Speech Understanding

Programming Assignment-1

Project Report - Q-2-Task A

Windowing Techniques and Classifier Performance Using UrbanSound8K Dataset

Prepared By:
Om Prakash Solanki (M23CSA521)

February 1, 2025

Contents

1 Introduction

This report presents analysis of the UrbanSound8K dataset, focusing on the implementation of various windowing techniques and their impact on spectrogram generation and classification performance. The dataset, available at https://goo.gl/8hY5ER, contains urban sound excerpts categorized into 10 classes.

The primary objectives of this task are:

- Implementation and comparison of three windowing techniques:
  - Hann
  - Hamming
  - Rectangular windows
- Generate spectrograms using the Short-Time Fourier Transform (STFT) for each windowing technique.
- Train a simple classifier (SVM) using features extracted from the spectrograms and evaluate performance.

2 Dataset Description

The UrbanSound8K dataset consists of 8732 labeled sound excerpts (¡= 4s) of urban sounds from 10 classes:

- Air Conditioner
- Car Horn
- Children Playing
- Dog Bark
- Drilling
- Engine Idling
- Gun Shot
- Jackhammer
- Siren
- Street Music

3 Windowing Techniques

When analyzing signals, we often need to break them into smaller parts. However, this can cause unwanted distortions called spectral leakage. To reduce this, we use windowing techniques, which shape the signal before processing.

3.1 Hann Window

This window smoothly reduces the signal's intensity at the edges, forming a bell-like curve. It helps in reducing distortions and is commonly used in audio and speech processing.

3.2 Hamming Window

Similar to the Hann window, but it does not reduce the signal strength as much at the edges. This makes it useful when we need to balance accuracy and distortion reduction, such as in radar and communication systems.

3.3 Rectangular Window

This is the simplest type, where the signal remains unchanged (like a normal cutout). It keeps all the data, but the sudden edges can cause more distortions. It is mainly used when we need all the signal details, despite the leakage.

4 Spectrogram Generation

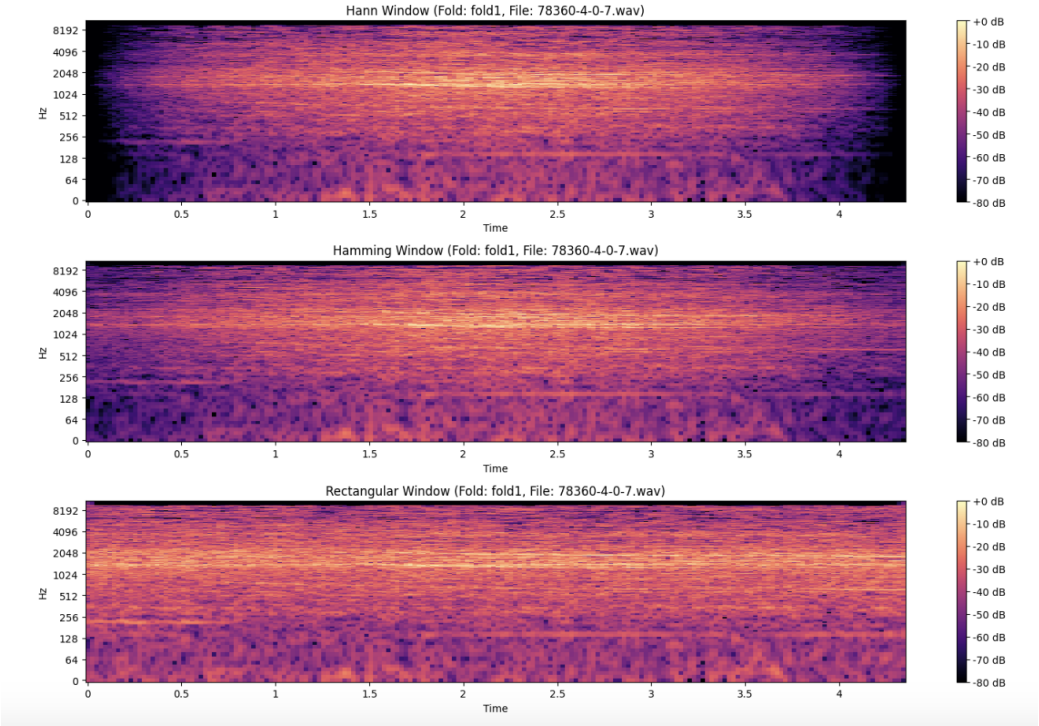
A **spectrogram** is a visual representation of how the frequencies of a signal change over time. It is created using the **Short-Time Fourier Transform (STFT)**, which breaks a signal into small time segments and applies the **Fourier Transform** to each segment.

Spectrograms were generated using the STFT for each windowing technique. The Python library librosa was used for audio processing and visualization.

