CSCI 566 - Final Project Report

Theodore Ho
teho@usc.edu

Shuye Huang
shuye.huang@usc.edu

Dongwook Kim
dkim2378@usc.edu

Abstract

Musical Source Separation(MSS) is the process of separating stems from a track. For example, given a music waveform with multiple instruments/audio sources, MSS outputs just the waveform for one source such as the track's vocals. MSS has been actively researched for decades with numerous models and different target sources. However, training data for MSS is difficult to obtain as it requires tracks with pre-separated sources. Many of the highest preforming models only train on the MUSDB-18 dataset (Rafii et al. [2019]) which only has 150 tracks. We aim to explore data generation and augmentation techniques to create a larger training dataset and investigate the performance impact on existing models. Specifically, we include conventional sound effects (delay, high and low pass filter) as customized augmentation methods and apply on MUSDB-18 dataset and perform vocal-instrument separation. We use rebased ResNeSt (Zhang et al. [2022]) as our experiment architecture, apply different augmentations and train the model from scratch. We observe Signal-To-Distortion Ratio (SDR, evaluation metrics for MSS) of 5.98, with a 7% improvement from the reference model, indicating our augmentation techniques are valid.

1 Introduction

Listening to music has become a daily basis lifestyle since the invention of the internet as it provided easy access to a wide variety of music libraries. Alongside the practice of remixing came, where one would mix different tracks of other songs to produce a new song. This is quite straightforward, given that one has access to individual instrumental tracks of the said songs. However, it is more often the case that songs are published as mixed single-track versions hence the attempts to separate music sources from the singular tracks have been made and are called Musical Source Separation(MSS). Key challenges in MSS root from the fact that the final recording mixes not only contain various instruments, but are additionally filtered and non-linearly mastered. Additionally, in stereo mixes, the sources may move or alternate between channels thus making MSS harder. Solving MSS can benefit a wide range of individuals, from content creators such as DJs and samplers, to listeners with an interest to listen to instrumental (vocal-removed) versions of their favorite songs.

2 Related Work

Deep learning approach for MSS has been developed rapidly in recent years, mainly applied on temporal domain (i.e., waveform) or temporal-frequency domain (i.e., spectrogram). Multiple architectures have been proposed, for instance, waveform U-Net (Défossez et al. [2019]), domain

Preprint. Under review.

densely connected dilated CNN on spectrogram (Takahashi and Mitsufuji [2020]), spectrogram RNN (Luo and Yu [2023]), local-attention-based dual U-Net on hybrid input (Défossez [2021]), etc. The growth of architecture's sophistication results in requirement of larger dataset for model training.

Unfortunately, limited open dataset of source separated tracks are available, in which case data augmentation is of importance. Methods to enrich an audio dataset include temporal or frequency domain masking or warping (Park et al. [2019]), pitch-shifting and spectral transformation (Cohen-Hadria et al. [2019]), remixing channels in batch (Défossez et al. [2019]). It is noticeable that these approaches may not generate realistic music segments, with potential caveats such as beat miss alignment, tempo distortion and break down of melodic structures, leading to possible worse model training. Some previous work has discussed this issue (e.g., Chiu et al. [2021], Défossez [2021]). We here want to systematically evaluate the goodness of these augmentation strategies and their potential implications for different model architectures. While many data augmentation techniques have already been explored and used to train the SOTA models, our project aims to explore new data augmentation techniques and measure them against existing techniques described in previous papers.

Performance is measured using SDR (signal distortion ratio) between the true source and the models predicted source. To standardize testing of model performance, researchers average the SDR score for all test examples in the MUSDB-18 dataset to compare performance between models.

3 Dataset

MUSDB-18 (Rafii et al. [2019]) is a widely used benchmark dataset for evaluating music source separation algorithms. It contains 150 tracks, including professionally mixed songs in various genres, such as pop, rock, and jazz. Each track is provided in stereo and separated into four distinct stems: vocals, drums, bass, and other instruments, enabling researchers to assess the quality of source separation for each component. In this study we focus on vocal separation, hence other stems are mixed. The dataset is split into a training set of 100 tracks and a test set of 50 tracks, making it suitable for supervised learning and benchmarking. Notably, several prominent MSS studies have utilized MUSDB-18 as their benchmark, including Open-Unmix (Stöter et al. [2019]) and Demucs (Défossez et al. [2019]), both of which demonstrated state-of-the-art performance on this dataset. Additionally, MUSDB18 is integral to the Signal Separation Evaluation Campaign (SiSEC), which standardizes the evaluation of MSS methods using established metrics such as SDR.

