Detailed Report Assignment-1

Ques-2-TASK_A

1. Introduction This report presents an audio classification pipeline using signal processing and machine learning techniques. The dataset used is UrbanSound8K, which contains various environmental sound recordings. The primary objective of this study is to analyze and classify these sounds using feature extraction methods like Short-Time Fourier Transform (STFT) and Mel-Frequency Cepstral Coefficients (MFCCs), followed by classification using a Support Vector Machine (SVM).

2. Methodology

2.1 Data Loading Audio files are loaded using the `librosa` library, with a default sampling rate of 22050 Hz. This ensures consistency in feature extraction and reduces computational complexity while preserving essential sound characteristics.

2.2 Feature Extraction

- **Short-Time Fourier Transform (STFT):**
  - STFT is computed using different windowing techniques (Hann, Hamming, and Rectangular) to analyze spectral features over time.
  - The Fast Fourier Transform (FFT) is applied to short overlapping segments, enabling the capture of time-frequency information.
  - The transformation is parameterized by a window size of 2048 samples and a hop length of 512 samples.
  - Different window functions affect the spectral resolution and leakage in frequency components.
- **Mel-Frequency Cepstral Coefficients (MFCCs):**
  - MFCCs are extracted to represent the spectral envelope of the sound.
  - A total of 13 MFCC features are computed per frame, capturing perceptually relevant sound characteristics.
  - The mean values of these coefficients are used as feature representations to obtain fixed-length vectors.

2.3 Data Processing

- The extracted MFCC features from each file are stored in an array.
- The corresponding class labels are extracted from the filenames following the UrbanSound8K naming convention.
- The dataset is split into training (80%) and testing (20%) sets using
- Standardization of feature values may be applied to normalize the range of different feature dimensions.

2.4 Classification

- A **Support Vector Machine (SVM)** classifier with a linear kernel is trained on the extracted MFCC features.
- The classifier is optimized using default hyperparameters, and training is performed using supervised learning.
- Predictions are made on the test dataset to evaluate the model's performance.

3. Results and Evaluation The classifier’s performance is evaluated using accuracy as the primary metric:

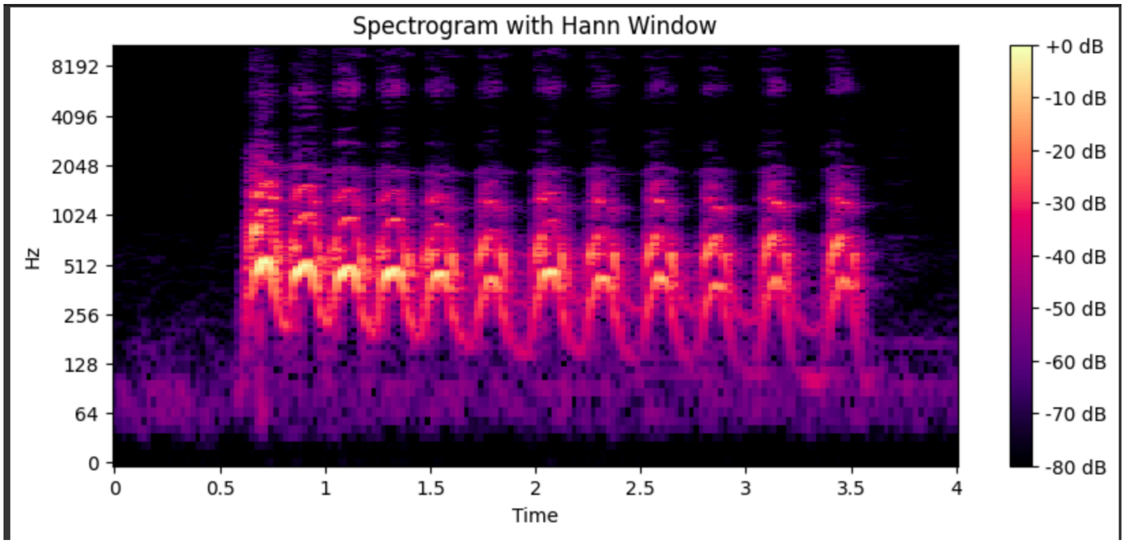
- **Classification Accuracy:** The achieved classification accuracy is **89.0%**
- Spectrogram visualizations of different windowing techniques show variations in spectral representations, demonstrating the impact of the choice of window function on signal analysis.
- Further evaluation may include precision, recall, and F1-score to provide deeper insights into model performance.