4.1 Generate a Spectrogram using STFT

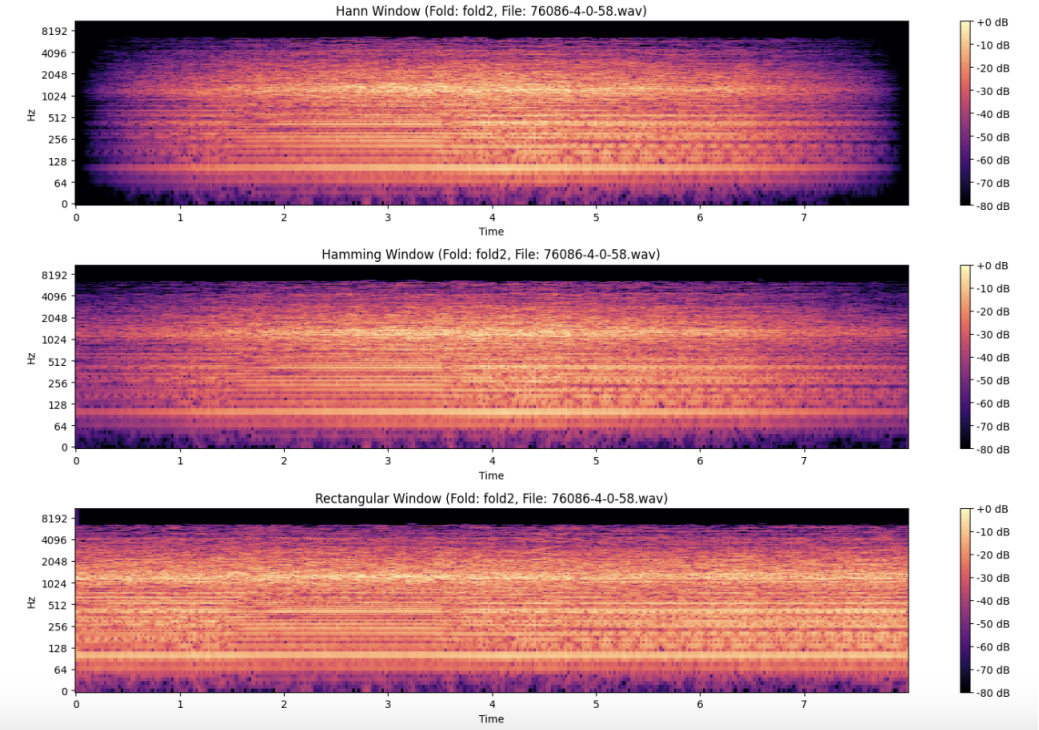
1. **Divide the signal** into short overlapping windows.
2. **Apply a window function** (Hann, Hamming or Rectangular) to each segment.
3. **Compute the Fourier Transform** for each segment to get frequency components.
4. Create a 2D plot:
    - **X-axis (horizontal)** $\rightarrow$ Time
    - **Y-axis (vertical)** $\rightarrow$ Frequency
    - **Color intensity** $\rightarrow$ Magnitude of frequencies

4.1.1 Fold 1 : Spectrogram



- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- Each spectrogram has a frequency resolution of approximately **8.28 Hz per pixel**.
- The total time duration in all spectrograms is **4.5 seconds**.
- Time resolution is **0.0033 seconds per pixel**, meaning fine time variations are captured.
- **Hann Window**: Smoother transitions, reduces spectral leakage, slightly lower intensity.
- **Hamming Window**: Similar to Hann but retains more sharpness in mid-frequencies.
- **Rectangular Window**: Higher intensity, but introduces more noise and artifacts.

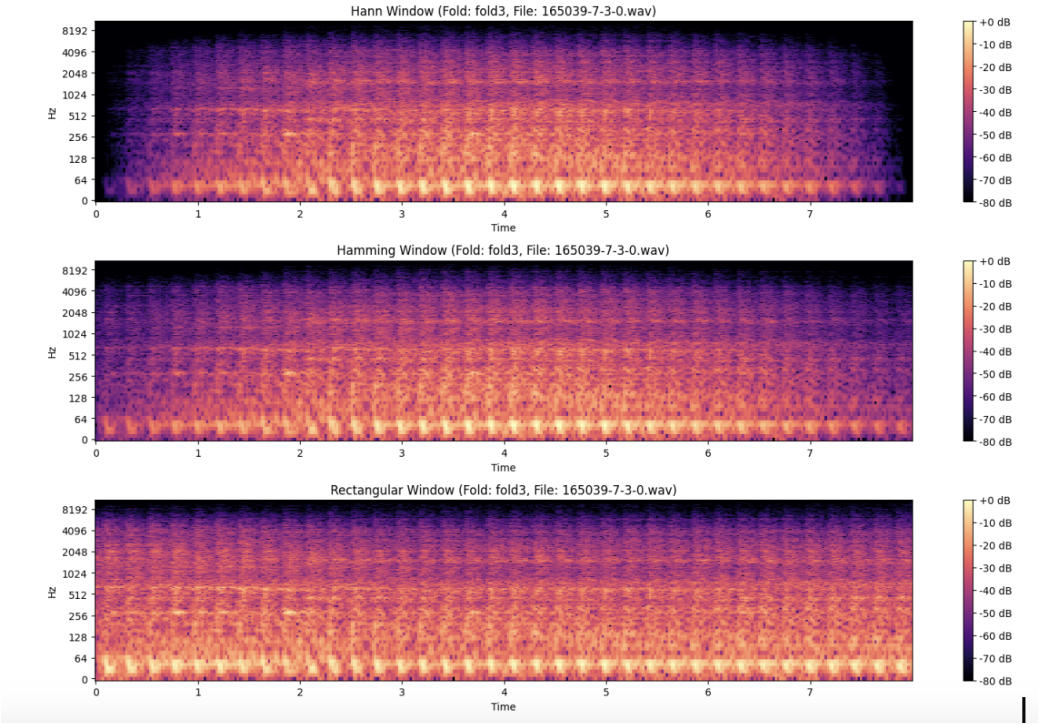
4.1.2 Fold 2 : Spectrogram



- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- Each spectrogram has a frequency resolution of approximately **8.28 Hz per pixel**.
- The total time duration in all spectrograms is **7.5 seconds**.
- Time resolution is **0.0033 seconds per pixel**, meaning fine time variations are captured.
- **Hann Window**: Smooth transitions, minimizes spectral leakage, reduces intensity slightly.
- **Hamming Window**: Similar to Hann but preserves more mid-frequency details.
- **Rectangular Window**: Strongest intensity but introduces more noise and spectral artifacts.

