Ukrainian Catholic University

Faculty of Applied Sciences

Business Analytics & Computer Science Programmes

Improving Audio Quality Through Denoising Techniques: A Study of Short-time Fourier Transform and Thresholding Methods

Linear Algebra final project report

Authors:
Andrii Pletinka
Mykhailo Humeniuk

16 May 2023

Abstract

We will discuss a denoising method that involves performing Short-time Fourier Transform (STFT) and thresholding. We implemented it and developed a Jupyter notebook containing the full process here.

1 Introduction

Denoising audio refers to the process of removing unwanted noise from an audio signal in order to improve its quality and intelligibility. In audio recordings, noise refers to any unwanted or undesired sound that is captured along with the desired audio signal. It can degrade the quality of the recording, making it less clear and more difficult to comprehend. Noise can be introduced into an audio signal at various recording, transmission, or storage stages and can take many forms. Audio denoising techniques remove these unwanted noise components from the signal while preserving as much of the original signal as possible.

2 What is sound

2.1 Sound from LA point of view

In linear algebra terms, a sound wave can be represented as a vector with elements corresponding to the amplitudes of the wave at different time points. Each element of the vector represents the magnitude of the wave at a specific time instant. The entire sound wave can be represented by a sequence of these amplitude values.

Sound waves can be decomposed into different frequency components using techniques such as the Fourier transform. This decomposition allows us to analyze the different frequency components present in a sound and their respective amplitudes. Each frequency component can be thought of as a separate vector in the frequency domain.

2.2 Frequency and time domains

The frequency domain is a mathematical representation of a signal in terms of its frequency components. It provides a different perspective on the signal by expressing it as a sum of sinusoidal components with different frequencies, amplitudes, and phases. The phase refers to the temporal relationship or timing of the sinusoidal components that make up a signal. It represents the shift or offset in the waveform relative to a reference point.

The frequency domain provides valuable insights into the underlying characteristics of a signal. It allows us to analyze and manipulate specific frequency components independently, which is useful for the task of noise removal. By analyzing signals in the frequency domain, we can gain a deeper understanding of their frequency content, identify dominant frequencies and remove unwanted noise or interference.

Unlike in the frequency domain, in the time domain, a signal is represented as a function of time, showing how the signal varies over time.

3 Related literature review

[1] describes the performance comparison of Time-frequency algorithms for removal of Additive White Gaussian Noise. This paper introduces an algorithm for solving the problem of noise reduction using STFT and thresholding techniques.

[2] introduces application of different denoising techniques, including Block Thresholding, on different data. The paper also describes the performance testing and comparison of the used algorithms.

[3] describes the simillar algorithm and also contains information about Hanning window and STFT.

4 Short-time Fourier Transform

4.1 Fourier Transform of a part of the signal

STFT is a time-frequency analysis technique used in signal processing and audio analysis that provides a way to analyze the frequency content of a signal as it changes over time.

The STFT is similar to the standard Fourier Transform. Instead of computing the Fourier Transform of the entire signal at once, the signal is divided into smaller segments, and the Fourier Transform is applied to each part separately. This allows us to analyze the frequency content of the signal over time, which is essential in many applications, such as audio analysis and speech processing.

The STFT is computed by dividing the signal into overlapping windows of a fixed length and then applying the Fourier Transform to each window.

The Fourier Transform of a signal looks like this:

$$X(\omega) = F[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,dt$$

But when we want to perform a Fourier Transform on just a part of the signal:

$$\hat{x}(t,\tau) = x(t) \cdot W(t-\tau)$$

where $\hat{x}$ is the sampled signal, W is a windowing function and τ is the length of the window, we will get the STFT function:

$$X_S = F[x(t) \cdot W(t-\tau)] = \int_{-\infty}^{\infty} x(t) \cdot W(t-\tau)e^{-i\omega t}\,dt$$

4.2 Hanning window

The commonly used windowing function in signal processing and spectral analysis is a Hann function (or Hanning window) defined as

$$W(n) = \frac{1}{2}\left[1 - cos(\frac{2\pi n}{N})\right]$$

Where $W(n)$ is the value of the window function at index n of the sample within the window and N is the total number of samples in the window. The Hann window function smoothly tapers the edges of a signal to minimize spectral leakage when performing Fourier analysis.

