Comparison and Synthesis of Methods of Musical Audio Source Separation

Jingfei Xia
Electrical & Computer Engineering
jingfeix@andrew.cmu.edu

Nathaniel Jenkins
Biomedical Engineering
nejenkin@andrew.cmu.edu

Prateek Gaddigoudar
Electrical & Computer Engineering
pgaddigo@andrew.cmu.edu

Abstract

Music audio source separation aims at extracting different tracks of sounds from a single music. However it is hard to separate each of the track since they have strong correlation with each other. In this project we will use both some traditional methods such as Principal Component Analysis, Independent Component Analysis, Linear discriminant Analysis, Non-Negative Matrix Factorization as well as some of the state-of-art methods like Wave-U-net to deal with the problem. We will review the different strategies that researchers have employed while attempting to solve this problem, and analyze the performance on different approaches. We will try to explain explain the result by identifying the strengths and weaknesses of various strategies.

1 Introduction

Audio Separation, which is popularly known in psychology as the 'Cocktail party problem' at the beginning. In the Cocktail party problem we want to extract speech from the noisy background music. Methods such as Independent Component Analysis has been proposed to deal with problem in order to separate the speech and music. Recently cover artists on Youtube have become increasingly popular. In order to make cover music, artists have to acquire partial records. For example they need a piece of music without vocal part from the original music, or they just need the drum track and recreate the music by themselves. However, in most cases, popular songs are not released with an off vocal version or a drum-only version. This has proposed a new challenge for us: instead of separating speech and background music as described in Cocktail party problem, we have to separate music into different tracks. In this project, we plan to separate our music into four tracks: drums, bass, vocals and others. We will try several methods that have been used in pattern recognition and analyze their performance on this newly developed problem. Additionally we will also use the state-of-art method like neural network to achieve this goal. By comparing the performance between different methods, we will have a better understanding on this problem and find an optimal solution.

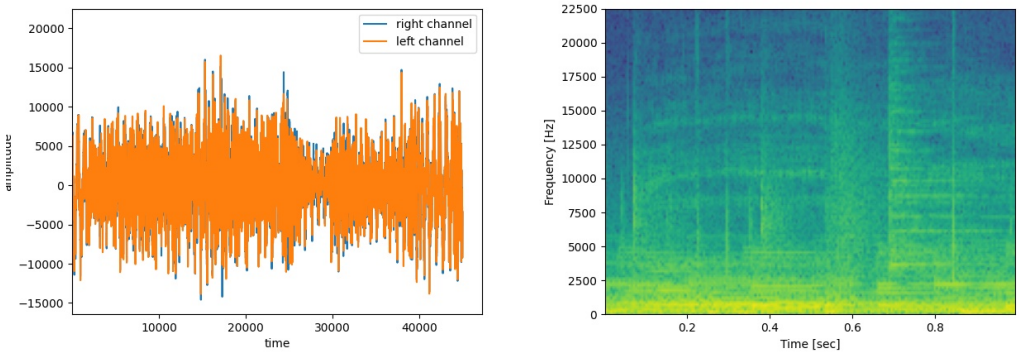
2 Dataset

We will be making use of Demixing Secret Dataset (DSD)[4] to train and test our model. DSD100 dataset has 100 full lengths music tracks of different styles along with their isolated drums, bass, vocals and others stems. The average duration for each song is approximately 4 minutes. The dataset consists of train and test sets with 50 songs each. For a same song, the mixture and the sources are stereophonic, have the same length and the same sampling frequency (i.e. 44,100 Hz).

3 Methodology

3.1 log10 spectrograph

The mean of the Log base 10 spectrograph of both the left and right audio channels were taken. The generated spectrographs have a temporal frequency of .201 kHz, and record 128 frequency responses from 0-5kHz. The following figures represent the left and right audio channels for 1 second of audio and the mean log base ten response to the same interval.



This representation has been proven to be adaptive for similar audio signaling applications by Jin et. al (2012), as it provides an effective framework for pre-emphasizing the more salient lower frequencies of a signal. In order to produce an adaptive representation of the data without producing n unrealistically large representation of the data in memory, the following method will be used:

The temporal resolution of the spectrogram will be fitted to the 45 kHz frequency of the raw data using spline based interpolation. The raw and spectral data from each time point will be vectorized and stored as a 130 dimensional feature (128 spectral responses, 2 raw audio recordings). Principal component analysis will be used to reduce this representation to 1 or 2 significant projections.

