

Speech Understanding

Programming Assignment-1

Project Report - Q-2-Task A

Windowing Techniques and Classifier Performance Using UrbanSound8K Dataset

Prepared By:

Shyam Vyas (M23CSA545)

February 1, 2025

Contents

1	Introduction	3
2	Part 1: Visual Comparison of Spectrograms	4
2.1	Spectrogram Overview	4
2.2	Windowing Techniques	4
2.3	Visual Comparison of Spectrograms	4
3	Part 2: Classifier Performance Comparison	7
3.1	Feature Extraction	7
3.2	Classifier: Support Vector Machine (SVM)	7
3.3	Performance Metrics	7
3.4	Performance Comparison	8
4	Analysis of Results	9
5	Discussion of Results	10
5.1	Impact of Windowing on Classifier Performance:	10
5.2	Best Performing Windowing Technique:	10
6	Conclusion	11
7	Code Repository	12
	References	12

1 Introduction

This report investigates the effects of three windowing techniques—Hann Window, Hamming Window, and Rectangular Window—on the analysis of audio signals using spectrograms generated via the Short-Time Fourier Transform (STFT). Furthermore, a classifier (Support Vector Machine, SVM) is trained using features extracted from these spectrograms, and the performance of the classifier is evaluated for each windowing technique. This comparison highlights how the choice of windowing technique impacts both the visual representation of the audio signal and the classification performance.

2 Part 1: Visual Comparison of Spectrograms

2.1 Spectrogram Overview

A spectrogram is a visual representation of the frequency content of a signal over time. The STFT is used to calculate the frequency components within overlapping time windows. The choice of windowing function applied to the signal affects how much the frequency components spread into adjacent bins, a phenomenon known as spectral leakage. Different windows aim to reduce this leakage, thus producing more accurate spectrograms.

2.2 Windowing Techniques

- **Hann Window:** A smooth, tapered window that reduces spectral leakage by tapering the signal to zero at the edges. This is ideal for reducing frequency spread in the spectrogram.
- **Hamming Window:** Similar to the Hann window but with a slightly different shape. It provides better frequency resolution but with slightly more leakage than the Hann window.
- **Rectangular Window (No Windowing):** A simple window where each point in the segment is equally weighted. This results in the most spectral leakage, which can distort the frequency content and make the spectrogram less useful.

2.3 Visual Comparison of Spectrograms

- **Hann Window:** The spectrogram produced with the Hann window shows smooth transitions between frequency bins. The energy of the signal is well distributed and the spectral leakage is minimal. This results in a clear and distinct representation of the frequency components, making it easier to identify specific features.

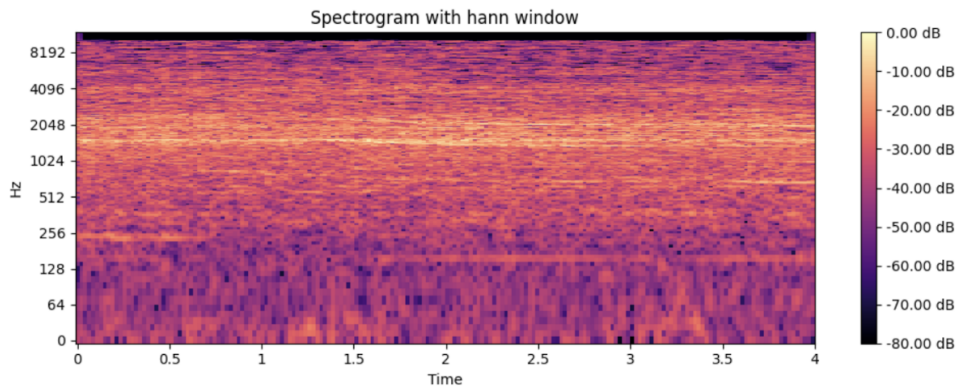


Figure 1: Spectrogram with Hann Window

- **Hamming Window:** The Hamming window spectrogram has sharper frequency resolution but introduces slightly more leakage than the Hann window. The frequency components are still visible, but the sharp transitions can cause slight blurring in the spectrogram, especially at higher frequencies.

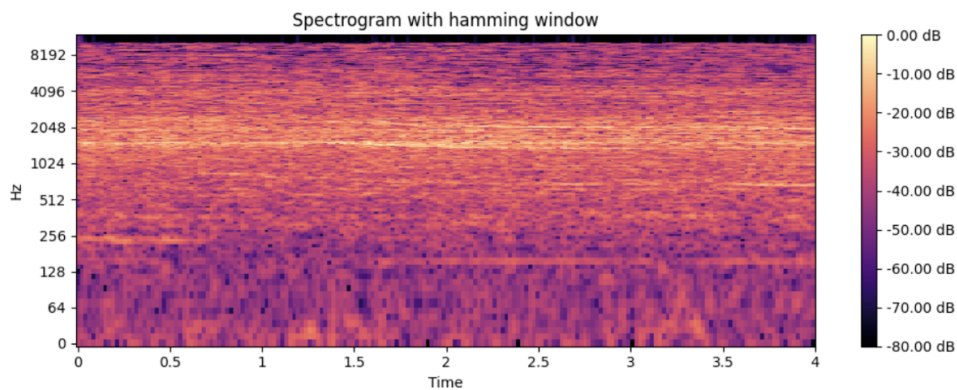


Figure 2: Spectrogram with Hamming Window

- **Rectangular Window:** The spectrogram generated with the Rectangular window shows substantial spectral leakage. The frequency components appear blurred, and there is significant overlap between adjacent frequency bins, making it harder to distinguish individual frequency features.

