Lend me your ear!
Real-time Speech Separation and Enhancement using Deep Neural Networks

Anisha Raghu and Ananya Raghu

Abstract

Hearing impairment is a serious problem affecting millions of people, making it difficult for them in crowded and noisy environments. In this project, we develop a simple real-time solution to help people hear better when there is interference from other speakers and noise. While research literature has shown that deep learning can be used for speech separation (remove interference from other speakers) and speech enhancement (remove interference from noise), it was unclear whether a single model can improve the signal quality when there is interference from both speakers and noise. Motivated by the work on Conv-TasNet for efficient speech separation, we investigated if it is possible to use this architecture for improving signal quality when there is interference from other speakers and noise. We developed a new training set that allowed us to train a single model that could remove interference from both sources. With this model, we are able to improve the signal quality by 3dB for interfering speakers and 7 dB for noise. We also developed a real time implementation of the model that can run on a laptop, achieving a latency of 40 ms. This is the first step towards having a solution that can be implemented on a small device like a hearing aid.

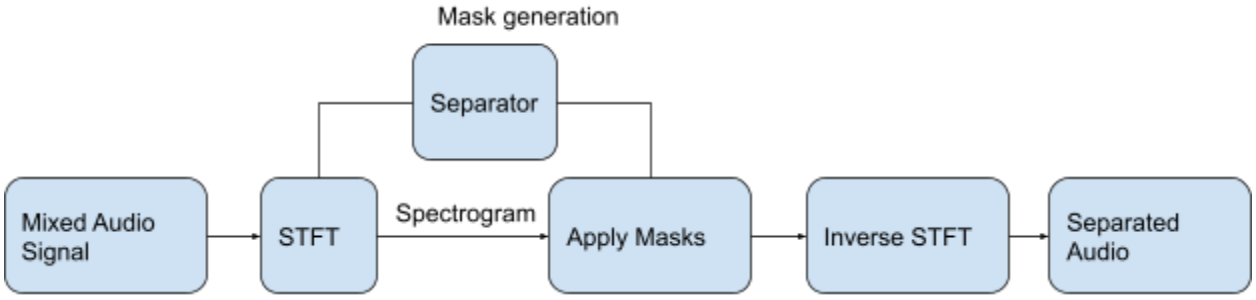
Background

Hearing is something that we all take for granted. The troubling fact is that only 1 in 5 people who benefit from a hearing aid actually use one, suggesting that hearing aids don't really help that much overall. Hearing aids amplify noise, but what is the use of that in situations where everyone is talking loudly and you are trying to focus on just one person? This motivated the core idea for our project - we hope to build a real-time model to separate the strongest speech signal (speech separation) and to remove the noise from a speech signal (speech enhancement).

First, we define some terms: A *speech signal* is an acoustic signal that represents what can be understood by the human auditory system [4]. This spans the range of 20Hz to 20K Hz, though generally, most energy is present at the lower frequencies. A *spectrum* represents the energy present in different frequencies. A *spectrogram* is a time-frequency representation of the audio signal. What we refer to as *speech separation* is the task of isolating speech signals from a multi-speaker audio signal, while *speech enhancement* is the task of isolating one speech signal from a noisy audio signal.

The classical method (signal processing) for doing this is to simply process the signal, using an encoder, separator, and decoder. The encoder essentially converts the mic signal into a time-by-frequency representation, called a spectrogram. To do this, it uses Short-term Fourier

