Exercise 4

Introduction to the Software and Configuration

In this project, we utilized the p5.js library along with its sound extension, p5.sound, to process audio signals for automatic speech recognition (ASR). The software was configured to preprocess audio files, reducing noise interference before feeding them into ASR systems. The primary aim was to enhance the accuracy of speech recognition in noisy environments.

Solution to Noisy Environment Issue

Noise Reduction Techniques: Our approach to mitigate the impact of background noise involved a combination of digital signal processing techniques. These included a low-pass filter to remove high-frequency noise and a dynamic range compressor to reduce volume disparities.

Implementation: The low-pass filter was set with a cutoff frequency strategically determined based on the noise profile of test samples. The compressor was configured to normalize the volume, ensuring consistent input levels for the ASR system. This preprocessing significantly improved the clarity of speech in the recordings.

Analysis of Test ResultsTest Setup: The evaluation involved processing a set of audio files provided by the client, along with additional recordings made in various noisy environments. Each file was processed through our noise reduction setup before being analyzed by the ASR system.

Observations: The results indicated a marked improvement in ASR accuracy post-processing. Audio files with low to moderate background noise saw the most significant enhancement, with a noticeable reduction in misinterpreted words. However, in extremely noisy environments, while there was improvement, some speech elements were still lost.

Evaluation of ASR Libraries

ASR Libraries Considered: We evaluated several ASR libraries, including Google Speech-to-Text and IBM Watson Speech to Text. These libraries were chosen for their advanced speech recognition capabilities and widespread use in the industry.

Comparison: Google's ASR exhibited superior performance in terms of accuracy and latency, especially in less noisy conditions. IBM Watson provided more consistent results across varying noise levels but at a slightly lower accuracy rate.

Pros and Cons: Google's ASR demonstrated high efficiency in optimal conditions but was more susceptible to errors as background noise increased. IBM Watson's ASR, while slightly less accurate overall, showed greater resilience to noise interference.

Conclusion

Summary of Findings: The implementation of noise reduction techniques significantly improved the performance of ASR systems in noisy environments. While not completely eliminating errors, the preprocessing steps made speech recognition more reliable and accurate under challenging conditions.

Future Developments: Future work could explore adaptive noise cancellation techniques, where the noise profile is dynamically analyzed and filtered. Additionally, experimenting with more sophisticated ASR libraries or custom-built neural network models could yield further improvements.