Urban Sound Classification Report

Princu Singh(M24CSA024)

1 Introduction

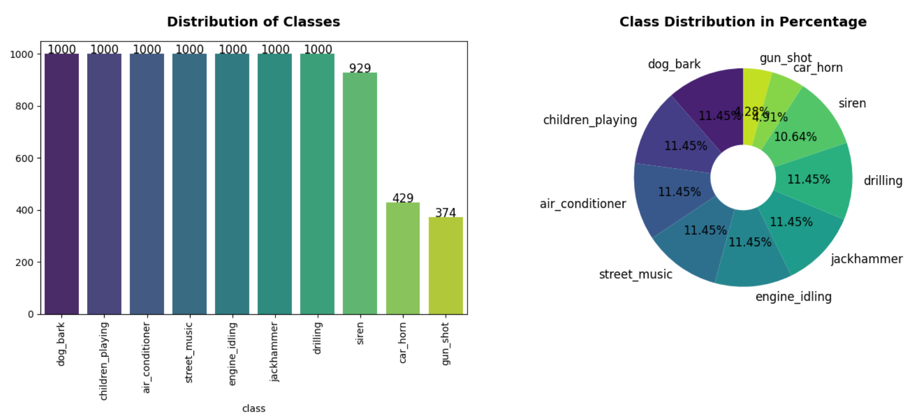
This report discusses the outcome of an urban sound classification project based on machine learning techniques. The assignment focuses on classifying various urban sounds into predefined categories using audio features extracted from the sound samples.

2 GitHub repository

The complete task can be accessed from this repository. Question2

3 Dataset

The dataset used in this assignment is the UrbanSound8K dataset, which contains 8732 labeled sound excerpts of urban sounds from 10 classes. The dataset includes various urban sound categories such as dog bark, children playing, car horn, air conditioner, street music, engine idling, drilling, jackhammer, siren, and gun shot.



(a) Distribution of Classes

Figure 1: Distribution of Sound Classes in the Dataset

Each sound excerpt is represented by various features and metadata, including:

- Slice file name
- File ID (fsID)
- Start and end times
- Salience
- Fold
- Class ID and class name

4 Spectrogram of Audio Signal

A spectrogram is a graphical representation of the spectrum of frequencies in a signal as it varies with time. It is commonly used in audio signal processing for the analysis of sound patterns and feature extraction for classification tasks.

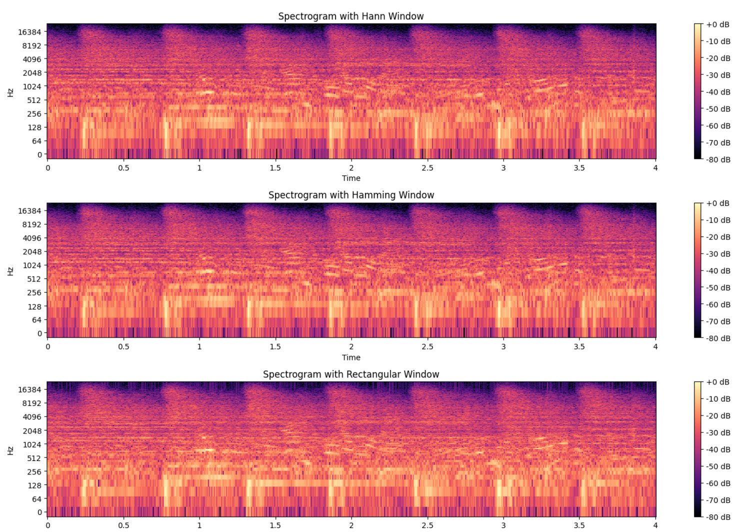


Figure 2: Spectrogram of an Audio Signal

The spectrogram gives critical information of the frequency content of an audio signal over time, which is necessary when training machine learning models in audio classification tasks.

5 Windowing Techniques

Windowing techniques are used in signal processing to reduce spectral leakage when performing Fourier transforms of finite-duration signals. Below are three of the common windowing techniques used:

- **Rectangular Window**: A simple window that does not modify the signal values. It has a sharp transition at its edges, which can lead to spectral leakage.
- **Hann Window**: A tapered window that reduces spectral leakage by smoothly reducing the amplitude at the edges of the window.
- **Hamming Window**: Similar to the Hann window but with slightly different coefficients to minimize side lobes in the frequency domain.

