Music Composition Using Deep Learning

Neil Bugeja
Computing and Business (Hons.)
University of Malta
neil.bugeja.19@um.edu.mt

***Abstract*—With recent developments in deep learning and artificial intelligence, computer-based music creation has witnessed remarkable advancements. This study focuses on evaluating the performance of Music Composition (MC) software by analysing the quality of the composed samples, specifically addressing the challenge of genre alignment. The goal is to establish an objective and reliable method for assessing the quality of the compsoed samples by incorporating a Music Genre Recognition (MGR) model.**

To achieve this objective, a Convolutional Neural Network (CNN) model is utilised for genre classification, using 40 Mel Frequency Cepstral Coefficients (MFCCs) as input features. The CNN model achieves a test accuracy of 70%, enabling the objective determination of genre alignment in the composed samples.

The findings reveal when evaluating Jukebox, an open-source music composer developed by OpenAI, the adherence to the requested genre is influenced by the specified parameters. When the artist and genre are from the same grouping (ex: Artist: Mozart, Genre: Classical), approximately 2/3 of the compositions demonstrate a positive genre match. However, when the artist and genre oppose each other (ex: Artist: Mozart, Genre: Pop), only 42% of the samples match the requested genre, with 58% aligning with the artist's genre instead. Moreover, when lyrics from a different genre are provided, none of the test cases yields a positive genre match.

This study highlights the significance of genre alignment in assessing the quality of computer-generated music and emphasises the importance of careful parameter selection. By objectively evaluating genre adherence, the study provides valuable insights into the strengths and limitations of MC software.

***Index Terms*—Computer-based music creation, deep learning, convolutional neural network, genre classification, music genre recognition, music quality assessment.**

I. Introduction

Applying computer-based technologies in music creation or aiding its creation is not a novel concept. Experiments to model music with neural networks (NN) started being developed in the 1980s and continued to the early 2000s [10] [17]. Eventually, with the growing development of technology and Deep Learning (DL), several studies attempted to model music using NNs. Certain NN architectures that perform well in fields, such as Natural Language Processing (NLP) and Computer Vision (CV), were proven to also be successful in music composition [11].

A. Problem Statement

The subjective nature of music quality and genre classification makes it challenging to objectively evaluate the performance of Music Composition (MC) models. This study seeks to develop an objective evaluation methodology using a Music Genre Recognition (MGR) model to determine the genre alignment of the generated samples. By incorporating this approach, the study aims to establish a reliable framework for assessing the quality of computer-generated music compositions.

B. Research Questions

This study aims to answer the following research questions:

1) Is it possible to objectively analyse the performance of MC models by evaluating the genre alignment of the generated samples?
2) How do specific parameters, such as artist selection and genre combinations, influence the quality and genre alignment of the composed samples?

C. Aims and Objectives

The main goal of this study is to evaluate the performance of MC models by analysing the quality of the generated samples. The author aims to establish an objective and reliable method for assessing the quality of samples produced by MC software. As this concept is relatively new, the study aims to determine the usefulness of the generated results. Furthermore, during the analysis of the MC model's outcomes, the study will investigate whether specific parameters substantially impact the quality of the generated songs.

II. Background

A. Music Genre

Music Information Retrieval (MIR), also called Multimedia Information Retrieval, encompasses the process of extracting valuable insights from music data. MIR finds application in various fields, including classification, recognising genres, separating music sources, and identifying instruments [7]. Songs are characterised by various distinguishing features, which can be utilised to categorise them into specific genres. These genres, assigned by the music industry, are typically derived from attributes in the songs or the time period they became popular. However, given the inherent subjectivity in this classification process and the lack of universally accepted standards, most academic studies [14] [15] agree that these definitions are somewhat ambiguous [6]. This ambiguity gives rise to alternative and non-traditional classification systems, such as the automatic classification of music into genres using Deep Learning (DL).

