Speech Assignment - Report

Rohan Frederick
(M23CSA525)

2nd Feb 2025

1 Objective

1.1 Task A

Understand and implement the following windowing techniques:

- Hann Window
- Hamming Window
- Rectangular Window

Apply the above windowing techniques and generate spectrograms using the Short-Time Fourier Transform (STFT). Train a simple classifier (e.g., SVM or neural network) using features extracted from the spectrograms and evaluate the performance results comparatively in different techniques.

1.2 Task B

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

2 Data Analysis

The audio sample rate and time for each sample vary as follows.

Time (in sec)	Count
4	7459
2	525
3	281
1	459
5	8

Table 1: Time Duration of audio samples

Sample Rate	Count
48000	2502
44100	5370
96000	610
11025	39
24000	82
8000	12
16000	45
22050	44
32000	4
192000	17
11024	7

Table 2: Sample Rate Distribution

Class	Count
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

Table 3: Audio Class Distribution

The classes are not balanced and the techniques that can be used to create a class balance are:

- Randomly duplicate a few samples.
- Duplicate after applying a band-pass filter to filter desired frequency.
- Reduce the sampling rate and apply FFT with a window size of half the sampling rate followed by IFFT to get the new signal (risk of aliasing).

Note: However the Techniques to create a class balance were not implemented.

3 Windowing

- The window size for comparison of different windowing techniques is 1024. This reduces our frequency resolution to a maximum of 512 Hz (Nyquist frequency), giving a total of 512 + 1 (direct current) frequency points. A

- standard 50% overlap was considered for comparison between different windowing techniques
- The time duration represented by the window size depends on the sampling rate.

Sample Rate	Time in ms
48000	21
44100	23
96000	11
11025	93
24000	43
8000	128
16000	64
22050	46
32000	32
192000	5
11024	93

