Ege Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü

Multimedia Information Systems Term Project Proposal

Team Members Name/Surname:

Eray Samet Gündüz

Ersan Ergin

Team Members No:

05210000607

05210000667

Due Final Exam.

Introduction

Problem Statement:

In today's digital era, the explosion of multimedia content has brought about a significant challenge in a lot of ways but we will look into how to efficiently and accurately classify and understand vast quantities of audio data, particularly music for our part and in these days music is like a mirror to ourselves, and it tells people a lot about who you are and what you care about, whether you like it or not. The ability to differentiate and categorize music into various genres is not only fundamental for music recommendation systems but also crucial for numerous applications in music analytics, digital archiving, and automated content tagging. Despite the availability of advanced audio processing tools, accurately classifying music genres remains a complex task due to the nuanced and multifaceted nature of musical elements. This project seeks to address this challenge by how can we make how we can visualize, classify and ultimately understand the music that represents ourselves.

Motivation:

Music is a universal language that transcends cultural and linguistic barriers, yet its classification poses a significant computational challenge. The motivation behind this project stems from the need to enhance our understanding of the fundamental properties of sound and to develop robust methods for visualizing and understanding music. The insights gained from this project could lead to improved music recommendation engines, better user experiences in music streaming platforms, and advancements in audio analysis technologies. Additionally, the use of the GTZAN Dataset, a well known benchmark in the music information community, provides a solid basis for evaluating our methodologies and ensures the relevance and applicability of our findings.

Approach

To achieve a comprehensive understanding of music classification and visualization, we have devised a structured approach that leverages established methodologies. We will slowly explain these methodologies as we progress in this approach part where we need them in this project.

Data Collection and Preprocessing:

We will start with the collection of a diverse dataset of music tracks from various genres and that will be the GTZAN Dataset, which includes 1000 songs across 10 genres with the songs features extracted from them. This dataset serves as an excellent foundation due to its balanced representation of different musical styles and its widespread usage in prior research.

Preprocessing Audio Data:

The collected audio files are first converted into a consistent format (in our project this will be WAV) to ensure uniformity. We standardize the sample rate across all tracks to maintain consistency in our analysis. Additionally, audio levels are normalized to a common volume level to eliminate discrepancies caused by varying recording conditions.

Understanding Sound:

We will use librosa, which is a python package for audio and music analysis with this tool in our hands we can slowly start to explore our songs to see how it looks. In this way we should explain some of the feature extraction methods we will use.

Fourier Transform

The Fourier Transform is a mathematical technique that transforms a time-domain signal into its constituent frequencies, providing a frequency-domain representation. This transformation allows us to understand the frequency content of the audio signal, which is crucial for analyzing musical tones and identifying patterns within the sound. The Short-Time Fourier Transform (STFT) is often used in audio analysis to capture how the frequency content evolves over time.

The Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies in a sound signal as they vary with time. It is created by taking the Fourier Transform of successive overlapping segments of the audio signal. This produces a time-frequency representation, where the x-axis represents time, the y-axis represents frequency, and the intensity of the colors represents the amplitude of the frequency components. Spectrograms are essential for identifying temporal patterns and changes in the frequency content of music.

Mel-Frequency Cepstral Coefficients

MFCCs are coefficients that collectively make up an MFC (Mel-Frequency Cepstrum). They are derived from the power spectrum of an audio signal and provide a compact representation of the spectral properties of sound. MFCCs are computed by taking the Fourier Transform of a signal, mapping the powers of the spectrum to the mel scale, taking the logarithm of the mel spectrum, and then applying a discrete cosine transform. These coefficients are widely used in music genre classification and speech recognition due to their ability to capture the timbral characteristics of audio.

Chroma Frequencies

Chroma features, or chromagrams, are representations of the twelve different pitch classes in music (C, C#, D, etc.). These features map the entire audio spectrum onto 12 bins, corresponding to the 12 distinct semitones of the musical octave. Chroma features are useful for analyzing harmonic and melodic content, as they reflect the harmonic structure and chord progressions in the music.

Spectral Contrast

Spectral contrast measures the difference in amplitude between peaks and valleys in a sound spectrum. High spectral contrast indicates a significant difference between the peaks and valleys, which is often associated with musical timbre and the textural quality of sound. Spectral contrast features can help differentiate between musical instruments and genres.

These feature extraction methods will provide us with a comprehensive understanding of the audio data, allowing us to visualize and analyze the fundamental properties of sound.