B. Audio Music Representation

Music analysis involves extracting features from songs to gain insights into various aspects of the music. The choice of representation, whether audio or symbolic, is crucial as it affects the quality and reliability of the analysis. This literature primarily focuses on audio representation and highlights two important derivatives: Mel spectrograms and Mel-frequency cepstral coefficients (MFCC). Mel spectrograms and MFCCs are widely used in machine learning due to their ability to capture information about changes in the power spectrum of sound over time. MFCCs are particularly useful for reducing data dimensionality and decorrelating features, making them beneficial for many machine learning tasks. On the other hand, Mel spectrograms maintain a more detailed representation of the power spectrum, which can be advantageous for tasks that require precise frequency information. Overall, both Mel spectrograms and MFCCs play significant roles in analysing and understanding music, providing valuable insights into the spectral content and temporal characteristics of audio signals.

C. Music Composition Based On Deep Learning

DL-based MC makes use of the results that are produced by MIR methods [12]. The most used NN architectures that are providing positive results in the approach to MC are; Recurrent Neural Networks (RNNs), Generative Models, such as Variational Auto-Encoders (VAEs) or Generative Adversarial Networks (GANs), and NLP-based models such as Long Short-Term Memory (LSTM) or Transformers [11].

III. Literature Review

A. Music Composition (MC)

The authors Civit, Civit-Masot, Cuadrado, and Escalona (2022) [8] conducted an analysis on Music Composition (MC) models to determine the prevalent AI-based techniques in the field. As of 2021, they observed a steady increase in the use of Transformer and GAN-based architectures [8].

Objective grading of music samples is challenging due to subjective opinions. During the development of Jukebox, manual evaluation was performed based on coherence, musicality, diversity, and novelty [9]. The authors Civit, Civit-Masot, Cuadrado, and Escalona (2022) [8] evaluated MCs by comparing the consistency of musical structure and motivic development among top-performing models, including Magenta Transformer, MuseNet, and PIA. They determined that although Transformer-based architectures initially showed promising performance, it was not consistent [8]. However, the launch of Google's MusicLN in 2023 may improve the consistency of Transformer-based architectures [3]. Jukebox, a unique model capable of working with audio music and composing songs with music, rhythm, and lyrics, was also evaluated. While it was praised for its ability to compose songs that included lyrics, music and rhythm, criticisms included the presence of recognizable artefacts in the output .wav files, limiting their professional usability [8]. GAN or LSTM-based systems were noted to exhibit good long-term structure but may lack complex or surprising musical content compared to Transformer-based models [8].

In another study by Hernandez-Olivan and Beltran (2021) [11], it was found that no specific neural network architecture can be considered superior to others in all situations. Different architectures have strengths and weaknesses, and some excel in generating specific types of output. Transformers and generative models were identified as having an advantage over other architectures, with notable examples being MuseGAN and Jukebox [11].

Overall, the analysis highlights the evolving trends in MC models, the challenges of objective grading, and the strengths and limitations of different neural network architectures in music composition.

B. Music Genre Recognition (MGR)

1) Data Preperation: Buttigieg Vella's paper [6] presents a variety of MGR projects and their results [5] [18] [4]. The majority of authors made use of 30 seconds of song length. However, Bishard & Laskar [4] split the 30-second samples found in the GTZAN dataset into ten seconds in an effort to inflate the dataset size. All authors used a file type of MP3, with the FMA Medium dataset being the most popular source of song samples. Bishard and Laskar [4], and Boxler [5] both performed a 70%, 15% and 15% split for training, validation and testing, respectively. Buttigieg Vella [6] performed a split of 60%, 15% and 25% for his data.

2) MGR Machine Learning Models: The Convolutional Neural Network (CNN) architecture is widely used in Music Genre Recognition (MGR) applications. Various studies [6] [5] [18] [4] have applied CNNs, incorporating techniques like batch normalization and the Exponential Learning Unit function, to achieve good results. Boxler [5] achieved a loss of 1.42 using CNNs, while Murauer and Specht [18] obtained a loss of 1.65, attributing it to hardware limitations.

Buttigieg Vella [6] compared different neural network models (ANN, CNN, RNN, GBM) for MGR and found that the CNN model performed the best, achieving a loss of 0.942 and an accuracy and F1 score of 0.69. Other papers have also demonstrated the effectiveness of CNNs in related tasks such as music tagging and instrument recognition [16] [13].

C. Conclusion

The evaluation of Music Composition (MC) models is challenging due to the subjective nature of assessing song quality. To address this, a novel approach was adopted, focusing on genre classification as a means to evaluate the quality of composed samples. Jukebox, with its ability to specify the genre of the desired composition, was chosen for this evaluation.

