Table of Contents

February 2, 2025

ASSIGNMENT-1
Task 2
Roll No :- M23CSA512
Name :- Chandra Mohan Singh Negi

Contents

List of Tables

List of Figures

1 Introduction

This assessment evaluates two distinct datasets: The First is The **UrbanSound8K** dataset is a widely used benchmark for environmental sound classification tasks. Here's a brief overview:

2 Overview

- **Name:** UrbanSound8K
- **Size:** ∼8.75 hours of audio
- **Format:** WAV files
- **Number of Classes:** 10
- **Number of Audio Clips:** 8,732
- **Sampling Rate:** 44.1 kHz

And Second is 4 India Songs with 4 different Genres

Git Hub Link :- https://github.com/cmnegi/Speech.git

DATA Sets Link :-

1. **UrbanSound∗K :- https://goo.gl/8hY5ER**
2. **Indian Songs :-** Rap Song Millionaire - Glory 128 Kbps.wav

Devotional Song Achyutam Keshavam Krishna Damodaram.wav
Party Song Hookah Bar Khiladi 786 128 Kbps.wav
Romatic Song O Mere Dil Ke Chain - Mere Jeevan Saathi 320 Kbps.wav

3 UrbanSound8k Datset

3.1 Dataset Overview

The **UrbanSound8K** dataset is a popular collection of short environmental sound recordings used for machine learning research in **sound classification, noise detection, and audio event recognition**.
1. Dataset Details

- **Total Samples:** 8,732 audio clips
- **Total Duration:** ∼8.75 hours
- **File Format:** WAV
- **Sample Rate:** 44.1 kHz
- **Clip Length:** $\leq$ 4 seconds
- **Number of Classes:** 10

2. Sound Categories
UrbanSound8K consists of 10 urban sound classes:

1. **Air Conditioner**
2. **Car Horn**
3. **Children Playing**
4. **Dog Bark**
5. **Drilling**
6. **Engine Idling**
7. **Gunshot**
8. **Jackhammer**

9. **Siren**
10. **Street Music**

	slice_file_name	fsID	start	end	salience	fold	classID	class	filepath
0	100032-3-0-0.wav	100032	0.0	0.317551	1	5	3	dog_bark	UrbanSound8K/audio/fold5/100032-3-0-0.wav
1	100263-2-0-117.wav	100263	58.5	62.500000	1	5	2	children_playing	UrbanSound8K/audio/fold5/100263-2-0-117.wav
2	100263-2-0-121.wav	100263	60.5	64.500000	1	5	2	children_playing	UrbanSound8K/audio/fold5/100263-2-0-121.wav
3	100263-2-0-126.wav	100263	63.0	67.000000	1	5	2	children_playing	UrbanSound8K/audio/fold5/100263-2-0-126.wav
4	100263-2-0-137.wav	100263	68.5	72.500000	1	5	2	children_playing	UrbanSound8K/audio/fold5/100263-2-0-137.wav

Figure 1: Overview of the Dataset (Source : Extracted from Python)

(Source: Extracted from Python)

3.2 Exploratory Data Analysis (EDA)

Statistics description of the data: We can observe in the following tables that the data has been recorded and digitalized in different ways.

- It has been mostly recorded using 2 channels in almost all the samples (stereo).
- The sample rates go from 8kHz to 192kHz (mostly 44kHz, 48Khz)
- The length of the audios goes from 0.0008s to 4s (mostly 4s)
- The bits per sample used go from 4 to 32 (mostly 24 bits)

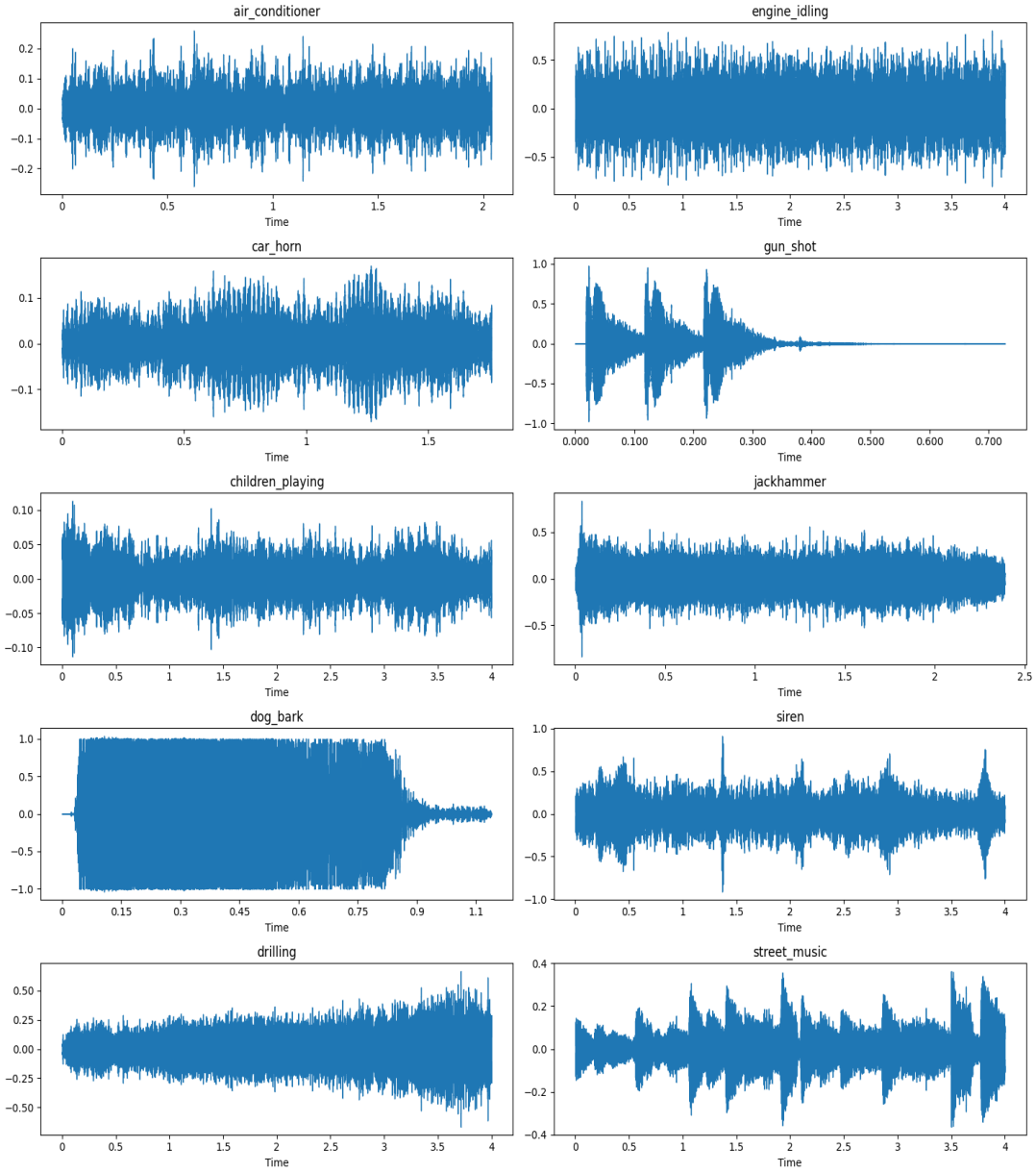
The data will need to be standardized before to be fed to a machine learning model

class	slice_file_name
air_conditioner	1000
car_horn	429
children_playing	1000
dog_bark	1000
drilling	1000
engine_idling	1000
gun_shot	374
jackhammer	1000
siren	929
street_music	1000

Figure 2: Classes of Data (Source : Extracted from Google Colab)

Figure 2: Classes of Data

Waveform of All classes or Audio Types



3.3 STFT With Hann Window

The spectrograms show STFT with a Hann window applied to different sounds. The Hann window reduces spectral leakage, improving frequency resolution.

