Audio Source Separation Using Deep U-Net

EE698R Course Project

Pratikhya Ranjit
Roll Number:190639
EE Department, IIT Kanpur

Rishabh Katiyar
Roll Number:190702
EE Department, IIT Kanpur

***Abstract*—In this project we perform audio source separation by separating vocal and instrumental components of audio signals. We compare the performance of the U-Net proposed by Jansson et al [1] and the baseline model on the MIR-1K dataset. We apply data augmentation to analyse the performance of models. The models are evaluated quantitatively using SDR, SIR and SAR metrics and qualitatively through listening.**

I. Introduction

In Audio Source Separation the audio which is to be examined can be described in terms of a main melody line (foreground) and the accompaniment (background) provided by the background musicians. In this paper we will focus on separating the singer from a musical mixture that contains one or more musical instruments. There are many different use cases of audio source separation in the field of Music Information Retrieval, some being : Music classification, Lyrics transcription, vocal activity detection and many more.

Audio can be represented in different forms. It can be represented as a waveform sampled at a particular frequency known as the Sampling Frequency (F_s). A waveform is a continuous time signal discretized in both time and amplitude. The signal can be monophonic or mono, if there is only one audio channel of the shape $x \in \mathbb{R}^{TX1}$. We say a signal is stereophonic, or stereo, if the array has two channels of shape $x \in \mathbb{R}^{TX2}$.

The audio signal can also be represented as a time-frequency representation or Short-time Fourier Transform (STFT). The absolute value of a Time-Frequency (TF) bin $|X(t, f)|$ at time t and frequency f determines the amount of energy heard from frequency f at time t. Importantly, each bin in our STFT is complex, meaning each entry contains both a magnitude component and a phase component. Both components are needed to convert an STFT matrix back to a waveform so that we may hear it. The STFT is invertable, meaning that a complex valued STFT can be converted back to a waveform. This is called the inverse Short-time Fourier Transform or iSTFT.

Masking is used to separate a single source from a multi-source audio. In source separation approach, a single mask is created to separate one source from the audio. A mask is a matrix that is the same size as a spectrogram and contains values in the inclusive interval [0.0,1.0]. Each value in the mask determines what proportion of energy of the original mixture that a source contributes. So a mask basically acts like a filter. For a particular TF bin, a value of 1.0 will allow all of the sound from the mixture through and a value of 0.0 will allow none of the sound from the mixture through.

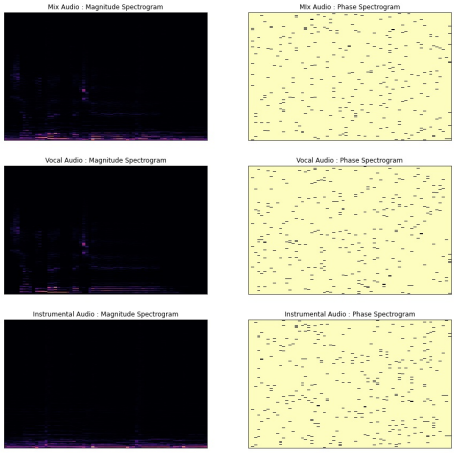

Fig. 1. Magnitude & Phase spectrograms of different types of single channel audio: Mixed audio, vocal audio, instrumental audio

We apply a mask to the original mixture audio by element-wise multiplying the mask to the mixture spectrogram. So if a mask, $\hat{M}_i \in [0.0, 1.0]^{TXF}$, represents the i^{th} source, S_i, in a mixture represented by a magnitude spectrogram, $|Y| \in R^{TXF}$, we can make an estimate of the source like so:

$$S_i = \hat{M}_i \odot |Y| \tag{1}$$

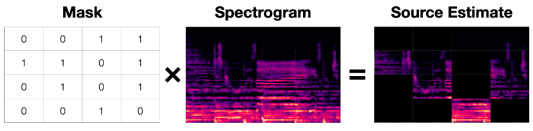

Fig. 2. Element-wise multiplication of the mask with the spectrogram to produce the source estimate

We do not model phase in our machine learning model but only the magnitude spectrogram. For a mask-based source separation approach, we copy the phase from the mixture and thus we can convert our source estimate back to a waveform. From (1), S_i represents the magnitude spectrogram of our source estimate. Now we can just copy the phase from the mixture over to the magnitude spectrogram of our source estimate, S_i:

$$\tilde{S}_i = S_i \odot e^{j.\angle Y} \tag{2}$$

where we use $j = \sqrt{-1}$, "$\angle$" to represent the angle of the complex-valued STFT of Y, and $\tilde{S}_i \in \mathbb{C}^{TXF}$ to indicate that the estimate for Source *i* is now complex-valued similar to an STFT.