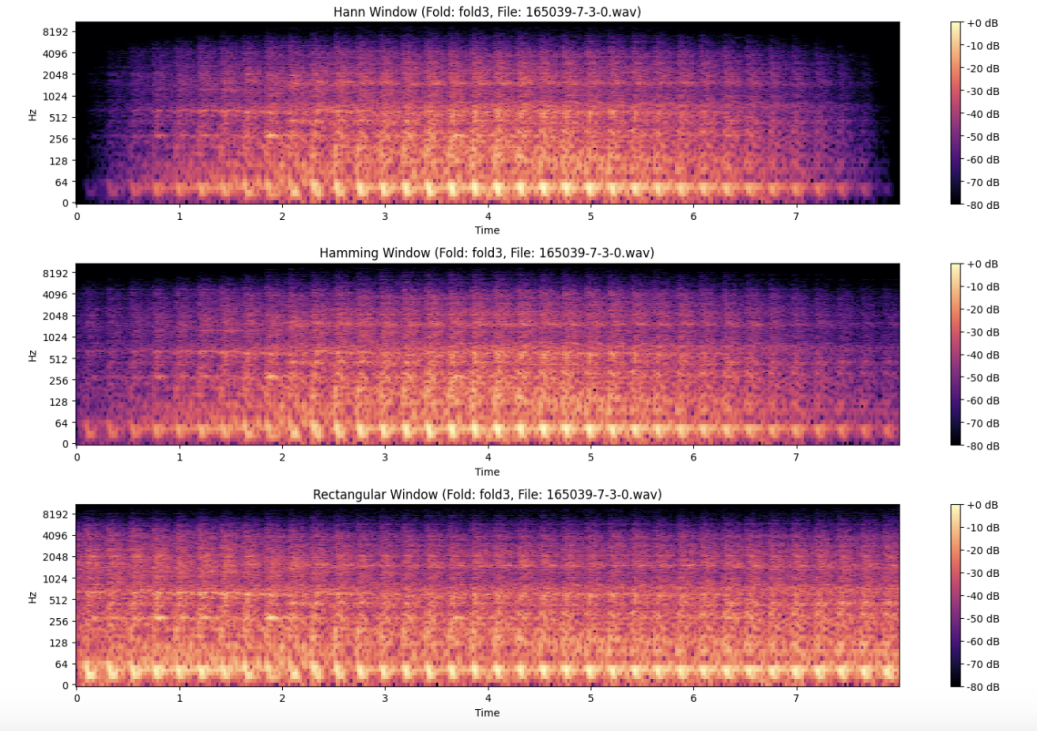
- For precise and clear analysis, **Hann or Hamming** is preferred over the **Rectangular** window.
- For a better trade-off between smoothness and detail, **Hann or Hamming** is preferred over **Rectangular**.

4.1.3 Fold 3 : Spectrogram



- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- Frequency resolution is **8.28 Hz per pixel**, ensuring fine frequency detail.
- The total time duration in all spectrograms is **7.5 seconds**.
- Time resolution is **0.0033 seconds per pixel**, which allows capturing rapid variations.
- **Hann Window**: Provides smooth transitions and minimizes spectral leakage, though it slightly reduces overall intensity.
- **Hamming Window**: Balances smoothness and detail retention, maintaining more mid-frequency information.
- **Rectangular Window**: Produces the highest intensity but introduces more spectral noise and artifacts.
- For optimal frequency analysis, **Hann or Hamming** is preferred over **Rectangular**.

