ELIMINATION OF BACKGROUND NOISE FROM SIGNAL

Munazza Begam, Swathi Prabhu, Kavya G , Bhoomika H , Asia Hazareena*

Department of Electronics and Communication, P. A. College of Engineering, Karnataka, Mangaluru, India

E-mail: hazareena_te@pace.edu.in

Abstract

By removing background noise through adaptive filtering, band-pass filtering, dynamic range compression, and normalization, this project seeks to enhance audio quality. Python libraries such as sound device, librosa, and scipy were used to process audio recordings that were recorded at 16,000 Hz. Waveform and spectrogram analysis verified that the system effectively reduced noise while maintaining the main signal's clarity. The project establishes the groundwork for effective noise reduction in real time, with the possibility of machine learning improvements in the future.

1 Introduction

In the field of audio signal processing, getting high-quality recordings is necessary for several uses, such as multimedia production, forensic analysis, and telecommunications. Unwanted noise frequently deteriorates the clarity and understandability of audio recordings, including background static, hiss, and interference from the environment. Effective noise

ISBN:97881-19905-39-3

reduction strategies are essential because these noise issues have the potential to hide key details. Complete audio enhancement demands both band-pass filtering and dynamic range compression normalization for adding noise reduction methods to the procedure.

The noise reduce library implements adaptive filtering in its complex noise reduction systems that blocks noise while protecting waveform integrity. Background noise reduction improves when using band-pass filtering procedures since these methods excel better than Butterworth filter implementations. The files in an audio project undergo level normalization to achieve uniformity while range compression operates on sound intensities to keep consistent output volumes.

These methods enable the project execution to produce audio content with high quality fidelity and equalized volume levels together with clarity in sound. The evaluation of this approach relies on aural and visual inspection methods that use waveform and spectrogram analysis. The project shows first how the topic matters to industry yet encompasses broad industrial applications before presenting methods to boost audio signal quality through modern techniques.

1.1 Aim and Objective

Our project aims to enhance audio recording quality using advanced processing techniques. Sound recording will proceed through the sound device library at 16,000 Hz. The noise reduction library includes robust noise reduction abilities that we will adjust parameters to reach optimal noise reduction objectives. The Butterworth band-pass filter enables frequency propagation between 300 to 3000 Hz while blocking all other frequencies. The application of dynamic range compression will normalize audio levels while dynamic range normalization keeps loudness consistent. Visual inspection together with aural tests will be used to evaluate the implemented methods. Our research team plans to refine adaptive noise identification through advanced filtering techniques for enhancing audio excellence in our device.

1.2 Problem and Summary

• Unwanted noises within audio recordings consisting of background static and hiss together with environmental interferences lead to severe reduction of audio clarity.

Unwanted background noises diminish essential information while decreasing the effectiveness of recorded material across multimedia production together with forensic analysis and telecommunication sector because the contents become difficult to understand.

• Poor audio quality in conversations can cause misconceptions and misinterpretations. Unwanted noise in multimedia production can lower the final product's audio quality and professionalism. Noise in audio evidence can damage its integrity and make it harder to distinguish important information in forensic analysis.

2 Literature Survey

Tamura, Shinichi. There is noise everywhere. In the majority of audio-related applications, including hands-free communications, voice over IP (VoIP), human-machine interfaces, hearing aids, teleconferencing/telepresence/telecollaboration systems, and many more, the signal of interest, which is typically speech, that is picked up by a microphone is typically contaminated by noise. Band-pass filtering when combined with dynamic range compression and normalization and noise reduction forms the basis of this project.

The project aims to enhance audio quality as a complete system. Functionality of adaptive filtering exists because complex noise reduction algorithms support these operations. The selection process performed by the noise-reducing library allows beneficial audio frequencies to pass through yet continues to protect the unaltered sounds. The application of band-pass filtering to vital speech frequencies receives particular benefit because it enhances important frequencies. Injection of Butterworth filters serves to minimize the disruptive effects of background sounds. Sudheer Kumar, E., et al. The loudness control aspect of dynamic range compression maintains the sound levels throughout a recording

ISBN:97881-19905-39-3

through its volume adjustment capabilities.

Through normalization all processed audio signals receive an amplitude normalization that establishes a consistent volume level. Implementation of these strategies leads the project to generate audio outputs with high fidelity and balanced and clear sound quality and recordings.This analysis incorporates aural comparisons with an additional use of visual inspection methods through waveform and spectrogram assessment techniques to analyze the results.

The introduction emphasizes how this project affects multiple sectors as well as its potential influence to follow. The audio signal quality receives remarkable enhancements through the developed methods. The outcomes of the experiments show that the suggested framework can significantly enhance the mathematical approach. It's also noteworthy to observe that the suggested method consistently lowers non-stationary noise.

