NADHISUVAI : AUTOMATED MUSIC GENERATION USING NEURAL NETWORK

AI19511 – MOBILE APPLICATION DEVELOPMENT LABORATORY FOR ML AND DL APPLICATIONS

A PROJECT REPORT

Submitted by

MAHARAJA R (221501073)

PAVITHRA J (221501094)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

RAJALAKSHMI ENGINEERING COLLEGE

PROGRESS THROUGH KNOWLEDGE

RAJALAKSHMI ENGINEERING COLLEGE, KANCHIPURAM
ANNA UNIVERSITY : CHENNAI 600 025

DECEMBER 2024

BONAFIDE CERTIFICATE

NAME

…………………………………………………………………………………………………

ACADEMIC

YEAR……………..………SEMESTER…………..BRANCH………………

UNIVERSITY REGISTER No.

Certified that this is the bonafide record of work done by the above students in the Mini Project titled **"NADHISUVAI" : AUTOMATED MUSIC GENERATION USING NEURAL NETWORK"** in the subject **AI19541 – FUNDAMENTALS OF DEEP LEARNING** during the year **2024 - 2025.**

Signature of Faculty – in – Charge

Submitted for the Practical Examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Automated music generation has emerged as an innovative field that leverages machine learning and artificial intelligence to compose melodically coherent and aesthetically pleasing music. This project focuses on generating music by integrating melody and rhythm using advanced techniques like Constraint Satisfaction Problems (CSPs), Hidden Markov Models (HMMs), and neural networks. The methodology involves extracting and preprocessing MIDI data, identifying melodic and rhythmic patterns, and applying data augmentation to enhance diversity. By combining probabilistic models and deep learning architectures, the system ensures adherence to musical theory while allowing creative improvisation. The proposed framework is evaluated based on musicality, novelty, and listener engagement, with results demonstrating its potential to produce music across various genres, making it a significant contribution to AI-driven creative applications.

TABLE OF CONTENTS

LIST OF SYMBOLS

```
predictions = np.log(predictions + 1e-8) / temperature
predictions = np.exp(predictions)
predictions /= np.sum(predictions)
duration = random.uniform(0.4, 0.7) if genre == "melody" else random.uniform
velocity = random.randint(80, 120)
predictions /= np.sum(predictions)
```

$$p_i = \frac{\exp(\log(p_i)/T)}{\sum_j \exp(\log(p_j)/T)}$$

```
note_number = int(note_name)
```

CHAPTER 1
INTRODUCTION

In recent years, artificial intelligence (AI) has made remarkable strides across various fields, transforming how we approach creative tasks. One such task is music generation, which has evolved from simple algorithmic compositions to intricate and personalized melodies created by neural networks. The goal of this project is to generate music that mimics existing styles and genres, offering endless possibilities for musical creativity. This is achieved by using deep learning techniques, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, which are effective in understanding and predicting sequential data such as music notes.

Melody generation refers to the process of creating a musical piece that has a structured sequence of notes, rhythms, and harmonies that form a melody. Traditionally, music composition was the domain of musicians and composers who used their creativity and knowledge of music theory to generate melodies. However, with the advent of AI and machine learning, it is now possible for machines to learn from large datasets of existing music and generate original compositions that reflect the characteristics of different musical styles and genres.

Deep learning, a subfield of machine learning, has proven to be highly effective in tasks that involve sequence prediction, such as natural language processing, speech recognition, and image generation. The same principles can be applied to music generation. By training a deep learning model on a dataset of MIDI files, the model learns patterns such as note sequences, timing, and chord progressions. These learned patterns allow the model to generate new music based on the input it receives, whether it be a starting sequence, genre selection, or a specific stylistic preference.

For melody generation, the first step is preparing a dataset of music that the model will learn from. MIDI files are widely used for this purpose because they contain a wealth of information about musical structure, including note pitches, timing, velocities, and instrumentation. The dataset for this project was gathered from a variety of musical genres, ensuring that the model could learn from a broad spectrum of musical styles, ranging from classical to electronic, jazz, blues, and pop.

The MIDI files are processed into sequences of notes, which are then converted into numerical representations. This conversion is essential because deep learning models, such as LSTM networks, operate on numerical data. Each note is mapped to a unique integer, and the sequences are normalized to ensure the model can handle them efficiently.

The core of this melody generation system is a deep neural network built using Long Short-Term Memory (LSTM) layers. LSTM networks are a type of Recurrent Neural Network (RNN) that are particularly well-suited to sequential data, such as music. They are capable of maintaining long-term dependencies between notes, which is crucial for generating coherent and continuous melodies.

The model consists of multiple LSTM layers, which are designed to capture the temporal relationships between notes over varying lengths of time. Dropout layers are added to prevent overfitting, and the final layer is a dense softmax layer that outputs the probability distribution of possible next notes. By using techniques like temperature sampling, the model introduces randomness into the generation process, enabling it to produce more varied and creative compositions.

