Assigment 1

Topic: Experimenting with Spectrograms and Windowing Techniques (Question 2).
Date: 20-01-2025
Submitted By: Arjun Arora (M24CSA003)

---

1 Objective

The objectives of this assignment are as follows:

1. **Task A:**
   (a) Use the UrbanSound8K dataset for this assignment. Download the dataset from - `https://goo.gl/8hY5ER`.
   (b) Understand and implement the following windowing techniques:
      - Hann Window
      - Hamming Window
      - Rectangular Window
   (c) Write a Python program to apply the above windowing techniques → Generate spectrograms using the Short-Time Fourier Transform (STFT).
   (d) Compare the spectrograms visually and analyze their differences. Discuss the correctness of windowing performed.
   (e) Train a simple classifier (e.g., SVM or neural network) using features extracted from the spectrograms and evaluate the performance results comparatively in different techniques.
2. **Task B:**
   (a) Select 4 songs from 4 different genres and compare their spectrograms.
   (b) Analyze the spectrograms and provide a detailed comparative analysis based on your observations and speech understanding.

2 Dataset

The UrbanSound8K dataset consists of 8732 labeled audio excerpts (les than 4s) categorized into 10 urban sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music. The dataset is sourced from Freesound.org and is organized into 10 stratified folds (fold1-fold10) for reproducibility in machine learning tasks [**?**]. Each sample varies in sampling rate, bit depth, and number of channels based on the original recording.

The dataset includes a metadata file, `UrbanSound8K.csv`, which provides essential details such as file names, Freesound IDs, timestamps, salience ratings, fold assignments, and class labels.

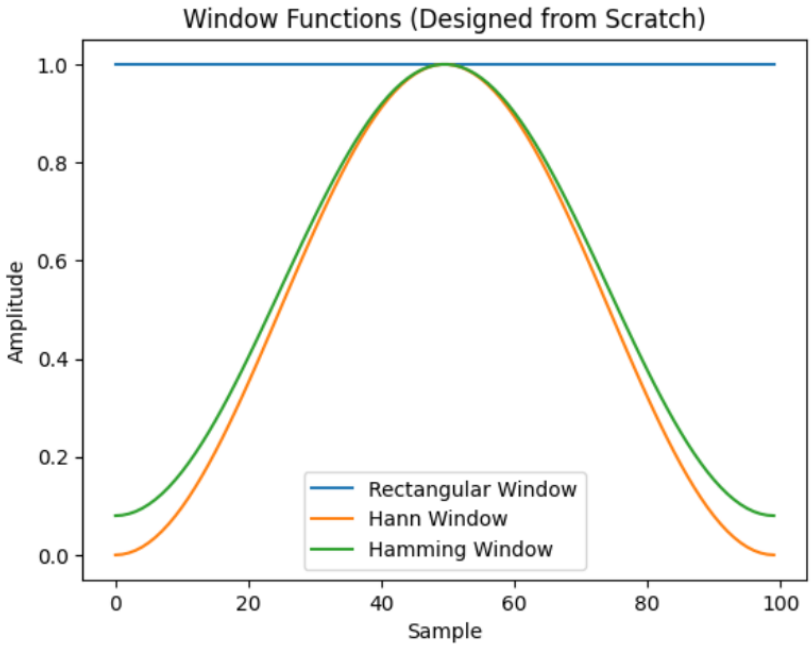
For this study, **fold 10 is used for training**, while **folds 1-9 are used for testing** to evaluate model performance.

3 Implementing Windowing Techniques

Windowing is a technique used in signal processing to reduce spectral leakage when performing Fourier Transforms. By applying a window function to a signal, discontinuities at the edges of finite-length signals are minimized, improving frequency domain analysis. Three commonly used windowing functions are the Hann window, Hamming window, and Rectangular window.

Let N be the length of the signal.

Figure 1 shows a comparison of these three window functions.



Figuur 1: Comparison of Rectangular, Hann, and Hamming Window Functions.