Noise reduction in audio signals is a crucial area of research, particularly for applications like hands-free communications, VoIP, human-machine interfaces, hearing aids, and teleconferencing systems. The primary challenge in these applications is to improve the signal-to-noise ratio (SNR) while preserving the quality of the original signal, typically speech. Over the years, the evolution of noise reduction techniques has seen a shift from basic filtering methods to more advanced machine learning approaches. Haykin, S. Early noise reduction methods focused on linear filters such as low-pass, high-pass, and band-pass filters, which attenuate unwanted frequencies typically associated with noise. However, these techniques have limitations, especially in complex acoustic environments where noise and speech overlap in the frequency domain. Wiener filters, for example, aim to minimize the mean square error between the estimated and original signal, showing effectiveness in stationary noise environments, but they struggle with non-stationary noise [3]. Boll, S. F. Another widely used technique is spectral subtraction, where the noise spectrum is estimated and subtracted from the noisy signal, although it can introduce artifacts, such as "musical noise," if not carefully implemented [4].

ISBN:97881-19905-39-3

Widrow, B., & Stearns, S. D. Adaptive filtering, using algorithms like Least Mean Squares (LMS) and Recursive Least Squares (RLS), adjusts filter parameters in real-time to match changing noise environments, making it suitable for non-stationary noise conditions often encountered in mobile or teleconferencing applications [5]. Kalman, R. E. Echo cancellation, particularly important in telecommunication systems, involves identifying and removing echoes that can interfere with speech clarity, with techniques like the Kalman filter being employed successfully [6].

Xu, Y., et al. In recent years, machine learning has become increasingly applied to noise reduction. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown significant promise in distinguishing between noise and speech in complex environments. These models are trained on large datasets to learn the characteristics of both noise and speech, leading to more accurate noise reduction [7]. Vincent, P., et al. Denoising autoencoders, specialized neural networks designed to reconstruct clean signals from noisy inputs, have been applied effectively in scenarios where traditional methods fall short, especially in highly non-stationary noise environments [8].

Dillon, H. Different apps need special ways to cut noise. Hearing aids need quick noise cutting with little wait and must keep speech clear. They use tools like spectral subtraction and adaptive filtering for this [9]. Meeting tools use tech with many microphones to make spoken words clear and cut background noise [8]. New tech keeps growing, even with the ongoing problem of random noise. Future study looks to mix machine learning with old DSP methods to build fast ways that boost noise cutting in different places. Recent tech jumps make better sound possible for many uses.

3 Requirement Specification

3.1 Software Specifcation

- Python

3.2 Python Deftnition

Python is a free language offering many tools for science computing, machine learning, and signal processing. Signal processing can be done with libraries like SciPy and NumPy. Use filtering methods and adaptive techniques like spectral subtraction and Wiener filtering. The implementation of machine learning noise reduction methods operates through frameworks TensorFlow and PyTorch and keras libraries.

The signal processing uses Wavelet transform through the PyWavelets software.

Toolboxes like SciPy, NumPy, PyWavelets, TensorFlow, PyTorch, and librosa in audio work.

4 Problem Statement

Background sounds hurt signals in phones, sound work, health tests, and checks on nature. Cutting sounds helps keep signals clear and real. Sound clutter shows up in three ways: change in pitch, change in loudness, and time trouble. Finding it is hard. To cut clutter, we use smart tools like filters, changeable filters, taking out parts, wave changes, and smart learning. Many ways are out there to cut sounds, but we need to keep signals safe while cutting the clutter. This stops us from messing up signals by mistake.

4.1 Objectives

Identification and Characterization of Noise: This work seeks to develop methods that could be used in the proper identification and characterization of the background noise of different types of signals. It involves looking into the frequency, amplitude, and time characteristics of the noise to find out how it interacts with the signal of interest. To this end, there will be a full understanding of the characteristics of the noise, hence opening ways for choosing suitable noise reduction methods.

Evaluation of Current Noise Reduction Methods:

One of the important objectives is the performance evaluation of old and new noise reduction techniques. This would include conventional techniques, like filtering, spectral subtraction, wavelet transforms, and some more modern techniques, such as adaptive filtering and machine learning-based methods.They want to evaluate these methods for different situations and determine their effectiveness with diverse signaling and noise types.

Designing and implementing noise reduction algorithms forms the main objective of this project while considering the specified signal types together with noise characteristics. The creation of an effective algorithm demands necessary noise reduction with the preservation of signal quality for the targeted content. The designed algorithms need optimization for real-time processing effectiveness and efficiency throughout the development phase.

The project will review the relationship between noise reduction success and resulting signal modification.

Making the reduction of noise work efficiently requires maintaining good signal integrity represents a primary design objective. The study analyzes different noise reduction strategies to preserve the fidelity of the wanted signal while developing approaches for suppressing distortions or artifacts that result from these methods. Achieving this exact balance stands as the essential prerequisite to guarantee the methods used for noise reduction do not degrade signal quality.

