\hat{y_i})) + (1-y_i) * log(1-\hat{y_i}) \tag{1}$$

where $y$ is the label, $\hat{y}$ is the predicted value and $N$ is the amount of pixels in the image. We see that when $y_i$ equals 1, we add $log(\hat{y_i})$ to the loss, a loss that will be big when $\hat{y_i}$ is close to 0 but decrease when $\hat{y_i}$ gets closer to 1. The same reasoning holds true when $y_i$ equals 0, then $log(1-\hat{y_i})$ is added to the loss. This loss is big when $\hat{y_i}$ is close to 1 but decreases when $\hat{y_i}$ gets closer to 0 (the ground-truth for the $i$-th pixel).

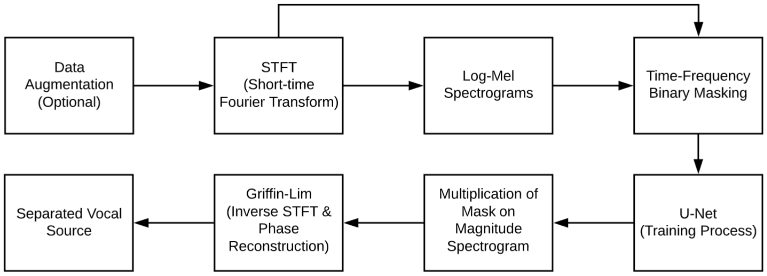
Data Augmentation (Optional)
STFT (Short-time Fourier Transform)
Log-Mel Spectrograms
Time-Frequency Binary Masking
U-Net (Training Process)
Multiplication of Mask on Magnitude Spectrogram
Griffin-Lim (Inverse STFT & Phase Reconstruction)
Separated Vocal Source

Figure 4: Data flow

## 3 Experiments

### 3.1 Implementation Details

Our network was trained on a single Nvidia Tesla T4 GPU, with the binary crossentropy loss function and the Adam [26] optimizer. Parameters of the optimizer incorporated an initial learning rate of $1e^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10e^{-8}$ and a learning rate decay of 0. The U-net model was implemented in Tensorflow and Keras, and was trained for 100 epochs with a batch size of 32. Early stopping was used with a patience set to 10, and the model was validated after each epoch. A model checkpointer was introduced to save the best model from the training process, to then use that same model for the evaluation process on the test data.

### 3.2 Evaluation Metrics

For evaluation, two of the measurements proposed by [27] were used, Source-to-distortion Ratio (SDR) and Sources-to-artifacts Ratio (SAR). We can categorize the distortions of our output from our vocal source separation network as interference and artifacts. SAR measures the network output in regards of the amount of artifacts produced in the processing stages of the network. Interference refers to remaining traces of other sources on the separated source. This and SAR are combined in SDR which can be viewed as an overall measurement of the quality of the source separation algorithm [28]. Both SDR and SAR is represented by logarithmic values and test accuracy is represented by a number between 0 and 1. The evaluation metrics were calculated using the museval toolkit[4].

### 3.3 Deciding on time-frequency representation

In 2.2 we decided on two different time-frequency representations of our dataset, short-time Fourier transform and the log-Mel-spectrum. To see which representation that was most suited for the study. The results are summarized in Table 1. We saw that using STFTs made the model perform better, since its SDR was higher than for log-Mel-spectrums, hence STFTs was used for our main experiments.

[4] https://github.com/sigsep/sigsep-mus-eval

### 3.4 Ablation Study

We conducted an ablation study where we tested 3 different models and evaluated them in terms of their SDR- and SAR score, as described in 3.2, as well as their test accuracy. Each model was trained with the same hyper-parameter values, stated in 3.1, but with different augmentations to the input data. As described in 2.3 we identified two different augmentation methods, basic augmentations and random amplitude scaling augmentations. We trained models 1-3 with, no augmentation, basic augmentation and basic augmentation together with random amplitude scaling augmentations, respectively. The results of the ablation study are summarized in table 2.

## 4 Results

Table 1: Pre-study, time-frequency representations

| Representation | SDR | SAR | Test accuracy |
|---|---|---|---|
| Short-time Fourier transform | -3.109 | -15.992 | 0.759 |
| log-Mel-spectrum | -20.732 | -13.912 | 0.749 |

We see in Table 1 that STFTs perform better than log-Mel-spectrums regarding SDR and test accuracy.

Table 2: Ablation study

| Model | Augmentation | SDR | SAR | Test accuracy |
|---|---|---|---|---|
| 1 | No augmentation | -3.077 | -16.448 | 0.748 |
| 2 | Basic augmentation | -2.892 | -16.290 | 0.755 |
| 3 | Basic augmentation + random amplitude scaling | -2.787 | -15.755 | 0.753 |

We see in Table 2 that basic augmentation together with random amplitude scaling performs best in regards to SDR and SAR, and only basic augmentation receives the highest test accuray.

## 5 Discussion and Conclusion

Augmenting input data seem to yield a small performance boost, but if we look at Table 2, we see that there is no significant difference between using augmenting techniques or not. This could possibly be because of the rather subtle changes to the spectrograms when using augmentation since we modify the signal, not the resulting spectrogram that the CNN uses. It could also be that these models generalize better to unseen data, since we only used augmented data in the training set.

