Music Source Separation with Noise

Yi-Ting Tsai
ytsai25@jhu.edu

Joshua Miller
jmill328@jhu.edu

Abstract

Music source separation is a unique task with powerful applications both in the music production and music information retrieval community. There are many commercially available applications of music source separation available with DJing as a primary motivation. However, there has been very little exploration of the effect of noise on this task. A practical problem to motivate this task would be a live music situation in which a studio quality recording does not exist, yet source separation is still desired for usage within a secondary task. Simulating a live environment provides both a practical motivation as well as novel challenges for this task. We present a comprehensive study of noise's effect on music source separation using the Spleeter toolkit. Both U-Net and Bidirectional LSTM neural architectures are explored as well as multiple variations of data augmentation to improve source separation performance, measured using standard blind source separation metrics. Ultimately, further work is needed to determine how to make music source separation most resilient to noise as data augmentation does not yield significant improvements within our experiments.

1. Introduction

Music source separation is a special instance of audio source separation involving the extraction of particular subcomponents of a musical track. The goal of this task is to extract vocals and instruments from a given musical track, a single stereo recording file. The goal of this source separation is to successfully extract individual components from a musical track to be used either in musical production or a music information retrieval based task. This work will focus on the neural approach to this problem although several alternative approaches do exist which rely exclusively on signal processing. For training a neural network for music source separation typically both a mixture file and each musical component are provided as isolated audio files. Training data usually consists of cleanly recorded individual tracks and their mix, produced in a studio. However, we would like to extend this approach to music recorded in a noisy, live environment so that music source separation is not limited to only studio quality recordings. This introduces challenges of the noise present in a live setting, mainly babble or cheering from a crowd, the additional acoustic variety introduced by a live venue, as well as the difficulty of separating vocals from crowd noise which both consist of human speech.

2. Objectives

With the goal of extending music source separation to a live or otherwise noisy environment in mind, we attempt a comparison of neural networks resilience to noise for this specific task. This extension requires placing noise into the mixture audio to later be separated. Comparisons will be primarily using the U-Net [9] and BLSTM [15] models provided with the Spleeter [8] toolkit. Additionally, in order to improve upon a given network's performance, we also investigate several data augmentation methods in order to train networks in a way which allows not only instruments and vocals to be modelled but also so that any noise present can be separated as its own unique component.

3. Background

Existing state-of-the-art methods for music source separation, as shown in the proceedings of Sisec 2018 [14], are composed of mainly U-Net and Long short-term memory (LSTM) based models. Although U-Net was originally proposed for usage within the task of image segmentation it yields impressive results for this audio based task. Likewise, there are several approaches which also implement DenseNet-like architecture with great success. In visually derived approaches such as U-Net, typically the spectrogram from training audio is modelled in the same way as a 2-D image. However, more recently approaches have begun to use only 1-D input waveforms in the temporal domain as well. This also allows for approaches that differ from the typical separation method of spectrogram masking.

The task of ranking music source separation networks is also not always straightforward. Despite the most commonly used metric Source to Distortion Ratio (SDR) being a singular statistical indication of performance, this metric varies across networks in terms of the instrument or musical component being separated, e.g. vocals, drums, bass, or other. Mean opinion scores (MOS) may be best suited to evaluate results from this task but due to the size of the dataset and both the noisiness and time requirements of human evaluation this is often not performed. Additionally, approaches achieving state of the art approaches on our dataset of interest, MUSDB18 [12], almost all use extra training data from private datasets often unavailable to the public due to copyright restrictions. Although there are currently some state-of-the-art approaches performing among the top 10 music source separation systems which only use MUSDB18 data, this does provide a difficult challenge due to the relatively small amount of training data included and is often addressed with simply adding external training datasets. Lastly, as far as we have found, there is no existing work which addresses the affect of noise on this task.