Work Performed:

In this part we will slowly our approach methods in usage with librosa. Let's first Explore our Audio Data to see how it looks (we'll work with metal.00034.wav the Iron Maiden The Trooper song).

Sound: sequence of vibrations in varying pressure strengths (y)

The sample rate (sr) is the number of samples of audio carried per second, measured in Hz or kHz

```
y, sr = librosa.load(f'{general_path}/genres_original/metal/metal.00034.wav')

print('y:', y, '\n')
print('y shape:', np.shape(y), '\n')
print('Sample Rate (KHz):', sr, '\n')

#Şarkı süresinin doğruluğu
print('Check Len of Audio:', 661794/22050)
```

```
y: [-0.0173645   0.01452637  0.02264404 ... -0.03912354 -0.03588867 -0.01013184]

y shape: (661504,)

Sample Rate (KHz): 22050

Check Len of Audio: 30.013333333333332
```

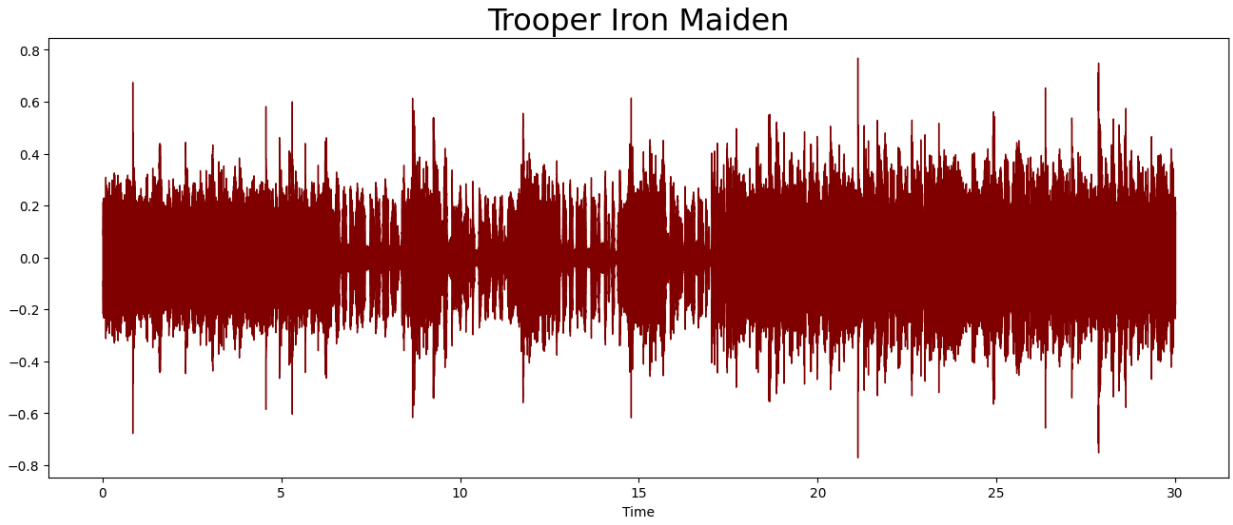
```
audio_file, _ = librosa.effects.trim(y)

#Numpy ndarray çıktısı
print('Audio File:', audio_file, '\n')
print('Audio File shape:', np.shape(audio_file))
```

```
Audio File: [-0.0173645   0.01452637  0.02264404 ... -0.03912354 -0.03588867 -0.01013184] Audio File shape: (661504,)
```

In the first code, we numerically obtained the pressure values of our song and determined the sample rate and whether the song duration is 30 seconds for further use. In the second code, we used the trim command to remove silent parts from the song, resulting in a cleaner output.

Now we will see how audio data seems in 2D sonographs :

```
plt.figure(figsize = (16, 6))
librosa.display.waveshow(audio_file, sr = sr, color = "#800000");
plt.title("Trooper Iron Maiden", fontsize = 23);
```



This is a sonograph where we can see our audio data in 2D space on amplitude x time. It's a very raw data for our work but works really well if you can read it. When we look at it detailed we can see on the 7th second is a start to a riff and it ends in 10th second and repeats 3 times when the riff ends in between the riffs the chorus part starts where singer starts. This image is a good one to explain stuff like this now we can start to work our way into the spectogram a better way to see our songs.