3.1 Rectangular Window

The Rectangular window is the simplest windowing function, where the amplitude remains constant over the entire window length:

$$w(n) = 1, \quad 0 \leq n \leq N-1 \tag{1}$$

3.2 Hann Window

The Hann window is a raised cosine window that smooths the signal towards zero at the edges:

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \quad 0 \leq n \leq N-1 \tag{2}$$

3.3 Hamming Window

The Hamming window is similar to the Hann window but has a slightly different weighting to reduce the side lobes:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \tag{3}$$

4 Generating & Comparing Spectrograms

A spectrogram is a visual representation of the spectrum of frequencies in a signal as it varies with time. It is commonly used in audio signal processing to analyze the frequency content of a signal. Spectrograms are generated using the Short-Time Fourier Transform (STFT), which breaks down the signal into short, overlapping time segments and computes the Fourier Transform for each segment. This process allows us to observe how the frequency content of the signal changes over time.

4.1 Generating Spectrograms using STFT

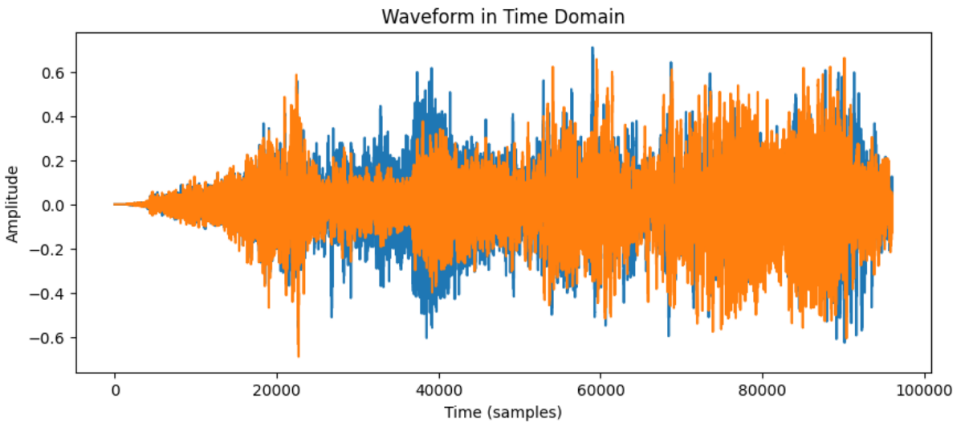
To generate a spectrogram, the following steps are typically performed:

- **Windowing**: The signal is divided into short, overlapping segments, and each segment is multiplied by a window function (e.g., Hamming, Hann, or Rectangular). Windowing reduces spectral leakage by tapering the edges of each segment.

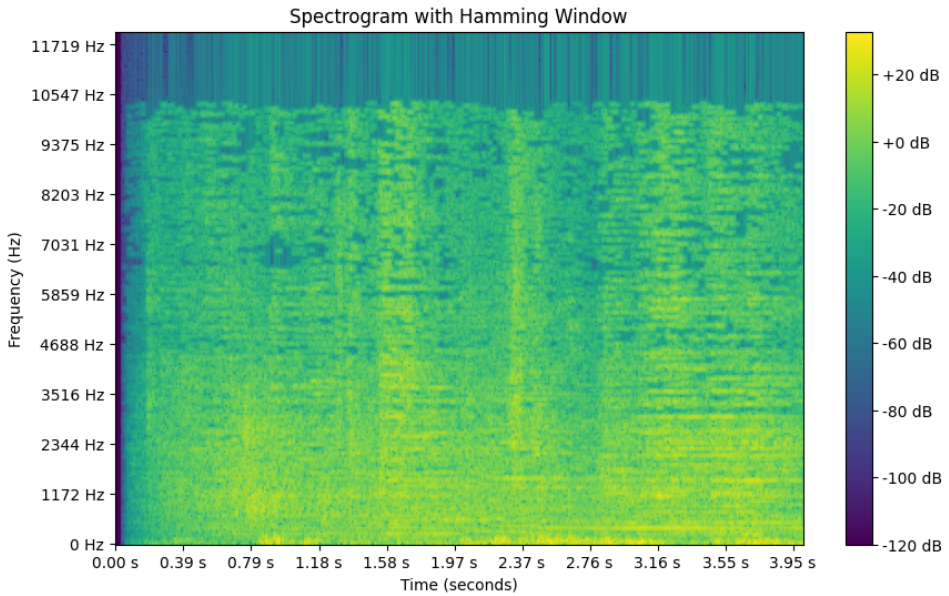
- **STFT Computation**: The Fourier Transform is applied to each windowed segment to obtain the frequency components.
- **Absolute Values**: The magnitude of the complex STFT output is taken to obtain the amplitude of each frequency component.
- **Decibel Conversion**: The amplitude values are converted to decibels (dB) to better visualize the dynamic range of the signal. This is done because the human ear perceives sound logarithmically, and the decibel scale compresses the wide range of amplitudes into a more manageable range.
- **Mono Conversion**: For multichannel audio, the signal is typically converted to mono by averaging the channels before generating the spectrogram. This simplifies the visualization and analysis.