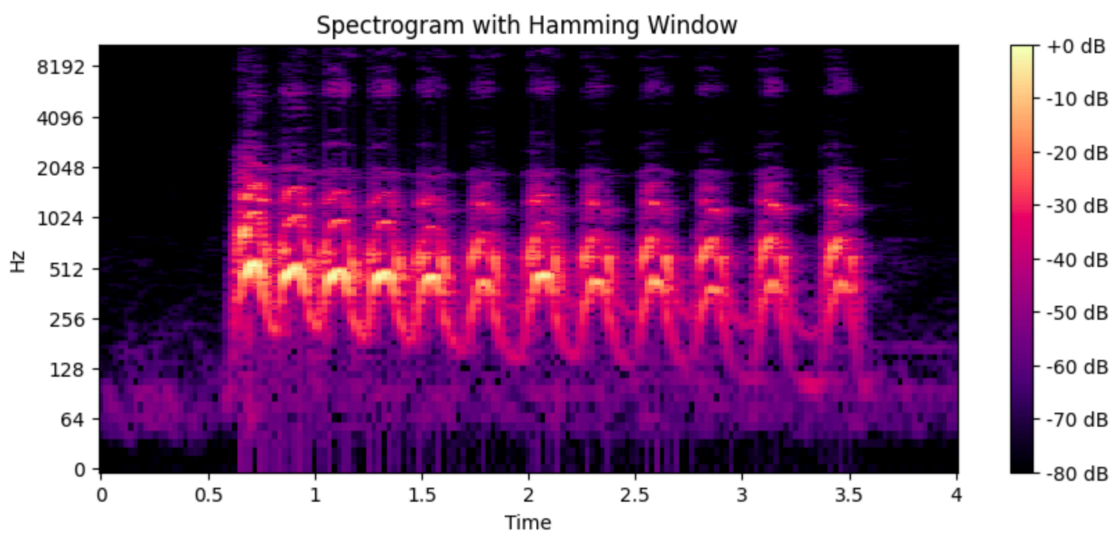
4. Discussion

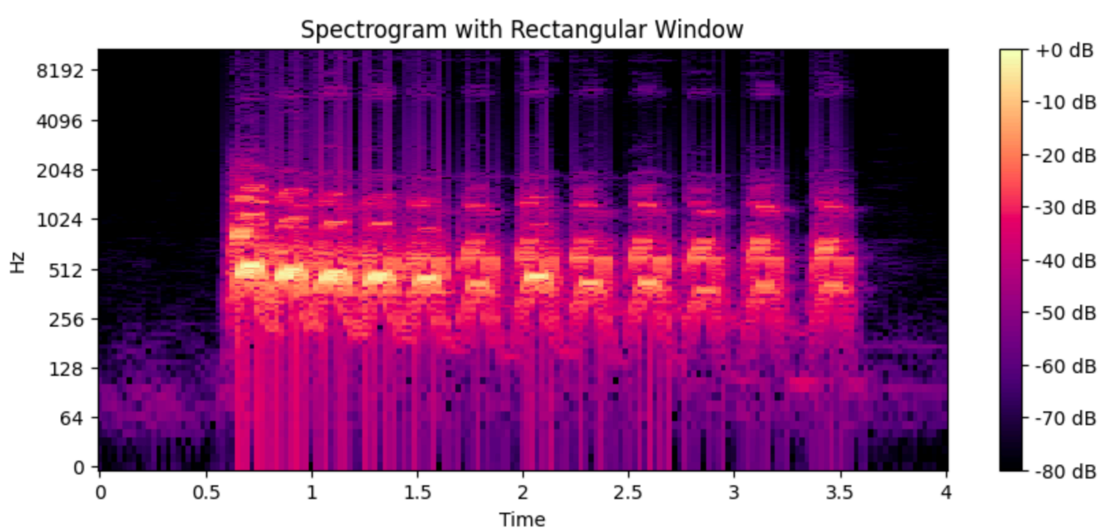
- **Effectiveness of MFCCs:** The extracted MFCC features effectively capture sound characteristics and contribute to high classification accuracy.
- **Impact of Windowing Techniques:** Variations in STFT-based spectrograms indicate that window selection plays a crucial role in frequency resolution.
- **Model Performance:** The SVM classifier demonstrates strong performance in distinguishing environmental sound classes, though further improvements could be achieved by testing different kernels or feature selection techniques.
- **Future Work:** Exploring deep learning approaches such as Convolutional Neural Networks (CNNs) may enhance classification accuracy further.