Quantitative evaluation of our model's output is done using Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), and Source-to-Artifact Ratio (SAR). An estimate of a Source $\hat{s}_i$ is assumed to actually be composed of four separate components,

$$\hat{s}_i = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{3}$$

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

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

This is interpreted as the amount of unwanted artifacts a source estimate has with relation to the true source.

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

This is interpreted as the amount of other sources that can be heard in a source estimate.

$$SDR = 10log_{10}(\frac{||s_{target}||^2}{||e_{interf} + e_{noise} + e_{artif}||^2}) \tag{6}$$

SDR is usually considered to be an overall measure of how good a source sounds.

II. Related Work

Several techniques have been proposed for source separation of musical audio. Successful results have been achieved with non-negative matrix factorization, Bayesian methods, and the analysis of repeating structures. Deep learning models have recently emerged as powerful alternatives to traditional methods. *Wave-U-Net* is an extension of the U-Net architecture that operates directly on waveforms. *Open-Unmix* is a more recent neural network architecture that boasts impressive performance. *Deep Clustering* maps each TF bin to a high-dimensional embedding space such that TF bins dominated by the same source are close and those dominated by different sources are far apart. The U-Net architecture used in this paper adapts the U-Net [1] architecture proposed by *Jansson et al* for the task of vocal separation. The architecture was introduced in biomedical imaging, to improve precision and localization of microscopic images of neuronal structures.

III. Methodology

We have used a Machine Learning Based Approach particularly the Convolutional Neural Networks for Audio Source Separation. In a convolutional layer, the input is convolved with filters so that the output matrix contains important features extracted from the input. U-Net performs a series of 2D convolutions, each of which produces an encoding of smaller and smaller representation of the input. The small representation at the center is then scaled back up by decoding with the same number of 2D deconvolutional layers (sometimes called transpose convolution), each of which corresponds to the shape of one of the convolutional encoding layers. Each of the encoding layers is concatenated to the corresponding decoding layers.
The dataset provided to us was the MIR-1K dataset which consisted of 1000 song clips & 2 channels in which the music accompaniment and the singing voice were recorded at left and right channels, respectively. U-Net inputs a spectrogram hence the dataset which consisted of wave files was first downsampled to 8192 Hz to reduce the input size. Then we computed the magnitude spectrogram for each wavfile with n_fft = 1024, hop length of 768 and a window size of 1024. The magnitude spectrogram is fed as input to our model which in turn produces the mask of the same size as input. The final mask is multiplied by the input mixture which is the output of our model and the loss is taken between the ground truth source spectrogram and mixture spectrogram with the estimated mask applied.

To compare the model's performance we took many approaches. Firstly,we pre-processed the MIR-1K dataset as described above. We prepared the training, validation and test dataset by randomly shuffling wavefiles in the original dataset. Next, we created a new dataset by putting some selected audio files in validation and test datasets and rest in training dataset. The second dataset was such that it ensured the model was never trained on the same song clip of the same singer on which it was being tested and validated. This guaranteed a fair evaluation of performance of our model unlike with the first dataset. Next, we did some data augmentation by adding random noise and changing the pitch of some of the audio files to train our model with more data. We trained both the U-Net Model and the baseline model on each of these datasets and analysed their performance.

We then qualitatively as well as quantitatively assessed the performance of our model. We manually listened to some of the audio files to test if the model is separating vocals from the accompanied musical instruments or not. We calculated the Signal to Distortion ratio(SDR), Signal to Interference ratio(SIR) & Signal to Artifact Ratio(SAR) to

measure performance. We computed performance measures using the mir_eval toolkit [2].

A. Network Architecture

The architecture consists of two blocks, first the encoder block and then the decoder block. The encoder block consists of a stack of convolution layers where each layer halves the size of the image but doubles the number of channels thus encoding the spectrogram into a small and deep representation. That encoding is then decoded to the original size of the spectrogram by a stack of upsampling or deconvolutional layers. Each deconvolutional layer doubles the input size and halves the number of channels. The U-Net adds additional skip connections between layers at the same hierarchical level in the encoder block and decoder block. This allows low-level information to flow directly from the high-resolution input to the high-resolution output.