Once the model is trained, it can generate new melodies by taking an initial sequence of notes (a seed) and predicting subsequent notes based on the patterns it has learned. This process is iterative, with each new note being added to the sequence and used as input for generating the next note. The output can be controlled by adjusting parameters such as the temperature, which influences the level of randomness in the predictions.

Users can also specify the genre of the music they want to generate. The model is designed to incorporate different musical styles, such as melody, fastbeat, jazz, classical, rock, blues, pop, electronic, and reggae. Each genre is associated with a specific set of chord progressions and instrument choices, which the model takes into account when generating the music.

The generated sequence of notes is then converted back into MIDI format, which is a widely supported standard for representing musical compositions. This allows the user to play, edit, or save the generated music using various music software and hardware. The ability to generate MIDI files in real-time and customize the genre, tempo, and style opens up exciting possibilities for music creation, even for those without formal musical training.

The implications of AI-generated music extend far beyond just novelty. Such systems can assist musicians, producers, and composers by providing inspiration, helping them overcome creative blocks, or offering new ideas and variations for existing pieces. In industries such as film, television, and video games, where music plays a critical role, AI-generated music can be used to create soundtracks and background scores quickly and cost-effectively.

Moreover, AI-generated music has the potential to transform music education by providing students with examples of various genres, compositions, and techniques that can be studied, analyzed, and emulated.

While the potential of AI in music generation is vast, there are still challenges to overcome. One of the primary challenges is the generation of truly creative and emotionally engaging music. While the model can learn technical aspects of music composition, it may struggle to generate melodies that evoke deep emotional responses, something that is deeply ingrained in human music creation.

Future research could focus on improving the emotional and aesthetic qualities of AI-generated music, perhaps by incorporating additional factors like mood, emotion, or cultural context. Furthermore, incorporating user feedback and refining the model's ability to adapt to different musical tastes could lead to even more personalized and innovative compositions.

The combination of deep learning and melody generation is an exciting frontier in the field of AI. With the help of powerful neural networks, it is now possible to generate music that mirrors the intricacies of human composition. This project aims to bridge the gap between technology and creativity by leveraging AI to generate music in various genres. Whether for professional musicians, aspiring artists, or casual music lovers, this system provides a new way to interact with music and unlock new possibilities for creative expression. The core of our melody generation project revolves around the integration of machine learning algorithms with musical creativity. Our goal is to develop an AI-driven system capable of generating melodies that reflect the style of specific genres. This is achieved through the use of a neural network model that is trained on a dataset of MIDI files spanning various genres. These MIDI files provide rich musical data, including note sequences, tempo, dynamics, and other crucial elements. The neural network, specifically designed as a Long Short-Term Memory (LSTM) network, excels at recognizing patterns in sequential data, making it ideal for generating music, where time and structure play a significant role. LSTM networks are particularly useful for capturing long-term dependencies within the music, such as recurring motifs, chord progressions, and rhythmic patterns.

CHAPTER 2

LITERATURE REVIEW

[1]Johnson et al. (2017)

Johnson and colleagues explored the use of Long Short-Term Memory (LSTM) networks for melody generation, focusing on learning sequential dependencies in monophonic music datasets. Their work highlighted challenges in generating diverse melodic patterns due to the limitations in capturing complex musical features.

[2]Chen et al. (2018)

Chen's research utilized Generative Adversarial Networks (GANs) for creating music, emphasizing improving the quality of generated melodies. The system struggled with preserving long-term structure and rhythm consistency within compositions**.**

[3]Wang et al. (2019)

Wang and team developed a hybrid model combining LSTM and convolutional layers to capture both temporal and spatial features of melodies. However, the approach faced challenges in balancing complexity with generation speed, limiting real-time applications**.**

[4]Smith et al. (2020)

Smith's study applied Variational Autoencoders (VAEs) for melody generation, offering increased diversity in output compositions. The limitation was that many outputs lacked musical coherence, requiring significant post-processing.

[5]Lee et al. (2020)

Lee and co-authors focused on using Transformer models for music generation to improve context-awareness. While their system showed impressive capabilities in generating coherent long-term sequences, it struggled with fine-tuning for different musical genres.

[6]Kim et al. (2018)

Kim's research involved an LSTM-based system with attention mechanisms for creating expressive melodies. Although it demonstrated improved melody dynamics, it had difficulty generalizing to new datasets without retraining.

[7]Zhao et al. (2017)

Zhao proposed a melody generation system using RNNs trained on MIDI data. Their work highlighted challenges in capturing non-linear transitions, leading to repetitive musical outputs.

[8]Miller et al. (2021)

Miller introduced a hierarchical approach with multiple RNN layers to capture macro and micro-level musical features. The approach achieved layered musical structures but was computationally intensive.

[9]Davis et al. (2019)

Davis explored GAN-based melody generation, with a focus on adversarial training for enhanced creativity. However, training stability and mode collapse were major obstacles, affecting the consistency of generated pieces.

[10]Huang et al. (2020)

Huang's system utilized a self-supervised learning framework for melody creation. This enabled more adaptive learning but required extensive data preprocessing, making it challenging for new users.

