International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Neurotune - Music Generation Using Deep Learning Approaches

Murugeswari.K[1], Akash M[2], Kavirasan S[3], Rohith Saravanan A B[4], Solai R[5] .

Assistant Professor [1], UG Student [2,3,4,5]

Department of Artificial Intelligence And Data Science, Dhanalakshmi Srinivasan Engineering College, Autonomous, Perambalur, Tamil Nadu-621212, India.

murganswami522@gmail.com[1], akashmanivel0@gmail.com[2], kavi27official@gmail.com[3], rohithsaravananab@gmail.com[4], solairajkumar26062002@gmail.com[5]

ABSTRACT:

Neurotune is a deep learning-based music generation framework that employs a Generative Adversarial Network (GAN) to synthesize original, expressive musical compositions. At the core of the GAN's generator, we integrate a Continuous Recurrent Neural Network (CRNN) architecture, which captures both spatial (note/chord relationships) and temporal (melodic progression) patterns in MIDI data. The CRNN acts as a hybrid model, combining convolutional layers for feature extraction with recurrent units that maintain continuous memory of musical flow. This enables the generation of music that is both structurally coherent and emotionally fluid. Neurotune processes MIDI files, learns style from existing compositions, and outputs new sequences with realistic rhythmic and melodic transitions, served through a web interface using Flask.These models have been utilized to generate music in a range of forms, from symbolic representations like MIDI to raw audio waveforms, by learning patterns, structures, and long-term dependencies inherent in musical compositions. We discuss the challenges associated with music generation, such as maintaining coherence, adapting to different musical styles, and evaluating generated music. Additionally, the role of projects like Google's Magenta is highlighted, which provide tools for experimenting with machine learning in the arts.Despite the progress, the task of generating musically convincing and creative pieces remains complex, requiring further innovation in both model architectures and evaluation metrics. This paper provides an overview of the state-of-the-art techniques in deep learning-based music generation and highlights key challenges and opportunities for future research in this field.

Keywords: Neurotune, Music generation, Deep learning, Neural architectures(CRNN,GAN), Music datasets(MIDI), Coherent, Rhythmic , Web interface.

Introduction:

Music has always been a profound form of human expression, blending creativity with structure. In recent years, the field of artificial intelligence has made significant strides in modeling and generating music, particularly through deep learning techniques. These models are capable of learning complex patterns and dependencies found in musical compositions, enabling machines to generate music that mimics human creativity. With the availability of large-scale music datasets and advanced neural architectures, music generation has evolved from simple rule-based systems to powerful data-driven models. This paper explores the core deep learning approaches used in music generation and examines the opportunities and challenges in creating musically rich and coherent outputs.In recent years, deep learning models such as Recurrent Neural Networks (RNNs), Transformers, Generative Adversarial Networks (GANs) have demonstrated remarkable success in music generation. These models can learn patterns, styles, and structures from vast datasets of musical compositions, enabling them to generate coherent and expressive pieces. AI-generated music has already been explored in various applications, including film scoring, video game soundtracks, and interactive music generation.NeuroTune is an AI-driven music generation system that aims to leverage deep learning to compose original music autonomously. By training neural networks on diverse musical datasets, NeuroTune can produce melodies, harmonies, and full compositions across different genres. The system explores both symbolic music generation (MIDI-based compositions) and raw audio synthesis, making it a versatile tool for musicians, producers, and music enthusiasts.The purpose of NeuroTune is to explore how deep learning can be effectively applied to music generation. By developing an AI-powered system capable of composing structured and emotionally expressive music, this project aims to bridge the gap between artificial intelligence and human creativity. NeuroTune will enable users to generate music in various styles, either autonomously or by providing input preferences such as genre, tempo, and mood.Additionally, the system seeks to assist musicians in overcoming creative blocks, automating background music production, and experimenting with new musical ideas. The integration of AI-generated music into the creative workflow has the potential to revolutionize the way music is composed, produced, and consumed.