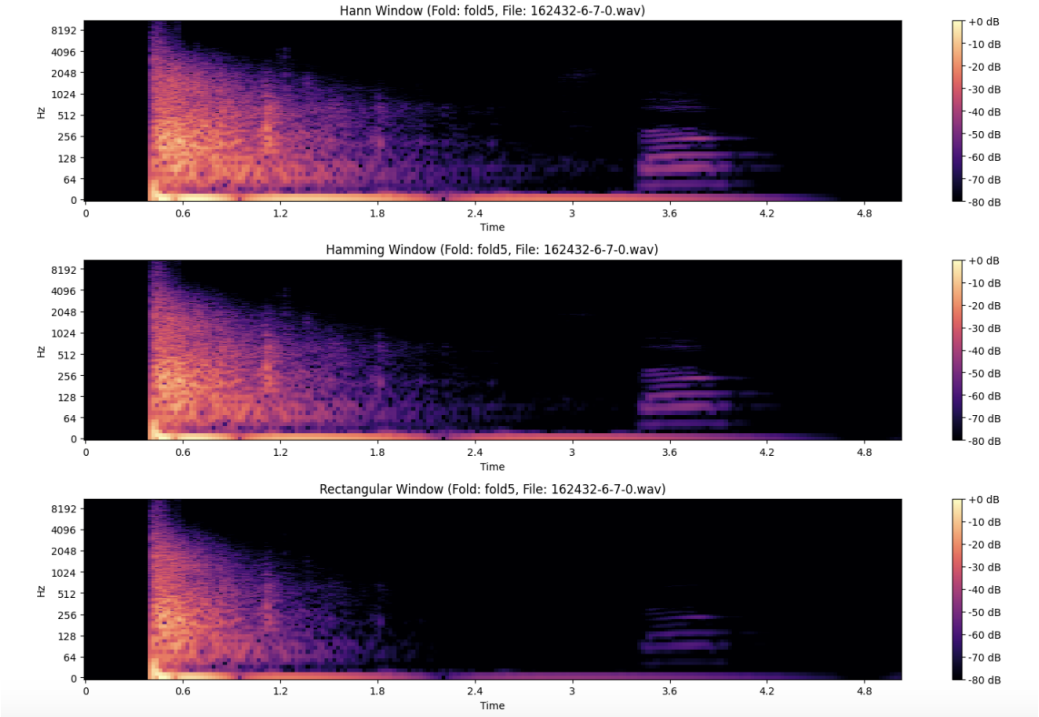
4.1.4 Fold 4 : Spectrogram



- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- Frequency resolution remains fine, allowing for detailed frequency analysis.
- The total time duration in all spectrograms is approximately **2.2 seconds**.
- Time resolution is high, capturing transient changes effectively.
- **Hann Window**: Smooth spectral transitions with minimized spectral leakage.
- **Hamming Window**: Slightly better intensity retention than Hann, while still reducing leakage.
- **Rectangular Window**: Produces the strongest intensity but introduces spectral artifacts and noise.
- The signal energy is concentrated in the **lower frequencies (below 1 kHz)**.
- Vertical patterns indicate periodic changes in amplitude.

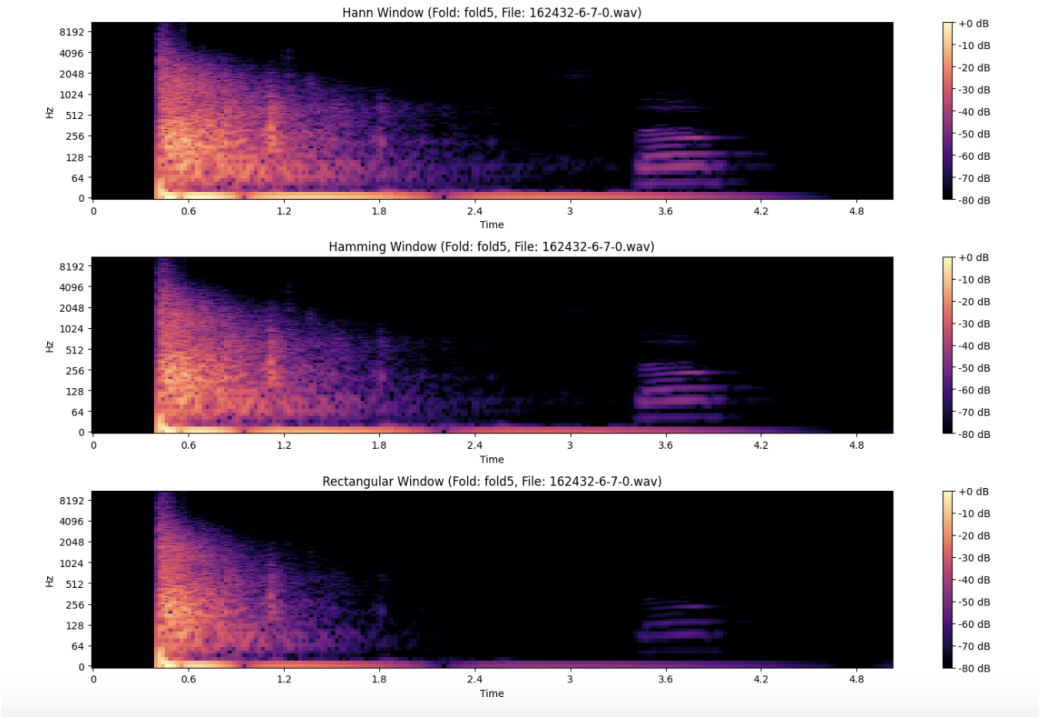
- The **Rectangular Window** has the most visible noise, while **Hann and Hamming** provide cleaner representations.