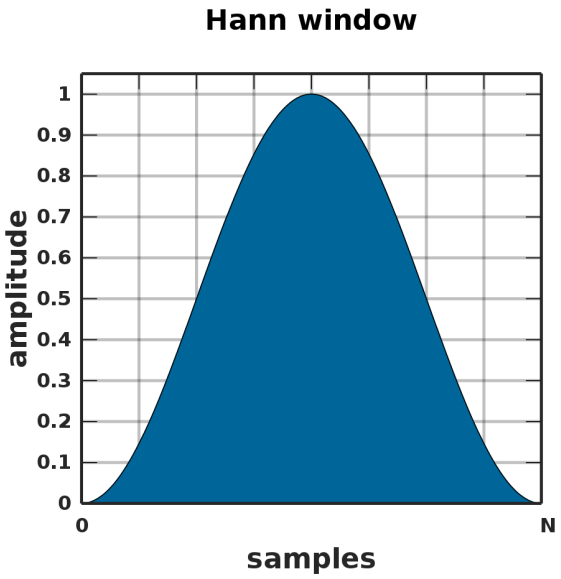

Figure 1: How a Hann window looks like.

4.3 Spectral leakage

Spectral leakage refers to a phenomenon when the frequency content of a signal spreads or leaks into neighboring frequencies in the frequency domain representation, that can occur during the process of analyzing or transforming a signal from the time domain to the frequency domain. Spectral leakage can lead to inaccurate frequency analysis and loss of resolution. It can mask or obscure smaller or adjacent frequency components, making them harder to detect or identify. This effect is particularly significant when analyzing signals with low-amplitude or closely spaced frequency components. The most common cause of spectral leakage is when the signal being analyzed does not have an exact integer number of periods within the analyzed window.

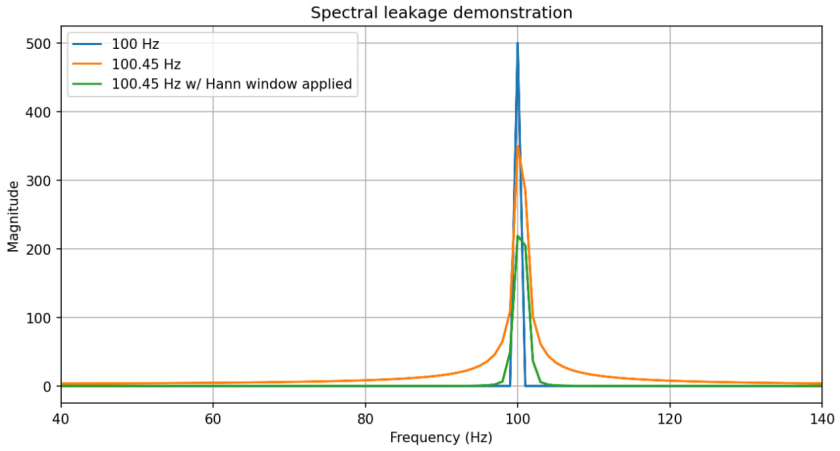

Figure 2: Example of a spectral leakage phenomenon.

To understand why this happens, recall that the FFT assumes that the analyzed signal is periodic over the length of the analyzed window. When this assumption is valid (i.e., the signal has an exact integer number of periods within the window), the frequency components line up precisely with the frequency bins of the FFT, and there

is no leakage. However, when the signal does not meet the periodicity requirement, a discontinuity occurs at the edges of the window. This discontinuity introduces high-frequency components in the signal, which were not present in the original signal. These high-frequency components contribute energy to nearby frequency bins during the spectral analysis, resulting in spectral leakage.

5 STFT, DFT and FFT

Fast Fourier Transform is an efficient algorithm which computes the Discrete Fourier Transform of a sequence of time-domain samples and converts it into a frequency-domain representation, revealing the frequency components present in the signal.

We have

$$\hat{f} = F_n f \tag{1}$$

where $\hat{f}$ is the Fourier Transform vector of frequency components, f is the vector of data and F_n is a DFT matrix.

