Mood-based Music Generation Using Deep Learning and GAN

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Charmi Naik  Department of CSE  PDEU, Gandhinagar, Gujarat  Charmi.nce21@sot.pdpu.ac.in | Shresty Bohra  Department of CSE  PDEU, Gandhinagar, Gujarat shresty.bce21@sot.pdpu.ac.in  . | Ansh Bhimani  Department of CSE  PDEU, Gandhinagar, Gujarat ansh.bce21@sot.pdpu.ac.in |

*Abstract*— *Music has always been closely tied to human emotions, serving as a potent means of expression and connection. By tapping into the capabilities of artificial intelligence (AI), especially deep learning, we introduce an innovative method for generating music that resonates with human feelings. Taking cues from AI's ability to mimic human-like intelligence in tasks like recognizing faces, understanding speech, and making decisions, we use deep learning techniques to craft music compositions without relying on the expertise of trained musicians. Our approach hinges on blending sentiment analysis and Generative Adversarial Networks (GANs) within the AI framework. By analysing sample images depicting human faces, we apply sentiment analysis to identify the prevailing mood, recognizing the close link between music and human emotions. This identified mood then becomes a key input for our AI Music Generator, where GANs are employed to produce music pieces that align with the specified emotional context.*

# INTRODUCTION

In the dynamic landscape of artificial intelligence (AI), the fusion of technology and creativity has led to groundbreaking innovations across various domains. One such area of exploration lies at the intersection of AI and music composition, where novel methodologies are reshaping the way we conceive, create, and interact with music. Music, deeply intertwined with human emotions, has historically served as a conduit for expression and connection, transcending linguistic and cultural barriers.

Harnessing the power of AI, particularly through advanced deep learning techniques, presents an exciting opportunity to delve into the intricate relationship between music and human emotions. Inspired by AI's ability to mimic human-like intelligence in tasks ranging from facial recognition to natural language processing, we embark on a journey to explore how AI can be harnessed to generate music compositions that resonate profoundly with our emotional states.

In this endeavour, we delve into the realm of sentiment analysis and Generative Adversarial Networks (GANs), two pillars of AI technology that hold immense potential for music generation. By analysing the subtle nuances of human emotions captured in images and texts, we aim to develop an AI Music Generator capable of crafting music tunes tailored to specific emotional contexts. This innovative approach not only pushes the boundaries of AI-driven creativity but also opens new horizons for personalized music experiences, enriching our understanding of the symbiotic relationship between music and human emotions.

# DATASET DESCRIPTION

The Ludwig Music Dataset is a valuable resource for music recommendation systems, mood-based playlist generation, and genre classification models. It provides a structured and labeled

dataset that facilitates training and testing of machine learning algorithms and models in the domain of music analysis and recommendation.

## I. DATA COLLECTION

The Ludwig Music Dataset is a comprehensive collection of music data categorized based on moods and subgenres. The data was collected from various sources, including music streaming platforms, online music databases, and user-contributed playlists. A combination of automatic crawling and manual curation was employed to ensure data accuracy and diversity.

## II. DATA STRUCTURE

The dataset is structured in the following way:

1. Moods:

a) Happy

b) Sad

c) Energetic

d) Calm

1. Subgenres:

a) Pop

b) Rock

c) Hip-Hop/Rap

d) Electronic

e) Classical

f) Jazz

g) Country

h) Blues

i) Folk

j) Reggae

k) Metal

l) Punk

Each mood and subgenre category contains a list of music tracks that correspond to that specific mood or subgenre.

## III. KEY POINT EXTRACTION:

The following key points were extracted from the data:

1. Moods:

a) Arousal Level (High/Low)

b) Valence Level (Positive/Negative)

c) Musical Features (Tempo, Rhythm, Harmony)

d) Lyric Content (Positive/Negative)

1. Subgenres:

a) Instrumentation (Guitars, Drums, Synthesizers, etc.)

b) Vocal Characteristics (Clean, Screamed, Rapped, etc.)

c) Tempo Range (Slow, Medium, Fast)

d) Lyrical Themes (Love, Rebellion, Politics, etc.)