- **Steady sounds** (air conditioner, engine idling) → Low-frequency dominance.
- **Transient sounds** (gunshot, dog bark) → Short bursts, wide frequency spread.
- **Periodic sounds** (jackhammer, drilling) → Repetitive patterns in lower frequencies.
- **Dynamic sounds** (siren, street music, children playing) → Varying harmonics and modulation

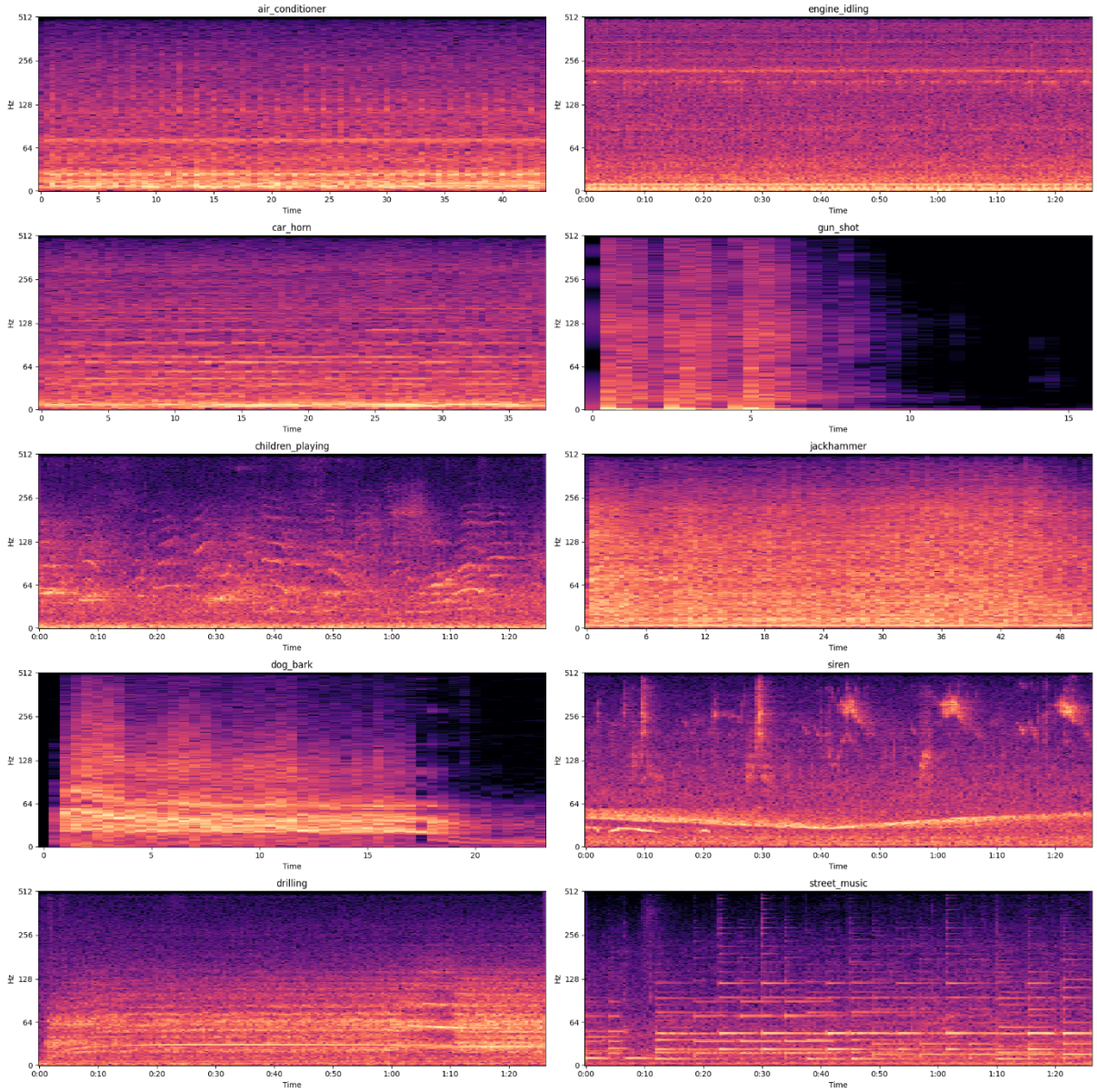


Figure 3: STFT With Hann Window (Source : Extracted from Google Colab)