Hybrid Approaches: The researchers must assess the various methods across multiple situations to understand their capabilities when processing different noise types.

The principal goal involves designing and executing noise reduction algorithms for specified signals which need noise characteristic evaluation. The design process of proper algorithms requires both noise reduction procedures along with quality maintenance for the desired content. Real-time processing optimization needs to be applied to the designed algorithms during the development period for effectiveness and efficiency.

The evaluation examines how successful noise reduction affects the modification of signals throughout the investigation.

ISBN:97881-19905-39-3

Good signal integrity maintenance functions as a main design objective required for efficient execution of noise reduction measures. The research examines multiple noise reduction techniques to protect original signal fidelity by building techniques that minimize the artifacts which develop from these approaches. Every noise reduction system needs precise signal to noise management which functions as the base requirement to prevent methodology-related signal quality deterioration.

Recommendations and Best Practices:

The research study creates guidelines which suggest appropriate noise reduction methods for different applications alongside best practices for their implementation. Guidelines along with best practices will be established to achieve effective application of these methods when dealing with real-world scenarios for maintaining good performance across disparate contexts.

The Signal Processing Field Receives Both Direct Positive and Negative Impact:

The project will assist signal processing development through research publication and knowledge distribution to the broader processing community. The research identifies various implementations and future advancements for noise reduction that serve as foundations for additional research projects in this field.

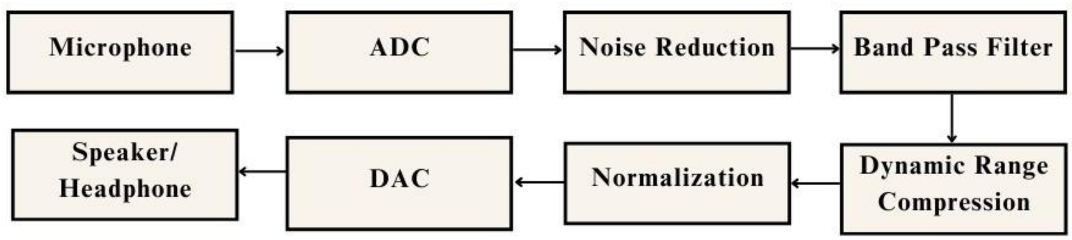
The project includes three key areas of development: improved signal processing abilities, better signal quality outcomes and practical solutions to diminish background noise in different applications.

4.2 Expected Outcome

The implementation of a reliable audio processing system that efficiently records, denoises, filters, and visualizes audio signals is the project's anticipated result. The audio recordings produced by the system will have far less background noise, more clarity, and higher quality. Through in-depth waveforms and spectrograms, users will be able to see the audio data, facilitating a deeper study and comprehension of the audio material. This system will

ISBN:97881-19905-39-3

produce a clean and professional audio output, making it very helpful for applications like voice recordings, podcasts, and other audio-related work in settings with a lot of background noise.



Block Diagram

Figure 1: 1 Block diagram of Elimination of background noise from signal

Figure 5.1 The construction of the audio recording and processing system is shown in the block diagram. The microphone captures sound before converting it into analog electrical signals until it reaches the processing system. A digital converter unit known as an analog-to-digital converter transforms the input electrical signal into digital form for processing by a computer system. The digital signal requires special algorithm processing to decrease noisy interference. The band-pass filter removes the frequencies within the 300-3000 Hz range that represents human vocal range. For standardizing volume levels the signal compression method affects the signal range. The signal undergoes compression before normalization

ISBN:97881-19905-39-3

to achieve a specific amplitude range which enables it to fit. The final digital output gets converted into an analog signal with the help of a Digital-to-Analog Converter followed by audio transmission through headphones or speakers.

