Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Agnete Djupvik

Music That Feels Just Right

Masteroppgave i Informatikk: Kunstig Intelligens
Veileder: Björn Gambäck
Juni 2020

NTNU
Kunnskap for en bedre verden

Agnete Djupvik

Music That Feels Just Right

Masteroppgave i Informatikk: Kunstig Intelligens
Veileder: Björn Gambäck
Juli 2020

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU
Kunnskap for en bedre verden

Abstract

This Master's Thesis explores the combination of two artificial intelligence tasks: the challenge of music emotion recognition (MER), and the automatic composition of new music by using the emotion-annotated music as its basis. Methods for music emotion classification and emotion taxonomies are explored, and used as the foundation for automatic music composition based on emotion-annotated music data.

For emotion classification, a deep neural network is used on a 900-sample dataset of popular music. Music is processed as raw waveforms, without any pre-processing of specific music features. Emotions are categorized within four quadrants in the X/Y plane of valence and arousal.

For music composition, a self-attention-based generative model named the Pop Music Transformer is used. Music is represented as sequences of MIDI-like events, facilitating for long-range coherence, rhythmic patterns and local tempo changes. Training is done on the MAESTRO dataset, a dataset consisting of classical piano pieces, containing both MIDI and MP3 file formats of each sample.

Between the classification model and the composition model, an automatic pipeline taking a desired emotion as input is set up. The emotion classification system is used to predict emotions on the MAESTRO dataset. In testing, the system could mostly only predict music belonging to low-energy quadrants, due to the naturally lower energy overall in the classical piano genre compared to pop music overall.

The classification system reaches testing accuracy of between 50 and 60% in different experimental setups described in this thesis. Music composition is evaluated by the use of a survey with 101 participants, with the main purpose of discovering whether the intended emotions were indeed conveyed by the composed music. Survey results proved that the composed music did not adhere directly to the intended quadrants. However, valence levels proved somewhat distinguishable in the music composed, proving the system's ability to learn characteristic features of valence in emotions.

Keywords: Music emotion recognition, computational creativity, automatic composition, deep neural network

Sammendrag

Denne masteroppgaven utforsker gjenkjenning av følelser i musikk, og automatisk komponering av ny musikk ved bruk av den følelse-klassifiserte musikken som grunnlag.

For klassifisering av følelser brukes et dypt nevralt nettverk på et datasett bestående av 900 sanger innenfor populærmusikksjangeren. Musikken prosesseres som rene lydbølger, uten preprosessering av eksplisitte overordnede trekk ved musikken. Følelsene som brukes til klassifisering er kategorisert som fire kvadranter i et X/Y plan over valens og energi.

For musikkomponering brukes en generativ modell basert på kunstig relativ selvbevissthet kalt Pop Music Transformer. Musikken representeres som sekvenser av MIDI-lignende hendelser. Dette fasiliterer for sammenheng over lengre tid i musikken, rytmiske mønstre og lokale tempoendringer. Trening gjøres på MAESTRO-datasettet, et datasett som består av klassisk pianomusikk i både MIDI- og MP3-format.

I skjæringspunktet mellom klassifiseringsmodellen og komposisjonsmodellen er det satt opp en automatisk sammenkobling som tar inn en ønsket følelse som parameter. Klassifiseringsmodellen ble brukt til å predikere følelsene uttrykt i MAESTRO-datasettet. Systemet klarte i hovedsak kun å bruke de to lav-energi-kvadrantene i denne klassifiseringen, grunnet den naturlige lavere energien man i hovedsak finner i pianomusikk sammenlignet med popmusikk generelt.

Klassifiseringsmodellen nådde testnøyaktighet på mellom 50 og 60% i en rekke eksperimentelle oppsett beskrevet i denne masteroppgaven. Musikkomponeringen ble vurdert ved hjelp av en spørreundersøkelse, som hadde som hovedmål å undersøke hvorvidt den ønskede følelsen virkelig ble formidlet i den komponerte musikken. Resultatene fra spørreundersøkelsen viste at den komponerte musikken ikke svarte til tersklene for de ønskede kvadrantene. Allikevel var det mulig å til en viss grad skille resultatene for de ulike kvadrantene, særlig innenfor valensaksen, noe som viser systemets evne til å lære seg særtrekk for hva som utgjør høy og lav valens i musikken.

Preface

This Master's Thesis was written during the Fall of 2019 and Spring of 2020 as part of my Master of Science (MSc) degree in Informatics at the Department of Computer Science (IDI) at The Norwegian University of Science and Technology (NTNU).

I would most like to thank my supervisor, Björn Gambäck, for help and guidance throughout this year.

I would also personally acknowledge Jussi Karlgren, Simon Durand and Ching Sung at Spotify for their interest and help, and for sharing insight into Spotify's methodologies to further my work.

I also appreciate Renato Panda at Universidade de Coimbra, Portugal, for his tremendous effort in providing data for my research, as well as being a resourceful discussion partner and for promoting collaboration across country borders.

I would also like to thank Marinos Koutsomichalis for additional guidance on relevant AI topics to get me started on this task, as well as for giving feedback on my work and experiments.

Agnete Djupvik
Trondheim, 31st July 2020

Contents

List of Figures

List of Tables

Acronyms

AI Artificial intelligence. 1, 6, 31

ANN Artificial neural network. v, 6, 27, 28, 45

API Application Programming Interface. 11, 32, 33, 35, 38, 46, 48, 49

CNN Convolutional Neural Network. 28

DAW Digital Audio Workstation. 12, 13

EA Evolutionary algorithms. 9

GAN Generative Adversarial Network. 12

GPU Graphics Processing Unit. 11, 44

MER Music Emotion Recognition. 2, 3, 17, 23, 32, 35, 64, 68

MIDI Musical Instrument Digital Interface. 13, 27, 30, 43

MIR Music Information Retrieval. 2, 33

MSE Mean Squared Error. 10

ReLU Rectified Linear Unit. 43

REMI Revamped MIDI-Derived Events. 30, 43

SLR Structured literature review. 4, 15, 16

SVM Support Vector Machine. 9, 12, 28, 32, 57

VA Valence and Arousal. 24, 35

VAE Variational Autoencoder. 12, 29

1. Introduction

Music is all around us every day, and so are the technologies driving its development. From the first synthesizer (Pinch and Trocco, 1998) to the first artificial holographic artists (Johnston, 2008), technology is a driving force within defining what music is and can be.

Musical Computational Creativity is a field within Artificial Intelligence (AI) which has seen considerable growth within the last decades (Dannenberg, 2006). Within the field, algorithmic composition of music first came to existence in the 1950s, initially with the Illiac Suite (described by Hiller and Isaacson, 1958). Since then, a plethora of solutions and architectures have been proposed to understand and synthesize music and other forms of art, and often with impressive results. Technology can contribute as an instrument not only to the performer or audio technician, but also to the composer, as a creative partner or even as an autonomous agent (Saunders, 2012). In this thesis, I explore automatic composition in the context of emotion, by attempting to understand emotions conveyed by music and using that information to compose new music in line with those emotions.

1.1. Background and Motivation

The notion of creativity, though a feature of human intelligence in general, is a fundamental challenge for artificial intelligence (Boden, 1998). Rather than following strict rules, creativity is what allows the connecting elements that are known, but hitherto considered unconnected, in new and novel ways. A creative process can depend on a great number of cognitive aspects (Lubart, 2001), such as personality (Wolfradt and Pretz, 2001), cultural background (Bruch, 1975), personal preferences (Houtz et al., 2003), and analogical thinking (Dahl and Moreau, 2002).

Boden (1998) distinguishes three forms of creativity. The first is "combinational" creativity: the novel combinations of familiar ideas and concepts. The second is "exploratory" creativity: the generation of new ideas within the exploration of already structured conceptual spaces. The third is "transformational" creativity, closely related to exploratory creativity, but with such results that new conceptual spaces arise. Cases of the latter form are often considered the "revolutions" or paradigm shifts within a field; one example being the transition from analogue to digital music production and performance. With the rise of artificial intelligence, one such paradigm shift in the future could be the automated composition of music in such a fashion that it is indistinguishable to that composed by humans.

Within creativity, AI has often come to a loss (Rowe and Partridge, 1993). Strict

and elegant algorithms are often not enough to discover the perfect trade-off between that which is considered genuinely novel or creative (Grace and Maher, 2019), and that which pleases an audience. In music, the individual listener may have widely different preferences depending on their mood, location or even time of the day. This Master's Thesis will be exploring the very notion of music that feels *just right* for the listener by the classification of emotion within music. This means exploring how AI models can "learn" which emotions are conveyed by music, and using that information to compose new music which hopefully can express the same emotion.

1.2. Goals and Research Questions

Goal: *Classify emotion in music and use the classifications in automatic music composition.*

The goal for this project is to explore methods of interpreting and generating music in accordance with a given emotion or mood, often referred to as Music Emotion Recognition (MER), as a subfield of Music Information Retrieval (MIR). To reach this goal, the state-of-the-art within the field will be identified, and experiments will be conducted in order to compare different methods and benchmark criteria.

The exploration will focus on the intersection between:

1. The process of computer-based "understanding" of the emotions expressed in music.
2. The process of synthesizing new music based on the acquired understanding.

Research question 1 *What are suitable methods for computer-based classification of emotion in music?*

This research question involves understanding how this task has been performed using different technical approaches, as well as comparing existing work to new approaches used in similar fields.

Research Question 2 *What are sets of emotion categories that are comprehensible and effective for machine learning use?*

What labels or categories are used in the classification of music can greatly impact the results; *happy* might suffice as a category in some regards, but sub-categories such as *exhilarated* or *peaceful* may be very different things. Categories of emotion can be both broad and narrow, and it is important to explore how the choice of categories affect the quality of the classification process. This research question mainly involves exploring different opinions in the field of emotion psychology, and addressing their potential usability in a machine learning context.

Research Question 3 *What are relevant and efficient methods for creating emotion-based computer-generated music, and evaluating it?*

This research question explores different novel methods used for the computer-based composition of music. This includes fully automatic composition, as well as composition based on specific input such as a given emotion, or musical data conforming to that emotion. In evaluating the output by a mood-based digital composer, the human opinion is naturally important. However, human evaluation is slow, labour-intensive and possibly prone to bias (Hashimoto et al., 2019). Different human evaluation methods, as well as automatic evaluation measures, perhaps in combination, should be compared.

Whether and how the goal, and subgoals in the research questions, have been achieved are summarized in Chapter 7.

1.3. Research Method

The three research questions require somewhat different methods. All include research on the state-of-the-art. Also, the conclusions of each RQ influence the answer of the others; if one set of emotion categories proves clearly superior to others, it should be expected to see that set used in work regarding MER.

In answering **RQ1**, an experimental approach will be taken. Experiments will be conducted on creating a emotion classification system for music, which creates results that can be compared to some approaches in the state-of-the-art, for example, by using common data sets. Different configurations with regards to data representation and machine learning architecture will be explored.

RQ2 is a research question which will take a more analytic approach, in the intersection between emotional psychology and computer science. This involves exploring different opinions in the field of emotional psychology, and attempting to uncover common ground in what is considered reasonable categories that can be used in classifying emotions. Finally, promising alternatives will be viewed in the context of usability within machine learning, namely how well the categories can be expressed in a technical context.

For **RQ3**, the objective is to discover existing methods of generating music, and how they can be implemented (if they are not already) to learn what constitutes emotions within music. This may mean training separate models to understand different moods, or more explicitly determining rules for different moods in music. An experimental approach will be taken here as well, by using classification results discovered in RQ1 to train a model to compose new music based on the created classification. To evaluate the results of the composed music, a survey will be performed where participants will be involved in determining whether the music played does indeed conform to the intended emotion. Also, musical quality overall will be assessed with the same method.

