Image Processing - Exercise 2

Achsaf Atzmon, 316129501

Introduction

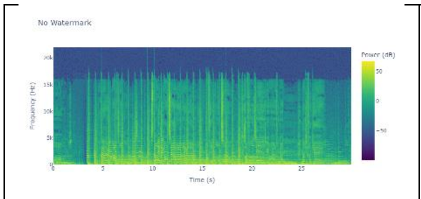
The goal of this exercise is to design, implement, and evaluate methods for creating and detecting audio watermarks. Audio watermarks are typically inaudible electronic identifiers embedded in audio signals to indicate copyright ownership.

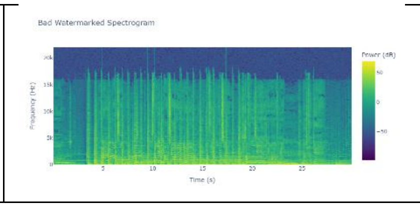
The main techniques utilized include signal processing concepts, such as Fourier transforms and filtering, as well as classification methods to distinguish watermarks. Spectrograms were used to help visualize the data and making it easier to analyze

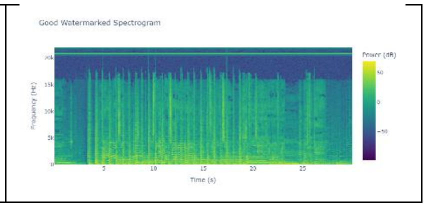
Adding Watermarks

Good Watermark:
The good watermark was designed to be detectable by algorithms while still being inaudible to humans. Therefore the watermark was placed above 20000 hz (The end of human hearing range)
The audio signal was transformed into the frequency domain using a Fourier transform. A pseudorandom noise pattern was added to specific high-frequency bands. The modified signal was then transformed back into the time domain.

Bad Watermark:
The bad watermark works similarly but was kept within the hearing range of humans and therefore, very noticeable.