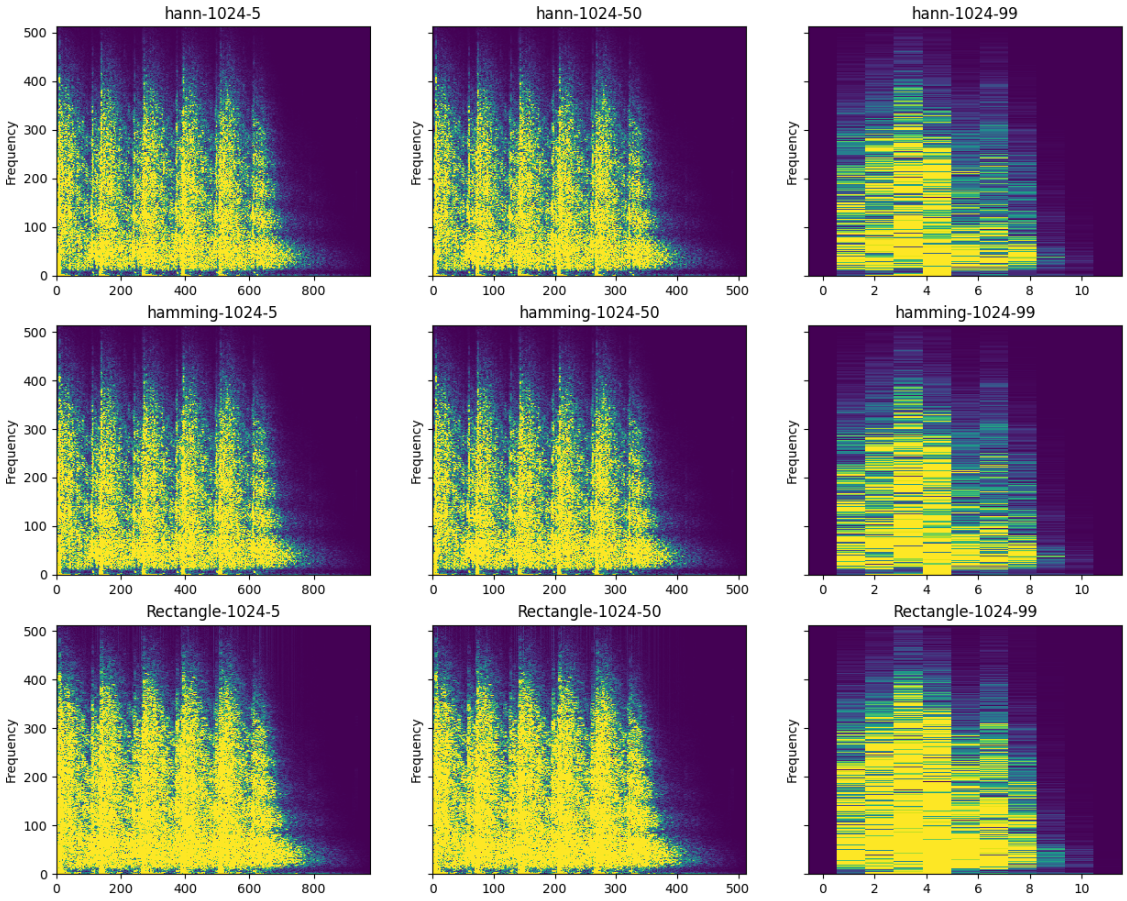
Table 4: Sample Rate vs. Time

The effects of windowing as we change the windowing types and overlap can be seen below:-
Where hann-1024-5 is:-

- Window type: Hann
- Window Size : 1024
- Hop Size :

$$\frac{5 * 1024}{100} = 51 samplepoints \tag{1}$$

(low number high overlap)



The frequency resolution is fixed as the window size is 1024 in all cases for an audio with sample rate as 44100 and data length as 99225 samples.

$$N = \left\lfloor \frac{L - W}{2H} \right\rfloor + 1 \tag{2}$$

Note - we are dividing by 2H as we are only plotting till the Nyquist Frequency.

The resolution of time varies as the hop size differs; for example, for 5% overlap, the hop size = 51 samples.

Note: Due to time and frequency trade off the decision was to either fix the frequency resolution or the time resolution. In order that the comparison is like to like the frequency resolution was given precedence.

4 Training Details – Approach

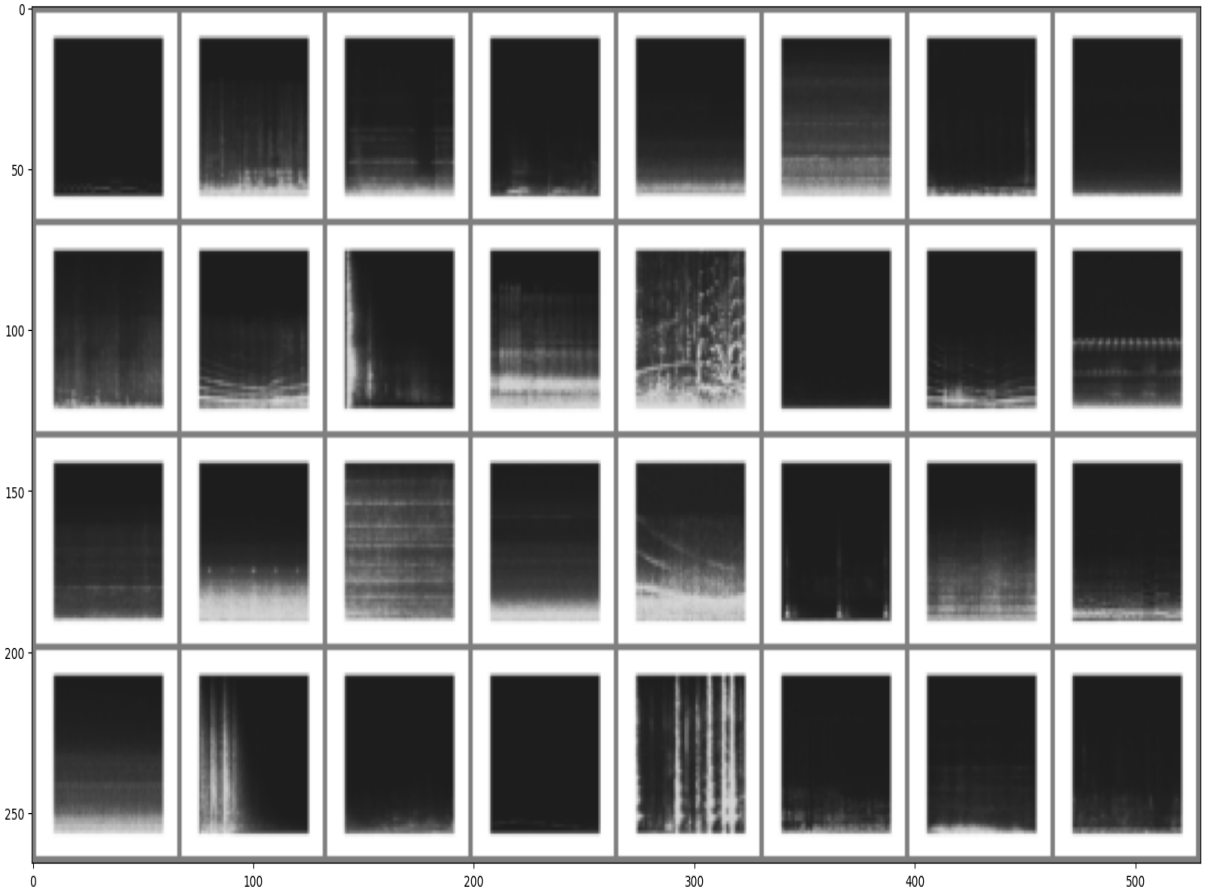
- All audio files are transformed into **spectrograms in grayscale** and saved as images.