The k-th Fourier coefficient is obtained by taking the sum over all data points like the following:

$$\hat{f_k} = \sum_{j=1}^{n-1} f_j e^{-i2\pi jk/n}$$

To retrieve the data from the Fourier coefficients we now need to take the sum over all frequencies:

$$f_k = \frac{1}{n} \sum_{j=1}^{n-1} \hat{f}_j e^{i2\pi jk/n}$$

So if you perform a DFT on a data vector $\{f_1, f_2, \ldots, f_n\}$ you will get $\{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n\}$ which will tell how much of each frequency you will need to add up to reconstruct the data. Notice that the terms $e^{-i2\pi jk/n}$ and $e^{i2\pi jk/n}$ are some multiples of the $e^{-i2\pi/n}$ which is some fundamental frequency ω_n (n-th root of unity) that relates to what kind of sines and cosines can be approximated with the n discrete values from the domain. An n-th root of unity refers to a complex number that, when raised to the power of n, equals 1. Using this ω_n we now can construct the DFT matrix F_n:

$$\begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega_n^1 & \omega_n^2 & \ldots & \omega_n^{(n-1)} \\ 1 & \omega_n^2 & \omega_n^4 & \ldots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{(n-1)} & \omega_n^{2(n-1)} & \ldots & \omega_n^{(n-1)^2} \end{pmatrix}$$

So the equation (1) will look like the following:

$$\begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega_n^1 & \omega_n^2 & \ldots & \omega_n^{(n-1)} \\ 1 & \omega_n^2 & \omega_n^4 & \ldots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{(n-1)} & \omega_n^{2(n-1)} & \ldots & \omega_n^{(n-1)^2} \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{pmatrix} \tag{2}$$

The DFT matrix has some important properties. The most significant property is that it is a unitary matrix, which guarantees that the transformation is reversible and enables efficient computation of the DFT and its inverse using the FFT.

The Fast Fourier Transform is the algorithm based on the divide-and-conquer strategy that allows us to compute the Discrete Fourier Transform in $O(nlog(n))$ time which is much faster than a straightforward matrix multiplication which takes $O(n^2)$ time.

If n is a power of 2, then there is a possibility to reorganize entries of matrix F_n in a way which will significantly reduce the number of multiplications:

$$\begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{pmatrix} = \begin{pmatrix} I_{n/2} & -D_{n/2} \\ I_{n/2} & -D_{n/2} \end{pmatrix} \begin{pmatrix} F_{n/2} & 0 \\ 0 & F_{n/2} \end{pmatrix} \begin{pmatrix} f_{even} \\ f_{odd} \end{pmatrix} \tag{3}$$

where

$$D_{n/2} = \begin{pmatrix} 1 & & & & \\ & \omega & & & \\ & & \omega^2 & & \\ & & & \ddots & \\ & & & & \omega^{n/2-1} \end{pmatrix}$$

The main difference between STFT and FFT is that the FFT computes the Fourier Transform of the entire signal at once, while the STFT computes the Fourier Transform of small, overlapping signal segments. This makes the STFT better suited for analyzing signals that change over time, while the FFT is better suited for analyzing stationary signals.

However, the STFT has some disadvantages compared to the FFT. One of the main disadvantages is that the STFT introduces temporal and frequency resolution trade-offs due to a fixed window length. This means that the STFT may not be able to accurately capture the fine details of the signal in both time and frequency domains simultaneously. In addition, the STFT requires more computational resources than the FFT due to the need to compute the Fourier Transform of multiple signal segments.

6 Thresholding and denoising

6.1 Intro to thresholding

Thresholding is a standard method used in audio denoising to remove noise from a signal. It works by setting a threshold value above which signal components are kept and below which they are discarded. This is based on the assumption that the signal components are larger in magnitude than the noise components, which are usually smaller in magnitude and spread across a wide frequency range.

The basic idea behind thresholding is to compare the magnitude of each frequency component of the signal to a predefined threshold value. If the magnitude exceeds the threshold, the piece is kept; otherwise, it is discarded. There are different types of thresholding methods.

6.2 Spectrograms and Mel spectrograms

Consider an example of a speech signal from an open-source library called [4]LibriSpeech (Fig. 3).