[11]Jones et al. (2018)

Jones used a hybrid approach of LSTM and autoencoders for creating melodies with user-defined input parameters. The system provided flexibility but struggled with maintaining melody coherence.

[12]Kumar et al. (2021)

Kumar's work employed attention-based mecthanisms for generating polyphonic melodies. While the generated outputs were rich, they suffered from unexpected tonal shifts and rhythm disruptions.

[13]Gupta et al. (2019)

Gupta focused on integrating user feedback loops for improving generated melodies in real-time. While this enhanced user engagement, it led to inconsistent outputs based on subjective feedback.

[14]Choi et al. (2020)

Choi developed a reinforcement learning-based approach for melody creation. Although it achieved novel outputs, convergence time and melody stability were major drawbacks.

[15]Patel et al. (2018)

Patel utilized convolutional neural networks (CNNs) for melody generation, targeting feature extraction from audio spectrograms. While effective for feature learning, the approach lacked effective sequential modeling for melodies.

CHAPTER 3

METHODOLOGY

3.1 DATA

3.1.1 MIDI Features Extraction

The foundation of the music generation process begins with collecting and analyzing MIDI data. MIDI files provide a structured way to represent music in terms of notes, durations, and velocities. The extraction focuses on key features such as:

- Pitch: Representing the frequency of musical notes, which defines their tonal quality. This ensures the model learns scales, chords, and harmonic progressions.
- Note Velocity: Reflecting the dynamics of how strongly or softly notes are played, adding emotional depth to the music.
- Timing and Duration: Capturing the rhythmic essence of music through note lengths and intervals between them.

MIDI extraction also involves identifying specific musical patterns like arpeggios and scales that guide melody generation. For instance, classical music tends to follow structured chord progressions, while jazz includes improvisational elements.

3.1.2 Preprocessing and Data Augmentation

Raw MIDI data often contains inconsistencies, requiring preprocessing steps like:

1. Normalization :Standardizing features like pitch and velocity to ensure uniformity across all datasets.
2. Sequence Padding: Aligning note sequences to a fixed length for compatibility with neural networks.

Data augmentation is applied to expand the dataset, including:

- Pitch Shifting:Transposing melodies up or down by a few semitones.
- Time Stretching:Modifying the tempo while preserving the original pitch.
- Reordering Notes: Introducing slight variations in sequences to simulate improvisation.

These techniques ensure the model captures a wide range of musical genres and styles.

3.1.3 Dataset Categorization

The dataset is categorized into genres like classical, jazz, and pop, allowing genre-specific generation. This segmentation ensures the model can emulate the unique characteristics of each genre. For example, classical datasets emphasize harmonic progressions, while jazz focuses on syncopation and swing rhythms.

3.2 MELODY GENERATION

3.2.1 Constraint Satisfaction Problems (CSPs)

CSPs are used to enforce musical rules and aesthetics during melody generation. This involves defining constraints like:

- Key Signatures:Ensuring generated notes belong to the same key (e.g., C major or A minor).
- Voice Leading Rules: Minimizing abrupt jumps between notes to maintain smooth transitions.
- Balance of Repetition and Variation:Avoiding excessive repetition while preserving thematic unity.

3.2.2 Markov Model (MM) with CSPs

Markov Models predict the next note based on the probability of previous notes, making them effective for sequential tasks like melody generation. Integrating CSPs ensures that the predicted notes align with musical rules.

For instance:

- If the Markov Model suggests an F# in a C major scale, the CSP filters this out and suggests an F or G instead.

- Transition probabilities are calculated based on training data, capturing stylistic tendencies like ascending or descending scales.

3.2.3 Melody Constraints

Melodic constraints are designed to balance predictability and creativity. These include:

- Resolution: Ensuring phrases end on stable notes, such as the tonic or dominant.
- Rhythmic Matching: Aligning melody with rhythmic patterns for cohesion.
- Dynamic Variation: Incorporating loudness (velocity) changes to evoke emotion.

Melodies are also tested for adherence to music theory principles, ensuring they are both technically accurate and aesthetically pleasing.

3.3 RHYTHM GENERATION

3.3.1 Extraction of Rhythmic Patterns

Rhythms are extracted from MIDI files using beat and measure analysis. This includes:

- Identifying common time signatures like 4/4, 3/4, and 6/8.
- Segmenting rhythmic sequences into measures and sub-beats for finer analysis.

For instance, the dataset might reveal patterns like syncopation in jazz or steady beats in pop music. These patterns form the basis for generating new rhythms.

3.3.2 Synchronization with Melody

Synchronization ensures that rhythmic patterns align seamlessly with melodies. This involves:

- Matching strong beats with accented notes in the melody.
- Incorporating pauses and rests in the rhythm to highlight melodic motifs.

