Speech Enhancement and Denoising Distorted Audio

Rahul Banerjee
ECE
UT Austin

Edwin Hernandez
ECE
UT Austin

Ryuichi Yanagi
ECE
UT Austin

Abstract

Recently, there has been growing in the field of speech enhancement using AI. Companies such as Meta and NVIDIA have published models such as Demucs and Clean-U-Net that accomplish this task well on audio samples that contain additive noise or echo. In this paper, we explore an area within speech enhancement that we feel has been understudied: removing non-echo distortions from audio. To do this, we first created a novel dataset that contains non-echo reverb with imprecise delays in order to simulate faulty microphones or lossy data transmission. In our experimentation, we explored multiple avenues of removing distortion from speech, such as fine-tuning pre-trained models and developing our own model from scratch. We had the most success with fine-tuning the Demucs model on our reverb dataset. We found that our model had a higher perceptual evaluation of speech quality (PESQ) score on our dataset compared to the pre-trained Demucs model. We also noticed our model struggled in the short-time objective intelligibility (STOI) metric on both of our evaluation datasets.

1 Introduction

Isolating speech waveforms despite having background noise or distorting effects is crucial to increasing intelligibility of speech signals. For example, a recording of a speaker can have noise from being in an automobile or being in transit, wind from being outside, or chatter and babble from others in a restaurant. Using traditional methods of automatic speech recognition (ASR), additive noise of this type even at high signal to noise ratios can have huge impacts on accuracy. Hidayat et al. [3] found that traditional MFCC based ASR systems have about a 91% accuracy on clean speech, but at a SNR of only 15db the accuracy decreases to 60%, and at 10db the accuracy is only 40%. As such, it is clear to see that cleaning an audio signal is a highly effective method to increase accuracy.

Historically, speech enhancement in this way has been done with signal processing methods. For example, given a binaural recording, an easy way to clean speech is to have a close microphone to the speaker and one farther out: background noise should hit both microphones at about the same intensity but the speech signal will hit the close microphone at a greater intensity than the farther one. As a result, subtracting the farther waveform from the closer would result in a more intelligible signal.

Modern methods have explored using more machine learning methods trained on artificially noised data. We focus on enhancement in monaural non-signal processing methods using transfer learning, specifically in the domain of distorted speech signals rather than additively noised signals.

For this purpose, we mainly investigate transferability and improvements to an older, waveform-based version of Demucs that was retrained on enhancement as seen in Defossez et al. [1].

Preprint. Under review.

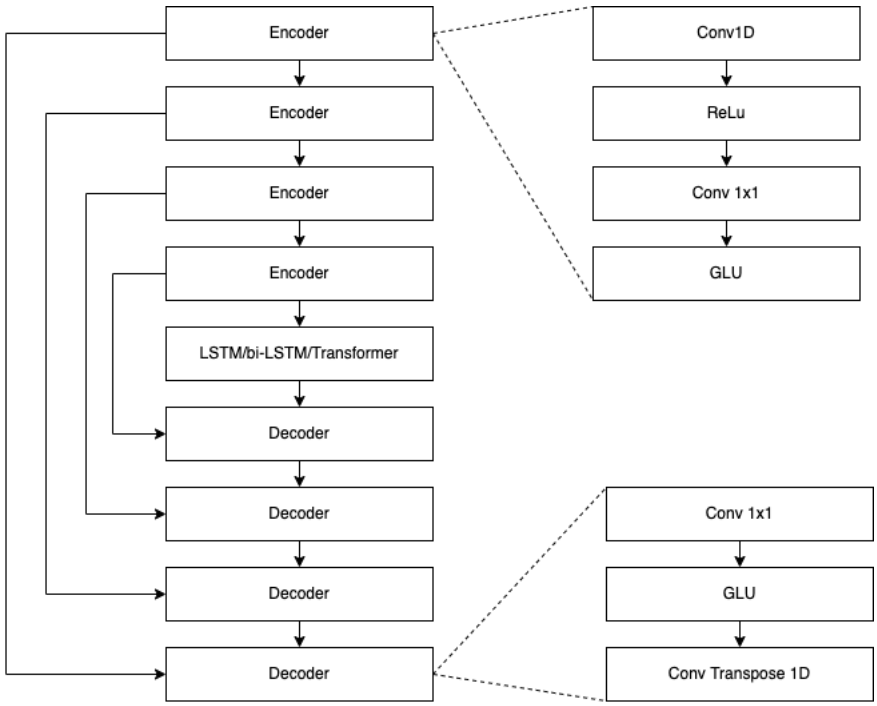

Figure 1: Architectures for the encoder-decoder, U-net style models.