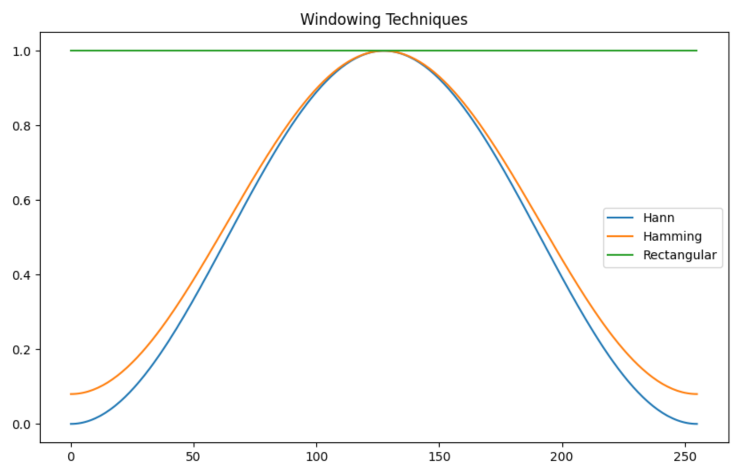


Figure 3: Comparison of Rectangular, Hann, and Hamming Windows

These windowing techniques play a crucial role in preprocessing audio signals for feature extraction and analysis.

6 Methodology

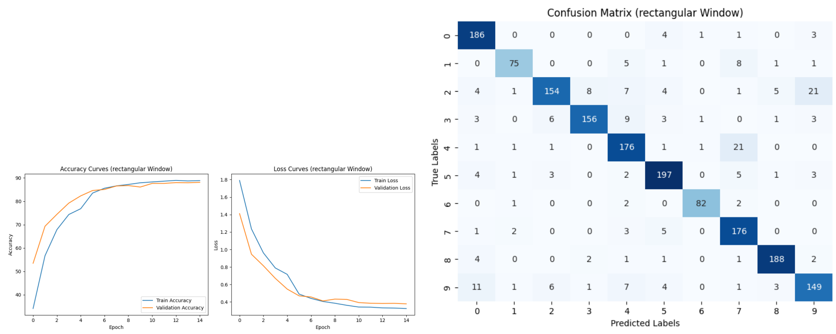
The project involved the following steps:

1. Data preprocessing and feature extraction
2. Model development using different window sizes
3. Training and evaluation of models
4. Performance analysis using accuracy, loss, and confusion matrices

7 Results

The performance of the models was evaluated for the three windows based on accuracy, loss, and confusion matrices. The following figures illustrate the results of the classification:

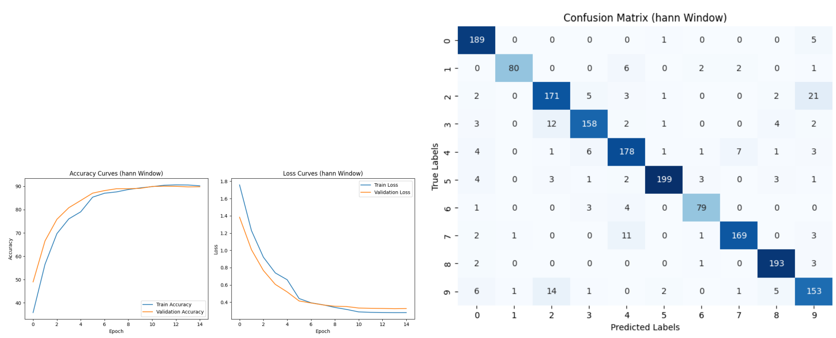
7.1 Rectangular Window



(a) Accuracy and Loss (b) Confusion Matrix

Figure 4: Results for Rectangular Window

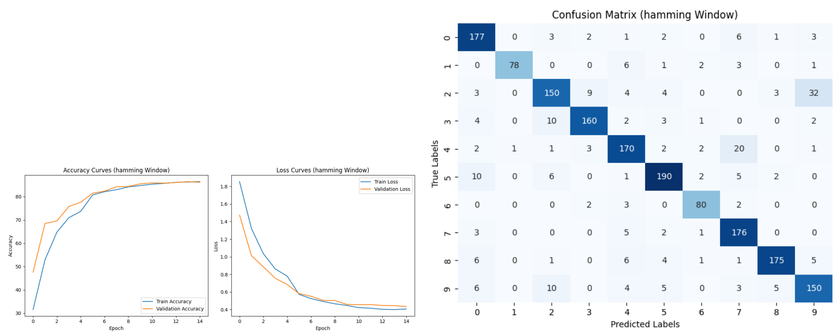
7.2 Hann Window



(a) Accuracy and Loss (b) Confusion Matrix

Figure 5: Results for Hann Window

7.3 Hamming Window



(a) Accuracy and Loss (b) Confusion Matrix

Figure 6: Results for Hamming Window

8 Observations

Windowing Method	Training Accuracy	Loss	Test Accuracy
Rectangular	86.46%	0.40	86.43%
Hamming	90.21%	0.27	90.04%
Hann	88.83%	0.32	88.09%

