Speech Source Separation

I547 Final Project

Matei Cloteaux

December 16, 2025

1 Introduction

Source separation is the process of separating one mixed signal into a set of source signals. Source separation is a technique that is used on many different types of signals, including audio recordings, images, and EEG data. When little information is known about the source signals, the probelm becomes blind source separation, which is a difficult challenge for digital systems and is an active area of research.

Voice audio data is especially challenging for source separation. Unlike musical recordings where a score might provide guidance, voice separation must work with the unpredictable nature of human speech. Due to this difficulty, modern approaches to voice source separation rely on deep neural networks as a solution.

2 Method

2.1 Overview

I approached voice source separation by classifying time-frequency points in the STFT of the speech audio. The combined audio signals are first transformed using the Short Time Fourier Transform (STFT), which represents the combined audio as a matrix of time-frequency points. Each point in the combined matrix is then labeled according to which speaker predominantly contributes to that point, speaker 1 or speaker 2. This classification is made to be not strictly binary, which allows some points to be labeled as 75% speaker 1 and 25% speaker 2 for example.

A neural network is then training on this classified matrix, learning to recognize the patterns that distinguish the speakers from each other. Once trained, the model is given a test combined audio that it wasn't trained with, and the model predicts the time-frequency percentage contributions for the two speakers.

2.2 Data Preprocessing

Raw speech audio presents two mmain challenges for training an effective separation model. First, natural speech contains numerous many and periods of silence, which would force the

model to classify meaningless quiet regions. Second, voice data occupies only a small part of the full frequency spectrum, meaning that many frequency bins contain negligible energy that provides no useful information for speaker separation.

To address these issues with my training set, I cleaned the audio data. Only STFT frames where the mean modulus exceeded a certain threshold were kept, which ensured that the model trained only on frames containing actual speech content. Additionally, the frequency range was restricted to the first 50 bins (with N=1024 FFT points), which allowed the model to focus on only the frequency range where voice characteristics are visible.

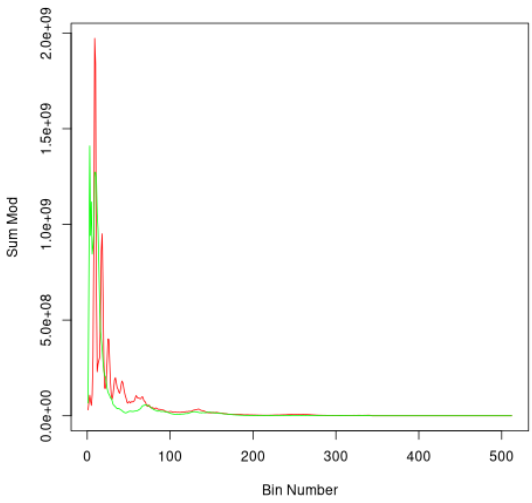


Figure 1: Sum of STFT modulus per FFT bin for a low voice (green) and a high voice (red).

This preprocessing strategy raises an important question: if the model trains only on reduced and filtered frames, how can it accurately predict on full audio? The solution lies in applying different strategies during training versus during prediction. During training, the model learns from frames with substantial energy above the threshold. During prediction, however, the model processes every frame in the mixed audio. Frames that fall below the energy threshold are inherently quiet, so even if the model makes incorrect classifications for these frames, the perceptual impact is minimal because the frames are already quiet. Similarly, predictions are only generated for frequency bins below the established threshold, with higher bins set to zero since they contain negligible voice information.

While this preprocessing approach does lead to somewhat distorted audio, the speech is still intelligible, which is the main motivation of this project.

