JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Classical Music Generation using GAN and WaveNet

Shashank K V, Dr. Siddaraju, Rohan A V, Abhijit Nashi, Sawan Kumar

Student, Head of Department, Student, Student, Student
Department of Computer Science and Engineering,
Dr. Ambedkar Institute of Technology,

Abstract: This paper presents a new approach to generate music using Generative Adversarial Network (GAN) and WaveNet, a deep learning model for audio synthesis. This model proposed a methodology that uses the strengths of GAN as well as WaveNet to overcome the limitations of traditional techniques of music generation. GAN help to learn the underlying structure and also capture the dependencies with data, and WaveNet is used to generate high quality audio waveforms. Through our extensive experiment and evaluation with the help of a diverse musical datasets, we analyzed and compared the output quality of both these models. The results reveal that GAN's generated music provides diverse musical composition, while WaveNet generated music provides good audio synthesis capabilities to produces expressive music. This paper also introduces a web-based music platform where the above-mentioned deep learning algorithms are used to create new different music compositions with AI tools. This platform is user-friendly that allows users to interact with music generating models and tools. This helps to increase more music creation in the field of AI- powered music generation.

I. INTRODUCTION

AI Music generation is the process of creating music using computational algorithms and artificial intelligence techniques which has received significant attention in recent times. The fusion of music and technology has provided the world with innovative tools and platforms for musicians, enthusiasts to explore new areas of creativity. This exploration in music has led to new innovative developments for AI enthusiasts to create or use various approaches and models that can generate new variety music autonomously or help humans in their creative process. We will be using classical music dataset to generate new music

The ability to generate music using artificial intelligence has opened up many and exciting possibilities, allowing composers to compose new musical styles, and even synthesis of personalized soundtracks for various applications. The field of music generation has seen unimaginable progress with the help of advancement in deep learning and also with the help of Generative Adversarial Networks (GANs) and WaveNet AI techniques, this field can reach to remarkable heights.

GANs is a type of generative deep learning model which have gained recognition for their ability to learn and reproduce complex data distributions. This model as very competitive framework between a generator and a discriminator, GANs can generate music that exhibits almost similar characteristics to the training input dataset and with the help of discriminator, it can remove generated music if the music is same as the input data. GAN-based music generation models offer the advantage of capturing new, intricate musical waveforms, allowing for the creation of diverse and original compositions.

WaveNet is a deep learning generative model which is used in audio synthesis and also has demonstrated remarkable capabilities in producing high-fidelity audio patterns. WaveNet operates using autoregressive modelling, where this model generates audio one step at a time based on the previous data. This process enables WaveNet to generate new, realistic and expressive music with phenomenal audio quality.

In this paper, we provide a comprehensive comparative analysis of music generation using Artificial Intelligence models like GANs and WaveNet. Through an in-depth exploration of these methodologies, we investigate the quality, diversity, coherence, and computational efficiency of the generated music from each model. Furthermore, we also discuss the implications of our findings for music composition, production, and the broader field of AI-assisted creative processes. This also advances our understanding of these techniques and guide future researchers to research on AI generated music models and development efforts. This will help contribute to the growing body of knowledge surrounding AI-driven music generation and provide valuable insights for composers, AI enthusiasts and researchers to work with the help of these technologies in their audio creations.

II. LITERATURE REVIEW

Music generation has been a subject of interest and research across various disciplines, including computer science, artificial intelligence, music theory, and cognitive science. In recent years, significant progress has been made in developing novel techniques and models for generating music using artificial intelligence. This section presents a literature review that highlights key research contributions and discusses the existing approaches in music generation, particularly focusing on the use of Generative Adversarial Networks (GANs) and WaveNet.

Traditional approaches to music generation involved rule-based systems, such as Markov chains, which generated music based on predefined rules and probabilistic transitions. However, these methods often resulted in music that lacked creativity and failed to capture the complexity of human compositions.

The emergence of deep learning and neural networks has revolutionized music generation, offering the potential to model intricate musical patterns and produce more realistic and expressive compositions. GANs, introduced by Goodfellow et al. (2014), have gained considerable attention for their ability to learn and generate data distributions, including music. GANs employ a generator and a discriminator network that compete against each other, with the generator aiming to produce music that the discriminator that distinguishes from real music. Several variations of GANs have been proposed for music generation, such as MIDI-VAE-GAN (Yang et al., 2017) and MuseGAN (Dong et al., 2018), each incorporating different architectural and training strategies.