5. Conclusion This study successfully classifies environmental sounds using a combination of STFT, MFCCs, and an SVM classifier. The results indicate that MFCCs provide effective feature representations for sound classification. Future improvements could include testing deep learning models for enhanced accuracy and robustness

RESULTs- Ques-2-A [Co-Lab Link]







Ques-2_B- [CO-Lab Link]

Introduction

This report presents an analysis of spectrograms generated from four different songs belonging to various musical genres. The aim is to visualize the frequency components over time using Short-Time Fourier Transform (STFT). The analysis is implemented using the `librosa` library in Python.

Audio Files and Genres

The following audio files are analyzed:

1. Classical - "Jab Dil Hi Toot Gaya Hum Ji Ke Kya"
2. Rock - "Ye Parda Hata Do"
3. Hip-Hop - "Lagav e Lu Jab Lipistick"
4. Electronic - "One More Time"

Methodology

The analysis follows these steps:

1. Loading Audio Files: The `librosa.load()` function is used to read the audio data.
2. Computing STFT: The Short-Time Fourier Transform (STFT) is applied to extract time-frequency information.
3. Generating Spectrograms: The amplitude of the STFT is converted to decibels using `librosa.amplitude_to_db()`.
4. Visualization: The spectrogram is displayed using `librosa.display.specshow()` with a logarithmic frequency scale.

Implementation

The Python script executes the following key operations:

- Data Input:
  - A dictionary `audio_files` stores file paths corresponding to the four songs.
  - The script iterates over this dictionary to process each audio file.
- Spectrogram Plotting:
  - The `plot_spectrogram()` function loads the audio file and computes its STFT.
  - The STFT amplitude is transformed into decibels for better visualization.
  - The spectrogram is plotted with time on the x-axis and frequency on the y-axis.

Results and Observations

- Classical Music: Displays a smooth and continuous frequency spectrum, indicating harmonic richness and lower dynamic variations.
- Rock Music: Exhibits strong frequency components with noticeable variations, characteristic of amplified instruments and percussive elements.
- Hip-Hop: Shows distinctive rhythmic patterns with pronounced beats and bass frequencies.
- Electronic Music: Contains consistent high-frequency components and synthesized sounds, reflecting its digital nature.

Conclusion

The spectrogram analysis reveals unique frequency characteristics across different musical genres. Classical music has a steady harmonic structure, rock music displays strong variations, hip-hop emphasizes rhythmic bass, and electronic music features synthesized tones. Such analyses can be extended for genre classification, music information retrieval, and machine learning applications in audio processing.