3.2 MFCC feature extraction

The audio files are analog signals so before we train on the dataset, we need to transform them into digital signals. Basically this analog-to-digital conversion consists of two parts: sampling and quantization. In this case, we have used MFCC, mel frequency cepstral coefficients as our features representation for other methods. There seven steps to construct MFCC features.

The first stage in MFCC feature extraction is to boost the amount of energy in the high frequencies. Usually when we look at the spectrum there is more energy at lower frequencies than the higher frequencies. Boosting the energy at high frequencies makes there formants more available to our model.

The second stage is windowing. The data we have is complete songs. It would very long utterances and more importantly it will be a non-stationary signal since the spectrum would change very quickly. In this case it would be better if we extract features from a small window of speech. In this project we just use the simplest window, the rectangular window instead Hamming window.

The next step is to extract spectral information for our windowed signal. The tool for extracting spectral information for a sampled signale is Discrete Fourier Transformation. The equation for DFT is:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\frac{\pi}{N}kn} \tag{1}$$

The result of the DFT will be the information about the amount of energy at each frequency band. However human hearings is not equally sensitive at all frequency bands. To modify this kind of feature we can warp the frequencies output by DFT onto mel scale. The mel frequency can be computed from the raw audio frequency as:

$$mel(f) = 1127 \ln\left(1 + \frac{f}{700}\right) \tag{2}$$

After this we take the log of each of the mel spectrum values.

It would be possible to use the mel specturm as feature representation but there is still some problems for the spectrum. So the next step is to do the computation of cepstrum. Cepstrum is a useful way of separating the source and filter. We can use the coefficients in cepstrum to represent our music data/ The cepstrum is defined as the inverse DFT of the log magnitude of the DFT of a signal.

Besides the coefficient we get from the cepstral, we also need another feature: the energy from the frame. The energy in each frame is the sum over time of the power of the samples in the frame.

3.3 Linear projection methods

3.3.1 Independent Component Analysis

Independent component analysis identifies aspects of a signal that vary independently of one another. This is ideal for the application of source separation, as the different sources will exhibit some degree of independence. While it is tempting to apply this directly to a data set and then attempt to map each identified component to a class, the projections identified with this method are not supervised, and as such, do not guarantee association with a class. Like PCA, the projections that are identified by this method can reduce the dimensionality of a data set while preserving important features.

3.3.2 Linear discriminant analysis

Unlike PCA, LDA is optimized for class separability. However, the ultimate measure of class membership similarly derives from the application of an additional algorithm. A metric similar to the probabilistic analysis of PCA is the application of the Mahalanobis distance between the class centers and the observed projection.The Mahalanobis distance is weighted by the variance of each class's distribution, and as such gives a reliable measure of membership probability. The combination of LDA and Mahalanobis analysis is theoretically much more powerful than probabilistic PCA, as it directly compares classes against one another, emphasizing components of a distribution unique to a given class.

3.4 Non-Negative Matrix Factorization

NMF is an unsupervised learning technique that has been applied successfully in several fields, including signal processing, face recognition and text mining [5]. NMF approximates a matrix $\mathbf{X}$ with a low-rank matrix approximation such that $\mathbf{X} = \mathbf{BW}$. Here $\mathbf{B}$ represents Base matrix and $\mathbf{W}$ represents Weight matrix.

Given a mixed audio signal, we make use of Short-time Fourier transform (STFT) to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. This is stored as magnitude spectrum $\mathbf{M}$. We can then make use NMF technique to obtain the weight matrix $\mathbf{W}$ which gives us the contribution of each element, provided we have the base matrices for each of the elements [6]. These base matrices for each element are obtained previously using KL Divergence technique. i.e $\mathbf{M} = [\mathbf{B_1}\mathbf{B_2}...]\mathbf{W}$

Once we obtain the individual contribution of each element in the mixed audio signal through matrix $\mathbf{W}$, we can construct the individual source signals through Inverse-STFT.

