Summer Fellowship Report

On

Scilab Case Studies

Submitted by

Aashwin Vaidya

Under the guidance of

Prof. Prabhu Ramachandran
Aerospace Engineering Department
IIT Bombay

Mentor:
Rashmi Patankar

July 8, 2026

Acknowledgment

I would like to sincerely thank the FOSSEE Team at IIT Bombay for granting me the valuable opportunity to participate in the FOSSEE Summer Fellowship 2026. My heartfelt appreciation goes to Prof. Prabhu Ramachandran for initiating this remarkable program that encourages learning, innovation, and open-source software development.

I am especially grateful to Mrs. Rashmi Patankar for her continuous guidance, valuable suggestions, and constant encouragement throughout the fellowship. Her support and technical expertise played a crucial role in the successful completion of this project and significantly enhanced my understanding of Scilab.

I would also like to express my sincere gratitude to the Institute of Engineering and Technology (IET), DAVV, Indore, for providing me with a strong academic foundation that helped me undertake and complete this work successfully.

Contents

Chapter 1

Introduction

The Free and Open Source Software in Education (FOSSEE) project, initiated by IIT Bombay, promotes the use of open-source software in education and research. Through activities such as workshops, internships, and fellowships, FOSSEE encourages students to learn and contribute to open-source technologies while solving practical engineering problems.

Scilab is an open-source numerical computing software used for scientific and engineering applications. It provides powerful tools for numerical computation, signal processing, data visualization, and GUI development. Its flexibility and ease of use make it a suitable platform for implementing engineering projects and exploring practical applications of digital signal processing.

Case Study 1: Music Genre Classifier Using Feed Forward Neural Network: Design and implementation of an acoustic feature extraction pipeline in Scilab to process raw audio files into structured feature vectors. The Feed Forward Neural Network model evaluated these extracted statistical characteristics to map and classify audio tracks into distinct musical genres with optimized processing efficiency.

Case Study 2: Single-Digit DTMF Signal Generator and Detector using Goertzel Algorithm in Xcos: Design and simulation of a telephonic signaling architecture using Scilab and Xcos, employing parallel IIR filter banks and the Goertzel algorithm for tone identification. The system achieved precise decoding of dual-tone inputs and robust signal isolation within the standard frequency grid.

Case Study 3: ECG Signal Processing and Heart Rate Frequency Detection: Design and evaluation of a digital filter cascade,including a biquad notch filter and a moving-average smoother to eliminate baseline wander and 50 Hz power line interference from patient data. The system successfully isolates QRS complexes and detects R-peaks to compute real-time heart rate (BPM) metrics.

Chapter 2

Music Genre Classifier Using Feed Forward Neural Network

2.1 Abstract

This project builds a multi-layer neural network to classify musical genres using 3-second clips from the GTZAN dataset. The model uses a csv file of 3-second audio clips which contain 57 features for every audio clip. This data is normalised using Z-score normalisation and fed into the neural network as matrices. The final architecture has a two-hidden-layer structure which uses Rectified Linear Unit (ReLU) activations, which kept learning signals sharp and eliminated gradient saturation. To prevent the model from getting stuck in local traps, standard gradient descent was replaced with a native mini-batch partitioner and an adaptive Adam optimization loop. This approach successfully stabilized training, dropping cross-entropy loss to near-zero and increasing testing accuracy to 88.09 %. The model's success was visually verified using an Epoch vs Categorical Cross Entropy Loss graph and confusion matrix heatmap. The project is inspired by the music genre classifier using convolutional neural networks published by Qiuqiang Kong, Xiaohui feng and Yanxiong Li.

2.2 Problem Statement

Building a high-performing audio classifier natively in Scilab presents severe data and structural challenges. First, deep learning models require a massive volume of data to generalize effectively, but standard music collections like the GTZAN dataset offer only 1,000 baseline audio tracks. Second, raw audio exists as a complex 1D time-domain signal, a long sequence of amplitude waves that standard neural networks cannot easily process. To build an optimized solution, these 30-second tracks must be pre-processed and systematically sliced into 3- second segments to expand the data pool into 10,000 distinct samples. Furthermore, the raw signal must be transformed into static multi-feature acoustic vectors so that mathematical gradients can be calculated stably. To achieve this, the project implements a multi-layer neural network through matrix linear algebra. The data matrix is first normalized using Z-score calculation to eliminate feature scale disparities. The system then routes a 57-feature input vector through a twohidden-layer network using Rectified Linear Unit (ReLU) activations to prevent gradient saturation. To improve convergence speed and prevent the network from getting trapped in local minima, standard gradient descent is replaced with a minibatch partitioner with an adaptive Adam optimization loop.