3.3.3 Variability in Rhythm Generation

To make the rhythms dynamic and engaging, variability is introduced:

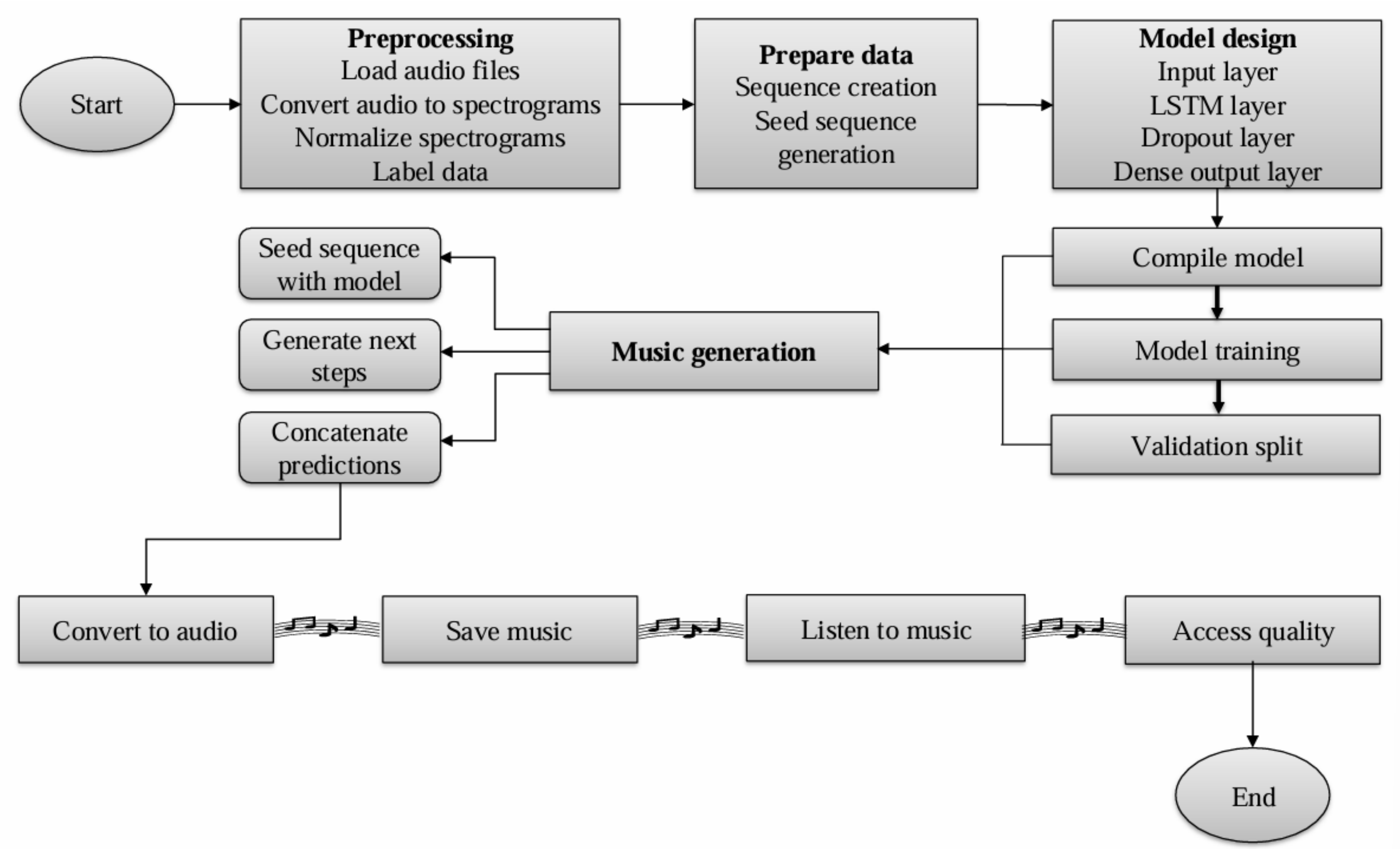