2 Model

2.1 Demucs v2

In the model by Defossez et al. [1], the base model is an encoder-decoder architecture inspired from Luo and Mesgarani [5] that has skip-connections as in U-net from Ronneberger et al. [7]. More specifically, it has layers of encoding and decoding layers. For example, we can use L layers of both encoders and decoders, with outputs of encoder i being passed to encoder $i + 1$ and also decoder i (the skip connection). Symmetrically, we have the outputs of decoder i being passed to decoder $i + 1$.

Each encoder layer consists of a one dimensional convolutional layer followed by a ReLu followed by a convolution with a 1 by 1 stride and step fed into a GLU. The decoder layers are the same layers in reverse. Between the encoder and decoder are LSTM units to model non-linear relations between the vector representations. The original source separation Demucs model used a kernel to stride ratio of 2:1, with a kernel of 8 and a stride of 4 for each encoder — the denoising speech enhancement version has a kernel of 40ms of audio and a kernel of 16ms of audio.

The model also resamples the waveform by a factor by upsampling and then recovers the sample rate by downsampling after. This resampling is done by interpolation and empirically increases the accuracy at the cost of additional computation for training, as the data size is effectively increased by the same factor of resampling.

3 Experiments and Methodology

In most of our experimentation, we used using the standard and premium GPUs offered by Google Colab Pro+, but we also experimented with Paperspace Gradient. The GPUs available to us ranged from NVIDIA K80 to NVIDIA V100.

3.1 Data Augmentation

Instead of just using echo based reverb as seen in Defossez et al. [1] and the augmentations used there, we wanted to experiment with higher degrees of distortion that may come from faulty microphones

or data corrupted during transmission. To do this, instead of just getting standard room impulse responses we recorded impulses in smaller rooms with echos with source and receiver in different locations. The impulses were then chopped up with non-precise delays before being normalized and convolved to create the noisy signals.

In addition to using a non-standard impulse response, we used a different dataset than the models were originally trained on. [1] uses the Valentini dataset, and all versions of Demucs were trained on musical data since it is a musical source separator model. We used a Spanish language dataset to hopefully generalize even further, and augmented it with a small amount of additional noise and the convolutional noise mentioned above.

3.2 Modifications

We took many different approaches and tried different strategies in parallel to look for promising leads before focusing our compute on the most encouraging results.

For one, we tried to take inspiration from architectures that we tested and saw results from and modifying parts of it. Within the encoder-decoder architectures seen in [1, 2, 8, 7, 4] and illustrated in figure 1, we tried taking different approaches to the encoders (and thus decoders). These all used rectified linear units (ReLU), gated linear units, and 1D convolutions.

We tried building our own model from scratch using similar encoder-decoder layers. Using parameters similar to previous work took very long to train, taking multiple hours per epoch. To be able to prototype more models, we tried reducing the parameters in the model by increasing the convolutional stride lengths and convolutional kernel. Although convolutions are used for their low relative computational cost, with audio waveforms at 16 kilohertz, even a short 6 second clip has almost 100k samples. While this did make training much faster, there was a pretty drastic negative impact on performance, and we did not see much improvement even with higher number of epochs. The other method we tried to decrease training times was to reduce the number of hidden unit/layers in the innermost layer of the model, but that was also non-performant.

Another approach we explored was to import weights and use the pretrained version of of the model from Defossez et al. [1], but replace the inner layer. We tried replacing the LSTM in the innermost layer with GRUs or a pure RNN or just a dense connected layer. However, none of these options yielded better results and increased the training cost as well. Varying the number of encoder and decoder layers similarly did not improve performance.

We also tried taking pre-trained models in both speech denoising and music source separation and fine-tuning them on our dataset. However, the pre-trained models were scaled to the computational resources of the groups developing them (Meta/Facebook, NVIDIA, etc). As such, we were a lot more limited in our ability to fine-tune them on both the amount of training done and the amount of data to train on.

4 Results

Table 1 summarizes the results of the fine-tuned model versus the pre-trained model. We ran initial experiments on the different modifications we describe above, but apart from fine-tuning the pre-trained Demucs model, they did not show much promise. Hence, we decided to dedicate our computational resources to fine-tuning Demucs on our custom dataset. For evaluation, we used the DNS benchmark in [6] as well as a testing dataset we created using the data augmentation technique described above. For metrics, we used perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI). PESQ is used to assess the speech quality as perceived by humans. STOI measures the intelligibility of degraded signals, where a higher score means that the signal is easier to understand. We used the wide-band version of PESQ in order to compare to the results achieved on [1]. We fine-tuned the model for 540 epochs of two batches over our custom dataset containing more than 5,000 samples.