2.3 Flowchart

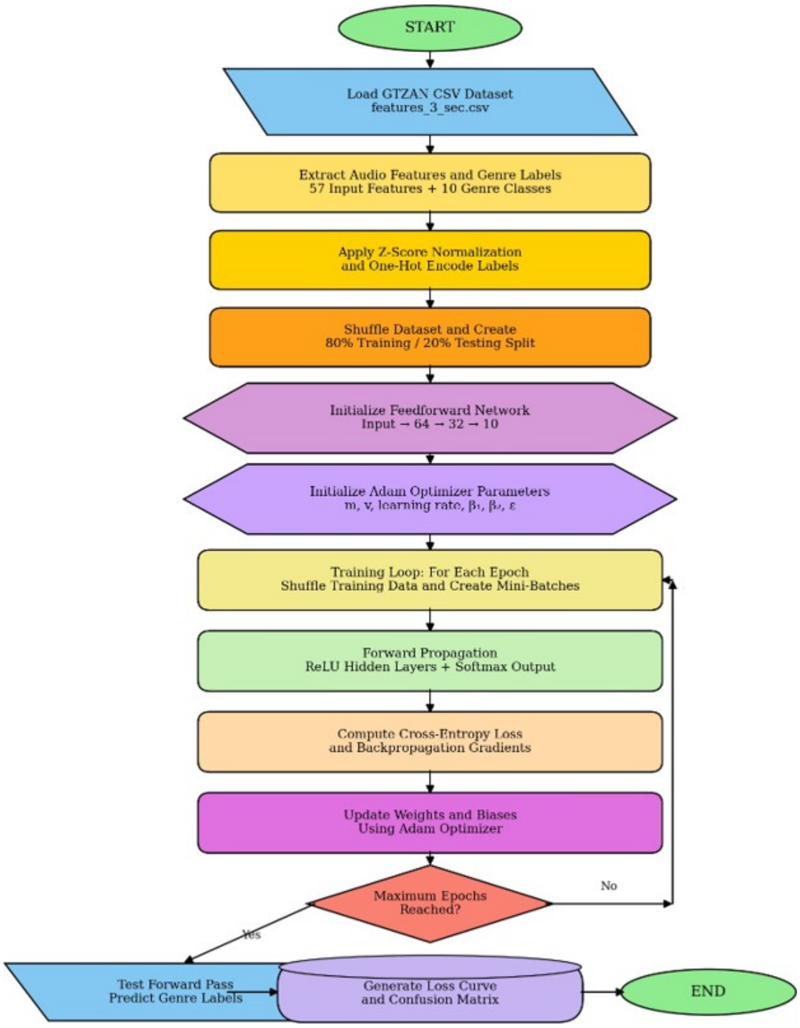

Figure 2.1: Flowchart of Music Genre Classifier Using Feed Forward Neural Network

2.4 Results

The successful completion of the training is validated by the neural convergence curve, which shows the plot between epochs and categorical cross entropy loss. Heat map of the confusion matrix is given which shows how accurate the model was in predicting the correct genre and finally, the accuracy is shown.