- To read the files in batches a custom torch dataset class is designed that can vary the windowing parameters. The class also allows for **manual windowing** techniques.
- The images are **resized into 64 X 64** resolution
- Pytorch CNN class is defined with below architecture. Due to training time the architecture is not kept deep.

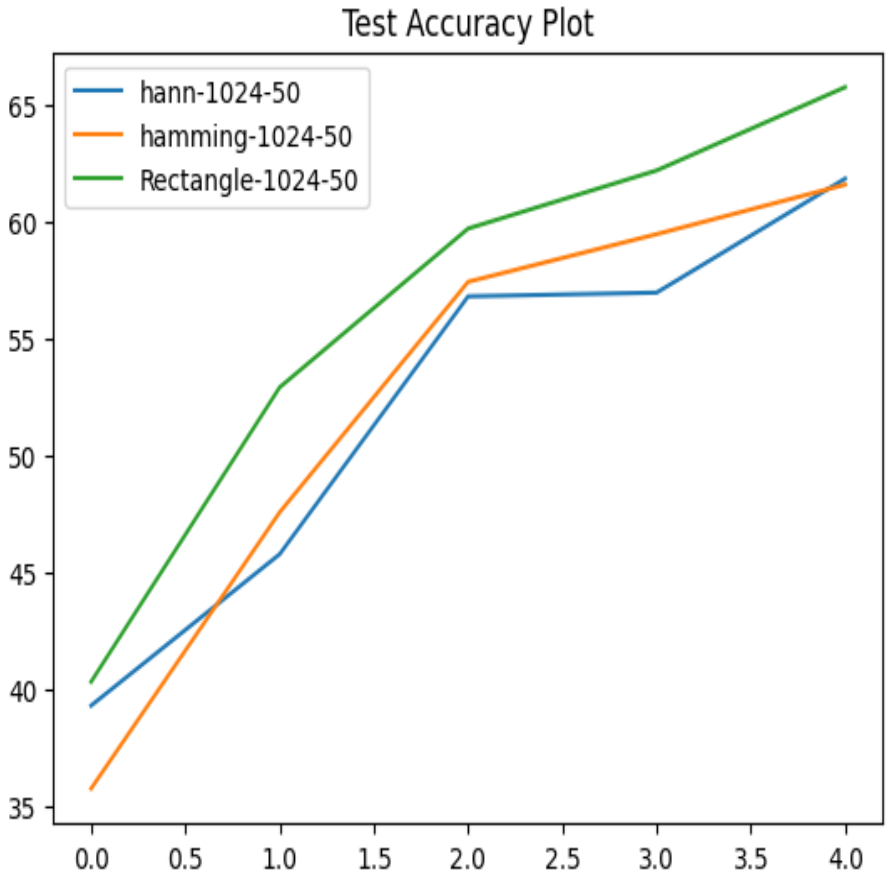
Layer (Type)	Output Shape	Param #
Conv2d-1	[-1, 32, 62, 62]	320
ReLU-2	[-1, 32, 62, 62]	0
MaxPool2d-3	[-1, 32, 31, 31]	0
Conv2d-4	[-1, 64, 29, 29]	18,496
ReLU-5	[-1, 64, 29, 29]	0
MaxPool2d-6	[-1, 64, 14, 14]	0
Linear-7	[-1, 128]	1,605,760
ReLU-8	[-1, 128]	0
Linear-9	[-1, 10]	1,290
Total Params		**1,625,866**
Trainable Params		**1,625,866**
Non-trainable Params		**0**
Metric	Size (MB)	
Input size	0.02	
Forward/backward pass size	3.03	
Params size	6.20	
Estimated Total Size	**9.25**	