4.1.5 Fold 5 : Spectrogram



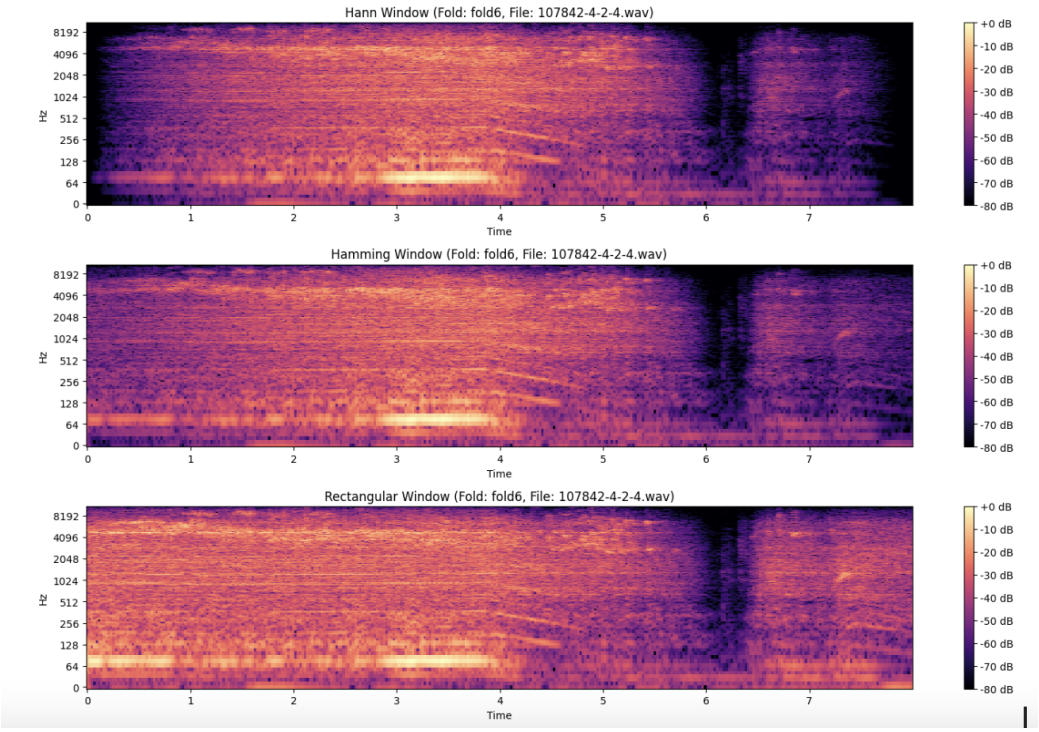
- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- They all show clear details in the frequency patterns.
- The total time duration of the signal is about **4.8 seconds**.
- The spectrograms capture quick changes in the sound over time.
- **Hann Window**: Smooth changes in sound with less unwanted noise.
- **Hamming Window**: Keeps sound details a bit better than Hann, with low noise.
- **Rectangular Window**: Shows the loudest sounds but adds more noise and unwanted patterns.
- Most of the sound energy is in the lower frequencies (below 1 kHz).
- Vertical lines show repeating changes in the sound.
- The Rectangular Window has the most noise, while Hann and Hamming show clearer sounds.

4.1.6 Fold 6 : Spectrogram



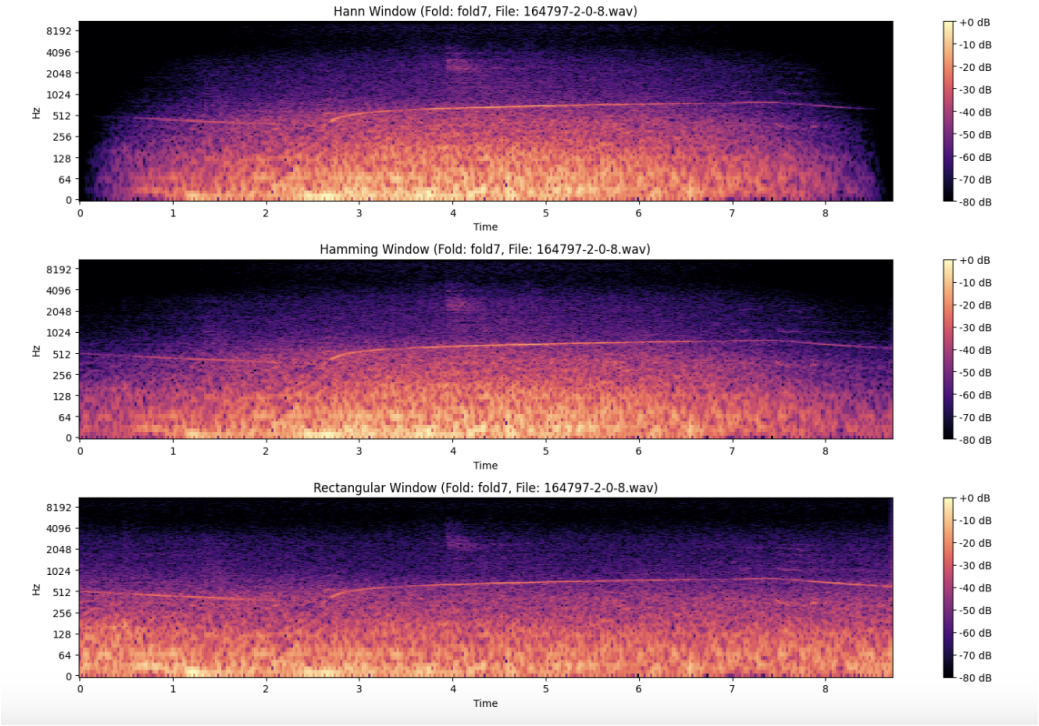
- All spectrograms cover a frequency range from **0 Hz to 8192 Hz**.
- They all show clear details in the frequency patterns.
- The total time duration of the signal is about **4.8 seconds**.
- The spectrograms capture quick changes in the sound over time.
- **Hann Window**: Provides smooth spectral transitions with reduced background noise.
- **Hamming Window**: Offers a good balance between detail retention and noise suppression.
- **Rectangular Window**: Displays higher intensity but introduces more spectral leakage and noise.