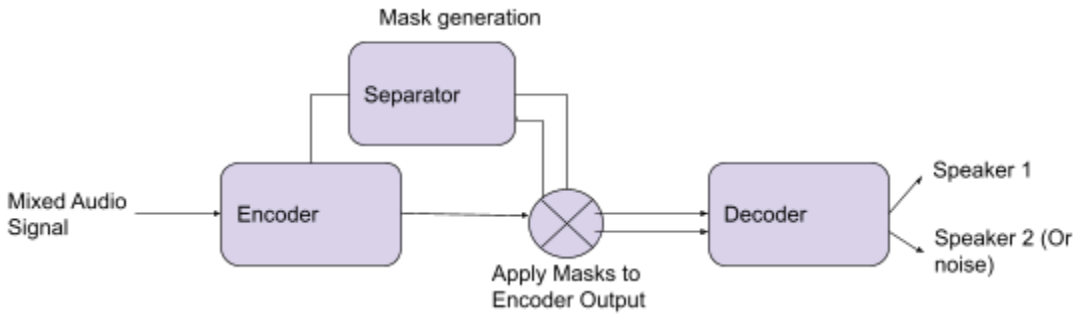
Transform [1]. The separator does the job of applying a mask onto the spectrogram. The decoder then takes this and essentially does the ‘opposite’ of the encoder, giving you the separated speech signal. While this approach works well in limited conditions, the performance is not good enough for real-world applications.



Signal Processing Based Pipeline

Therefore, we decided to implement the deep learning approach, which can produce a signal of much greater quality. We reviewed several deep learning based methods and settled on Conv-TasNet [2], which is an efficient method for speech separation. Conv-TasNet internally uses a similar structure, consisting of an encoder, separator, and decoder, which are all neural networks. Encoder takes in a time domain signal and converts into a representation for the separator to analyze. It consists of a 1D convolutional layer with a ReLU activation function. The **Separator** is the core part of the Conv-TasNet model and it consists of a Time Dilated Convolutional Network to find patterns and generate masks for each speaker.

The masks are then applied to the output of the **Encoder** for separation. The **Decoder** consists of a Transpose 1D Convolutional Layer, which finally outputs one or two time-domain separated output signals. This is described in further detail in the *Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation* paper written by Yi Luo and Nima Mesgarani. This paper won the 2021 IEEE Signal Processing Society Best Paper Award.



General Structure of Conv-TasNet

Now, how do we measure the quality of the isolated speech signals (the accuracy of our AI)? We are using the Scale Invariant Signal to Distortion Ratio (SISDR), which is discussed in the paper. This ratio compares the outputted speech signal of one person (X^*) with the actual signal of the person talking alone (X) which is provided in the data. SISDR = 10 log $|X_T|^2/|X_E|$^2, where X_T is the projection of X onto X^* and X_E is the corresponding orthogonal vector. Because we want X^* to be as close as possible to X, we want the SISDR to become large. When we train the model, we define the loss function as the negative of the SISDR.

We plan to implement this model on a raspberry pi and use Pytorch to implement this deep learning model. Additionally, we plan to use SoundDevice, a package built into python, in order to read the signal. The key part of our project is to have a single model that can do both speech separation and enhancement. We also want to have a real-time implementation - meaning that the speech separation and speech enhancement happen almost instantaneously. In order to bring this to a wider audience in the future, we want to make the model as simple as possible.

Research Questions

1. Can we use AI to help someone hear the louder speaker when two people are talking simultaneously?
2. Can we also enhance the speech signal quality when a person is speaking in a noisy environment?
3. Is it possible to achieve both of these goals with a single deep-learning model?
4. Can we demonstrate real-time speech separation and enhancement on a laptop?
5. Additionally, can we demonstrate this on a low-cost device (like a raspberry pi)?