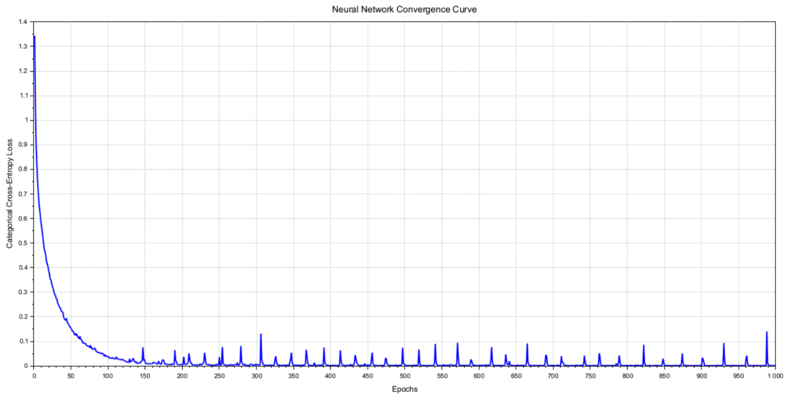

Figure 2.2: Neural Convergence Curve

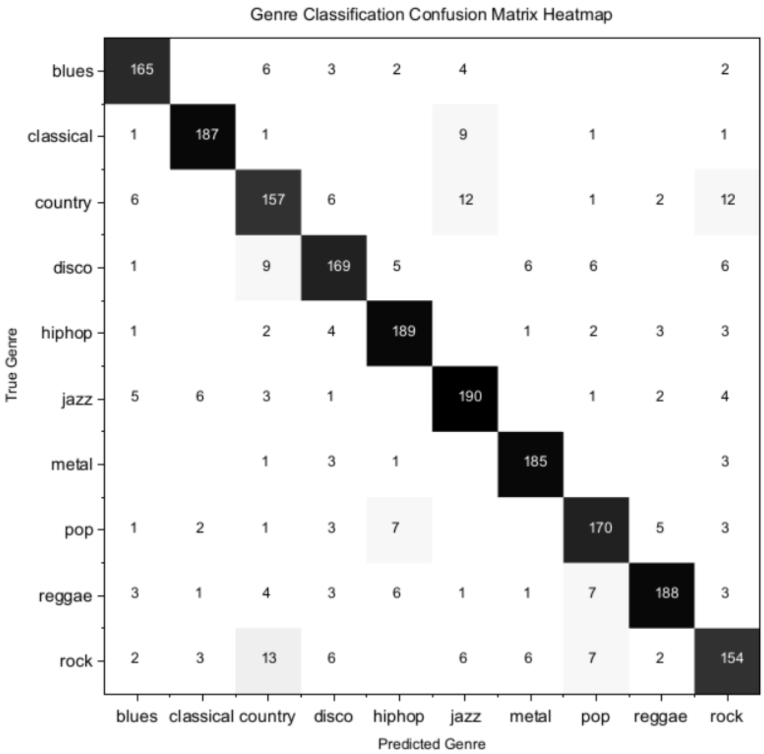

Figure 2.3: Heat Map of the Confusion Matrix

```
Training the Neural Network...
Epoch 50/1000 - Cross-Entropy Loss: 0.1522
Epoch 100/1000 - Cross-Entropy Loss: 0.0360
Epoch 150/1000 - Cross-Entropy Loss: 0.0130
Epoch 200/1000 - Cross-Entropy Loss: 0.0035
Epoch 250/1000 - Cross-Entropy Loss: 0.0361
Epoch 300/1000 - Cross-Entropy Loss: 0.0028
Epoch 350/1000 - Cross-Entropy Loss: 0.0021
Epoch 400/1000 - Cross-Entropy Loss: 0.0014
Epoch 450/1000 - Cross-Entropy Loss: 0.0060
Epoch 500/1000 - Cross-Entropy Loss: 0.0040
Epoch 550/1000 - Cross-Entropy Loss: 0.0011
Epoch 600/1000 - Cross-Entropy Loss: 0.0016
Epoch 650/1000 - Cross-Entropy Loss: 0.0013
Epoch 700/1000 - Cross-Entropy Loss: 0.0017
Epoch 750/1000 - Cross-Entropy Loss: 0.0010
Epoch 800/1000 - Cross-Entropy Loss: 0.0017
Epoch 850/1000 - Cross-Entropy Loss: 0.0034
Epoch 900/1000 - Cross-Entropy Loss: 0.0010
Epoch 950/1000 - Cross-Entropy Loss: 0.0015
Epoch 1000/1000 - Cross-Entropy Loss: 0.0020

>> Upgraded 2-Hidden-Layer Network Accuracy: 88.09%
```

Figure 2.4: Cross-Entropy Loss for every 50 epoches and training accuracy

Chapter 3

Single-Digit DTMF Signal Generator and Detector Using Goertzel Algorithm in Xcos

3.1 Abstract

This case study performs a partial replication of the reference paper focusing on DTMF generation and Goertzel-based detection in Scilab/Xcos. Dual Tone Multi Frequency (DTMF) is the signaling method used in telephones to represent keypad digits using sound. When a key is pressed, two tones are produced together, one from the low-frequency group and one from the high-frequency group. This project focuses on the generation and detection of DTMF signals using the Goertzel algorithm. In the generator section, each keypad digit is assigned its standard pair of frequencies, and the final DTMF signal is created by adding the two sine waves. In the detector section, the generated signal is passed through Goertzel blocks designed for all standard DTMF frequencies. The Goertzel algorithm is useful here because it can efficiently measure the energy present at selected frequencies without calculating the full spectrum. After calculating the energy values, the frequency with the highest energy in the low group and the frequency with the highest energy in the high group are selected. This pair is then mapped back to the corresponding keypad digit. The project helps in understanding how tone-based communication works in real telephone systems. It also shows how signal processing techniques can be used to detect information from an audio signal in an efficient and reliable way

3.2 Problem Statement

DTMF signaling is widely used in telephone and automated communication systems to identify keypad inputs. Each key on a telephone keypad is represented by two tones, one from the low-frequency group and one from the high-frequency group. The main problem in this case study is to design a system that can generate these DTMF tones for different keypad digits and then correctly detect the original digit from the received signal. The solution should first generate a valid DTMF signal by selecting the correct pair of frequencies for a given input digit and combining the two sinusoidal signals. After generation, the detector should analyse the signal and identify which two frequencies are present with the highest energy. Based on this detected frequency pair, the system should map the result back to the corresponding keypad digit. The accuracy of the detector can be improved by using proper sampling frequency, suitable signal duration, thresholding, and clear comparison of frequency energy values. To achieve this aim, the project uses the Goertzel algorithm for DTMF detection. Instead of calculating the complete frequency spectrum, the Goertzel algorithm checks only the required DTMF frequencies, making the detection process simpler and more efficient. In the project, the generator produces the DTMF tone and the detector computes energy values for all standard DTMF frequencies. The final digit is obtained by selecting the highest energy frequency from both the low and high frequency groups. Lastly, the validity of the detected signal is tested by energy threshold test, twist test, relative peak test.