Results show an increase in the STOI score on the reverb dataset (the dataset that we made) when compared to the Demucs model but not in PESQ. This means that the fine-tuned model was able to remove the echo better, but struggled to maintain speech quality. This was evident during the training process, as the fined-tuned model was getting better at reconstructing the speech without any echo,

Table 1: Results

Results				
Model	DNS 2020		Reverb Dataset	
	PESQ	STOI	PESQ	STOI
Demucs	2.42	92.57	1.78	18.51
Fine-tuned Demucs	1.13	46.79	1.233	27.65

but still did not reached the level of the original model. However, the performance on noise removal significantly decreased as a result, showing that the fine-tuned model was specializing too much on speech with echo.

5 Discussion

5.1 Computation Costs

Running models on waveforms is expensive — even at lower sampling rates for speech (at 16kHz) compared to music (at 44kHz), training the model takes a long time. The most performant models in the space that have been developed by researchers at companies like Meta and NVIDIA have been trained using 8 V100 GPUs for 1200 epochs of 800 batches [8], which is outside of the computational capacity of our team. Further, our dataset was smaller than other datasets typically used in this field, and we did this to help our model train faster. If we expanded our dataset to be a standard size, each epoch would take longer to complete. Moreover, usual datasets in the field can go up to one terabyte in size, storage we didn't have available.

5.2 Loss Functions

Many of the models we investigated before we began our own work used L1 and L2 losses to model the waveform loss. While this makes sense for source separation, for speech itself it lacks some nuances that the human ear and brain have. For example, those losses do not reflect how certain frequencies are more in tune with what people hear. Further, L1 and L2 losses are not time invariant at all — if we had a model that output the correct waveform that starts one millisecond later in the audio clip, to a human it would sound identical but the loss from the loss functions would be huge.

6 Conclusion

In this work, we explored multiple avenues of trying to remove non-echo distortion from audio samples. After testing these different paths, we narrowed our efforts to fine-tuning a pre-trained Demucs model on a novel reverb dataset. We found that our fine-tuned model outperforms Demucs in the PESQ metric on our dataset, but falls short in other areas. One of the big things we found while experimenting was that in exchange for better performance on reducing distortion in audio, our models would perform worse at removing additive noise. One way that this issue could be addressed would be to use a different model architecture, such as a transformer. The Demucs model we experimented used LSTMs, but there is a new version available that uses a hybrid transformer architecture and may have performed better in both tasks. This should be something explored in the future. Additionally, if we had more computational resources, we would have liked to see how our fine-tuned model would perform after more training epochs.

One of the main contributions from this work is the novel reverb dataset we created. Our dataset differs from others Defossez et al. [1] because we simulate distortion by recording impulses with echo with varying distances between source and receiver and chopped them up with non-precise delays. We see this dataset potentially being used in future research either as a way to fine-tune a speech enhancement model or to evaluate a speech enhancement system on realistic scenarios.

References

[1] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain, 2020. URL `https://arxiv.org/abs/2006.12847`.

[2] Alexandre Défossez. Hybrid spectrogram and waveform source separation, 2021. URL https://arxiv.org/abs/2111.03600.

[3] Risanuri Hidayat, Agus Bejo, Sujoko Sumaryono, and Anggun Winursito. Denoising speech for mfcc feature extraction using wavelet transformation in speech recognition system. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 280–284, 2018. doi: 10.1109/ICITEED.2018.8534807.

[4] Zhifeng Kong, Wei Ping, Ambrish Dantrey, and Bryan Catanzaro. Speech denoising in the waveform domain with self-attention, 2022. URL https://arxiv.org/abs/2202.07790.

[5] Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, aug 2019. doi: 10.1109/taslp.2019.2915167. URL https://doi.org/10.1109/taslp.2019.2915167.

[6] Chandan K. A. Reddy, Ebrahim Beyrami, Harishchandra Dubey, Vishak Gopal, Roger Cheng, Ross Cutler, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, Puneet Rana, Sriram Srinivasan, and Johannes Gehrke. The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework. *CoRR*, abs/2001.08662, 2020. URL https://arxiv.org/abs/2001.08662.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

[8] Simon Rouard, Francisco Massa, and Alexandre Défossez. URL https://arxiv.org/abs/2211.08553.