Music Generation Through Live Situation Recognition

Ilse Mariana Córdova Sánchez, José Luis Gutiérrez Espinosa, Andrés Gómez de Silva Garza
Computer Engineering Department
Instituto Tecnológico Autónomo de México (ITAM)
Río Hondo #1, CDMX, México
{icordov1, jguti109, agomez}@itam.mx

Abstract

Music can be treated as an autonomous language that oftentimes aids humans with experiencing and conveying different emotions with even more precision than words. At the same time, creating music can be considered one of the most creative activities humans carry out since it not only requires a precise technique and particular execution but also an appropriate interpretation that encompasses a vast range of dynamics to be able to communicate that which is left unspoken. This paper describes, proposes and implements a system called NIMA (Neural Image-to-Music Algorithm). This system was designed to compose and play background music in order to provide atmosphere in different situations. The generation of the aforementioned musical pieces is based on video feedback given to the system from a camera. The NIMA system drew inspiration from the fact that a considerable number of pieces that are digitally generated usually carry a monotone sound throughout their duration, and we wanted to avoid this.

Introduction

Creating music requires a precise technique, a particular execution and an appropriate interpretation that encompasses a vast range of dynamics to be able to communicate different messages. We can call this musical interpretation: the artistic communication that a performer is able to convey. If a musical piece were to lack adequate interpretation, it would sound rather monotonous and flat. The system we describe in this paper, NIMA (Neural Image-to-Music Algorithm), falls within the musical category.

In NIMA we decided to focus on generating background music. The aim of NIMA is to imitate the work of a musician creating music to be played in the background of a given situation, for example while having a meal or while studying. Said task is not simple to accomplish, as music creation requires several complex cognitive abilities in the human brain and is even difficult for humans to do.

One of the main reasons why we decided to take on this project is because we wanted to create a tool that could reach several users, be it to listen to the background music generated by the system for simple enjoyment, or to use it as an aid when it comes to drawing inspiration from the musical pieces that it creates. Furthermore, we wanted to achieve the creation of dynamic and emotive music that could adapt to the environment that is captured live via a camera.

This paper is organized as follows. We start by analyzing two past projects that are related to ours and have looked further into concepts related to the dynamics of musical pieces. Then we provide a general description of our system. Afterwards, we will proceed to detail the design and implementation of NIMA. Finally, we present the conclusions we have reached and discuss the improvements we plan to implement in the near future.

Previous Work

Numerous systems that deal with computerized musical composition have been built previously across the globe in different ways. In this paper, we present and detail two previous solutions that we have analyzed and drawn inspiration from. These previous projects are mainly related to the generation of a musical style or interpretation. We could not find an existing system that processes live video and uses it to compose music.

Neural Translation of Musical Style

Iman Malik identifies two important elements when it comes to music generation. The first one is the composition, which focuses on the pitches and harmonies that define a song, and the second one is the performance of the piece or musical style. Inspired by the rather monotone results that previous music generation projects yielded, Malik decided to create StyleNet. StyleNet is a system that "should be able to synthesize and inject dynamics into monotonous MIDI files" (Malik 2017); this system mainly focuses on adding musical style to compositions in order to imitate stylized human performances.

The architecture of her bot is implemented by using two main layers.

1) The Bi-Directional LSTM layers "…provide memory for learning dependencies, and the bi-directional

architecture allows the model to take the future into consideration" (Malik 2017).
2) The linear layer that performs a linear transformation on the given input, which is the output of the Bi-Directional LSTM layer that needs to be scaled in order to represent a larger range.

The results yielded by Malik's project are limited to learning the dynamics the system is trained on. The initial data set that was used by the model is the Yamaha e-Competition files that contain human-recorded dynamics.

Once her system was completed, Malik designed an experiment inspired by the Turing test in order to evaluate StyleNet. Said experiment consisted of a survey in which people were asked to correctly identify the human performance. The results that were obtained show that only 46% of the participants could correctly identify the human performance; therefore, her system successfully passed the Turing-style test it was subjected to.

Music Transcription Modelling and Composition Using Deep Learning

This Cornell University project is based on the use of long short-term memory (LSTM) networks, "which increases the number of parameters to be estimated in training, but controls the flow of information in and out of each cell to greatly help with convergence" (Sturm et al. 2016). Musical applications can benefit heavily from an LSTM since the perception of time in this type of network is often regarded as much better than other types of networks. It is likely we will implement this type of neural network in NIDA in the future.

This system's purpose is to create music transcription models to facilitate music composition and it does so by using the LSTM networks which are likely to generate rhythmically good sound clips since time is the main focus of this network. Once this is achieved, a music composer can build a music piece based on said clips.

In order to achieve the music composition goals that the team that developed it set, around 23,000 music sheets were used in order to train the model and obtain music transcriptions and music sheets that could be actually used by real life musicians and also featured interesting musical dynamics.

NIMA System Description

After evaluating some of the musical composition models that have been created, and having drawn some inspiration from them, we created the NIMA system. This system uses real time video input and processes it in order to analyze the type of environment the computer, presumably being transported by its user, finds itself in. Once this recognition has been completed, the system proceeds to evaluate the gathered data in order to find its relationship with different musical aspects related to dynamics that will allow it to compose and generate a musical piece featuring expressive timing and dynamics that are inspired by the type of environment.

System Design and Implementation

In order to fulfill our goals, we decided to use Python, since many tools are available in this programming language, and it is quite versatile when it comes to AI. We also used Google's Colab Notebooks, which helped greatly since it is not necessary to set up environments and, because of its modularity, makes the process of working with neural networks much easier.

General Operation

The operation of the current version of NIMA can be summarized by the following flow of steps:
1) TensorFlow, Magenta and other dependencies are installed. The necessary libraries, SoundFont and Magenta models are downloaded.
2) The length of the musical piece is defined by the user and a picture is taken.
3) The picture is processed to identify the overall brightness level, and the HSV (Hue, Saturation, and luminosity Value) of the six most prominent colors are defined.
4) The level of brightness of the image is used to decide how many notes per second will be played in the piece.
5) The HSV values are used to decide which pitches from the chromatic and diatonic chords (Inspired Acoustics 2020), that can be found in the music pieces stored in the datasets, will have more presence within the musical piece.
6) The context that the neural network associates with the picture defines the type of musical piece the system will generate. NIMA is capable of identifying 7 different types of environments: bedrooms, exteriors, vegetation, highways, living rooms, skylines and water bodies.
7) A musical piece is created that takes into account the decisions that were made in Steps 4-6.

Computer Vision

The first big section of the project is related to computer vision, as it is through this that the type of environment in which the user is immersed is detected. The computer vision part of NIMA can be divided into three phases: obtaining the image's brightness, finding the six colors most dominant in the image, and identifying the type of environment (situation).

For the first two, the software uses a frame taken from the video feed every 30 seconds (this parameter value can be changed) and analyses:

Image brightness is obtained by converting the frame to HSV format and extracting the "Value" value of each pixel and then calculating their average. The image brightness is used to determine the music's speed (beats per minute).

Image dominant colors are obtained through k-clustering the pixels' colors, finding the six cluster centroids with the most pixels near to them, and identifying their color (OpenCV 2020).

The third phase, **situation detection**, passes the image through a **convolutional neural network** which determines which of seven environment types the image represents. The number and detail of the environment classification can be further improved; however, in the system's current state, these seven states are meant to be linked to one of the seven musical genres (classical piano, contemporary piano, pieces from original soundtracks, pop music, rhythm and blues pieces, salsa pieces, and jazz) that have been selected.

Neural networks

Neural networks are the basis for countless machine learning algorithms and programs. In NIMA we make use of the following types of neural networks.

Deep neural networks (DNN's) have more internal layers than a regular/simple neural network and can usually offer more accurate guesses on the inputs given. In NIMA we have used a DNN to help us achieve accurate classification of environments.

Convolutional neural networks (CNN's) work by analysing inputs in small portions and then scaling up to analyse the full input (in this case a picture). By following this process, CNN's achieve a deeper understanding of the dataset they are trained from, in comparison to a regular/simple neural network, greatly improving the versatility and reliability the network has. In pictures and videos specifically, CNN's allow the network to identify a class's characteristics in unexpected positions or orientations (Rosebrock 2017).

The CNN used in this part of NIMA was fed images of the following environments/situations: bedroom, kitchen, living room, highways, exteriors, skyline, vegetation, and bodies of water. This network was built using only three convolutional layers to minimize processing times and will be optimized in the following months.

Music Generation

The second major section of the project is the music generation section. This part of our project heavily relies on the tools provided by Magenta (DeBreuil 2020), specifically its Performance RNN tool and Multi Conditioned Performance with Dynamics model.

Performance RNN is based on LSTM and RNN's, and it is designed to model polyphonic music that features the expressive tempo and dynamics that characterize music. This tool focuses on determining which pitches should be played, when to play them, and the amount of force that should be used when playing a note (Google Brain 2020). Performance RNN was trained using the Yamaha e-Piano Competition data set ; said data set features human-performed pieces, which are essential when training the model. Furthermore, the original data set was modified in order to be able to have more training examples. Specifically, the piano performances were modified by applying time stretching and transpositions (Ian and Sageev 2017).

Multi Conditioned Performance with Dynamics is a specific Performance RNN model that features more complex dynamics since it allows the user to determine the number of notes per second that the system should take into consideration when generating a musical piece. This model also takes into consideration a pitch class histogram that can be given by the user in order to control the primary pitches that will be featured within the musical piece. Furthermore, Multi Conditioned Performance with Dynamics has a feature called "temperature" that defines how random a track must be. Less than 1.0 makes a track less random and generates musical pieces that can be quite repetitive. 075 is the value set by default.

The music generating section is made up of two rule-based methods and a generating method that allow us to obtain the number of notes per second that should be played and the primary pitches that the musical piece should feature:

The **notes per second** are obtained by taking the brightness value. A range of 5 to 15 notes per second has been defined and the brightness value goes from 0 to 160. The brighter an image is, the more notes per second will be played in order to create a more cheerful and lively piece.

The **pitch class histogram** is obtained by taking the six predominant HSV values from the image and associating them with a chromatic scale value. The first person to ever associate color with music was Sir Isaac Newton with his Musical Prisms when he came up with the musical divisions of the prism back in 1670 (Hutcherson 2006). The idea behind it is that colors have a frequency of light in wavelengths within the visible spectrum that can be converted into Hertz and associated with one of the notes in the chromatic scale (Newton 1952). In order to associate a note with a color, we relied on Rodgers and Mehra's table of HSV values shown in Figure 1 (Ian and Sageev 2017).

Color	Hue	Saturation	Value
Black	$0° < H < 360°$	$0 < S < 1$	$V < 0.1$
White	$0° < H < 360°$	$S < 0.15$	$V > 0.65$
Gray	$0° < H < 360°$	$S < 0.15$	$0.1 < V < 0.65$
Red	$H < 11°, H > 351°$	$S > 0.7$	$V > 0.1$
Pink	$H < 11°, H > 351°$	$S < 0.7$	$V > 0.1$
	$310° < H < 351°$	$S > 0.15$	$V > 0.1$
Orange	$11° < H < 45°$	$S > 0.15$	$V > 0.75$
Brown	$11° < H < 45°$	$S > 0.15$	$0.1 < V < 0.75$
Yellow	$45° < H < 64°$	$S > 0.15$	$V > 0.1$
Green	$64° < H < 150°$	$S > 0.15$	$V > 0.1$
Blue-green	$150° < H < 180°$	$S > 0.15$	$V > 0.1$
Blue	$180° < H < 255°$	$S > 0.15$	$V > 0.1$
Purple	$255° < H < 310°$	$S > 0.5$	$V > 0.1$
Light Purple	$255° < H < 310°$	$0.15 < S < 0.5$	$V > 0.1$

Figure 1. Table of colors and their HSV values (Rodgers, L., & Mehra, S., 2013)

Finally, the **performance method** takes the values output by the two prior methods, sets the temperature to 0.75 by default, sets the length of the piece to 60 seconds, and passes the information to the model in order to generate the musical piece.

Conclusions and Future Work

While NIMA is not as complete as we intend to make it, we believe it has been set on the right track to accomplish creative music generation when paired with situations the system can detect. As we continue to further develop this software, it will grow closer to being a music composer and one that users could enjoy listening to in many given situations.

As of today, the software is capable of generating a piece of music by analysing images detected through a camera; however, these pieces can sometimes have musical mistakes or a melody or harmony that does not live up to the expectations of the user. These imperfections are currently being worked on and will hopefully be fixed in the near future.

In the months to come, we strive to better the situation detection section of the project. We are currently working on a model suited to our needs, however, fine tuning said model has presented a challenge. Additionally, we also want to find a data set or develop a synthesizer that will be capable of playing several instruments instead of just a piano, which is the instrument that the system is currently limited to; this would help us achieve our goals and produce more interesting musical pieces. However, this is not our main focus. Once the system is improved, we would like to subject it to a series of evaluations and tests.

Acknowledgements

This work has been supported by the Asociación Mexicana de Cultura, A.C. The making of this project was greatly facilitated by using the tools provided by Google Brain, and the Magenta team.

References

DeBreuil, A. (2020). *Hands-On Music Generation with Magenta*. Birmingham, UK: Packt Publishing. pp. 64 -70.

Google Brain. (2020). Performance RNN. Retrieved May 15, 2020. Available at: https://github.com/tensorflow/magenta/tree/master/magenta/models/performance_rnn

Hutcherson, N. (2006). Music for Measure: On the 300th Anniversary of Newton's "Opticks". Retrieved May 15, 2020. Available at: https://web.archive.org/web/20061219094447/http://home.vicnet.net.au/~colmusic/opticks1.htm

Ian, S. and Sageev, O. (June 29, 2017). Performance RNN: Generating Music with Expressive Timing and Dynamics. Retrieved May 15, 2020. Available at: https://magenta.tensorflow.org/performance-rnn

Ian, S. and Sageev, O. et al. (2017). Performance RNN. Retrieved May 15, 2020. Available at: https://colab.research.google.com/notebooks/magenta/performance_rnn/performance_rnn.ipynb#scrollTo=s9wvam1f7UnL

Inspired Acoustics. (2020). MIDI Note Number and Center Frequencies. Retrieved May 15, 2020. Available at: https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies

Malik, I. (2017). Neural Translation of Musical Style. Master of Engineering Dissertation, Cornell University.

Newton, I. (1952). *Opticks or A Treatise of the Reflections, Refractions, Inflections & Colours of Light.* New York: Dover Publications Inc.

OpenCV. (2020). OpenCV Tutorials. Retrieved May 2, 2020. Available at:

https://docs.opencv.org/master/d9/df8/tutorial_root.html

Rodgers, L., & Mehra, S. (2013). DressBest ColorPal. Retrieved May 15, 2020. Available at:

https://mehrarodgers.wordpress.com/2013/05/05/final-project/

Rosebrock A. (2017). *Deep Learning for Computer Vision*. PyImageSearch.

Sturm, B. L., Santos, J. F., Ben-Tal, O. and Korshunova, I. (2016). Music Transcription Modelling and Composition Using Deep Learning. First Conference on Computer Simulation of Musical Creativity. Retrieved May 15, del 2020. Available at: https://arxiv.org/abs/1604.0872