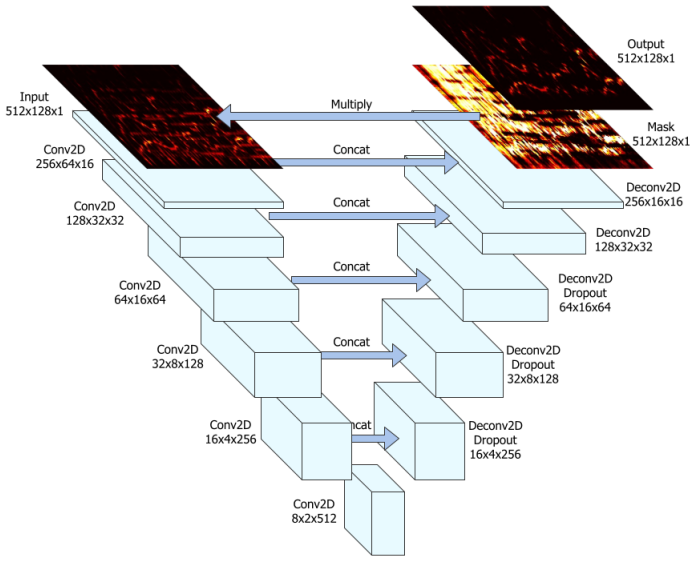

Fig. 3. U-Net Architecture taken from [1]

The Encoder block consists of 6 strided 2D convolution layers with same padding of stride 2 and kernel size 5x5 with batch normalization and activation function as leaky ReLu with leakiness 0.2. The Decoder block consists of 5 strided deconvolution layers with stride 2, kernel size 5x5 with batch normalization and activation function as ReLu with a dropout of 50% in the first 3 layers. The Final layer has a sigmoid activation function that makes the mask. The optimizer used is Adam.

The Loss function is L_1norm loss

$$L(X;Y;\theta) = ||f(X,\theta) \odot X - Y||_{1,1} \tag{7}$$

where $f(X,\theta) \odot X$ is the output of the network model where $f(X,\theta)$ is the mask generated that is applied to the input X which is the input mixture spectrogram, while Y is the target spectrogram.

In the original U-Net implementation by *Jansson*, the final layer of the model outputted only the mask $f(X,\theta)$. The model trained itself by calculating L_1-loss between $f(X,\theta) \odot X$ and input spectrogram. In our paper, we have introduced a variation where the model itself computes the $f(X,\theta) \odot X$ and outputs the predicted spectrogram for the vocals. The model then trains by simply computing L_1-loss between the output and input spectrograms.

The Base Model Architecture was also similar to the U-Net Architecture with the only difference that the skip connections on each level between the encoder and decoder block were removed. This was done to show the utility of the skip connections. The rest of the architecture was same as of the U-Net with our variation.

B. Training

The model is fed the magnitude spectrogram of mixed audio as input and is then trained for 20 epochs. The training loss as well as the validation loss keep on reducing as the number of epochs increased from which we can infer that the model is learning how to create the masks to separate the vocals from the background music. For getting the masks for the accompanied music we are not training a separate model, instead after getting the predicted estimate of the magnitude spectrogram of the vocals, we are doing a element-wise subtraction of the estimate from the magnitude spectrogram of the mixture to get the predicted estimate for magnitude spectrogram of the accompaniment or the background music.

$$\hat{S}_{accompaniment} = X - f(X,\theta) \odot X \tag{8}$$

where $\hat{S}_{accompaniment}$ is the magnitude spectrogram of the accompaniment background music, X is the magnitude spectrogram of the mixture(vocals + accompaniment) and $f(X,\theta) \odot X$ is the predicted magnitude spectrogram of the vocals.

C. Dataset & Experiments

1) Dataset: The MIR(Music Information Retrieval)-1K dataset is developed for the task of singing voice separation. It contains 1000 audio clips extracted from randomly selected Chinese pop songs. Each audio clip in the dataset consists of two channels, the left channel consisting of the instrumental component and the right channel consisting of the voice component.