4.2 Sample Signal and Spectrograms

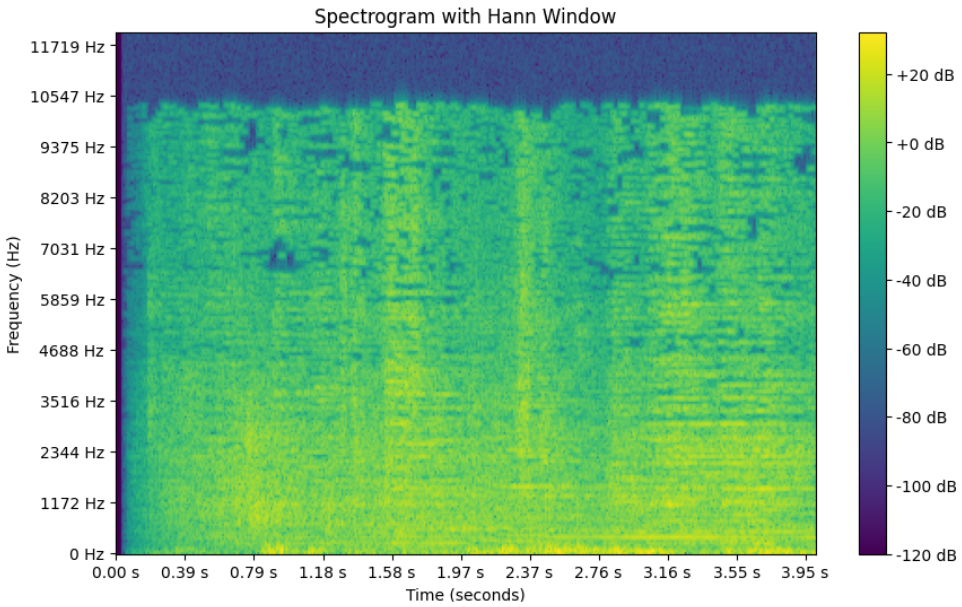
Below is the waveform of a sample signal in the time domain, followed by the resulting spectrograms generated using different windowing techniques for an audio which belong to street music class: Hamming, Hann, and Rectangular windows.



Figuur 2: Waveform in Time Domain.