3.5 Wave-U-net

In order to separate different music tracks in a single piece of music, we could also use the neural network to achieve the goal. A recent method called Wave-U-net is used to achieve this goal. The network structure is similar to the structure used in image processing. In the paper of Daniel et al., they have propsed a u-net network which is used for speech separation. Unlike in image processing, they change all 2D convolutional layers into 1D. It does the downsampling and upsamling procedure as in image processing. The upsampling procedure is done by transpose convolution. After the upsampling procedure, the features from the corresponding layers in downsampling will be cropped and concatenated to the output of upsampling layers. We use a single convolution layer to connect the downsampling part and upsampling part. In the downsampling block, it performs several convolution to get the shortcut and then performs downsampling. In the upsampling block it performs the

upsampling operation first. Then, it crops the shortcut and upsampling value and concatenate them to keep the input size same as the downsampling procedure. After that it will pass through several convolution layers before input to the next block. Finally there will be a convolution layer to output the final result. The final output will have two channels, representing the right voice and left voice of the audio file.

We have two options for training. We can train one single neural network and set the output channel to be 8 (2 channels for each track and in total there are four tracks). Another solution is that for each track we have the same structure of network. In this problem we have four tracks which are drum, bass, vocals and others. So in total we have four same neural networks. We generate the result for both structure. We train the neural network separately with mixed sound as input and separated track as target output. The structure of network is shown as below.

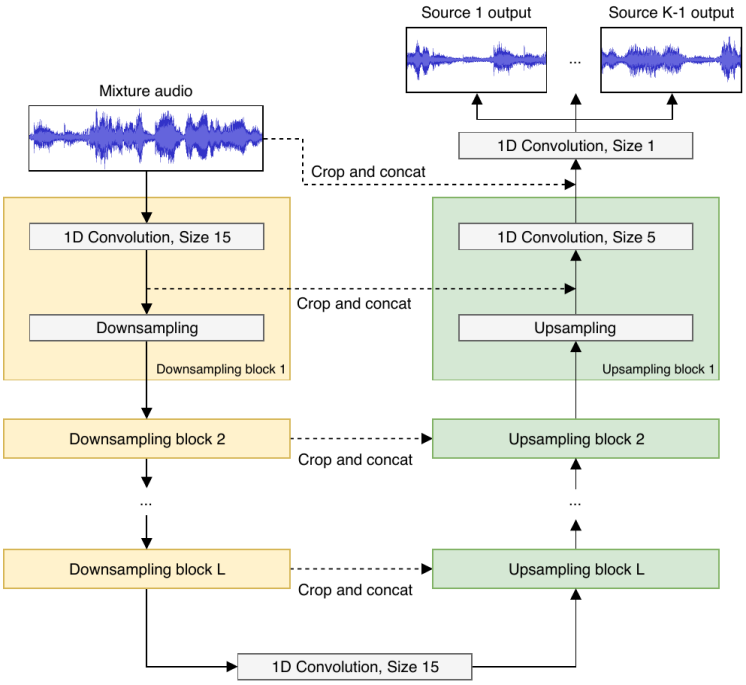


Figure 1: Wave-U-net Architecture

3.6 Open-Unmix

This is a deep neural network reference implementation for music source separation based on a three-layer bidirectional deep LSTM[7]. The model learns to predict the magnitude spectrogram of a target, like vocals, from the magnitude spectrogram of a mixture input. Internally, the prediction is obtained by applying a mask on the input. The model is optimized in the magnitude domain using mean squared error and the actual separation is done in a post-processing step involving a multichannel wiener filter. LSTM doesnot operate on the original input spectrogram resolution, but instead it compresses the frequency and channel axis of the model to reduce redundancy and make the model converge faster. Due to its recurrent nature, the model can be trained and evaluated on arbitrary length of audio signals.After applying the LSTM, the signal is decoded back to its original input dimensionality. In the last step, the output is multiplied with the input magnitude spectrogram so that the model is asked to learn a mask.