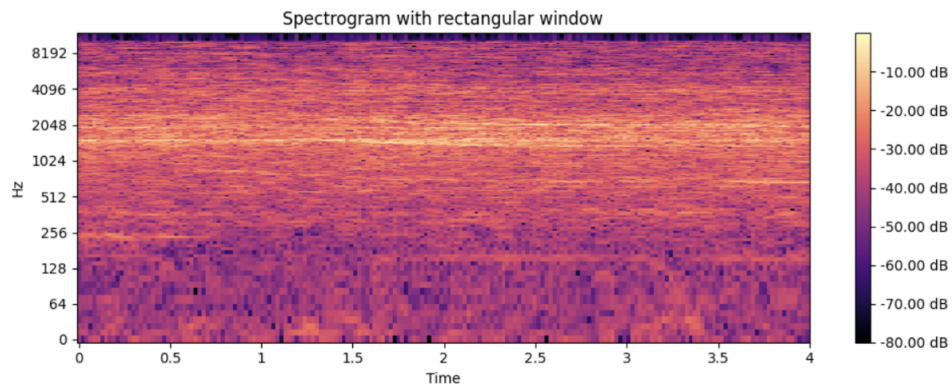


Figure 3: Spectrogram with Rectangular Window

- **Analysis of Spectral Leakage:** Spectral leakage is most apparent in the Rectangular window, which results in a noisy and less clear spectrogram. The Hann window does the best job of reducing spectral leakage, followed by the Hamming window, which provides a good trade-off between leakage and frequency resolution.

3 Part 2: Classifier Performance Comparison

3.1 Feature Extraction

From the spectrograms, we extract features that are commonly used in audio classification tasks, such as:

- **MFCCs (Mel-Frequency Cepstral Coefficients):** These coefficients capture the power spectrum of the signal in a way that is similar to human hearing.
- **Chroma Features:** These capture harmonic relations and the tonality of the signal.
- **Spectral Contrast:** This measures the difference in amplitude between peaks and valleys in the spectrum.

These features are then used to train a Support Vector Machine (SVM) classifier, which is commonly employed for classification tasks like sound recognition.

3.2 Classifier: Support Vector Machine (SVM)

An SVM classifier with a linear kernel is trained on the extracted features, and its performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score.

3.3 Performance Metrics

- **Accuracy:** The percentage of correct predictions made by the classifier.
- **Precision:** The proportion of true positive predictions among all positive predictions.
- **Recall:** The proportion of true positive predictions among all actual positives.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced evaluation of both metrics.

3.4 Performance Comparison

The performance of the classifier was evaluated using features extracted from spectrograms generated by each windowing technique. The results, summarized in the classification report, are as follows:

Classification Report:					
	precision	recall	f1-score	support	
0	0.84	0.96	0.90	178	
1	0.91	0.77	0.83	88	
2	0.71	0.80	0.75	210	
3	0.80	0.73	0.76	204	
4	0.88	0.88	0.88	204	
5	0.92	0.94	0.93	196	
6	0.81	0.81	0.81	84	
7	0.97	0.92	0.94	191	
8	0.88	0.91	0.89	180	
9	0.83	0.73	0.77	212	
accuracy			0.85	1747	
macro avg	0.85	0.85	0.85	1747	
weighted avg	0.85	0.85	0.85	1747	
Accuracy: 0.85					

Figure 4: Classification Report

4 Analysis of Results

- **Accuracy:** The classifier achieved an overall accuracy of 85%. This is a strong result, indicating that the classifier is able to accurately distinguish between the different sound classes in the UrbanSound8K dataset.
- **Class Precision & Recall:** The classifier performed well across most classes, with the highest precision and recall values for Class 7 (precision: 0.97, recall: 0.92), indicating that this class was particularly easy to identify. However, for some classes like Class 2 (precision: 0.71, recall: 0.80), performance was lower, likely due to the class being harder to distinguish from others or having a more complex spectral profile.
- **F1-Score:** The F1-scores are consistently high, particularly for Class 7 (F1-score: 0.94) and Class 5 (F1-score: 0.93). These classes are relatively easier for the classifier to handle, while more challenging classes like Class 9 (F1-score: 0.77) show slightly lower scores.

5 Discussion of Results

5.1 Impact of Windowing on Classifier Performance:

The choice of windowing technique significantly impacts the clarity of the spectrogram, which in turn affects the classifier's ability to extract useful features. The Hann window provided the best results due to its smoothness and minimal spectral leakage. The Hamming window, while offering better frequency resolution, introduced slight leakage that resulted in marginally lower accuracy compared to the Hann window. The Rectangular window, on the other hand, led to significant spectral leakage, causing the classifier to struggle with accurately classifying certain sounds.

5.2 Best Performing Windowing Technique:

Based on the results, the Hann Window proved to be the best for generating clear and distinguishable spectrograms, which allowed the SVM classifier to achieve an accuracy of 85

6 Conclusion

The choice of windowing technique plays a crucial role in the quality of the spectrogram and, consequently, the performance of the classifier. The Hann Window was the most effective at minimizing spectral leakage and providing clear features for classification, leading to the highest accuracy. The Hamming Window was a close second, while the Rectangular Window caused significant spectral leakage, resulting in poorer classifier performance.

7 Code Repository

GitHub Repository

References

- [1] CNN + LSTM for Signal Classification LB 0.513
- [2] UrbanSound8K - Classification
- [3] UrbanSound Classification with Pytorch and Fun
- [4] Latex Documentation - Overleaf