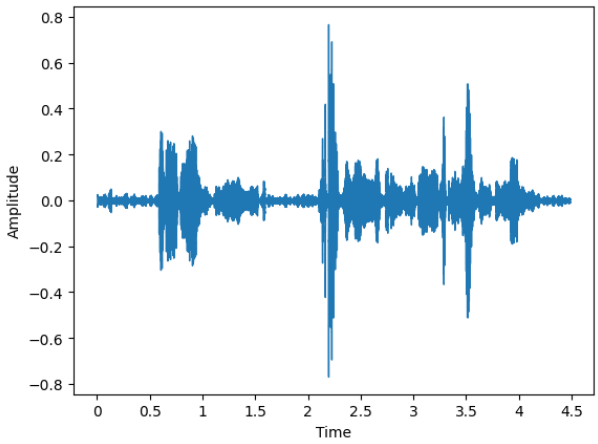

Figure 3: Waveform of a speech signal.

Let's calculate the STFT coefficients for this signal and represent them as a spectrogram (Fig. 4).

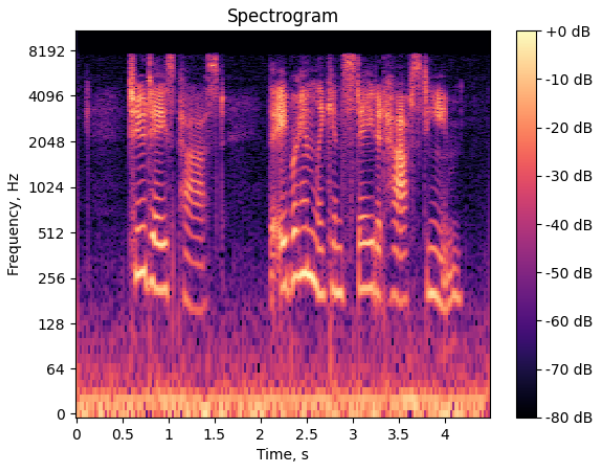

Figure 4: STFT coefficients represented as a spectrogram.

The spectrogram represents the signal in the time-frequency domain. To get a spectrogram one should calculate FFT for small frames of input signal - Short-time Fourier Transform. More on STFT you can read in our report. After computing STFT, we get the intensity (the measure of presence) for different frequencies in different time periods. This gives us an opportunity to represent the audio in both time and frequency domains and present this representation as the spectrogram. The y-axis represents the frequencies and the x-axis represents the time. The lighter colors indicate more present frequencies while the darker colors indicates less intense presence of the frequency.

There is also another type of a spectrogram called Mel spectrogram. The main difference between Mel and regular spectrogram is that it more 'human-oriented'. When we listen to sounds, we don't perceive all frequencies equally. We are more sensitive to lower

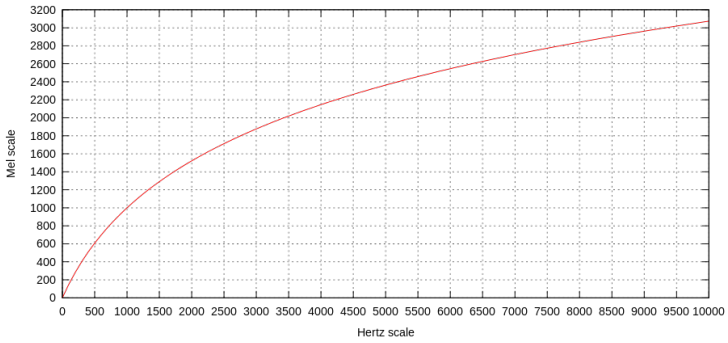

Figure 5: Hz to Mel scaling.

frequency ranges than higher. The Mel scale reflects this by emphasizing lower frequencies and gradually spreading out the higher frequencies, as shown on this picture.

Mel spectrogram gives way less information about higher frequencies than the regular spectrogram.

6.3 Analyzing audio corrupted with noise

The Mel spectrogtams shown on Fig. 6 and Fig. 7 are used to compare original and corrupted files. You can notice that the dB scales changed for corrupted file, since there is way less variation in intensity for different frequencies. Thus, our dB scale should be more informative to correctly represent the difference for different frequencies. Also, the dark parts original audio are no longer dark in corrupted. This indicates the presence of noise.

