Speech Denoiser (GitHub)

Team (Name, netID):

- George Evans Daenuwy (netID: grgevans)
- Nels Martin (netID: nelsm2)
- Akash Savitala (netID: akash18)

Background:

For this project, our focus was on improving speech recognition on noisy audio by cleaning the sound before it is sent into an automatic speech recognition (ASR) model. First off, the dataset we used comes from the Denoising Audio Collection on Kaggle and contains **paired clean and noisy versions** of the same speech recordings. Essentially, this setup is important because it allows a model to directly learn how to map noisy audio back to clean audio. And as part of our data preparation process, all audio was converted to mono, resampled to 16 kHz, and normalized to match the expected format. The main model used in this project is a 1D Convolutional U-Net, which operates directly on raw audio waveforms. The U-Net is a learned model that actually trains on examples of noisy and clean speech to figure out how to remove noise on its own.

Our background motivation comes from the trade-off between model size and accuracy in Whisper. While large variants of Whisper achieve strong performance, smaller versions like Whisper-tiny, base, or small suffer a noticeable drop in accuracy. Transcription quality without relying on a large model, we introduce a neural denoiser that cleans the input audio first. Using classical DSP denoisers to remove parts of the speech and using a neural denoiser to create cleaner audio while keeping the speech natural.

Methodological Approach:

Finding Standardized Dataset

We standardized the dataset for all experiments. We used the Denoising Audio Collection dataset, which provides paired clean and noisy speech. As part of our data preparation process, all audio was converted to mono, resampled to 16 kHz, normalized, and split into training, validation, and test sets. Longer clips were segmented into fixed-length chunks.

Designed Three Comparables Approach

To fairly evaluate different approaches, we created three pipelines that all feed into the same ASR model, Whisper-small. **Pipeline A** sends noisy audio directly into Whisper as a baseline. **Pipeline B** applies a classical DSP denoiser using Wiener Filter before

ASR. **Pipeline C** applies our learned 1D Conv U-Net denoiser before ASR. Structuring the project this way allows us to directly compare classical and neural methods under identical conditions.

Comparing DSP Denoiser and Noisy Audio
Develops a speech denoising and recognition pipeline that integrates classical signal processing with modern machine learning. A Wiener-like spectral denoiser is first implemented, which estimates the noise power from a noise-only segment of the signal and applies a frequency-dependent gain controlled by tunable parameters alpha and gain floor to suppress noise in the STFT domain before reconstructing a cleaner waveform. The denoised audio is then passed into the Whisper-small automatic speech recognition (ASR) model to generate transcripts, which are compared against those obtained from both noisy and clean audio. Finally, the denoiser's parameters are optimized either by minimizing waveform-level error relative to clean references.

1D Convolution U-Net
This neural network architecture performs a series of 1D convolutions on an audio waveform vector to decrease the length of the vector(s) and increase the number of channels. The latent vector is then upsampled to the original size of the input vector, and the loss function is the MSE of the output vector and the clean target vector. Another key feature of the U-Net is skip connections. During the up sampling, activations from previous layers are added to the activations of later layers, increasing the resolution of the final output.

Concepts from Class
Our project includes the U-Net architecture (and therefore an encoder and decoder), data normalization, train/test splits and validation, the MSE loss function, and data preprocessing, all of which are concepts from class.

Evaluation
Whisper-small was used only as an inference model and was never trained or modified. The only difference across pipelines was the version of audio fed into Whisper. Moreover, we compared transcripts using Word Error Rate (WER) as the main quantitative metric and also examined waveform plots, spectrograms, and listening examples for qualitative assessment.

Results:

DSP Denoiser
To begin our evaluation, we tested the classical DSP approach to understand how much improvement a non-learned denoiser can provide to speech recognition. Using the

same noisy test set, we compared Whisper-small's transcription performance under two conditions:

Pipeline	**Input to Whisper**	**Description**
Baseline (Noisy)	Noisy audio only	Whisper receives unprocessed noisy speech.
DSP Denoiser	Wiener-filtered audio	Noise suppressed using spectral filtering based on estimated noise power.

Word Error Rate Comparison
Example input : *Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob.*

Pipeline	**Word Error**	**Result**
Baseline (Noisy)	1 word	*Six spins of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob.*
DSP Denoiser	3 words	*Six bins of fresh snow peas, 5-6 slabs of blue cheese, and maybe a snack for her brother Bob.*

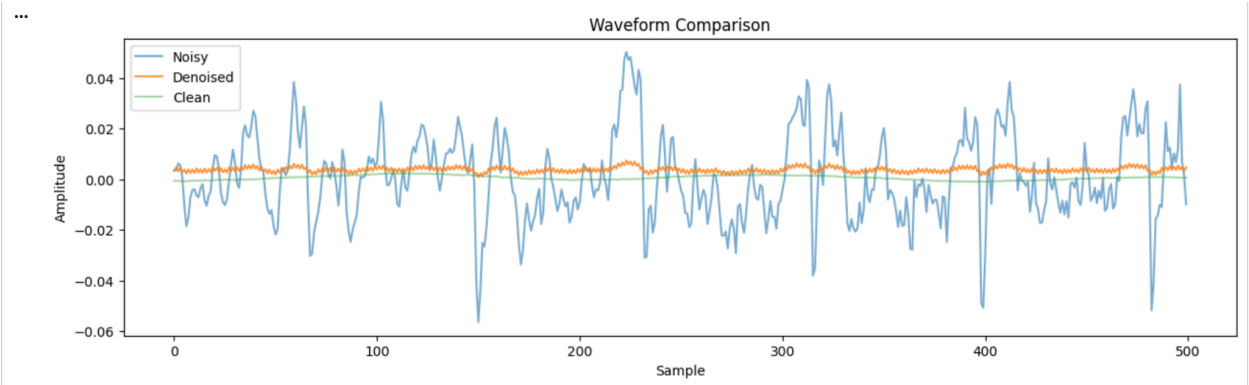
The DSP Denoiser has a higher error rate because of :

1. The Wiener filter successfully suppresses broadband stationary noise, However it over suppressed high frequency energy.
2. Consonants such as /s/, /t/, /f/, /k/ have higher frequency energy which make the word errors as shown in the table.

U-Net
To test the effectiveness of the U-Net denoiser, we evaluated the Whisper model on a noisy audio sample and on the de-noised audio sample. Whisper produced nearly identical results for the two audio samples:

Pipeline	**Word Error**	**Result**
Baseline (Noisy)	0 words	*Ask her to bring these things*
U-Net Denoiser	0 words	*Ask her to bring these things.*

With the only difference being the addition of a period in the second sentence. We also evaluated the performance of the U-Net model by analyzing the waveforms of the clean, noisy, and denoised audio samples:



The denoised audio waveform is much closer to the clean waveform to the noisy waveform, but still contains some of the fluctuations from the noise in the original audio file.

Conclusion:

Both the U-Net and the DSP architectures decreased the noise in the audio files they processed. However, when evaluated with the Whisper audio-to-text model, the denoised files did not necessarily produce better output then the noisy files. In the case of the DSP pipeline, this is likely because the Weiner filter removes high-frequency sounds that Whisper needs to be able to identify words. In the case of the U-Net pipeline, the audio was able to be understood in its noisy state, so more analysis would need to be done to determine exactly how effective the U-Net architecture is at increasing speech intelligibility.

References:

OpenAI Whisper GitHub: https://github.com/openai/whisper

Scipy Signal Processing – Wiener Filter Documentation :
https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.wiener.html

PyTorch STFT Documentation :
https://pytorch.org/docs/stable/generated/torch.stft.html