SINGING VOICE SEPARATION WITH DEEP U-NET CONVOLUTIONAL NETWORK

Anthony Zhang V00018885

Eric Yvorchuk V00864667

Mek Obchey V00880355

Ryan Chen V00842774

Vincent Potrykus V00917465

1. INTRODUCTION

A song often has two parts: the part that contains the vocal content, and the part that contains the non-vocal instrumental content. Separating the vocal and accompaniment parts from each other is useful for several MIR tasks, such as singer identification and lyric transcription as it would be much easier to perform these tasks on a clean vocal signal than a mixture signal [1]. Singing voice separation is also useful for karaoke applications, as the instrumental component allows users to sing along with the songs with or without the original singer; however, this application is generally quite difficult as the separation needs a high degree of cleanliness in order for the karaoke track to be useful for singing in public settings. Our group was interested in singing voice separation because of the aforementioned practical applications for the general public.

While there are traditional algorithms, such as auto-correlation-based, filter-based, and pitch-based for doing source separation of musical audio, deep learning models have emerged as powerful alternatives. Our group members are very interested in deep learning topics and how deep learning techniques can be applied to musical information retrieval tasks. Thus, for our project, we aimed to use U-Net, a neural network-based algorithm, to implement a system that performs singing voice separation by training models that predict the vocal and backing track components from a piece of music's audio signal, and applying the trained models to input music tracks to generate the separated vocal and instrumental tracks. Figure 1 below shows a diagram of the architecture.

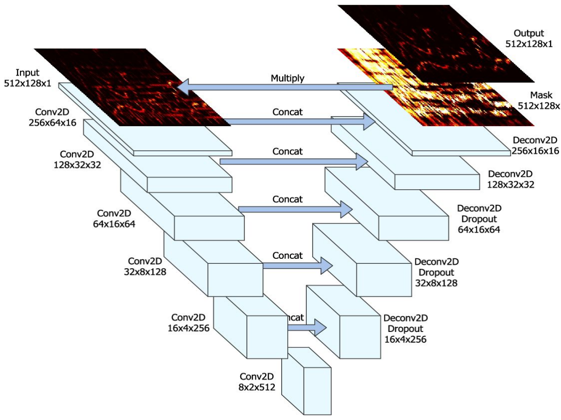

Figure 1: U-Net architecture

U-Net is a fully convolutional network for image segmentation. It consists of two parts, encoding and decoding. As shown in figure 1, the network appears to be U-shaped, hence the name U-Net. In the encoding stage, the input image gets down-sampled in each layer, it allows the neural network to learn the "WHAT" information and loses the "WHERE" information by encoding the images into smaller and deeper representations (smaller in image size but bigger in the number of channels) through multiple convolutional layers, and during the decoding stage, it restores the "WHERE" information by gradually applying up-sampling so that the image recovers to its original size.

2. RELATED WORK AND BACKGROUND

In [1], Jansson et al introduced their implementation of U-Net for performing singing voice separation tasks. Their data indicates that the qualities of separated vocal component and accompaniment component that were obtained from the U-Net algorithm have a significant improvement over other state of art methods, particularly for being able to recreate the small details needed for high quality audio reproduction. This paper also serves as our main reference for the implementation of our project and is the original source for figure 1. As such, we used a very similar architecture and evaluation methods that are described in later sections.

In [2, 3], researchers introduced the U-Net architecture as a method for biomedical image segmentation and discussed the superior performance of U-Net in visual recognition tasks compared to other state of the art algorithms, which later inspired researchers to apply the U-Net method to MRI tasks such as singing voice separation to achieve better performance. In particular, they found that they were able to use a fully convolutional network and a large degree of data augmentation on a limited training dataset to achieve better performance than other networks, which were found to be slow and did not simultaneously have good localization and use of context.

In [4], the authors proposed and implemented Wave-U-Net, which is a modified version of U-Net, for performing end-to-end audio source separation. Wave-U-Net aims to solve the drawbacks that it is difficult for traditional U-Net to produce high quality separation results due to high sampling rate. Their results yielded an improvement compared to traditional U-Net under the same settings, however, the improvement of results was not very significant, and the implementation was more complicated. Thus, due to the limited time we had to implement the system, we chose to implement our system using the traditional U-Net architecture.