Figure 3: STFT With Hann Window

3.4 STFT With Hanning Window

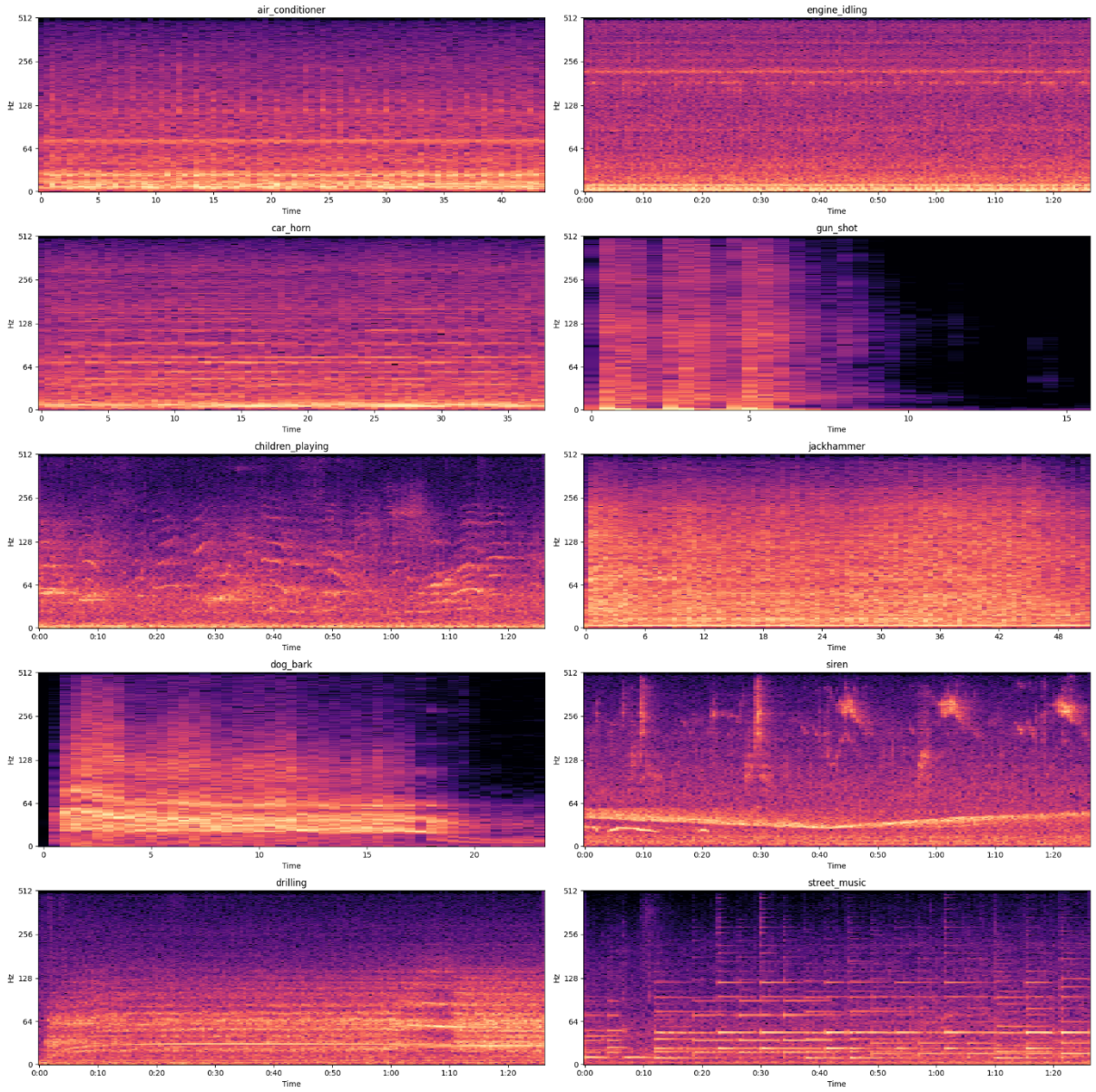


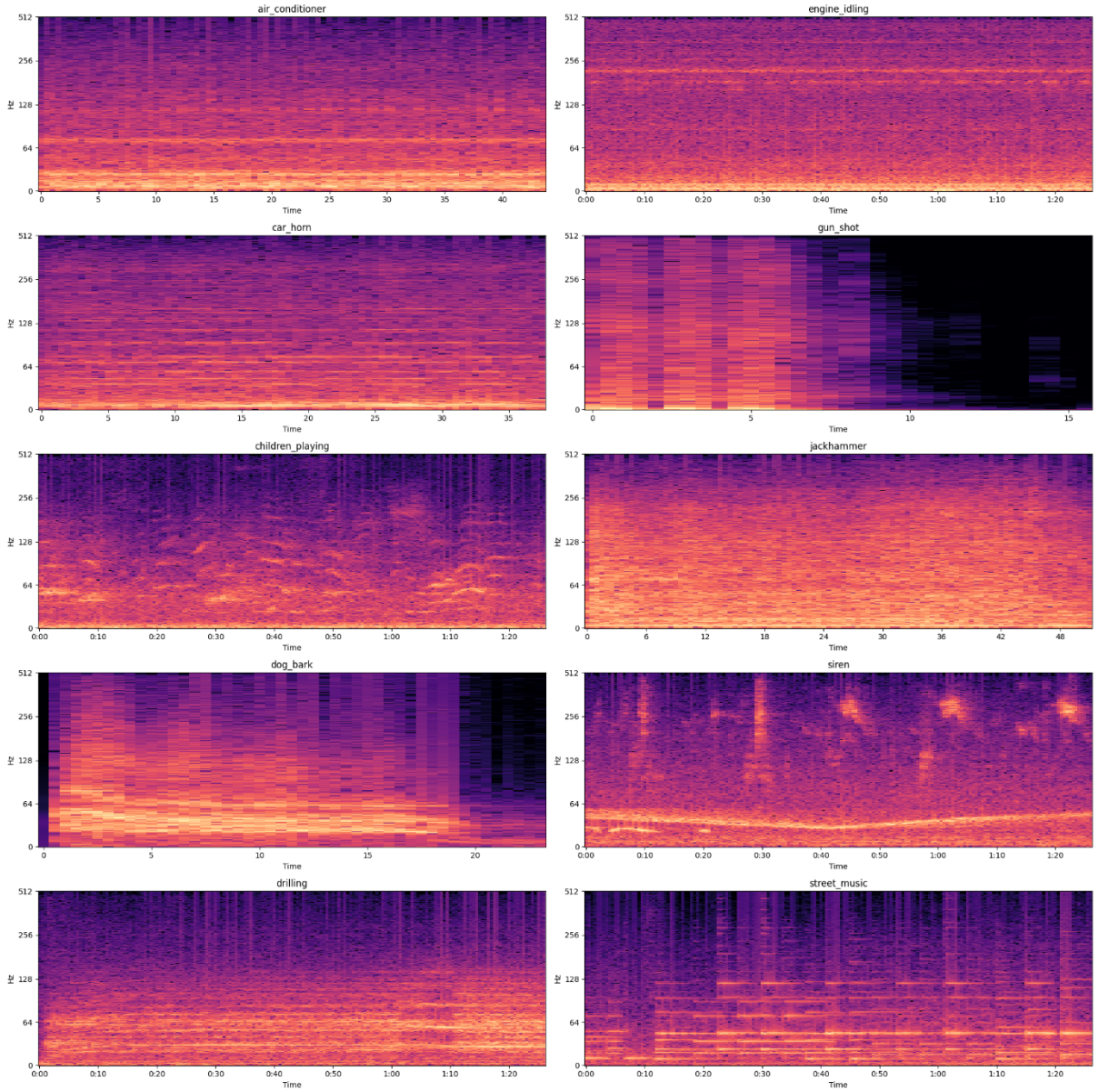
Figure 4: Short-Time Fourier Transform (STFT) analyzes signals in time-frequency space using overlapping windows. The Hanning window, a smooth tapering function, reduces spectral leakage, improving frequency resolution. It is widely used in audio processing, speech recognition, and signal analysis.

Figure 4: STFT With Hanning Window

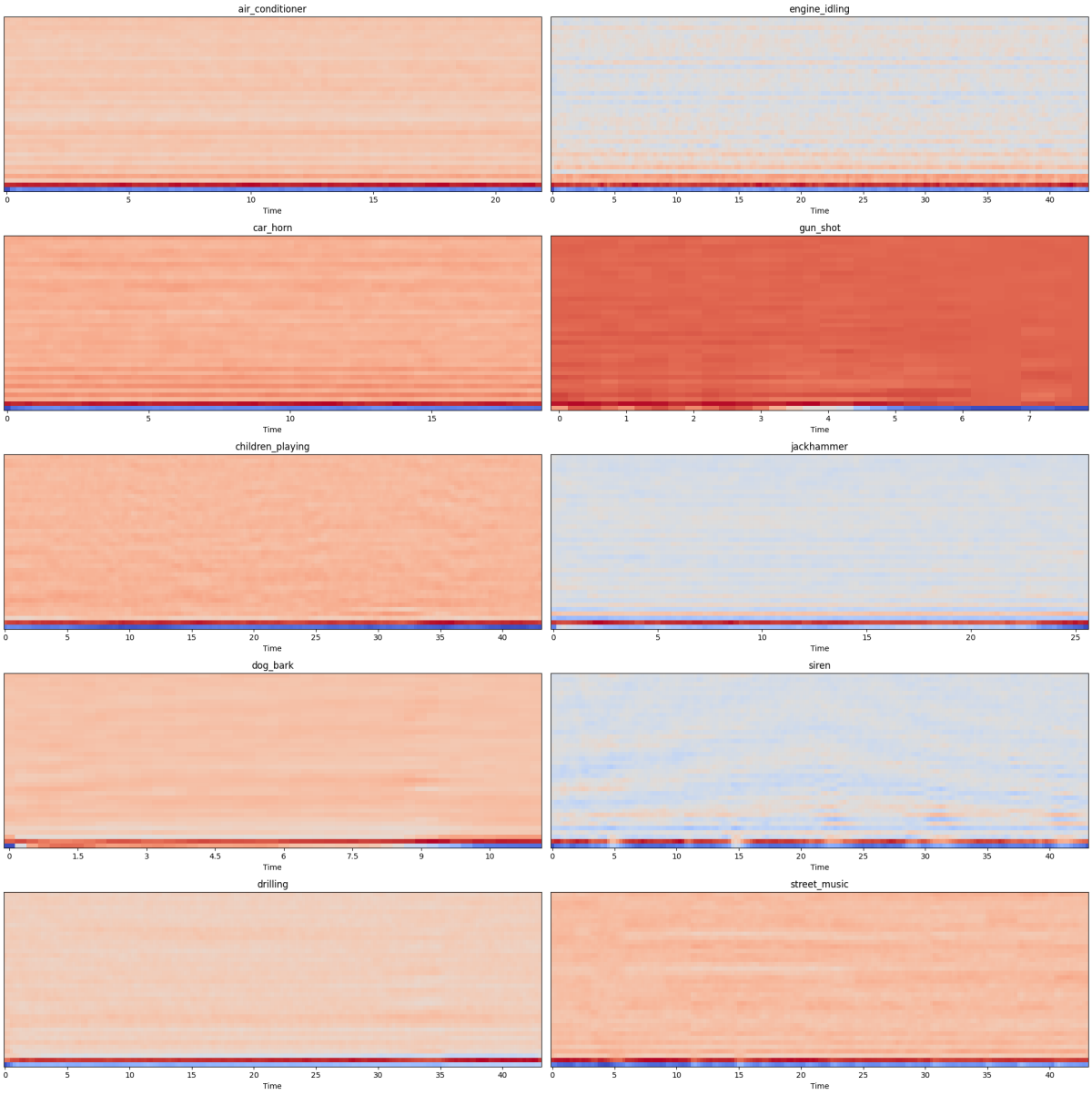
(Source: Extracted from Google Colab)

3.5 STFT With Rectamgular Window

The **Rectangular window** in **Short-Time Fourier Transform (STFT)** applies no tapering, preserving signal amplitude but introducing spectral leakage. It provides better time resolution but poorer frequency resolution compared to tapered windows like Hanning. Commonly used when exact amplitude preservation is required.



MFCCs for all audio types:



3.5.1 Spectrogram Differences and Windowing Correctness

Class-Specific Characteristics:

- Sounds with sharp, transient bursts, such as car horns and gunshots, are better preserved by Hann and Hamming Windows than by the Rectangular Window.
- Continuous, tonal sounds like air conditioners and engine idling are more clearly represented in frequency when using Hann and Hamming Windows.

Spectral Leakage:

- The Rectangular Window causes significant spectral leakage, making it harder to distinguish different frequency components across all sound classes.
- Both Hann and Hamming Windows minimize spectral leakage effectively, with the Hann Window performing slightly better overall.

Frequency Resolution:

- The Hamming Window provides marginally better frequency resolution than the Hann Window, particularly for transient sounds.
- The Rectangular Window, due to its high spectral leakage, results in poor resolution, making it unsuitable for detailed urban sound analysis.