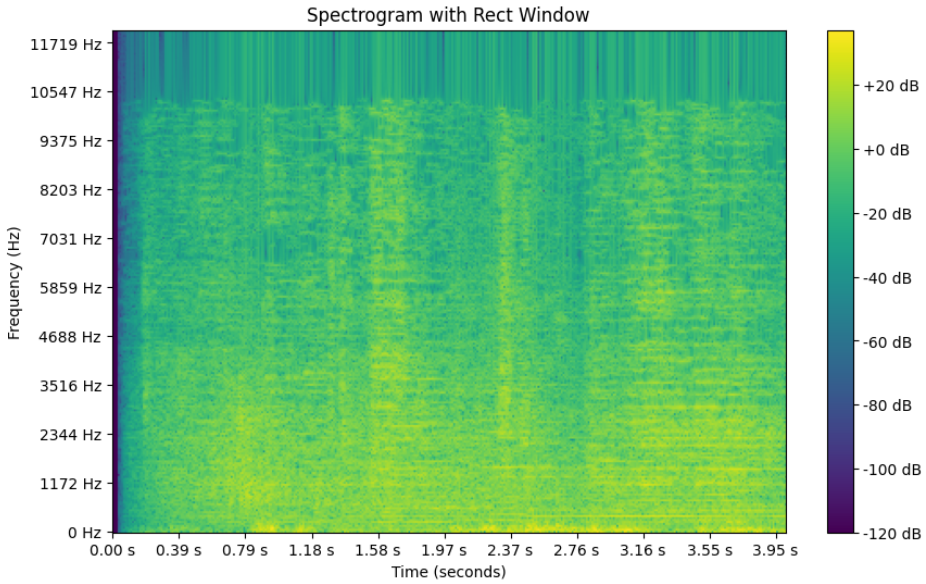
Figuur 3: Spectrogram with Hamming Window.



Figuur 4: Spectrogram with Hann Window.

4.3 Comparison of Spectrograms

The spectrograms generated using different windowing techniques exhibit distinct characteristics:



Figuur 5: Spectrogram with Rectangular Window.

- **Hamming Window**: Provides a good balance between frequency resolution and spectral leakage. The spectrogram shows clear frequency components with minimal leakage. In Figure 3, around 1 second, we can notice an activity at 7031 Hz which is better visible as compared to spectrogram generated from rectangular window.
- **Hann Window**: Similar to the Hamming window but with slightly better frequency resolution. The spectrogram is smoother, with reduced sidelobe levels. In Figure 4, we can see the same activity more sharply as compared to hann window.

- **Rectangular Window**: Offers the best frequency resolution but suffers from significant spectral leakage. The spectrogram shows sharp peaks but with noticeable artifacts due to leakage.

The choice of windowing technique depends on the specific application. For general-purpose audio analysis, the Hamming or Hann window is often preferred due to their balanced performance.

The spectrograms were generated using Python and PyTorch. The attached Jupyter Notebook (`script.ipynb`) contains the implementation details and code used to produce these visualizations.

5 Classification

In this task, a classifier was implemented using a simple Convolutional Neural Network (CNN) architecture. The model consists of:

- **2 Convolutional Layers**: These layers extract spatial features from the input spectrograms.
- **2 Fully Connected Layers**: These layers perform the final classification based on the extracted features.

The dataset was divided into training and testing samples, with 7895 samples used for training and 837 samples used for testing. The model was trained for 5 epochs, and a 50% overlap was applied during the windowing process to generate the spectrograms. The model was trained using three different windowing techniques: Hann, Hamming, and Rectangular. The performance of the model was evaluated using accuracy and loss metrics, and confusion matrices were plotted for each windowing technique.