4. Approach

In order to quickly test various networks as well as the effect of different data augmentation regimes, we choose to use the Spleeter toolkit [8], created by the music streaming platform Deezer. For each part of a song provided as a .wav file, a spectrogram is calculated to be used for input to the network with the following parameters: frame length = 4096 samples and step size = 1024 samples. With the MUSDB dataset provided with a sampling rate of 44.1kHz, these values correspond to roughly 0.1 and 0.025 seconds respectively. These values are fairly typical for music-related processing, perhaps slightly on the longer end. While speech recognition processing usually uses smaller frames, most music-related applications require a finer frequency analysis, particularly for lower frequencies which are not of much interest for speech recognition but contain valuable information for discriminating between a bass guitar and bass drum for example.

The Spleeter pipeline accepts these spectrograms as input and models each instrument or component's frequency mask with a U-Net or BLSTM. Therefore, for our experiments separating the music into vocals, drums, bass, and other, 4 unique networks are trained. U-Net models are trained with 12 layers, 6 encoding and 6 decoding with skip connections while BLSTM models consist of 3 layers. Input to these networks is a 3-D tensor consisting of channels (left and right), time steps, and frequency bins. For the purpose of our experiments, noise is treated as an ”other” component within the mixture. The loss used for training is the L1-norm between masked input mix spectrograms and source-target spectrograms and weights are updated by the Adam optimizer. At test time, the mixture spectrogram is masked to produce separation for a desired component.

Both U-Net and BLSTM networks were trained for multiple different update steps in order to see the baseline performance and training behavior for each. Our initial experiments showed the BLSTM began to overfit more quickly than the U-Net and produced lesser performance when trained for a greater number of training steps (40k vs. 200k). In comparison, U-Net showed greater stability in training and was initially chosen to be the primary network for our experiments as it may more clearly show the effect of our data augmentation procedures. However, upon further testing of each model, both the U-Net and BLSTM models show a similar trend in performance with data augmentation. Therefore, results are presented for both models with baseline as well as data augmented performance.

4.1. Database

The MUSDB18 database includes a training set of 100 songs and a test set of 50 songs. The songs are of different genres, and the total duration of all 150 songs is around 10 hours. Each song is provided with their isolated drums, bass, vocals and other tracks.

For training and testing the baseline U-Net and BLSTM models, the original MUSDB18 training and test sets were used. For our goal of extending music source separation to noisy environments, cheering noise (around 2 hours) and babble noise (around 8 hours) derived from multiple Youtube videos [2, 4, 5, 3, 1] were mixed with MUSDB18 songs. In the specific task of testing trained models' performances on noisy songs, a new test set was created by adding noise to the original MUSDB18 test set with a 12 dB signal-to-noise ratio (SNR).

For training the improved U-Net and BLSTM models, a new training set was used. The new training set was created by applying multiple data augmentation techniques on the original MUSDB18 training set. In the end, we expanded 100 training songs into 800 songs.

4.2. Data augmentation

Since the duration of training data from MUSDB18 is limited to less than 10 hours, expanding the dataset is important and three particular data augmentation methods were applied to see if the U-Net and BLSTM models' performance on noisy songs can be improved.

Method 1. Previous experiments showed that traditional audio data augmentation methods such as time stretching, pitch shifting, track scaling, and filtering have very limited impact on music source separation results [11]. In most cases, these data augmentation techniques even lowered the separation performance. Among the traditional audio data augmentation methods tested in [11], only channel swapping (swapping the left and right channels of stereo songs) achieves a stable improvement in some evaluation metrics without damaging the overall separation performance. Improvements made from channel swapping was also observed in [15]. Thus, along with another technique that has been popular among music source separation tasks – random mixing of instruments from different songs (also tested in [15]), channel swapping is used in our first data augmentation method.

The specific steps for generating new mixtures are as follows: (1) Split original songs into pairs. (2) Each pair has song 1 and song 2. Tracks from songs assigned as 1 are channel swapped. (3) Next, within each pair of songs, two track types (out of bass, drums, vocals, other) are selected. Tracks of the selected types are exchanged between song pairs. (4) New songs are created by mixing the new track combinations. An example of this method could be of the form: new mixture = song 2 vocals + song 2 bass + song 1 drums (channel swapped) + song 1 other (channel swapped).