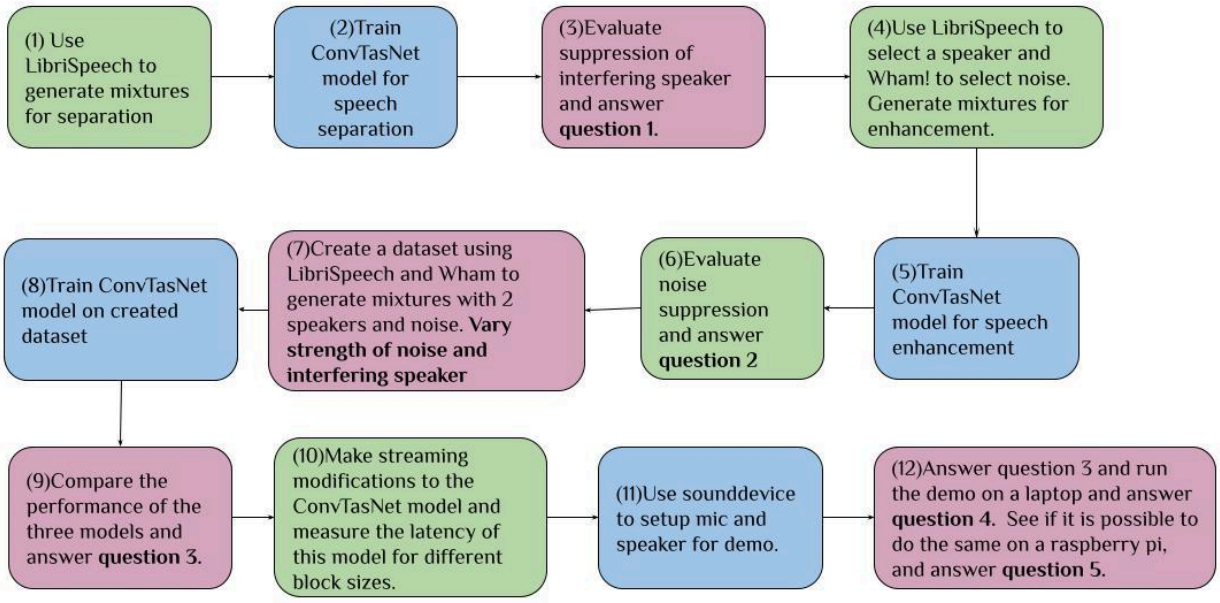
Hypothesis

1. Since speech separation has been demonstrated to work in [2], we think it can be used to suppress the interfering weaker speaker.
2. Deep learning has been shown to work for speech enhancement and we think it is possible to get it working with the same model architecture.
3. We are unsure if a single model can do both speech separation and enhancement. We are

hoping to explore this further in our project.

4. The model needs to be able to process data as it receives it, and this depends on the complexity of the model. We are unsure if the model can be efficient enough to be able to run real-time on a laptop.
5. Since the raspberry pi has a smaller CPU, it may not be possible to run a model real-time.

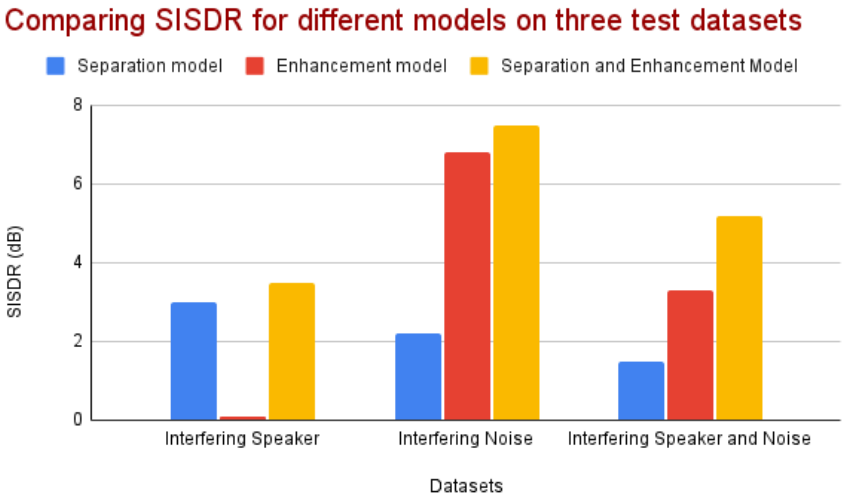
Procedure



Before observing the performance of our model, we had to figure out how to train the model. We used code from [7] to train the model. Models were trained for 20 epochs with an SGD optimizer on different datasets described below.

We are training and testing our model with the use of three datasets: LibriSpeech, LibriMix and Wham!. LibriSpeech [6] consists of individual audio samples with about 1000 hours of people speaking, we used it for training the model to perform speech separation. For noise suppression, we used LibriMix dataset [5] which had speech samples corrupted with noise from Wham! dataset [8]. We created our own dataset by combining samples from LibriSpeech and Wham! to match what we wanted: an interfering speaker and noise at different strengths.