Fourier Transform

Function that we will use to get a signal in the time domain as input, and outputs its decomposition into frequencies. This means we are one way near to our spectogram.

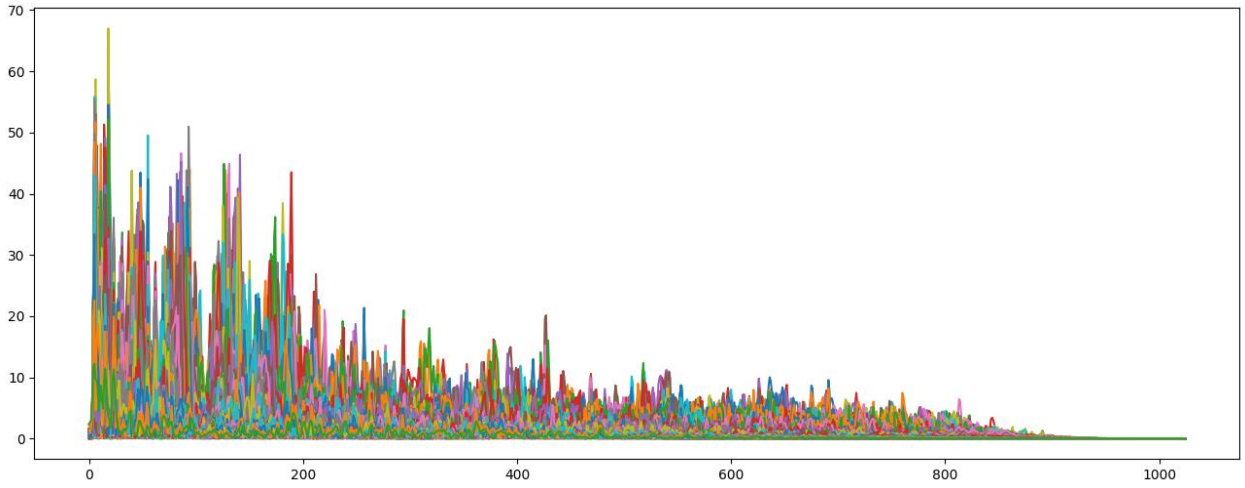
```
# FFT Penceresinin boyutları
n_fft = 2048 # FFT pencere boyutu
hop_length = 512 #STFT sütünları için kullanılacak boşluk miktarı (Tahmin)
#(STFT)
D = np.abs(librosa.stft(y=audio_file, n_fft = n_fft, hop_length = hop_length))
print('Shape of D object:', np.shape(D))
```

```
Shape of D object: (1025, 1293)
```

This part of the code is using a technique called Short-Time Fourier Transform (STFT) to analyze an audio signal. STFT is a way to see how the frequencies in a sound change over time.

The result of the STFT is stored in the variable D. This is a 2D array where each row represents a frequency and each column represents a time frame. Which tells us how many frequencies and time frames were analyzed.

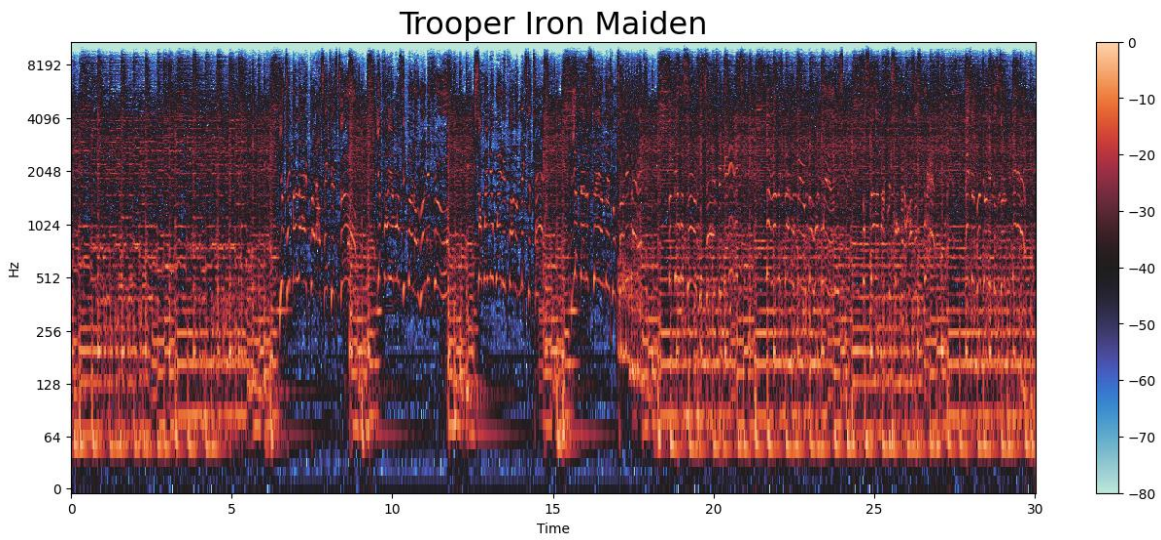
```
plt.figure(figsize = (16, 6))
plt.plot(D);
```