Method 2. Mix-audio data augmentation, a new audio data augmentation method specifically designed for music source separation tasks [13], is applied in method 2. Different from previous remixing instruments methods that remix tracks from different songs, this method randomly mixes audio segments from different times of the same track as an augmented segment for that track:

$$S_{mix} = \sum_{j=1}^{J} S_j$$

where $S_j, j = 1, ..., J$ are audio segments from the same track, and J is the number of segments to be mixed. Our J is set to either 2 or 3. S_{mix} is the mixed track. If S_j are vocals, then, their addition S_{mix} is also vocals. The mix-audio data augmentation provides a large amount combinations of one source. A new song *xmix* is then created by mixing the new mixed tracks S_{mix} vocals, S_{mix} drums, S_{mix} bass, and S_{mix} other. This augmentation method makes separating tracks from mixtures even more challenging. Intuitively, a system that is able to separate multiple vocals from a mixture is also able to separate a single vocal from the mixture [13].

Method 3. The last augmentation method is adding noise to input songs. This is inspired by denoising autoencoders. Since our goal is to improve our model's performance on noisy songs, we want to force the networks to learn to ignore noise while reconstructing tracks by adding noise to the training mixtures. Noise files from our noise dataset are used with varied SNRs.

Combination. The three data augmentation methods are combined to expand the original MUSDB18 training set. The process is shown below in Fig. 1. In-song mixing refers to method 2. Across song mixing refers to method 1.

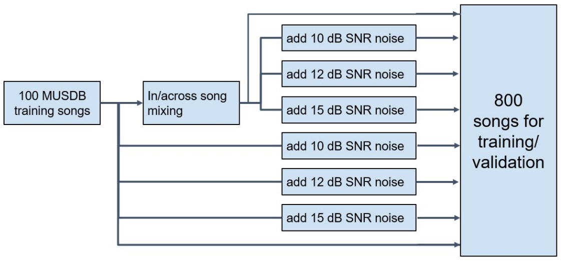

Figure 1. Data augmentation process

4.3. Training

The input training dataset is split into a sub training set and a validation set using a percent ratio of 86%:14%. During training, songs are further divided into 20 second chunks and shuffled. Within each 20 second chunk, a sub 12 second segment is then randomly cropped. This is to ensure efficient spectrogram caching while keeping some randomness in the selected segments [6]. The spectrograms of these 12 second segments are the real inputs to the training networks.

Specific configuration and parameters used for training are as follows: learning rate = $1e^{-4}$, batch size = 4, max training steps = 120k (for both baseline U-Net and augmented U-Net), 41k (for baseline BLSTM) and 160k (for augmented BLSTM). An NVIDIA GTX 1660 Ti (6 GB) was used for training. The total training time was around 22 hours.

4.4. Evaluation

Instead of simply using SDR as our sole evaluation metric, all of the three standard source separation measures [16]–Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR) and Signal to Artifact Ratio (SAR) are derived. Given estimated tracks t'_j and ground truth tracks t_j, where $j = 1...4$, SDR indicates how close t'_j is to t_j. SIR indicates how discriminative t'_j is with all other ground truth tracks. SAR indicates how much of the ground truth t_j t'_j has with relation to unwanted burbling artifacts.

The same evaluation method used in [11] is applied: During evaluation, separated tracks from test songs are divided into 1-sec segments, and SDR, SIR, and SAR are computed for each 1 second segment using the Museval toolbox by [14]. The median SDR, SIR, and SAR for each track are then derived for each test song. After that, another set of median SDR, SIR, and SAR are derived (from the previously computed medians) for each track across all songs. At the end, one SDR, one SIR, and one SAR are kept per track.

5. Results & Analysis

The evaluation results of the baseline models in Table 1 and Table 2 show that SDR is generally better for the BLSTM model, and SIR is better for the U-Net model. Although SDR is most commonly chosen as the one metric to evaluate source separation, BLSTM's improved performance in this respect does not directly translate to better sounding results. BLSTM is however, able to train more quickly than U-Net as it contains only 3 layers compared to U-Net's 12 layers and therefore may be preferable for some applications for this reason.