4.1.7 Fold 7 : Spectrogram



- Also spans 0 Hz to 8192 Hz.
- Maintains detailed frequency patterns similar to Fold6.
- Covers approximately 4.8 seconds.
- Effectively captures transient changes in the audio signal.
- **Hann Window**: Ensures smooth spectral representation with minimal noise artifacts.
- **Hamming Window**: Retains important sound details while keeping noise low.
- **Rectangular Window**: Emphasizes louder components but at the cost of increased noise and spectral leakage.

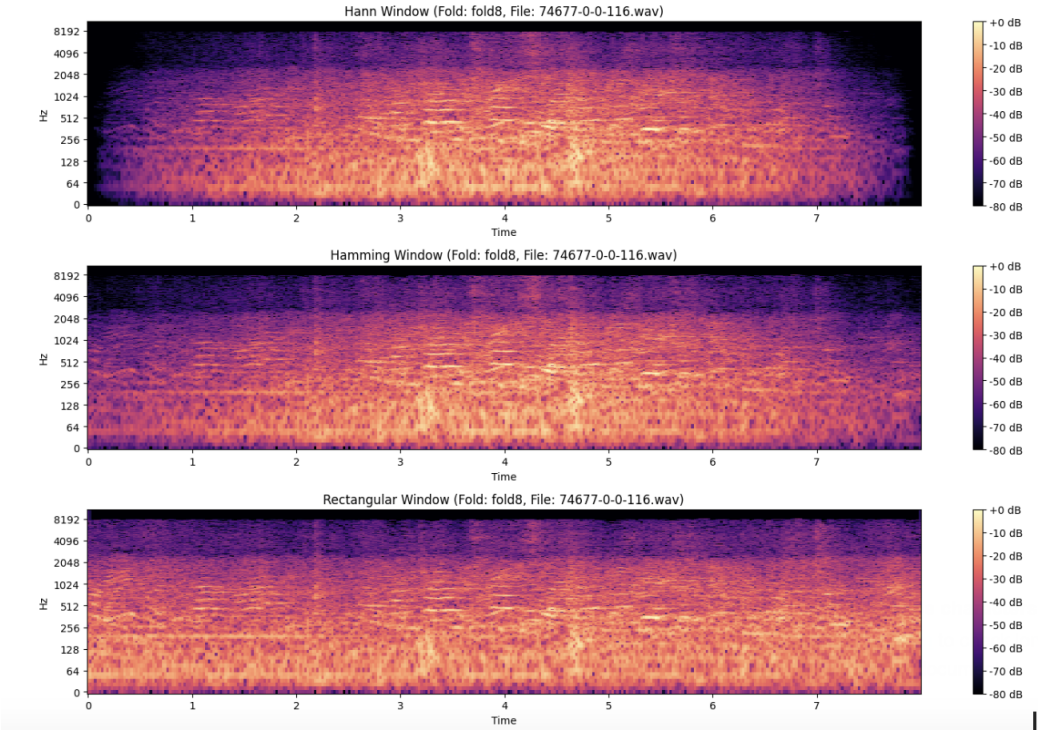
4.1.8 Fold 8 : Spectrogram



- All spectrograms show frequencies from **0 Hz to 8192 Hz**.
- The frequency details are clear in all three cases.
- The spectrograms cover a total time of about **7 seconds**.
- They show changes over time clearly and with good detail.
- The signal energy is predominantly concentrated in the **lower frequency range (below 1 kHz)**, as indicated by the higher intensity in these regions.
- Vertical patterns in the spectrograms suggest periodic changes in the signal's amplitude or modulated frequencies.
- The **Rectangular Window** exhibits the most visible noise and artifacts, making it less suitable for accurate spectral analysis.
- **Hann Window**: delivers the cleanest output, followed by the **Hamming Window**.

- **Energy Focus**: Most of the energy is in lower frequencies, especially below **1 kHz**.
- **Patterns Over Time:** Vertical lines show periodic changes in the signal.
- **Noise:** The Rectangular Window has the most noise and artifacts, while the Hann and Hamming windows give cleaner results.