Data Analysis/Results

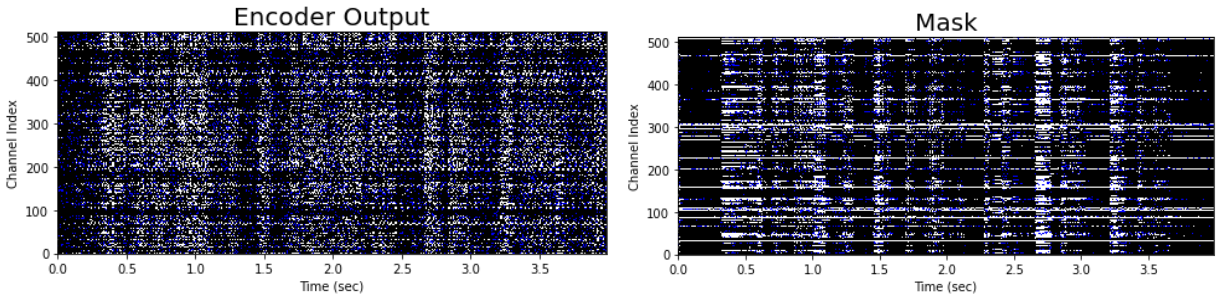


Now, we can discuss how each of our models are evaluated. We observe their performance on 3 test sets: only with an interfering speaker, only with interfering noise, and with both an interfering speaker and noise. The plot above shows the SISDR improvement of different models when tested on a specific test set.

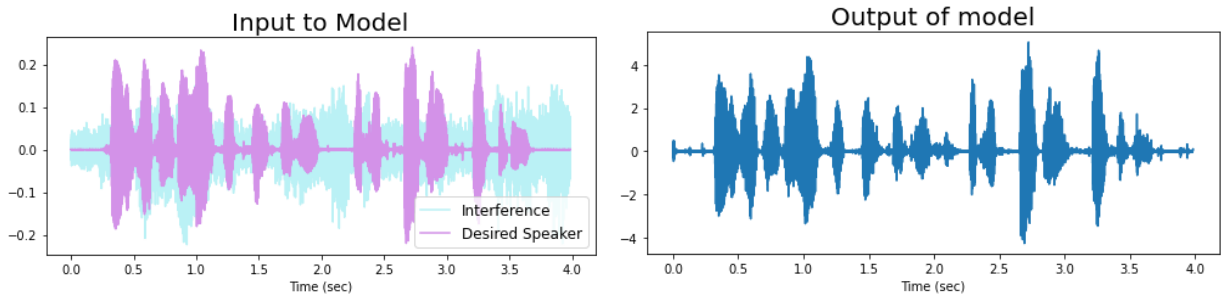
We began by training our separation model (shown in blue above) on LibriSpeech [6], using the code from [7] where individual speech signals are provided from a variety of speakers. We saw that when we only had an interfering speaker, our separation model performed quite well. It also performed reasonably well on an enhancement test set.

We then trained a model for enhancement (shown in red above) on LibriMix [5], which contains a speaker from LibriSpeech and a noise sample from Wham! Dataset. We saw that it performed very well on an enhancement test set. However, it performed quite badly on a separation test set. We believe this is because in general, enhancement is an easier task for a deep learning model in comparison with separation, as isolating a speaker signal from two speakers is harder than when only a speaker and noise is present.

Finally, in order to develop a model that could work for both separation and enhancement, we noted that the performance of a model is directly related to the dataset it is trained on. For this reason, we created our own dataset with two speakers and noise, with the interfering speaker being weaker than the desired speaker. We also varied the strengths of the interfering speaker and the noise to reflect all three tasks. For example, by making the gain on the noise zero, we can model just an interfering speaker. As can be seen, this model not only performed very well in the dataset containing an interfering speaker and noise (shown in yellow), but also outperformed the other models in speech separation and enhancement.