In order to train the U-Net model for voice separation, we must have three types of audio dataset: the mixed audio, vocal component of audio and the instrumental component of the audio. Since the MIR-1K dataset has different vocal and instrumental component channels, we can separate them easily into single channel audios. The mixed audio is computed as the element-wise sum of the voice and instrumental channels.

2) Experiments:

1) In order to build a robust model, we create a bigger dataset using data augmentation techniques like noise addition and pitch-shifting. We compare the performance of our model by training on both datasets in order to have a clearer picture of the effect of augmented dataset on models.
2) The MIR-1K dataset consists of audio clips with a single singer singing at a time. We train our model using that data only. In order to find out how the model behaves with different kinds of input, we test our model with input audio files with two people speaking simultaneously in different languages.

IV. Evaluation

The U-Net model proposed by Jansson et al. was tested on Medley DB and iKala dataset. The evaluation scores of the model on both the datasets is given in Table I.

Table II, III & IV shows the various evaluation scores for the U-Net Model and the Baseline Model for the datasets. Table II shows for the original dataset. Table III shows for the new dataset created by putting some selected audio files in validation and test datasets and the rest in training dataset. Table IV shows the scores for the augmented dataset created by adding random noise and changing the pitch of some of the audio files.

We qualitatively evaluated our model for two persons speaking together by listening to its output and found that the model was able to separate both the voices by suppressing background noise properly. The output was comparable to single voice audio results.

TABLE I
ORIGINAL U-NET MODEL MEAN SCORES

	MedleyDB	iKala
NSDR vocal	8.681	11.094
NSDR instrumental	7.945	14.435
SIR vocal	15.308	23.960
SIR instrumental	21.975	21.832
SAR vocal	11.301	17.715
SAR instrumental	15.462	14.120

TABLE II
ORIGINAL MIR-1K DATASET MEAN SCORES

	U-Net(M1)	Baseline(M6)
SDR vocal	11.158	7.866
SDR instrumental	10.167	6.942
SIR vocal	19.604	13.367
SIR instrumental	17.819	12.126
SAR vocal	11.995	9.728
SAR instrumental	11.179	9.068

The graphs for the losses of various models are plotted below

TABLE III
NEW MIR-1K DATASET(WITHOUT AUGMENTATION) MEAN SCORES

	U-Net(M2)	Baseline(M4)
SDR vocal	8.052	5.692
SDR instrumental	7.154	4.606
SIR vocal	14.249	10.562
SIR instrumental	13.701	9.225
SAR vocal	9.707	8.152
SAR instrumental	8.690	7.436

TABLE IV
NEW MIR-1K DATASET(WITH AUGMENTATION) MEAN SCORES

	U-Net(M3)	Baseline(M5)
SDR vocal	4.799	2.703
SDR instrumental	6.242	3.950
SIR vocal	11.185	7.030
SIR instrumental	11.594	7.472
SAR vocal	7.596	7.819
SAR instrumental	11.807	12.051

V. Conclusion and Future Work

The U-Net Model gives satisfactory results as can be seen from the table of scores provided. The performance of U-Net on the original MIR-1k dataset shows that the vocals are nicely separated from the mixed signal. We can see from the table that on the changed dataset that ensured the model was never trained on the same song clip of the same singer on which it was being tested and validated, the performance degraded a bit but is still satisfactory. We tested the model on our own voice recordings with instrumental music being played in the background and found that the model was able to separate our voice from the instrumental music. All the three tables show that U-Net significantly outperforms the Base Model on all the 3 measures except the SAR measure on the Augmented Dataset. We feel that it happens because on augmenting the dataset by adding noise and changing the pitch produces artifacts in the original data and hence a lower SAR. The skip connections may flow this information from the convolutional block to the deconvolutional block. Since there are no skip connections in the baseline model hence the baseline model performs better than U-Net here.

The loss functions clearly show that the model is learning how to separate the vocals from the mix audio. Again a clear distinction can be seen between the U-Net and the Base Model, where the loss at which the Base Model settles is higher than that of the U-Net Model.

For future work, we will train our models on various audio datasets and analyse the performance on them. Due to a constrain on the computational resources, we could not train our model on a very large dataset and thus did not augment the whole dataset. We will inspect the performance of our model by training on a large dataset in the future.

