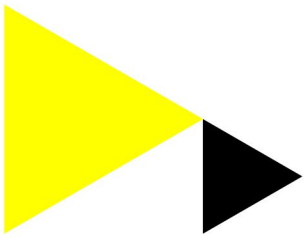
From Novice to Composer

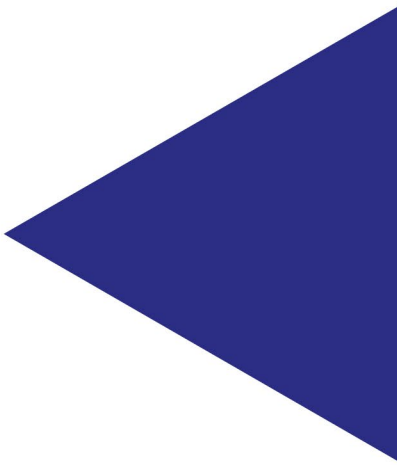
Using AI to Facilitate Music Creation with MIDI Generation and Sample Extraction

Author(s)
Slingerland, Erik; Fuckner, Mario

Publication date
2023

Document Version
Final published version





From Novice to Composer: Using AI to Facilitate Music Creation with MIDI Generation and Sample Extraction*

Erik Slingerland and Dr. Marcio Fuckner

Amsterdam University of Applied Sciences, 1091 GH Amsterdam, Netherlands
erik.slingerland01@gmail.com, m.fuckner@hva.nl

Abstract. Concerns have been raised over the increased prominence of generative AI in art. Some fear that generative models could replace the viability for humans to create art and oppose developers training generative models on media without the permission of the artist. Proponents of AI art point to the potential increase of accessibility. Is there an approach to address the concerns raised by artists, while still utilizing the potential these models bring?. Current models often aim for autonomous music generation. This however makes the model a black-box that users can't interact with. By utilizing an AI pipeline combining symbolic music generation and a proposed sample creation system, trained on Creative-Commons data, a musical looping application has been created to provide non-expert music users a way to start creating their own music. First results show that it assists users in creating musical loops and shows promise for future research into the field of human-AI interaction in art.

Keywords: Audio and Speech Processing · Responsible AI · Generative Art · Computational Creativity · Deep Learning.

1 Introduction

In the recent years, discussions have arisen about the use of AI in art. Due to the increased potential, some fear a trend where artists can be replaced by AI models [1]. There are also concerns over ownership of AI art and training models on data without the original creators' knowledge or approval [2] [3]. Proponents of using AI in art cite the idea that AI can improve accessibility towards art, encourage creativity [2] and could positively impact mental health [4]. Within this context, the project Hitloop was conceived to research possible applications that allow for cooperation between humans and AI in creating music while addressing two aspects of responsible AI: Transparency and Human Autonomy. The first version of Hitloop was an application to assist users in musical looping for Electronic Dance Music (EDM). In music looping, audio fragments of different recordings are combined and rearranged to create music. These audio fragments are called samples and are selected by the musicians [5]. Most musicians arrange

* Supported by Responsible AI lab, Amsterdam University of Applied Sciences

these loops in the Musical Instrument Digital Interface (MIDI) protocol. Within EDM, music looping is a standard approach for creating songs [6]. To achieve the goals of Hitloop, an AI pipeline comprised of an AI model for generating MIDI patterns and a model for extracting samples out of audio recordings was created. To address transparency, the AI models were trained fully on open source and non-copyrighted data. To address human autonomy, these systems were implemented into an application where users could create their own music and change every aspect of the composition.

2 AI models

2.1 MIDI generation

Generating MIDI patterns is a form of symbolic music generation [7]. These models are often trained on either datasets of classical music or non-public datasets. Previous researchers have successfully used a combination of models to create a pipeline for transforming audio into a symbolic representation with the purpose of creating their own dataset [8]. To increase transparency of the MIDI generation model for Hitloop, a dataset was created by emulating this process with non-copyrighted EDM. Roughly 90 songs with a CC-BY, CC-BYSA, CC-BYNC, CC-BYNCSA or CC Zero licence [9] were included. A list of the songs used has been added to the Hitloop application. The process of transforming music into a MIDI dataset, involves three steps. Firstly, the audio track is split into separate instrument channels using Meta's Demucs model [10]. Afterwards, the channels are transcribed into MIDI.Drums were transcribed with Omnizart [11] and the pitch-based instruments with Basic Pitch by Spotify [12]. As a final step the resulting MIDI files are combined into one big array, this will allow the MIDI generator to learn specific patterns for the instrument groups. Because EDM is characterised by small 1-4 bar repeating patterns [6], the decision was made to split the array into 1-bar sections for decreased recourse demand during training. To generate the MIDI patterns, two architectures were tested. The initial testing was with a GAN based on MuseGAN [13]. Secondly, a VAE based on MusicVAE [14]. These were chosen due to their publically available information on training parameters. Both models were trained for 1000 epochs and on the same training data.