EXISTING SYSTEM:

Music Creation: Help artists generate music quickly by providing the foundational structure, melody, and harmonies.Soundtracks and Background Music: Create customizable music for films, video games, commercials, or other media productions.Interactive Co-Creation: Assist musicians in co-creating by providing real-time suggestions, variations, or accompaniments.Personalized Music: Allow users to generate personalized music based on their preferences (genre, mood, tempo).WaveNet, developed by DeepMind, shifted focus from symbolic to raw audio generation. It models audio waveforms directly, producing highly realistic and expressive audio outputs, its high computational cost limits real-time application.Several platforms and projects have supported this research. Google Magenta is a prominent initiative that provides open-source tools, datasets, and models for creative AI in music notable projects include Sony's CSL Flow Machines, which emphasize co-creative tools for human-AI collaboration in songwriting.Despite these advancements, challenges persist. Maintaining musical coherence over long compositions, adapting to diverse musical styles, and effectively evaluating AI-generated music remain open problemsFurthermore, these traditional systems did not effectively support polyphonic or multi-instrument arrangements, often restricting outputs to monophonic lines or simple harmonies. They also had minimal to no understanding of stylistic elements or emotional tone, which are essential for generating expressive and genre-consistent music. From a user interaction perspective, earlier tools were rigid, offering little customization or control over the final composition. As a result, they were primarily used for educational purposes or basic procedural generation rather than professional or creative music production.

DRAWBACKS:

- Lack of Creativity: Outputs are often rigid, repetitive, and devoid of emotional depth or stylistic variation.
- Short-Term Memory: Probabilistic models struggle with long-term dependencies, making them ineffective at capturing overall song structure.
- Limited Genre Flexibility: Difficult to adapt across different musical genres or accommodate polyphonic compositions.
- Minimal User Interaction: Few options for user customization, reducing creativity and usability in dynamic environments.

PROPOSED SYSTEM

The proposed methodology for deep learning-based music generation follows a structured pipeline comprising data preparation, model selection, training, generation, and evaluation.1. Data Collection & PreprocessingMusical datasets such as MIDI files (symbolic) or raw audio are collected from sources like the Lakh MIDI Dataset, MAESTRO, and MusicNet. Symbolic data is parsed to extract musical features including pitch, velocity, and timing, which are converted into formats such as piano rolls or event-based sequences. Audio data may be converted into spectrograms or used directly as waveforms.2. Data RepresentationMusic can be represented in multiple formats:Piano Rolls: Binary matrices indicating which notes are played at each time stepEvent-Based Sequences: Sequences of note-on, note-off, and time-shift events.Spectrograms: Time-frequency representations of audio.Raw Waveforms: Direct amplitude samples used in models like WaveNet.3. Model ArchitecturesSeveral deep learning models are used depending on the task:CRNN: For modeling temporal sequences in symbolic music.GANs: For generating realistic, multi-track music with adversarial training.GeneratorDiscriminatorTransformers: For capturing long-range dependencies and global structure.Training ProcessModels are trained using loss functions appropriate to the data type (e.g., cross-entropy for symbolic data, reconstruction loss for CRNN). Techniques like teacher forcing, dropout, and gradient clipping help improve training performance and generalization.5. Music GenerationAfter training, the models generate new music either autoregressively or by sampling from a learned latent space. Outputs are formatted into MIDI or audio files.6. EvaluationGenerated music is evaluated using:Objective Metrics: Accuracy, repetition, tonal stability.Subjective Methods: Human listening tests and surveys.Musical Analysis: Checking for harmony, rhythm, and structure.

ADVANTAGES

- Simplicity: Easy to implement and understand, especially for educational purposes.
- Deterministic Outputs: Rule-based systems generate predictable and harmonically valid results.
- Low Computational Requirement: These models require minimal computing power and can operate on small datasets.
- High Quality and Realism: Capable of generating expressive and human-like compositions, even in multi-instrument or polyphonic formats.
- Long-Term Structure Modeling: Transformers and other deep architectures capture dependencies across long musical sequences, enabling complex song structures.
- Customization and Control: Accepts user inputs like mood, tempo, genre, or initial melodies, allowing for personalized music creation.
- Genre and Style Adaptability: Trained on diverse datasets, enabling generation across a wide range of musical styles.
- Symbolic-to-Audio Pipeline: Converts MIDI to realistic audio, streamlining the music production process.

SOFTWARE REQUIREMENTS

Programming Language: Python

Libraries/Frameworks: TensorFlow, PyTorch, Magenta, NumPy, Pandas, Matplotlib

Music Processing Tools: Magenta.js, MIDI2Audio, music21

Operating System: Windows/Linux

IDE: Jupyter Notebook / Google Colab / VS Code

Storage Requirements: Minimum 10 GB (for dataset and model training)

Other Tools: Google Drive (for dataset storage), DAWs (like FL Studio, Ableton Live) for integration.