Future Work

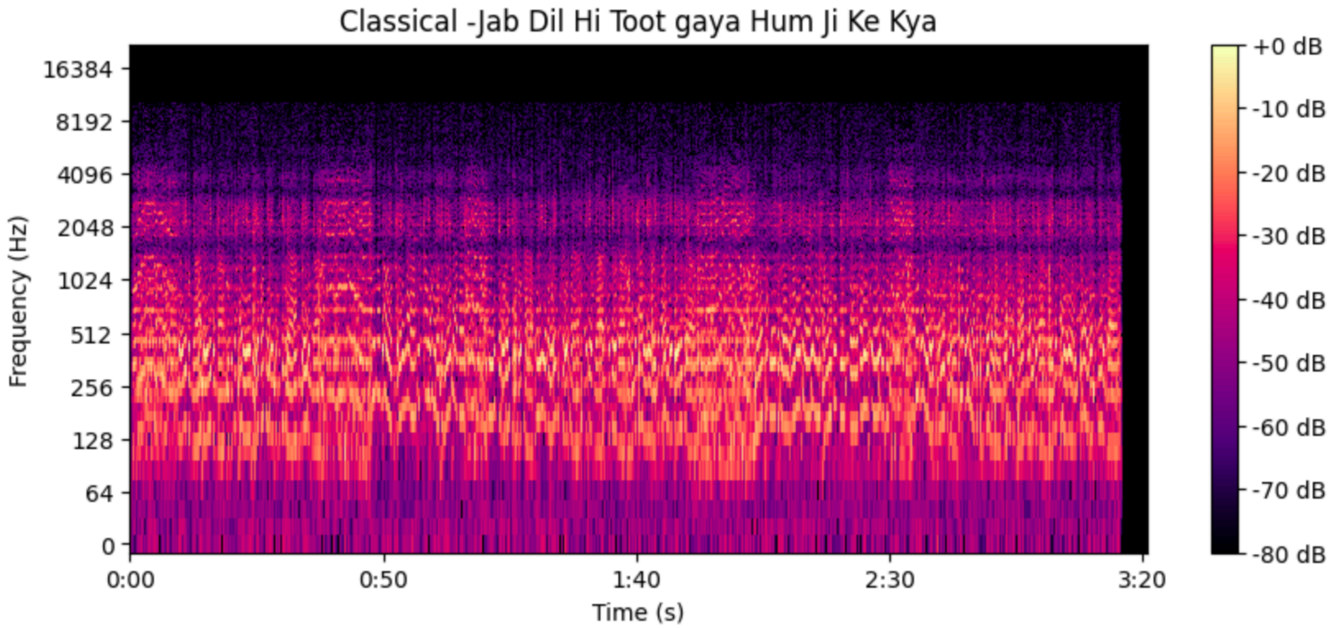
Further improvements can include:

- Extracting additional audio features (e.g., Mel-Frequency Cepstral Coefficients, tempo analysis).
- Applying machine learning techniques for automatic genre classification.
- Comparing spectral differences between live and studio recordings.