To create a Music Genre Recognition (MGR) model, a Convolutional Neural Network (CNN) was selected based on its effectiveness in previous studies. Two model versions were developed, one using Mel Frequency Cepstral Coefficients (MFCCs) and the other using Mel spectrograms. This choice was made to explore any potential influence of the more detailed representation provided by the Mel spectrogram on the model's accuracy.

IV. Methodology

This project consists of three distinct sections. First, a model is created that is capable of identifying the genre of an audio file. Following that, music is composed using OpenAI's Jukebox, with a request of the genre. Finally, the model detects if the piece of music belongs to the requested genre.

A. Music Genre Recognition

1) Dataset: For the purpose of Music Information Retrieval (MIR), various datasets were evaluated to determine the most suitable option for this project. The selected dataset needed to meet certain criteria, including having high audio quality, containing commonly used music genres, and maintaining a balanced distribution. After careful consideration, a modified version of the FMA (Free Music Archive) dataset was chosen. as it offers a high quantity and quality of audio, making it an appealing choice for this study. The modified version of the FMA dataset chosen contains six genres: classical, folk, hip-hop, jazz, pop, and rock. Each genre in the dataset is represented by 619 samples, except for Jazz, which has 384 samples. To further increase the dataset size, all samples were split in 10-second snippets. Overall, the final dataset consists of 10,437 10-second song samples across the six selected genres.

2) Feature Extraction: Pre-processing was executed on the given dataset by extracting 40 Mel Frequency Cepstral Coefficients (MFCCs) every 2048 samples, while also down-sampling the snippets to 22.05Khz. The chosen audio features for this analysis were MFCCs and Mel-spectrograms, as they are widely regarded as the most effective acoustic features. Two similar models were developed, each utilising one of these Music Information Retrieval (MIR) techniques, and their respective performances were evaluated.

3) Machine Learning Model: For this project, a Convolutional Neural Network (CNN) was selected as the machine learning model. CNNs have demonstrated promising results in various applications, including music genre classification. To ensure a systematic evaluation, the dataset was divided into three sets: training, validation, and testing, with proportions of 70%, 15%, and 15%, respectively. The training set was shuffled and used to train the model over a specified number of epochs. After each epoch, the model's performance was evaluated using the validation set. This iterative process allowed for monitoring the model's progress and making necessary adjustments. Once the model was adequately trained, its architecture and parameters were saved for future use. The final evaluation was conducted using the testing set, which consisted of previously unseen samples. The objective was to classify the song tracks into the predefined genres that the model was trained on. The model's detections were compared with the actual genre labels, and the results underwent thorough analysis to assess the model's performance and accuracy.

B. Jukebox

Upon installation of Jukebox, the author gained access to three distinct models: the 1B Lyrics, 5B, and 5B Lyrics. These models primarily differ in the resources required for their operation and the volume of data utilised during their training. The 1B Lyrics model underwent training using a dataset comprised of one billion words from various sources. Conversely, the 5B models were trained with five billion words. The 5B and 5B Lyrics models differ in their function; the latter generates songs with lyrics, while the former solely produces the accompanying music [9]. The larger 5B models were chosen as the primary models to evaluate.

C. Genre Detection

This section synthesises the work conducted in the previous two chapters. The highest-performing CNN model, responsible for identifying the genre of a music snippet, is loaded alongside the Jukebox samples. Subsequently, the model detects the genre of the song. Users can then assess the accuracy of the genre detections through the calculation of an accuracy percentage.

V. Implementation

The artefact consists of five modules. The first module focuses on data preparation, involving cleaning and categorising the dataset. The second module is responsible for extracting the MFCCs and Mel Spectrograms from the dataset. The third module encompasses the development, optimisation, testing, and evaluation of the Convolutional Neural Network (CNN) model.

The fourth module operates independently and utilises OpenAI's Jukebox to generate original musical compositions. It produces three 30-second WAV files in each iteration. Finally, the fifth module integrates the CNN model with the music samples generated by Jukebox to determine the corresponding genre for each sample.

