SCSU Journal of Student Scholarship

Volume 1
Issue 2 *Special Issue 2: 10th Annual Minnesota State Conference of Undergraduate Scholarly and Creative Activity*

Article 3

June 2021

Deep Learning Based Music Source Separation

Ryan Henning
Winona State University

Abdullah Choudhry
Winona State University

Ming Ma
Winona State University

Introduction

This work aims to cope with the problem of separating an audio source into several audio tracks. Three architectures based on existing models, with implementations using 1D and 2D convolution, can separate stems from any song out of the full mix, where a stem is the separated track of a full mixed audio source. This is done using existing songs with access to the separated stems. Analysis of different music source separation architectures opens doors for artists and musicians to sample or create new content from old tracks. Comparisons between the architectures and their domains also allow us to see which architecture is effective as well as which domain yields the best results. One architecture is improved by implementing a refinement U-Net (i.e., an encoder/decoder convolutional neural network). This problem plays an important role in Music Information Retrieval (MIR), allowing researchers to analyze vocal lyrics, transcribe music, classify music genres, and extract vocal melodies [1]. Figure 1 shows a U-Net architecture for the application of singing voice separation.

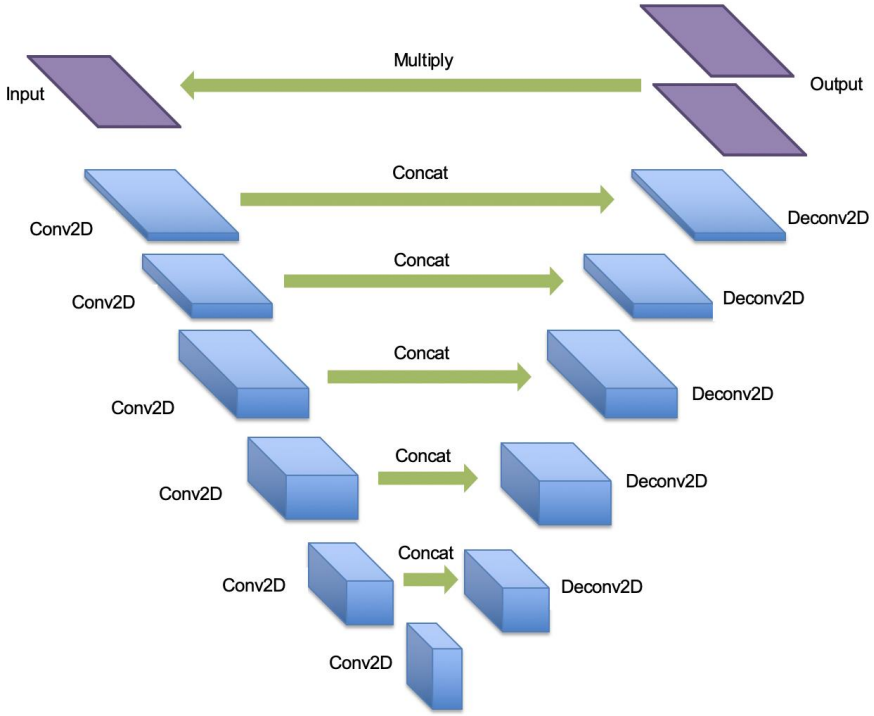

Figure 1. U-Net architecture for singing voice separation. The input is music audio and the output are vocal and instrumental components.

Related Work

Researchers have proposed several solutions for this problem. These solutions all include some variation of a U-Net. The U-Net is used for estimating a soft-mask for each source/stem, it was originally developed for medical image segmentation, however this model has also been proven effective in music source separation. In 2019 the music streaming company Deezer released Spleeter, which is an open source music separation tool [1]. Facebook Research also released their own tool in early 2020 which was called Demucs [2]. One more solution is the f90 implementation of the Wave-U-Net. The Wave-U-Net is an adaption of the U-Net to the one-dimensional time domain to perform end-to-end audio source separation [3].