4.1.9 Fold 9 : Spectrogram

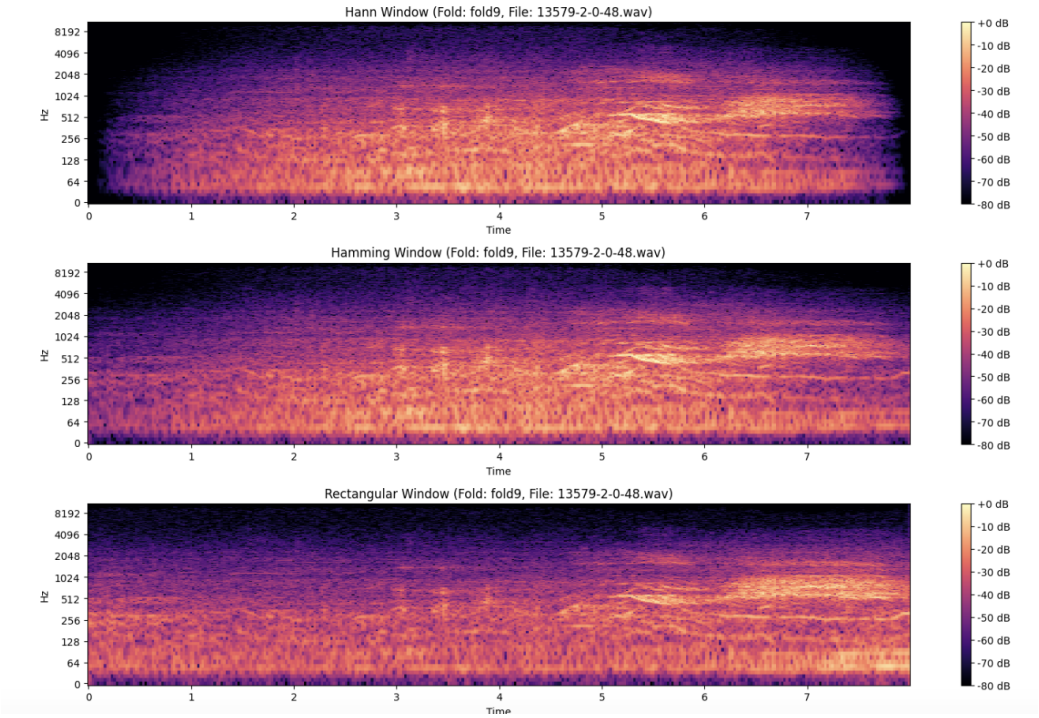


- All spectrograms show frequencies from **0 Hz to 8192 Hz**.
- Frequency details are well-resolved in all three cases, showing clear variations across the range.
- The total signal duration is approximately **7 seconds**.
- The spectrograms capture the signal's time-dependent changes clearly.
- **Hann Window:** Best for reducing noise and ensuring a smooth representation of the signal.
- **Hamming Window:** Offers a balance between strong intensity and smoothness, making it a versatile option.
- **Hann Window:** Shows higher intensity but is less suitable for analysis due to more noise and artifacts.
- The **Rectangular Window:**

  - Shows the strongest intensity of all three techniques.
  - However, it introduces noticeable noise and artifacts, making the spectrogram less clean.

- **Energy Focus:** Most of the energy is concentrated in lower frequencies, primarily below **1 kHz**, with some activity extending beyond.
- **Periodic Patterns:** Vertical variations in intensity indicate periodic changes in the signal's characteristics over time.

4.1.10 Fold 10 : Spectrogram



- All spectrograms show frequencies from **0 Hz to 8192 Hz**.
- Frequency details are clearly visible, showing good resolution.
- The signal duration spans approximately **8 seconds**.
- The spectrograms capture changes over time effectively, allowing for clear time-frequency analysis.
- **Energy Focus:** The energy is concentrated in lower frequencies, particularly below **1 kHz**, with noticeable harmonic structures at higher frequencies.
- **Harmonic Patterns:** Horizontal lines in the spectrograms represent harmonics of the signal.
- **Noise Levels:**
  - The **Rectangular Window** exhibits more noise and less-defined harmonic structures.

  - The **Hann and Hamming Windows** produce cleaner and more defined spectrograms.

- **Hann Window:** Best for smooth and noise-free spectral analysis.
- **Hamming Window:** A good compromise between smoothness and intensity, providing clear harmonics.
- **Rectangular Window:** Stronger intensity but less suitable due to increased noise and artifacts.

5 Training a SVM Using Spectrogram Features

We train a simple classifier (SVM) using features extracted from spectrograms of audio signals. The performance of the classifier is evaluated for different windowing techniques (Hann, Hamming, Rectangular), and the results are compared to determine the most effective technique.

5.1 Methodology

5.1.1 Dataset:

- The UrbanSound8K dataset is used, which contains audio files from 10 different classes
- The dataset is divided into 10 folds, and a subset of the data is used for training and testing.

5.1.2 Feature Extraction:

- Spectrograms are generated using the Short-Time Fourier Transform (STFT) for each audio file.
- Three windowing techniques are applied: Hann, Hamming, and Rectangular windows.
- Features are extracted from the spectrograms by computing the mean and standard deviation of the frequency bins.

5.1.3 Classifier:

- A Support Vector Machine (SVM) with a linear kernel is used as the classifier.
- The dataset is split into training and testing sets (80
- The classifier is trained and evaluated for each windowing technique.

5.1.4 Evaluation Metrics:

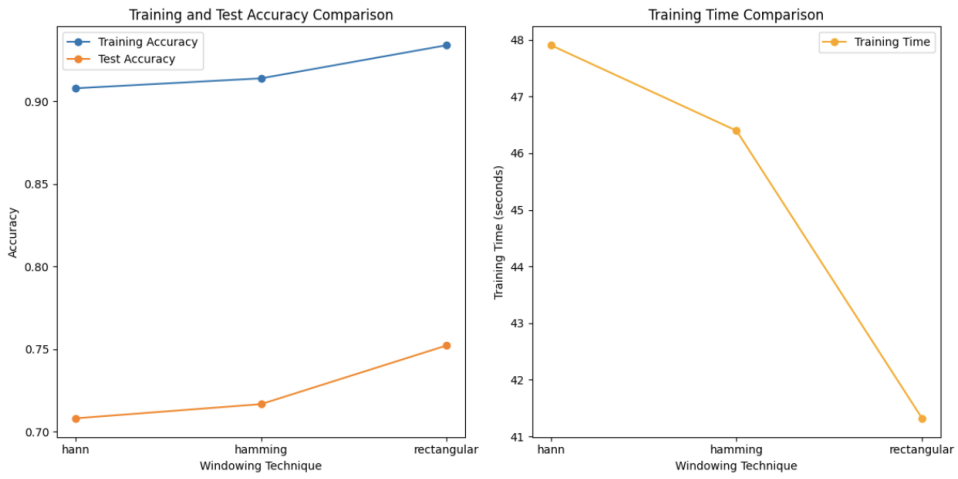
- Accuracy: Measures the percentage of correctly classified samples.
- Classification Report: Provides precision, recall, and F1-score for each class.