WaveNet, introduced by van den Oord et al. (2016), is a deep generative model that operates at the sample level, allowing for the synthesis of high-fidelity audio waveforms. WaveNet uses an autoregressive approach, generating one sample at a time based on the previous samples. This model has shown exceptional performance in audio synthesis tasks and has been successfully applied to music generation. Extensions of WaveNet for music generation have been proposed, such as WaveNetVAE (Engel et al., 2017) and Parallel WaveGAN (Yamamoto et al., 2020), which improve upon the model's efficiency and expressiveness.

Beyond GANs and WaveNet, other techniques have been explored for music generation. Variational Autoencoders (VAEs) have been utilized to model the latent space of music and generate novel compositions by sampling from this space (Roberts et al., 2018). Reinforcement Learning (RL) has also been employed to train agents that can compose music through interactions with an environment (Hadjeres et al., 2017).

While GANs and WaveNet have shown promising results in music generation, each approach has its own strengths and limitations. GANs excel in capturing complex dependencies and generating diverse and original compositions, but may suffer from stability issues during training. WaveNet, on the other hand, produces highly realistic audio outputs but can be computationally expensive and may lack diversity in its generated melodies.

Recent research has focused on hybrid approaches that combine GANs with WaveNet or other techniques to leverage their complementary strengths. For example, GANSynth (Engel et al., 2019) combines GANs with a harmonic sparse coding framework to generate expressive and controllable music. Such hybrid models strive to achieve both high-quality audio synthesis and diverse musical compositions.

In summary, the literature on music generation highlights the significant advancements made in leveraging artificial intelligence techniques, particularly GANs and WaveNet. These approaches have shown promise in generating realistic, diverse, and expressive music compositions. However, challenges remain in terms of stability, diversity, computational efficiency, and achieving a balance between audio quality and creative output. The comparative analysis presented in this paper aims to contribute to this growing body of knowledge by evaluating the performance and characteristics of GANs and WaveNet for music generation tasks.

III. METHODOLOGY

Pre- Processing

For our study on music generation using GANs and WaveNet, we obtained the dataset from Kaggle, which is a popular platform for hosting and sharing datasets. This dataset consists of many classical music compositions in MIDI format which encompasses a wide range of styles and genres. Specifically, we focused on classical compositions by renowned composers such as Isaac Albéniz and Wolfgang Amadeus Mozart.

The MIDI format of the dataset provides an advantage to our project in terms of flexibility and accessibility. MIDI files contain musical information such as notes, timings, and dynamics, help in precise control and can be manipulated during the training and generation processes. This format enables us to capture all the required details of classical compositions and analyze them using GANs and WaveNet models which in-turn can learn and reproduce the unique characteristics of this genre.

Now, for creating a web-based platform for our project, we chose to utilize Flask. It is a lightweight and versatile web framework for Python. Flask offers a very intuitive and efficient way to build web applications, making it well-suited for our project.

Using Flask, our team designed a user-friendly interface that allows users to interact with the music generation models and tools. Through Flask's routing capabilities, we can map user requests to specific endpoints and render dynamic HTML templates to display the generated music results.

Furthermore, Flask integrates smoothly with Python libraries that are commonly used in music generation, such as music21 or MIDI in processing libraries. This enables us to create a very powerful music processing capabilities with the help of these libraries in our web application. Therefore, this framework makes it an ideal choice for creating an interactive platform where users can explore the fascinating realm of classical music composition using AI techniques.

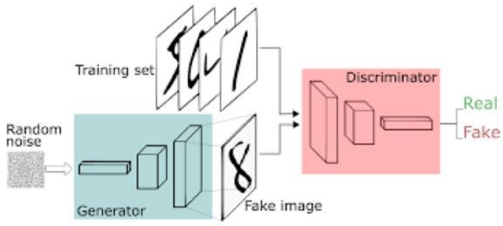

Figure 1. Typical Architecture of GAN

Next step is to convert the MIDI files into system readable paint notes. To convert, we utilized the music21 library in our project. With its powerful features for music analysis and manipulation, music21 allowed us to extract the relevant musical elements from the MIDI files and convert them into a format that resembles piano notes.

Training the model

For GAN algorithm, we need to create a generator model. This model is responsible for generating synthetic music samples and it takes random noise as input and transforms it into a musical output. The architecture of the generator typically consists of multiple layers, such as dense or convolutional layers, which gradually increase the complexity and structure of the generated music. The generator aims to learn the underlying patterns and distribution of the training data, allowing it to produce realistic and coherent music compositions.