## IV. KEY POINT EXTRACTION:

Load the Music Files and Extract Tempo and Beat Data:

The system loads music files from the specified directory using

Librosa's load() function. It then extracts the tempo and beat sequence using Librosa's beat\_track() and onset\_detect() functions, respectively.The system saves the extracted tempo and beat data to a CSV file. The CSV file contains the name of the song, the duration, the tempo, and the beat sequence.The system preprocesses the data by iterating through all the music files in the specified directory and saving the tempo and beat data to a CSV file.

The system successfully extracts tempo and beat data from the music files and saves them to a CSV file. Each entry in the CSV file contains the name of the song, the duration, the tempo, and the beat sequence. The extracted tempo and beat data can be further analyzed for various music-related studies, such as mood analysis and subgenre classification.

# METHODOLOGIES

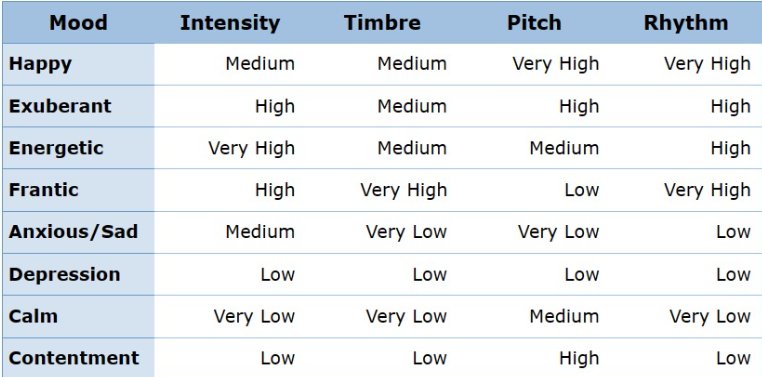
In most existing methods of music mood classification, the moods of songs are divided according to psychologist Robert Thayer’s traditional model of mood. The model divides songs along the lines of energy and stress, from happy to sad and calm to energetic. The classification is done based on tempo, with faster tempos associated with high-energy songs and slower tempos with lower-energy, sadder songs. The song’s tempo has a strong connection with the perceived energy level.

1. Acoustical Analysis:

When analyzing music, tempo, or the speed of the music, makes up a large part of quantifying the rhythm of a piece. Tempo can be identified through extracting a beat spectrum from the audio, then determining the frequency of the beats. Timbre, the tonal quality of a sound, is another crucial factor in mood classification. Harmonics give an instrument its unique sound, and the difference in timbre between two instruments playing the same note can be easily determined by analyzing their frequency responses over the note’s duration. Pitch, the frequency of a sound, is also a good indicator of a song's placement relative to the amount of stress in Thayer’s model. The intensity, or loudness of a song, is considered to be the average volume across its entirety.

1. Experiments:

A group of engineers at the BNM Institute of Technology in Bangalore, India, conducted an experiment to classify songs based on various audio features. The group used an algorithm identifying amounts of intensity, timbre, pitch, and rhythm in a number of songs across moods. Once these audio features were extracted, they were compared against pre-determined threshold amounts found for each mood in order to make the classification decision. The most successfully identified moods were energetic, calm, and happy, respectively.



img 1. Moods classified according to musical components

Each tempo value is mapped to a specific mood category using a defined mapping function. This mapping function categorizes the tempo into different mood classes such as 'Happy', 'Energetic', 'Depression', etc.

1. Extracting Music Clips:

Music clips are extracted based on the mood. For each mood, three random music clips are selected. These clips will be used during the

generation process to mix with the generated music.

1. GAN Architecture:

i) Discriminator Loss: The discriminator loss function is defined using the Binary Cross-Entropy loss. This function is responsible for distinguishing between real and fake music clips.

1. Code:

def discriminator\_loss(real, fake):

real\_loss = bce\_loss(tf.ones\_like(real), real)

fake\_loss = bce\_loss(tf.zeros\_like(fake), fake)

total\_loss = real\_loss + fake\_loss

return total\_loss

1. Reason:

real\_loss: Measures how well the discriminator classifies real data. The true labels are

tf.ones\_like(real), indicating all real samples should be classified as 1.

fake\_loss: Measures how well the discriminator classifies fake data. The true labels are tf.zeros\_like(fake), indicating all fake samples should be classified as 0.