4 Methods

We plan to employ data augmentation techniques used as audio producer effects such as EQ and delay to compare with existing data augmentation techniques. We will select models, train them on the MUSDB dataset with no augmentations, existing augmentation techniques, and our augmentation techniques separately. We will use the SDR score of the models trained on different datasets to compare the performance improvement across different augmentation techniques. For this project, we will only focus on the vocal separation stem.

4.1 Model

Many of the models for MSS are very large and require lots of GPU power. For example, HDemucs was trained on 8 16-GB V100 GPUS (Défossez [2021]). However, we were able to achieve reasonable SDR results with a light-weight U-Net mode. Our model is a U-Net architecture from the 68M paramater ResNeSt (Zhang et al. [2022]) encoder provided in (Iakubovskii [2019]). Following the specifications of the U-Net model in (Solovyev et al. [2023]), our model takes in a 6 second audio segment STFT spectrogram with a hop length and window size of 512 and 8192 respectively. This results in a 4096x512 input and output dimension for our model as audio spectrograms.

4.2 Data Augmentations

We used previous data augmentation techniques published in Défossez [2021] for reference against our own musical effects inspired data augmentation techniques.

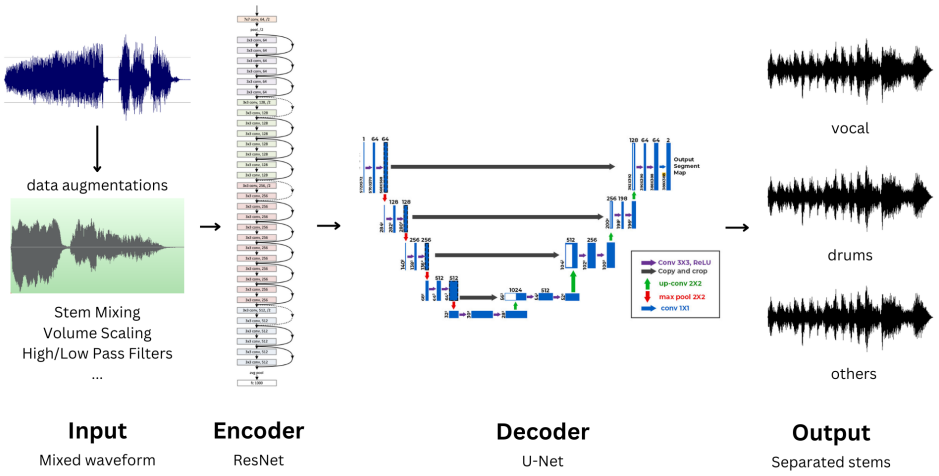

Figure 1: Overview of the ResNet-UNet Model Architecture

Stem Mixing (reference): Mix stems from other tracks in the dataset into the current mixture. For example, each training sample is a pair (vocals, mix). If stem mixing is applied, a random non-vocal stem from another training sample will be added to the mix.

Volume (reference): Scale the vocals and mix by a random scalar sampled uniformly from the range (.5 to 1.5)

High Pass Filter (Proposed): Apply a Pedalbord high-pass filter with a cutoff frequency of 50Hz.

Low Pass Filter (Proposed): Apply a Pedalbord low-pass with a cutoff frequency of 15kHz.

Delay (Proposed): Apply a Pedalbord half second delay.

All data augmentation techniques were implemented as a PyTorch data-loader. If an augmentation is enabled for the model instance during training, each training sample has the probability of the data augmentation applied. We used the same loudness and stem mixing probabilities used in Défossez [2021]. The probabilities for our proposed augmentations were the probabilities that produced the best results during training.

Augmentation	Probability per Sample
Loudness	.1
Stem Mixing	.2
High-Pass Filter	.05
Low-Pass Filter	.05
Delay	.05

Table 1: Augmentation Probabilities

4.3 Training Procedure

We trained five separate instances of the same model using different data augmentation techniques. No augmentations for reference, loudness, stem mixing, loudness with stem mixing, and finally our proposed method, high/low-pass filter with delay.

We used an AdamW optimizer with a $5e^{-5}$ learning rate and a batch size of 18. A batch size of 18 was the largest we could use for our available hardware; 40GB A100GPU. We trained each model instance for 15,000 steps. After each 1000 step we tested the models performance on the test set. Training each model took around 5 hours on an 40GB A100 GPU.