In addition to the quantitative evaluation, a more qualitative evaluation, where we essentially listen to the audio of our models predictions, was also completed. Due to the subjective nature of this evaluation, we did not include this assessment in the section on experiments. Nevertheless, we still felt that the models qualitative performance carried value in our personal appraisal towards our models. What we could deduce was that our model was very able to separate the vocal source from a full mixture, despite a low SDR- and SAR score.

From our pre-study we saw that STFTs performed better than using log-Mel-spectrum representations of our input data. If we look at Figure 2 we can see a hint of why, as the ground-truth mask for the log-Mel-spectrums are almost all of pixel-value 1. This could be because of the lower resolution with using 128 filters compared to 513 in STFT or because perhaps the filters do not cover the vocal frequency spectra with enough accuracy.

For future work, we remain positive that more research into the effect of data augmentation can yield progress in the field of musical source separation. New frameworks and tools for more coherent evaluation metrics would also be very helpful for future progress in the field. We also believe that more research could be made in investigating the possibility of integrating a structured, qualitative framework for measuring quality of separation, as questions can be raised against the sole use of quantitative metrics when assessing specifically musical arrangements.

## A Peer review feedback

### 1. Relevance for the learning outcomes

Both reviewers were in consensus that our report had met most of the learning outcomes. One reviewer suggested an additional feature extraction method, MFCCs, for the initial decision of the time-frequency representation. We did not incorporate this suggestion into practice, because we wanted representations with correlated data, utilizing the power of our deep architecture.

### 2. Literature Study

One reviewer stated that some of our references could be excluded, and suggestions were made on specific literature. After examining our complete list of references we decided not to incorporate this suggestion, and we kept the suggested references as we did not believe them to be redundant. Another reviewer suggested us to modify our footnotes and turn them into full references. We partly put this suggestion into practice, for the resources where we could find proper references.

### 3. Novelty/Originality

The reviewers agreed that our project carries some level of novelty, but that it is also inspired by related works, which we agree with. One reviewer reiterated a suggestion to us to add an additional experiment including MFCCs, but as explained earlier we did not incorporate that suggestion.

### 4. Correctness

One reviewer pointed out a divergence in our report from the NIPS template, which we have now adjusted.

### 5. Clarity of presentation

One reviewer suggested that a certain section of the report could benefit from more visuals. We partly incorporated this suggestion by adding both a data flow chart over the complete process and graphs displaying the considered time-frequency representations of our audio data. The same reviewer also suggested some changes to the introductory section of the report. We closely examined the specific parts that were pointed out, but decided not to change the section as we did not agree with the reviewers assessment.

## References

[1] J. S. Downie, “Music information retrieval,” *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 295–340, 2003.

[2] N. Orio, “Music retrieval: A tutorial and review,” *Found. Trends Inf. Retr.*, vol. 1, p. 1–96, Jan. 2006.

[3] Z. Rafii, A. Liutkus, F. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, “An overview of lead and accompaniment separation in music,” *CoRR*, vol. abs/1804.08300, 2018.

[4] P. Chandna, M. Miron, J. Janer, and E. Gómez, “Monoaural audio source separation using deep convolutional neural networks,” vol. 10169, pp. 258–266, 02 2017.

[5] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, “Singing voice separation with deep u-net convolutional networks.” October 2017.

[6] A. J. R. Simpson, G. Roma, and M. D. Plumbley, “Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network,” *CoRR*, vol. abs/1504.04658, 2015.

[7] Z. Rafii, A. Liutkus, F. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, “An overview of lead and accompaniment separation in music,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, 2018.

[8] J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David, "Main instrument separation from stereophonic audio signals using a source/filter model," 01 2009.

[9] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," in *IN: PROC. ISMIR2005. (2005) 337–344*, pp. 337–344, 2005.

[10] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," pp. 477–482, Jan. 2014. 15th International Society for Music Information Retrieval Conference, ISMIR 2014 ; Conference date: 27-10-2014 Through 31-10-2014.

[11] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, pp. 1652–1664, June 2016.

[12] A. J. R. Simpson, "Probabilistic binary-mask cocktail-party source separation in a convolutional deep neural network," *CoRR*, vol. abs/1503.06962, 2015.

[13] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[16] S. Bhardwaj, "Audio Data Augmentation with respect to Musical Instrument Recognition," Nov. 2017.

[17] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, vol. 117, 2013.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[19] M. Miron, J. Janer Mestres, and E. Gómez Gutiérrez, "Generating data to train convolutional neural networks for classical music source separation," in *Lokki T, Pätynen J, Välimäki V, editors. Proceedings of the 14th Sound and Music Computing Conference; 2017 Jul 5-8; Espoo, Finland. Aalto: Aalto University; 2017. p. 227-33.*, Aalto University, 2017.

[20] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 261–265, IEEE, 2017.

[21] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[22] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2135–2139, IEEE, 2015.

[23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," dec 2017.

[24] R. A. Solovyev, M. Vakhrushev, A. Radionov, V. Aliev, and A. A. Shvets, "Deep learning approaches for understanding simple speech commands," *CoRR*, vol. abs/1810.02364, 2018.

[25] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, p. 279–283, Mar 2017.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[27] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] S. Venkataramani, R. Higa, and P. Smaragdis, "Performance based cost functions for end-to-end speech separation," 2018.