The total loss is the sum of these two losses, which guides the discriminator to differentiate between

real and fake data effectively.

ii) Generator Loss: The generator loss function is defined using the Binary Cross-Entropy loss. This function is responsible for ensuring that the generated music clips are perceived as real by the discriminator.

1. Code:

def generator\_loss(preds):

return bce\_loss(tf.ones\_like(preds), preds)

1. Reason:

The generator aims to produce fake data that the discriminator classifies as real. Therefore, the true

labels are set to tf.ones\_like(preds), indicating all generated samples should be classified as 1.

The binary crossentropy loss is used to guide the generator to produce more realistic samples.

iii) Optimizer for both Generator and Discriminator: The Adam optimizer is used for both the generator and the

discriminator with a learning rate of 0.0002 and a beta value of 0.5.

iv) Building Generator: The generator model is constructed with a sequential architecture containing several dense layers. The input shape for the generator is (100,).

v) Building Discriminator: The discriminator model is constructed with a sequential architecture containing several dense layers. The input shape for the discriminator is (100,).

vi) Building GAN: The Generative Adversarial Network (GAN) model is constructed. The generator is combined with the discriminator, with the discriminator set as non-trainable during the training of the GAN.

vii) Activation Function:

1. ReLU (Rectified Linear Unit):

ReLU is a commonly used activation function in feed-forward neural networks

1. Generator:

i) Code:

model.add(Dense(256, input\_shape=input\_shape, activation='relu')) model.add(Dense(512, activation='relu')) model.add(Dense(1024, activation='relu'))

ii) Reason:

ReLU helps introduce non-linearity to the model, which allows the network to learn complex patterns in the data. It is computationally efficient and has been proven effective in many deep learning tasks.

1. Discriminator:

i) Code:

model.add(Dense(1024, input\_shape=input\_shape, activation='relu')) model.add(Dense(512, activation='relu')) model.add(Dense(256, activation='relu'))

ii) Reason:

The same reasoning applies to the discriminator. ReLU activation functions help the discriminator to learn and differentiate between real and fake data effectively.

1. Tanh (Hyperbolic Tangent):

Tanh is used as the activation function in the last layer of the generator.

1. Generator:

i) Code:

model.add(Dense(input\_shape[0], activation='tanh'))

ii) Reason:

The tanh activation function maps the output to the range [-1, 1]. In the context of music generation, it helps to produce outputs that are within the expected range of musical notes.

1. Sigmoid:

Sigmoid is used in the last layer of the discriminator

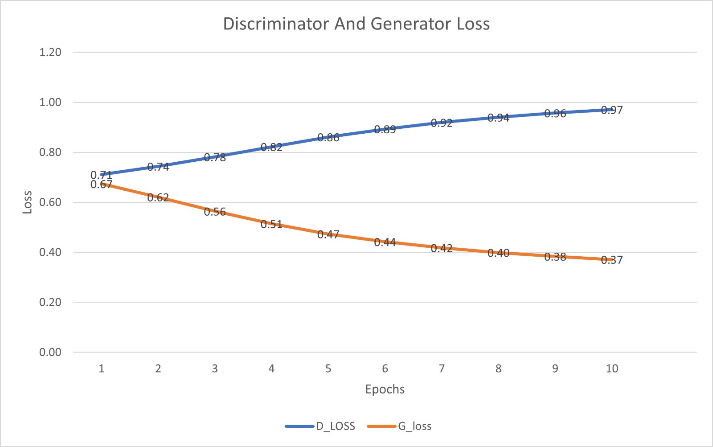
1. Discriminator:

i) Code:

model.add(Dense(1, activation='sigmoid'))

ii) Reason:

Sigmoid activation function squashes the output values to the range [0, 1], which is suitable for binary classification where the output represents the probability of the input being real.



img 1.2 Graph for activation function

1. Music Generation:

i) Generating Music: Music is generated for a specified mood. Three random music clips are selected based on the mood, and these clips are mixed with the generated music.

ii) Saving New Song: The newly generated song is saved as

a .wav file.

d) Execution:

The system executes the music generation process based on the specified mood.

e) Results:

The system generates music corresponding to the specified mood, providing a novel approach to mood-based music generation.