Methods

I. Deezer's Spleeter

Spleeter uses a spectrogram based approach for music source separation and is based on Tensorflow, allowing training and inference to be run on a central processing unit (CPU) or the graphics processing unit (GPU). It was designed to be a state-of-the-art model for music separation. The method used in Spleeter is a U-Net model with 12 layers, six for the encoder and six for the decoder. The model uses a kernel size (i.e., the size of convolutional filter) of five and a stride size (i.e., the step size of convolution kernel movement) of two, is trained using a first-order gradient-based optimization algorithm called Adam optimizer, and uses Signal Distortion Ratio (SDR), Signal to Artifacts Ratio (SAR), Signal to Interference Ratio (SIR), and Source Image to Spatial Distortion Ratio (ISR) as metrics for evaluation [1].

Deezer's testing showed that Spleeter is able to separate a song into four stems with 100 seconds of stereo audio in less than one second, using an RTX 2080 GPU and a double Intel Xeon Gold 6134 CPU at 3.20GHz. Although the model is not trained or optimized using the standard benchmark dataset called musdb18, when comparing it to another state-of-the-art model Open-Unmix, Spleeter proves to be a competitive alternative for most metrics [1]. Refer to [1, 4] for more details.

II. Facebook's Demucs

Demucs is a waveform-to-waveform model based on the Conv-Tasnet. Demucs is based on Pytorch, allowing it to also be run on either the CPU or GPU. It uses a U-Net with four encoder layers and four decoder layers. It uses a kernel size of eight and a stride size of four, uses the average mean square error (MSE) for loss [5], and has a recommended number of 180 epochs (i.e., 180 passes of the training data through the network).

According to Facebook Research, a single batch of size 16 with 10 seconds of audio took Demucs 1.6 seconds per batch, where Open-Unmix took 0.2 seconds per batch, and Wave-U-Net takes 1.2 seconds per batch. When testing separation quality, the Conv-Tasnet has the highest SDR while Demucs beats both Open-Unmix and the Wave-U-Net. Additional testing with larger datasets showed the SDR gap between Demucs and the Conv-TasNet shrank [5, 6]. Refer to [2] for more details.

III. F90 Wave-U-Net

F90 is an implementation of the Wave-U-Net. The goal of this model is to do music sources separation in the one-dimensional time domain. This method allows for modelling phase information and it avoids fixed spectral transformations (used when designing filters). The model has implementations in both Tensorflow and Pytorch. Tensorflow is an end-to-end open source platform for machine learning which is developed by Google Brain Team. Pytorch is also an open source machine learning library developed by Facebook. The model can be run on both the CPU and GPU. This model uses 24 layers, 12 for the encoder and 12 for the decoder. It also has a kernel size of five and a stride size of two, and it is trained using the ADAM optimizer. It performs early stopping after 20 epochs of no improvement on the validation set, measured by mean squared error (MSE) loss. The final model is then fine tuned with the batch size doubled until 20 epochs without improvement in validation loss. Refer to [3] for more details.

IV. Comparison Between 1D and 2D Domains

To evaluate the effectiveness of 1D and 2D convolution we take the architectures of Spleeter, Demucs, and f90 and implement them with both 1D and 2D convolution. Spleeter (using a spectrogram-based approach) was used as the basis for the implementation of our 2D models and f90 (using a waveform-based approach) as the basis for our 1D models. To train under similar conditions we used 100,000 training steps and a batch size of four. Training on a GPU under these conditions took about six hours. These models were then evaluated upon completion of the training to compute their metrics. The result in terms of the metrics allows for comparisons to be made to determine which architectures are effective and what domain is most effective for a given architecture. From these comparisons, the most effective model among the six models can be obtained.

V. Phase Two Model

For our improved phase two model, a stack U-Net implementation was used. To create this model, the best one of the six trained models is selected for improvement. Based on the metrics, the one with the lowest score needed to be improved. To achieve the performance improvement, a post-separation processing stage is introduced. In this stage, each stem undergoes further processing to improve its quality. This stage takes the form of another U-Net, resulting in the stack U-Net model.