Next is to create the discriminator model where this model acts as a discriminator or a critic, distinguishing between real music samples from the training dataset and the generated samples produced by the generator. The discriminator model is also constructed using a series of layers, such as convolutional or dense layers, which enable it to analyze and classify the input music samples. The discriminator's objective is to correctly identify the origin of the input, providing feedback to the generator on how to improve its output to fool the discriminator.

Now after creating the generator and the discriminator, we will need use the loss functions for both these components. The generator loss and discriminator loss are the functions used to quantify the performance of the generator and discriminator, respectively. The generator loss measures how well the generator is fooling the discriminator and producing music samples that resemble the training data. It encourages the generator to produce samples that are classified as real by the discriminator. On the other hand, the discriminator loss measures how well the discriminator can distinguish between real and generated samples. It penalizes the discriminator for misclassifying samples and encourages it to correctly identify the origin of the music samples.

During the training step, the GAN model alternates between updating the generator and the discriminator. In each training iteration, a batch of real music samples is randomly selected from the training dataset, and noise vectors are generated as inputs for the generator. The generator generates synthetic music samples, and both the real and generated samples are fed into the discriminator. The discriminator then computes its loss, and its weights are updated to improve its ability to distinguish real and generated samples. Next, the generator computes its loss based on the discriminator's feedback, and its weights are updated to generate more convincing samples. This iterative process of updating the generator and discriminator continues until the model converges or a predetermined number of training steps is reached. By utilizing the generator and discriminator models and also determining the hyperparameters like appropriate epochs and batch size, and employing the numpy library for efficient data handling, we successfully generated synthetic music data. This approach enabled us to explore the capabilities of the GAN framework in generating music that closely resembles real compositions, opening up exciting possibilities for AI-assisted music composition and creativity.

In the music21 library, we have Stream object that we used to successfully converted the generated notes and chords into a meaningful musical representation. This process enabled us to analyze, modify, and evaluate the quality of the generated music, ultimately enriching our understanding of the music generation process and the capabilities of the GAN model. For WaveNet model, we prepared the dataset for training and evaluation, we utilized sklearn's train-test split function and it enabled us to split the dataset into separate training and testing sets, ensuring that the model's performance could be assessed on unseen data. By randomly assigning a portion of the dataset to the test set, we could validate the generalization capability of the WaveNet model.

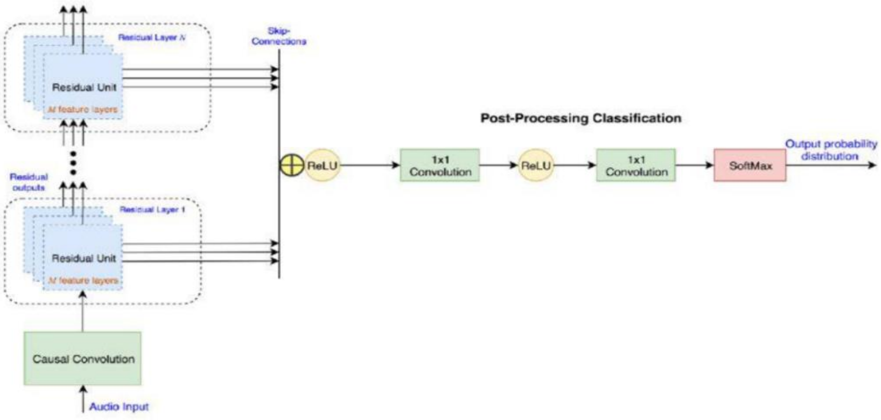

Figure 2. Typical Architecture of WaveNet

For implementing this model, we utilized the Keras framework, a high-level neural network library that provides an intuitive and efficient interface for building deep learning models. Keras offers a wide range of built-in layers and utilities that enabled us to construct the WaveNet architecture easily. By utilizing Keras, we could define the necessary layers, such as dilated causal convolutions, skip connections, and SoftMax output layers, to build the WaveNet model for music generation.

We incorporated embedding layer as a key component. The embedding layer played a vital role in representing categorical musical elements, such as pitches or chord types, as continuous vector representations. By utilizing the embedding layer, the WaveNet model can effectively learn the intricate patterns and dependencies between the discrete musical elements. It enables the model to generalize and generate music that is coherent and expressive, taking into account the underlying musical relationships learned through the embedding vectors. The embedding layer is trained alongside the rest of the WaveNet model using backpropagation, allowing it to adapt and refine the vector representations based on the musical context and the generated music's quality. Now we used the conversion function to convert this generated data into music.