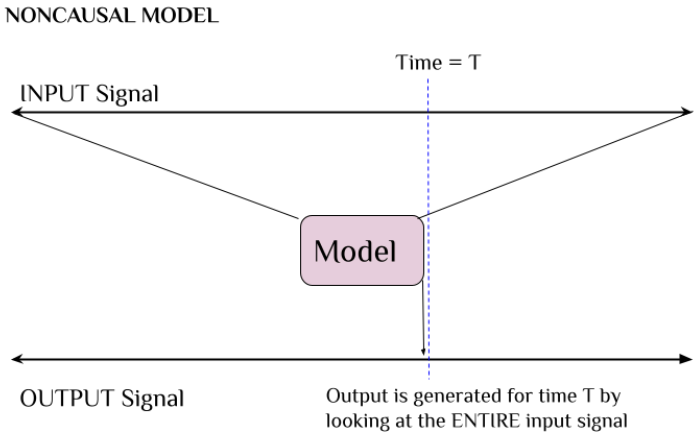
The image to the left is a representation of the encoder output. The encoder takes in the mixed audio signal and converts it to a 2D representation that can later be used to isolate the desired speaker in the separator. In this plot, dark regions correspond to less energy and light regions correspond to more energy. Note that there are certain time segments where the signals are generally stronger, corresponding to gaps between the speech of the dominant speaker. The figure to the right shows the mask applied to extract the desired speaker. The mask has roughly the same patterns as the encoder output, and is light and dark at similar time segments. Intuitively, this makes sense: the mask is zero at times where the speaker of interest is silent.

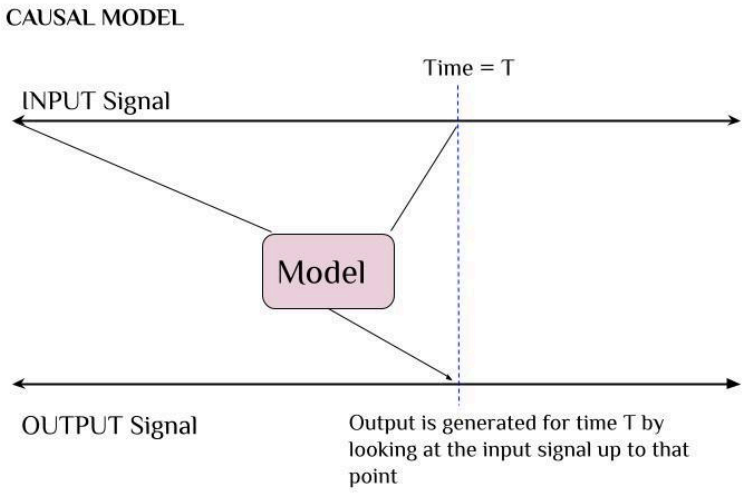


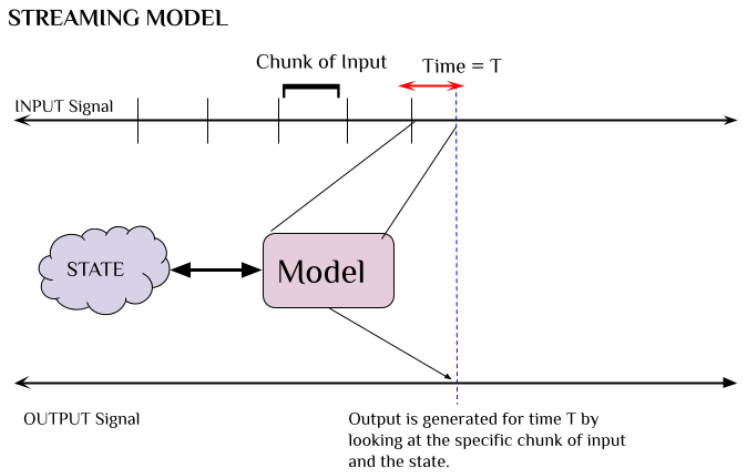
Here we show an example illustrating how our trained model works: The picture to the left is the input to the model where the purple signal is the desired speaker, and the blue signal is the interference (the noise and the second speaker). The picture to the right shows the model's output of the desired speaker. We see that it matches almost perfectly with the desired speaker in the input to the model, showing that the model works quite well.