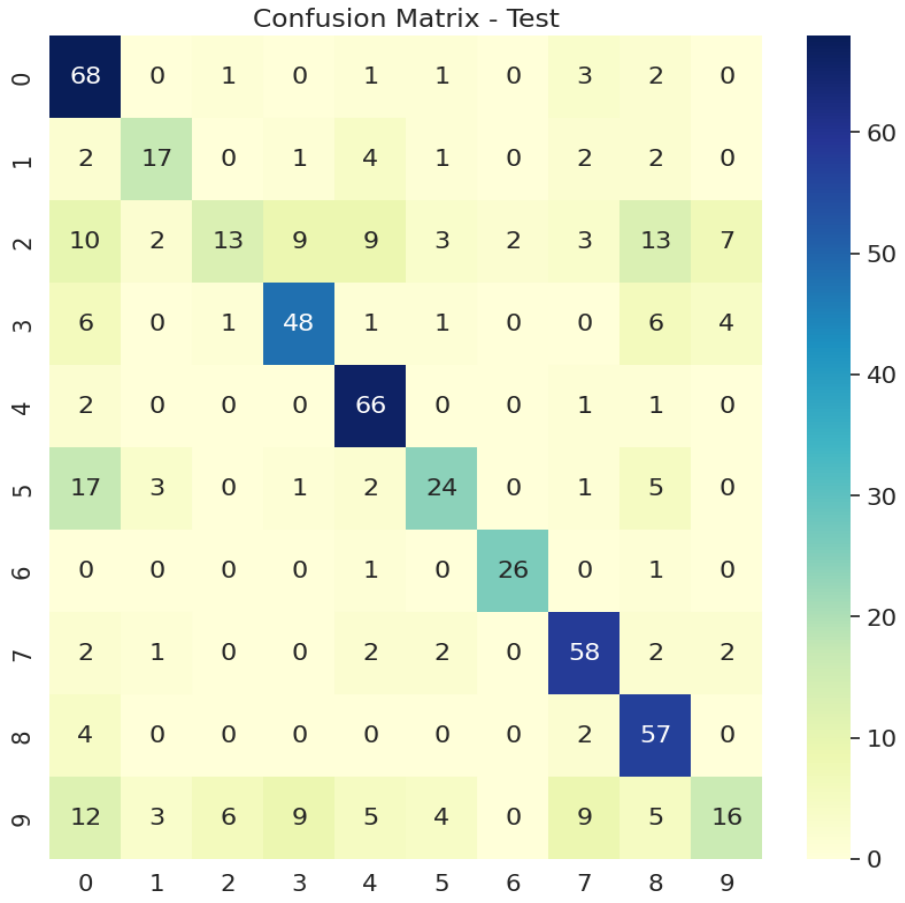
Table 5: Neural Network Architecture

- Due to long runtime, models were trained with only 5 epochs.
- A Sample of Training Data as below:


0
50
100
150
200
250
0
100
200
300
400
500

5 Training Results





5.1 Hypothesis

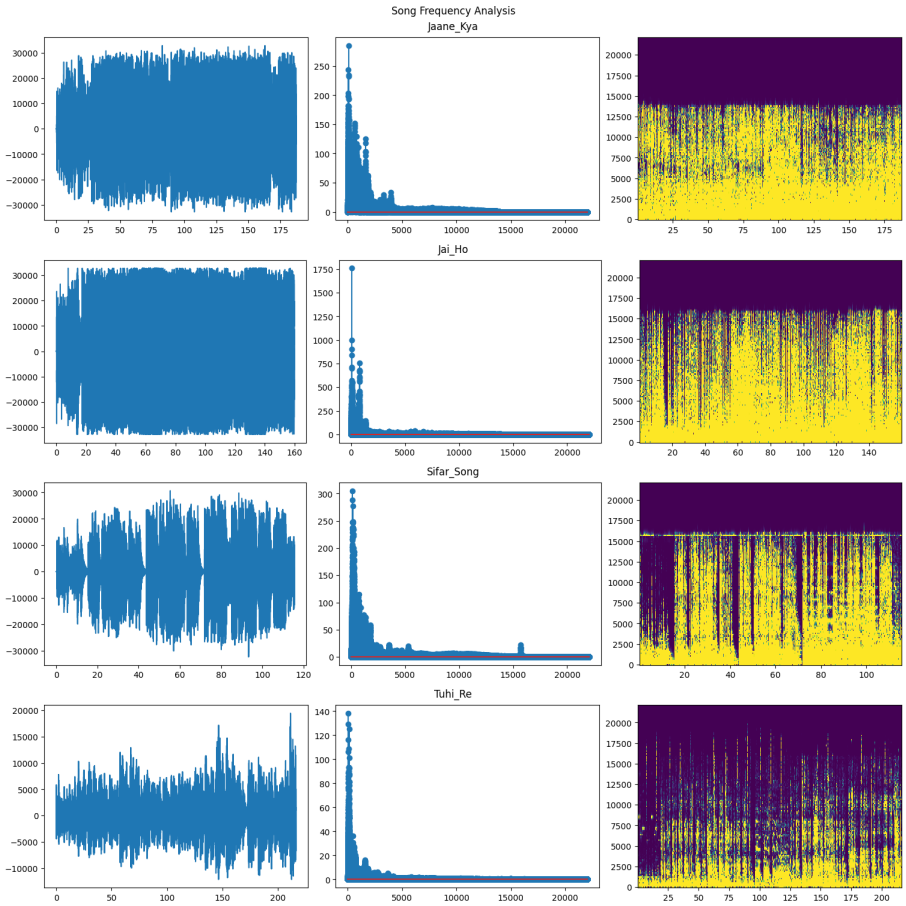
It is hypothesized that rectangle is giving higher accuracy as both hamming and Hanning have a smoothening effect on the signal. The Rectangular window causes more spectral leakage (which also creates a pseudo smoothening effect) however the spectral leakage is mitigated by the window overlap. Hence in case on rectangular window the jumps are more pronounced.

6 Task B: Song Analysis

The selected songs are:

- Jaane_Kya.wav
- Jai_Ho.wav
- Sifar_Song.wav
- Tuhi_Re.wav

6.1 Spectral analysis:



- **Jai Ho** has higher power with amplitude reaching up to 1750 units.
- **Tu Hi Re** has the highest frequency reaching up to 17500 Hz, while others peak around 15KHz.It has less bass (low frequency) than all other songs.
- Lower frequency elements generally have higher power across all songs.

7 References

- https://www.geeksforgeeks.org/hide-axis-borders-and-white-spaces-in-matplotlib/
- https://stackoverflow.com/questions/16120481/matplotlib-grayscale-heatmap-with-visuall
- https://pytorch.org/tutorials/beginner/data_loading_tutorial.
  html
- https://stackoverflow.com/questions/2148543/how-to-write-a-confusion-matrix-
- https://medium.com/data-science-in-your-pocket/calculating-precision-recall-for-multi-
- https://www.youtube.com/watch?v=er3LoYljvsU

8 Packages and Libraries

- numpy
- matplotlib
- scipy
- torch
- torchaudio
- pandas
- PIL
- torchvision
- pickle
- seaborn