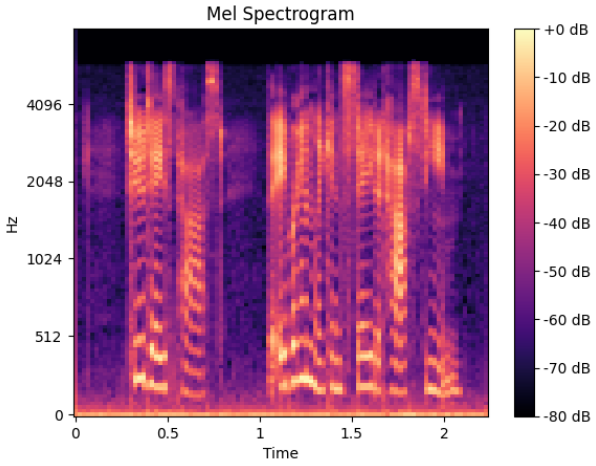

Figure 6: Mel spectrogram of original audio.

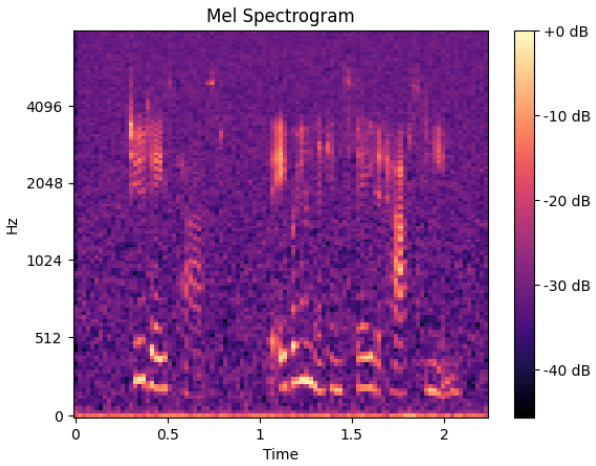

Figure 7: Mel spectrogram of corrupted audio.

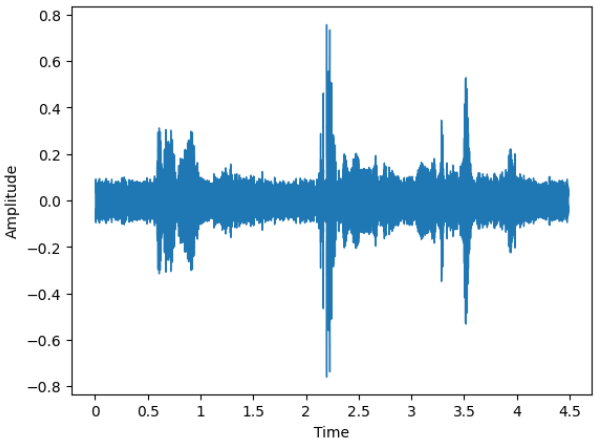

Figure 8: Waveform of corrupted audio.

6.4 Performing trivial denoising

In this section we perform a trivial denoising technique. The main idea is to compute STFT and for each frame of STFT we determine the mean and standard deviation. Then for every coefficient in this frame we apply hard thresholding:

$$f_k(x) = \begin{cases} x, & \text{if } x > \mu_k + 2.5 \cdot \sigma_k. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Where k is the number of the frame and $\mu_k + 2.5 \cdot \sigma_k$ is a threshold which is usually denoted λ. Please note that we chose this threshold just for example. After setting coefficients of unwanted frequencies to zero, we perform Inverse STFT. ISTFT is done by computing IFFT for each frame of STFT. This operation helps us to retrieve the samples of denoised audio.

6.5 The results of trivial denoising

Let us determine the quality of denoised audio by its spectrogram.

The spectrogram shows that there some 'dots' in the upper part. These dots will represent an unpleasant noise in the audio. This unwanted sound is called a 'musical'

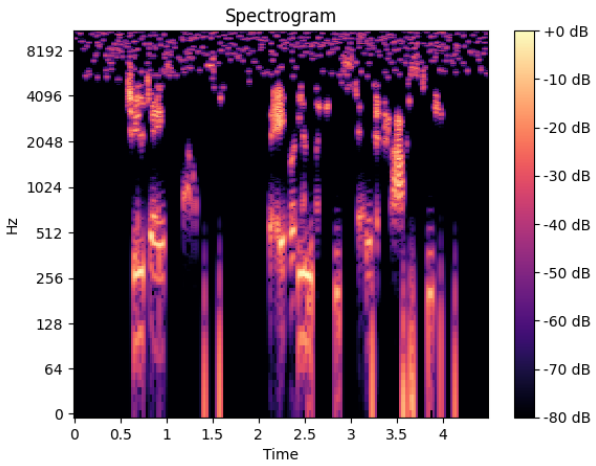

Figure 9: Mel spectrogram of denoised audio using trivial method.

noise. This phenomenon occurs when the distinct sinusoid are being present in the audio. If you are familliar with Star Wars, the great example of musical noise would be the sound made by R2-D2 robot. Since, it is really hard to encounter the pure sinusoids in nature, it is rather unpleasant to hear. So, the trivial denoising technique is proved to deliver bad results.