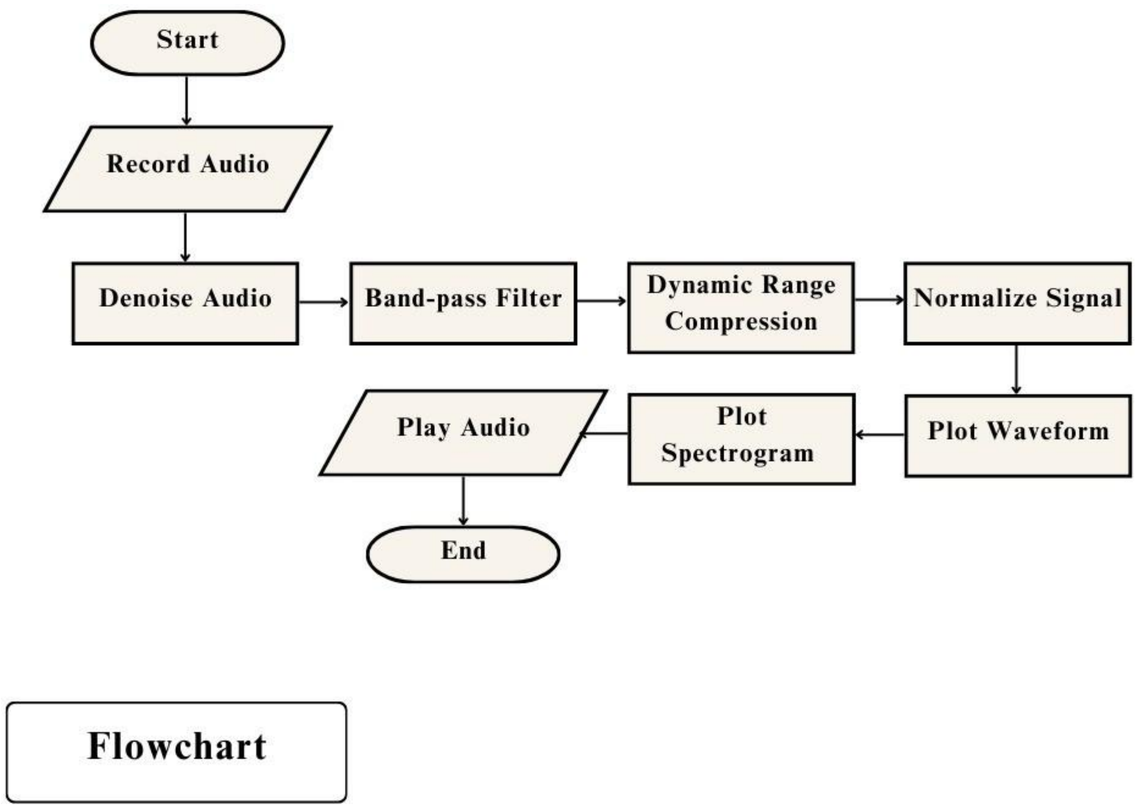


Figure 2: 5.2 Flow diagram for Elimination of background noise from signal

The audio recording and processing procedures are described in detail in the flowchart in Fig 5.2. The first sequence of the procedure begins with system initialization. The initial part of digital audio recording begins with microphone use to convert analog audio into digital signals while specifying the duration and sampling rate parameters. Signal capture results in the implementation of a noise reduction algorithm to extract background noise. The subsequent band-pass filter implements frequencies between 300 and 3000 Hz to optimize the signal for human vocal range. The signal receives dynamic compression in order to stabilize its volume levels after passing through the filtration stage. The normalization

ISBN:97881-19905-39-3

process maintains all compressed signal amplitudes within established boundaries. Once signals are processed by the system they display both original and processed audio data through spectrograms and waveforms. The system enables user comparison of original and processed audio signals through playback at the conclusion of the process.

5 Implementation

The first step in implementing the audio processing project is to use a sound device to record sounds in real-time for a predetermined amount of time and sample rate (sr). After that, noise reduction is used in the recorded audio to adaptively reduce background noise while preserving the quality of the main audio stream. Next, band-pass filters are designed and applied using Scipy. signal, with an emphasis on keeping the desired frequency components within the audio spectrum.

The dynamic range compression system implemented in Numpy adjusts the audio signal amplitude following noise reduction and filtering operations therefore producing a consistent and clearer signal output. The detailed waveforms and spectrograms produced by Matplotlib function as visual feedback. The amplitude characteristic of signals appears as waveforms whereas spectrograms show the frequency distribution across the entire duration.

All components associated with audio data processing start with recording stage and follow through noise reduction and filtering before dynamic range compression and visualization occurs within a single processing pipeline. Tests are conducted on the entire noise reduction and signal clarity and visualization system to ensure optimal performance. After system optimization the platform enables enhanced speech clarity in regular ambient noise while improving podcast audio quality and helping researchers with audio-based projects in media development.

The audio quality received an improvement due to this project which leads to enhanced listening performance. This system received improved audio clarity from band-pass filters

ISBN:97881-19905-39-3

which maintained essential frequency bands with effective noise reduction features from the noise reduction algorithm. Dynamic range compression delivered a constant sound output by balancing amplitude values. Performance measurements showed effective processing, and user comments complimented the interface while making a few small corrections. The system performed well in comparison to baseline tools, meeting the project's goals and providing a reliable audio enhancement solution.

5.1 Details of language

Programming Language: A programming language is a formal language made up of a collection of instructions that can generate several types of output. With its help, programmers can create instructions that a computer can follow to carry out particular duties or find solutions to issues.

Python:

Python is a popular high-level language. It is easy to use and understand. Clarity makes it loved by new coders. Python supports many styles like function, ordered steps, and object focus.