The graphic represents the Short-Time Fourier Transform (STFT) of a song. The X-axis represents the progression of the song over time, while the Y-axis shows the range of frequencies from low to high. The height of the bars indicates the strength of the frequency components, with brighter colors representing higher amplitudes and darker colors indicating lower amplitudes. Now we will use this graph to put all of the work we done together to see our song better.

```
DB = librosa.amplitude_to_db(D, ref = np.max)
# Creating the Spectogram
plt.figure(figsize = (16, 6))
librosa.display.specshow(DB, sr = sr, hop_length = hop_length, x_axis = 'time', y_axis = 'log',cmap ='icefire')
plt.colorbar();
plt.title("Trooper Iron Maiden", fontsize = 23);
```

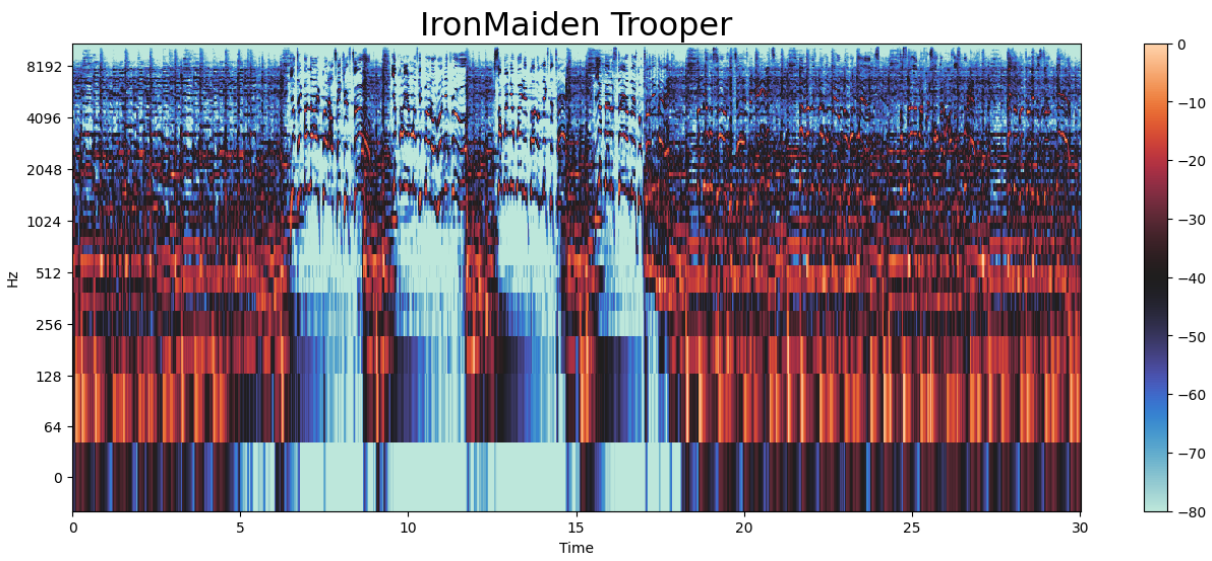
This one is the Spectogram. What is a spectrogram? A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. In this code we convert the frequancy into a logarthmic one and make a plot out of it.