3. METHODS

Our goal was to create a neural network based on the U-Net architecture in order to predict the vocal and accompaniment components of mixed audio tracks, where each component would be represented by its own model. In particular, we wanted the predicted tracks to maintain the respective target component and filter out the other component with high accuracy and clarity. The plan was to take the data we have gathered, consisting of their individual vocal stems, instrumental stems and the original mixture, and process them in a way so that the data can be utilized in order to train our models. We have implemented our U-Net architecture using the Keras library from Tensorflow since it is easy to use for deep learning tasks [5].

3.1 Datasets

As stated above, the training data we used for our models comes in the form of a triplet, in which we have the mixture, vocal, and instrumental stems. We chose to make use of two datasets that fit this criteria: MedleyDB and DSD100. These two datasets provide us with 120 tracks collectively, encoded at 44.1kHz.

3.1.1 MedleyDB

MedleyDB is a dataset of multitrack audio originally created for music research on melody extraction, aiming to address the shortcomings of existing collections [6]. This dataset provided us with 20 royalty-free multitracks (mix, processed stems, raw audio and metadata) of different genres, including pop, rap, jazz, and more.

3.1.2 DSD100

The DSD100 dataset contains 100 full length audio tracks, each of which consisted of both its vocal and instrumental tracks along with the original mixture [7, 8]. This dataset contained two folders, one which contained 50 songs used for training, and another folder also containing 50 songs used for testing. The data from the DSD100 dataset is derived from The ‘Mixing Secrets’ Free Multitrack Download Library.

In order to acquire more training data, we employed data augmentation techniques such as gain control, pitch-shifting, and time-stretching on the existing datasets using Scaper [9]. Scaper is a Python library that provides an open source tools data for music source separation.

3.2 Implementation

In order to make sure that the models could train the data smoothly, we preprocessed both the MedleyDB and DSD100 datasets to make them more manageable. We first downsampled the input audio to 8192 Hz, and then computed the Short Time FFT with a window size of 1024 and a hop size of 768 samples. The magnitude spectrograms were normalized and then saved in a .npz format to be used to train the models.

The U-Net architecture consists of an encoder component and a decoder component. The encoder component is a set of convolutional layers that reduces the input dimensionality while preserving prevalent information. Localization is applied, which combines high resolution features from the contracting path with the unsampled output. Each encoder layer consists of 5x5 convolutions (filters) with stride 2, and ReLU with leakiness alpha = 0.2. The first layer has 16 filters which is doubled at each downsampling layer, with each downsampling step doubling the number of features map.

The decoder uses strided deconvolution, which is symmetric to the encoder. As such, it maintains the same number of filters, sizes, strides, and output dimensions. The decoding component uses a 50% (p = 0.5) dropout to the first 3 layers, 5x5 convolutions with stride 2, batch normalization, and plain ReLU. In the final layer, we use a sigmoid activation function, which is useful for models where we have to predict the probability as an output. Finally, the model is trained with a patch of 128 frames, which is approximately 30 seconds, for mini-batch training, utilizing the ADAM optimizer.

3.3 Tools

Our implementation was done in Python, with the bulk of it done through Jupyter Notebooks. We also used Google CoLab to speed up the training phase, as well as to use the added benefits of established sharing capabilities. We also maintained a GitHub repository to allow simultaneous collaboration, which contains the source code for preprocessing the data, training the models, generating the predicted tracks, and evaluating the results [10].

For this project we made use of TensorFlow, which is an open-source library developed by Google. It is primarily used for deep learning applications, and supports traditional machine learning. We made use of a high-level API built on top of TensorFlow 2.0 known as Keras. Since the core data structures of Keras are layers and models, the Keras API was incredibly useful for creating our U-Net architecture.

4. RESULTS

We used both objective and subjective evaluation methods for assessing our results. The objective methods provided quantifiable information about the quality of our results, while the subjective methods provided a more intuitive and typical way for humans to assess the results.

4.1 Objective Evaluation

For the objective evaluation we used the mir_eval library [11, 12]. Following the voice separation paper by Jansson et al, we used the following metrics:

- **Normalized Signal to Distortion Ratio (NSDR):** In order to account for the fact that the track might contain periods of no data (eg. when the singer isn't singing), NSDR compares the predicted (`estimated_src`) track to the test (`reference_src`) track, compares the mix track to the predicted (`reference_src`) track and computes the difference of those two:
  (, ,) = (,)−(,)
- **Signal to Interference Ratio (SIR)**
- **Signal to Artifacts Ratio (SAR)**

For example, to compute the SIR of the vocal part of the tracks, we use the `bss_eval_sources` method, where `reference_src` is an array of test tracks and `estimated_src` is an array of predicted tracks:

```
_, sir, _, _ =
bss_eval_sources(reference_src,

estimated_src)
```

This method returns an array of float values corresponding to the SIR of each predicted track - test track pair. Other metrics are calculated in a similar way.