A. Environment Settings

To facilitate the project, connecting to Vast.ai was essential in order to rent hardware resources [19]. Vast.ai offers various Docker options that simplify environment setup. For music composition using Jukebox, the author utilised a PyTorch Docker, and only a Conda installation of Mpi4py was necessary to complete the environment.

On the other hand, for the Music Genre Recognition (MGR) model, the TensorFlow Docker was required. This environment included several installed libraries such as Keras, Matplotlib, and Librosa, which were necessary for optimisation, plotting, and generating the required Music Information Retrieval (MIR) features.

Locally, a Conda environment with Python version 3.9.16 was set up on the author's machine. The Streamlit library was installed to run the final application (module 5), along with Pandas, Pydub, TensorFlow, Librosa, and Matplotlib.

B. Module 1 - Datset

The medium subset of the FMA dataset and its associated metadata folder were downloaded from a GitHub repository and stored locally. The data was then organized into genre-specific folders using a Python script and the 'tracks.csv' file provided in the metadata. Songs that did not meet the required length (as determined from a wiki page) were removed. After examining the organized dataset, a decision was made regarding which genres to include based on the number of songs and their popularity. The resulting dataset contains all the samples from which Music Information Retrieval (MIR) features will be extracted.

C. Module 2 - Feature Extraction

A Vast.ai instance was set up using the TensorFlow Docker. The dataset was uploaded to this instance, and a Python program was developed. The program first split each sample into three snippets. Then, using the Librosa and Pandas libraries, it generated the MFCCs for each snippet and saved them in a .csv file. This process was repeated to generate and save the Mel Spectrograms for each snippet as well.

D. Module 3 - Model Creation

An ipynb program was developed using Jupyter Notebook to perform tasks on two CSV files containing MFCCs and Mel Spectrograms of the dataset. The program follows the same process for both files, first using the MFCC file and then the Mel Spectrogram file.

The chosen CSV file is split into training, validation, and testing data using a stratified split of 70/15/15. The model is then trained iteratively 1000 times using KerasTuner, which optimises the model's hyperparameters and architecture, including layers and learning rate. The CNN model is built using the Keras library within TensorFlow. Since the model is run on Vast.ai, the built-in TensorFlow Docker is utilised, ensuring the correct configuration of the environment.

After each iteration, the program plots a graph of training and validation accuracy, as well as loss, using TensorBoard. This allows for the determination of the best-performing neural networks based on hyperparameters. Additional manual tweaking was performed, and the testing accuracy of the best-performing model is ultimately determined.

E. Module 4 - Music Composition (Jukebox)

This module operates independently from the previous ones and requires significant computational resources. To meet this requirement, a new instance is connected to Vast.ai using the PyTorch docker. The core functionality of this module is based on OpenAI's Jukebox, which is installed through their git repository [2]. A pre-downloaded Jupyter Notebook [1] serves as the necessary code interface for Jukebox.

Each iteration of the program composes three 30-second .WAV samples and the model, genre, artist, and lyrics were changed based on the test case. Jukebox initially provides a level 2 sample, which is grainy and lacks clarity. Substantial processing power is required to render the sample into its final form, level 0. This process is repeated multiple times, resulting in songs inspired by different artists, and genres. After each iteration, the Jukebox samples are saved locally for use in the next module.

F. Module 5 - Music Genre Classifier

This module represents the program's final solution, leveraging the solutions derived from modules 3 and 4 to determine the genre of music samples generated via Jukebox. Streamlit is used to create a user-friendly web experience. The user is prompted to input a 30-second WAV song sample, which the application segments into three distinct 10-second snippets. For each snippet, the model predicts its genre and a visual representation of its confidence is provided via a bar chart. The program also provides the MFCCs of the snippets and allows the user to listen to the snippet