3.3 Flowchart

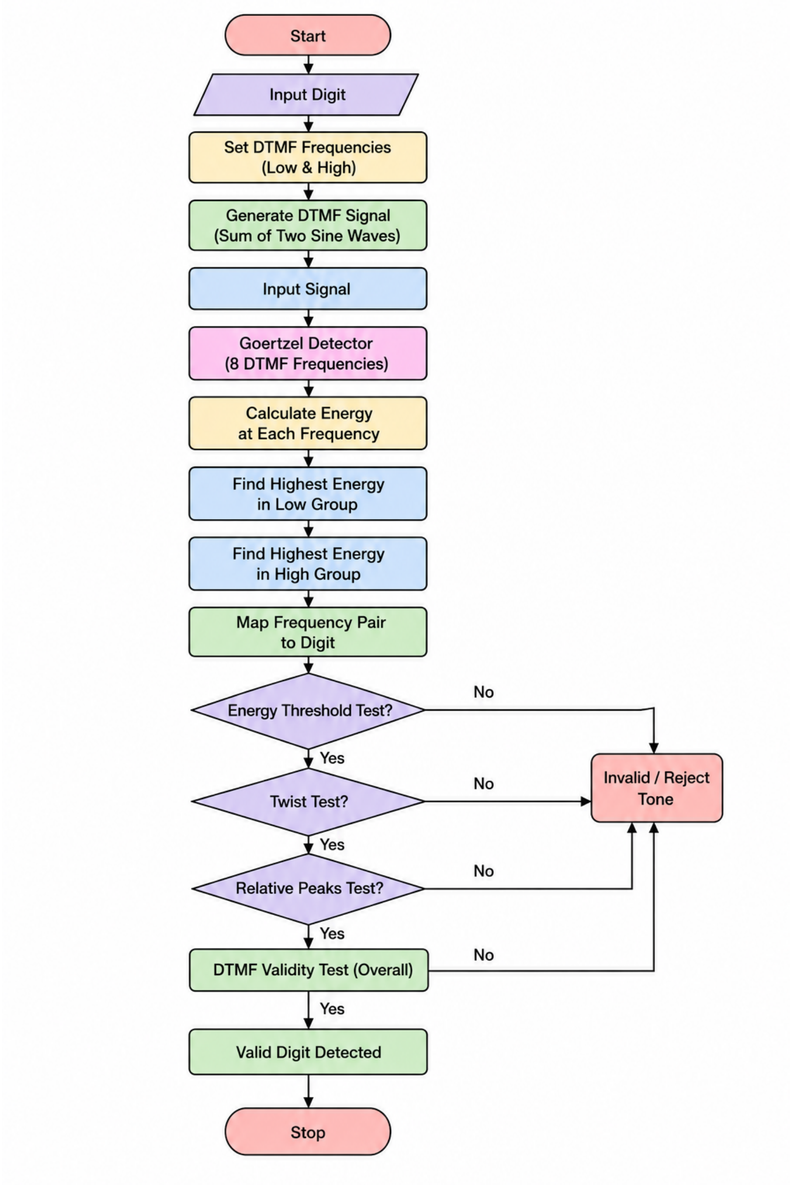

Figure 3.1: Flowchart of Single-Digit DTMF Signal Generator and Detector Using Goertzel Algorithm in Xcos

3.4 Results

The xcos diagram, the Goertzel detector is shown below. The generated DTMF Signal for digit 6 and 7 and their energy outputs are given. Also, their validation test results are shown.