For each track we calculated 6 metrics:

- Vocal SIR
- Vocal SAR
- Vocal NSDR
- Accompaniment SIR
- Accompaniment SAR
- Accompaniment NSDR

The tables below show the metrics obtained by our model compared with MIREX and Spleeter for one sample track.

Comparison with MIREX

	MIREX	Our model
Vocal SIR	15.308	-15.354
Vocal SAR	11.301	-4.897
Vocal NSDR	8.681	-0.635
Accompaniment SIR	21.975	20.996
Accompaniment SAR	15.462	11.373
Accompaniment NSDR	7.945	34.382

Comparison with Spleeter

	Spleeter	Our model
Vocal SIR	35.000	30.706
Vocal SAR	11.709	9.082
Vocal NSDR	13.116	10.473
Accompaniment SIR	31.716	31.103
Accompaniment SAR	11.833	10.221
Accompaniment NSDR	10.1221	8.516

4.2 Subjective evaluation results

The subjective evaluation method for assessing the perceived quality of source separated audio is similar to the protocols proposed by Emiya et al [13], though we designed our subjective assessments to be much simpler due to the inability to recruit evaluators on campus. Additionally, recruiting online evaluators from paid platforms is not in our budget, so we evaluated the results by ourselves. A sample of our source-separated vocal and accompaniment tracks is in our GitHub repository so readers can subjectively evaluate our models' performance as well. For each input mixture, we provide the predicted vocal and accompaniment parts along with their corresponding ground truth audio for comparison.

We evaluated our vocal and accompaniment models by the following metrics from Emiya et al:

1. Global quality,
2. Preservation of the target source,
3. Suppression of other sources, and
4. Absence of additional artifacts.

The global quality of the source separated vocal and accompaniment audio are consistent with their corresponding ground truth. While some vocal parts are perceived to have a slightly lower volume, their overall quality is relatively the same.

For the preservation of the target source, the predicted vocal parts are much clearer for music tracks with less Beats-Per-Minute (BPM). In particular, the lyrics from pop, reggae, and country tracks are clearer and more understandable than the lyrics from heavy metals tracks with higher BPM. However, this problem doesn't seem to happen in any of the predicted accompaniment parts in any music genre.

The suppression of the vocal component by the accompaniment model gives the impression that the perceived vocal parts are highly suppressed to the point where lyrics are hard to hear. Especially for pop and country tracks, the vocal components are close to a faint background noise and can be easily filtered out by our ears. Unfortunately, we cannot say the same for the vocal model as its predictions are inconsistent. For some of the predicted vocal tracks, the accompaniment components can still be heard clearly, and which musical instruments or chords were played can also be exactly identified.

The absence of additional artifacts does not happen for the vocal model. Most, if not all the multiple musical instruments perceived in the predicted vocal parts are transformed to a distorted sound that is present in many predicted vocal tracks. Interestingly, this particular sound is very noticeable in rock and heavy metal tracks that have longer and continuous bass and drums, but much less so for tracks with a single musical instrument. Similarly, the predicted audio of the accompaniment model also contains the same distorted sound that appears to be masking the vocal part.

5. DISCUSSION AND CONCLUSION

Our project was successful in the sense that we were able to develop a working U-net architecture to separate an audio track into its vocal and accompaniment components, though the quality of the results had mixed success. In particular, as evidenced by the large deviation of the metrics between our model and MIREX for the vocal tracks, the issue of the lyrics being difficult to understand for certain genres and tempos, and the fact that there is inconsistency in how much of the accompaniment is suppressed in the vocal track, the accompaniment model tends to perform much better than the vocal model. A possible reason for this is that since there are fewer components in the vocal part than the accompaniment part, noise and distortion are more noticeable in the vocal part. Another possible reason is that since there are more accompaniment components, it is thus more difficult to filter those out from the vocal track. Our project was also unsuccessful in preventing additional distortion from both models, compromising the tracks' clarity. Despite this, the project was successful in having the predicted tracks from both models maintain the quality of the original mixture, having the accompaniment model filter out the vocals significantly, and having the metrics be consistently close between the model and Spleeter.