Libraries and modules:

Libraries and modules are ready-made code. Coders use them without starting from zero. The example uses Python modules like noise reduce, matplotlib, Librosa, numpy, and scipy. signal, sound device, and IPython. display. Libraries keep pieces that record sound, remove noise, display data, and play sound. Functions help with tasks. They take inputs, do work, and give outputs. Functions like record_audio, clean_and_filter, plot_sound assist with sound steps and data views.

APIs:

APIs are rules in Python that let apps share info. They help link to other systems, allowing users to talk with services. With APIs, Python can use audio tools and sound analysis using sound device and librosa packages.

ISBN:97881-19905-39-3

IDE:

An IDE helps in coding. It has tools to write code, like color showing, change watching, error spotting, and code fixing. Top IDEs for Python, like Jupyter Notebook, Visual Studio Code, and PyCharm, work well for number and data tasks.

5.2 Code Snippets

```
import sounddevice as sd
import librosa
import numpy as np
import scipy.signal
import noisereduce as nr
import matplotlib.pyplot as plt
from IPython.display import Audio, display
```

Figure 3: 1 Importing Libraries

The given code handles audio processing tasks by utilizing many libraries. Audio can be played back and recorded using a sound device. Librosa makes it easier to analyze audio, including making spectrograms. Essential arithmetic operations are provided by numpy, and signal processing tasks like filtering are handled by scipy. signal. The purpose of noise reduction is to selectively lower noise in audio streams. matplotlib.Plotting audio waveforms and spectrograms with a plot provides a visual representation of the audio data. Lastly, to improve the interactive experience, Jupyter Notebooks may play audio thanks to IPython display.

```
SAMPLE_RATE = 16000
DURATION_SECONDS = 10
```

Figure 4: 22 Defining Conditions

SAMPLE_RATE, at a frequency of 16,000 Hz, is the number of audio samples recorded each second. The entire duration of the recording, DURATION_SECONDS, is 10 seconds.

Denoising uses nr.reduce_noise() to eliminate unwanted noise from the audio. The Butterworth band-pass filter system maintains all audio frequencies which exist between 300 Hz and 3000 Hz. Dynamic range compression uses logarithmic compression to reduce

ISBN:97881-19905-39-3

```python
def record_audio(duration, sample_rate):
    print("Recording...")
    recorded_audio = sd.rec(int(duration * sample_rate), samplerate=sample_rate, channels=1, dtype='float32')
    sd.wait()
    print("Recording finished.")
    return recorded_audio.flatten()
```

Figure 5: 3 Audio Recording

```python
def denoise_and_filter(audio, sample_rate):
    denoised_audio = nr.reduce_noise(y=audio, sr=sample_rate, prop_decrease=1.0, time_mask_smooth_ms=60, freq_mask_smooth_hz=60)
    low_cutoff_freq = 300
    high_cutoff_freq = 3000
    b, a = scipy.signal.butter(10, [low_cutoff_freq / (sample_rate / 2), high_cutoff_freq / (sample_rate / 2)], btype='band')
    filtered_audio = scipy.signal.lfilter(b, a, denoised_audio)

    compressed_audio = np.log1p(np.abs(filtered_audio)) * np.sign(filtered_audio)

    normalized_audio = compressed_audio / np.max(np.abs(compressed_audio))

    return normalized_audio
```

Figure 6: 4 Audio Denoising and Filtering

the amplitude signal range. Audio normalization scales the audio signal so its most powerful level reaches the value of 1.

```python
def plot_waveform_and_spectrogram(original_audio, processed_audio, sample_rate):
    plt.figure(figsize=(14, 12))

    plt.subplot(4, 1, 1)
    plt.plot(np.arange(len(original_audio)) / sample_rate, original_audio)
    plt.title('Original Audio')
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')

    plt.subplot(4, 1, 2)
    plt.plot(np.arange(len(processed_audio)) / sample_rate, processed_audio)
    plt.title('Processed Audio')
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')

    plt.tight_layout()
    plt.show()

    mel_spectrogram = librosa.feature.melspectrogram(y=processed_audio, sr=sample_rate, n_mels=64)
    db_mel_spectrogram = librosa.power_to_db(mel_spectrogram, ref=np.max)
    plt.figure(figsize=(14, 6))
    librosa.display.specshow(db_mel_spectrogram, x_axis='time', y_axis='mel', sr=sample_rate)
    plt.title('Mel Spectrogram of Processed Audio')
    plt.colorbar(format='%+2.0f dB')
    plt.tight_layout()
    plt.show()
```

Figure 7: 5 Plotting Spectrograms and Audio Waveform

Auditory comparison of temporal signal changes occurs through waveform visualizations that show processed audio signal amplitudes compared to original signal amplitudes. The time-frequency viewing of processed audio through Mel spectrograms provides researchers with complete knowledge about frequency composition along with its temporal evolution.