Experiments

I. Datasets

MUSDB18 is the standard benchmark dataset for music source separation [7]. The MUSDB18 dataset contains 150 songs. The dataset has 50 songs for testing and 100 for training. Each song is saved in a folder containing the original mix, the bass, drums, vocals, and other as .wav files (original and its four stems). Existing methods separate the original mix and then compare their separated stems with the originals from the dataset. Using this dataset, the six models were compared.

II. Evaluation Metrics

The evaluation metrics are the source-to-distortion ratio (SDR), source-to-artifacts ratio (SAR), source-to-interference ratio (SIR), image-to-spatial ratio (ISR), and an average of these ratios. The SDR is used to evaluate the total distortion (unwanted changes in the waveform of an audio) in the signal. The SAR evaluates the total amount of artifacting (accidental or unwanted sounds caused from digital altering of a sound) in the signal. The SIR evaluates the total interference (unwanted noise bleeding in from other stems) in the signal. The ISR evaluates the spatial distortion (perceived distance / how accurate the sound is to the original) in the signal. These metrics are computed using the Museval package. Museval computes these metrics by comparing the separated stem to the original. Spleeter and f90 both use this package to evaluate their models.

In the equations below, s is the clean source signal. $\hat{s}$ is the estimated source signal given by $\hat{s} = s + e_{spat} + e_{interf} + e_{artif}$ with e_{spat} the error due to spatial distortions, e_{interf} the error due to interference with other sources and e_{artif} the error due to artifacts. SDR takes all error types into account in its computation which is why this metric is the most commonly used in model

comparisons. The SDR is the most important metric. The metrics above are expressed in decibel units (symbol dB).

$$SDR = 10\log_{10}\frac{\|s\|^2}{\|e_{spat}+e_{interf}+e_{artif}\|^2} \quad (1)$$

$$ISR = 10\log_{10}\frac{\|s\|^2}{\|e_{spat}\|^2} \quad (2)$$

$$SIR = 10\log_{10}\frac{\|s+e_{spat}\|^2}{\|e_{interf}\|^2} \quad (3)$$

$$SAR = 10\log_{10}\frac{\|s+e_{spat}+e_{interf}\|^2}{\|e_{artif}\|^2} \quad (4)$$

III. Phase One Results

The ratios computed for each model allow us to make comparisons. The resulting ratios are not the state-of-the-art, but that was to be expected with only 100,000 training steps. To get results matching existing models, 1.5 million training steps should be used (which is set in the original Spleeter code), and it would take about a week to train each model. These metrics allow comparisons to be made since all six models were trained under the same conditions. Table 1 below shows the ratios computed for each of the six models.

Table 1. Comparison of computed ratios.

Dimension	Model	SDR (dB)	SAR (dB)	SIR (dB)	ISR (dB)	Average Overall (dB)	Average w/o SIR (dB)
1D	Spleeter	2.105	10.134	241.616	2.156	64.003	4.798
	Wave-U-Net	2.086	9.422	250.360	2.217	66.021	4.575
	Demucs	1.892	7.664	249.816	1.934	65.329	3.833
2D	Spleeter	0.326	-3.707	-0.877	6.769	0.628	1.129
	Wave-U-Net	1.826	-3.782	0.673	6.509	1.306	1.518
	Demucs	3.901	2.950	7.057	8.098	5.502	4.983

For models using 2D convolution, the Demucs architecture had the best ratios in every category. For models using 1D, convolution the Spleeter architecture had the highest SDR and SAR, and the f90 architecture had the highest SIR and ISR. For some ratios under the 2D models, negative scores were observed. These ratios mean that the signal power is lower than the comparison power (e.g., a negative SAR means that the artifacts overpower the original signal). The 1D models also exhibit very high SIRs. With a 1D domain, the modelled phase information allows the stems to be better distinguished. However, the other metrics are more important than the SIR. So, the average ratio score without SIR was computed and the average was not so skewed. Figure 2 shows a graph of the adjusted averages.

