Deep Learning to Isolate Vocals from Stereo Music - Signal processing for AI in Audio Final Project Report

Idan Kashtan
Computer Science Department,
Reichman University
idan.kashtan@post.runi.ac.il

Abstract

Source separation is a field of signal processing that aims to recover or reconstruct one or more source signals. Through some linear or convolutive process, have been mixed with other signals. It has numerous applications in music, speech, and audio processing, including music remixing, noise reduction, and speech recognition. Source separation is a challenging problem because the sources often overlap in the time-frequency domain and share common spectral characteristics. Deep learning techniques, particularly convolutional neural networks (CNNs), have shown promising results in source separation tasks. In this project, I focus on using a U-NET-like architecture to separate vocals from music recordings. I use the MUSDB18 dataset, which contains professionally produced multitrack music recordings, for training and testing the model. I compare my model's performance against a model called Spleeter, created by Deezer, to evaluate its effectiveness.
Link to my notebook: https://colab.research.google.com/drive/1JkLIl--72RQlPTkdEBYgIaTbkp5Yos_O?usp=sharing
Link to the presentation: https://drive.google.com/file/d/1GYiI2ym8-g_2o4NF9aXL2uhJmjBgHNZP/view?usp=sharing

1 Related Works

The U-Net architecture was first introduced in the context of biomedical image segmentation by Olaf Ronneberger et al. (2015), where it achieved state-of-the-art results on several benchmark datasets. Since then, the U-Net architecture has been widely used in computer vision tasks, such as object detection, semantic segmentation, and image restoration. In audio processing, the U-Net architecture has been used for music source separation, where it has achieved promising results. Another great work in the field is Wave-U-Net by Daniel Stoller et al. (2018). Their work is an extension of the U-Net architecture that operates directly on the waveform. They use a series of 1D convolutions on the audio instead of 2D convolutions on the spectrogram. Another work in source separation is Spleeter by Deezer, a deep learning-based source separation system that can separate vocals, drums, bass, and other sources from stereo music recordings. Spleeter uses a U-Net architecture and achieves state-of-the-art performance in various source separation tasks.

2 Architecture

The architecture used in this project is based on the U-Net network, which has shown promising results in various image and audio processing tasks. The U-Net architecture (Figure 1) is a fully convolutional network that consists of an encoder and a decoder networks, connected through skip connections that concatenate the feature maps from the encoder to the corresponding decoder layers. The encoder network consists of a series of convolutional layers followed by max-pooling layers for downsampling, which allows the network to learn increasingly abstract features at multiple scales.

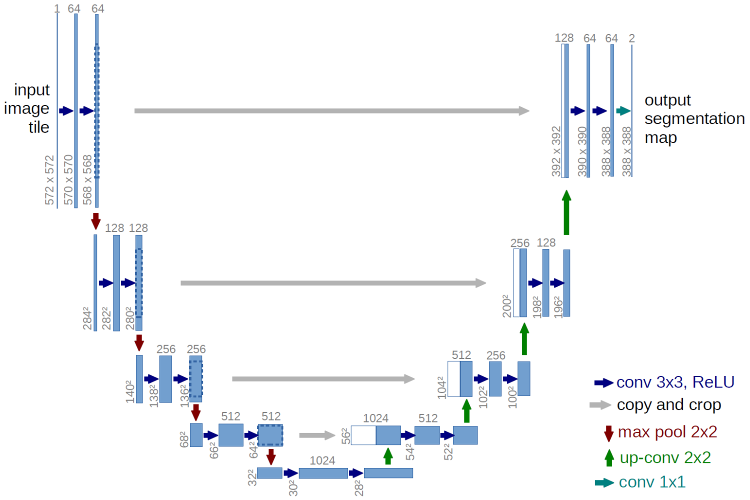


Figure 1: The original U-Net architecture from the paper.