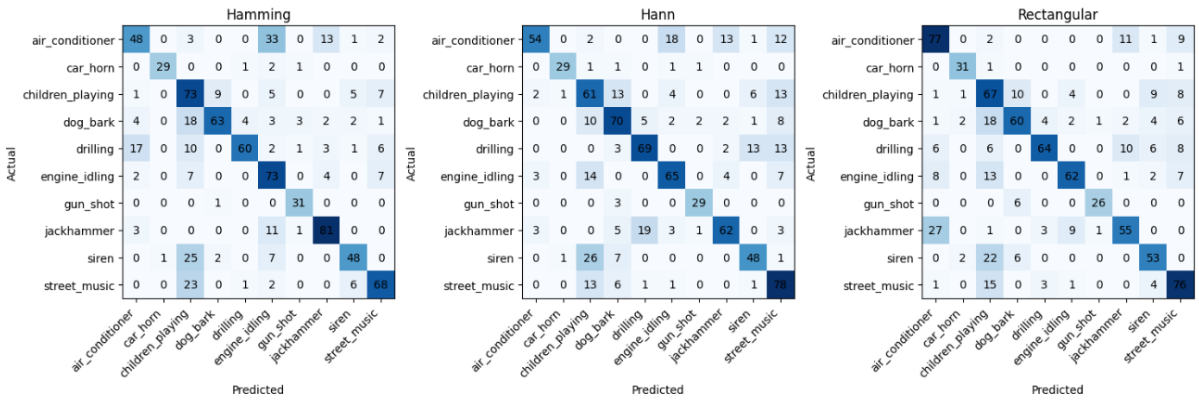
The final results for each windowing technique are summarized in the Table 1.

Window	Epoch	Train Acc	Test Acc	Test Loss
Hann	5	0.9121	0.6750	1.5118
Hamming	5	0.8965	0.6858	1.5333
Rectangular	5	0.8595	0.6822	1.5251

Tabel 1: Performance metrics for the classifier using different windowing techniques.

6 Analysis

In this section, confusion matrices were plotted for models trained using all three windowing techniques. Figure 6 shows the confusion matrices for the models using Rectangular, Hann, and Hamming windowing. By analyzing these confusion matrices, we can identify the classes that were misclassified, which provides valuable insights into the model's performance.



Figuur 6: Confusion Matrices for Models with Rectangular, Hann, and Hamming Windowing.

For example, with the Hamming window used in classification, we observe that the number of misclassifications of `jackhammer` as `drilling` is notable. `Jackhammer` sounds are characterized by impulsive,

broadband noise, whereas `drilling` sounds have more periodic, tonal components. Since the Hamming window allows more spectral leakage, it tends to blur the spectral details, introducing unwanted frequency components. This makes `jackhammer` sounds appear more similar to `drilling` sounds, leading to more frequent misclassifications of `jackhammer` as `drilling`. Therefore, the Hamming window is more prone to this type of error.

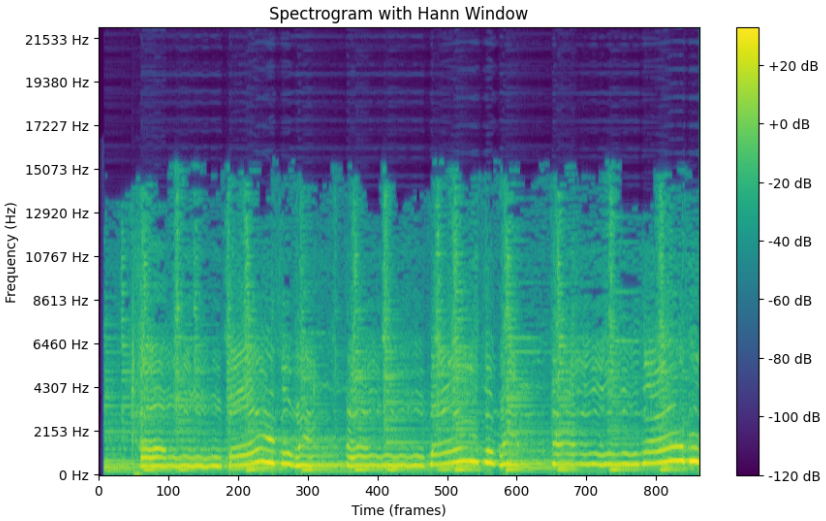
In the case of the Rectangular window, a significant number of `jackhammer` sounds are misclassified as `children playing`. This is likely due to the lack of smoothing at the edges of the Rectangular window, which does not handle the frequency content as well as the Hann or Hamming windows. On the other hand, the Hann and Hamming windows provide better spectral resolution and less leakage, which results in better classification of `jackhammer` sounds.