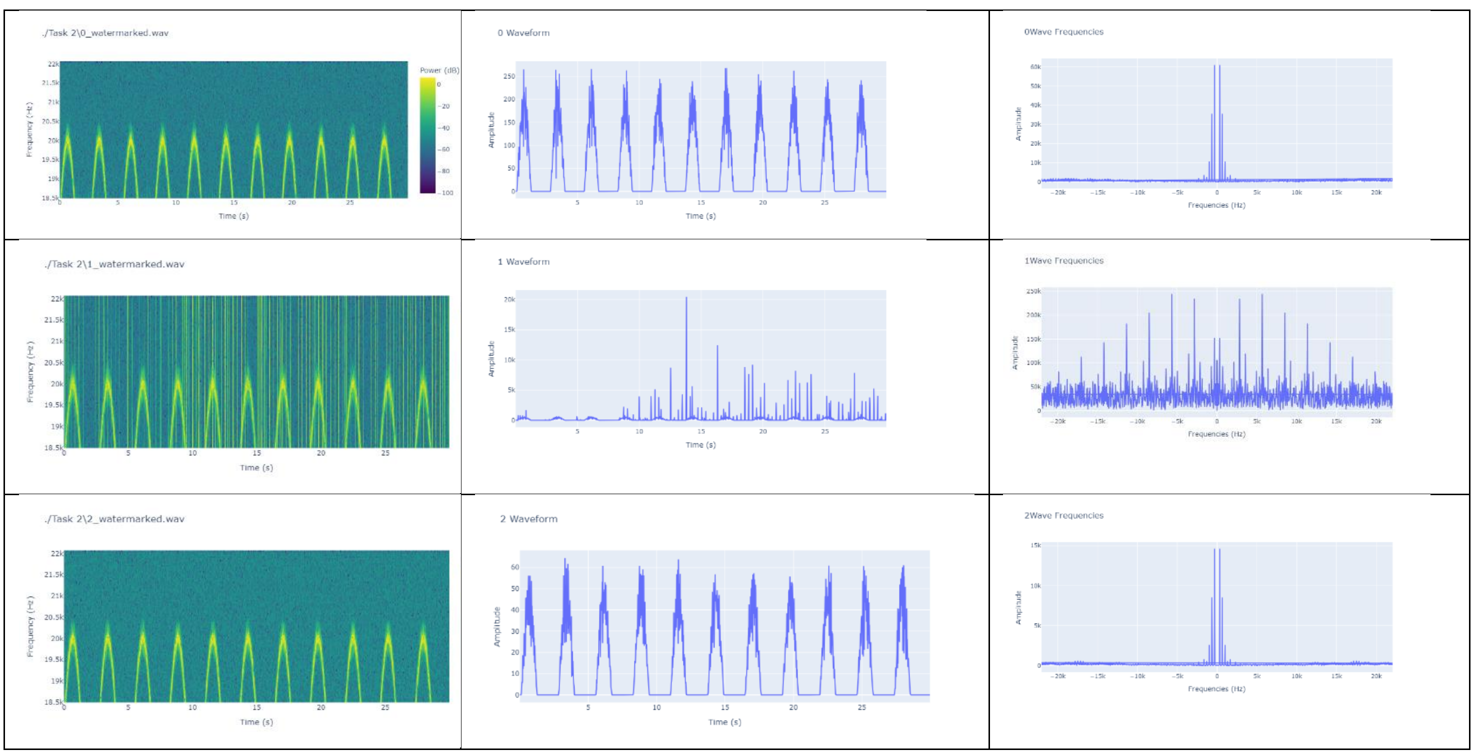


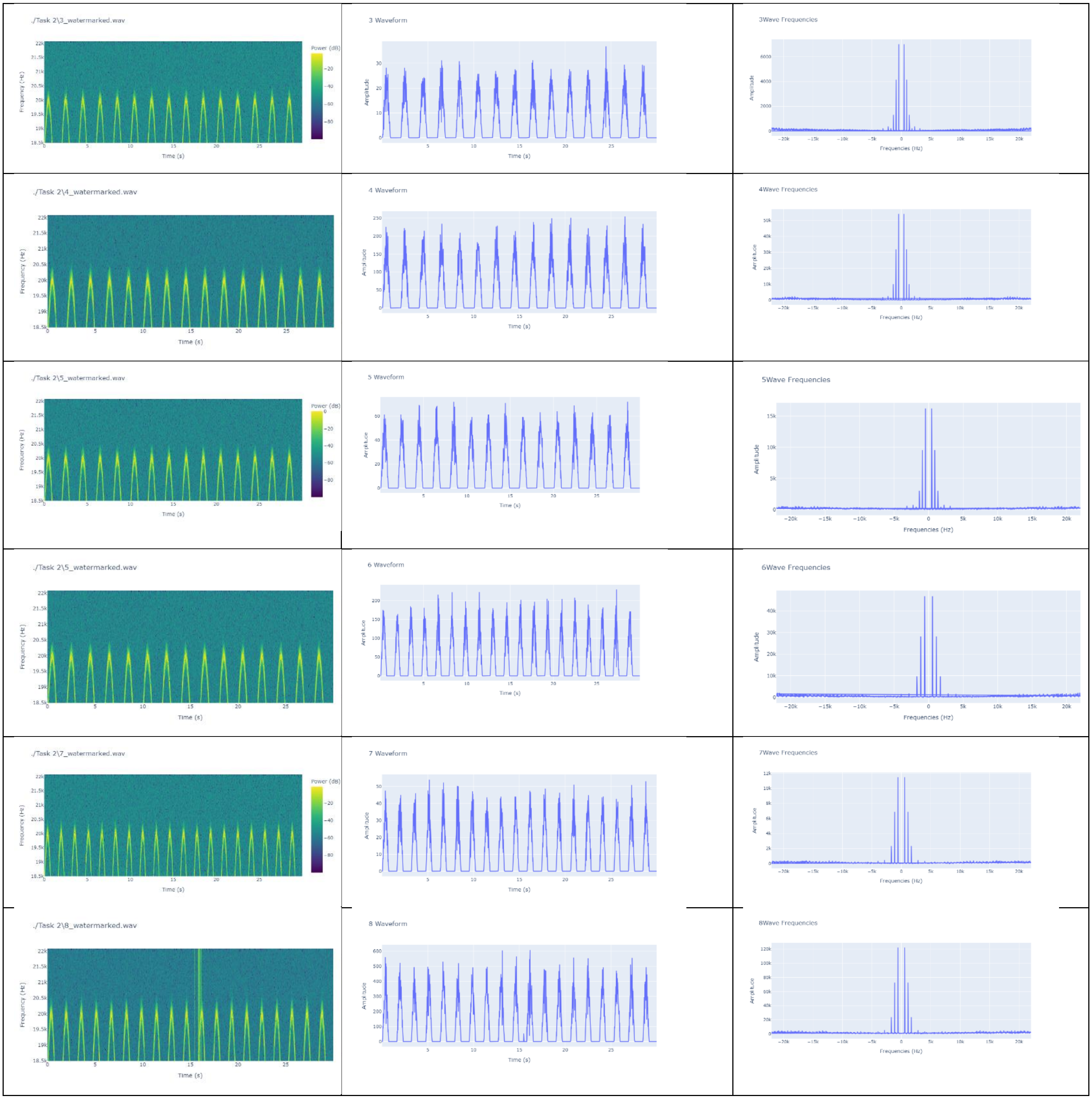
Originally I added a integer value to the frequency, but that just raised the overall amplitude of the audio instead of adding a frequency like I wanted. It was required to add an oscillation in order to affect the sound.

For Images I would likely use a similar feature, but It would include sin waves that are imperceptible to humans. This would involve using very high frequency waves, much higher than the pixel length, so as to not appear on screen.

Classifying watermarks

I started by creating a spectrogram of the audio to see if there were any obvious watermarks. Among the higher frequencies a very clear sin wave can be seen on each of the wav files. I clipped the data to include only frequencies above 18.5k to get the cleanest version of the watermark. I converted this sin wave into a graph and then did a Fourier transform on each one to determine what the frequency of each one was. The results of the grouping were (0,1,2) , (3,4,5) , (6,7,8). We can see this very clearly in the graphed transform.