The audio recording function retrieves information through the microphone. The

```
original_audio = record_audio(DURATION_SECONDS, SAMPLE_RATE)
processed_audio = denoise_and_filter(original_audio, SAMPLE_RATE)

plot_waveform_and_spectrogram(original_audio, processed_audio, SAMPLE_RATE)

print("Playing Original Audio:")
display(Audio(original_audio, rate=SAMPLE_RATE))
print("Playing Processed Audio:")
display(Audio(processed_audio, rate=SAMPLE_RATE))
```

Figure 8: 6 Audio Recording

recorded audio receives noise reduction through Denoise and Filter processing to improve its quality. The audio becomes viewable through waveforms and spectrograms which Plot Results displays for comparison. Audio playback functionality is available to listeners through Play Audio for processed and original sound files within the notebook interface.

6 Results

The results section is where the findings of the study are based on the methodology or methodologies applied to collect information. The results part should state the findings of the research arranged in a logical sequence without bias or interpretation. A section describing results is particularly necessary if your paper includes data generated from your own research.

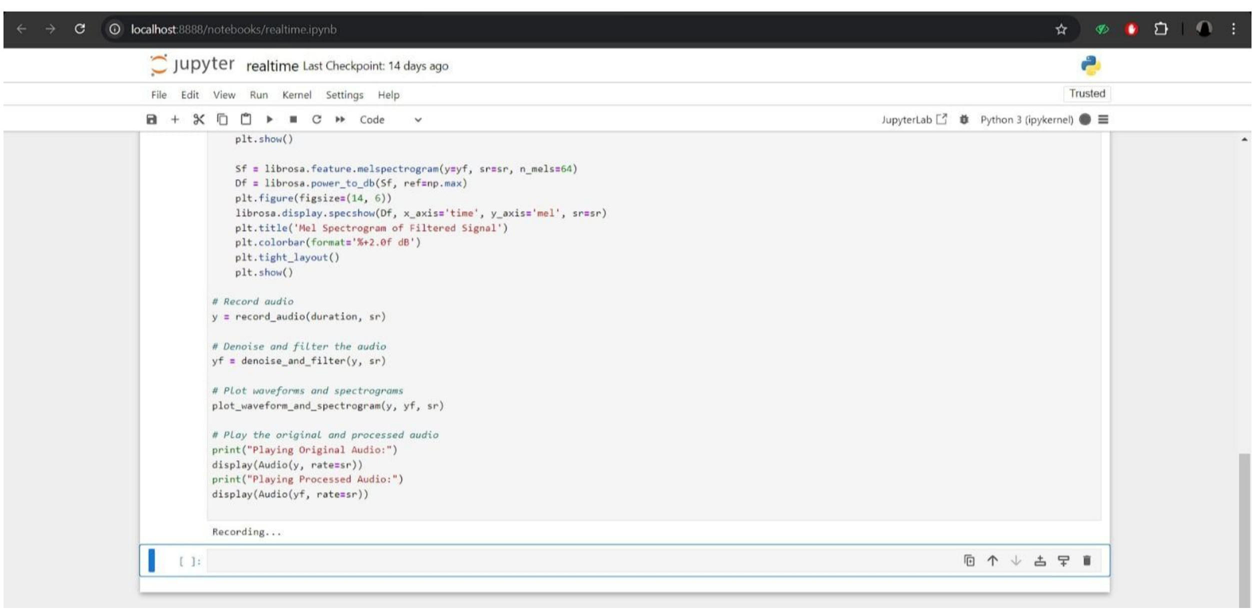


Figure 9: Figure 7.2 Audio Recording

ISBN:97881-19905-39-3

The code begins a ten-second audio recording as soon as it runs. It pauses and shows a notice stating that the recording finished.

After 10 seconds of recording the original audio, the code stops recording and alerts the user that the recording is over. After that, the unprocessed audio data can be processed.