2.2 Sample extraction

During the research, no significant publications exploring the task of sample creation were found. This paper will therefore propose it's own method. The proposed sample creation pipeline works by finding audio fragments within a recording that are similar to instrument sounds. This is done by first extracting auditory features from the instrument samples. The audio recording is then be

examined for potential samples using a sliding window. For every window, the features get extracted are compared against the feature patterns of all the instrument samples. When features have a certain similarity, the audio fragment is extracted as a sample. Different researchers have shown that spectral representations of audio fragments are sufficient features to categorize instruments [15] [16]. Therefore mel-spectrograms were used as the audio features. The cosine distance was used to establish the similarity between the window and instrument sample. In different machine learning applications, cosine similarity is the industry standard [17]. To extract a sample, the cosine similarity between a window and sample must be higher than a pre-defined threshold. By using this threshold, it is possible to influence the amount of- and similarity of the extracted samples. The lower the threshold, the more dis-similar extracted samples will be. This sample extraction pipeline requires instrument samples and audio recordings to be extracted. The instrument samples were collected from the Wikimedia open-CC audio library. This included snares, hi-hat's, cymbals and basses. The used recordings were from the Re:vive Amsterdam dataset [18].

3 Evaluation

The two AI models utilised different evaluation methods. The MIDI generation pipeline was evaluated with a mix of quantitative and qualitative measures, while the sample creation pipeline was only tested with qualitative measures. Because of the lack of established quantitative measures to determine a 'good sample'.

In the first phase, quantitative measures were used to establish what MIDI generation model was most similar to the dataset. This model was used in phase two, where user testing is conducted with non-expert users to establish whether the AI tool assisted in creating musical loops. The measures used were the average of unqualified notes [13] (noise) and the average silent ratio for 1000 randomly generated MIDI patterns. To indicate possible mode collapse, the 1000 generated patterns were also compared for cosine similarity. These scores where then compared to the same measures on the dataset. By doing this, it could be established whether the generated music had characteristics behavior to EDM. When evaluating the results of the test, both the VAE and GAN performed closest to the dataset in a specific test. In the end the decision was made to use the GAN because it had more variation in the generated segments.

For the user testing, a prototype musical looping application was built using JavaScript and the tone.js [19] module for audio playback. In this prototype the user could create and adjust both the MIDI pattern and samples for a 1-bar loop. The participants received two versions of Hitloop. Firstly a version utilizes the proposed AI systems and secondly a version where both systems operate using purely randomly extracted samples and MIDI patterns generated with a random number generator. For both versions, the users received two tasks: 'Create a sequence that you enjoy when you play it on repeat for 2-3 times'

and 'Adjust the sequence so it feels either calmer or more energetic'. By using these two questions, the users would emulate tasks that a DJ would do when creating EDM, without explicitly telling what the goal of our research is. For both versions the users had to answer four questions on a Likert scale. The responses were compared between the two versions to see if the AI systems contributed to the music creation process. 18 users were used during testing and the average scores are listed in figure 1.

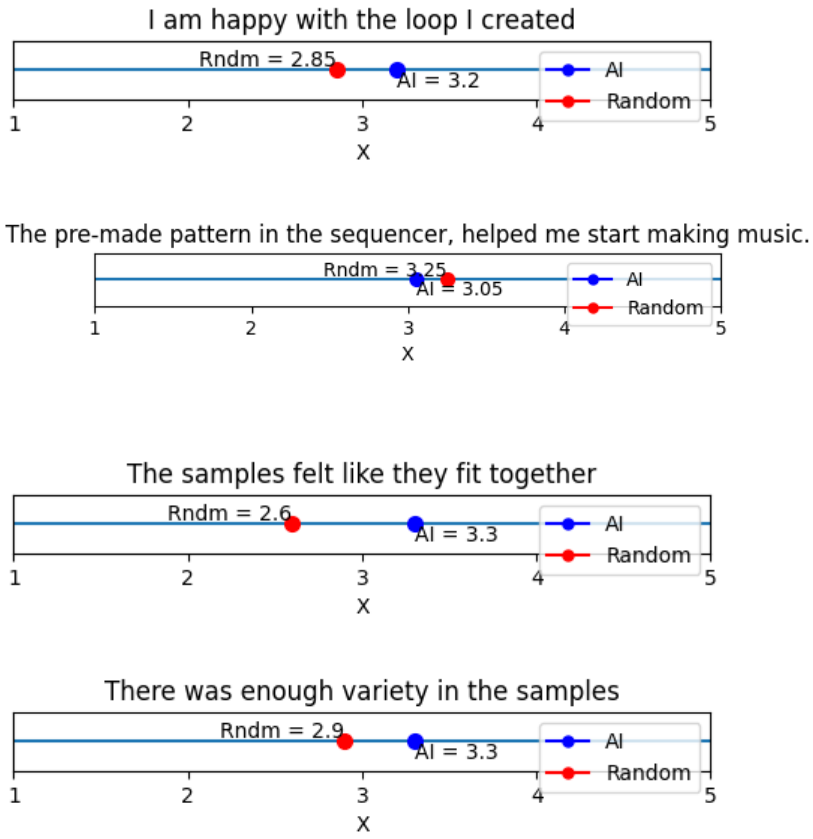


Fig. 1: Average scores of the random and AI pipeline of Hitloop. 1= strongly disagree, 5 = strongly agree

The User testing revealed that the AI-generated MIDI patterns did not outperform the random baseline. Therefore it can be questioned whether the MIDI generation component is suitable for the goal. The goal of Hitloop is to prompt a non-expert user to create musical loops. If this could be achieved with more low-tech solutions like a random system or going back to pre-deep learning rule-based systems for music generation [20], this could be an alternative. Because these models are simpler and often require less training data, this could be a way to address the concerns of transparency. The sample creation pipeline was evaluated more favourably. The AI pipeline outperformed the random baseline in perceived cohesion and variety. Because the AI pipeline scored higher while receiving worse pre-made patterns. It is also possible this could indicate that a perceived higher quality of samples results in the users creating a better musical loop. To conclusively determine this, more research is required.

4 Conclusion

The proposed pipeline for Hitloop has achieved it's goal of exploring Transparency and Human Autonomy questions whilst assisting users in creating musical loops. While the individual models used are not yet sufficient for deployment, it can be considered a first step to spark the discussion on the development of transparent and human in the loop AI music applications.

References

1. J.-W. Hong and N. M. Curran, "Artificial intelligence, artists, and art: Attitudes toward artwork produced by humans vs. artificial intelligence," vol. 15, no. 2, pp. 58:1–58:16. [Online]. Available: https://doi.org/10.1145/3326337
2. E. Cetinic and J. She, "Understanding and creating art with AI: Review and outlook," issue: arXiv:2102.09109. [Online]. Available: http://arxiv.org/abs/2102.09109
3. M. Senftleben, "A tax on machines for the purpose of giving a bounty to the dethroned human author – towards an AI levy for the substitution of human literary and artistic works," issue: 4123309 Place: Rochester, NY. [Online]. Available: https://papers.ssrn.com/abstract=4123309
4. D. Williams, V. J. Hodge, and C.-Y. Wu, "On the use of AI for generation of functional music to improve mental health," vol. 3. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2020.497864
5. T. Rodgers, "On the process and aesthetics of sampling in electronic music production," vol. 8, no. 3, pp. 313–320. [Online]. Available: https://www.cambridge.org/core/journals/organised-sound/article/abs/on-the-process-and-aesthetics-of-sampling-in-electronic-music-production/236F868DD4AF8926152E6182C005E78E
6. A. Behr, K. Negus, and J. Street, "The sampling continuum: musical aesthetics and ethics in the age of digital production," vol. 21, no. 3, pp. 223–240. [Online]. Available: https://doi.org/10.1080/14797585.2017.1338277
7. J.-P. Briot, "From artificial neural networks to deep learning for music generation – history, concepts and trends," issue: arXiv:2004.03586. [Online]. Available: http://arxiv.org/abs/2004.03586
8. G. A. C. d. Santos, A. Baffa, J.-P. Briot, B. Feijó, and A. L. Furtado, "An adaptive music generation architecture for games based on the deep learning transformer mode," issue: arXiv:2207.01698. [Online]. Available: http://arxiv.org/abs/2207.01698
9. About CC licenses. [Online]. Available: https://creativecommons.org/about/cclicenses/
10. S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," issue: arXiv:2211.08553. [Online]. Available: http://arxiv.org/abs/2211.08553
11. Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, "Omnizart: A general toolbox for automatic music transcription," vol. 6, no. 68, p. 3391, publisher: The Open Journal. [Online]. Available: https://doi.org/10.21105/joss.03391
12. R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
13. H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," issue: arXiv:1709.06298. [Online]. Available: http://arxiv.org/abs/1709.06298
14. A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," issue: arXiv:1803.05428. [Online]. Available: http://arxiv.org/abs/1803.05428

15. L. Haidar-Ahmad, "Music and instrument classification using deep learning technics."
16. K. Racharla, V. Kumar, C. B. Jayant, A. Khairkar, and P. Harish, "Predominant musical instrument classification based on spectral features," in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 617–622. [Online]. Available: http://arxiv.org/abs/1912.02606
17. P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A comparison of semantic similarity methods for maximum human interpretability," issue: arXiv:1910.09129. [Online]. Available: http://arxiv.org/abs/1910.09129
18. Re:vive. [Online]. Available: https://revivethis.org/
19. Tonejs/tone.js: A web audio framework for making interactive music in the browser. [Online]. Available: https://github.com/Tonejs/Tone.js
20. D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," vol. 50, no. 5, pp. 1–30. [Online]. Available: http://arxiv.org/abs/1812.04186