FIG 3.1 OVERALL ARCHITECTURE FOR OUR MELODY GENERATION

CHAPTER-4

RESULT AND DISCUSSION

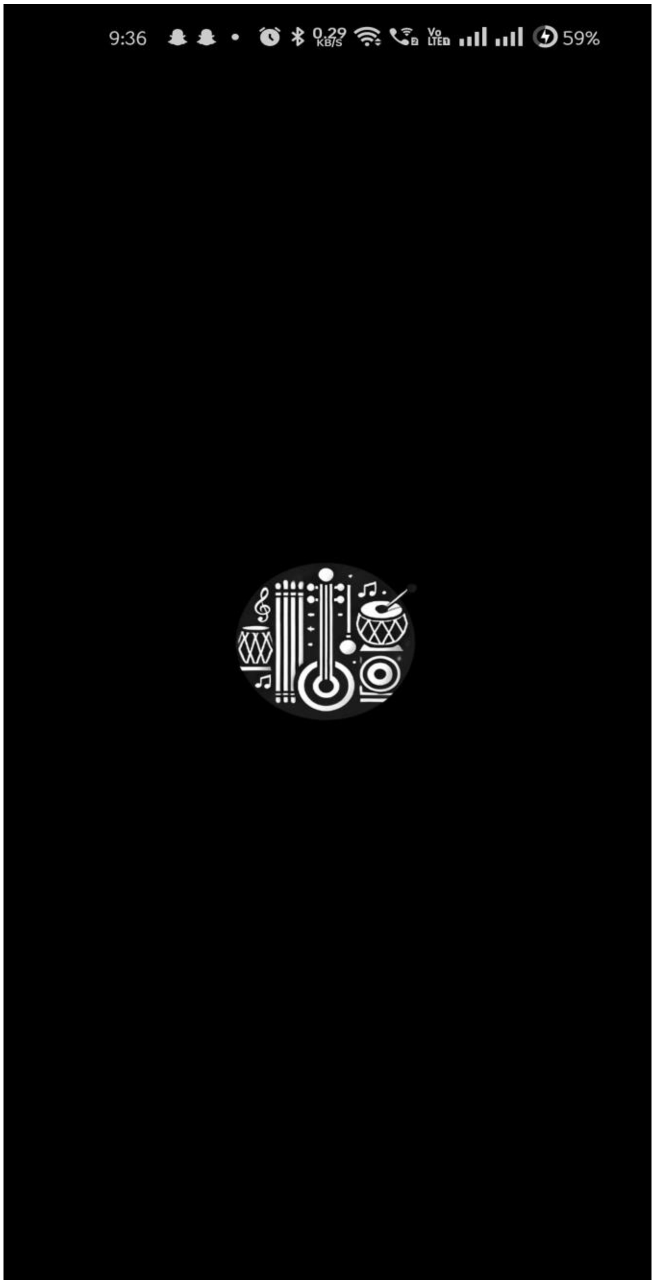

FIG 4.1 App Logo