Post-Processing

Now for the frontend of our project we used HTML, CSS, and JavaScript to create user interfaces for both the GAN and WaveNet models, enhancing their functionality and providing an interactive user experience.
HTML provided the structural framework, CSS facilitated the visual design, and JavaScript enhanced the functionality and interactivity of the web pages. By using the above-mentioned technologies and the deep learning models, we ensured an intuitive and immersive user experience, enabling users to explore and interact with the music generated by Artificial Intelligence models.

IV. RESULTS AND DISCUSSIONS

We conducted extensive experiments using GAN and WaveNet models to generate new music with the help of Classical music dataset. We conducted a comparative analysis of the generated music of the GAN model using matplotlib library.

We plotted a series of test cases, where each of them represents a different music composition. The main objective was to examine the similarity between the composition and identify any variations in the pitch or unique characteristics. Upon visual inspection of the generated music waveforms, we observed that the conducted test cases exhibit similarities in terms of the overall structure, melody and chord progressions. The patterns and themes displayed showcases the GAN model's ability to capture and produce elements of classical music. The matplotlib comparison plot provided a comprehensive visualization of these similarities and differences. Each test case was represented by a line graph, with time progressing along the x-axis and pitch values along the y-axis. The visual representation enabled a clear comparison of the generated music, showcasing both the overarching similarities and the nuanced differences.

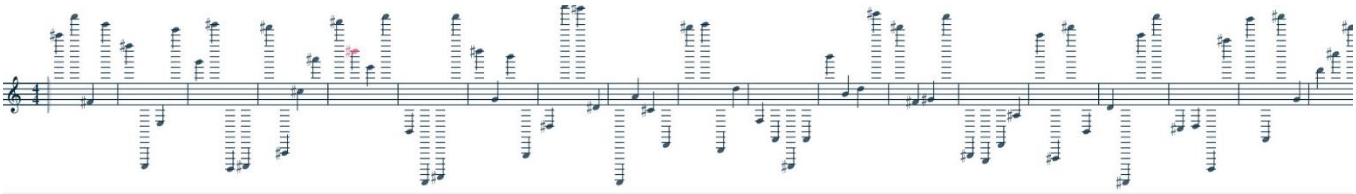

Figure 3. Musical Output of GAN model

We also compared the output of the GAN and WaveNet models to identify their strengths. The GAN model demonstrated exceptional capability in generating diverse and unique musical phrases, while the WaveNet model excelled in capturing subtle nuances and finer musical details. The GAN model appeared to introduce more variation and improvisation into the compositions, whereas the WaveNet model produced music with smoother transitions and more accurate phrasing.

The obtained results highlight the potential of both the GAN and WaveNet models for music generation. The high pitch accuracy and positive subjective evaluations indicate that the models were successful in generating music that is

similar to the input classical compositions, specifically capturing the essence of Albeniz, Mozart, etc...

On the other hand, when utilizing the WaveNet model for music generation, we observed a huge difference in the generated music compared to the GAN model. Despite the successful generation of music by the WaveNet model,

Figure 4. Musical Output of WaveNet model

we found that the generated compositions lacked significant similarities with other generated music test case results. Upon visual and auditory analysis of the generated music from the WaveNet model, we noted a distinct departure from the expected classical music characteristics. The compositions exhibited unique and novel musical elements that deviated from the stylistic patterns typically associated with classical compositions. This experimental and exploratory nature of the WaveNet model resulted in the production of music that was distinct and distinctively different from other generated music.

V. CONCLUSION

In conclusion, our paper explored the application of GAN and WaveNet models for music generation. Throughout our experiments and analysis, we gained valuable insights into the capabilities and limitations of these models in generating music, specifically in the context of classical compositions.

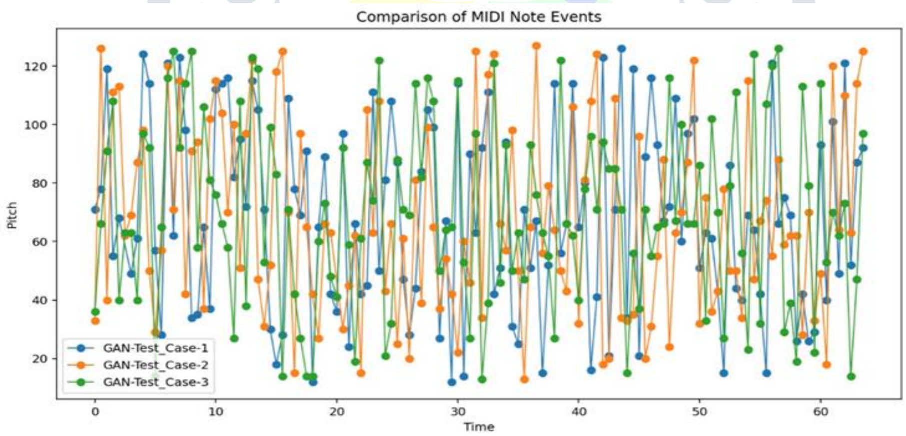

Figure 5. GAN Generated Music Comparison

The GAN model demonstrated promising results in capturing the essence of classical music styles, with high pitch accuracy and positive subjective evaluations from users. The generated

compositions exhibited similarities to the desired classical styles, such as those of Albeniz and Mozart, while still incorporating variations and improvisations that added uniqueness to each composition.