After data augmentation, improvements can be seen in bold in Table 1 and Table 2. There is a small improvement in vocals SIR when it comes to the augmented U-Net model. Comparing between the mix-audio approach shown in Table 1 and the approach of simply adding noised training data, there is not a clear improvement shown by the mix-audio approach. This confirms the work of Pretet et. al [9] in that typical approaches to data augmentation involving waveform manipulations do not yield significant improvements for the task of source separation.

Although separation metrics are on average less than for the baseline evaluation, data augmentation does yield improvements particularly visible for Drums and Bass with the BLSTM model. One issue for the BLSTM model is that, it begins to overfit after certain amounts of training steps. This phenomenon is extremely obvious in the Drums and Bass parts of the networks. This may be caused by the lack of regularization techniques used in the BLSTM model. Unlike the U-Net model, which utilizes several dropout layers, the BLSTM model only has three BLSTM layers. As mentioned above, from the augmented BLSTM results shown in Table 2, we can see that even though the overall performance of the model does not improve much on the noisy dataset, the separation results for the bass track do improve in general. This shows that data augmentation still helps with overfitting problems in music source separation tasks.

Lastly, as shown by bolded numbers in the difference column in Table 3, the BLSTM model without any data augmentation is less affected by noise introduced to the test set than the U-Net model. This result is generalized across instrument or musical component suggesting the choice of model may be more meaningful in terms of performance than the method of data augmentation employed for this task.

Additionally, upon evaluation of spectrograms resulting from each separation, it seems that the network does not prioritize frequency content much above 10 kHz. Although the bulk of musical information is not carried above this frequency range, it does contribute to the naturalness of sound, especially vocals. This is likely due to the wider variation in this frequency range in the case of vocals due to to microphone choice, studio acoustics, and the upper timbral register of each individuals voice compared to the more easily recognizable and therefore learn-able harmonic structure present below this range.

		UNet	
		Base (120k)	Augmented (120k)
Vocals	SDR	3.609	3.545
	SAR	3.812	3.461
	SIR	9.381	**9.775**
Drums	SDR	4.187	4.132
	SAR	4.956	4.764
	SIR	9.493	9.487
Bass	SDR	3.792	3.565
	SAR	4.708	**4.766**
	SIR	8.313	7.572
Other	SDR	3.204	3.134
	SAR	4.133	**4.163**
	SIR	5.698	5.358
Average	SDR	3.698	3.594
	SAR	4.40225	4.2885
	SIR	8.22125	8.048

Table 1. Baseline vs augmented U-Net models evaluation results on noisy test dataset. The number 120k corresponds to the max training steps taken to train the models.

		BLSTM	
		Base (41k)	Augmented (160k)
Vocals	SDR	3.833	3.724
	SAR	4.734	4.584
	SIR	8.368	7.769
Drums	SDR	4.123	4.058
	SAR	5.101	4.979
	SIR	8.296	**8.711**
Bass	SDR	3.552	**3.592**
	SAR	4.954	**5.084**
	SIR	6.69	**6.862**
Other	SDR	3.446	3.337
	SAR	4.392	**4.449**
	SIR	5.138	4.976
Average	SDR	3.7385	3.67775
	SAR	4.79525	4.774
	SIR	7.123	7.0795

Table 2. Baseline vs augmented BLSTM models evaluation results on noisy test dataset. The numbers 41k and 160k correspond to the max training steps taken to train the models.