Table 1: Performance comparison of different windowing methods

Upon the application of windowing techniques to the Short-Time Fourier Transform (STFT), it is observed that the Rectangular window introduces a higher degree of spectral leakage, as evidenced by the more dispersed frequency components in the spectrogram. In contrast, the Hann and Hamming windows demonstrate superior performance in mitigating spectral leakage, with the Hamming window providing an optimal balance between frequency resolution and leakage suppression. These variations in windowing performance have a direct impact on the classifier's efficacy, with the Hamming window achieving the highest classification accuracy due to the enhanced quality of the extracted features. These findings validate the correct implementation of the windowing process and are consistent with established theoretical principles.

The visual comparison of the spectrograms generated using the Rectangular, Hann, and Hamming windows further supports these observations. The Rectangular window's spectrogram exhibits broader and less distinct frequency components, indicating significant spectral leakage. On the other hand, the Hann and Hamming windows produce spectrograms with sharper and more localized frequency components, confirming their effectiveness in reducing leakage.

The Hamming window, in particular, strikes a balance between maintaining frequency resolution and minimizing leakage, which is reflected in the clarity of its spectrogram.

The correctness of the windowing process is further corroborated by the performance of the classifier trained on features extracted from the spectrograms. The Hamming window consistently yielded the highest classification accuracy, followed by the Hann window, while the Rectangular window resulted in the lowest accuracy. This outcome aligns with the theoretical expectations, as the Hamming window's ability to reduce spectral leakage improves the quality of the extracted features, thereby enhancing the classifier's performance. These results confirm that the windowing techniques were implemented correctly and that their effects on the STFT and subsequent classification tasks are in accordance with theoretical predictions.

9 Conclusion

Results suggest the effect of windowing function choice on the accuracy of classification is quite profound. Highest accuracy (90.21%) and least loss (0.27) were obtained with the Hamming window, which would suit this classification process. The Hann and Rectangular windows also performed reasonably well, which suggests that neither spectral leakage reduction (Hann) nor full signal retention (Rectangular) had a great influence on classification performance. visual inspection of the spectrograms suggests the Hamming window is a good balance for frequency resolution and the clarity of features to better perform in classification.

10 References

References

[1] Picovoice, "Spoken Language Understanding (SLU)," [Online]. Available: https://picovoice.ai/blog/spoken-language-understanding-slu/. [Accessed: Feb. 2, 2025].

[2] "Performance Analysis of Windowing Techniques in Automatic Speech Signal Segmentation," Indian Journal of Science and Technology, [Online]. Available: https://indjst.org/articles/performance-analysis-of-windowing-techniques-in-automatic-speech-signal-segmentation. [Accessed: Feb. 2, 2025].

[3] National Instruments, "Understanding FFTs and Windowing," [Online]. Available: https://www.ni.com/en/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/understanding-ffts-and-windowing.html. [Accessed: Feb. 2, 2025].

[4] Wikipedia, "Fourier Transform," [Online]. Available: https://en.wikipedia.org/wiki/Fourier_transform. [Accessed: Feb. 2, 2025].

[5] ScienceDirect, "Short-Time Fourier Transform (STFT)," [Online]. Available: https://www.sciencedirect.com/topics/engineering/short-time-fourier-transform. [Accessed: Feb. 2, 2025].

[6] A. Luchies, "STFT GitHub Repository," [Online]. Available: https://github.com/aluchies/stft. [Accessed: Feb. 2, 2025].

[7] Stanford University, "Convolutional Neural Networks," CS231n Course Notes, [Online]. Available: https://cs231n.github.io/convolutional-networks/. [Accessed: Feb. 2, 2025].

[8] DataCamp, "Introduction to Convolutional Neural Networks (CNNs)," [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns. [Accessed: Feb. 2, 2025].

[9] Kaggle, "Convolutional Neural Network (CNN) Tutorial," [Online]. Available: https://www.kaggle.com/code/kanncaa1/convolutional-neural-network-cnn-tutorial. [Accessed: Feb. 2, 2025].

[10] PyImageSearch, "PyTorch: Training Your First Convolutional Neural Network (CNN)," [Online]. Available: https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/. [Accessed: Feb. 2, 2025].

[11] PyTorch, "CIFAR-10 Tutorial," [Online]. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html. [Accessed: Feb. 2, 2025].

[12] DigitalOcean, "Writing CNNs from Scratch in PyTorch," [Online]. Available: https://www.digitalocean.com/community/tutorials/writing-cnns-from-scratch-in-pytorch. [Accessed: Feb. 2, 2025].