The average frequency 0hz was manually set to zero to make finding the watermark easier.

These are the results of taking the highest amplitude frequency from each transform

File 0's watermark is 328.66hz

File 1's watermark is 5680.68hz
File 2's watermark is 328.66hz
File 3's watermark is 448.47hz
File 4's watermark is 448.47hz
File 5's watermark is 448.17hz
File 6's watermark is 567.68hz
File 7's watermark is 567.68hz
File 8's watermark is 567.68hz

As you can see the only one that this method didn’t work for was File 1 since it had many other frequencies, making the watermark less significant.

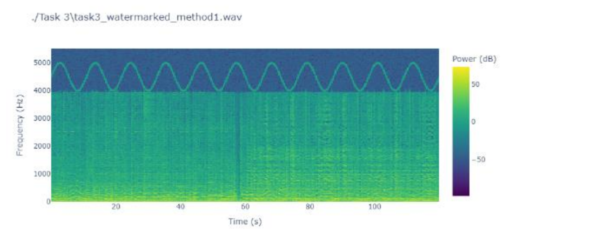
To remove the watermark I would subtract the value of the sin wave at each point in time from the spectrogram of the audio, then do ISTFT to get a clean audio signal with no watermark.

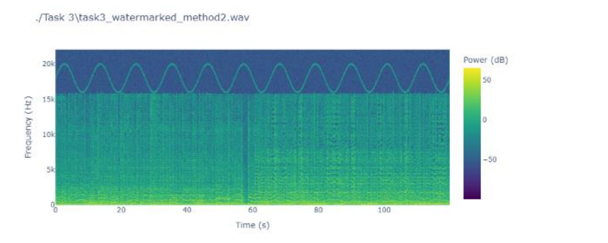
Determining speedup method

Time Speedup:
This method looks at the original .wav file and changes the sample rate. In order to speed up the audio it will skip the relative number of samples giving the illusion of speed up. This method also causes the frequencies to expand/contract linearly to the rate of change. The final output will be a faster/slower recording with higher/lower pitch.

Frequency Speedup:
This method converts the audio into a spectrogram then only manipulates the time dimension to speedup or slowdown the desired amount. An ISTFT is then performed to convert it back to playable audio. This method preserves the original frequencies since we didn’t change them, while still giving us the speedup/slowdown, the pitch remains the same and more natural sounding.

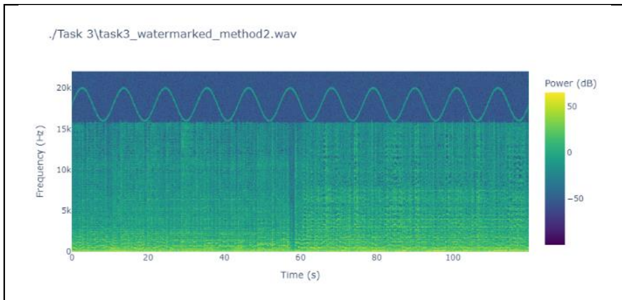
When listening to the audio it is clear that method_1 is the time slowdown, since it sounds very bassy and unnatural, looking at the spectrograms of the two method_1 is mostly very low freqencies which is differen’t from other audio files. In contrast method_2 has frequencies that are much more common amongst recordings and sounds more normal.

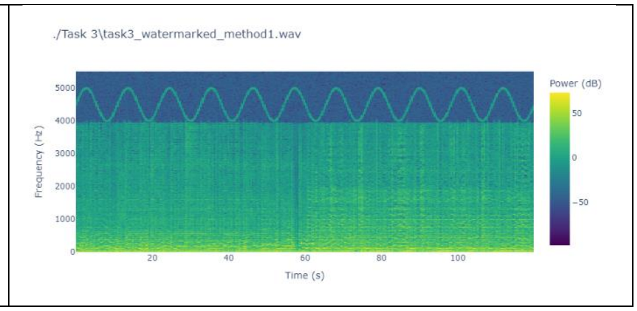




Since the time slowdown has a direct relationship with the effect of the frequencies, if we figure out how much it effected the frequencies we can find out what the speedup ratio is. We can find the original frequencies in the frequency graph. To make our life easier we'll compare the watermarks in both graphs to determine the speedup ratio.

The normal watermark frequency maxes out at 19939hz and after the slowdown at 4984hz. This means the speedup ratio is ¼. (The audio has been slow down by 4)





Conclusion

In this exercise, we explored the process of adding and classifying watermarks and analyzed methods to improve computational efficiency. Key insights include the practical application of watermarking techniques for secure image handling, the classification challenges posed by diverse patterns, and the potential for speedup using optimized algorithms. These findings demonstrate the interplay between image processing and performance optimization, emphasizing their significance in real-world applications. Future work could focus on enhancing accuracy and further reducing processing time.