6. Conclusions

In summary, introducing noise that resembles a desired separation component, in our case vocals, presents a very difficult task. We confirmed that data augmentation does not necessarily improve separation metrics when the original dataset is small. Through various evaluations of both U-Net

		Base UNet (120k steps)			Base BLSTM (41k steps)		
		Plain Test Data	Noisy Test Data	Difference	Plain Test Data	Noisy Test Data	Difference
Vocals	SDR	4.7	3.609	-1.091	4.516	3.833	**-0.683**
	SAR	4.574	3.812	-0.762	5.45	4.734	**-0.716**
	SIR	11.083	9.381	-1.702	10.028	8.368	**-1.66**
Drums	SDR	4.466	4.187	-0.279	4.321	4.123	**-0.198**
	SAR	4.474	4.956	**0.482**	4.707	5.101	0.394
	SIR	9.282	9.493	**0.211**	8.326	8.296	**-0.03**
Bass	SDR	3.784	3.792	0.008	3.504	3.552	**0.048**
	SAR	4.896	4.708	-0.188	5.087	4.954	**-0.133**
	SIR	6.994	8.313	**1.319**	6.304	6.69	0.386
Other	SDR	3.047	3.204	**0.157**	3.382	3.446	0.064
	SAR	3.9	4.133	0.233	3.99	4.392	**0.402**
	SIR	5.311	5.698	0.387	4.621	5.138	**0.517**
Average	SDR	3.99925	3.698	-0.30125	3.93075	3.7385	**-0.19225**
	SAR	4.461	4.40225	-0.05875	4.8085	4.79525	**-0.01325**
	SIR	8.1675	8.22125	**0.05375**	7.31975	7.123	-0.19675

Table 3. Baseline U-Net and BLSTM models evaluation results on plain vs noisy test dataset. The numbers 41k and 120k correspond to the max training steps taken to train the models.

and BLSTM networks we found the latter to be generally less affected by noise for music source separation within our experiments although U-Net does achieve greater separation performance for both bass and drums in some cases. Lastly, despite the relatively standard separation metrics employed, we found that these may not be fully indicative of separation quality. Despite the statistical differences we report between our baseline and augmented models, with evaluation through listening it is difficult to say which is better performing. Therefore, a human-based judgement such as Mean Opinion Score (MOS) may be more useful as a way to quantify the performance of music source separation.

Future work involving music source separation with noise could involve alternate sequential models such as Gated Recurrent Unit (GRU) networks or a Transformer-like architecture to utilise attention as a learning mechanism. Although there are already some high performing models of this type, we would also like to explore models which perform separation using only the waveform domain, for example: Demucs [7] or Conv-Tasnet [10]. Different separation methodologies such as separating a music track into just vocals, instrumentation, and noise could also be explored as well as implementing a cascaded system of sorts, first feeding the mixture into a noise reduction network then performing source separation on the de-noised output.

References

[1] Ambience soothe [youtube channel]. https://www.youtube.com/channel/UCv63vUuCjJoQBM8hJABZjfQ. 2

[2] Ambience [youtube channel]. https://www.youtube.com/c/Ambience/featured. 2

[3] Relax&sleep [youtube channel]. https://www.youtube.com/channel/UCpHODrWjWtm06H9AlXDoHDw. 2

[4] Sleep sounds express - meditation & relaxation [youtube channel]. https://www.youtube.com/channel/UC7lKPm75KvH73aVhJYp5WXQ. 2

[5] Womb sounds and white noise for babies [youtube channel]. https://www.youtube.com/user/WhiteNoiseGuru/videos. 2

[6] Deezer. Improving config for musdb18, 2019. Last accessed 10 May 2021. 3

[7] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain, 2021. 5

[8] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5:2154, 06 2020. 1, 2

[9] Andreas Jansson, Eric J. Humphrey, N. Montecchio, Rachel M. Bittner, A. Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *ISMIR*, 2017. 1, 4

[10] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, Aug 2019. 5

[11] Laure Pretet, Romain Hennequin, Jimena Royo-Letelier, and Andrea Vaglio. Singing voice separation: A study on training data. *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019. 2, 3

[12] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, Dec. 2017. 2

[13] Xuchen Song, Qiuqiang Kong, Xingjian Du, and Yuxuan Wang. Catnet: music source separation system with mix-audio augmentation, 2021. 3

[14] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. In *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK*, pages 293–305, 2018. 1, 3

[15] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. 03 2017. 1, 3

[16] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006. 3