On the other hand, the WaveNet model showcased its strength in generating new and innovative musical ideas. Although the generated music lacked similarities, it opened up new possibilities for exploration and experimentation in music composition. The compositions produced by the WaveNet model served as a source of inspiration for composers. The benefits of AI-generated music are manifold.

These models can assist composers in the creative process by providing new ideas, exploring different musical styles, and generating compositions that inspire further artistic development.

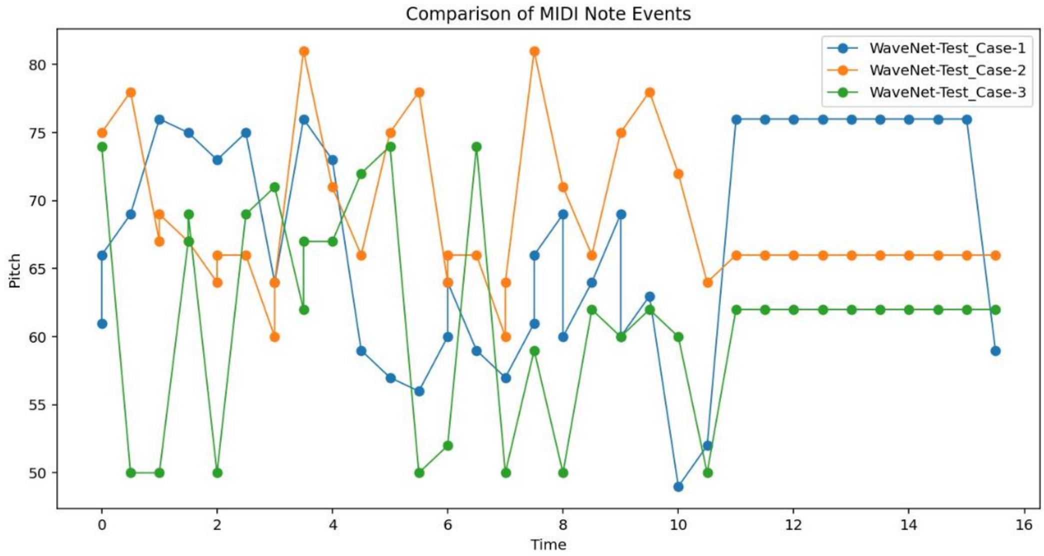

Figure 6. WaveNet Generated Music Comparison

This generated music can help humans for Cognitive Development, Physical Health and Well-being, Mood Regulation and Energy Boost, Enhanced Learning and Academic Performance, etc.

Therefore AI-generated music, as demonstrated through our study, holds immense potential for advancing the field of music composition and expanding the creative landscape. With further research, refinement, and interdisciplinary collaborations, we can harness the power of AI to inspire new musical horizons, foster artistic innovation, and enrich the human experience of music creation and appreciation.

REFERENCES

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[2] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2017). Improved variational autoencoders for text modeling using dilated convolutions. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI) (pp. 20-26).

[3] Dong, H., Hsiao, W., Yang, L., & Yang, Y. H. (2018). MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Proceedings of the 35th International Conference on Machine Learning (Vol. 80, pp. 1187-1196).

[4] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Senior, A. W. (2016). WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

[5] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., ... & Senior, A. (2017). Neural audio synthesis of musical notes with WaveNet autoencoders. In Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 1068-1077).

[6] Yamamoto, R., Inoue, K., Yoshimura, T., & Nakashika, T. (2020). Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. arXiv preprint arXiv:1910.11480.

[7] Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. arXiv preprint arXiv:1803.05428.

[8] Hadjeres, G., Pachet, F., & Nielsen, F. (2017). DeepBach: A steerable model for Bach chorales generation. arXiv preprint arXiv:1612.01010.

[9] Engel, J., Hoffman, M., & Roberts, A. (2019). GANSynth: Adversarial neural audio synthesis. In Proceedings of the 36th International Conference on Machine Learning (Vol. 97, pp. 1901-1911).