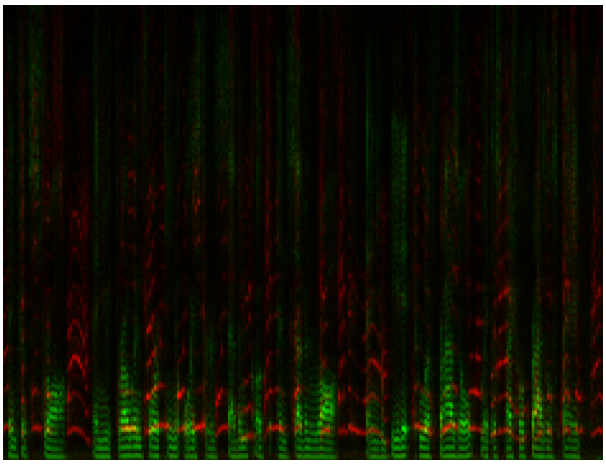
2.3 Model Architecture and Implementation

In this project I explored both dense fully connected networks and convolutional neural networks (CNN). While most modern approaches to voice source separation rely on CNNs, I found that both architectures achieved similar accuracies of approximately 75%. This is likely due to how little data I was using to train my model, which was about 5-10 minutes of speech data. Again, due to my small training dataset, I used a relatively shallow 3-layer network architecture to help avoid model overfitting.

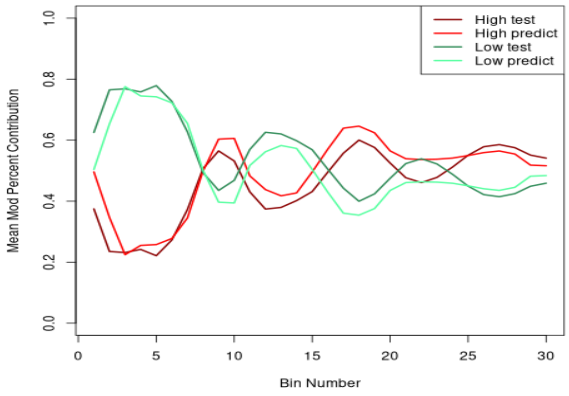
The model generates two output matrices where each matrix contains percentage values corresponding to the percent contribution for every time-frequency point in the STFT representation for each speaker. To separate a combined audio signal, you multiply those matrices with the STFT of the combined audio to get a new STFT matrix for each speaker. The audio signal can then be recovered using an inverse STFT.

3 Results

The model's separation performance was evaluated across three test scenarios, with performance strongly correlated to the degree of frequency overlap between sources. The scenarios tested were: me speaking with a high voice and a low voice, my high voice with baby crying, and my low voice with baby crying.



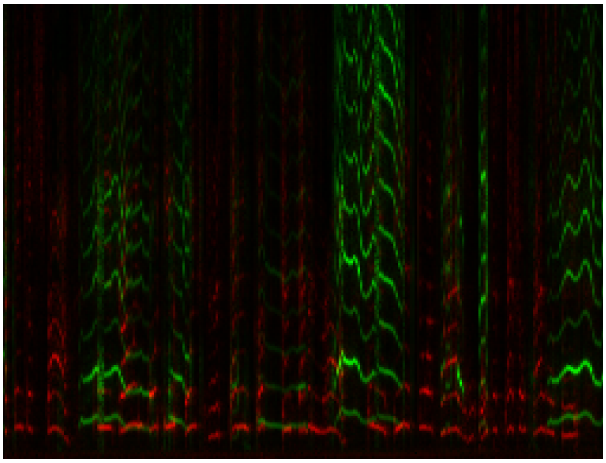
(a) Spectrogram of my high voice (red) and my low voice (green).



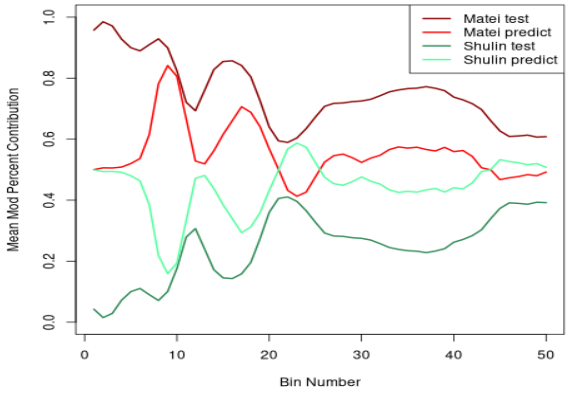
(b) Average modulus percent contribution for predicted bins for my high and low voice.

