Noise Suppression Model for Real-Life Human Audio-Project Report

Team Name-AI Innovators

Authors: Aryan Gupta (230150003), Vibha Gupta (230150029), Tejas Deshmukh (230150027), Vaishnavi Agarwal (230150028)

1. Introduction

Background:
Real-world audio recordings are frequently contaminated by various types of background noise such as traffic, machinery, and crowd chatter. This unwanted noise degrades speech intelligibility and audio quality, affecting applications ranging from telecommunication to assistive listening devices.

Objective:
The goal of this project is to design an effective noise suppression model for real-life human audio. We compare two methods: a neural network-based autoencoder and a classical spectral subtraction approach. While the autoencoder offered a data-driven solution, it ultimately lacked robustness, prompting us to adopt spectral subtraction as the final method.

2. Dataset Used

Dataset Source:
We utilized the “Noisy speech database for training speech enhancement algorithms and TTS models” from Edinburgh DataShare (https://datashare.ed.ac.uk/handle/10283/1942). This dataset provides paired clean and noisy WAV files captured under diverse real-world noise conditions.

Key Characteristics:
• Approximately 30 hours of speech data covering various speakers and environments.
• Noisy samples include common urban noises (traffic, crowd, air conditioning) at multiple SNR levels.
• Clean speech recordings serve as ground truth for evaluation.

Usage:
For the neural network approach, 3-second clips were extracted and converted into 128×128 mel-spectrogram patches. For spectral subtraction, full-length audio and an external test pair were processed to compute objective SNR improvements.

3. Technology Stack

Programming Language & Environment:
• Python 3.12.3 on Windows/Linux
• Jupyter Notebook for prototyping

Core Libraries:
• NumPy & SciPy: Numerical computations and signal processing
• Librosa: Audio I/O, STFT/ISTFT, mel-spectrogram extraction
• SoundFile: Reading and writing WAV files
• Matplotlib: Visualization of waveforms and spectrograms

Machine Learning (Approach 1):
• PyTorch: Defining and training the convolutional autoencoder
• scikit-learn: Train/test split and basic preprocessing

Additional Tools:
• Git for version control
• Microsoft Word for report generation

4. First Approach: Neural Autoencoder (Brief Overview)

We followed an online tutorial to test out this approach. However, the lack of results in the approach forced us to move to an alternate approach. We have used neural networks in this initial approach which is showcased in alternate_failed_approach.ipynb.

5. Second Approach: Spectral Subtraction (Detailed)

Methodology:
Spectral subtraction is a classical noise reduction technique operating in the frequency domain. By estimating the noise spectrum and subtracting it from the noisy signal's magnitude, we can attenuate stationary noise effectively.

Algorithm Steps:
1. Estimate noise magnitude spectrum (average over first 0.5 s).
2. Compute STFT of the full noisy signal (window=2048, hop=512).
3. Subtract noise spectrum from each frame's magnitude, flooring at zero.
4. Combine modified magnitude with original phase.
5. Perform inverse STFT (ISTFT) and fix signal length.
6. Insert 5 ms of zeros at start to correct edge artifacts.
7. Save denoised audio as 16-bit PCM WAV.

Key Advantages:
• No training required – deterministic and fast.
• Handles stationary noise well, such as hums and constant background sounds.

• Controls distortion by flooring and edge correction tricks.

Practical Considerations:
• Choice of window/hop impacts smoothness and residual noise.
• Noise profile updates could address non-stationary noise.
• Combined with adaptive thresholds, further improvements are possible.

6. Results & Evaluation

External Test Case (p232_003.wav):
We evaluated SNR improvements on a held-out noisy/clean pair:
• SNR before denoising: 2.1 dB
• SNR after spectral subtraction: 8.7 dB
• Net improvement: +6.6 dB

Visual Analysis:
• Waveform plots show reduced noise floor and retained speech peaks.

Perceptual Feedback:
• Preliminary listening reported clearer speech.

7. Contributions of Team Members

• Aryan Gupta:
 - Designed the STFT/ISTFT parameters and noise-profile estimation routine.

 -Completed the implementation of first approach.

• Vibha Gupta:
 - Preprocessed audio files, handled format conversions.

 - Implemented edge correction and final audio saving logic.

• Tejas Deshmukh:
 - Developed visualization scripts for waveform and spectrogram comparisons.

 - Documented findings and contributed to report synthesis.

• Vaishnavi Agarwal:
 - Assisted with hyperparameter tuning.

 - Conducted objective SNR computations and compiled evaluation results.