The Rectangular window model classifies `air conditioner` class it mostly correctly, while both Hann and Hamming windowing models tend to confuse it with `engine idling` more frequently. This could be due to the fact that both Hann and Hamming windows better capture subtle details in the frequency domain, which might overlap more between the `air conditioner` and `engine idling` classes, while the Rectangular window, with its sharper cutoff, is less sensitive to such fine details and classifies it more accurately.

7 Task B

For Task B, four audio tracks were taken for analysis:
1) `Tip Toe Through the Tulips (Hig pitched)`
2) `Pink Panther Theme Song (Instrumental)`
3) `Ra.One Theme Song (Electronic)`
4) `Tom and Jerry Theme (Jazz)`

Since these audio tracks exhibit similar behavior throughout, only the first 5 seconds of each track were considered for comparison and analysis. The respective spectrograms for these 5-second segments are plotted. (Note: The duration of analysis is a hyperparameter and can be adjusted as needed.) The audio files can be found in the attached folder.
The spectrograms of these songs show differences in how the sound energy is spread across frequencies and time. Each song has patterns that reflect its musical style and the instruments used. By looking at these spectrograms, we can analyse sounds with different types of genre.

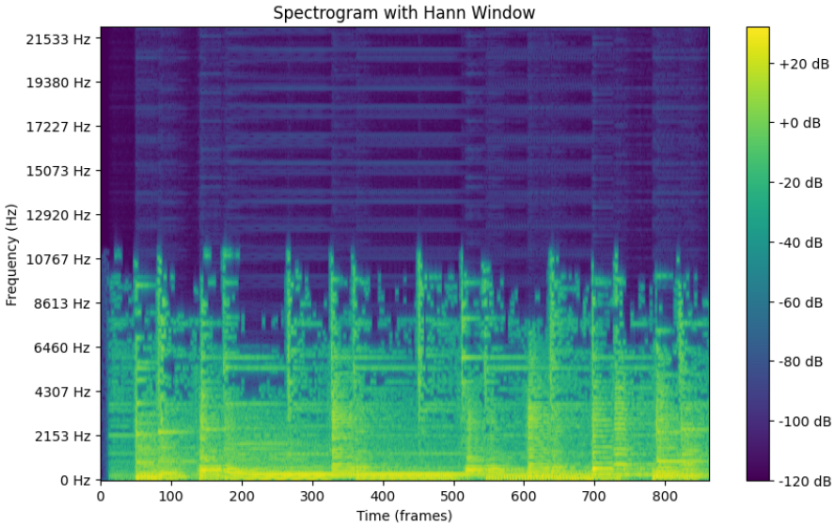


Figuur 7: Spectrogram for `Tip Toe Through the Tulips`

In `Tip Toe Through the Tulips`, the spectrogram has smooth, beight yellow lines at lower frequencies, and as we move higher, the color fades, showing that the energy decreases. This pattern suggests

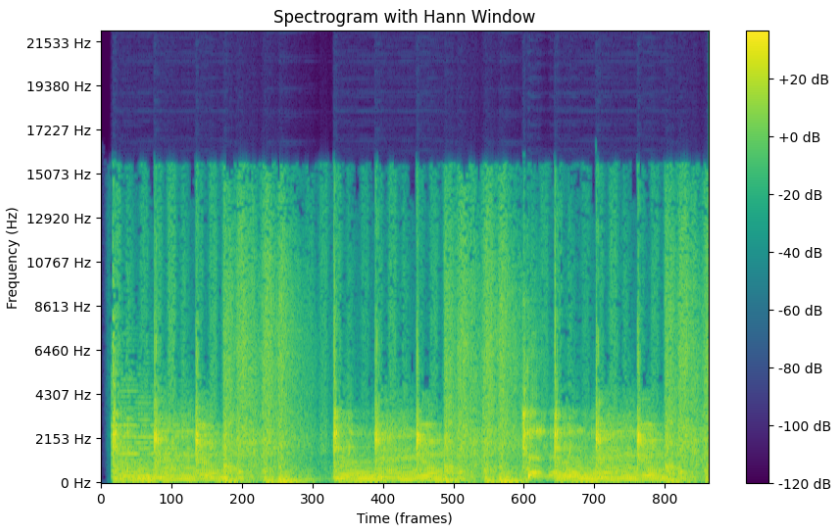
that the song has a strong and steady tone, mostly concentrated in the lower frequency range. The voice in this song is high-pitched, but its energy does not spread too much across different frequencies. The smoothness of the spectrogram means that the song has continuous sounds rather than sudden, sharp changes.