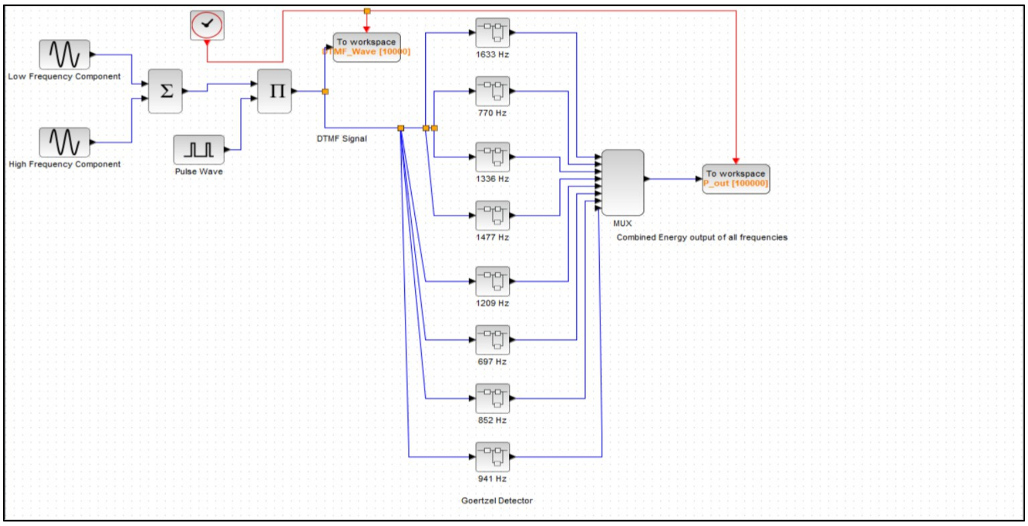

Figure 3.2: Xcos Block Diagram

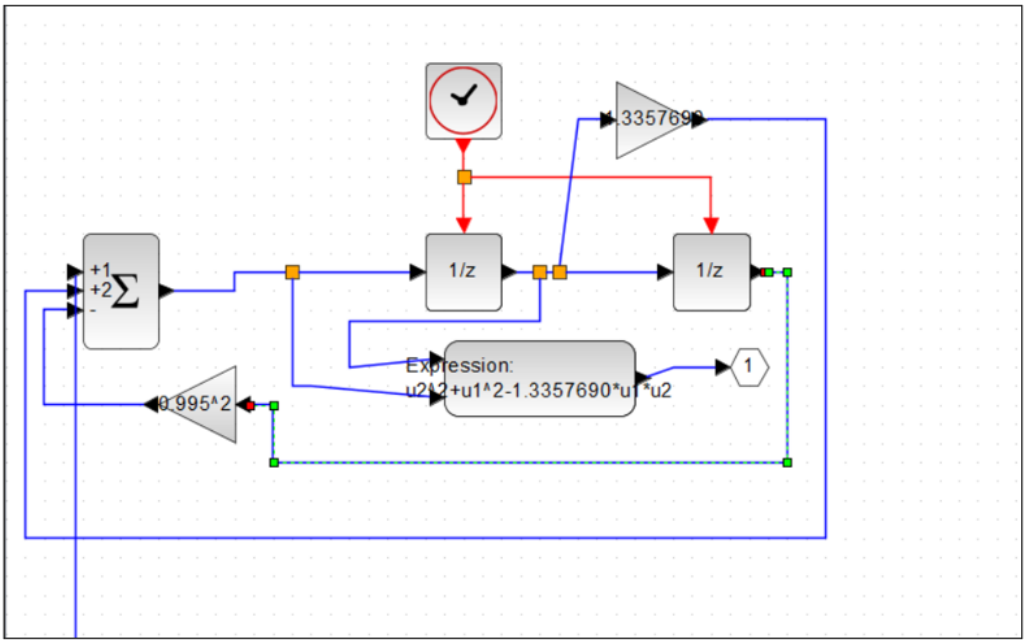

Figure 3.3: Goertzel Detector

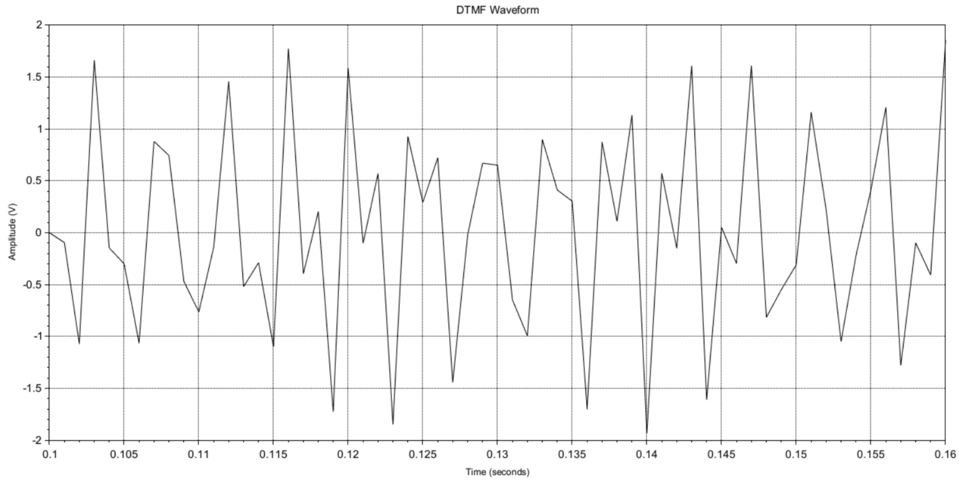

Figure 3.4: Digit ’6’

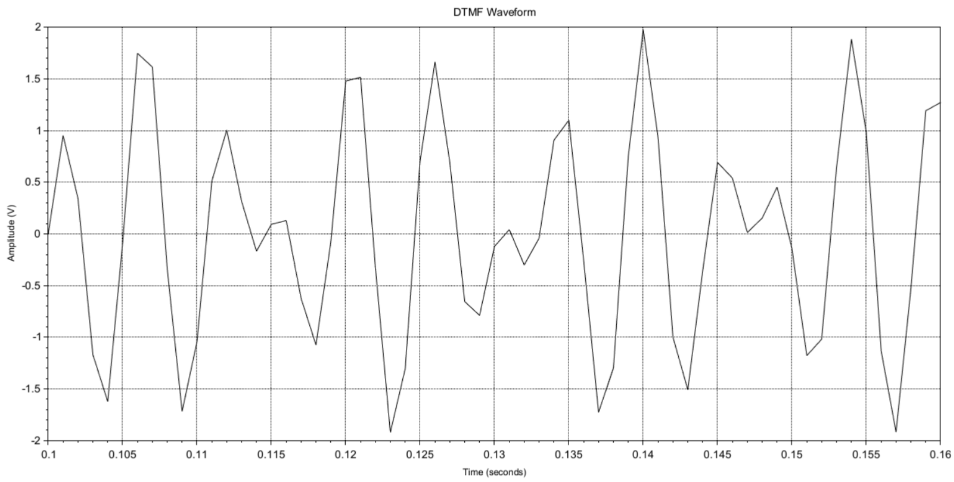

Figure 3.5: Digit ’7’

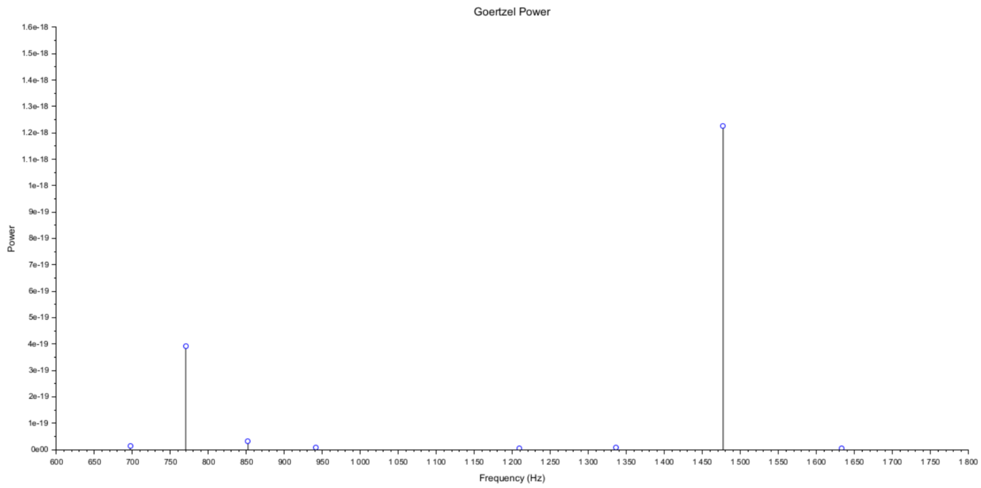

Figure 3.6: Output Energy for Digit ’6’