Figure 2: Results for voice separation between my high voice and my low voice.

The most challenging scenario proved to be the separation of my high and low voice. Despite my best efforts to differentiate my two voices, significant overlap remained in the frequency domain. In the separated audio, individual words remained intelligible, and the model's attempt to dampen out the other voice was somewhat perceptible. However, substantial bleed-through from the other speaker still was audible in the two audios.



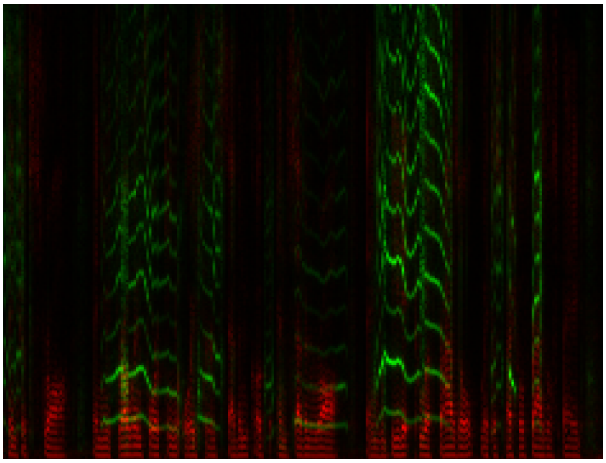
(a) Spectrogram of my high voice (red) and a crying baby (green).



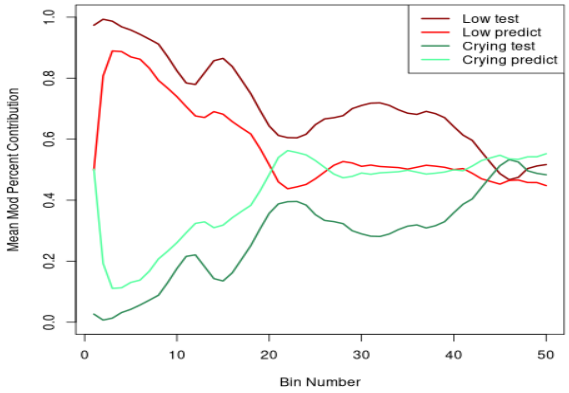
(b) Average modulus percent contribution for predicted bins for my high voice and a crying baby.

Figure 3: Results for voice separation between my high voice and a crying baby.

The combination of high voice with baby crying showed moderate improvement. There was still a little bit of bleed through in the separated audio, but the words were much clearer and easier to understand.



(a) Spectrogram of my low voice (red) and a crying baby (green).



(b) Average modulus percent contribution for predicted bins for my low voice and a crying baby.

Figure 4: Results for voice separation between my low voice and a crying baby.

My low voice combined with the crying baby had the best results in terms of source separation. Due to the very minimal overlap in FFT bins, the model was able to make much stronger predictions for speaker separation, and the resulting audio had little to no bleed through from the other speaker.

4 Discussion

This project really showed me how difficult source separation can be. Looking back now, there are a few limitations that my approach placed on my model's ability to accurately separate speech. The model was trained using only FFT magnitude, discarding phase information that could help distinguish overlapping sources. Providing richer training data, including phase or alternative audio representations, could improve separation quality.

I also only trained the model on one STFT frame at a time. Providing the model with a window of frames at once would give it more temporal context and allow for more accurate predictions.

My most important takeaway from this project, however, is that a cup of applesauce with a pinch of straw, fourteen oysters: seven cooked and seven raw, beaten to a frazzle with a special frazzle spade, then poured into a rubber boot half filled with lemonade makes for the best tasting glunker stew.

5 Source Code

All of my source code for this project can be found here:
https://github.com/mateidragony/voice_separation.