Aman Khan Sprint Task 2 – Project Echo Engine Team (task 3)

Audio files used from /GitHub/Project-Echo/src/Prototypes/data/WebScrapeAndStoreSounds/Downloaded\_sounds"

**red fox screeching** for Bandpass Filtering

**cute artic white fox laughing vine** for Noise Gate

**Chris Fox KRLD** for Spectral Subtraction

**Audio Filtering Documentation**

In this project, we focused on cleaning audio recordings by applying a bandpass filter. This method helps remove noise and irrelevant sounds, improving the quality of the audio for further analysis, such as animal call classification. Below is a detailed explanation of the process.

**Objective**

The goal was to reduce unwanted noise from the audio files while preserving the key features of the animal calls. This step is crucial to ensure the audio data is clear and accurate before it is used in machine learning models.

**Method**

We used a bandpass filter, which isolates frequencies within a specified range. Based on our analysis of the audio files and their spectrograms, we determined that most animal calls occur between 300 Hz and 3000 Hz. Frequencies outside this range, such as very low (background hum) or very high frequencies (static noise), were removed.

The bandpass filter was implemented using Python’s scipy.signal library. It was applied to the audio files to suppress noise while retaining the sounds of interest.

**Results**

The filtering process produced significant improvements:

A diagram of a sound waveform

Description automatically generated

**Waveform Analysis:**

The original waveforms showed a lot of low-amplitude signals, which were likely background noise.

After filtering, these low-amplitude noises were removed, leaving only the prominent signals, which are likely to be the animal calls.

A screenshot of a sound recording

Description automatically generated

**Spectrogram Analysis**

The original spectrogram showed energy spread across a wide range of frequencies, including areas irrelevant to animal calls.

The filtered spectrogram concentrated energy within the 300–3000 Hz range, reducing noise while retaining the relevant audio features.

These changes ensure that the audio data is now much cleaner and better suited for downstream tasks, such as classification.

Key Takeaways

Bandpass filtering is a simple yet effective method for cleaning audio files.

By targeting a specific frequency range, we successfully removed irrelevant noise without compromising the important features of the animal calls.

The filtered audio files are smaller in size and clearer, making them more efficient and reliable for further analysis.

**Noise Gate**

**Technique Overview**: The Noise Gate method filters out signals below a specific amplitude threshold, assuming they are noise. This threshold was carefully set to ensure that low-amplitude noise was removed while retaining relevant parts of the audio signal.

**Implementation**:

The Noise Gate function was implemented in Python using NumPy to zero out values below a threshold.

A threshold of 0.02 was used, determined experimentally to strike a balance between noise removal and signal preservation.

The filtered audio was visualised through waveform and spectrogram comparisons to assess the effectiveness of the technique.

**Data**: The selected audio file was loaded from the project dataset. The file contained both prominent target sounds and faint background noise.

**Results**

**Waveform Analysis**:

The original waveform showed low-amplitude fluctuations between the target sounds, indicating background noise.

After applying the Noise Gate, these low-amplitude regions were significantly reduced, leaving a cleaner waveform with only the key features intact.

A group of blue sound waves

Description automatically generated

**Spectrogram Analysis:**

The original spectrogram displayed energy spread across a wide range of amplitudes.

Post-filtering, the spectrogram revealed a more focused representation of the target frequencies, with reduced noise in low-energy regions.

A close-up of a sound recording

Description automatically generated

**Audio Quality:**

The filtered audio was clearer, with most background noise removed.

The main features of the audio (e.g., animal calls) were preserved, though faint signals near the threshold may have been attenuated.

**Spectral Subtraction**

**Methodology**

**Technique Overview**: Spectral Subtraction removes noise by identifying a noise-only region in the audio and subtracting its spectrum from the entire audio signal. This approach assumes that the noise is steady and consistent across the recording.

**Implementation**: A noise-only region (the first 2 seconds of the audio) was identified to calculate the noise profile. Using the Spectral Subtraction function, the noise spectrum was subtracted from the audio spectrum. The cleaned signal was reconstructed using the modified spectrum.

**Data**: The selected audio file contained both prominent target sounds and background noise. The first 2 seconds of the file were used as a noise reference.

**Parameters**: Noise region: 0–2 seconds. The subtraction ensured that negative values in the spectrum were clamped to zero to avoid artefacts.

**Results**

**Waveform Analysis**:

The original waveform showed low-amplitude noise throughout the recording, particularly in regions between the main target signals.

After applying Spectral Subtraction, the low-amplitude noise was significantly reduced, resulting in a cleaner waveform with better-defined target signals.

A close-up of a sound wave

Description automatically generated

**Spectrogram Analysis:**

The original spectrogram displayed widespread energy across low-frequency and high-frequency bands, indicating background noise.

The filtered spectrogram revealed a reduction in noise, particularly in frequencies outside the target range. The energy is now concentrated in the regions corresponding to the target signals.

A screenshot of a computer screen

Description automatically generated with medium confidence

**Discussion**

**Strengths:**

Spectral Subtraction was effective in reducing steady, consistent noise from the recording.

The technique preserved the primary features of the audio, such as animal calls, while removing irrelevant sounds. This method is ideal for audio recordings with clearly distinguishable noise and signal regions.

**Limitations:**

Over-subtraction can sometimes distort target signals, particularly when the noise spectrum overlaps with the target frequency range. This method may be less effective for dynamic or inconsistent noise.

**Conclusion**

After applying and analysing the three audio filtering techniques—Bandpass Filtering, Noise Gate, and Spectral Subtraction—it is evident that each method has its strengths and limitations depending on the type of audio and noise present. Bandpass Filtering was effective in isolating frequencies within a specific range (e.g., 300–3000 Hz), making it highly suitable for signals with clear frequency boundaries, but it struggled with noise overlapping within the target range. Noise Gate efficiently removed low-amplitude background noise, resulting in a cleaner waveform, but it risked cutting off faint signals near the threshold. Spectral Subtraction demonstrated the best balance by targeting steady background noise and preserving most of the key audio features, though it introduced minor distortions when noise frequencies overlapped with the target. Overall, Spectral Subtraction proved to be the most versatile and effective method for this project, as it delivered significant noise reduction while retaining clarity in the primary signals, making it ideal for diverse audio scenarios.