4 Performance analysis

Measuring the results of a source separation system is a challenging problem. There are two main categories for evaluating the outputs of a source separation system: objective and subjective. Objective measures rate separation quality by performing a set of calculations that compare the output signals of

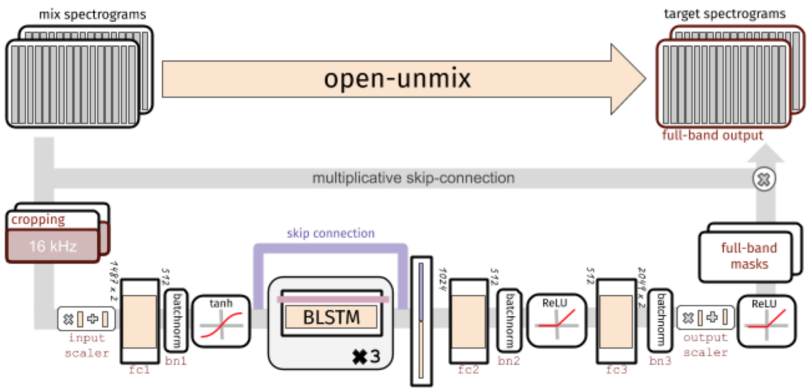


Figure 2: Open-Unmix Architecture

a separation system to the ground truth isolated sources. Subjective measures involve having human raters give scores for the source separation system's output.

Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR) and Source-to-Artifact Ratio (SAR) are the most widely used methods for evaluating a source separation system's output. An estimate of a Source s_i is assumed to actually be composed of four separate components;

$$\mathbf{s_i = s_{target} + e_{interf} + e_{noise} + e_{artif}}$$

where $\mathbf{s_{target}}$ is the true source, and $\mathbf{e_{interf}}$, $\mathbf{e_{noise}}$ and $\mathbf{e_{artif}}$ are error terms for interference, noise, and added artifacts, respectively.

- Source-to-Distortion Ratio (SDR) An overall measure of how good a source sounds.

$$\text{SDR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2} \right) \tag{3}$$

- Source-to-Artifact Ratio (SAR) The amount of unwanted artifacts a source estimate has with relation to the true source.

$$\text{SAR} := 10 \log_{10} \left(\frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2} \right) \tag{4}$$

- Source-to-Interference Ratio (SIR) The amount of other sources that can be heard in a source estimate. That is the amount of leakage.

$$\text{SIR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \right) \tag{5}$$

- Mean Squared Error

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I(x,y) - K(x,y))^2 \tag{6}$$

  This similarity function has been noted to be more sensitive towards the magnitudes present in a signal than the Cosine similarity function, making it an adaptive measurement distinct from that of the cosine similarity function. As a result, the mean response from both of these functions will serve as an adaptive loss function that is robust to differences in both shape and magnitude.

5 Experiments and Results

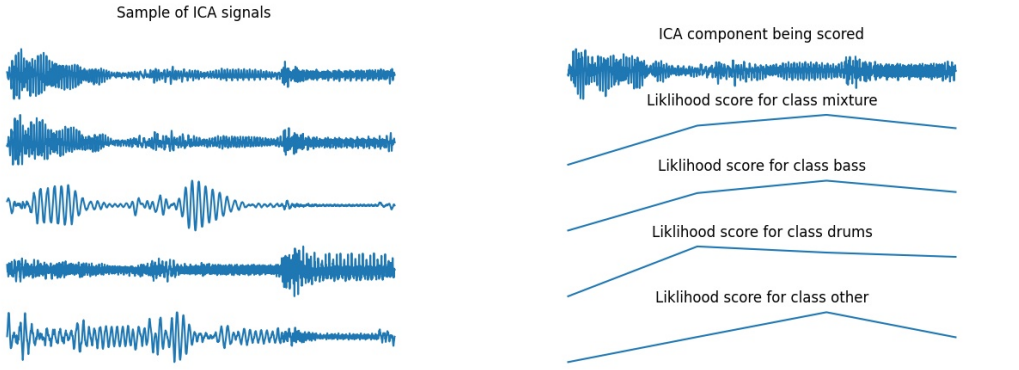
5.1 Linear projection methods

5.1.1 Independent Component Analysis

As we have mentioned before for each piece of music there are both left voice and right voice. So we use these two separated tracks as "observation" in ICA algorithm. Then we try to derive the vocal part and non-vocal part for each piece of music. The result is not very good. For most of the music it cannot separated the vocal part out of the mixed music. Only for a few pieces of music such as "MilkCow Blues" in the dataset it can generate a promising non-vocal part from the mixed music. For the vocal part it still contains a lot of instrumental elements inside it. The reason why the algorithm fails here is that in our dataset for most of songs there is not much difference between right voice and left voice. And another reason is that for a single piece of music we cannot have enough samples for the observation and in this case it makes result not that accurate