The 2D Demucs-based architecture has the best score. Based on this we hypothesize that our 2D Demucs-based architecture would continue to be the best model if all models were trained more. Perhaps in the future proper training (1.5 million training steps, or about one week) can be conducted to confirm this prediction.

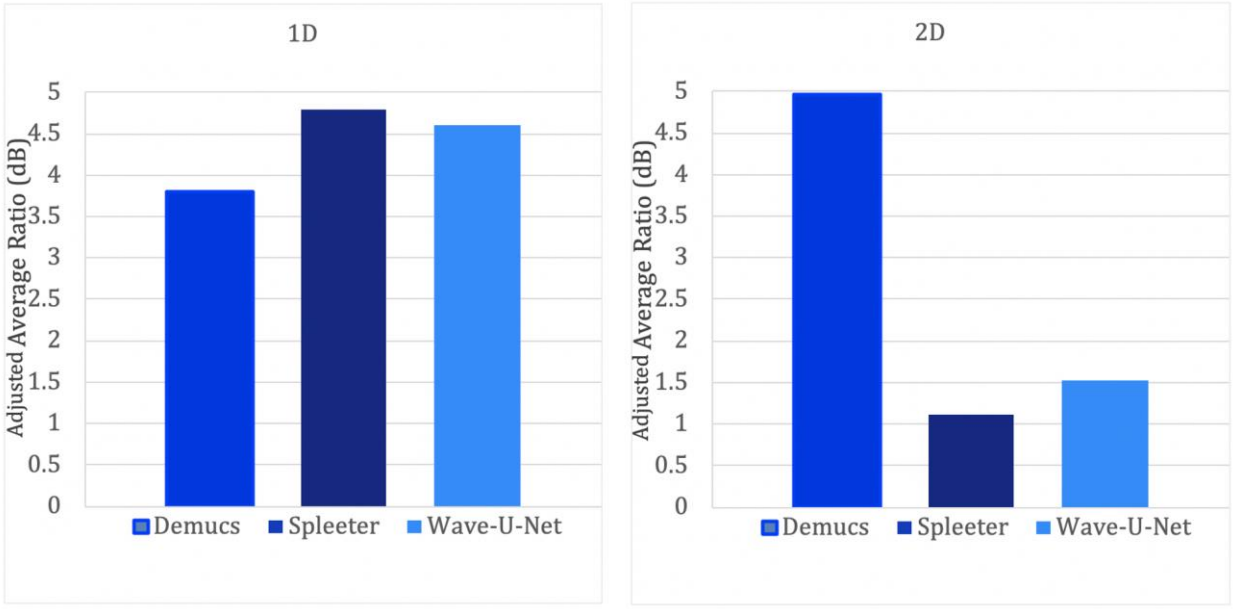

Figure 2. Graph of adjusted averages

IV. Phase Two Implementation

To implement the refinement U-Net, Spleeter was used as our main framework. From the results in Table 1, the 2D Demucs-based architecture was the best of the six, so 2D Demucs-based architecture was used as the structure for both U-Nets. The SAR was lowest for this architecture, so our goal is to reduce the artifacting to improve the quality of separated stems. To achieve the improvement, a post-separation processing phase was introduced, which filters out as much artifacting as possible for each stem. After training this model, museval was used to compute the metrics from Equations (1-4). With these metrics, this new model was compared with the models from our experiment to see if this second phase improves the quality of the stems. Training for a much longer period of time allows us to compare our model to the proposed model. Figure 3 shows the basic pipeline of our model.