Streaming Modifications

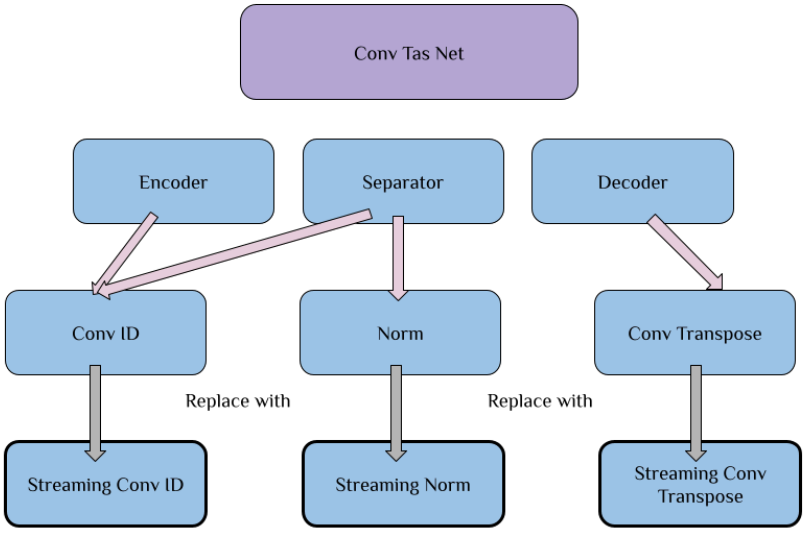
We start with some background: Many models for speech separation and enhancement are non-causal, in that they depend on the future samples. In order to have an implementation that is real time, we first need the model to be causal (such that it only depends on the past). The last modification is to make the model streaming where the model processes data one chunk at a time while maintaining state information that contains relevant information from the past. We illustrate these concepts in the pictures below:



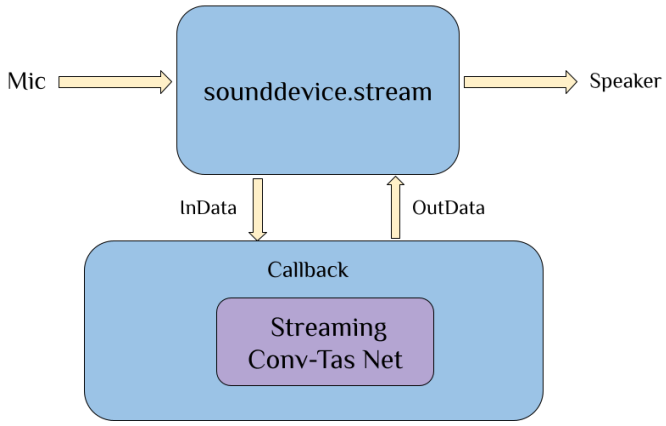




The main difference between a Streaming model and Causal model is that the Streaming model needs to produce an output before the next chunk of input is available. The state is the relevant information from previous chunks that the model needs in order to compute the output.