5.1.2 Combined ICA and LDA model

One novel method for source separation is to generate a series of related signals by convolving the original audio with a series of signals with frequencies in the audible range (20-20000)Hz, with associated frequencies increasing exponentially. By then applying an ICA projection to these signals, a series of mixed signals will be created that describe independent components of the original signal. The likelihood of each observed component being an observation of a given class can be measured by using an LDA projection space and class associated probability distributions of each class within that space by employing the Mahalanobis distance. This likelihood can be measured at all points along a given ICA component, yielding a likelihood score corresponding to each class that changes over time. An approximate signal for each class can then be estimated by creating a linear combination of each component, weighted by their corresponding LDA likelihood scores.



(a) Sample of observed ICA components

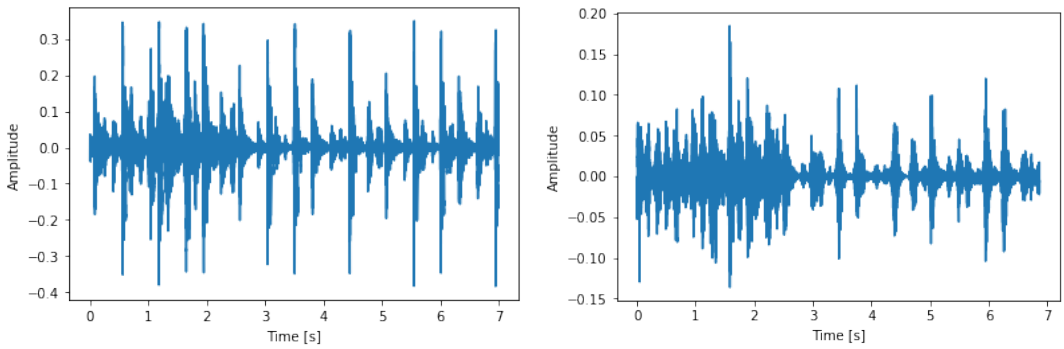
(b) Associated Likelihood scores with ICA component

Figure 3: Summary of ICA analysis

5.2 Non-Negative Matrix Factorization

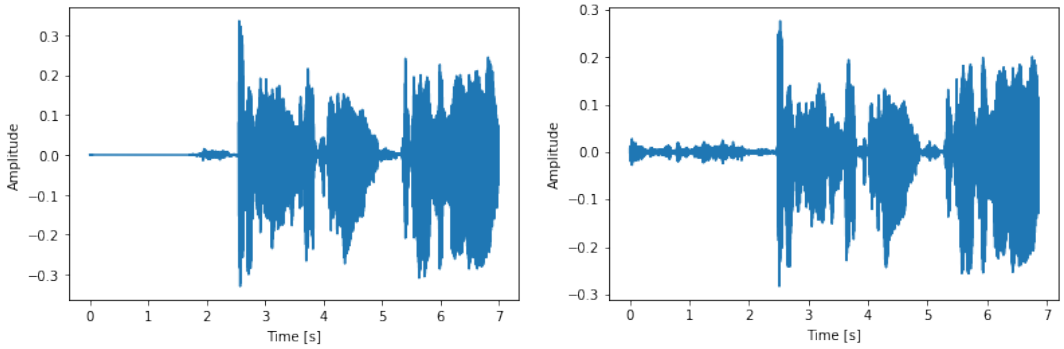
We have implemented 4-way source separation of a given song into 'Vocals', 'Drums', 'Bass' and 'Other' using Non-negative Matrix Factorization. It is observed that NMF implementation gives impressive results when comparing the output to ground truth qualitatively i.e listening to the audio files. In order to quantify this comparison, this project makes use of Mean-Square Error between source files provided by DSD100 dataset and the output generated by model. We can see the mean-Square error values for four elements of a song from Table 1 proposed here.