The `Pink Panther Theme` also has high energy at lower frequencies, but unlike `Tip Toe Through the Tulips`, the pattern is not smooth. Instead, we see vertical lines appearing at different time points. These vertical lines indicate short and sudden sounds, likely from instruments like saxophone. Since these sounds appear at different times and cover a wide range of frequencies, they create a more unpredictable spectrogram, showing that the music has many quick changes in pitch and intensity.



Figuur 8: Spectrogram for `Pink Panther Theme Song`

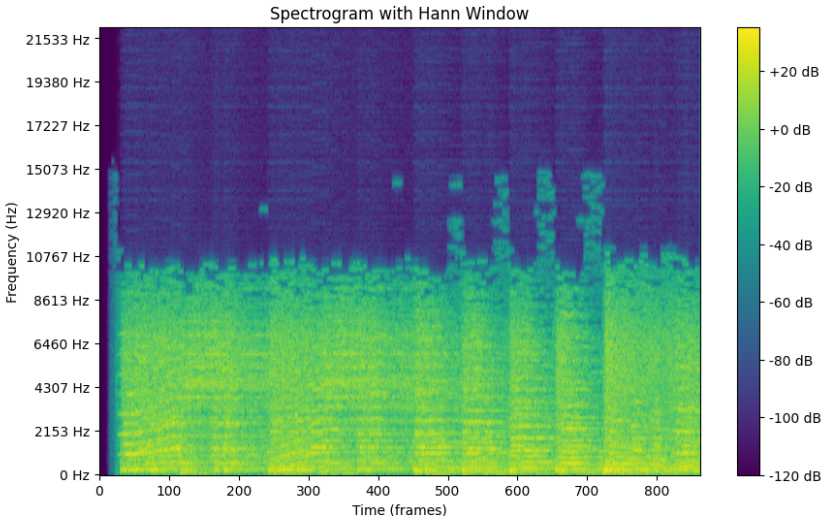
The `Ra.One Theme` covers a wide range of frequencies in its spectrogram. The sound energy is spread across different frequencies, meaning that both low and high frequencies are active at the same time. This happens because the song uses many different instruments, including deep bass and high-pitched electronic sounds. The energy is not concentrated in just one area but is spread out, making the spectrogram appear more filled and intense compared to the other songs.



Figuur 9: Spectrogram for `Ra.One Theme Song`

The `Tom and Jerry Theme` shows strong energy from 0Hz to 10,767Hz, similar to the `Pink Panther Theme`. However, it also has horizontal lines that extend across many frequencies. These horizontal lines

mean that some musical notes are held for a longer time instead of changing quickly. This suggests that the song has a mix of steady tones and sudden changes. Additionally, some energy is present even beyond 10,767Hz, showing that high-pitched sounds also play a role in the music.



Figuur 10: Spectrogram for `Tom and Jerry Theme`

These differences show how various types of sounds create different spectrogram patterns. The dataset can be downloaded from the link provided in the objective

8 References

PyTorch Documentation. Available: `https://pytorch.org/docs/stable/index.html`

J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research,ïn *22nd ACM International Conference on Multimedia*, Orlando, USA, 2014.

Speech Processing Book. Available: `https://speechprocessingbook.aalto.fi/`