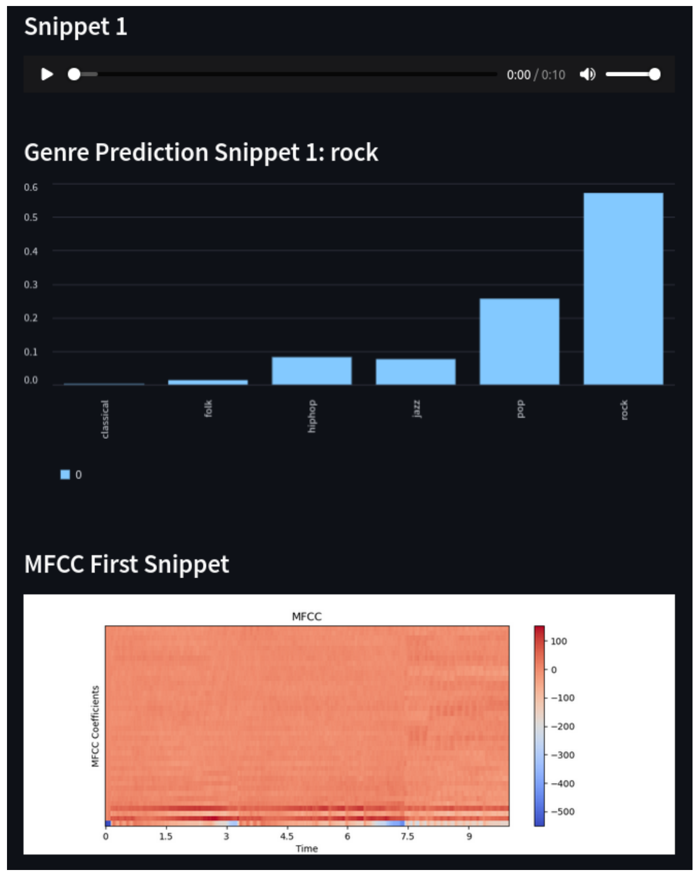

Fig. 1. Jukebox Model Genre Correctness

VI. Results and Discussion

A. Genre Classification Model

The final model for this project was determined through an extensive search using KerasTuner. After conducting 1000 trial runs, the best-performing model was selected based on its performance metrics. The specifications of the final model are as follows:

- Input Layer: number of neurons is dependent on input size

- Convolutional layers: The model has four convolutional layers, each with 64 filters of size (3,3) and ReLU (Rectified Linear Unit) activation.
- MaxPooling Layers: After each convolutional layer, there is a max pooling layer with pool size (3,3) and stride (2,2). The padding is set to 'same'.
- BatchNormalization Layers: The model has three batch normalization layers.
- Dropout Layers: The model has two dropout layers each with a dropout rate of 0.3.
- Dense Layers: The model has two dense layers. The first has 64 neurons and uses the ReLU activation function. The second is the output layer and has as many neurons as there are unique target classes. It uses softmax activation.

Upon evaluation, the MFCC model and the Mel Spectrogram model had an insignifant difference in accuracy. The MFCC model was chosen for further analysis due to its slightly higher accuracy and smaller file sizes.

The final version of the model had a validation accuracy of 71%, validation loss of 0.84, training accuracy, precision and F1 score of 0.702.

B. Jukebox Samples

During the song composition process using Jukebox, the author specified various parameters such as the model, genre, artist, and lyrics. To evaluate the genre alignment of the composed songs, the Music Genre Recognition (MGR) model was employed. Since the MGR model is designed to detect the genre of 10-second songs, each 30-second song from Jukebox was divided into three snippets, and the genre was determined for each snippet. A "genre match" was recorded if at least two out of the three snippets matched the requested genre. On the other hand, if the MGR model detected the genre of the artist instead, it was considered an "artist match". Additionally, the author also personally rated the songs based on quality from 0 to 7. A bar chart displaying the full results can be viewed in figure 2.