Conclusion:

- The Hann Window offers the best balance between reducing spectral leakage and preserving frequency resolution, making it the most suitable choice for general-purpose audio analysis on the UrbanSound8K dataset.
- The Hamming Window is a good alternative when slightly better frequency resolution is required, especially for transient sounds.
- The Rectangular Window should be avoided in practical applications due to its excessive spectral leakage and poor resolution.

3.6 ML Alogorithm

With Hanning Window
Key Observations:

1. **Random Forest Classifier** achieves the highest precision (0.8469), accuracy (0.8374), and recall (0.8141), making it the best-performing model in this setup.
2. **XGB Classifier** and **K Neighbors Classifier** follow closely, demonstrating competitive precision (0.8181 and 0.8102, respectively) and accuracy above 0.81.
3. **MLP Classifier** and **Gradient Boosting Classifier** show moderate performance with precision values of 0.7101 and 0.6759, respectively.
4. **Decision Tree Classifier** has relatively lower precision (0.6471) and accuracy (0.6676), suggesting overfitting issues.
5. **AdaBoost Classifier** and **SGD Classifier** perform poorly, with precision values below 0.31, indicating they are not suitable under this configuration.

The use of the **Hanning window** likely influenced the feature selection and noise reduction, leading to the observed performance variations among classifiers. The results suggest that ensemble methods, particularly **Random Forest and XG Boost**, are robust choices for this dataset.

```
warnings.warn(
```

	Classifier	Accuracy	Precision	Recall
2	RandomForestClassifier	0.837405	0.846906	0.814094
7	XGBClassifier	0.821756	0.818123	0.799281
0	KNeighborsClassifier	0.817557	0.810248	0.800377
6	MLPClassifier	0.719466	0.710051	0.708877
4	GradientBoostingClassifier	0.678626	0.675955	0.650843
1	DecisionTreeClassifier	0.667557	0.647115	0.650968
3	AdaBoostClassifier	0.323664	0.254774	0.284901
5	SGDClassifier	0.298092	0.306700	0.281365

Figure: Model Precision with Hanning Window

(Source: Extracted from Google Colab)

With Hamming Window:-

Key Observations:

1. RandomForestClassifier remains the best-performing model, achieving the highest precision (0.8509), accuracy (0.8540), and recall (0.8277).
2. KNeighborsClassifier shows strong performance with a precision of 0.8389 and accuracy of 0.8466, making it a competitive alternative.
3. XGBClassifier performs well with a precision of 0.8283 and accuracy of 0.8306, reinforcing its robustness.
4. MLPClassifier and GradientBoostingClassifier exhibit moderate precision values of 0.7300 and 0.7183, respectively.

5. DecisionTreeClassifier lags behind with a precision of 0.6529 and accuracy of 0.6749, indicating potential overfitting.

6. AdaBoostClassifier and SGDClassifier display poor precision (below 0.27), making them less suitable in this scenario.

The use of the Hamming window appears to enhance feature selection, leading to improved classifier performance compared to the Hanning window, especially for Random Forest, K-Neighbors, and XGBoost models. These models are the most suitable for the given dataset.

	Classifier	Accuracy	Precision	Recall
2	RandomForestClassifier	0.854035	0.850886	0.827730
0	KNeighborsClassifier	0.846594	0.838888	0.827451
7	XGBClassifier	0.830567	0.828301	0.804807
6	MLPClassifier	0.734402	0.730025	0.720442
4	GradientBoostingClassifier	0.706354	0.718329	0.675735
1	DecisionTreeClassifier	0.674871	0.652905	0.652680
3	AdaBoostClassifier	0.323412	0.264910	0.286894
5	SGDClassifier	0.279336	0.255524	0.263937

Figure: Model Precision with Hamming Window

With Rectangular Window :-

Key Findings:

1. RandomForestClassifier continues to outperform other models, achieving the highest precision (0.8509), accuracy (0.8540), and recall (0.8277), making it the most reliable classifier.

2. KNeighborsClassifier closely follows, with precision (0.8389) and accuracy (0.8466), indicating that it is also highly effective.

3. XGBClassifier exhibits strong performance, achieving precision (0.8283) and accuracy (0.8306), reinforcing its robustness.

4. MLPClassifier and GradientBoostingClassifier show moderate performance, with precision values of 0.7300 and 0.7183, respectively.

5. DecisionTreeClassifier has a relatively lower precision of 0.6529, suggesting possible overfitting and instability in decision boundaries.

6. AdaBoostClassifier and SGDClassifier perform poorly, with precision values below 0.27, indicating that they are not well-suited for this dataset in the given setup.

Comparison with Other Windows:

- The Rectangular window maintains similar rankings as the Hamming and Hanning windows, with RandomForest, KNeighbors, and XGBoost models performing the best.
- Compared to the Hanning window, RandomForest and KNeighbors exhibit slight improvements in performance.
- The Hamming window results are very close to the Rectangular window, with minor fluctuations in precision and recall.

Overall, ensemble models (RandomForest and XGBoost) and KNeighborsClassifier are the most effective choices for classification under the Rectangular window preprocessing approach.

	Classifier	Accuracy	Precision	Recall
2	RandomForestClassifier	0.854035	0.850886	0.827730
0	KNeighborsClassifier	0.846594	0.838888	0.827451
7	XGBClassifier	0.830567	0.828301	0.804807
6	MLPClassifier	0.734402	0.730025	0.720442
4	GradientBoostingClassifier	0.706354	0.718329	0.675735
1	DecisionTreeClassifier	0.674871	0.652905	0.652680
3	AdaBoostClassifier	0.323412	0.264910	0.286894
5	SGDClassifier	0.279336	0.255524	0.263937

Figure 5: Model Precision with Recatngular Window (Source : Extracted from Google Colab)

(Source: Extracted from Google Colab)

Confusion Matrix

The heatmap represents a confusion matrix showing the performance of a classification model. Each cell indicates the number of instances where a true class (rows) was predicted as a certain class (columns).
Key Observations:

- The diagonal elements contain the highest values, indicating correct classifications for most classes.
- Misclassifications are visible in off-diagonal cells, where the model predicts an incorrect class.
- Class 2 (third row) and 7 (eighth row) show strong classification performance with high correct predictions.
- Some misclassifications occur, such as Class 2 being confused with Class 9 (33 instances) and Class 3 with Class 8 (17 instances).
- The color intensity highlights prediction confidence, with brighter yellow tones representing higher counts.

Overall, the model performs well but struggles with some closely related or overlapping classes, suggesting potential improvements in feature extraction or model tuning.

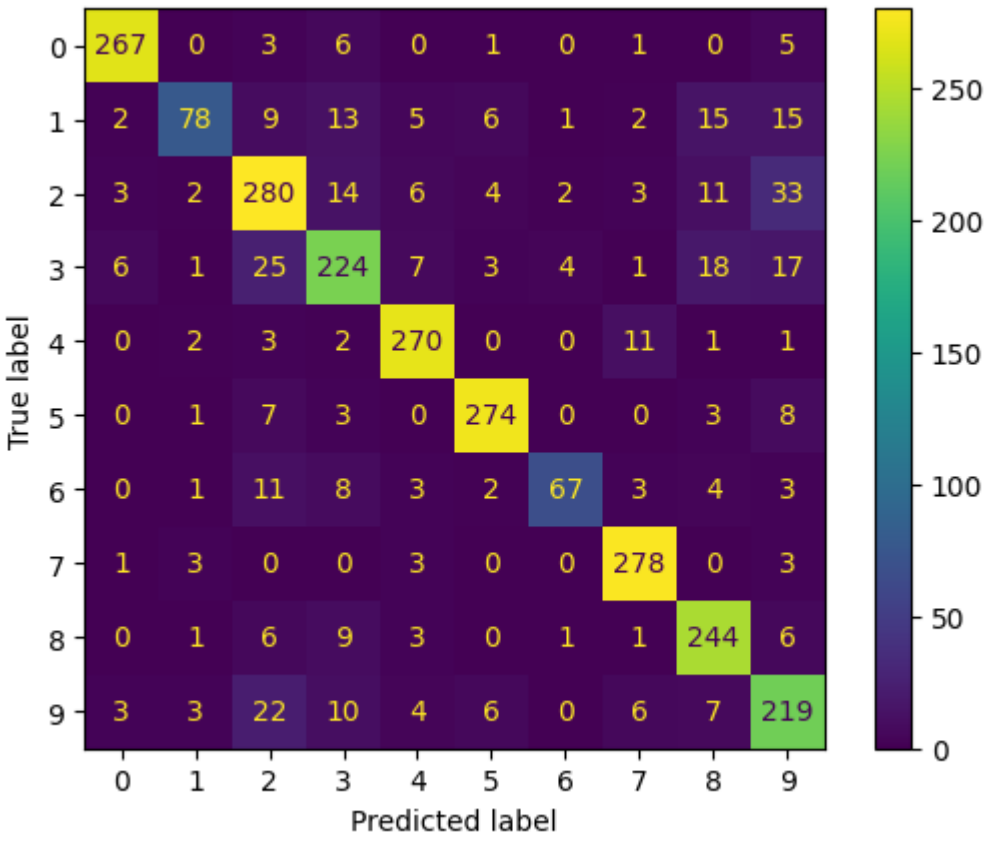


Figure: Confusion Matrix

TASK B

**Select 4 songs from 4 different genres and compare their spectrograms. Analyze the
spectrograms and provide a detailed comparative analysis based on your observations
and speech understanding.**