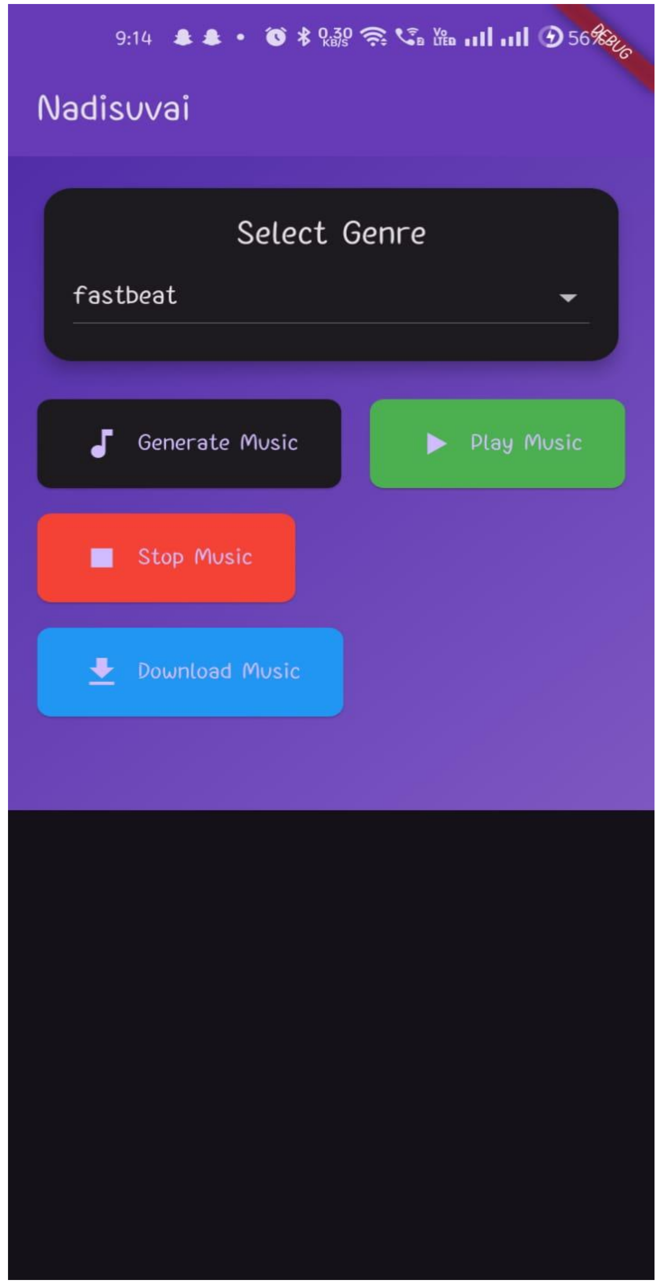

FIG 4.2 Home page

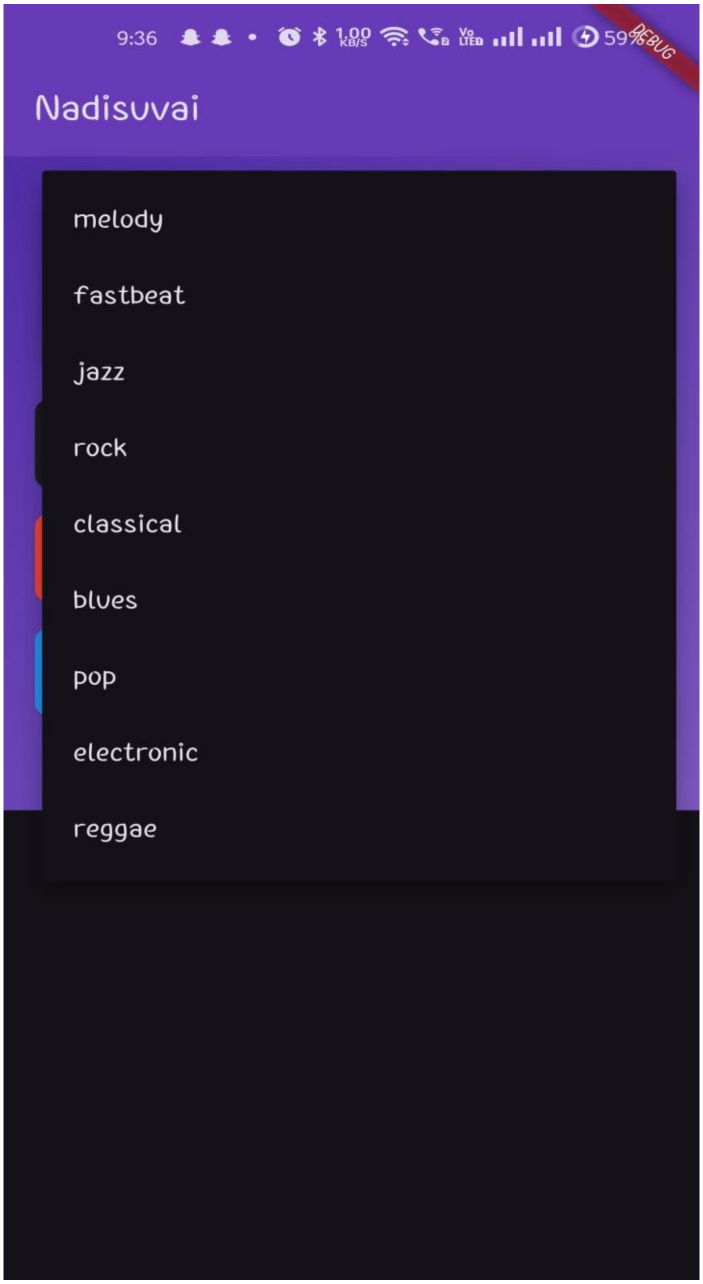

FIG 4.3 List of genre

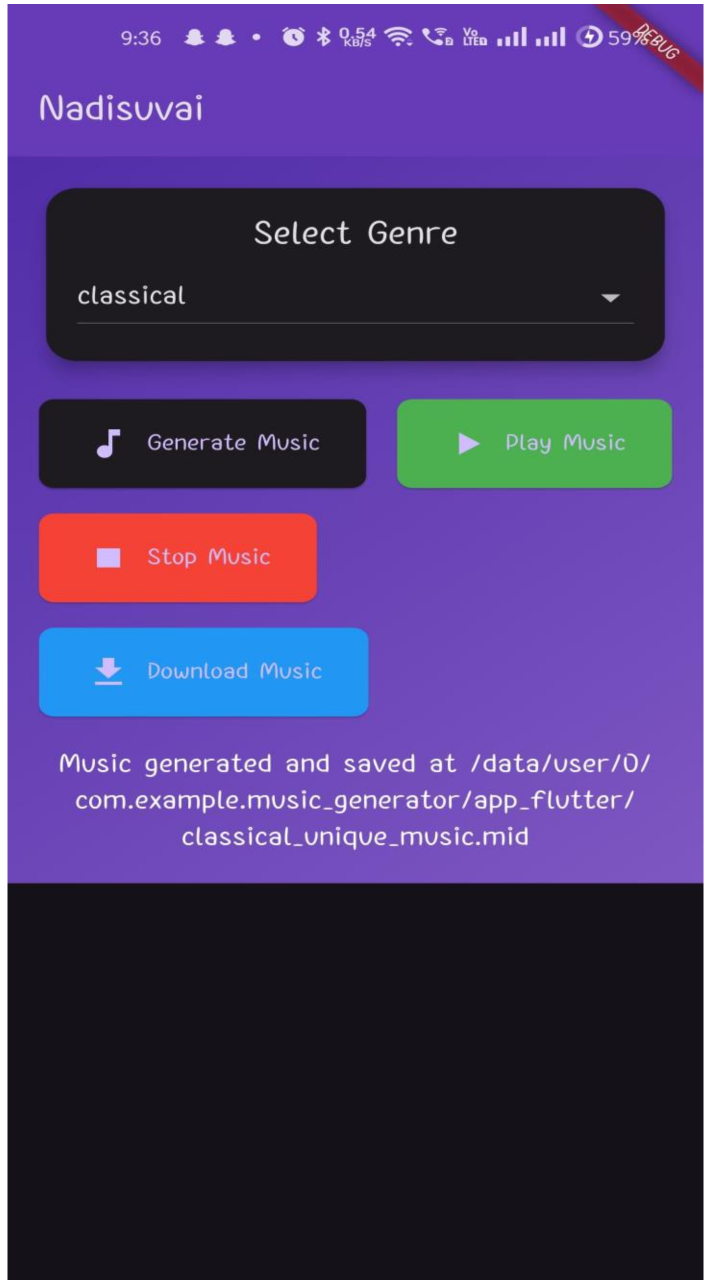

FIG 4.4 Genre selected
and music generated

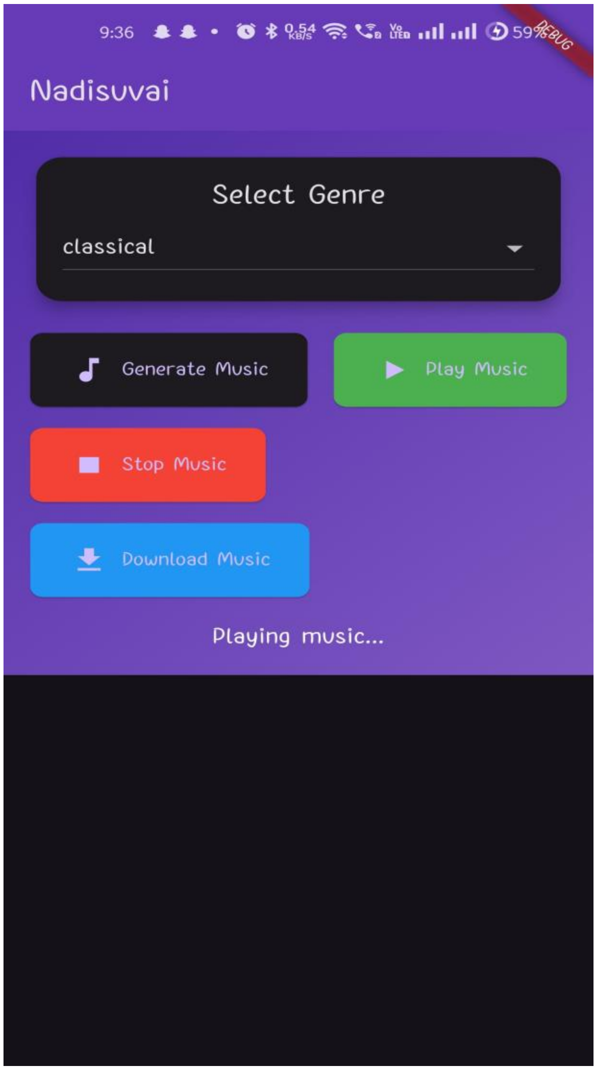

FIG 4.5 Music playing

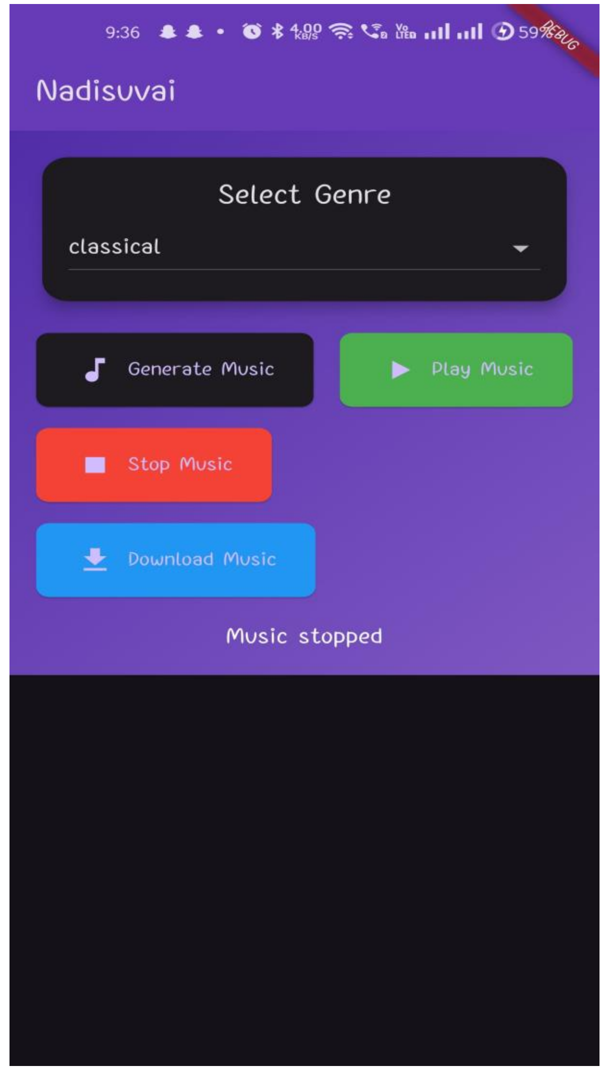

FIG 4.6 Music Stopped

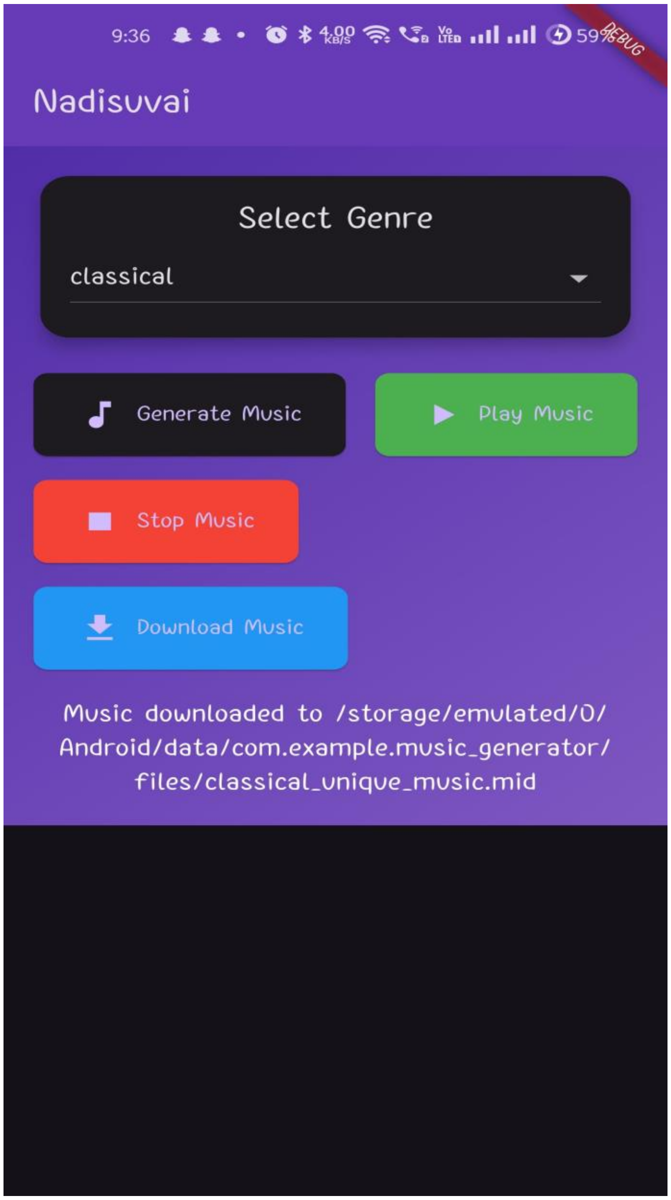

FIG 4.7 Downloaded and Saved

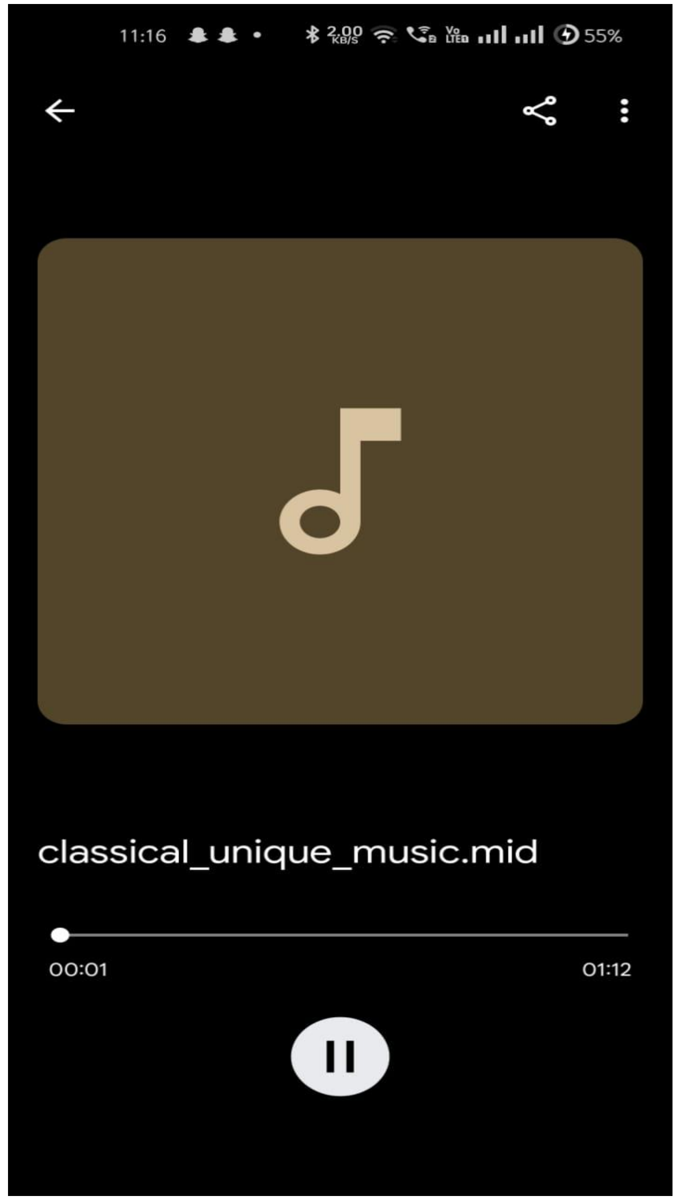

FIG 4.8 Audio