The decoder network uses transposed convolutional layers for upsampling, allowing the network to reconstruct the original resolution of the input. The skip connections help to preserve the spatial information and finer details of the input, enabling the decoder to use the high-level features learned by the encoder to reconstruct the individual sources from the mixture.

Soft masks A common way to separate a source from the mixture is using a mask on the TF representation. In this project, I focused on a type of mask called "Soft Masks (Ratio Masks)" they take values between [0.0, 1.0], trying to assign only part of the energy from the mix to the source. In this project, the mask is calculated like so:

$$mask = \frac{target_mag}{Max(target_mag, mixture_mag) + 1e-9}$$

This formula calculates a soft mask by dividing the magnitude spectrogram of a target audio source by the element-wise maximum of the target and mixture magnitude spectrograms plus a small constant to avoid division by zero errors.

Data Augmentation When using a small dataset like MUSDB18, it's essential to use data augmentation to help the model learn the desired invariance and robustness properties. To achieve that, I created more augmented data by mixing STEMs from different songs and adding random white noise to the mixture.

3 Experiments and Results

I evaluated the performance of the model on singing voice separation. The model was trained on the MUSDB18 dataset with additional augmented data, as described earlier. The model was trained to estimate the mask applied to the mixture. It returns an object containing a mask and the estimation of the vocals. During training, I used L1 loss from Nussl, also known as the Absolute Error. I used an ADAM optimizer with a learning rate of 1e-4, weight decay = 1e-5, a batch size of 8, and I use Xavier as the initialization of the model . In order to prevent overfitting, I used drop out layers and an early stopping after 15% of the total epochs if the model didn't improve on the validation set. The input of the model is the magnitude of the mixtures' spectrogram with the shape (8, 2, 1025, 587) where the first dimension is the batch number, the second is the number of channels, the third is the number of frequency bins, and the last one is the time steps.

We can see in figure 2 that the model learned and applied a mask to the input to reduce the energy from the original mixture to try to match the vocals' magnitude. The model still needs to learn more to get a perfect estimation or "Ideal mask" for now, we can observe that the model can identify the