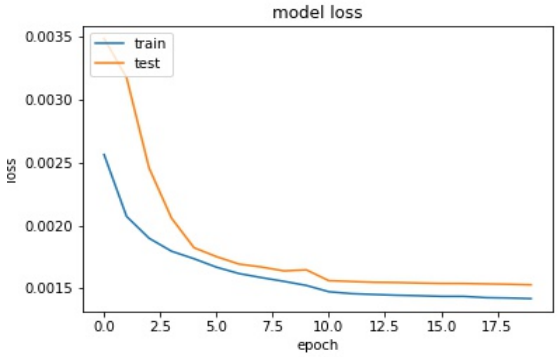

Fig. 4. Old Dataset: M1 train and validation loss

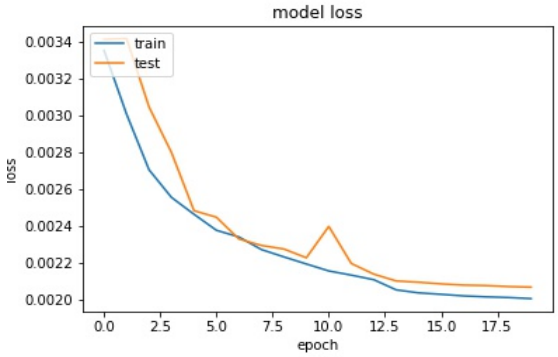

Fig. 5. Old Dataset: M6 train and validation loss

REFERENCES

[1] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks." October 2017.

[2] C. Raffel, B. Mcfee, E. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. Ellis, "mir$_eval$: $Atransparentimplementationofcommonmirmetrics$," 102014.

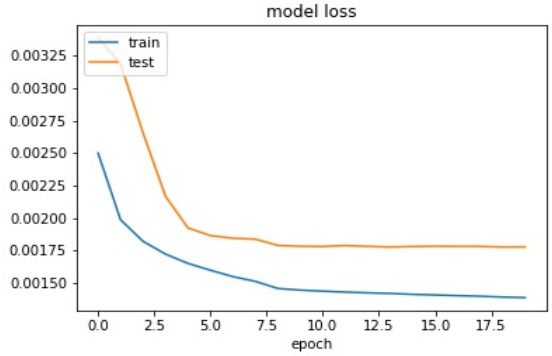

Fig. 6. New Dataset(without augmentation): M2 train and validation loss

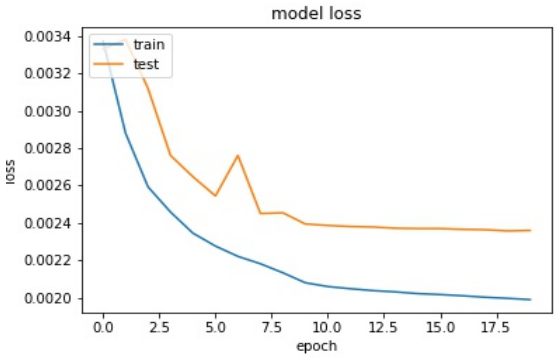

Fig. 7. New Dataset(without augmentation): M4 train and validation loss

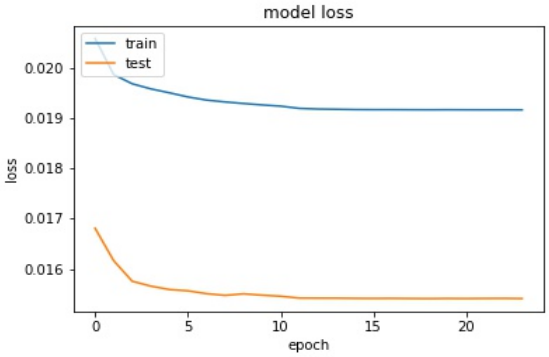

Fig. 8. New Dataset(with augmenatation): M3 train and validation loss

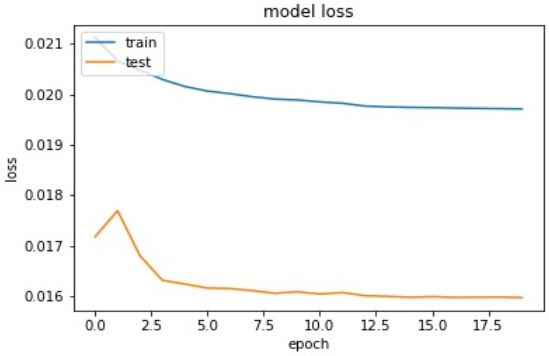

Fig. 9. New Dataset(with augmenatation): M5 train and validation loss