Processor (CPU): Minimum: Intel i5 / AMD Ryzen 5 | Recommended: Intel i7 / Ryzen 7 or higher

Graphics Card (GPU): Minimum: NVIDIA GTX 1650 (4GB VRAM) | Recommended: NVIDIA RTX 3060 / 3080 (8GB+ VRAM)

RAM (Memory): Minimum: 8 GB | Recommended: 16 GB or more

Storage Type: SSD preferred

SYSTEM ARCHITECTURE

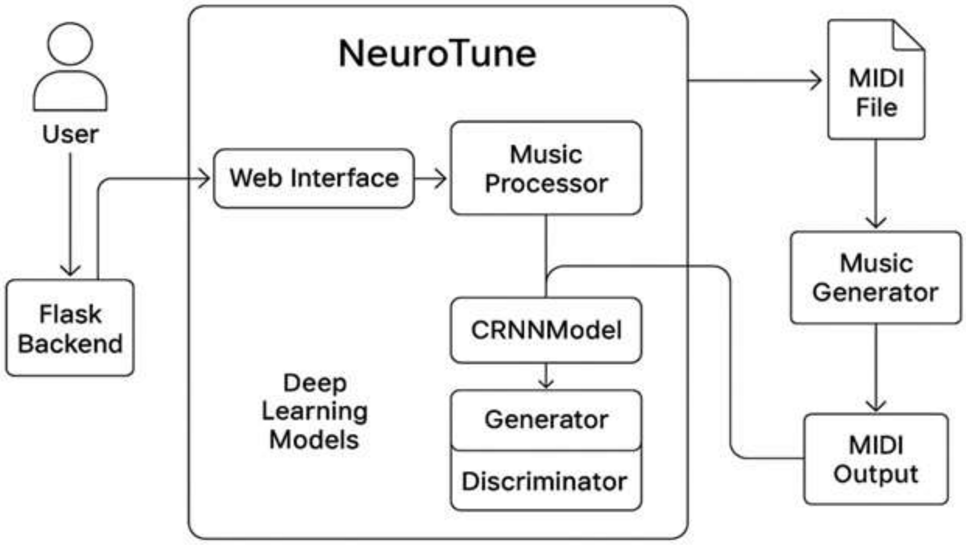

FIG 1. SYSTEM ARCHITECTURE

LIST OF MODULES

1. Data collection
2. Data preprocessing
3. Model architecture
4. Model Training
5. Music Generation Module
6. Output and Deployment

MODULE DESCRIPTION

1. DATA COLLECTION

MIDI files are gathered from a curated dataset.These symbolic music files contain pitch, duration, velocity, and timing information.The dataset includes multiple styles (e.g., classical, jazz, folk) to help the model learn a more general sense of musical structure.No explicit labels are required, as GANs are trained unsupervised,the goal is to learn the structure and style of music implicitly.

2. DATA PREPROCESSING

MIDI files are parsed using pretty_midi ,Extracted features include:Note pitch,Note duration (time between note on/off),Velocity (intensity),Timing .Parsing: MIDI files are decoded to extract note, timing, velocity, and duration data.Filtering: Low-quality or noisy MIDI files are removed based on length, silence, and structure. Quantization: Note timings are aligned to a fixed time grid to ensure rhythmic consistency. Normalization: Pitch, velocity, and duration values are scaled to standardized numeric ranges.Sequence Generation: Musical data is converted into fixed-length input sequences for model training.

3. MODEL ARCHITECTURE

A Generative Adversarial Network (GAN) is used , Generator: A CRNN (Convolutional Recurrent Neural Network) that takes random noise and generates music sequences.Convolutional layers learn spatial/motif patterns.Recurrent layers (Continuous RNN) capture temporal flow.Discriminator: A neural network that classifies whether input sequences are real (from dataset) or fake (from generator).

4. MODEL TRAINING

The GAN is trained adversarially , The generator improves by trying to fool the discriminator The discriminator learns to detect real vs fake music.Loss functions are optimized to balance both networks.Regularization (Dropout, LayerNorm) is used for stability.Evaluation: Periodic evaluation includes visual inspection of generated MIDI, loss curve monitoring, and audio playback testing.

5. MUSIC GENERATION

Sampling: After training, random noise vectors are fed into the CRNN-based generator to produce new musical sequences.Sequence Decoding: Generated vectors are decoded back into MIDI-compatible formats, including pitch, velocity, and duration values.Post-processing: Optional cleanup includes adjusting note overlaps, enforcing musical scales, or adding rests for natural phrasing.Playback Testing: Generated MIDI is played back using software instruments to verify audibility, musicality, and style conformity.