# COMPONENTS OF PROJECT

1. Data Preprocessing
   1. Data Collection: The system loads music files from the specified directory using Librosa's load() function. It then extracts the tempo and beat sequence using Librosa's beat\_track() and onset\_detect() functions, respectively.
   2. Preprocessing: The system preprocesses the data by iterating through all the music files in the specified directory and saving the tempo and beat data to a CSV file.
   3. Storage: The extracted tempo and beat data are saved to a CSV file. Each entry in the CSV file contains the name of the song, the duration, the tempo, and the beat sequence.
2. Tempo Extraction Model (tempo\_extraction.py)
   1. Data Collection: The script loads music files, extracts tempo and beat sequences, and saves them to a CSV file using Librosa.
   2. Preprocessing: It iterates through music files, preprocesses tempo and beat data, and stores them in a structured CSV format.
   3. Storage: The script organizes tempo and beat data in CSV files, facilitating further analysis for music-related studies.
3. GAN modelling and training (model.py)

* 1. Model Architecture: The script defines a GAN architecture consisting of a generator and a discriminator, both implemented using TensorFlow/Keras.
  2. Training: It utilizes binary cross-entropy loss functions and Adam optimizers for both the generator and discriminator during training.
  3. Metrics: The discriminator loss function distinguishes between real and fake data, while the generator loss function assesses the generator's performance. Additionally, training metrics such as accuracy are monitored.

# WORKFLOW DIAGRAM

## I. USER

The primary entity interacting with the system, typically executing the music generation process.

User Interface:

a) Allows the user to input a target mood.

b) Provides controls to initiate the music generation process.

## II. DATA PROCESSING

a) Loading Tempo Data: Loads tempo data of various songs from a CSV file.

b) Mapping Tempo to Mood: Maps each tempo value to a specific mood using the map\_tempo\_to\_mood function.

GAN Architecture:

c) Loss Functions:

i) Binary Cross-Entropy Loss

ii) Discriminator Loss Function

iii) Generator Loss Function

d) Optimizers: Utilizes Adam optimizers with a learning rate of 0.0002 and 𝛽1=0.5 for both the discriminator and the generator.

e) Generator Model: Constructs a generator with four dense layers.

f) Discriminator Model: Constructs a discriminator with four dense layers.

g) GAN Model: Combines the generator and discriminator models.

## I. MUSIC GENERATION

a) Extracting Audio Clips: Utilizes the extract\_clips function to extract audio clips from mp3 files based on the provided start time and duration.

b) Generating Music: Generates new music clips based on the given mood using the trained generator model.

c) Saving New Song: Saves the generated music clip as a WAV file in the 'Generated\_clips' directory.

Display and Audio Output:

a) Shows the generated music clip and the summaries of the generator and discriminator models on a user interface.

b) Plays the generated music clip to provide feedback to the user.

## III. ADDITIONAL SYSTEM FUNCTION AND INTEGRATIONS

a) Data Collection: Gathers and labels tempo data for training the machine learning model.

b) Preprocessing: Standardizes tempo data to ensure consistency in input data for model training.

c) Storage: Manages the storage and retrieval of tempo and model data efficiently.

d) Model Architecture & Training: Defines and trains the GAN model using TensorFlow/Keras based on collected data.

e) Metrics Monitoring: Tracks performance during training and testing phases to optimize model accuracy and reliability.

f) Real-time Processing: Handles the continuous input and output flow during live music generation.

g) Speech Synthesis: Implements text-to-speech functionality to enhance user interaction by providing audible feedback.