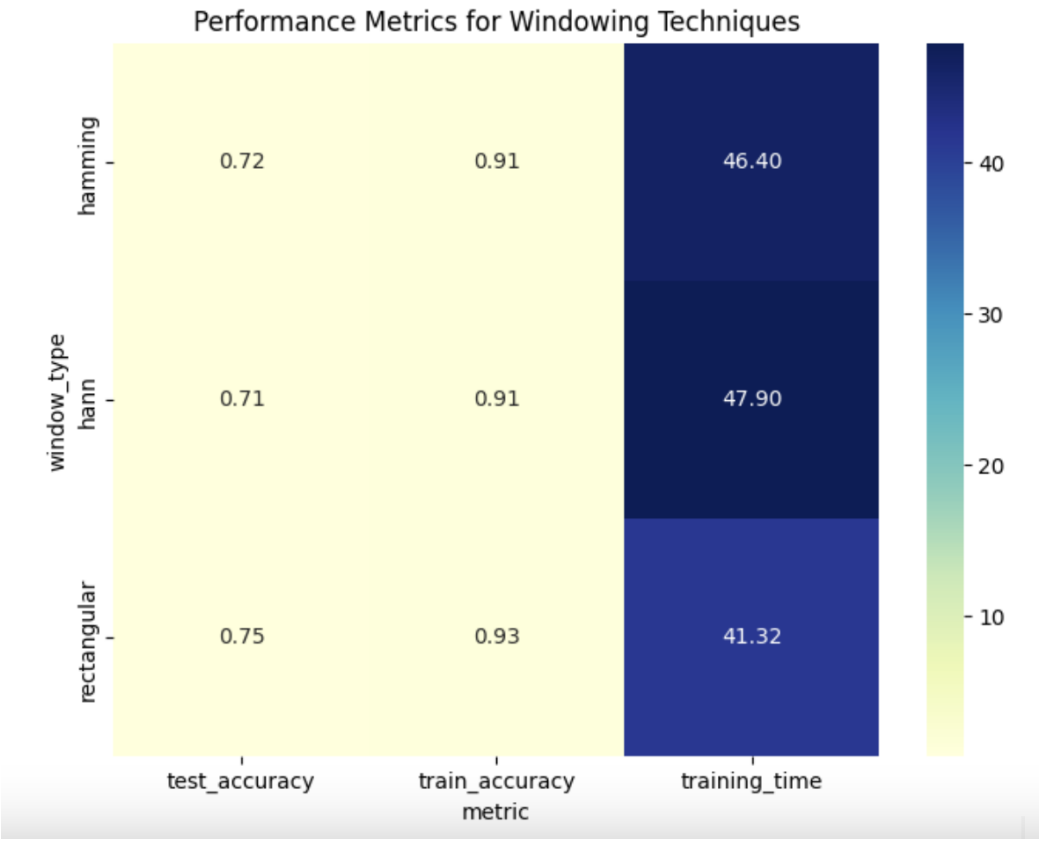
- Confusion Matrix: Visualizes the performance of the classifier for each class.
- Training Time: Measures the time taken to train the classifier.

5.2 Results

5.2.1 Performance Comparison of Windowing Techniques

Metric	Hann Window	Hamming Window	Rectangular Window
Training Accuracy	0.91	0.91	0.93
Test Accuracy	0.71	0.72	0.75
Training Time (s)	47.90	46.40	41.32
Precision (Macro Avg)	0.72	0.73	0.77
Recall (Macro Avg)	0.69	0.71	0.76
F1-Score (Macro Avg)	0.70	0.71	0.76
Precision (Weighted)	0.72	0.73	0.76
Recall (Weighted)	0.71	0.72	0.75
F1-Score (Weighted)	0.71	0.72	0.75





5.2.2 Key Observations from the Table

1. Test Accuracy:
    - Rectangular Window achieved the highest test accuracy (0.75), followed by Hamming Window (0.72) and Hann Window (0.71).
2. Training Time:
    - Rectangular Window was the fastest to train (41.32 seconds), while Hann Window took the longest (47.90 seconds).
3. Precision, Recall, and F1-Score:
    - The Rectangular Window consistently outperformed the other two techniques in terms of precision, recall, and F1-score (both macro and weighted averages).
4. Overall Performance:

- The Rectangular Window provided the best balance between accuracy, training time, and classification performance.

5. Confusion Matrix:

   - The confusion matrices for all three windowing techniques showed that the classifier performed well for most classes but struggled with some (class 3 and class 9).

5.3 Conclusion

- The Rectangular window outperformed the Hann and Hamming windows in terms of test accuracy, training time, and classification performance.
- Despite its simplicity, the Rectangular window provided the best balance between accuracy and computational efficiency for this task.
- The Hamming window performed slightly better than the Hann window, but both were less effective than the Rectangular window.

6 Code Repository

GitHub Repository URL

References

[1] Audio classification using spectrograms

[2] UrbanSound8K - Classification

[3] UrbanSound Classification with Pytorch and Fun

[4] Latex Documentation - Overleaf

[5] Windowing (signal processing)

[6] Understanding spectrograms