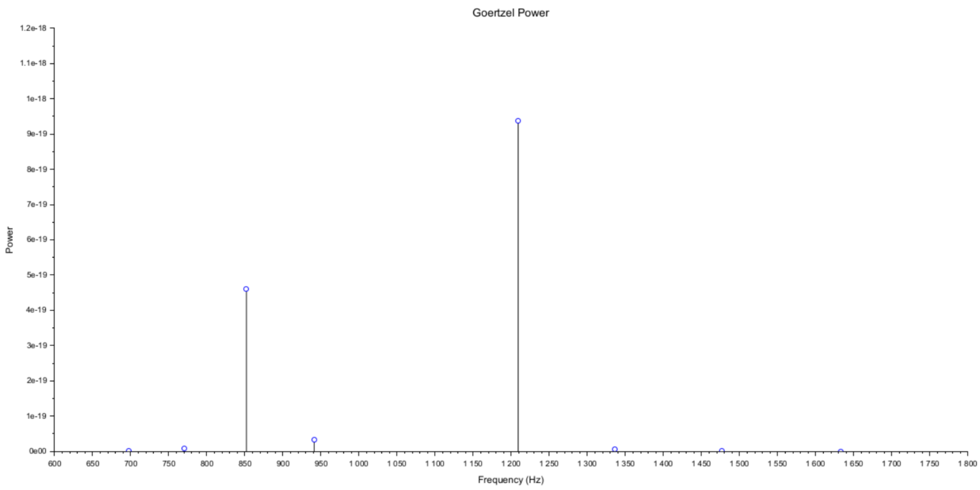

Figure 3.7: Output Energy for Digit ’7’

```
--> exec('C:\Users\vaidy\OneDrive\Desktop\dtmf.sce', -1)
  "1. Energy Threshold Test : T"
  "2. Twist Test: T [3.245822 dB, Reverse Twist (High > Low)]"
  "3. Relative Peaks Test: T"
  "Dominant Low Frequency: 770 Hz"
  "Dominant High Frequency: 1477 Hz"
  "Detected Digit:6"
```

Figure 3.8: Validation tests for Digit ‘6’

```
--> exec('C:\Users\vaidy\OneDrive\Desktop\dtmf.sce', -1)
  "1. Energy Threshold Test : T"
  "2. Twist Test: T [1.8344268 dB, Reverse Twist (High > Low)]"
  "3. Relative Peaks Test: T"
  "Dominant Low Frequency: 852 Hz"
  "Dominant High Frequency: 1209 Hz"
  "Detected Digit:7"
```

Figure 3.9: Validation tests for Digit ‘7’

Chapter 4

ECG Signal Processing and Heart Rate Frequency Detection

4.1 Abstract

Electrocardiogram (ECG) signals are used to monitor the electrical activity of the heart, but they are often affected by noise and interference that make analysis difficult. This project uses a GUI-based ECG signal processing system developed in Scilab for visualizing, filtering, and analyzing ECG recordings. The application allows users to upload ECG data, add controlled noise, view the raw signal, and apply digital filtering techniques. A notch filter is used to reduce 50 Hz power-line interference, while smoothing operations improve signal quality. The filtered signal is then converted into an energy signal that shows the R-peaks of the QRS complex. Based on the detected R-peaks, the heart rate is estimated in beats per minute (BPM). The project partially replicates the reference paper ECG Signal Processing and Heart Rate Frequency Detection Methods as it does not implement all the heartbeat detection algorithms presented in the paper. Instead, it focuses on ECG filtering, R-peak enhancement, and a threshold-based heart rate method.

4.2 Problem Statement

ECG signals are widely used for monitoring heart activity and detecting cardiac abnormalities. However, ECG recordings are often affected by power-line interference, random noise, and other signal distortions, making it difficult to accurately identify important features such as the R-peaks of the QRS complex. Therefore, there is a need for a simple and interactive system that can process ECG signals, reduce noise, and assist in basic heart rate analysis. The project is a GUIbased ECG signal processing system developed in Scilab. It allows users to upload ECG data, visualize the raw signal, introduce controlled noise, and apply digital filtering techniques. The processed signal is further analysed to enhance R-peaks and estimate heart rate from the detected peaks. The system can be improved in the future by implementing advanced heartbeat detection algorithms, adaptive filtering techniques, and support for real-time ECG acquisition. To achieve these objectives, digital signal processing methods such as notch filtering, signal smoothing, R-peak energy enhancement, and threshold-based peak detection are used. A graphical user interface is made to provide a platform for ECG visualization and analysis.