6.6 Block threshholding

For the block tresholding technique we also apply the hard tresholding. The main difference is that now we do it on the blocks, not time frames as in previous method. For easier understanding the block can be interpreted as a $n \times m$ part of the spectrogram. For each of these blocks we calculate the norm (putting all STFT coefficients that belong to the block to the power of 2 and sum them up). Then we compare the norms with some treshold and if the norm of the block is lower than the treshold, we delete the whole block out of spectrogram (set STFT coefficients to zero). Then we do the same procedure as in trivial method, by computing ISTFT and retrieving the denoised audio.

6.7 The results of Block threshholding denoising

Let us again determine the quality of denoised audio by its spectrogram. The obtained result does not contain the 'dots'. Also, this method does not cancel out the low frequencies. While comparing to the spectrogram of the original audio (Fig. 6), a part of frequencies with low and medium magnitudes were canceled out. Those low magnitude frequencies can be interpreted as the noise which is present in original audio, meanwhile the medium magnitude frequencies are most probably the harmonics of the speaker's voice. The harmonics of human speech are the integer multiples of the fundamental frequency, which contribute to the distinctive timbre and pitch variations in speech. Since, our block thresholding method removed those frequencies with medium magnitudes, the voice of the speker in denoised audio is slightly different from the original audio.

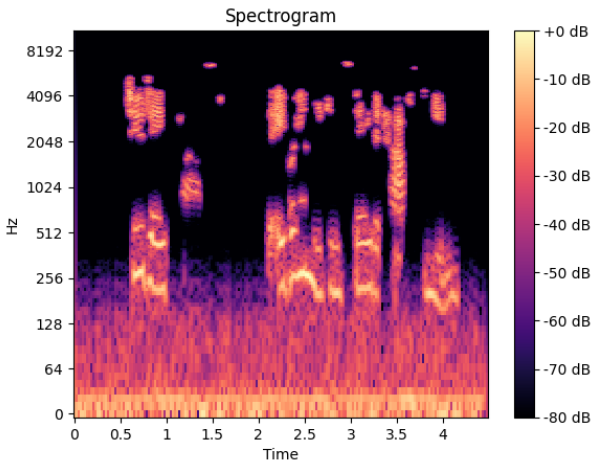

Figure 10: Mel spectrogram of denoised audio using block thresholding method.

6.8 A few remarks on block thresholding method

The parameters that determine the effectiveness of block thresholding method are the size of the block and the threshold. In our project we used 5×5 blocks. We did not perform the analysis of different sizes for blocks and their influence on algorithm performance. We did perform the analysis of different thresholds by computing the MSE between the original samples and the samples of the denoised audio. This method has some disadvantages. The main one is that lowering threshold (to a certain degree) results in lower MSE. For lower thresholds our algorithm started to include those frequencies with medium magnitudes which are good for the speaker's voice, but it also started to include some musical noise (the 'dots' appeared). We concluded that a slight downgrade in speaker's voice is better then the presence of musical noise.

7 Conclusions

Short-time Fourier Transform is a widely used time-frequency analysis technique that provides a way to analyze the frequency content of a signal as it changes over time. Thresholding is also a common method used in audio denoising along with the Fourier Transform to remove unwanted noise from a signal.

In this project, we got to know a widely-used spectral analysis tool called Short-time Fourier Transform and performed a denoising process using block thresholding method.

References

[1] Apoorva Athaley and Papiya Dutta, *Audio Signal Denoising Algorithm by Adaptive Block Thresholding using STFT*, Published in International Journal of Trend in Scientific Research and Development, 2017

[2] Guoshen Yu, Stephane Mallat and Emmanuel Bacry, *Audio Denoising by Time-Frequency Block Thresholding*, 2008

[3] Marie de Masson d'Autume, Christophe Varray and Eva Wesfreid, *Block thresholding audio denoising algorithm*, 2013

[4] https://www.openslr.org/12