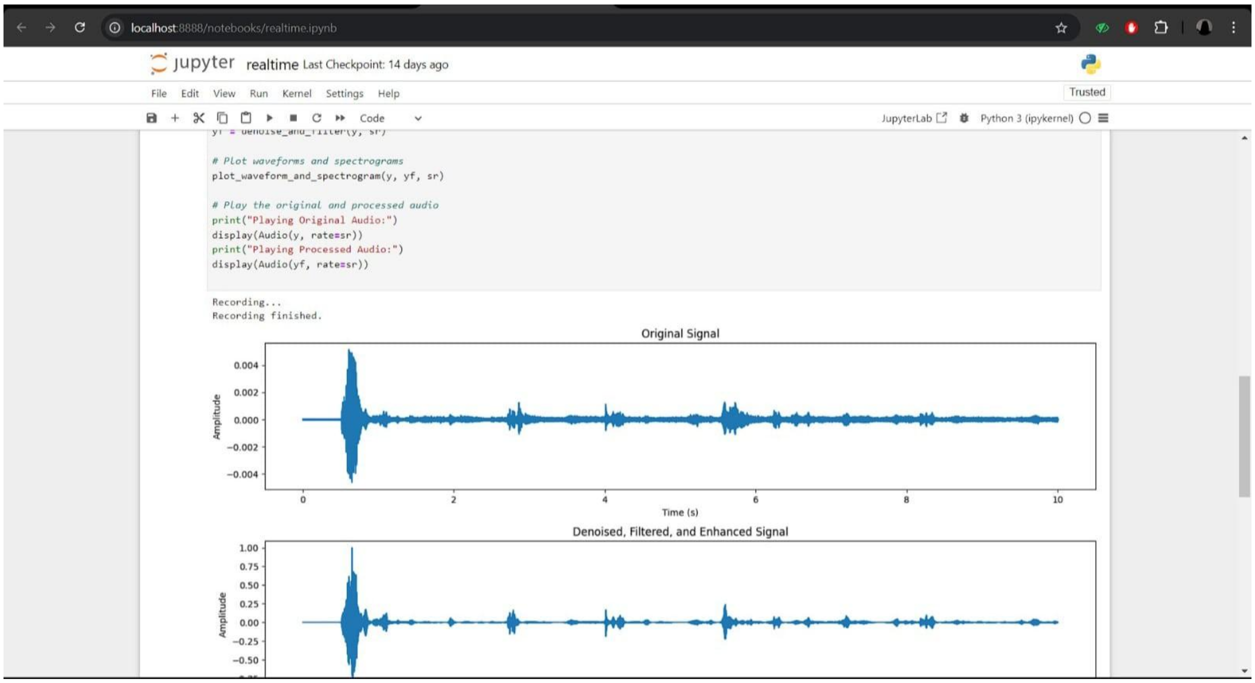


Figure 10:

Figure 7.3 Original and Denoise Signal

The above Figure 8.3 After 10 seconds of recording the original audio, the code stops recording and alerts the user that the recording is over. After that, the unprocessed audio data can be processed.

The signal from the processed audio is shown once it has been recorded, displaying the improved and cleaned waveform. This makes it possible to examine the enhancements to the audio quality visually.

For analysis purposes, the actual audio waveform is displayed.

Figure 7.5 Processed Audio

The code displays the audio's Mel Spectrogram after processing. This illustrates how