4.3 Flowchart

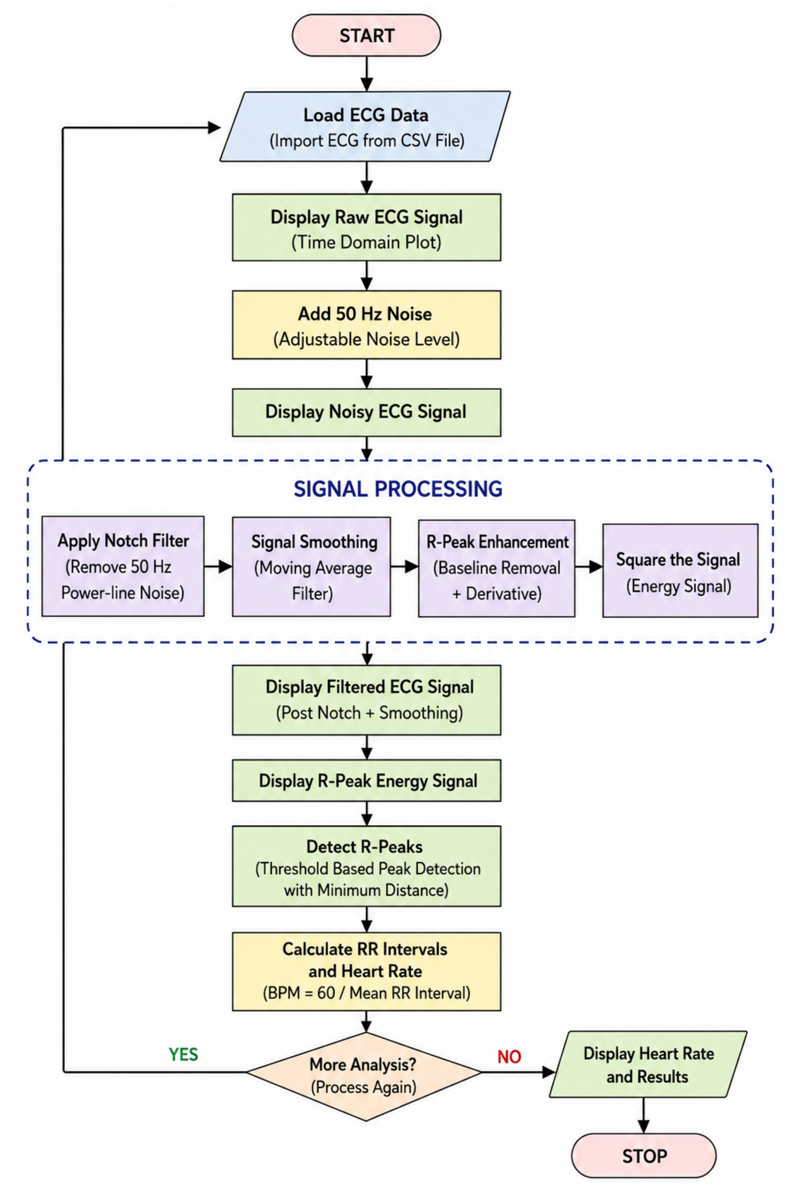

Figure 4.1: Flowchart of ECG Signal Processing and Heart Rate Frequency Detection

4.4 Results

The GUI has been created which allows the user to upload ECG Signal, add noise to it and apply digital filters. The raw ECG Signal, ECG signal with 50Hz noise is given below. The ECG signal is then filtered using signal mean, followed by 50 Hz notch filtering to suppress power-line interference and moving-average smoothing operation. R-peak filtering is done to enhance the R peaks and finally the heart rate was estimated.

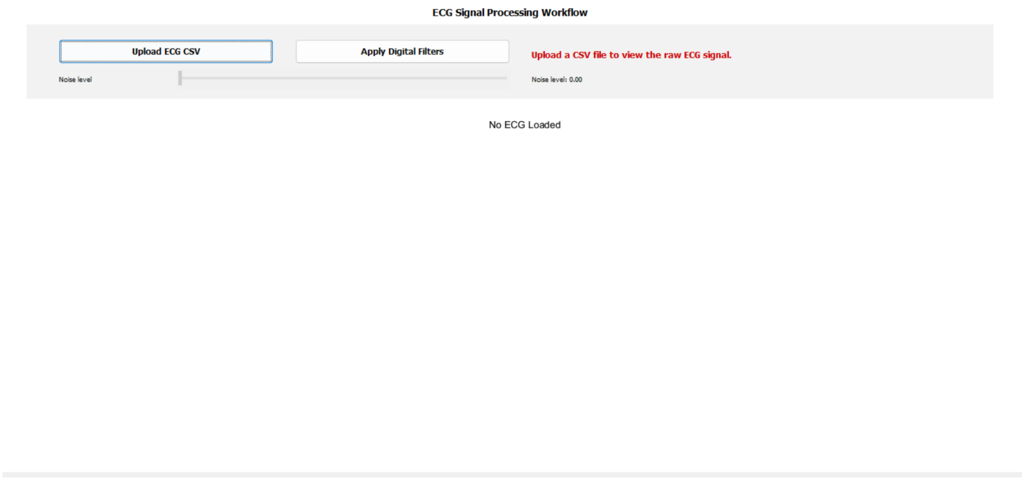

Figure 4.2: The GUI Window Before Uploading ECG Signal.

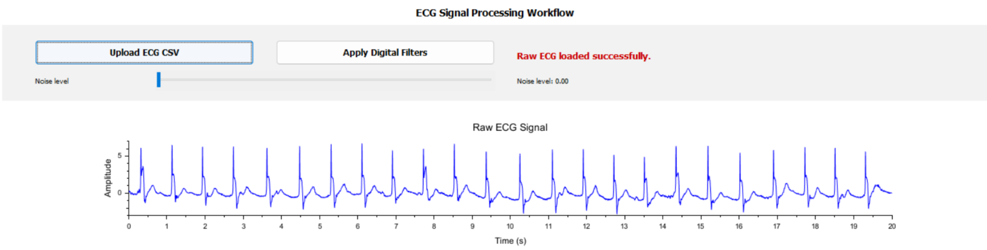

Figure 4.3: Raw ECG Signal.