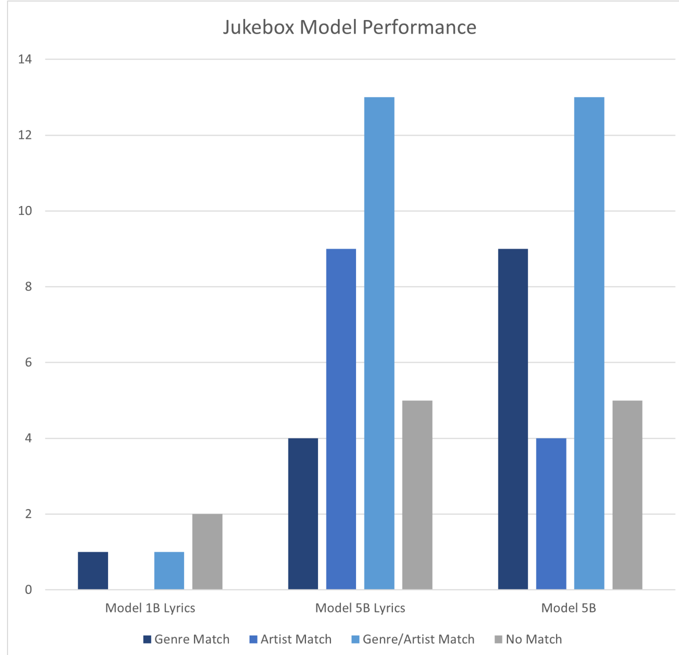

Fig. 2. Jukebox Model Genre Correctness

The 5B and 5B lyrics models each produced 18 samples, while only 3 were generated by the 1B lyrics model, making a total of 39 samples generated. This is due to the 1B lyrics model being established as less effective than the others, hence, it is more advantageous to evaluate the 5B models.

Numerous samples within the dataset exhibit different detected genres for their snippets. While part of this inconsistency may be attributed to the MGR model, some can be attributed to the quality of the song snippets themselves. For instance, consider test case 14, where the sample begins with silence and then transitions into a brief melody. Although the genre classification model meets certain standards, it is not 100% accurate, and running these test cases on a different model could yield vastly different outcomes.

1) Genre Correctness: Based on the results, in the simplest test cases where the genre aligns with the artist (ex: genre: Classical and artist: Mozart) and no lyrics are specified, Jukebox achieved a 67% genre match rate. Since there is no comparable research conducted on other music composition models, the author considers a two-thirds success rate in test cases to be acceptable.

However, when the requested genre contradicts the artist (ex: genre: Classical and artist: Lady Gaga), the detected genre tends to favour the artist rather than adhering to the desired genre. This discrepancy may arise from Jukebox's tendency to compose songs that closely resemble the artist's style rather than aligning with the specified genre. While it is commendable that all samples generated by the superior 5B models exhibit alignment with an artist or genre, it is crucial to acknowledge that the genre accuracy has decreased to 42%.

Furthermore, the inclusion of lyrics as a parameter has shown a significant influence on the sample's genre classification. Songs with matching genres and lyrics achieve a 100% genre match rate, while songs with opposing genres and lyrics have no matches. This observation may once again stem from the fact that songs are composed with substantial influence from the original song containing those specific lyrics.

2) Sample Quality: In evaluating the quality, the primary emphasis will be on the two polar ends of the song ratings: the 0-1 and 6-7 ranges. This is based on the assumption that most listeners would likely agree upon the reasons behind these categorisations.

The project focused primarily on composing classical songs due to the resource-intensive nature of song composition. Out of the 14 low-rated songs, half of them were set to the classical genre by the author, but only two were correctly identified as classical. Conversely, three out of the top five highest-rated samples were detected as being classical songs, even when a different genre was requested. These findings suggest a tendency towards generating high-quality classical samples within the Jukebox system.

A notable observation was the frequent occurrence of low-rated samples that still managed to match the predicted genre. One such example is Test Case 12, which consists of applause and unintelligible noises. It could be that Jukebox falsely believed it had produced a correct outcome. Alternatively, it could be attributed to certain genres and artists not harmonising well together. For instance, all test cases involving classical music from Kanye West yielded consistently poor results.

C. Conclusion

Jukebox's performance is significantly impacted by a multitude of parameters, and different hyper-optimisations can alter these results. Additionally, the quality of the samples produced by Jukebox varies considerably even in the same batch of song composition.

VII. Conclusion

In conclusion, this study aimed to evaluate the performance of Music Composition (MC) software by analysing the quality of the composed samples. The main objective was to establish an objective and reliable method for assessing the quality of the samples produced by MC software. To achieve this, a Music Genre Recognition (MGR) model was developed, enabling the evaluation of genre alignment.

Throughout the study, it became evident that assessing the quality of a song remains subjective, to the point where even the genre of certain songs is subjective. However, the MGR model provided a valuable tool for objectively determining whether the composed samples aligned with the requested genre. This approach allowed for a more comprehensive and unbiased assessment of the MC software's performance.