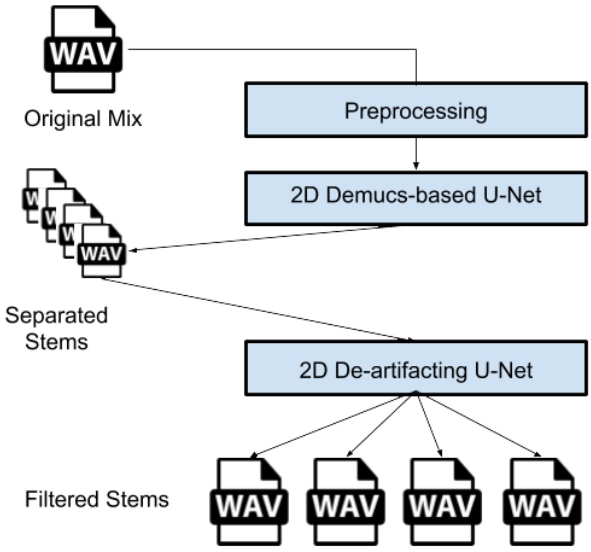

Figure 3. Stack U-Net pipeline.

V. Phase Two Results

After implementing our second U-Net, we trained it as for our previous models. Metrics were computed for comparison. Table 2 shows the results of the stack U-Net model and the 2D Demucs model.

Table 2. Comparison between Stack U-Net and 2D Demucs models.

	SDR (dB)	SAR (dB)	SIR (dB)	ISR (dB)	Average Overall (dB)	Average w/o SIR (dB)
Stack U-Net	3.927	3.149	6.630	8.103	5.452	5.060
Demucs 2D	3.901	2.950	7.057	8.098	5.502	4.983

The SAR in the stack U-Net model did improve by about 0.2 dB. The SDR improved by about 0.02 dB. The SIR dropped by about 0.4 dB, resulting in the average of the metrics for the stack U-Net model dropping by 0.5 dB. Looking at the adjusted average without SIR, it can be seen that an improvement of about 0.08 dB. Figure 4 below shows the SDR and adjusted average.

Figure 4. Graph of SDR and adjusted average.

The most important metric is SDR. It and the adjusted average show that the stack U-Net model improved a little. This model introduces a second U-Net, so there are many more weights that need to be adjusted. We only trained for 100,000 training steps, but with 1.5 million training steps, it is believed that the improvements will be much more noticeable.

Conclusion

Three different neutral network architectures extract tracks from music recordings using both 1D and 2D convolution. Six models were trained and compared on four metrics. The 2D Demucs architecture showed the best results. An architecture was built off to further enhance it. Our phase two model implemented a second U-Net to reduce artifacting in each of the stems. This new model gave a little improvement, but with more training, we believe the improvements will be more noticeable. In the future, we propose fully training a stack U-Net model to compare the results with single U-Net models.

References

1. R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam. “Spleeter: a Fast and Efficient Music Source Separation Tool with Pre-Trained Models”, *Journal of Open Source Software*, **5**(50), pp. 2154 (2020). doi:10.21105/joss.02154.
2. A. Défossez, N. Usunier, L. Bottou, and F. Bach. “Demucs: Deep Extractor for Music Sources with Extra Unlabeled Data Remixed”, *arXiv preprint arXiv*:1909.01174 (2019).

3. D. Stoller, S. Ewert, and S. Dixon. “Wave-U-Net: A Multi-scale Neural Network for End-to-end Audio Source Separation”, *arXiv preprint arXiv*:1806.03185 (2018).
4. R. Hennequin, and A. Khlif. “Spleeter”, *Deezer Research Publications*, (2020), https://research.deezer.com/projects/spleeter.html. (Accessed: 1 July 2021)
5. A. Défossez, N. Usunier, L. Bottou, and F. Bach. “Music Source Separation in the Waveform Domain”, *arXiv preprint arXiv*:1911.13254 (2019).
6. Y. Luo, and N. Mesgarani. “Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(8), pp.1256-1266. doi:10.1109/TASLP.2019.2915167.*
7. Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, and R. Bittner. “MUSDB18 Corpus for Music Separation”, (2017). doi:10.5281/zenodo.1117372