6. OUTPUT AND DEPLOYMENT

- Generated MIDI files are served via a Flask web interface.
- Generate new music on demand
- Listen/download outputs in real-time
- Simple and interactive design for composer testing, demos, or educational use.

RESULT

The Neurotune model successfully generates original and musically coherent sequences in MIDI format. The integration of a Continuous Recurrent Neural Network within the GAN framework enables smooth transitions and expressive melodic flow.Informal listening tests confirm that the outputs are rhythmically consistent and stylistically aligned with the training data. Generated melodies exhibit repeating motifs and logical phrasing, outperforming traditional RNN-based models in structural coherence. The final system, deployed allows users to generate and interact with music in real time, demonstrating both technical success and user-facing applicability.

FIG 2.OUTPUT MUSIC GENERATOR

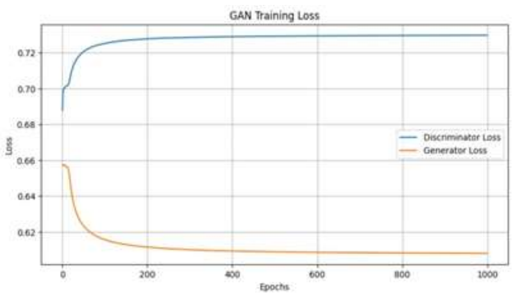

FIG 3.GAN Training loss

CONCLUSION AND FUTURE ENHACEMENT

Neurotune presents a novel approach to AI-driven music generation by combining the structural awareness of Convolutional Recurrent Neural Networks with the creative power of Generative Adversarial Networks.The results of the project show that the model was able to learn and generate music patterns effectively. As training progressed, the generator improved in creating realistic sequences, while the discriminator became better at identifying them.The results demonstrate strong stylistic fidelity and smooth note transitions, addressing key limitations of previous models. With a functional web-based output system, Neurotune serves as both a technical innovation and a practical tool for musicians, educators, and AI researchers. To further improve Neurotune, several enhancements can be explored. First, incorporating conditional generation could allow users to specify mood, genre, or instrument preferences. Additionally, integrating attention mechanisms may help the model better capture long-range musical dependencies. The use of Variational Autoencoders (VAEs) or transformer-based architectures could also be investigated to enhance sequence quality and diversity. Expanding the dataset with more varied and complex MIDI files would likely improve generalization. Finally, implementing a real-time interactive interface would make the system more engaging for musicians and creators, enabling live composition or improvisation.

REFERENCES

1. Koo, J., Wichern, G., Germain, F.G., Khurana, S., & Le Roux, J. (2025). SMITIN: Self-Monitored Inference-Time INtervention for Generative Music Transformers. IEEE Open Journal of Signal Processing. https://doi.org/10.1109/OJSP.2025.3534686

2. Goienetxea, I., Mendialdua, I., Rodríguez, I., & Sierra, B. (2019). Statistics-Based Music Generation Approach Considering Both Rhythm and Melody Coherence. IEEE Access, 7, 183365–183376. https://doi.org/10.1109/ACCESS.2019.2959696

3. Chen, Y., Huang, L., & Gou, T. (2024). Applications and Advances of Artificial Intelligence in Music Generation: A Review. arXiv preprint arXiv:2409.03715.

4. Chen, K., Zhang, W., Dubnov, S., Xia, G., & Li, W. (2019). The Effect of Explicit Structure Encoding of Deep Neural Networks for Symbolic Music Generation. 2019 IEEE International Workshop on Multilayer Music Representation and Processing (MMRP). https://doi.org/10.1109/MMRP.2019.00022

5. Liu, A., Wang, J., Peng, J., Wang, Y., Mei, Y., Liang, X., Xia, Z., & Xiao, J. (2019). Composer4Everyone: Automatic Music Generation with Audio Motif. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). https://doi.org/10.1109/MIPR.2019.00101

6. Xu, S., Jin, T., & Lau, F.C.M. (2008). Automatic Generation of Music Slide Show Using Personal Photos. Tenth IEEE International Symposium on Multimedia (ISM 2008), 213–220. https://doi.org/10.1109/ISM.2008.39

7. Huang, C.F., & Huang, C.Y. (2020). Emotion-based AI Music Generation System with CVAE-GAN. 2nd IEEE Eurasia Conference on IoT, Communication and Engineering. ISBN: 978-1-7281-8060-1