Although the resultant source files obtained by the model seem satisfactory to a normal listening ear, we observe slight noises from other elements in the background. We need to mitigate these noises to



(a) Original Drums track (b) Generated Drums track

Figure 4: Drums Comparison in NMF



(a) Original Vocal track (b) Generated Vocal track

Figure 5: Vocal Comparison in NMF

obtain lucid source separation. Also, it is observed that the code can be easily upscaled to further separate the song into more than 4 sources, but that comes at the cost of very high run time. This is not affordable when we have multiple sources to separate. Our next approach in this project is to look at other techniques which can be efficiently upscaled to separate more sources and can provide better accuracy than NMF.

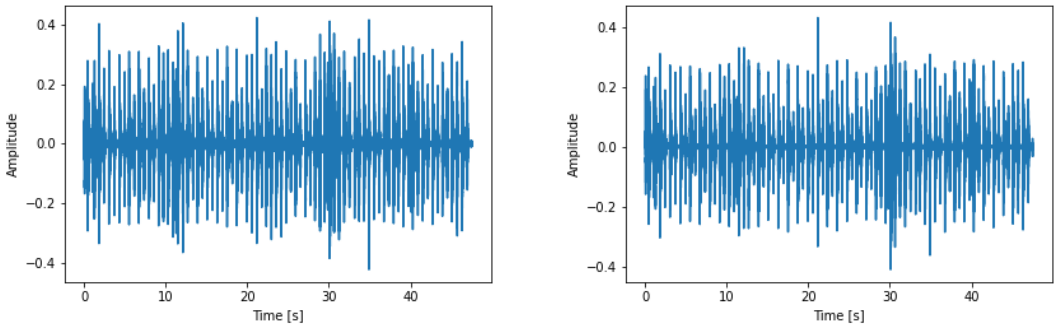
5.3 Wave-U-net

We use the source code from original Wave-U-net and modify the dataloader to fit our dataset. There are four tracks so in total there are four networks. For each of the network there will be five downsampling blocks and five upsampling blocks. In the downsampling block there will be one convolution layer before the shortcut and one convolution layer after the shortcut. For the upsampling block, there will be one convolution layer before concatenating the shortcut and one after that. During the training we use mean square error as our measurement. The kernel size is set to be 5 for each convolution layer. The learning rate is set to be 0.001.

For the single network with multiple output channels, we can see the result of Wave-U-Net (not separated) from Table 1. We can see mean square error is much smaller than NMF we have discussed before and it shows a pretty good result.

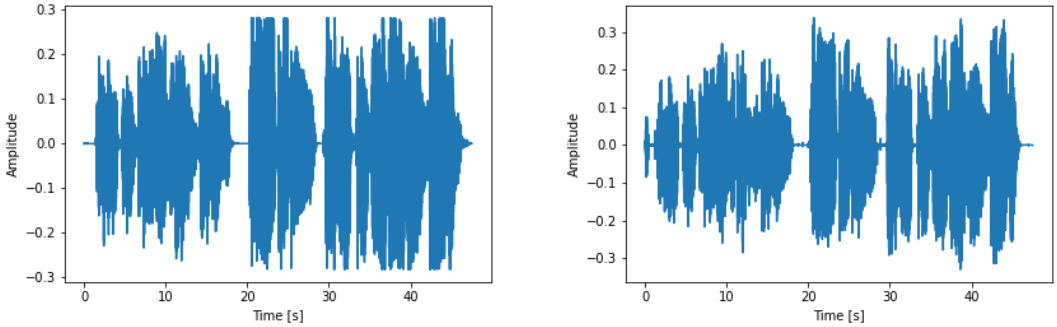
For the multiple network we use the model that has trained on MUSDB18 dataset, which is a much larger dataset than our DSD100 dataset.

