DELFT UNIVERSITY OF TECHNOLOGY

MULTIMEDIA SEARCH AND RECOMMENDATION 2020/2021 Q4
CS4065

CATMuG

Context Aware Tuneable Music Generation
Group 5, June 2021

Authors

Contents

1 General Information

1.1 Introductory description of the topic

Our chosen topic is Conditional Music Generation. Computational music composition is not a new idea, but it has seen a resurgence recently with the rise of deep neural networks. It is still quite difficult to produce realistic sounding music, however, and to the best of our knowledge there is no system yet that is situationally aware, i.e. it takes a certain context into account. However, there are certain moments where one prefers a certain kind of music over another. So, there is a need for a context-aware system.

To this end, we introduce CATMuG: Context-Aware Tunable Music Generation. CATMuG is a system that can generate new music based on some form of user input. This input can be used to tune the output of the music generation towards a certain desired context (i.e. emotion or mood). In our case, this context is encoded in values from the Valence and Arousal (VA) domain, but it could fairly easily be extended to work with different representations of context. The idea is that a user can directly influence the kind of music which is generated by supplying some form of a tunable parameter.

1.2 Motivation

This is an interesting topic for several reasons. First of all, the entire field of music generation is a fairly young one, so there are many unexplored avenues of which this project is one. Conditional music generation based on a specified context could be a solution for several problems. For example, the specific piece of music which a user would like to hear may not exist yet. It is especially difficult for someone to find new content if they have an exceedingly specific taste, a problem conditional music generation could solve. A different example is the problem of copyright. This is a big problem for content creators on social media platforms like YouTube or Twitch, who can't put copyrighted music in their videos or streams. Context-aware music generation could provide them with music that they can play with no problem, and still fit within a certain context. Lastly, it could be a good fit for video game developers. There has been a huge increase in games that have procedurally generated levels, but the music still requires composers. A system such as the one we are proposing could add another dimension of procedurally generated content, but it has to fit the level which was generated. Therefore, there could be a way to link the parameters used for the creation of both to ensure they fit together well. This is a very challenging topic, because music generation, in general, is far from solved, let alone with the inclusion of some input parameters to encode a context. Nevertheless, it is still a very interesting area to explore.

1.3 Link to CS4065

This project is related to many topics which have been covered during the course. The main topic it relates to was covered during the first lecture on multimedia search systems, specifically context-aware and affect-based search systems. While this project covers music generation, rather than music search, it deals with the same considerations and complexities related to the modelling of emotions. The built system directly uses the valence and arousal paradigm covered in this lecture, which provides a 2D affect space onto which the affect of the user can be mapped. This provides a more palpable and feasible way of reasoning about user affect and mapping system and musical features to affect. Another relevant topic is that of user modelling. As the project's topic relates to music generation based on mood and context, it is imperative to consider how this can be best captured and modelled for every user. Additionally, the same user-centred considerations regarding the system and its output which apply to search systems also apply in the current system. For example, whether or not the system output satisfies the criteria of the user. Finally, this project also relates to the lecture regarding musical feature extraction, as the Mel-frequency spectrum is used by the part of the system which handles VA value predictions. Audio is converted to a 2-dimensional mel-spectrogram which is used as the input for the VA regression models to predict VA values.

2 System Documentation

CATMuG

2.1 Link to video and repository

Link to our GitLab repository:
https://gitlab.ewi.tudelft.nl/cs4065/2020-2021/team05/catmug
Link to our video:
https://youtu.be/b5674pPCnLQ

2.2 System Documentation

2.2.1 Problem description

Music generation is music that is composed by computational means. Only since the rise of deep neural networks has music generation become a trending topic [28]. In recent years, a lot of deep neural network models have been proposed for music generation [7, 28, 17, 6]. In general, Recurrent Neural Networks (RNNs), Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs) and transformer-based architectures are often used architectures to compose music. Music generation in itself targets several problems, including copyright-related issues and exploring unheard music. However, what we focus on is generating music for a specific context or mood.

In the case of context-specific generated music, take the following examples. First of all, a game studio might want to accompany automatically generated levels with generated music, the music must, however, fit the context of the level. Therefore, being able to generate music that is specific to the context of the level would be required from the model. Secondly, people often listen to a playlist of songs. The songs can, however, become repetitive and bore the user. Most music platforms tackle this by recommending unheard songs. Yet, another option would be to feed the user-generated songs that are specific to the context of the playlist.

Some challenges arise when attempting to generate new music based on some context as input. For one, what kind of input should be used to represent a mood or context. Several approaches could work, but due to the limited scope of this project we have opted to use Valence and Arousal values [1, 2, 11]. There are, however, limitations to using this approach, which we will discuss in section 2.2.3.

2.2.2 Related work / products

Many generative models have been used to generate music. Especially deep learning models have had much success. One of the most popular models, from the Google Magenta Project, is MelodyRNN [25]; an RNN-based symbolic music (i.e. MIDIs) generator. Later on, two improved variants were introduced which attempted to capture long-term structure: the LookbackRNN and the AttentionRNN. Both of these allow the user to train the model on their MIDI collection and generate melodies based on a primer sequence. Another contender from the Magenta project is MusicVAE [20], which uses a sequential autoencoder with a hierarchical recurrent decoder to learn a latent representation of music. They then generate music by sampling from the latent space, operating with the assumption that close points in the latent space map to semantically similar data points. MidiMe [5] is a model which allows users to personalize MusicVAE using their data. It allows users to upload MIDI files and use the latent space (256 dimensions) from pre-trained MusicVAE models to train a subsection of this latent space (4 dimensions), allowing users to sample music similar to the input. With an online tool, the user can control sliders to interpolate between values in the 4 dimensions to generate music based on their input MIDI file.

There are currently very few mood-based music generation systems. Mood-based systems which do exist often use the valence arousal (VA) plane, as first proposed by Russell, [21] as a way to model affect (i.e. emotion). In this plane, emotion is modelled in terms of a two-dimensional plane containing 2 orthogonal dimensions, the valence dimension, which measures pleasure and displeasure, and the arousal dimension, which measures the degree of intensity. Kim et al. [12] leverage music mood tags with VA values to assign representative moods (e.g. exciting or sad) to each region in the VA plane and obtain a mood characterization for these regions. Interestingly, most regions were found to contain overlapping moods. Unfortunately, it is nontrivial to obtain these VA values for musical pieces. Yang et al. [30] attempt to predict VA values by using regression models on extracted features from song samples. It is found that predicting valence is much harder than predicting arousal. Oliveira & Cardoso [19] use extracted musical features to predict VA values for different musical segments from MIDI files.

Using VA values, systems have been created which can give effective music recommendations, possibly based on an input video [22] or image [23]. These systems map music and the input media to the VA plane and then recommend music based on the similarity between the music and the input media within this plane. Similarly, Kim et al. [13], use the VA-mood relation to recommend songs based on mood by exploiting relations between common emotional tags and the VA plane. Finally, Lopez et al. [15] used musical data with VA labels to create an emotion-driven music engine that can generate music related to a specific emotion by combining short MIDI segments into a music stream. However, this system works by combining musical segments from music that already exists, rather than generating completely new music.

2.2.3 Main functionalities, challenges and contributions

In this project, we propose **CATMuG: Context-Aware Tuneable Music Generation**. The **main functionalities** of our solution can be best described by the different stages in the pipeline of our program. Table 1 presents an overview of our system. We will first talk a little about the generation of music based on VA values, secondly, talk about the VA prediction model we trained, and finally discuss how this VA prediction model can be used to evaluate the generated samples from our music generation model.

Table 1: system overview

	functionality	method
VA prediction module	finding a song which has the matched VA value provided by a user	CNN
Music generation module	generating a music sample with the first bar from the song provided by VA prediction module	CGAN

Let us first discuss the most significant functionality of our project: the actual generation of music based on input VA values. We worked upon an existing model, namely MidiNet [28]. MidiNet is a Convolutional Generative Adversarial Network, which means that CATMuG is as well. we did not make any major changes to the actual model structure of MidiNet, so CATMuG follows the same architecture as Figure 1 from [28]. What we did extend on was the pipeline to get from input to output. Instead of a priming melody or chord sequence which is required for MidiNet, CATMuG can take VA values as input to generate samples. This process should, in theory, result in samples that have VA values similar to the values supplied as input by the user. This allows us to generate music based on context encoded in VA values. On a deeper level, this works by using VA predictor models we trained to give all of the input songs we have available (10,148 songs) VA labels. When a user subsequently provides VA values, as input, CATMuG finds the closest matching song available to it, and then uses that song as the priming sequence. This VA model was trained with a dataset [8] we found online of 200 MIDI labelled piano pieces of games. Each piece was annotated by 30 human subjects with VA values. Finally, our evaluation process consists of comparing the VA values of the input song with the VA values of the generated sample. Both of which are predicted using the VA predictor model mentioned before.

In the building phase of all the aforementioned functionalities, our group faced some **challenges**. We will elaborate further on a few of the biggest ones, motivating why these particular challenges were the ones we decided to focus on, and how we approached solving them. Additionally, there are also several limitations and potential pitfalls which we will discuss:

1. The biggest challenge we faced was the lack of available datasets. Both for the music generation model but especially for the VA prediction models. There just are not many VA labelled music datasets out there, and creating one ourselves would be far too costly and difficult given the scope of the project. In the end, we did find a dataset of VA labelled music [8], however, it was for video game music, and there were only 200 labelled tracks. Despite these domain and size limitations, we still used it because it was all we had available. For music generation, there was a decently sized dataset accompanying the MidiNet which was sufficient for our purposes. It was, however, all pop songs resulting in a mismatch between the data used for training the VA model (game music), and the data is would be used to predict values for (pop songs). However, as all data is in the midi format this discrepancy is a little less important.

2. Which brings us to our next challenge that relates to the data formats, which might not seem obvious initially. While we may have found a VA labelled dataset, one could ask themselves how reliable these ratings still are for the actual midi files. When the people originally annotated the data, they listened to it in .wav format, only after which it was converted to midi. This is a problem that persists throughout the entire pipeline of CATMuG. There are several points of data conversion, which is not a lossless process. Some information gets lost, be it actual bits of information, or just the general ”feeling” of the resulting audio. At this moment we do not have a direct solution to this challenge. However, it is extremely important to keep this matter in mind, as it may influence results.

3. Another challenge is the evaluation of the generated samples. This stems from the fact that evaluating music, in general, is quite difficult due to its highly subjective nature. Generated music has this same problem, although there does seem to be something 'off' about a lot of generated music that makes it sound fake. There are two main approaches to the evaluation of music, subjective and objective music evaluation. As a subjective evaluation would have to involve users in some way (e.g. some form of experiment or a more involved user study), and the scope of this project is fairly limited, we decided to only focus on objective evaluation. We take a closer look at some of the musical aspect of the music used as primers, and their respective generated samples. The idea is that there will be some correlation between these aspects, but more on this can be read in section 2.2.5.

4. Generating music based on context is a challenge in itself. It is a very young area of research with a lot of experimentation, so there are very few right or wrongs set in stone. Certain decisions have to be made. In our case, we decided to use VA values as a representation of context, but there are other systems. We will discuss this more in later sections, but we do not want to give the impression that this is the ideal choice in general. We picked it because it worked with the tools we had available to us at the time. This is the same for our generation model's architecture (Convolutional GAN). It worked best for us at the time given the tools we had available, but future research is necessary to create a better comparative analysis.

Finally, we give a short overview of our **main contribution**. Our main contribution is a proof of concept of a system that can generate music based on context. We present CATMuG, which uses a GAN to create new music from scratch based on some user input in the form of valence and arousal values. No such system existed before.

2.2.4 Techniques and technologies used

Given the limited time and resources available to us for the implementation part of this project, we decided very early on to use an existing model. There were a few options we looked into, like MusicVAE [20], MidiNet [27] and MuseGAN [6]. While the resulting generated samples of MusicVAE and MuseGAN did sound better in general, we settled on MidiNet in the end. There were several reasons for this. Firstly, MidiNet's code was much easier to interpret and easier to get a deeper understanding of. Especially as quickly getting into the project was a high priority for us. Relating to this, a PyTorch implementation existed, which our whole team has more experience with than with TensorFlow. Finally, MidiNet makes use of .midi files, which are easier to work with and more flexible and less complex than most other audio file formats like .wav or .mp3. The MidiNet repository also included a decently sized dataset of midi songs for training purposes. MidiNet is a Convolutional Generative Adversarial Network [27] and works by taking a (user-chosen) primer song as input and generating new music based on that song.

To also fulfil the goal of having the generated music be context-aware, a system had to be picked to translate this desired context to some form of input that can be given to the model. As mentioned before, we use the Valence Arousal domain [1, 2, 11] for this. This model was suggested to us by our TA and is one of the most widely used models to map emotions or moods to a 2-dimensional space. Additionally, we found a publicly available dataset of midi files with VA labels which further solidified our choice.

This dataset came in especially handy because, to evaluate whether our generated samples resembled the desired context, a system that could label music with VA values was required. Both because neither the different input songs for our music generation model nor its outputs were labelled. For this reason, we found a VA predictor regression model [4] which we trained using the dataset we found. This is a deep convolutional network that uses mel-spectrograms as input to predict VA values. This model was chosen because it obtained state-of-the-art performance VA predictions, showing improvements over classical approaches with more feature engineering (e.g. [30] [19]), especially when predicting based on only audio. Additionally, it achieved similar performance with other deep learning models and was one of the only models for which the code was published.

2.2.5 Evaluation methodology and results

In this project, we adopted objective evaluation methods based on music domain knowledge. Due to the limitation of time and resources, subjective evaluation methods are difficult to be implemented in our project.

Moreover, subjective evaluation methods are not suitable for evaluating large scale datasets. Thus, objective methods were implemented. Our evaluation methods include the three aspects: **(1) evaluating whether generated music samples are pleasing and variable; (2) evaluating the similarity between generated output music samples and priming music samples; (3) evaluating whether the generated music sample is in accordance with the user-defined mood.**

Before presenting evaluation methods and results, a brief explanation of music representations used in the music generation model is given in this paragraph. As mentioned in section 2.2.3, a priming melody encoding certain VA values chosen from a song was used as a condition in the music generation model. The song in the format of MIDI was divided into several bars. The bar was then represented by an h by w binary matrix X, where h represents the number of MIDI notes, and w represents the number of time steps in a bar. In addition, the first bar was chosen as the priming melody. The music generation model was trained on the conditions of these priming melody. Once the model was trained, the first output bar was generated on the condition of a priming melody from a real song, and the following output bars were generated on the condition of their previous bar, which is not from a "real" song.

The metrics of note count **(NC)** and average note duration **average ND** were implemented to evaluate our results from the first two aspects. These metrics are derived from previous research on the evaluation of generative models in music [29]. Brief explanations of these metrics are given as follows:

- Note count (**NC**) indicates the number of note classes used in a music sample. It shows the variability of a piece of music.
- Average note duration (**average ND**) indicates the average duration of consecutive notes in a music sample. It shows whether a piece of music is fragmented or not.

These metrics were applied to the test dataset consisting of 471 songs. To be specific, the metrics of NC and average ND were computed from the priming melody from a real song in the test dataset, the entire bars of the real song, and each output bar respectively.

Afterwards, a further analysis derived from [29] was implemented on the values calculated by evaluations metrics. Firstly, the mean and the standard deviation (STD) were computed from these values. The mean shows the overall characteristic in terms of NC or average ND, and the standard deviation indicates the reliability of the mean value. Secondly, distributions of these metrics of different groups, which include the group of priming melodies and the group of generated music sequences, were estimated by **Gaussian kernel density estimation**[1]. Thirdly, to compare the similarity between different groups, the Kullback-Leibler divergence (**KLD**) were calculated. It measures how the distribution of one group is different from the distribution of the other group in terms of **NC** or **average ND**, and a smaller **KLD** indicates that the two distribution share more similarities. The definition of KLD is given below [3]:

$$D_{KL}(P||Q) = \sum_{x \in \chi} P(x) log(\frac{Q(x)}{P(x)}) \tag{1}$$

Table 2: comparison between priming melodies and generated music samples

	mean	STD	KLD		mean	STD	KLD
the song of the priming melodies NC	5.96	1.67	2.33	priming melodies NC	2.89	1.20	0.01
whole generated music NC	14.71	2.73	-	first generated bar NC	2.51	0.65	-
the song of the priming melodies ND	5.68	3.01	0.97	priming melodies average ND	6.06	4.13	0.21
whole generated music average ND	1.33	0.33	-	first generated bar average ND	5.09	2.46	-

Table 1 presents the results of these evaluation metrics. A comparison between the group including the songs of the priming melodies and the group including the whole generate music is given below. In addition, the whole generated music consists of eight bars for each music sample. Compared with the songs of priming melodies, the whole generated music samples tend to have more variation in note classes and are more fragmented. It can be also observed in Figure 1 and Figure 2.

[1] implemented by sklearn.neighbors.KernelDensity with the bandwidth of 0.5

However, priming melodies and the first generated bars share certain similarities in terms of NC and average ND. It indicates that the first generated bar has a similar variation as its priming melody. Moreover, the first generated bar is not over fragmented. It might sound as pleasant as its priming melody.

Besides, comparing the statistics of the first generated bars with statistics of the other seven generated bars presented in Figure 1 and Figure 2, the other seven generated bars are more variable and fragmented. The characteristics of these seven generated bars are different from the priming melodies and the first generated bars.

In summary, the first generated bars share more similarities with their priming melodies from real songs, but they share more differences with their following generated bars. Meanwhile, compared with the real songs, the whole generated music samples tend to be more variable and fragmented.

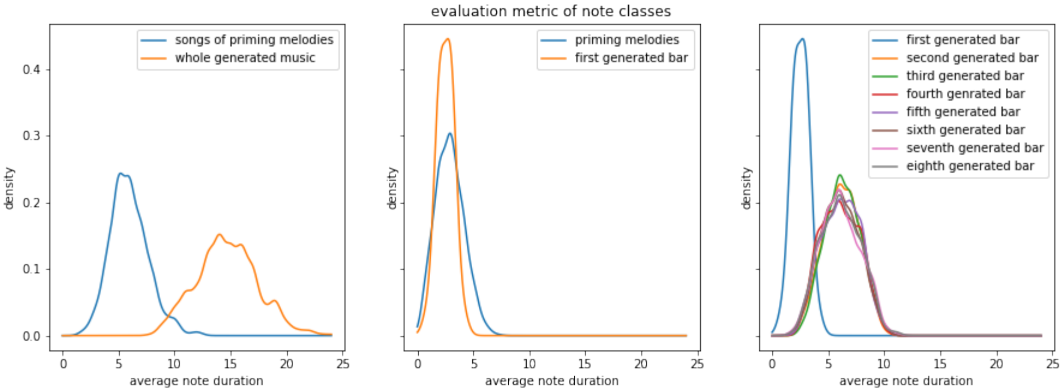


Figure 1: evaluation metric of note classes

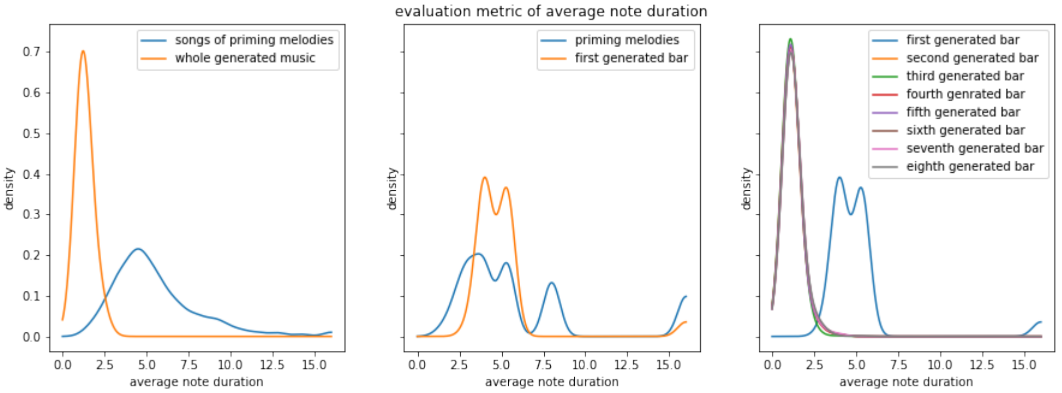


Figure 2: evaluation metric of average note duration

These findings can be explained by the music generation model. As previously mentioned, the music generation model was only trained on the condition of the first bar from a real song from the training dataset, which was used as a priming melody. Besides, the first bar of a piece of music may be different from other bars of the piece. For example, the first bar is a prelude and has a smaller **NC** and a longer **average ND** compared with other following bars. And once the model was trained, only the first bar was generated by the first bar in the input music sample, other bars were generated by the previously generated bar, which is a “fake” music sample. The music generation model might not be able to generalize well on these fake music samples. Thus, compared with the other seven generated bars, the first generated bar is more similar to the priming melody from a real song. In addition, there might also exist mode collapse in the music generation based on GAN, and it could result in the similarity in the other seven output bars.

With regard to the music generation model, several improvements could be made in the future. The music generation model could be trained on conditions of other bars rather than the first bar from a real song. For example, the bar which is in the middle of a song might possess more patterns, and the music generation model may capture more information about a real song.

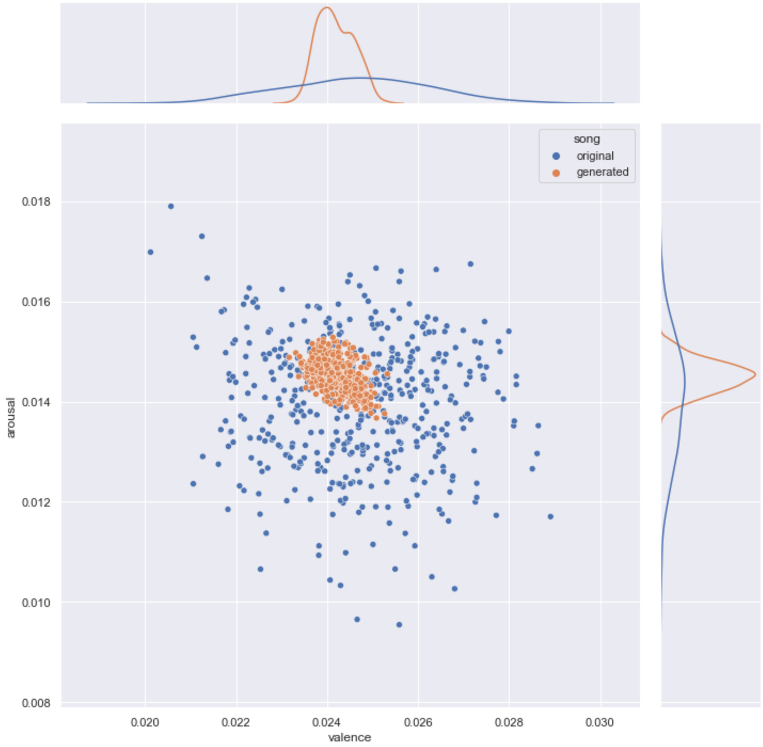


Figure 3: predicted valence and arousal values for all input and output songs

For the third objective of evaluating whether the generated music samples are in accordance with the user-defined mood, a VA predictor was used to predict VA values for the generated music so this could be compared to the VA of the music samples used as a primer. To compare the distributions of the input songs and generated songs, we plotted the VA values for all the songs. As can be seen from the figure, all of the VA values are very close to 0. One suspicion for this phenomenon was that the model was not generalizing beyond its training set. Due to a difficulty in acquiring data sets, the VA predictor model was trained with annotated videogame soundtrack songs, while the music generator was trained with pop songs. Retraining the VA predictor model by applying early stopping as a regularization technique gave very slight improvements to the diversity in values, but the model still suffered from the same effects. The leading hypothesis for this is simply that the model is not able to generalize to pop songs, as the videogame soundtrack data set does not contain enough overlapping features with the pop song data set to allow the model to learn to classify pop songs. Another possible factor could be the information loss associated with the conversion of MIDI to WAV of the sound files to allow them to be inputted into the predictor model.

Regardless, from the figure, it is clear that the generated songs exhibit much less variation and do not cover as much of the VA space as the input songs. One possible reason for this is that the generator model is suffering from modal collapse. Manual inspection of the generated music shows that the music patterns in the generated songs are often very similar, with the result almost always containing many short notes combined. This can also be seen from the previously displayed evaluation metrics, which show that the results each contain many notes with relatively short durations. This is especially pronounced for the latter seven bars, which have nearly identical distributions of **ND** and **NC**.

It is important to note when interpreting these results that even though the VA predictor model achieved relatively high performance on its training and testing set, it is a confounding variable in this evaluation, as it is difficult to separate which differences in the results are due to the ”true” VA values, and which are due to the predictor’s lack of performance. To settle this, a user study would need to be conducted.

2.3 Process

2.3.1 Team

The process of building the system had its ups and downs but overall went fairly smoothly. We started early by exploring different music generation models which were out there, and after some discussion decided on MidiNet. We then got it working, and gradually worked on finishing our pipeline by including other elements such as the VA prediction model. One of the biggest hurdles was finding datasets, and incorporating them into our solution. All in all, it was a gradual process slowly getting closer to being finished. It was sometimes challenging to get all the different components we use to work together properly, but we got a pipeline working in the end, albeit a little 'hacky' in some places. The final weeks were mostly spent on evaluation and the code necessary to support that and generate results. Everyone has worked on the code for a roughly equal amount, but different people focused on different things as will be explained below. Overall, we communicated well and at all times everyone knew what everyone else was working on.

2.3.2 Ricardo Jongerius

In the beginning, I was responsible to get the PyTorch implementation of MidiNet we found working, and get a first generated song as output. This proved quite difficult actually, as the model was made in Pytorch 0.4.1, which is very old now. This caused some problems with environments, but I got it to work in the end by using two different environments for training and the actual generation. I also wrote the "main functionalities, challenges and contributions" section, as well as "techniques and technologies used". Towards the end, I was also responsible for generating all the plots and visualisations in the report.

2.3.3 Gedeon d' Abreu de Paulo

For the system, I was in charge of modifying the different parts of the MidiNet code base and integrating it into a basic pipeline for CATMuG to allow for music generation based on a song primer. I also fixed some data preprocessing issues in the code and prepared the data for usage. Additionally, I created a system for choosing VA and for using selected songs as a primer for generation based on the chosen VA. I also worked with Jimmy on the VA prediction model to set it up for usage and integrate it into the generator model. Finally, I created a setup and generated data of all the preprocessed input songs and generated music for them to be used later in the evaluation step. For the report, I wrote the prior research section, researching many music generator models and VA related MMSR systems. Finally, I wrote the VA section of the evaluation.

2.3.4 Hang Ji

I was in charge of the evaluation of generated music samples from the music generation model. Firstly, I researched different evaluation methods from previous work. Then, objective evaluation methods were chosen for our project. Since there were no software tools for calculating these metrics, I wrote the code of these objective metrics for our music generation system. I also analyzed these metrics and wrote the **NC** and **average ND** section of the evaluation. I also included several explanations in our GitLab repository.

2.3.5 Jimmy Vlekke

Once our group settled on the mood-based music generation as our topic, I did a lot of research, including searching for a system that we could use as a base for our system. Once we found MidiNet, we still had to find a VA prediction model for which I was responsible. Eventually, after a lot of research, I found a paper with code. I immediately started to work on this model and integrating it into our system's pipeline. This task was not trivial, also because it was hard to find a dataset that we could use. Luckily, our TA hinted us a dataset, VGMIDI that is, which we ended up using. After training the VA predictor on VGMIDI, I used the trained VA models on all our valid songs for CATMuG to label them with VA values. I did the same of the generated songs that we used for analysis. I also added the VA prediction in the mood-based generation of songs. Finally, I cleaned up the code and updated the README. For the system documentation, I wrote the "problem description" section.

3 Research Proposal

CATMuG

3.1 Proposal summary

Music is an important aspect of the lives of many individuals and has the capability of greatly affecting the emotional state of the listener. Thus, it is no surprise that users are constantly on the lookout for new music, and being able to discover this music based on mood is a promising prospect. While the building of search and recommendation systems to aid the discovery of new music is an active area of research, a more obscure, yet promising area of research is that of music generation. CATMuG is the first mood-based music generation system that can generate completely new music using the music-domain-specific Geneva Emotional Music Scale (GEMS) while also being evaluated with user studies as well as objective metrics. To the best of our knowledge, there are currently no mood-based music generation models openly available to users. Furthermore, the existing models are either not evaluated with user studies, do not use an appropriate emotion classification scheme, do not generate new sounding music, or require the prior mapping of musical features to the emotional dimension.

CATMuG uses the GEMS system for mood classification, as it was built specifically for the music domain, is more preferred by users than the often used discrete emotion or VA models, is better at accounting for musical emotions and also takes into account the difference between perceived and felt emotions. It generates music using a deep learning model, which have achieved state-of-the-art performance in music generation, and uses a user study to properly evaluate the mood and quality of the generated music. Scientifically, this project would provide a baseline for other future models and provide an avenue for the analysis and evaluation of how well different moods can be captured. Finally, for users, it would be the only openly available mood-based music generation model, provide developers of media content a source of copyright-free music, and allow users to discover completely new music.

3.2 Extended synopsis of the project proposal

3.2.1 Problem description

It is commonly believed that music is capable of affecting our emotional state. The research field of music and emotion, a branch of music psychology, is devoted to understanding the psychological relationship between human affect and music. It has actually already been shown by means of experiments that music helps people satisfy their emotional needs [18]. Thus, music stimulates mood and, in general, music tends to be context dependent. This means that users on a music platform might only want to listen to certain songs within a specific context, such as working out or studying. Therefore, the user experience would greatly benefit from music platforms that can recommend songs well given the context and/or mood of a user. However, when assembling a music platform that would perform well at this task, there are some major issues that have yet not been successfully tackled yet.

First of all, there is a lack of common mathematical frameworks to describe all the relevant elements of emotion representations [24]. Yet, numerous attempts have been made to describe and classify emotions such as the Valence Arousal (VA) model. Therefore, the question we will research regarding this is as follows: **How should music be represented in the music domain?**

Secondly, music recommendation is a difficult task. An alternative to recommending a song that complements the given context is generating an unheard song. Music generation is, nonetheless, a difficult task too. But when generating music, there is more freedom to generate a song that complements the context well. Moreover, generated songs do not have the problem of possible copyright infringements. Furthermore, even though there are a lot of songs out there, this is definitely a finite set whilst the set of songs to be generated is infinitely large. The main question we want to research regarding context dependent music generation is as follows: **How can mood be captured to generate new music?**

Thirdly, computationally generating sound is not that hard, however, generating sound that sounds like an actual song is a lot harder. Generating music by means of deep learning models requires a robust and good working method of evaluating the generated songs. That's why the final sub-question is as follows: **How can the generated music be evaluated?**

And finally, the overarching research question related to all of these issues is: **How do you generate music based on mood such that the resulting music is satisfying to the user in terms of quality and context?**

As of now, there are some models that partly tackle these issues, however, such models either do not really generate music but combine existing pieces of songs, do not properly evaluate the generated songs, or use a subpar emotion labelling scheme such as VA. These will be discussed in more detail in the next section.

3.2.2 Previous work

To reflect the main components of the research question, the prior research will be divided in three parts: mood representation, mood-based music generation and model evaluation.

- **Mood representation.** There are multiple ways of assigning emotion to music. Many current systems use valence and arousal [21] as the basic dimensions for emotion. This circumplex model can be used to map emotion labels to the VA space. However, this might not be emotionally accurate, might be unintuitive for users (e.g. anger and fear are close in the circumplex, even though people perceive them as different emotions [26]) and might not be appropriate for music based emotion classification, because music-induced emotions have special characteristics [31]. Finally, it does not consider the difference between felt and perceived emotions. To create a domain-specific emotion classification model, an extensive study was conducted to create the GEMS (Geneva Emotional Music Scale), a 9 factorial model of emotions [31] that captures music-induced emotions better than basic discrete emotions and dimensional emotion models such as the VA space. The 9 dimensions are wonder, transcendence, tenderness, nostalgia, peacefulness, power, joyful activation, tension, and sadness. These can be further factored down to 3 factors: sublimity, vitality and unease.

- **Mood-based music generation.** There are very few mood-based music generation systems, and of the ones that exist, almost none of them are openly available for use and they all use the VA space. EDME [15], generates music by transforming and combining existent mood-labeled music pieces, based on emotion labels or VA values inputted by the user. One problem with this model is that the generated music stays recognizable. Wallis et al. [26] generate music from scratch by using rule-based algorithmic composition, but this algorithm requires the linear mapping of different musical parameters (e.g. pitch register, tempo) to the VA space beforehand. EmoteControl [10] allows users to control emotional expression in music by manipulating emotionally-relevant musical parameters. Unfortunately, the users need to input a MIDI file beforehand and there are numerous assumptions made for the input (e.g. on note durations and instruments) that were made in order to obtain reasonable results.

The field of style-based music generation is relatively unexplored. DeepJ [16] is an end-to-end generative model that can compose music for specific composer styles, providing users with multiple tunable parameters to control the generated music. This deep model uses a biaxial LSTM and can compose polyphonic music conditioned on composer styles by enforcing the musical style in the model's output. The only other major model is by Lim et al. [14], who use a variational autoencoder to generate music based on style.

- **Model evaluation.** In terms of model evaluation, Wallis et al. [26] evaluated how well their mood-based model captured mood by computing the correlation between perceived VA values and primer VA values for multiple generated music pieces. DeepJ evaluated both the generated music quality, and style. A subjective experiment was conducted to compare the quality of the results to another state-of-the-art model by using Amazon Mechanical Turk to allow users to listen to pairs of random samples between outputs of multiple models, and indicate which one they preferred. To evaluate the style, 20 individuals with musical backgrounds were asked to classify generated music into multiple classical styles. Half of the individuals were given real music as a control, and the resulting classification accuracies were compared. Additionally, they use t-SNE to visualize the style embedding space. Finally, Lim et al. [14] analyzed the reconstruction loss, the performance of a machine learning model which classified the styles of the generated output, and the number of chords and unique pitches.

3.2.3 Scientific contribution

The project aims to establish a music generation system which could generate music based on the desired mood of a user. The contributions include the following points: (1) implementing a music classification system which is interpretable to users and could classifies music into different emotions appropriately; (2) implementing a music generation system which could generate music based on desired emotion; (3) developing a user study and objective evaluation metrics to evaluate generated music samples with desired emotion. Besides, several innovations and challenges are also stimulated in this project.

- **Innovations.** The first innovation is the adoption a domain-specific emotion classification model. As mentioned in section 3.2.2, emotion classification models based on valence and arousal values have been broadly used in the field of music mood classification [4, 9, 12], and no emotion classification model based on the GEMS has been adopted. Compared with valence and arousal, GEMS is specifically designed to classify emotions triggered by music. Thus, music emotion classification models may alleviate the issue of domain mismatch. Moreover, the GEMS, a categorical emotion labelling method, is more interpretable to users compared with a dimensional emotion labelling method, such as valence and arousal values. Therefore, users are more easily able to interact with the music generation system with their desired mood.

  The second innovation is the implementation of a style-based music generation model for incorporating users' desired mood. As mentioned in 3.2.2, other music generation models require users specifying music parameters which represent certain emotion, and generate music based on conditions of these music parameters [10]. Compared with these models, our model could automatically discover the relationship between a piece of music and its emotion by the SOTA deep learning architectures. Thus, users could simply specify their desired mood instead concerning music parameters which are obscure for layman.

  The third innovation is putting forward both subjective and objective evaluation metrics to the field of music generation. From previous work, subjective experiments have been commonly used to evaluate generated music samples [16, 28]. However, subjective experiments cannot be conducted on a large scale dataset due to limitation of time and resources, and they are not suitable to compare different music generation models. Therefore, we put forward objective metrics to make evaluation efficient and comparable.

- **Challenges.** Several challenges need to be responded to during this project. These challenges are related to different aspects, which include dataset, music generation, and music evaluation. One challenge is acquiring a training dataset with emotion annotations. Since annotating music dataset with emotion labels requires much human labour and is subjective to different individuals, there are only a few datasets with emotion annotations. Meanwhile, these datasets are in different representations which include symbolic representations (i.e., piano rolls, midi events) and audio representation (i.e., spectrogram, waveform). Thus, a dataset which uses a different representation need to be converted to the desired representation used in our proposed system. However, the characteristics of the music could be degraded and distorted, which would change the mood and make the original music mood label inaccurate.

  The other challenge is how to evaluate the generated music. Although subjective evaluation methods are broadly used, they require a lot of human resources, and are difficult to reproduce, validate and compare the results. Thus, an objective evaluation method needs to be applied as a supplement of subjective evaluation methods in the project. However, there are no standard evaluation metrics introduced in the field of music generation.

The successful achievement of music generation system based on desired mood could have several impacts as well. The objective metrics which take mood into consideration would allow researchers make analysis and comparison between different models. Meanwhile, the music generation system based on desired mood is able to realize a cross-modal music generation system with the extracted emotion labels from other multimedia content forms including images and videos. Finally, the system could also generate copyright free music.

3.2.4 Methodology

In order to realise our goal of creating a system capable of context-aware music generation, we are faced with a few challenges which are also pointed out in previous section 3.2.3. These challenges are listed below. For each problem, we describe which techniques will be used to address them, and why this is the appropriate choice.

- **Data format.** First of all, there is a variety of different music format, each with their own advantages and disadvantages. For this project, we will make use of MIDI files. The biggest benefit to this that .midi files lend themselves very well to being converted to a datastructure which is easy to work with for code. This make sense, as MIDI files are essentially a list of time-stamped commands that are recordings of musical actions. While MIDI files contain less musical information, or can lack a musical 'feel' to them, they are a better fit for the system we are proposing. Another added benefit is that MIDI files are typically much smaller than digital audio files, which also eases the process of handling the data.

- **Emotion labelling.** Secondly, we have to select an appropriate emotion labelling system. In this research, we will make use of the Geneva Emotional Music Scale (GEMS) [31], as described in section 3.2.2. The reason this system will be used is that Zentner et al. found that it accounted for music-elicited emotions better than the basic emotion and dimensional emotion models. Additionally, the studies by Zentner et al. were performed specifically with music-elicited emotions in mind, instead of attempting to classify emotions in general. Therefore, we feel that GEMS is a good fit for our proposed system.

- **Dataset.** Third, we need to find available MIDI datasets, and label them accordingly. Luckily, there is a good amount of MIDI data available online[2], the main problem is labelling them. It is probably best to focus on a single genre to maintain some form of cohesion between all the music. The system could then be extended to a wider domain after it has proven to be successful. We suggest to take a classical piano rolls[3] dataset with around 1500 songs, as this is a workable amount to label. However, if there is more budget available the video game music[4] domain could also be picked. The budget is relevant because

[2] `https://github.com/wayne391/symbolic-musical-datasets` is a good starting point for example.

[3] `http://www.piano-e-competition.com/`

[4] `https://www.vgmusic.com/`

labelling will be done using Amazon Mechanical Turk[5], one of the easiest ways to get a high number of responses in a survey. The results will then be used to annotate each track with one of the nine labels from GEMS [31], as well as one of three of the overarching categories (unease, vitality, sublimity) from the GEMS paper to have access to a lower fidelity label if the performance proves insufficient.

- **Model.** For the music generation, the deep learning architecture of DeepJ [16] is used. This model can generate music based on composer styles through tunable parameters by using a generative Biaxial LSTM. This model was chosen because unlike other music generation models [17] [28] [25], it can capture multiple styles. Additionally, LSTMs have achieved state-of-the-art performance in music generation tasks. Furthermore, it is the only one which has been validated through user evaluations and comparisons with other deep learning models (unlike the other style based music generation model by Lim et al. [14].) It accepts a one hot encoded style vector, which is easily adapted to encode emotion labels instead.

- **Evaluation.** Finally, we will evaluate the system through a user study. This will be done once again through Amazon Mechanical Turk. Ideally, it will be the exact same people taking part. This is possible in Amazon Mechanical Turk. The idea of this second user study is to have the participants listen to the generated samples, and have them label those tracks again according to the GEMS system. We can then perform our analysis to see whether the generated samples' label correspond to the input label which was used to generate them. This set-up is similar to the user study from Wallis et al. [26]. In addition to this subjective user study, we also do some objective analysis comparing different (musical) properties of the tracks in order to measure sound quality. First we establish a distribution of certain properties like note class, duration, frequency etc. for each label from the GEMS system. Then, we compare the generated samples' values to the corresponding distribution to see how well it compares to the training data.

3.2.5 Project objectives, organization and planning

In this section we will translate the challenges from the previous section to concrete activities for each team member. We also provide a project timeline in Figure 4 as a global overview of the project. The bulk of the work on this research is going to be in three main areas: Data collection and annotation, Music generation model development, and Evaluation. These are the three phases this section will primarily focus on.

- **Data Collection and Annotation.** This problem is twofold, first we need to gather data, and afterwards we have to annotate each entry.

  Data collection is fairly straightforward, and can be done by a single person. The amount of data which can be used depends on available funding, as we will use Amazon Mechanical Turk (MTurk) for annotations which costs money per assignment (question on the platform). Depending on available resources, we propose to either use these 1573 classical piano rolls[6] in the case of low funds, or (a subset of) this video game music[7] dataset, which has over 28k songs in it. Of course, other datasets exist so further exploration could definitely be worthwhile. It is highly likely that someone also has to do some data cleaning and preparation to have it play nicely with the other components in the pipeline. These tasks relate to the Data section in the project timeline.

  Now we have data ready to go, but it still has to be annotated. As mentioned before, we will be using Mturk for this. One person should create a so-called 'Task' for this in Mturk. This has to be done sensibly though, so that there are at least a few annotations for each track in the dataset. This annotation process is, together with the evaluation, the most costly part of this research, but it is extremely valuable.

  The set-up of this Mturk task is as follows. First, the GEMS system will be explained thoroughly, so the users understand what each label means. After that, they will listen to around 10 songs, and then for each label say how well the song corresponds to that particular label. This is similar to the user studies

[5] https://www.mturk.com

[6] http://www.piano-e-competition.com/

[7] https://www.vgmusic.com/

conducted by Wallis et al. [26]. This will yield us a label, the one with the highest score, for each song in the dataset.

It is crucial that the GEMS system is explained meticulously. All participants have to fully understand the labelling system, in order to get good, valid results. This introduction requires a lot of attention. In the end, all the answered are consolidated to two labels for each song in the dataset. One specific label, which is one of the 9 labels from GEMS, and a more broad label, which is the hierarchical 'parent' of the label from GEMS (unease, vitality, sublimity).

- **Model development.** The next tasks relate to the development of the music generation model we will be using. As mentioned in the methodology, the choice of model is the deep learning architecture of DeepJ as defined in [16]. The implementation of this model is a task which one specific person from the team should focus on. A significant portion of the work required for this is self-reliant, and therefore can be done in parallel to other team members.

  For DeepJ, we have to use their model architecture, and instead of it using composer style vectors as input have it learn emotion vectors instead, based on the annotated training data from the previous point. Subsequently, such an emotion vector should be able to act as input to the model, so it generates music specific for that context. A big part of this task is also integrating the data into the model's pipeline, and some pre-processing might be required depending on specific issues which arise during model implementation. The final step is to train the model using our data. Depending on how big the dataset is in the end, this might take some time. Although even on consumer hardware, this should still be within the realm of days.

- **Evaluation.** Finally, we get to evaluation. Initially, some manual evaluation will have to be done by the team. This is to notice any big flaws in the system, and to play with hyperparameters in order to yield the best results for the final survey. This could take quite some time, as each iteration requires retraining of the model and reevaluation of the results. The objective evaluation metrics defined in Section 3.2.4 will play a huge role here. Each member of the team will also have to spend some time on manually evaluating the system.

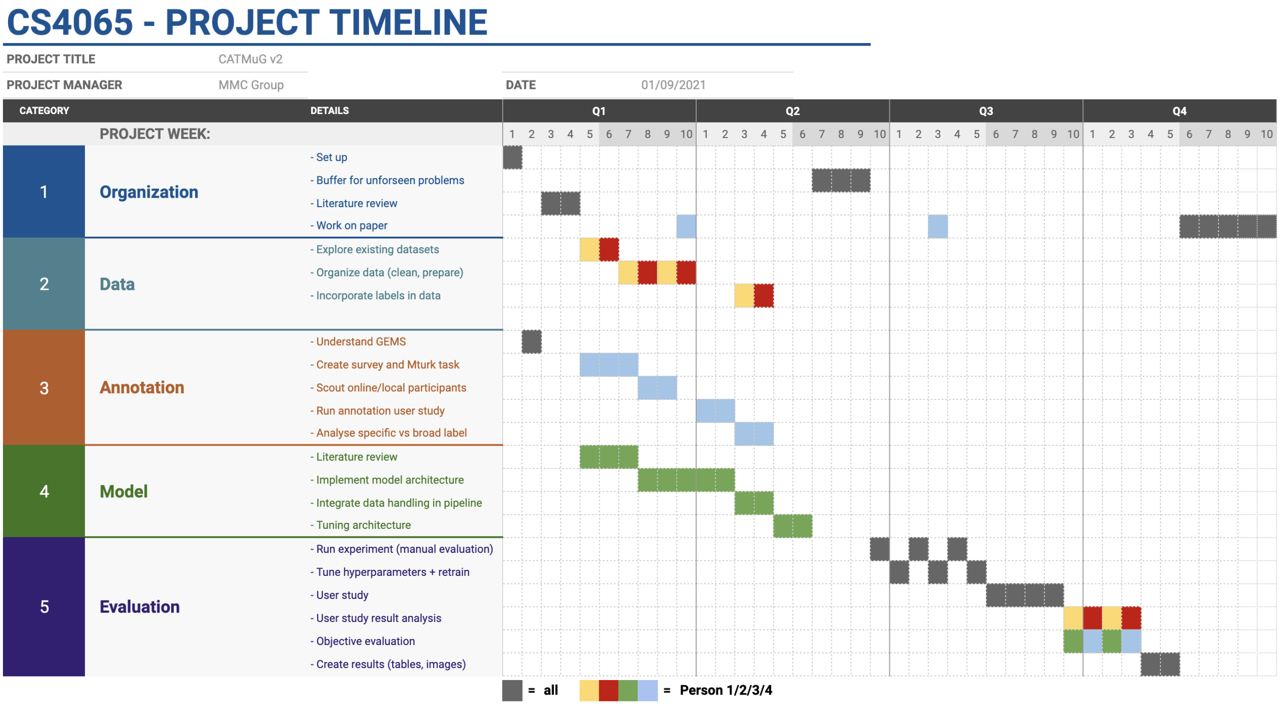
  After a few weeks, when the team is hopefully happy with the result, the evaluation user study will take place. This is once again done using Mturk. The setup is fairly similar to the annotation survey, except now the participants get to hear some of the generated pieces. They will hear both generated samples of music, as well as some original songs. They are then asked to label them with the GEMS labels, as well as say whether they think the music is real or generated. This provides valuable information on both the label of the generated pieces, as well as whether the system can convincingly generate new music. Once again, an exceptional explanation of the GEMS system is crucial to get reliable results.

In order to provide a detailed and comprehensive overview, we have created a Gantt chart to illustrate the project schedule shown in figure 4. It shows the timeline of the project, from start to finish, with all the weeks in an academic year represented as columns. Each section has different rows with the different sub-tasks for that section. Each person has a different colour, indicating what they will be working on each week.

3.2.6 Risk and mitigation plan

There are numerous risks that may occur which could prevent the successful completion of the project. Some of the most important risks relating to the perception of emotion, user experience and the music generator model are described below, as well as how they could be mitigated.

Risk(1): different people can experience different emotions for the same music, making it difficult to capture music related to a specific emotion. **Mitigation(1):** only include input data for which there is a certain degree of consensus among annotators. **Risk(2):** there are differences between countries and cultures with respect to how emotions are perceived. **Mitigation(2):** limit the scope of the project to a smaller region which is culturally and linguistically relatively homogeneous. **Risk(3):** there is a difference between perceived and felt emotions. This could confuse users and lead to the output not being what they expect. **Mitigation(3):** inform users on this difference, and explicitly mention that the project relates to felt emotions. The GEMS used was also explicitly developed to model felt emotions. **Risk(4):** the method of specifying mood might not be



CS4065 - PROJECT TIMELINE

PROJECT TITLE	CATMuG v2
PROJECT MANAGER	MMC Group
DATE	01/09/2021

CATEGORY		DETAILS
	PROJECT WEEK:	
1	Organization	- Set up - Buffer for unforseen problems - Literature review - Work on paper
2	Data	- Explore existing datasets - Organize data (clean, prepare) - Incorporate labels in data
3	Annotation	- Understand GEMS - Create survey and Mturk task - Scout online/local participants - Run annotation user study - Analyse specific vs broad label
4	Model	- Literature review - Implement model architecture - Integrate data handling in pipeline - Tuning architecture
5	Evaluation	- Run experiment (manual evaluation) - Tune hyperparameters + retrain - User study - User study result analysis - Objective evaluation - Create results (tables, images)

Figure 4: Project timeline in a Gantt chart.

pleasant to the user. **Mitigation(4):** conduct small HCI studies to evaluate and find a method which is most enjoyable users (e.g. using a smaller version of the GEMS model). **Risk(5&6):** the generator model may suffer from modal collapse or might not properly capture emotion. **Mitigation(5&6):** modal collapse is a notoriously difficult problem to solve, but one way of mitigating it is ensuring that the data set is diverse enough. For the second risk, this will be validated with the previously described evaluation studies in order to investigate whether this happens, why it happens, and the emotions for which it happens.

3.3 Process

3.3.1 Team

The process of writing this proposal was a bit hectic. As we realized that it was 50% of the grade, we attempted to start early. In the beginning, we had a great amount of difficulty coming up with a valid research direction, as we were also not sure what was expected, and how the research proposal would differ from the system documentation. This was compounded by the fact that none of us had much experience in the subject matter. As we worked on the system and did more research, we acquired more perspective on the subject matter, the pitfalls and the possibilities within the domain. Unfortunately, we had to start from scratch after the midterm, as it became clear that what we had written fit more in the system documentation. In the end, we decided to make the proposal a continuation of the work we did on the system. While we discussed the different sections together, ultimately each person in the group was responsible for writing a different section.

3.3.2 Gedeon d' Abreu de Paulo

I was in charge of writing the ”previous work” section and the ”risk and mitigation plan” sections. To this end, I did research on other mood-based music generation systems, (style-based) music generation systems, emotional classification studies and systems, the evaluation of generation systems and user studies for music labeling. After getting an idea of the intricacies involved with these subjects, I also wrote the issues that could come up and how we could deal with them. Finally, I also stayed up-to-date with the different sections and provided help with them.

3.3.3 Hang Ji

I was in charge of the section of scientific contribution. I did research on several topics related to the section of scientific contribution, which include music generation and music evaluation. By comparing our work with previous work, I summarized main contributions proposed by us. Meanwhile I also concluded several innovations which are the gap between our proposal and other previous work. Besides, I also pointed out related challenges that we might respond to during the process of implementation.

3.3.4 Ricardo Jongerius

I was in charge of two sections, ”methodology” and ”project objectives, organization and planning”. These were two fairly large sections, but they were also closely related. For the ”methodology”, I had to come up with a proposed approach to the difficulties of the research, and motivate why this is the correct approach. This was pretty difficult, as I have no experience in planning such a proposal whatsoever. Normally in projects, I tend to ’go with the flow’ which generally works out pretty well. For ”project objectives, organization and planning”, I had to come up with a more realised planning of concrete tasks relating to the methods which were proposed in the previous section. Again, this was new to me and not trivial, but I am content with my result. I also made a Gantt chart which I think helps convey the idea we had in mind regarding the planning and organisation.

3.3.5 Jimmy Vlekke

I was responsible for writing the ”problem description” section. In order for this section to be aligned with the other sections, I did my fair share of research on the topics covered in our research proposal. This includes research on music generation, music emotion recognition and evaluation using user studies. Since this section was restricted to be not that big compared to some of the other sections, I had the challenge of writing a dense text that contains all the necessary information without too much detail. Furthermore, I gave feedback on and adjusted the other sections. Finally, I also checked everything on language errors.

4 Reflection

4.1 Gedeon d'Abreu de Paulo

I did not have a lot of experience in MMSR topics, so much of what I had to do was new to me. I was able to follow most of the lectures, and did gain a nice overview of a wide variety of topics (e.g. audio features, factorization models, user modeling etc.), although I already had of experience with the machine learning related topics. When we were trying to choose a topic, we were considering music recommendation and music generation, so I also did research on music recommendation systems and learned about the main type of models (CBF, CF, multimodal systems etc.). I did not have any experience with music generation, so I learned a lot about the challenges in the field, and read papers on many models that are used. I learned about mood classification systems, such as the VA model (which I first heard of in a lecture) and the GEMS. I learned about problems with the system, and difficulties with classifying mood (e.g. differences across cultures, along time axis of music, how VA might not be extensive enough, differences between felt and perceived emotions etc.). I also learned about user evaluation, how user studies can be set up (such as for music generation systems, or the different studies conducted to develop the GEMS)

It was a bit frustrating in the beginning trying to find a clear direction of what to do as a project, as it was not completely clear what exactly was expected. I also do not have much experience writing research documents, much less research proposals, so I did get some experience from writing that. I do feel that I would've learned more if there was more guidance on writing a research proposal, and there was more frequent feedback moments on this. Regardless, I did learn to think more concretely about which challenges I am specifically tackling, as we were rather broad in the beginning. This includes reading literature to discover what difficulties current techniques have, identifying which problems still require a convincing solution, and what the intricacies are that make it hard to deal with each problem. Additionally, I learned to think more critically about what I propose in terms of how it solves a certain problem, justifying why I am doing it (including how it compares to the state-of-the-art) and to pay attention to and be critical when evaluating.

4.2 Hang Ji

In terms of the knowledge learnt during this project, I have acquired new knowledge which include music generation, evaluation of generated music, and emotion classification. Compared with other tasks which I carried out before (for example, hand written digit recognition, speech recognition and image classification), the task of context aware music generation is more difficult. The difficulties could be summarized in the following points: (1) there are no direct evaluation methods for music generation, and it makes comparison hard among different generation models. (2) the system has to take users into consideration, and a system which leaves users alone cannot perform well in real life. After realizing these difficulties, I found out that there are only a few products which could generate music just for users to listen. Most products related to music generation are only supplements to users. For example, Melody Mixer[8] based on the music generation model MusicVAE[9] from Google is design as a palette for composers or musicians. *In terms of the technical parts of this project*, I have gained several experience for formulating a research proposal. A concrete research problem cannot be formulated at the very beginning, it requires conducting some literature review with some possible directions in mind. Moreover, challenges and possible methods could be only put forward by proposing a concrete problem.

4.3 Ricardo Jongerius

I thought that this course was very interesting. While the lectures did not always correspond exactly to our specific project, they were still interesting and taught me a lot. I now understand the search and recommendation domain much better, and really liked the examples of when it does or does not work. Aside from the lectures, I also learned a lot about the topic of our project. I was completely new to music generation, but now know quite a lot. It was interesting to play around with the different existing models, and try to extend one with custom functionality. One big takeaway for me is that it's very difficult to convincingly generate real sounding music,

[8] https://experiments.withgoogle.com/ai/melody-mixer/view/

[9] https://magenta.tensorflow.org/music-vae

even when using state-of-the-art models. I also learned a lot from writing the proposal. While it was a little vague at times to understand what was expected of us, it taught me that all such research proposals take a lot of planning and consideration. Most of the time I would just 'go with the flow' in my coursework, and not plan too far ahead. However, I can now see the value in taking a critical look at what you are going to achieve, and how to get there in a concrete manner.

4.4 Jimmy Vlekke

When our group had to think of a subject for our project, I already had some ideas. I was mostly interested to research and on work on a project related to music recommendation as I'm not a huge fan of the recommendation algorithms of some of the music platforms out there. Thus, I coined this subject, however, we ended up picking slightly different subject, music generation. In my opinion, this turned out to be a great choice as I didn't know that much about this interesting subject. Researching this topic in combination with the lectures gave me a much better overview of multimedia search and recommendation and how the topics covered are relevant. I really enjoyed this method of learning.

Regarding music generation, I learned that this is not a trivial task mostly because generating sound that sounds like music is hard because defining evaluation metrics that can evaluate this is not easy.

For music emotion recognition, I learned that classifying emotions is also a difficult task. This is due to a lot of reasons which includes the difficulties of annotating music and issues with mathematical frameworks to robustly describe all relevant elements of emotion representations.

Finally, for the system we tried combining two relatively simple techniques to achieve a mood-based music generation model. This also turned out to be harder than I thought. But maybe due to the issues I have learned even more during this process than I initially expected.

The challenges that we faced in creation of our system but also during the process of writing the system documentation and the research proposal have learned me several key things. First of all, formulating a research proposal takes a lot of time and is not an easy task. It requires a lot of research and know-how of the field. Secondly, keeping things from becoming vague and making them concrete is also important. Thirdly, write down anything you come up with whenever it seems like an interesting or important thing as it might get lost in the process when you do not do this. Finally, evaluation plays a key-role, not only in a system but also when you are thinking about a research proposal.

References

[1] L. F. Barrett, *Discrete emotions or dimensions? the role of valence focus and arousal focus*, Cognition & Emotion, 12 (1998), pp. 579–599.

[2] S. Basu, N. Jana, A. Bag, M. Mahadevappa, J. Mukherjee, S. Kumar, and R. Guha, *Emotion recognition based on physiological signals using valence-arousal model*, in 2015 Third International Conference on Image Information Processing (ICIIP), IEEE, 2015, pp. 50–55.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[4] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, *Music mood detection based on audio and lyrics with deep neural net*, arXiv preprint arXiv:1809.07276, (2018).

[5] M. Dinculescu, J. Engel, and A. Roberts, *Midime: Personalizing a musicvae model with user data*, (2019).

[6] 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*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[7] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, *Neural audio synthesis of musical notes with wavenet autoencoders*, in International Conference on Machine Learning, PMLR, 2017, pp. 1068–1077.

[8] L. N. Ferreira and J. Whitehead, *Learning to generate music with sentiment*, (2019).

[9] J. Grekow and Z. W. Raś, *Emotion based midi files retrieval system*, in Advances in Music Information Retrieval, Springer, 2010, pp. 261–284.

[10] A. M. Grimaud and T. Eerola, *Emotecontrol: an interactive system for real-time control of emotional expression in music*, Personal and Ubiquitous Computing, (2020), pp. 1–13.

[11] E. A. Kensinger and D. L. Schacter, *Processing emotional pictures and words: Effects of valence and arousal*, Cognitive, Affective, & Behavioral Neuroscience, 6 (2006), pp. 110–126.

[12] J. Kim, S. Lee, S. Kim, and W. Y. Yoo, *Music mood classification model based on arousal-valence values*, in 13th International Conference on Advanced Communication Technology (ICACT2011), IEEE, 2011, pp. 292–295.

[13] J. Kim, S. Lee, and W. Yoo, *Implementation and analysis of mood-based music recommendation system*, in 2013 15th International Conference on Advanced Communications Technology (ICACT), IEEE, 2013, pp. 740–743.

[14] Y.-Q. Lim, C. S. Chan, and F. Y. Loo, *Style-conditioned music generation*, in 2020 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2020, pp. 1–6.

[15] A. R. Lopez, A. P. Oliveira, and A. Cardoso, *Real-time emotion-driven music engine.*, in ICCC, 2010, pp. 150–154.

[16] H. H. Mao, T. Shin, and G. Cottrell, *Deepj: Style-specific music generation*, in 2018 IEEE 12th International Conference on Semantic Computing (ICSC), IEEE, 2018, pp. 377–382.

[17] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, *Samplernn: An unconditional end-to-end neural audio generation model*, arXiv preprint arXiv:1612.07837, (2016).

[18] A. C. North, D. J. Hargreaves, and S. A. O'Neill, *The importance of music to adolescents*, British journal of educational psychology, 70 (2000), pp. 255–272.

[19] A. Oliveira and A. Cardoso, *Modeling affective content of music: A knowledge base approach*, in Sound and Music Computing Conference, 2008.

[20] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, *A hierarchical latent vector model for learning long-term structure in music*, in International Conference on Machine Learning, PMLR, 2018, pp. 4364–4373.

[21] J. A. Russell, *A circumplex model of affect.*, Journal of personality and social psychology, 39 (1980), p. 1161.

[22] S. Sasaki, T. Hirai, H. Ohya, and S. Morishima, *Affective music recommendation system reflecting the mood of input image*, in 2013 International Conference on Culture and Computing, IEEE, 2013, pp. 153–154.

[23] ——, *Affective music recommendation system based on the mood of input video*, in International Conference on Multimedia Modeling, Springer, 2015, pp. 299–302.

[24] V. Sethu, E. M. Provost, J. Epps, C. Busso, N. Cummins, and S. Narayanan, *The ambiguous world of emotion representation*, arXiv preprint arXiv:1909.00360, (2019).

[25] E. Waite et al., *Generating long-term structure in songs and stories*, Web blog post. Magenta, 15 (2016).

[26] I. Wallis, T. Ingalls, E. Campana, and J. Goodman, *A rule-based generative music system controlled by desired valence and arousal*, in Proceedings of 8th international sound and music computing conference (SMC), 2011, pp. 156–157.

[27] L. Yang, S. Chou, and Y. Yang, *Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions*, CoRR, abs/1703.10847 (2017).

[28] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, *Midinet: A convolutional generative adversarial network for symbolic-domain music generation*, arXiv preprint arXiv:1703.10847, (2017).

[29] L.-C. Yang and A. Lerch, *On the evaluation of generative models in music*, Neural Computing and Applications, 32 (2020), pp. 4773–4784.

[30] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, *Music emotion classification: A regression approach*, in 2007 IEEE International Conference on Multimedia and Expo, IEEE, 2007, pp. 208–211.

[31] M. Zentner, D. Grandjean, and K. R. Scherer, *Emotions evoked by the sound of music: characterization, classification, and measurement.*, Emotion, 8 (2008), p. 494.