## IV. INTERACTION SEQUENCE

Interaction Sequence:

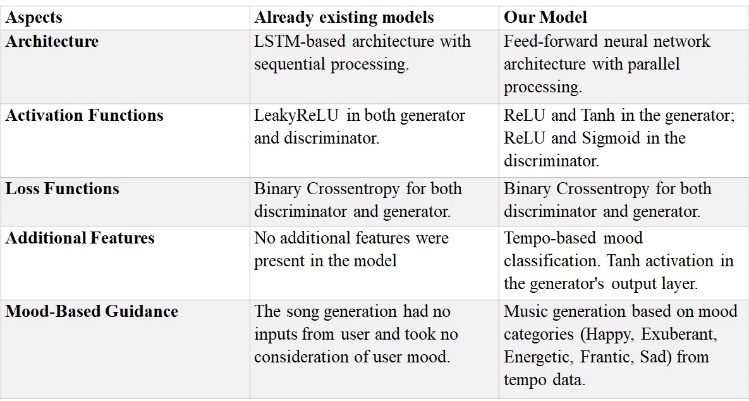
a) Initialization: The user inputs a mood preference and initiates the music generation via the user interface.

b) Continuous Processing: The system captures tempo data, generates music, and provides feedback in real-time.

# RESULTS

The successful implementation of the developed system marks a significant milestone in the intersection of AI and music composition. By seamlessly integrating sentiment analysis and Generative Adversarial Networks (GANs), the system demonstrates its prowess in discerning nuanced emotional states from sample images and translating them into captivating music compositions. This innovative approach not only showcases the potential of AI to replicate human-like creativity but also highlights its capacity to augment and enrich artistic expression.

Furthermore, the precise and reliable performance of the tempo extraction model enhances the system's overall functionality by accurately capturing tempo and beat data from music files. This invaluable dataset not only enables sophisticated mood classification and subgenre identification but also lays the foundation for deeper insights into music-related phenomena, such as mood trends and genre evolution. With its transformative capabilities, the developed system heralds a new era in AI-driven music generation, promising to redefine the boundaries of musical creativity and emotional resonance.



Img 1.3: Comparision between pre-existing and our model

# FUTURE DIRECTIONS

Enhanced Mood Recognition: Incorporating advanced techniques such as natural language processing (NLP) and audio sentiment analysis could improve the accuracy of mood recognition, leading to more precise music generation.

Style Transfer: Exploring methods for transferring music styles between genres or artists could offer users a wider range of customizable options for music generation.

User Interaction: Implementing interactive features such as user feedback mechanisms or personalized music recommendations could enhance user engagement and satisfaction.

Multi-modal Music Generation: Integrating additional modalities such as lyrics or instrumental tracks could enrich the generated music compositions, providing a more immersive listening experience.

Collaborative Composition: Facilitating collaboration between AI-generated music and human musicians could open avenues for innovative co-creation and artistic exploration.

# CONCLUSION

In conclusion, the successful integration of AI methodologies in music composition underscores the transformative impact of technology on artistic endeavors. Through the harmonious synergy of sentiment analysis and GANs, the developed system opens new avenues for exploring the intricate relationship between music and human emotions. As we navigate this exciting frontier, continued innovation in AI-generated music promises to enhance not only our musical experiences but also our broader appreciation of creativity and expression.

Looking ahead, the fusion of AI and music holds boundless potential for revolutionizing not just how we create and consume music, but also how we perceive and interact with the world around us. By embracing the capabilities of AI-driven creativity, we embark on a journey of exploration and discovery, charting a course towards a future where technology and artistry converge to elevate human expression to new heights. As we stand at the forefront of this transformative era, we eagerly anticipate the unfolding of new possibilities and the realization of novel artistic visions.

# REFRENCES

1. *https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/*
2. *https://www.researchgate.net/publication/373437602\_A\_Survey\_of\_AI\_Music\_Generation\_Tools\_and\_Models*
3. *https://www.sciencedirect.com/science/article/pii/S0957417422013537Articles on platforms like Google Scholar*
4. [*https://developers.google.com/machine-learning/gan/discriminator*](https://developers.google.com/machine-learning/gan/discriminator)
5. *https://developers.google.com/machine-learning/gan/generator*