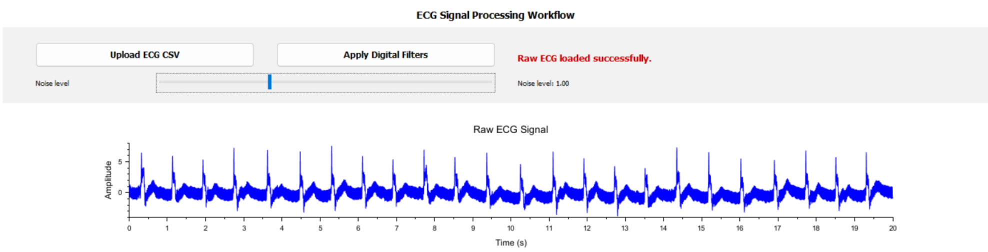

Figure 4.4: Raw ECG Signal with 50Hz Noise

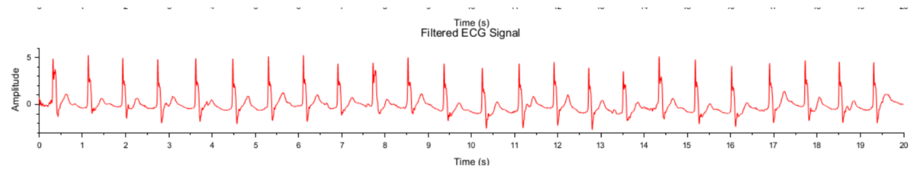

Figure 4.5: Filtered ECG Signal

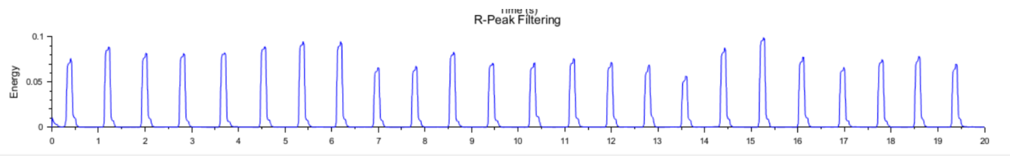

Figure 4.6: R-Peaks Filtering

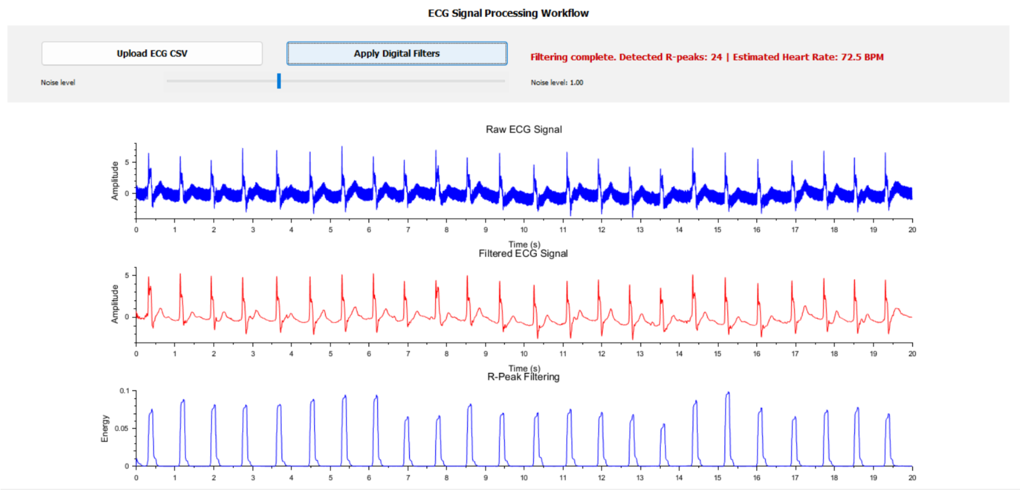

Figure 4.7: Detected R-peaks and Estimated Heart Rate

Learning and Conclusion

The FOSSEE Summer Fellowship has been an enriching learning experience that provided me with the opportunity to explore different domains using Scilab and open-source tools. Through the three case studies, I gained practical experience in machine learning, digital communication, and biomedical signal processing while strengthening my programming and analytical skills.

In Case Study 1, a Music Genre Classification System was developed using a Feed Forward Neural Network. The project involved extracting audio features from music files, training the neural network, and classifying songs into different genres. This study demonstrated the application of machine learning techniques for audio classification and achieved an accuracy of 88.09

In Case Study 2, a Dual Tone Multi Frequency (DTMF) Detection System was implemented using the Goertzel algorithm. The project successfully identified telephone keypad tones by analyzing their frequency components, demonstrating the practical application of digital signal processing techniques in communication systems.

In Case Study 3, an ECG Signal Processing and Heart Rate Detection System was developed using Scilab. The project implemented digital filtering, R-peak detection, and heart-rate estimation through an interactive graphical user interface, illustrating the application of signal processing techniques in biomedical engineering.

Overall, the fellowship significantly enhanced my understanding of Scilab, GUI development, digital signal processing, and machine learning. The knowledge and hands-on experience gained through these projects have strengthened my technical skills and prepared me to confidently apply open-source technologies to solve real-world engineering problems in future academic and professional pursuits.

References

- Qiuqiang Kong, Xiaohui Feng, Yanxiong Li. Music Genre Classification using Convolutional Neural Network PDF Link
- Pamuk, N., & Pamuk, Z. (2015). Dual Tone Multi Frequency (DTMF) signal generation and detection using MATLAB software.PDF Link
- Parák, Jakub, and Jan Havlík. "ECG signal processing and heart rate frequency detection methods." Proceedings of Technical Computing Prague 8 (2011): 2011.PDF Link
- Scilab - An Open Source Software https://scilab.in/
- FOSSEE Project, IIT Bombay https://fossee.in/