The result is better than single network. As we can see metrics of Wave-U-Net (separated) from the Table 1, the MSE has reached a quite small value comparing with other methods. As we can also



(a) Original Drums track (b) Generated Drums track

Figure 6: Drums Comparison in Wave-U-Net



(a) Original Vocal track (b) Generated Vocal track

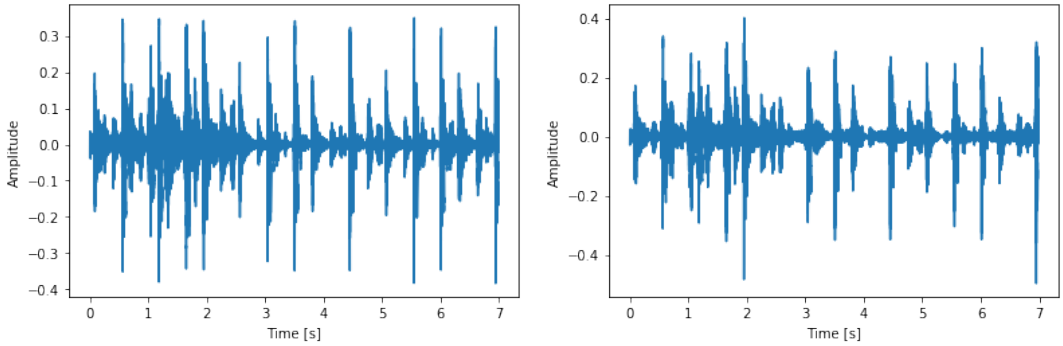
Figure 7: Vocal Comparison in Wave-U-Net

see from the time domain of different tracks from Figure 6 to Figure 7, the generated tracks and the target track are similar to each other. However if we listen to the audio file there is still some noise inside the audio. As the table of SDR indicates, This may be because the size of the dataset is small. Another possible reason is that we have to re-design our loss function in the network since we only consider MSE in this case, and if we add some noise to the deprecated audio, it would possibly reach amplitude of the target output.

5.4 Open-Unmix

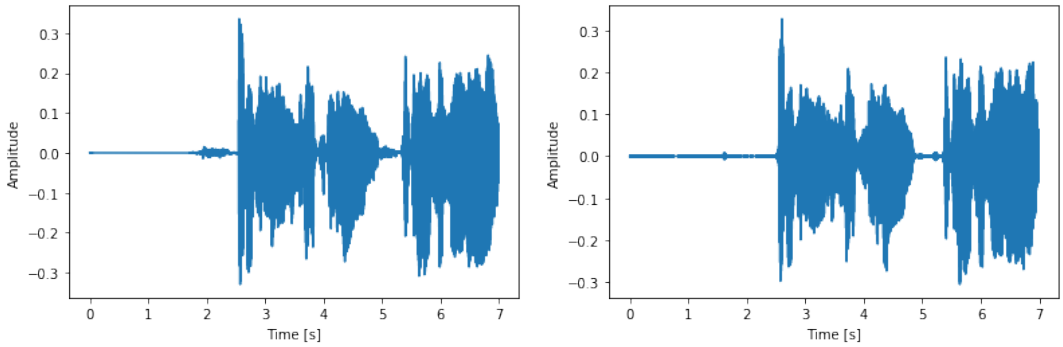
It is observed that Open-Unmix achieves comparably much better results compared to traditional feature-extraction techniques. Source code made available by nussl was used for this approach. nussl is audio source separation python library created by Interactive Audio Lab at Northwestern University. The reason behind choosing this library is it's easy-to-use framework for prototyping and adding new algorithms. Open-Unmix algorithm comes pre-trained on a larger MUSDB dataset in this framework. The testing was done on our DSD100subset and results tabulated in the following tables. As mentioned, Open-Unmix achieves smaller mean square error values between the source estimates and the ground truths. Also, the ratios SDR, SAR and SIR are high and positive indicating that there was very little disturbance in the source estimate quantitatively.

Subjectively, the estimates obtained by this approach sound much more clearer to the listeners. If listened keenly, one can still hear the background interference. But, that's negligibly small making the source estimate sound almost similar to the ground truth source.



(a) Original Drums track (b) Generated Drums track

Figure 8: Drums Comparison in Open-Unmix



(a) Original Vocal track (b) Generated Vocal track

Figure 9: Vocal Comparison in Open-Unmix

6 Metrics Evaluation

We have used MSE, SAR, SIR and SDR to evaluate different methods we have implemented. The metrics are shown in the following table.