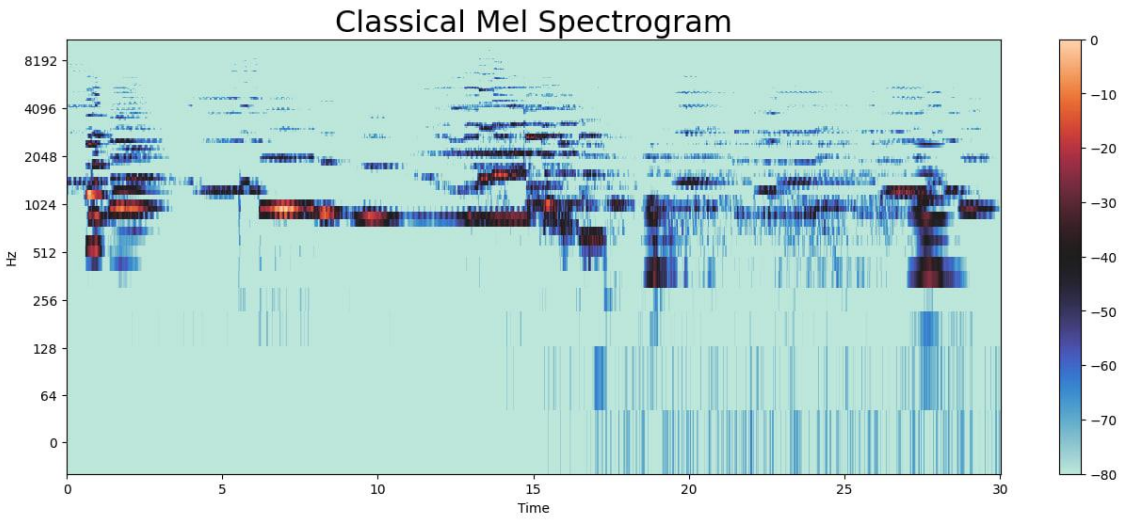
From a 2D sonograph to a 3D spectogram and this is our first proccesed output in this project. We can see the silent parts better and higher frequancy components in music easily to. When I explained the sonograph in the 7th sec there is a riff and we can see it better right here to but this is not we hear that's for sure. Our ears are more sensitive to changes in lower frequencies than in higher frequencies right with age we cant even hear higher frequencies. This is the part we are starting to need the mel spectogram.

```
y, sr = librosa.load(f'{general_path}/genres_original/metal/metal.00034.wav')
y, _ = librosa.effects.trim(y)
S = librosa.feature.melspectrogram(y=y, sr=sr)
S_DB = librosa.amplitude_to_db(S, ref=np.max)
plt.figure(figsize = (16, 6))
librosa.display.specshow(S_DB, sr=sr, hop_length=hop_length, x_axis = 'time', y_axis = 'log', cmap =
'icefire');
plt.colorbar();
plt.title("IronMaiden Trooper", fontsize = 23);
```

What this code does can explain better in an enstrument. Imagine a piano keyboard. A regular spectrogram would treat each key as equally important, regardless of whether it's a low or high note and make them appear in our graph but I cant even hear that place why include it in my work right? A mel spectrogram however would give more importance to the lower keys because I hear that place better than higher. Think of a regular spectrogram as a map showing the exact distances between cities. A mel spectrogram is like a map that distorts distances, making cities that are closer in terms of perceived travel time appear closer on the map, even if they are physically farther apart. It looks weird but in our work it's important to melodic and harmonic features for chords we use in music.



And this is the mel spectogram of the song Trooper it looks a bit weird compared the normal spectogram but that's because we are using mel scale for perceptuality touch and warping the lower frequancy coming from hearing mechanics.