4.4 Evaluation

The metric we are using for evaluation is Signal to Distortion Ratio (SDR). SDR is the standard metric used for evaluating MSS models and measures the magnitude of error scaled by source magnitude. The Signal Distortion Ratio (SDR) is given by $10 \cdot \log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{target-pred}\|^2}\right)$.

To evaluate our model, we take the average SDR of the test set at each 1000 steps and report the best average of the model.

5 Results

Augmentation Technique	SDR Vocals
No Augmentations (Reference Model)	5.7
Stem Mixing	5.57
Volume Scaling	5.9
Volume Scaling + Stem Mixing	6.06
High/Low-Pass Filter + Delay	5.98

Table 2: Performance of Data Augmentation Techniques on MUSDB18 Test Set

6 Discussion

Compared to the model trained with no augmentations as a reference, the conventional augmentation techniques stem mixing and volume scaling resulted in the SDR increase of -2.28% and 3.51% accordingly, when applied individually. When applied in combination, the model produced an SDR of 6.06, increased by 6.91% from the reference model. This suggests that the augmentation techniques may not be independent, and output better performances when used in combinations.

Meanwhile, our proposed augmentations, the high/low pass filter in combination with delays, resulted in an SDR of 5.98, increased by 4.91% from the reference model which was on par with the model augmented with the conventional techniques in combination. Our model failed to outperform the Meta Demucs model, which is understandable as Meta's model uses a different architecture and computing resources.

Hence we have to think about the wide variety of models available. Although we chose one model to work with for comparisons, it is possible that the techniques and combinations of techniques may affect the model performances differently, depending on the architecture. Furthermore, it is possible that some augmentations work better on specific stages of the training, which was not tested in our project. For example, one augmentation may work better if applied in the early stages of training, while others may work better in the later stages.

It may also be beneficial to use K-fold validation of our data augmentation performance as this would provide greater certainty for our results. However, evaluating model performance using a different test split than the one provided by MUSDB-18 would make it impossible to compare with other published models. This is because published models use the default MUSDB-18 test split for evaluation. Furthermore, the test split is quite large including 50 of the 150 songs in the dataset. However, it still may be beneficial to use K-fold validation.

In the future, we plan on testing more data augmentation techniques such as pitch shifting and time-stretching, to further measure the effectiveness of dataset augmentation on MSS models and different combinations. The baseline results of the current data augmentation techniques shown in this report will be used for us to measure our new data augmentation techniques.

7 Conclusion

We have confirmed that our proposed data augmentation techniques, high/low pass filter and delays, can improve the performance of MSS models, to the point at which was on par with the existing techniques. We also found that the techniques work better if used in combinations. We plan on

testing more audio producer effects such as pitch shifting and time stretching as data augmentation techniques. We would also like to investigate further about the effects of the data augmentation techniques on other various models and on different stages of training.

References

C.-Y. Chiu, J. Ching, W.-Y. Hsiao, Y.-H. Chen, A. W.-Y. Su, and Y.-H. Yang. Source separation-based data augmentation for improved joint beat and downbeat tracking. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 391–395. IEEE, 2021.

A. Cohen-Hadria, A. Roebel, and G. Peeters. Improving singing voice separation using deep u-net and wave-u-net with data augmentation. In *2019 27th European signal processing conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

A. Défossez. Hybrid spectrogram and waveform source separation. *arXiv preprint arXiv:2111.03600*, 2021.

A. Défossez, N. Usunier, L. Bottou, and F. Bach. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*, 2019.

P. Iakubovskii. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2019.

Y. Luo and J. Yu. Music source separation with band-split rnn. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1893–1901, 2023.

D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner. MUSDB18-HQ - an uncompressed version of musdb18, Dec. 2019. URL `https://doi.org/10.5281/zenodo.3338373`.

R. Solovyev, A. Stempkovskiy, and T. Habruseva. Benchmarks and leaderboards for sound demixing tasks, 2023.

F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji. Open-unmix-a reference implementation for music source separation. *Journal of Open Source Software*, 4(41):1667, 2019.

N. Takahashi and Y. Mitsufuji. D3net: Densely connected multidilated densenet for music source separation. *arXiv preprint arXiv:2010.01733*, 2020.

H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2736–2746, 2022.