Due to the mixed success of the project, it is clear that the existing models could be improved significantly. The most obvious way of doing so would be to acquire more training data. While the data augmentation techniques helped in this regard, they led to a marginal improvement in the results. Thus, other entirely new datasets are likely necessary, such as the iKala dataset used by Jansson et al [14], though it is difficult to find sets in which the tracks come with the mixture as well as the vocal and accompaniment components. Jansson et al described a different method of acquiring training data, where they searched for tracks where the original mixture and instrumental parts were available, and they used the corresponding magnitude spectrograms to yield the vocal magnitude spectrogram by taking their half-wave rectified difference. Doing so yielded sufficient ground truth vocal parts and could have also been used for this project.

There are also many ways that the project could be built upon for further applications. One way would be to figure out how to efficiently train the models using the tracks at the original sample rate so that the resulting predicted tracks could be of a much higher quality, which is desirable for karaoke tracks. Another way would be to further separate the vocal and accompaniment components into their subcomponents. For example, the vocals could be further separated into lead and backing vocals so that the latter could be merged with the accompaniment for a more accurate karaoke track. The accompaniment component could be separated into its bass, drums, and other instruments, similarly to Spleeter, into its instrumental families, or even into its individual instruments. The last application would be particularly useful for someone wishing to follow along with a track using whatever instrument they specialize in. This further separation would be difficult, however, as the individual ground truth components are not always readily available.

Singing Voice separation is a typical MIR task because of its practicality in applications such as lyric transcription and karaoke track generation, both of which are commonly used by the general public. While there are many techniques to carry out this task, using a U-net architecture is notable for its high efficiency and production of high-quality results. In general, our accompaniment model performs better than our vocal model, but both models have noticeable distortion in the predicted tracks. However, predicted tracks from both models maintain the quality of the original mixture and the lyrics for predicted vocal tracks from some of the genres are understandable. Overall, our group feels that our U-net architecture was a commendable first version for singing voice separation with many clear areas for improvement and expansion.

6. REFERENCES

[1] Jansson, A et al. *Singing voice separation with deep U-Net convolutional networks*. 18th International Society for Music Information Retrieval Conference. Suzhou, China. 23-27 Oct 2017. https://openaccess.city.ac.uk/id/eprint/19289/1/7bb8d1600fba70dd79408775cd0c37a4ff62.pdf

[2] Ronneberger, O et al. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* University of Freiburg, Germany. 18 May 2015. https://arxiv.org/pdf/1505.04597.pdf

[3] "MICCAI BraTS 2017: Scope | Section for Biomedical Image Analysis (SBIA)". Perelman School of Medicine, University of Pennsylvania. 2017. https://www.med.upenn.edu/sbia/brats2017.html

[4] Stoller, D et al. *Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation.* Queen Mary University of London. 8 June 2018. https://arxiv.org/pdf/1806.03185.pdf

[5] Chollet, F. “Image segmentation with a U-Net-like architecture”. *Keras*. Created 20 March 2019. Last Updated 20 April 2020. https://keras.io/examples/vision/oxford_pets_image_segmentation/

[6] Bittner, R et al. “MedleyDB 2.0: New Data and a System for Sustainable Data Collection”. New York, NY, USA: International Conference on Music Information Retrieval (ISMIR-16).

[7] Liutkus, A et al. “DSD100”. GitHub. Last updated 2019. https://sigsep.github.io/datasets/dsd100.html

[8] Liutkus, A et al. Edited by Tichavski, P et al. “The 2016 Signal Separation Evaluation Campaign”. *Latent Variable Analysis and Signal Separation*. Pg 323-332. Published by Springer International Publishing. Proceedings of the 12th International Conference on Music Information Retrieval. Liberec, Czech Republic. 25-28 August 2015.

[9] Manilow, E et al. “Generating mixtures with Scraper”. *GitHub*. Created 2020.

https://source-separation.github.io/tutorial/data/scaper.html

[10] Zhang, A et al. Singing Voice Separation with U-Net. University of Victoria, Victoria. 13 December 2020. [Online]. Available: https://github.com/Zhz1997/Singing-voice-speration-with-U-Net.
[11] Raffel, C et al. “mir_eval Documentation”. GitHub. https://craffel.github.io/mir_eval/

[12] Raffel, C et al. *mir_eval: A Transparent Implementation of Common MIR Metrics*. Proceedings of the 15th International Conference on Music Information Retrieval. 2014. https://colinraffel.com/publications/ismir2014mir_eval.pdf

[13] Emiya, V et al. "Subjective and Objective Quality Assessment of Audio Source Separation," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 7, pp. 2046-2057, Sept. 2011, doi: 10.1109/TASL.2011.2109381.

[14] Chan, T-S et al. “Vocal activity informed singing voice separation with the iKala dataset”. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 718–722. IEEE, 2015.