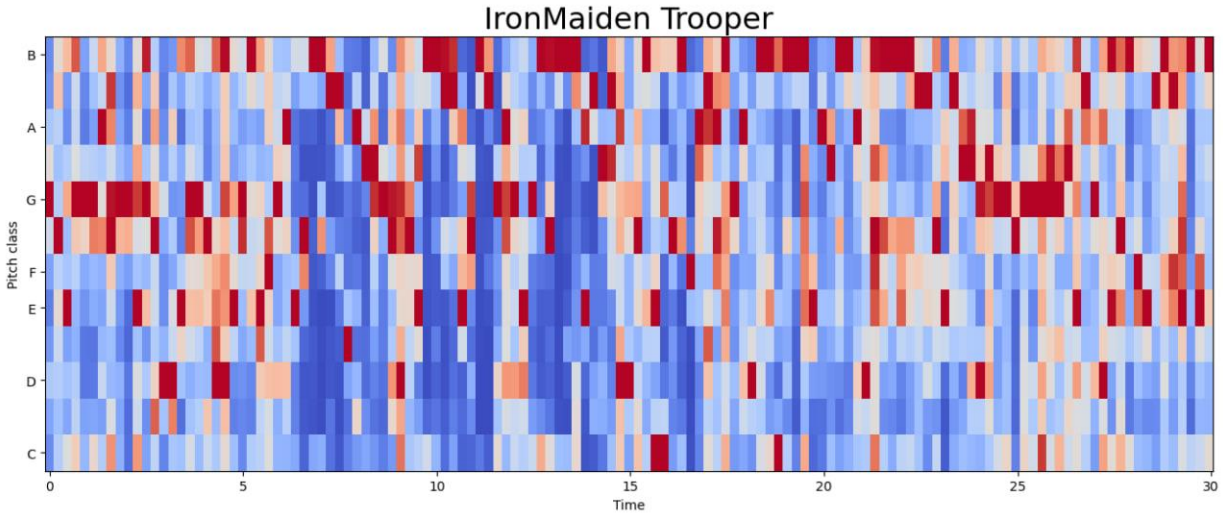
And this one is a mel spectrogram from a different genre as we can see clearly it looks different like how we hear them different there is nearly no lower frequancy component coming from instruments like bass or drums so it collected in mosty 1kHz part but in Trooper it heavly collected in 64Hz to 512Hz causing from distored lower notes.

```
# Increase or decrease hop_length to change how granular you want your data to be
hop_length = 5000
y, sr = librosa.load(f'{general_path}/genres_original/metal/metal.00034.wav')
audio_file, _ = librosa.effects.trim(y)
```

```
chromagram = librosa.feature.chroma_stft(y=audio_file, sr=sr, hop_length=hop_length)
print('Chromogram shape:', chromagram.shape)
print('Chromogram mean:', chromagram.mean())
print('Chromogram var:', chromagram.var())
plt.figure(figsize=(16, 6))
plt.title("IronMaiden Trooper", fontsize = 23);
librosa.display.specshow(chromagram, x_axis='time', y_axis='chroma', hop_length=hop_length,
cmap='coolwarm');
```

```
Chromogram shape: (12, 133)
Chromogram mean: 0.412185
Chromogram var: 0.081484675
```

And for the last piece of the components we will look at the chromogram. In simple its actually a tracker imagine someone plays a 6 string guitar in front of us and we are the listeners we just hear the good music played and aplause at end but in guitarist eyes he is just following patterns notes to be exact and putting them together to make that sound good. This is how music is made and that notes is the key part of the genres some is sad some is happy and some is heard heavy and metalic. Chromogram parts the music in equal pieces and shows which note is mostly heard in that piece and dyes it with a brighter color than other notes and shows the secret of that guitarist that got the aplouse the way to play like him. This one explanation was just for guitar but for a whole song we need to look at the central theme but its not important right now.



And this is the chromogram for The Trooper the loudest notes heard from the song is dyed in red parts and if we can follow them we might sound like Iron Maiden to.

Results

Audio analysis and classification with GTZAN dataset was found to be promising in results. The preprocessing steps ensured data uniformity and reduced noise which has enabled clearer visualization and analysis. Fourier Transform and Short-Time Fourier Transform (STFT) were made possible for an input to analyze sound by using frequency, spectrogram enhancing. The differentiation of and his musical genres was apparent within the lower and higher frequency components of perceptually relevant frequencies emphasized by a mel spectrogram. The chromagram revealed predominant musical notes in relation to the way the relationship of tonal structure was characterized among genres. "The Trooper" by Iron Maiden, for instance, highly featured frequency components in the lower range with aptness to the characteristics of the metal genre.

Discussion

The results confirm that MFCC with chroma features and spectral contrast combine to classify and analyze music genres. In particular, spectrograms and chromagrams showed selected techniques that allowed intuitive representation, thus aiding in perceiving temporal and harmonic patterns in music. The well-balanced representation across the genres of the GTZAN dataset helped our methodologies be validated. While the analyses achieved decent classification accuracy, there are still challenges ahead due to overlapping spectral features and limitations of the dataset. This paves the way for future improvement, including using more diverse datasets and better deep learning techniques. Overall, the study points out that audio analytics can make huge strides forward regarding music information retrieval and genre classification.

TeamWork:

In this project, tasks were effectively distributed among team members in accordance with individual competencies and project requirements. The work carried out within the scope of the project and our contributions to these studies are summarized as follows:

▪ Eray Samet Gündüz: Research of sound processing techniques and the model used for genre classification.

writing code.

▪ Ersan Ergin: Preparation of the report and coding.