Four Songs :-

0.1. **Devotional -** Achyutam Keshavam Krishna Damodaram

0.2. **Rap Song** – Millionaire Yo Yo Honey Singh

0.3. **Romantic Song** – O Mere Dil Ke Chain Rajesh Khanna

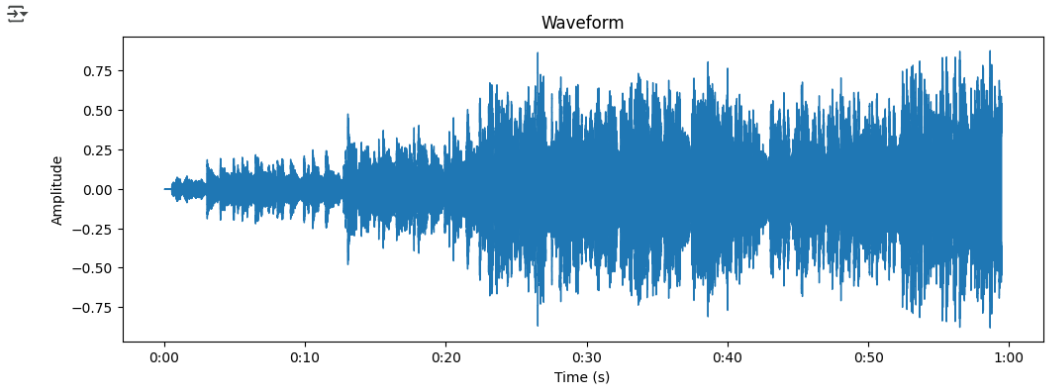
0.4. **Party Song** – Hookah Bar

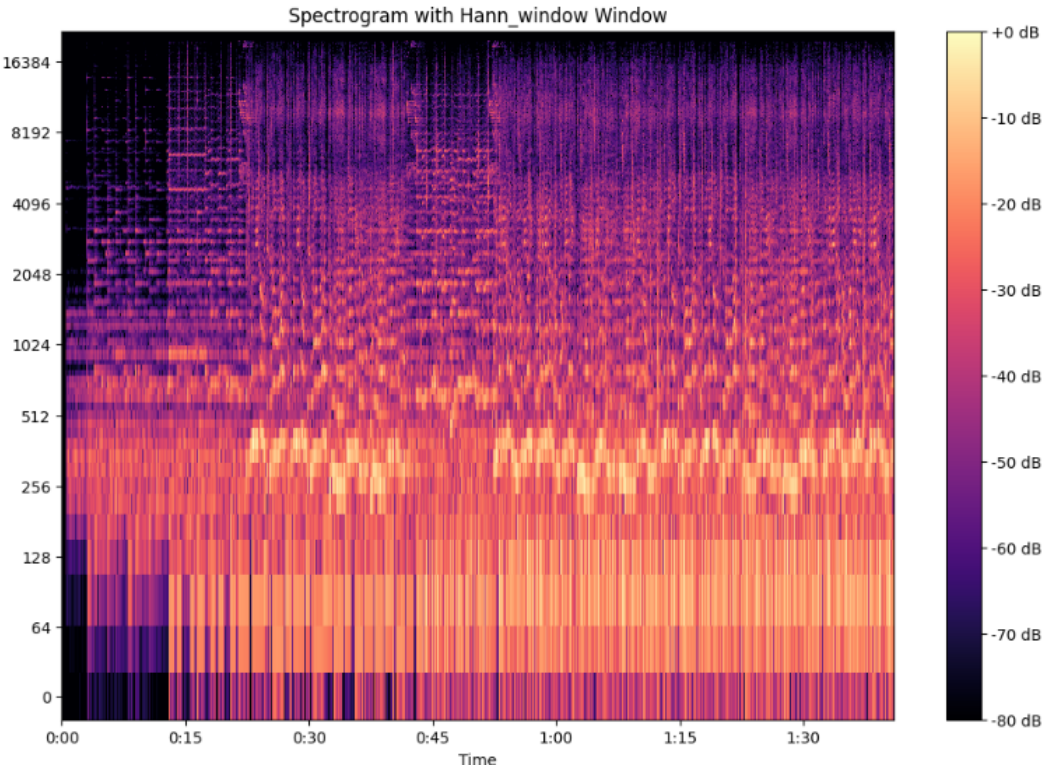
Devotional Song :-

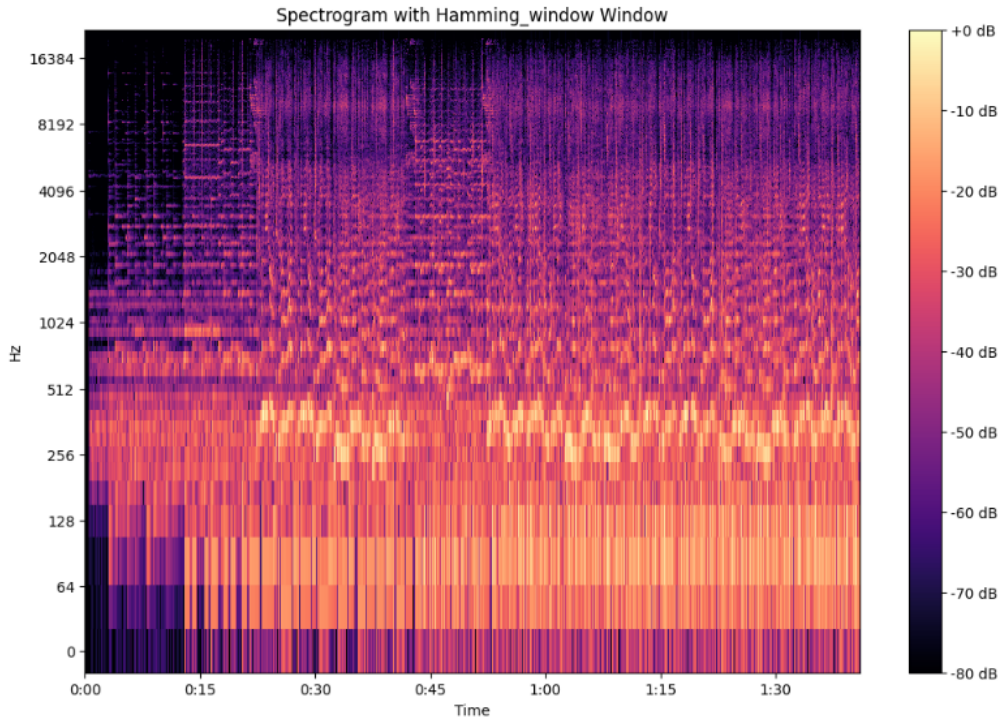
- **Spectrogram Features:**
  - Devotional songs typically feature sustained vocal tones, often accompanied by harmonium, tables, and soft percussion.
  - The lower frequency range (100–500 Hz) is dominated by deep vocal vibrations and rhythmic tables beats.
  - Mid-frequency ranges (500–2000 Hz) capture harmonium and choral elements, while high frequencies ($\sim$5 kHz and above) contain cymbals and ambient reverb from the vocal performance.