The findings of this study demonstrate the effectiveness and importance of incorporating the MGR model into the evaluation process. By comparing the predicted genre with the requested genre, it was possible to assess the MC software's ability to compose samples that align with the desired genre. This innovative approach addresses the need for a more objective evaluation methodology in the field of MC software, providing a viable approach.

Furthermore, the study focused on analysing the impact of specific parameters on the genre of the composed songs. When evaluating Jukebox, it was observed that contradictory genre and artist combinations often resulted in the MC models favouring the genre of the artist. Additionally, when the lyrics of an existing song were included in the parameters, the composed music always aligned with the genre of the original piece.

These findings highlight the complexities involved in genre prediction and the influence of different parameters on the output of MC models. In addition, the study provides valuable insights into the strengths and limitations of these models in composing songs of the desired genre.

Overall, this research contributes to the understanding of MC model performance and offers insights for future improvements in composing high-quality music samples that align with specific genre requirements.

1) Limitations and Future Work: The project faced limitations in time and resources, impacting its progress. Efforts were focused on developing a reliable Music Genre Recognition (MGR) model to evaluate Jukebox. Attempting to detect genre in full 30-second songs required further optimization for comparable performance to the 10-second model. The project's research approach, grading music composition models through genre validation, is novel with potential for further exploration. Conducting additional test cases with Jukebox, exploring its ability to continue composition from a song sample, or conducting comparative tests using different models could provide valuable insights. Further studies in this field can expand the understanding of music composition models and their capabilities.

References

[1] Interacting with Jukebox - Colaboratory.

[2] openai/jukebox: Code for the paper ”Jukebox: A Generative Model for Music”.

[3] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank. MusicLM: Generating Music From Text.

[4] D. Bisharad and R. H. Laskar. Music genre recognition using convolutional recurrent neural network architecture. *Expert Systems*, 36(4):e12429, 8 2019.

[5] D. Boxler. MACHINE LEARNING TECHNIQUES APPLIED TO MUSICAL GENRE RECOGNITION by. 2020.

[6] J. Buttigieg, V. Supervisor, and J. Bonello. Analysing Diverse Algorithms performing Music Genre Recognition. 2022.

[7] K. Choi, G. Fazekas, K. Cho, and M. Sandler. A Tutorial on Deep Learning for Music Information Retrieval.

[8] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona. A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, 209, 12 2022.

[9] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A Generative Model for Music. 4 2020.

[10] D. Eck. A network of relaxation oscillators that finds downbeats in rhythms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2130:1239–1247, 2001.

[11] C. Hernandez-Olivan and J. R. Beltrán. Music Composition with Deep Learning: A Review. *Signals and Communication Technology*, pages 25–50, 8 2021.

[12] L. A. Iliadis, S. P. Sotiroudis, K. Kokkinidis, P. Sarigiannidis, S. Nikolaidis, and S. K. Goudos. Music Deep Learning: A Survey on Deep Learning Methods for Music Processing. *2022 11th International Conference on Modern Circuits and Systems Technologies, MOCAST 2022*, 2022.

[13] P. Li, J. Qian, and T. Wang. AUTOMATIC INSTRUMENT RECOGNITION IN POLYPHONIC MUSIC USING CONVOLUTIONAL NEURAL NETWORKS.

[14] T. L. Li and A. B. Chan. Genre classification and the invariance of MFCC features to key and tempo. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6523 LNCS(PART 1):317–327, 2011.

[15] T. Lidy, A. Rauber, A. Pertusa, and J. Manuel. IMPROVING GENRE CLASSIFICATION BY COMBINATION OF AUDIO AND SYMBOLIC DESCRIPTORS USING A TRANSCRIPTION SYSTEM. 2007.

[16] V. Lostanlen and C.-E. CelláCellá. DEEP CONVOLUTIONAL NETWORKS ON THE PITCH SPIRAL FOR MUSICAL INSTRUMENT RECOGNITION.

[17] M. C. Mozer. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *http://dx.doi.org/10.1080/09540099408915726*, 6(2-3):247–280, 2007.

[18] B. Murauer. Detecting Music Genre Using Extreme Gradient Boosting. 2018.

[19] Vast.ai. FAQ — Vast.ai, 2023.