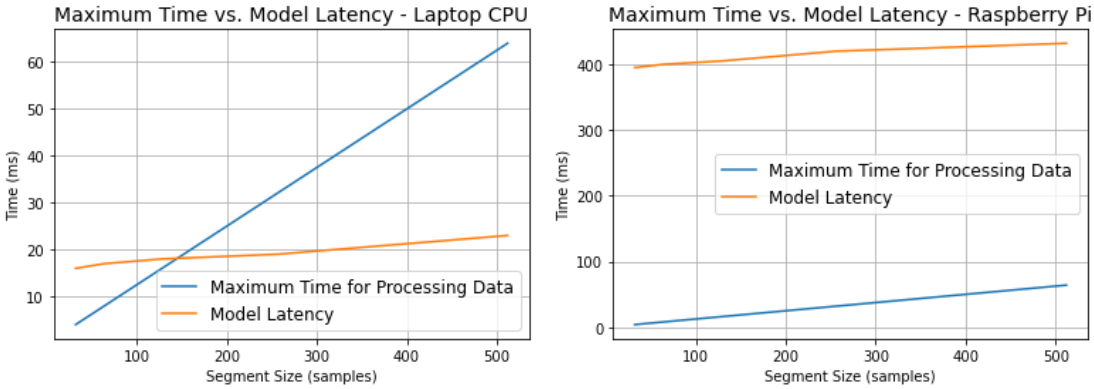


To make it streaming we just had to change three basic building blocks of the model: Conv ID, GlobalNorm and ConvTranspose to their streaming counterparts. This resulted in a streaming model.



*To make the ConvTasNet streaming, we replaced Conv1D, Normalization, and Conv Transpose1D with our streaming implementations.

For our demo, we made use of the python package sounddevice in order to interface with the mic and speaker.



These two plots show the maximum time available for processing data versus the latency of the model, on a computer as well as on a Raspberry Pi. Since our sample rate is 8000 samples/sec, the time available is the segment size/8 ms. We saw that on a computer, the latency of the model increased slowly as the segment size was increased and our model was realtime for segment sizes above 150 samples. However, on a Raspberry pi, the latency was much higher than the maximum time available, even when the segment size was large. This meant that it was not possible to have a real time model on a Raspberry pi. We plan to investigate this further in the future.

Conclusion

1. Yes, we were able to suppress the weaker speaker by about **3** dB, showing that it is possible to create a model to do speech separation.
2. Yes, we were able to suppress the noise by about **6.8** dB, showing that it is possible to create a model for speech enhancement.
3. Yes, based on the results above, we can create a **single model** that can do simultaneous separation of noise and interfering speaker. It is surprising to see that it outperforms the speech separation and enhancement models (that have the same architecture but are trained for individual tasks)
4. Yes, we measured the time it took for the model to run on a laptop and saw that it can be real-time for latencies of about 20 ms. This showed that it can run real-time on a laptop. Please check out our demo!
5. No, we cannot run it on a raspberry pi, as the model is too slow.

Next Steps

We want to make our model run faster so that it can run real time on a raspberry pi. We are puzzled about why the latency on the raspberry pi is so high, and want to investigate it further. Once we do that, we plan to look at even smaller devices like hearing aids and figure out how to run our model there. We also want to improve the signal quality produced by the model so that it can be more useful in a variety of situations.

References

1. *Fourier Transform*, https://www.thefouriertransform.com/#introduction. Accessed 16 January 2023.
2. "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation." *Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation*, https://arxiv.org/pdf/1809.07454.pdf.
3. "Hearing loss." *World Health Organization (WHO)*, https://www.who.int/health-topics/hearing-loss#tab=tab_1. Accessed 16 January 2023.
4. "The Speech Signal - Acoustic Beamforming for Speech Separation." *Google Sites*, https://sites.google.com/site/thuyntranthesisunisa/introduction/the-speech-signal. Accessed 16 January 2023.
5. "LibriMix: An open-source dataset for generalizable speech separation." *Hal-Inria*, 25 September 2021, https://hal.inria.fr/hal-03354695/document. Accessed 5 March 2023.
6. "LIBRISPEECH: AN ASR CORPUS BASED ON PUBLIC DOMAIN AUDIO BOOKS Vassil Panayotov, Guoguo Chen*, Daniel Povey*, Sanjeev Khudanpur." *Dan Povey*, http://www.danielpovey.com/files/2015_icassp_librispeech.pdf. Accessed 5 March 2023.
7. "tky823/DNN-based_source_separation: A PyTorch implementation of DNN-based source separation." *GitHub*, https://github.com/tky823/DNN-based_source_separation. Accessed 5 March 2023.
8. *WHAM! Extending Speech Separation to Noisy Environments*, https://arxiv.org/pdf/1907.01160.pdf. Accessed 5 March 2023.