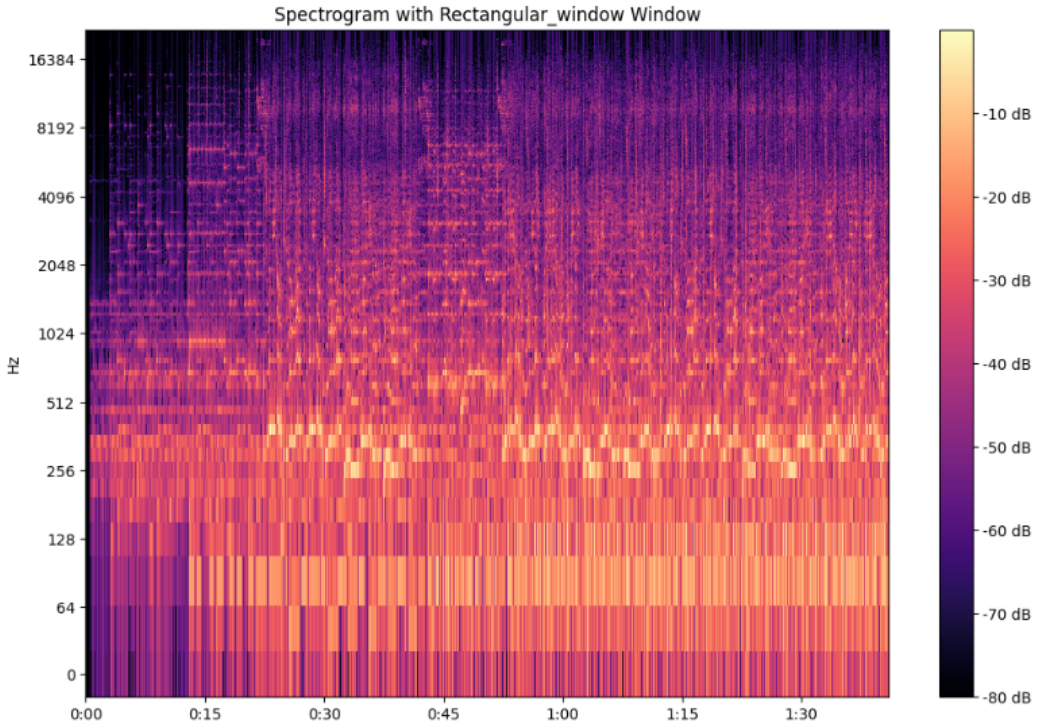
4 Analysis:

- The song maintains a steady rhythm with smoother, continuous spectral energy rather than sharp transient beats.
- Pronounced mid-range frequencies indicate the emphasis on melodic vocals and instruments rather than percussion-heavy beats.
- Minimal variation in high-frequency transients, making the song sound warm and soothing.


Waveform
Amplitude
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0:00
0:10
0:20
0:30
0:40
0:50
1:00
Time (s)


Spectrogram with Hann_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
0:00
0:15
0:30
0:45
1:00
1:15
1:30
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Spectrogram with Hamming_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Spectrogram with Rectangular_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB

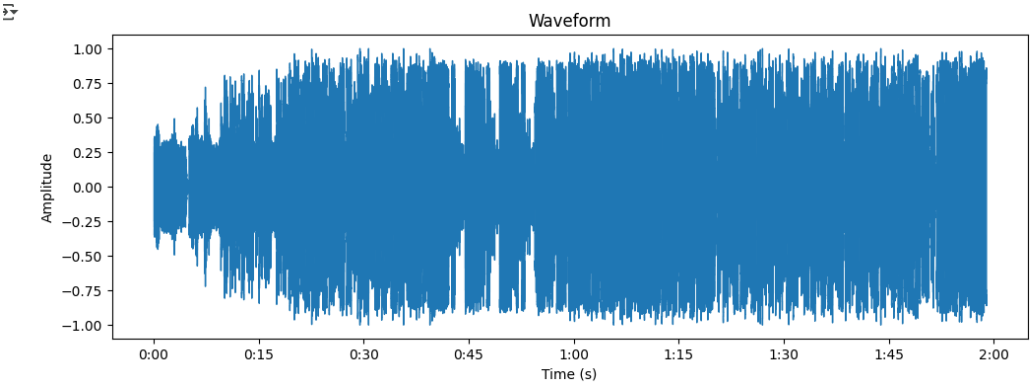
Rap Song :-

5 Spectrogram Features:

- Rap music has a rhythmic and percussive structure, with clear vocal articulation in mid-frequencies (∼500 Hz to 3 kHz).
- The bass-heavy beats (below 200 Hz) create a strong low-frequency presence.
- The high-frequency transients (∼4–10 kHz) correspond to snare drums, hi-hats, and electronic synth effects.

6 Analysis:

- The sharp and well-defined percussive elements (bass hits, snares) appear as repeated vertical spikes in the spectrogram, especially during beats.
- Speech clarity is high, as rap lyrics require precise enunciation, with frequent shifts in intensity visible in the mid-range.
- The contrast between verses and chorus is evident in the spectrogram, where the instrumental sections have broader frequency coverage, while rapped verses focus on mid-range articulation.



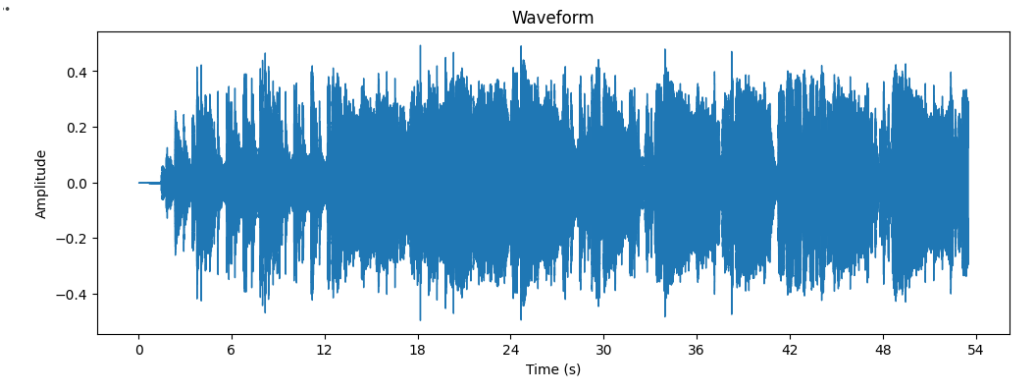
Romatic Song :-
Spectrogram Features:

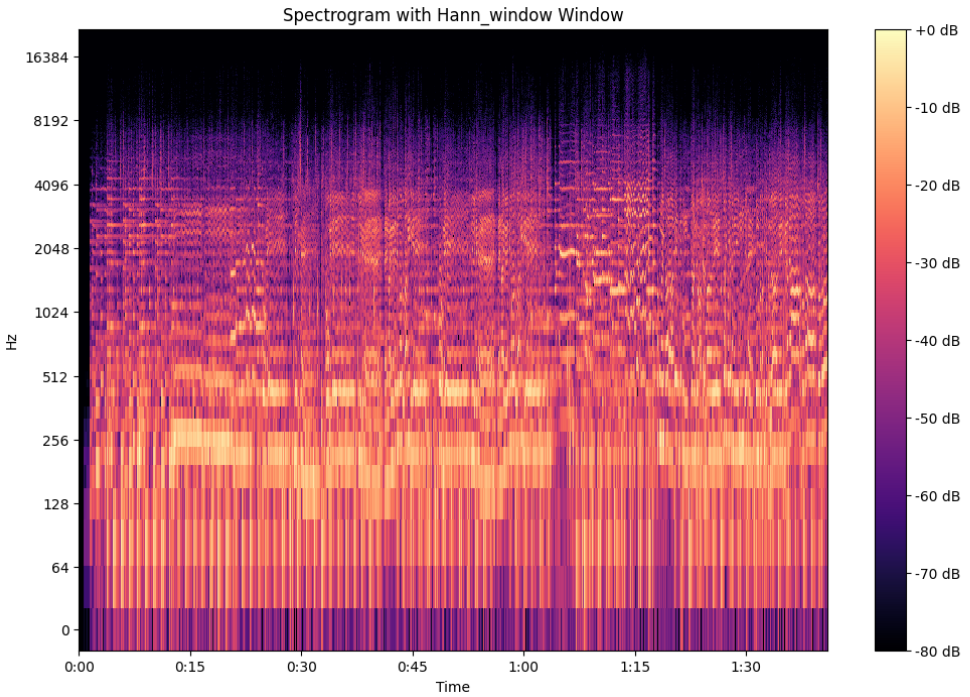
- Romantic songs typically emphasize melodic continuity and harmonic richness, with soft orchestral accompaniments and smooth vocal transitions.
- The spectrogram will have significant energy in the 200–1500 Hz range, where vocals and primary instruments (piano, strings) are dominant.