Table 1: MSE of different methods on DSD100

Method	Bass	Drums	Other	Vocal
ICA/LDA	0.0103	0.0027	0.0031	0.0049
NMF	0.0031	0.0032	0.0044	0.0024
Open-Unmix	0.0006	0.0007	0.0011	0.0002
Wave-U-Net (not separated)	0.0022	0.0015	0.0016	0.0026
Wave-U-Net (separated)	0.0003	0.0004	0.0010	0.0012

Table 2: SAR (in dB) of different methods on DSD100

Method	Bass	Drums	Other	Vocal
ICA/LDA	-31.3	-31.5	-36.3	-38.2
NMF	-42.92	-61.30	-42.92	-40.58
Open-Unmix	+6.3	+6.0	+4.7	+6.5
Wave-U-Net (not separated)	-0.81	-0.19	0.20	-2.50
Wave-U-Net (separated)	+5.48	+5.26	+3.32	+6.04

Table 3: SIR (in dB) of different methods on DSD100

Method	Bass	Drums	Other	Vocal
ICA/LDA	-0.41	-5.00	-1.21	-0.55
NMF	+6.75	-11.31	+6.75	+6.86
Open-Unmix	+9.2	+11.1	+6.5	+13.3
Wave-U-Net (not separated)	+13.50	+10.3	+3.92	+14.03
Wave-U-Net (separated)	+15.01	+16.0	+9.7	+18.57

Table 4: SDR (in dB) of different methods on DSD100

Method	Bass	Drums	Other	Vocal
ICA/LDA	-31.3	-31.6	-36.2	-38.2
NMF	-42.92	-61.30	-42.92	-40.58
Open-Unmix	+5.2	+5.7	+4.0	+6.3
Wave-U-Net (not separated)	-0.82	-0.33	-1.47	-0.88
Wave-U-Net (separated)	+6.18	+7.24	+2.42	+5.63

7 Conclusion

We have implemented the traditional Independent Component Analysis(ICA), combined ICA/LDA, Non-Negative Matrix Factorization(NMF), Open-Unmix method and Wave-U-Net to solve the problem. And we can see from the result that neural network really improve the performance of separating different tracks in a single music. Traditional method like ICA, LDA and NMF is very hard to solve this problem. Even though we implement different neural network in this problem, we can still find that the generated audio file also contains a lot of noise. In the future it might be a possible way to solve the problem with better-designed neural network.

8 Future Work

As we can see from the resuflft, the Wave-U-net has generated terrific result in our problem. However, the DSD100 dataset we use is mainly composed of pop and rock music. We can still try to generate more results on other genres of music and see the performance from that. Additionally we can try to re-design the loss function in Wave-U-net in order to better estimate the divergence between the generated audio file and the ground turth. Additionally we could try to implement the algorithms in series / parallel to improve the performance. We could combine the methods like what we have done in LDA/ICA in order to achieve better performance. Further we could continue to try other SOTA neural networks to do the separation.

References

[1] Luo, Y., Chen, Z., Hershey, J. R., Le Roux, J., Mesgarani, N. (2017). Deep clustering and conventional networks for music separation: Stronger together. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2017.7952118

[2] Daniel S., Sebastian E., Simon D.,. (2018). Wave-U-Net: A multi-scale neural network for end-to-end audio source separation. arXiv.org. https://arxiv.org/abs/1806.03185v1

[3] Liutkus, A., Stöter, F., Rafii, Z., Kitamura, D., Rivet, B., Ito, N., Ono, N., Fontecave, J. (2017). The 2016 signal separation evaluation campaign. *Latent Variable Analysis and Signal Separation*, 323-332. https://doi.org/10.1007/978-3-319-53547-0_31

[4] The why and how of Nonnegative matrix factorization. (2014). *Regularization, Optimization, Kernels, and Support Vector Machines*, 275-310. https://doi.org/10.1201/b17558-15

[5] Lee, D. D., Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791. https://doi.org/10.1038/44565

[6] J. Huang, P. C. Yuen, W. Chen and J. H. Lai.(2007) *Choosing Parameters of Kernel Subspace LDA for Recognition of Face Images Under Pose and Illumination Variations* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 4, pp. 847-862

[7] Stöter, F., Uhlich, S., Liutkus, A., Mitsufuji, Y. (2019). Open-unmix - A reference implementation for music source separation. Journal of Open Source Software, 4(41), 1667. https://doi.org/10.21105/joss.01667