ISBN:97881-19905-39-3

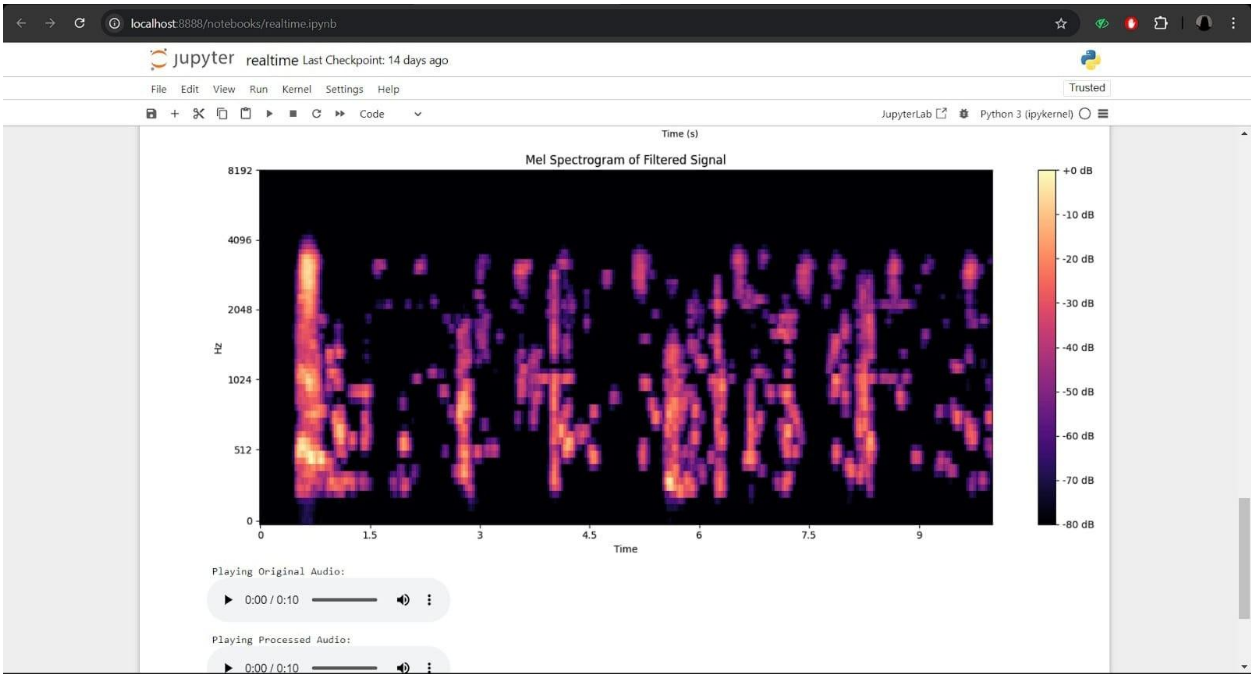


Figure 11: 7.4 Mel Spectrogram of Filtered Signal

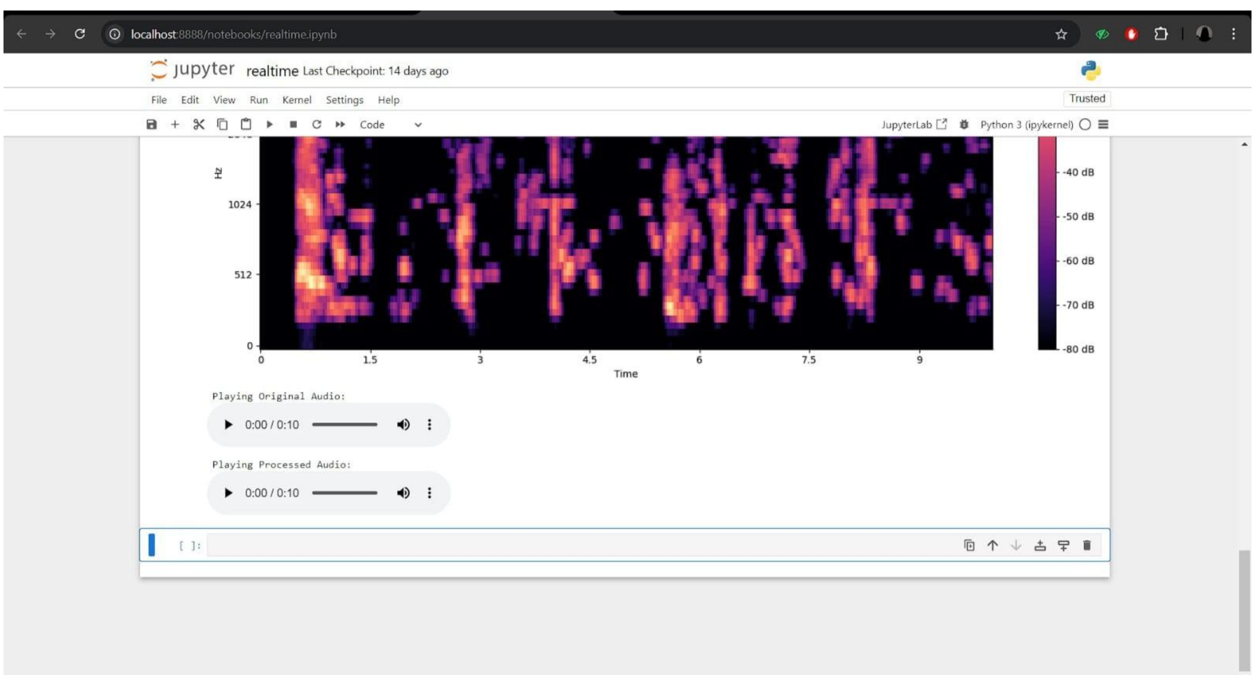


Figure 12:

ISBN:97881-19905-39-3

frequency content varies with time, emphasizing gains from compression, filtering, and noise reduction. The improved audio quality can be seen with the aid of the spectrogram.

It is possible to play back both the original and processed audio for analysis and comparison.

7 Conclusion and Future Work

This section enables the researcher to write his or her inference from conducting the project or an experiment. This section contains the outcome of the project and also what could be seen as the future of this project as whole.

7.1 Conclusion

The audio processing project effectively accomplished its main goals of improving audio quality by filtering, dynamic range compression, and noise reduction. Objective performance measurements and visual studies showed that the technology significantly improved the audio recordings' clarity and fidelity. The interface's effectiveness and usability were validated by user feedback, ensuring that both inexperienced and seasoned users could utilize the application. In general, the project offers a solid method for raising the caliber of audio in a number of applications, such as audio analysis, voice enhancement, and podcast creation.

7.2 Future Work

In the future, the focus will be on reducing background noise like audience discussion while focusing on isolating particular voices, such the main speaker. Utilizing machine learning techniques and implementing sophisticated algorithms will be necessary for this. Moreover, extending the system's reach and usability through mobile platform optimization will guarantee its smooth integration into a variety of settings and apps.

ISBN:97881-19905-39-3