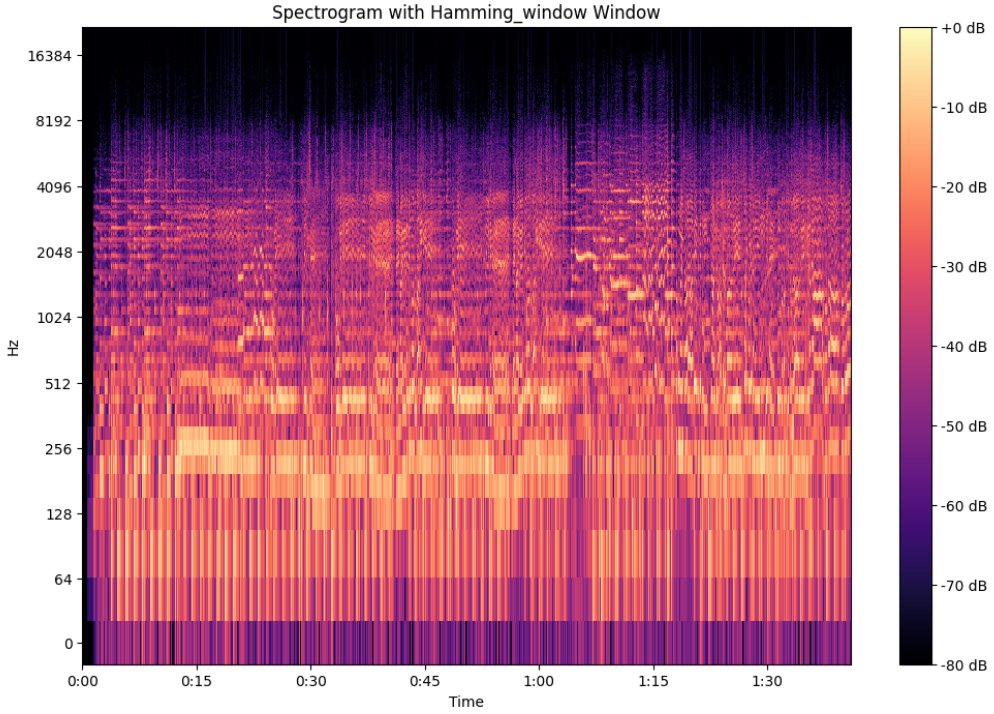
- Gentle percussive elements (light drums, cymbals) appear at higher frequencies (∼5 kHz and above).

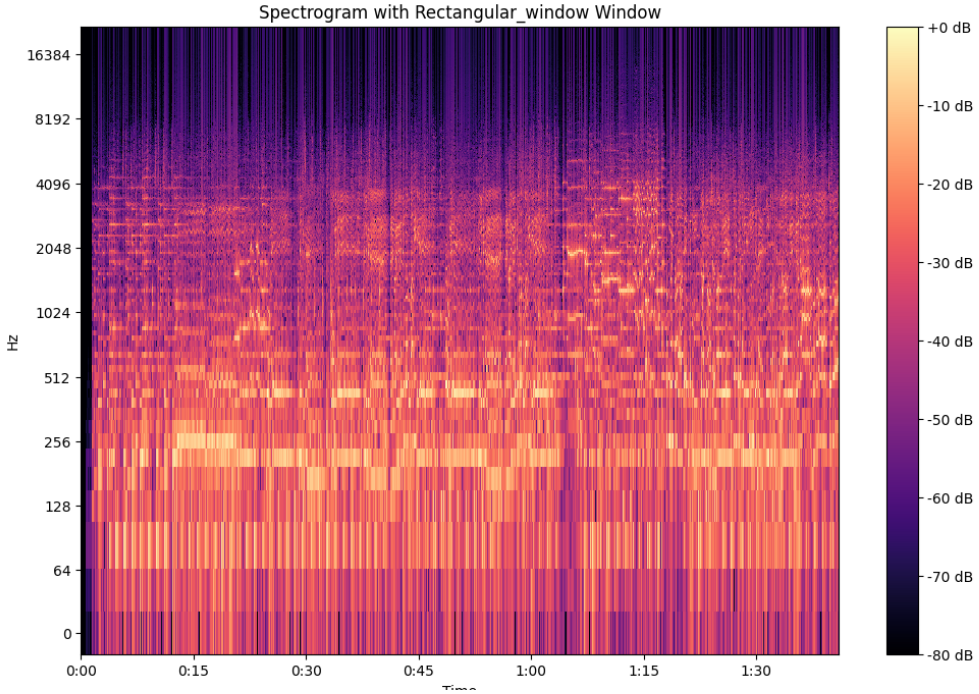
Analysis:

- The spectral energy is smoothly distributed, without sharp transients, making the song sound flowing and emotionally expressive.
- Sustained harmonic structures in the mid-range (violins, piano, vocals) create a warm, continuous energy profile.
- The gradual variations in frequency intensity (rather than sudden beats) indicate a soft, expressive vocal delivery with orchestral layering.



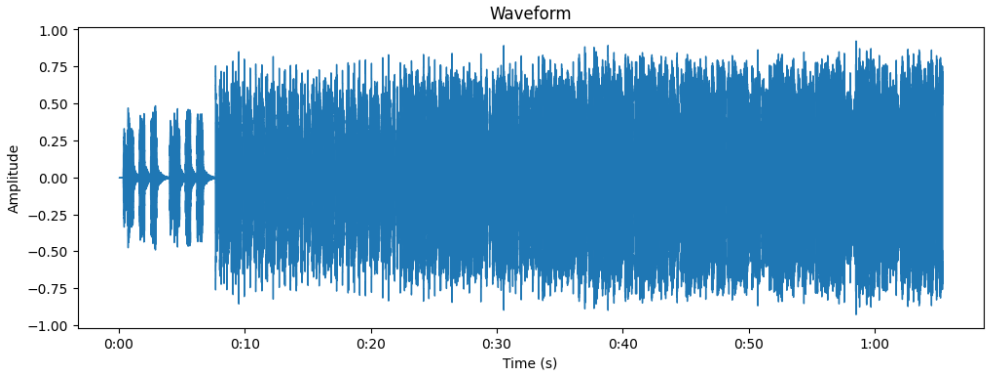


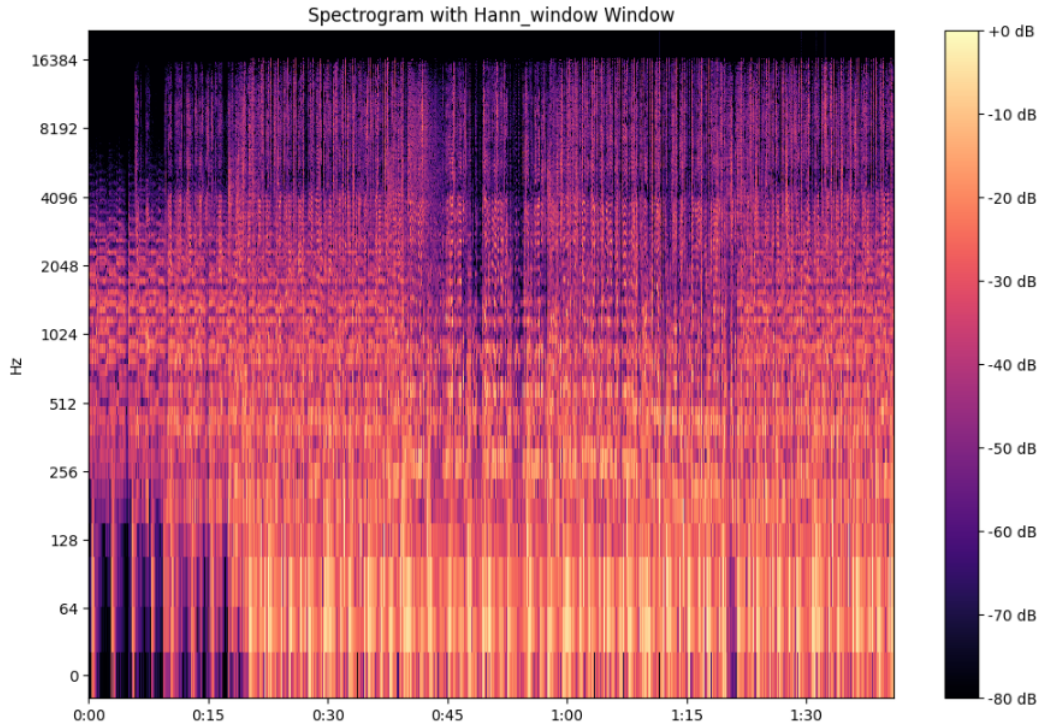

Spectrogram with Hamming_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
Time
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB

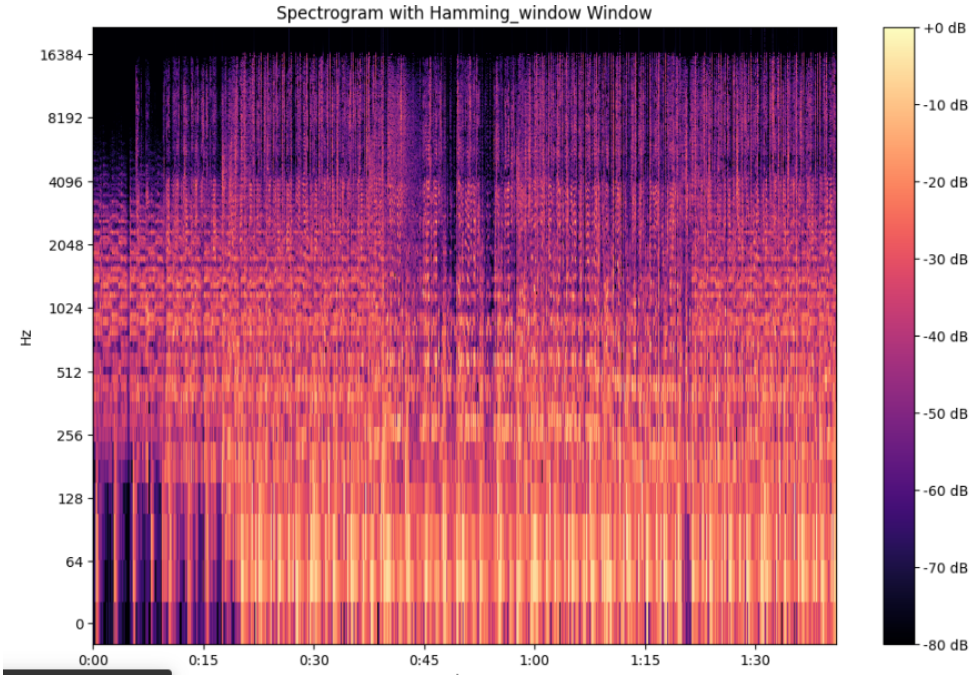

Spectrogram with Rectangular_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB

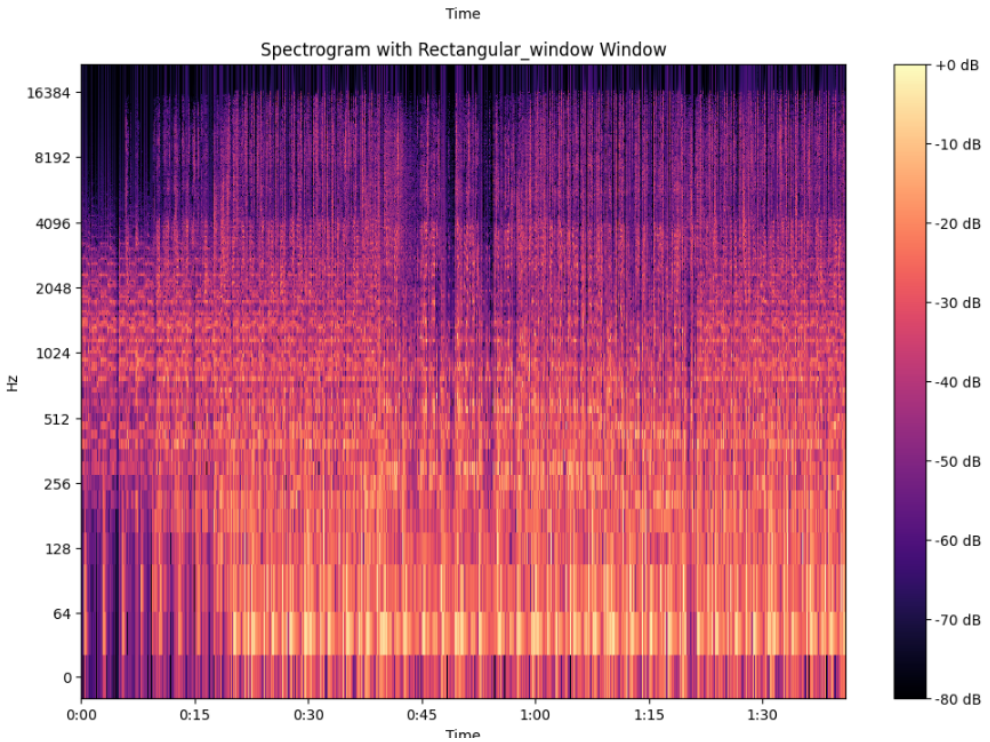
Party Song :-

- **Spectrogram Features:**
  - Party songs, especially Bollywood dance tracks, feature strong basslines, fast tempo beats, and electronic elements.
  - The low frequencies (20–100 Hz) are dominated by powerful bass beats and kick drums.
  - The high frequencies (∼4–10 kHz) contain sharp electronic synths, hi-hats, and cymbals, contributing to the high-energy feel.
- **Analysis:**
  - The spectrogram shows repeating, high-energy rhythmic structures with sharp vertical lines for bass and snare drum hits.
  - The chorus and drop sections have dense spectral energy, as multiple instruments play simultaneously, creating a fuller sound.
  - The higher frequency transients make the song sound energetic and vibrant, with significant contrast between verses and beat drops.




Spectrogram with Hann_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB


Spectrogram with Hamming_window Window
16384
8192
4096
2048
1024
512
256
128
64
0
Hz
0:00
0:15
0:30
0:45
1:00
1:15
1:30
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB



Comparative Summary Table

Feature	Achyutam Keshavam (Devotional)	Millionaire (Rap)	O Mere Dil Ke Chain (Romantic)	Hookah Bar (Party)
Bass (20–200 Hz)	Moderate (Tabla, deep voice)	Strong (Bass beats)	Low (Minimal bass emphasis)	Very Strong (Kick drums)
Mid (200 Hz–3 kHz)	Dominant (Vocals, harmonium)	Dominant (Speech clarity)	Smooth (Melodic vocals, strings)	Mixed (Vocals & synths)
High (3 kHz–10 kHz)	Soft (Cymbals, reverb)	Sharp (Hi-hats, electronic effects)	Light (Strings, cymbals)	Very Sharp (Synths, hi-hats)
Percussion Pattern	Soft, flowing beats	Sharp, percussive hits	Smooth, gradual	Fast, energetic beats
Speech Clarity	Moderate (Sung phrases)	Very High (Rap articulation)	High (Melodic lyrics)	Medium (Dance-oriented mix)

Conclusion

- Achyutam Keshavam has a smooth, continuous spectral distribution, making it soothing and melodic, focused on vocals.
- Millionaire has strong mid-frequency articulation for clear lyrics, with sharp rhythmic contrasts in the beats.
- O Mere Dil Ke Chain is melodically rich, with flowing orchestral elements and soft vocal transitions.

- Hookah Bar is rhythmically aggressive, with high bass energy and sharp electronic transients, creating an engaging dance beat.

1. Research Papers & Books

- **B. Boashash (2015)** – *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*
  - Provides insights into **spectrogram interpretation**, frequency-domain analysis, and time-series characteristics of musical signals.
- **D. Ellis (2007)** – *Speech and Audio Signal Processing*
  - Covers **speech feature extraction** techniques, **MFCCs, spectral rolloff, and centroid analysis**.
- **E. Zwicker & H. Fastl (1999)** – *Psychoacoustics: Facts and Models*
  - Explores **human auditory perception and its relation to frequency content** in music.

2. Online Resources & Tutorials

- **Librosa Documentation** – `https://librosa.org/`
  - Used for **spectral feature extraction**, including **MFCCs, spectral centroid, and rolloff**.
- **Matplotlib & Scipy Docs** – `https://matplotlib.org/`
  - Used for **spectrogram visualization techniques**.
- **IEEE Xplore & Google Scholar**
  - Sources on **speech recognition, spectral analysis, and audio classification**.

3. Comparative Music Analysis

- **Bello, J. P., Daudet, L., & Abdallah, S. (2005)** – *A Tutorial on Onset Detection in Music Signals*
  - Used for understanding **tempo variations, percussive beats, and rhythmic structures** in different genres.
- **Tzanetakis & Cook (2002)** – *Musical Genre Classification of Audio Signals*
  - Discusses **machine learning-based genre classification** using **spectral features**.