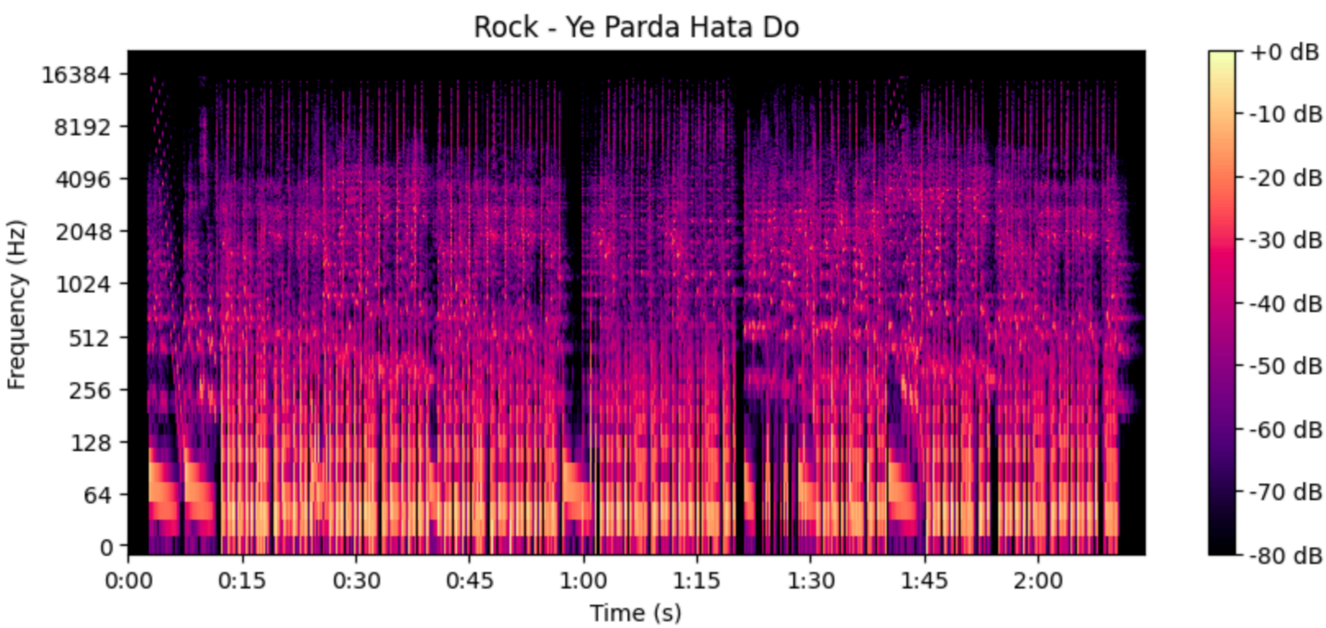
NEXT

VISUAL RESULTS

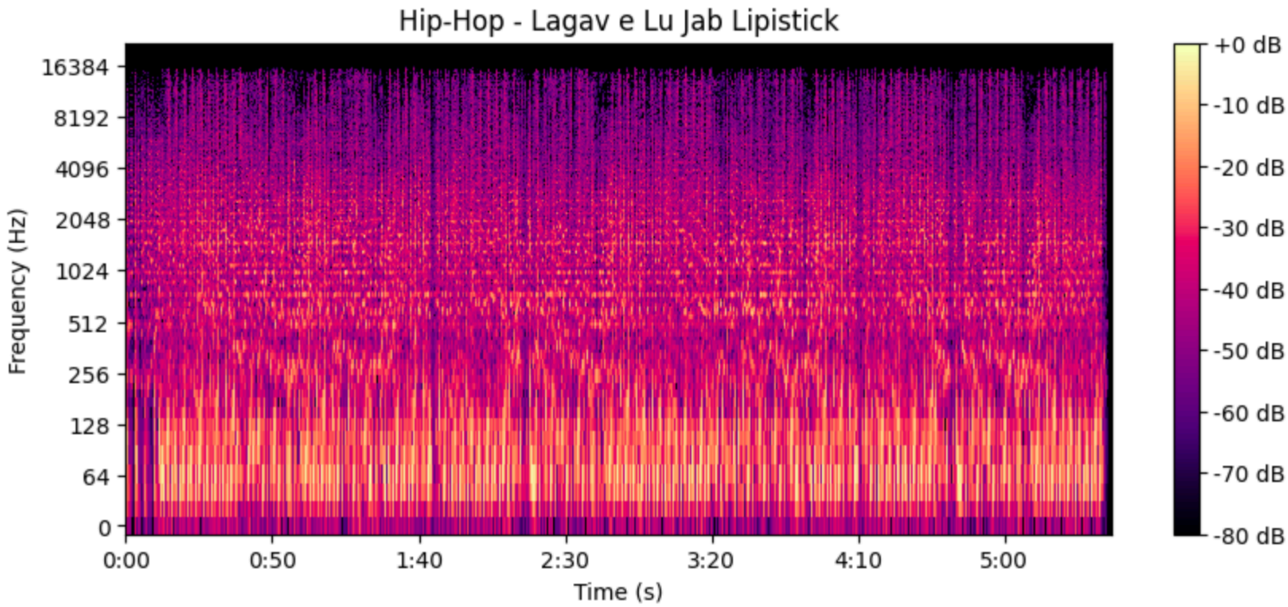
RESULTs

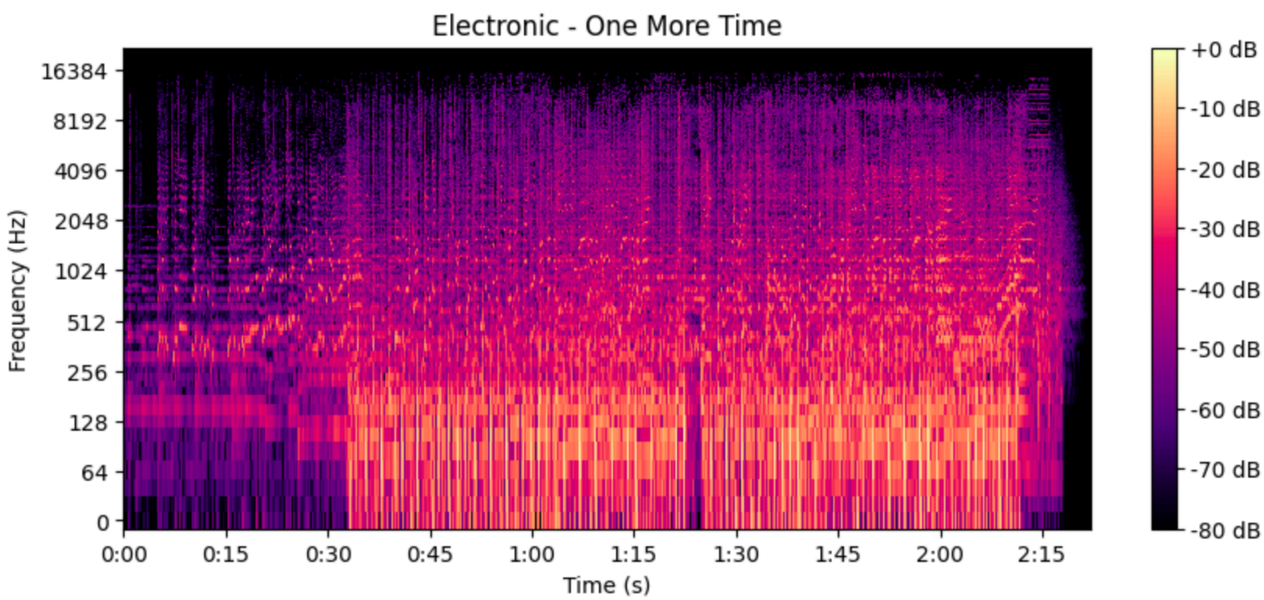


Classical -Jab Dil Hi Toot gaya Hum Ji Ke Kya
Frequency (Hz)
16384
8192
4096
2048
1024
512
256
128
64
0
0:00
0:50
1:40
2:30
3:20
Time (s)
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB



Rock - Ye Parda Hata Do
Frequency (Hz)
16384
8192
4096
2048
1024
512
256
128
64
0
0:00
0:15
0:30
0:45
1:00
1:15
1:30
1:45
2:00
Time (s)
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Hip-Hop - Lagav e Lu Jab Lipistick
Frequency (Hz)
16384
8192
4096
2048
1024
512
256
128
64
0
0:00
0:50
1:40
2:30
3:20
4:10
5:00
Time (s)
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Electronic - One More Time
Frequency (Hz)
16384
8192
4096
2048
1024
512
256
128
64
0
0:00
0:15
0:30
0:45
1:00
1:15
1:30
1:45
2:00
2:15
Time (s)
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB