International Journal of Research in Engineering, Science and Management
Volume 8, Issue 8, August 2025
https://www.ijresm.com | ISSN (Online): 2581-5792

Parameter Based Music Generation

Manav Bhanushali[1], Aryan Nathani[2], Sanimar Manghera[3*], Jyoti Ramteke[4], Chandrashekhar Gajbhiye[5]
[1,2,3]Student, Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India
[4,5]Professor, Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

***Abstract*: In this project, we propose a system for text-to-music generation, using AI techniques such as Natural Language Processing (NLP) and deep learning models. Our goal is to empower users to create music compositions using textual descriptions. The work is divided into three phases: literature survey, dataset preprocessing, and model implementation. This system makes music generation more accessible, even to users with no musical expertise, leveraging models such as LSTM, Bi-LSTM, and pre- trained models like MusicGen.**

***Keywords*: AI, NLP, music generation, deep learning, LSTM, Django, MusicGen.**

1. Introduction

In the Text to Music Generation project, we leverage Artificial Intelligence to revolutionize music production. Using advanced techniques such as NLP and Generative AI, our goal is to enable users to generate personalized music compositions based on textual input. This system bridges the gap between user input and AI-generated music, democratizing music production for users of all backgrounds.

2. Problem Definition

Creating high-quality music is a challenge for both be ginners and experienced musicians. Mastery over melodies, rhythm, and harmonics requires significant time and experience. Our project aims to bridge the gap between creativity and usability, inspiring innovation in tools and techniques for music composition. All musicians seek efficient methods to unleash creativity while meeting artistic expectations and deadlines.

A. Scope

Design a parameter-based music generation system, using AI techniques. Designing a user-friendly interface for seamless interaction. This involves sophisticated NLP models used to obtain relevant features from user-provided text inputs. It combines Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models for music generation in alignment with textual inputs. Optimization of the training of our model, its performance and scalability to respond in real time and have high quality music.

B. Objectives

Its overall primary goals are to democratize the creation of music, to use AI-based instruments in the process of sound engineering and composition, and thus bring people closer to building easy ways to create composition. All the traditional barriers separating humans and art in the sense that traditional creativity often requires a minimum or medium level of understanding from various aspects of what that form of art implies make up what this initiative seems to accomplish.

3. Literature Survey

The first direction of research is in the use of advanced deep learning models for AI-based music generation. Shanmukh Krishna (2023) explored the application of Long Short-Term Memory (LSTM) and Transformer models for music generation. They highlighted the gap in the field related to the lack of standardized evaluation metrics for assessing the quality of generated music [1]. In 2023, Ning Zhang et al. conducted research on AI techniques for pop music creation, identifying challenges in maintaining coherence during conditioned generation and aligning the generated music with specific requirements [2]. Shansong Liu et al. (2023) introduced MUSIC KNOWING LLAMA, a system for text- to-music generation using question answering and captioning. Their research encountered difficulties in capturing emotional elements, tempo variations, and genre-specific characteristics in the generated music [3]. Similarly, Olga Vechtomova et al. (2023) focused on real-time composition with LyricJam Sonic, shedding light on the challenges of assessing the quality and coherence of AI generated musical compositions in real-time scenarios [5]. Chunhui Bao and Qianru Sun (2022) discussed the generation of music with emotions, highlighting the need to explore the diversity in generated music and addressing biases in the generation process [6].

4. Design and Methodology

A. Block Diagram

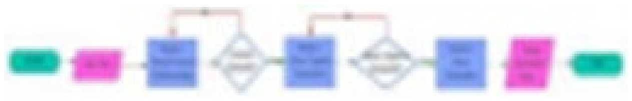

Fig. 1. Block diagram

B. System Overview

The system is composed of two main modules: text processing and music generation. In the text processing module, we use NLP techniques like tokenization and sentiment analysis to extract features such as mood and genre from the input. The

*Corresponding author: sanimar.manghera@spit.ac.in music generation module utilizes models like LSTM, Bi-LSTM, and MusicGen to convert the textual features into music sequences.

5. Implementation

A. Data Collection

Data collection is the most critical step and the one that usually is most difficult in any deep learning project. A model makes good predictions only when it has a well labeled dataset, which is big enough. In our case, we had two main choices: create our own dataset or use some existing publicly available datasets. For music generation, the dataset has to meet some criteria that include being large enough in order to capture diverse and complex musical patterns, have a wide range of styles, genres, instruments, and compositions, and be accessible in standard formats such as MID, WAV, MP3, or symbolic representations as MusicXML or mel-spectrograms. Furthermore, the dataset has to consist of annotations or metadata, which will guide the generated music to comply with specified criteria.

That means creating our own dataset seemed highly imprac tical, as creating both an audio and language annotation proved resource-intensive. The actual creation of the annotations from a model was feasible to some extent, but to get started with a completely custom dataset would require significant amounts of resources. Instead, we decided to rely on datasets that are available, a large number of which comprise music files and their accompanying captions or annotations. These among many will be one of the more suitable for our project- MusicCaps (Google),

MagnaTagATune (University of London), MU-LLaMA, Music QA (MU-LLaMA) or Song Describer Dataset. So here go very important datasets based on robust background for further music model generations.

The dataset for this project consists of two main types of data: textual data, including captions and aspect lists, and musical data, comprising audio samples. Both of these types were preprocessed with specialized processing to prepare them for the model. We focused on two major columns for preprocessing textual data: “Captions,” which hold descriptive multi-sentence text regarding the music, and “Aspect List,” which specifies specific aspects of the Different NLP techniques were used, such as tokenization, stemming, lemmatization, TF-IDF, and Word2Vec, to refine the text. Then, the “caption” and “aspect list” columns were combined into a new column called “augmented caption,” which was vectorized using Word2Vec in preparation for the model.

B. Models Tried

We experimented with models such as LSTM and Bidirectional LSTM. After downloading 60% of the data, we initiated model training.

C. MusicGen Model

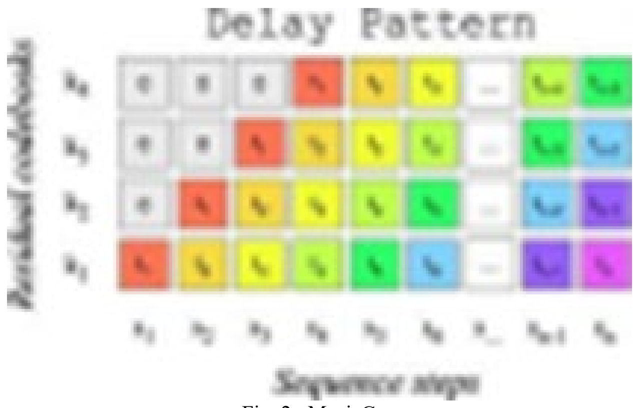

Fig. 2. MusicGen

MusicGen is a Transformer model developed by Jade Copet and colleagues at Meta AI, as introduced in the paper on Simple and Controllable Music Generation. The model pro- duces high-quality music from any text or audio input, doing so through three main stages of operation: text encoding, music decoding, and audio decoding. At text encoding, for example, a frozen text encoder processes text descriptions to produce such hidden-state representations. The hidden states are then used by the MusicGen decoder to predict discrete audio tokens, and audio decoding provides a waveform from the compressed audio model, EnCodec, using these tokens. In contrast to hierarchical models, MusicGen makes an allowance for efficient token interleaving patterns that greatly speed up inference.

To train MusicGen we utilized the Tesla T4 16GB GPU at Colab, initializing a basic environment with the installed pack- ages like Transformers, Datasets, and pre configured check- points at the Hugging Face Hub. It can be set for two music generation modes: greedy and sampling; the sampling is enabled by default because usually, it better generates output. We were able to run experiments within both unconditional and text conditional generation modes. The model generates a 3D Torch tensor shaped as (batch size, channels, sequence length), which is playable or can be saved to produce a .wav file later.

D. Backend Implementation

The backend, built with Django, handles user input, music generation, and content delivery. The workflow starts at home.html, where users submit prompts via POST requests. These prompts are processed and passed to the MusicGen model. While music is being generated, the interface displays a status update. Once complete, result.html presents the output and metadata. Django’s messaging framework and Celery enable real-time updates and asynchronous processing. The generated music is served via Django’s static files. JavaScript manages user interactions like play controls, timer, and progress bar.

E. Implementation for Emotion-Based Generation

In addition to the standard Text-to-Music generation method, our platform introduces an innovative feature: Emotion-to-Music generation. Utilizing a live video feed, this feature detects the user's facial expressions and infers their emotional state. Based on the detected emotion, the system generates relevant AI-based music prompts tailored to match the user's mood.

Fig. 3. Selection between emotion or text

The project utilizes a strong technology stack for full functionality and user interaction. The backend is done through Django, which takes charge of API requests and drives the AI-based emotion analysis pipeline. Django provides excellent communication between the user interface and the emotion detection system. For emotion recognition, this project uses Python libraries like OpenCV for accurate face detection and DeepFace for very precise emotion analysis. These tools work in tandem to deliver reliable and efficient emotion detection capabilities.

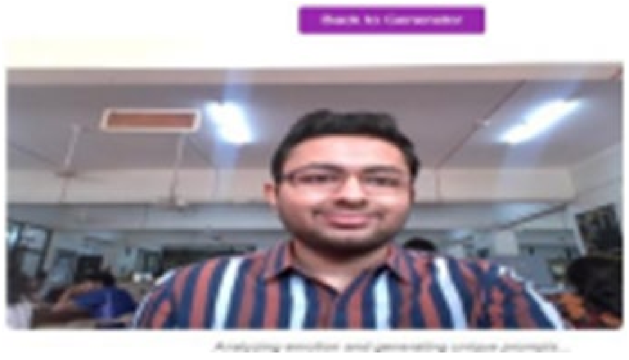

Fig. 4. An image with happy face captured by site

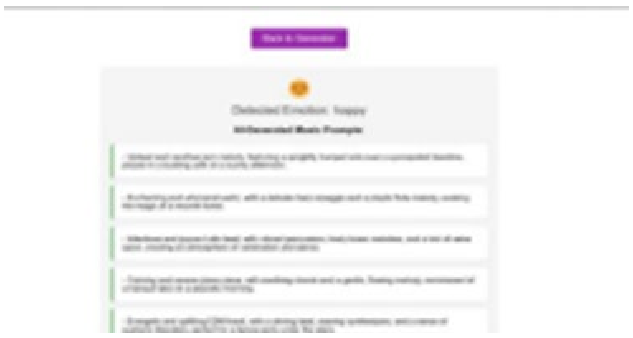

Fig. 5. Prompts from a happy face

The process starts with capturing the user's face via webcam. OpenCV's Haarcascade classifier detects facial features, and DeepFace analyzes the Region of Interest (ROI) to determine the dominant emotion—such as happiness, sadness, or anger. Based on the detected emotion, Django generates a suitable music prompt. For example, a smiling user triggers suggestions like "Upbeat jazz melodies" or "Energetic EDM tracks" to match the positive mood.

6. Technology Stack

The programmatic language used in this project is Python, and some other libraries are used to add on to its functionality. For instance, it uses deep learning techniques from PyTorch; handling musical data is enabled using some MIDI processing libraries. Finally, it uses RNN-LSTM and BERT (Bidirectional Encoder Representations from Transformers) to handle understanding input text. For transforming text to formats readable by machines, there exist tools such as Word2Vec, GloVe, and FastText. Models of GANs (Generative Adversarial Networks), VAEs (Variational Autoencoders), and Diffusion Models are used to create music from textual prompts. For the evaluation, there are tools like MIDITrail, MIDIculous, and MuseScore that are helpful for determining the quality of the music created. The pretrained model of MusicGen is used in the task of generating music.

For the frontend, HTML, CSS, and JavaScript are used to build up the user interface, while the backend is built using Django and Python for handling user requests and processes for generating music.

7. Conclusion

This paper presented the implementation of parameter-based music generation.

Acknowledgement

The authors would like to express their gratitude to Sardar Patel Institute of Technology for providing the necessary resources and support for this research. Special thanks to Prof. Jyoti Ramteke for valuable insights and technical assistance in the development of this platform.

References

[1] Shanmukh Krishna B, Satya Sai Roopa Sree Chiluvuri, Sai Sri Harshini Kosuri, Sai Sree Harsha Chimbili, Sai Sree Harsha Chimbili, "The Application of Long Short-Term Memory and Transformers for Music Generation," 2023.
[2] Ning Zhang, Junchi Yan, Jean-Pierre Briot, "Artificial Intelligence Techniques for Pop Music Creation: A Real Music Production Perspective," 2023.
[3] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, Ying Shan, "Music Knowing LLAMA: Advancing Text-to-Music Generation with Question Answering and Captioning," 2023.
[4] Kenta Suzuki, Jinyu Cai, "A Comparative Evaluation on Melody Generation of Large Language Models," 2023.
[5] Olga Vechtomova, Gaurav Sahu, "LyricJam Sonic: A Generative System for Real-Time Composition and Musical Improvisation," 2023.
[6] Wenzhao Liu, Dechao Meng, "Musical Elements Enhancement and Image Content Preservation Network for Image to Music Generation," 2023.
[7] Chunhui Bao, Qianru Sun, "Generating Music with Emotions," 2022.
[8] Guangwei Li, Xuenan Xu, Lingfeng Dai, Mengyue Wu, Kai Yu, "Diverse and Vivid Sound Generation from Text Descriptions," 2023.
[9] Sneha Adhikari, "Automatic Music Generation of Indian Classical Music Based on Raga," 2023.