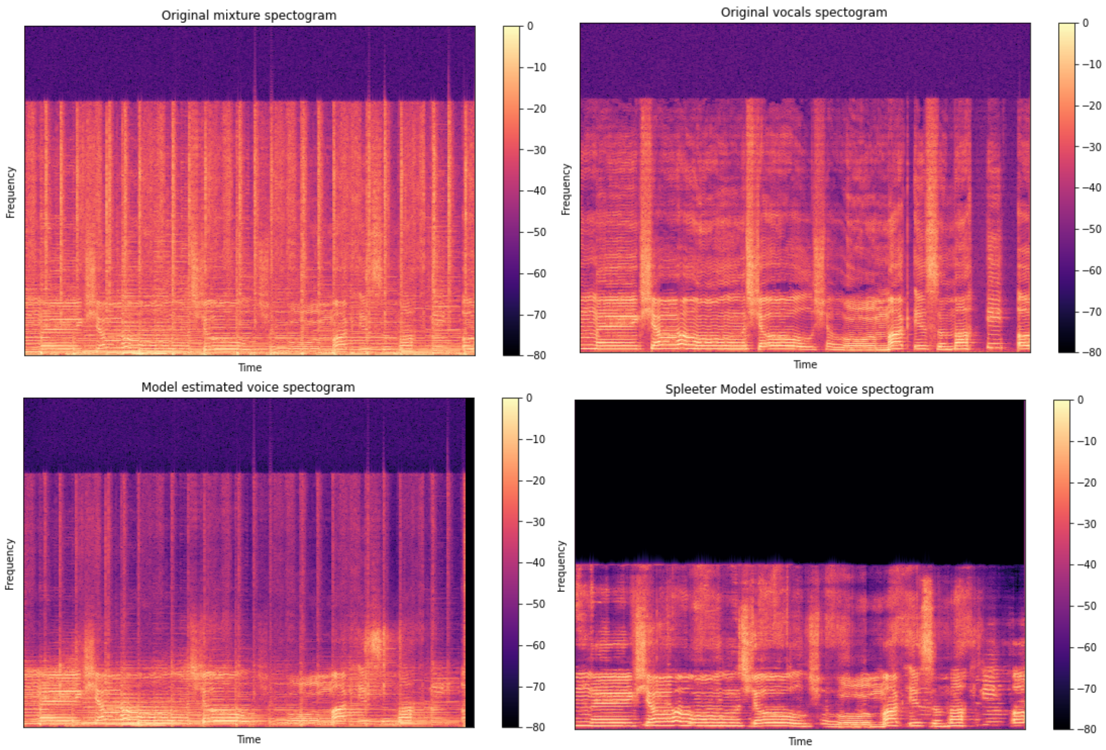


Figure 2: Sample spectrogram of the original mixture/vocals compare to the model's estimation.

vocals from the mixture, and it tries to reduce the noise. Inside the notebook, under "Test the model" and "More examples from the model", you can find examples demonstrating the model estimation.

3.1 Subjective Evaluation

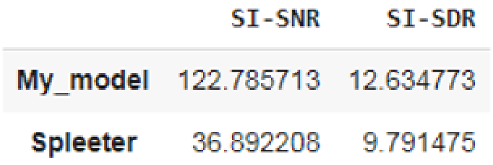
When using subjective evaluation, we want to listen to the estimation and try to evaluate the models' performance by hearing. We can hear from the notebook that the model is dealing with low notes quite well. It removes the bass sound from the original mixture. On the other hand, it has a hard time removing the higher notes like the guitar sound or the snare.

3.2 Objective Evaluation

When we want to compare two source separation models usually, we use some quantitative metrics like SDR, SNR, SI-SDR, SI-SNR, PESQ, and more. It's important to note that using these metrics gives you a rough idea of how good the estimation sounds. It doesn't capture everything, so we need to use a combination of the two evaluation methods. To use objective evaluation, we want to ensure that the waveforms are in a suitable format for calculation and that any differences in performance between the two models are not due to differences in the waveforms. To ensure that, I preprocessed the waveforms by centering and normalizing them. We can see in table 1 that my model got better results in both SI-SNR and SI-SDR. But when you listen to the estimation of Spleeter, we can hear the vocals much better than in my models' output. I need to further investigate these results.

4 Conclusion

In this project, I proposed a U-Net-based source separation model for isolating vocals from stereo music. The model learned a soft mask to apply to the mixture magnitude to assess the correct amount of energy of the vocals. I used data augmentation to help the model generalize and describe the training process. After that, I talked about evaluation methods and the models' results. In addition,



	SI-SNR	SI-SDR
My_model	122.785713	12.634773
Spleeter	36.892208	9.791475

Figure 3: Objective evaluation between my model and Spleeter. I used SI-SNR and SI-SDR.

the proposed model architecture and training procedure can be adapted and extended to other audio source separation and audio processing tasks, such as speech separation, denoising, and enhancement.

4.1 Future work

In future work, I plan to investigate several directions for improving the performance and robustness of the proposed model. First, I plan to explore the use of additional input features, such as the raw waveform or spectrogram phase, to enhance the models' ability to separate sources with complex temporal and spectral structures. Second, I plan to investigate different architectures for the encoder and decoder modules, such as dilated convolutions, attention mechanisms, and residual connections, to improve the model's expressiveness and scalability. Third, I plan to experiment with various data augmentation techniques, such as time and frequency masking, pitch shifting, and audio effects simulation, to increase the model's robustness to different types of input signals and environmental conditions.

References

1. `https://arxiv.org/pdf/1505.04597.pdf`
2. `https://arxiv.org/pdf/1806.03185.pdf`
3. `https://towardsdatascience.com/audio-ai-isolating-vocals-from-stereo-music-using-convolut`
4. `https://source-separation.github.io/tutorial/basics/tf_and_masking.html`