1.4. Contributions

This section describes the contributions made to the field of musical computational creativity, and Music Emotion Recognition in particular, which can be found in this thesis.

1. A Structured literature review (SLR), data extraction and synthesis studying the state-of-the-art within musical computational creativity related to emotion and mood understanding.
2. A comparison of sets of emotion categories, and their features and drawbacks with regards to use in machine learning.
3. A study of relevant machine learning methods and data sources for the task of understanding mood in a music data set, and an experimental system for classifying emotions in music, created to compare performance with relevant approaches.
4. An implementation of a system architecture producing music in accordance with some given emotion, and a survey designed and conducted to uncover to which extent the emotion is recognized in the newly composed music.

1.5. Thesis Structure

- **Chapter 2** introduces background theory useful for the readers, both related to music and to relevant technologies.
- **Chapter 3** presents reviewed related work within the field, as well as the protocol for a Structured literature review (SLR).
- **Chapter 4** presents a system architecture suited for experiments to provide answers to the research questions posed.
- **Chapter 5** presents the conducting of the experiments, and a survey designed and conducted for evaluation of the composed music.
- **Chapter 6** discusses the results, their strengths and weaknesses, possible improvements, and possible sources of bias.
- **Chapter 7** presents the conclusion and final answering of the research questions, and a description of future work.

2. Background Theory

This chapter introduces some important concepts that are used in related work and throughout this thesis. Section 2.1 presents basic musical concepts. Section 2.2 presents some technologies that are or have been used in the understanding, classification or composition of music.

2.1. Musical Background Theory

This section is intended for the reader with limited familiarity with musical concepts and terms, providing a basic framework required for the discussions in this project.

Notes

Music is made up of notes, each one with a given *pitch* and *duration* (Strayer, 2013). A pause is the absence of pitch for a given duration. A pitch is the frequency of the sound wave emitted, and the pitch is higher with higher frequency, and vice versa. Select, discrete pitches (notes) are named in the scale pattern A, B, C, D, E, F, G, A, where the return to the starting note constitutes an octave. However, the frequency has doubled, meaning that the two notes are distinct. This is named in the pattern A1, A2, and so on. As audio frequency is a continuous measure, *semitones* can be found halfway between the standard tones. As an example, a semitone between A and B is denoted as $B\flat$. Even smaller intervals are denoted as microtones (Botros et al., 2002).

Chords

A chord is the combination of three (triad) or more notes played simultaneously. One of the notes is denoted as the chord root, and the other two (in a common triad) are most often two and four steps above the chord root in the scale.

Musical key

The key of a piece indicates the chord that forms the basis of the music. In most Western, popular music, music starts and comes to rest in one key, while notes and chords other than the initial key creates tension and variation. A more permanent change of key within a song is called a modulation. A key is in a mode, most commonly *major* or *minor* mode. In Figure 2.1, a chord in C major is presented, followed by a chord in C minor.

Duration and measures

As previously mentioned, all notes and pauses have a duration. To denote this, the concepts of beats and measures are used. A note's duration is represented as a fraction, most often $\frac{1}{1}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$. The notes can be combined to form more complex durations.

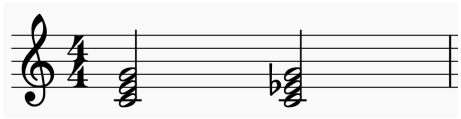

Figure 2.1.: A chord in C major, followed by a chord in C minor.

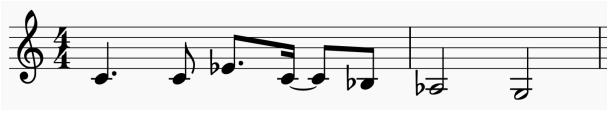

Figure 2.2.: The theme for The White Stripes' *Seven Nation Army*.

A *measure*, or bar, is a segment of time corresponding to a certain number of beats. A *time signature* is a notational convention specifying the number of beats contained in one measure, and which note value is equivalent to a beat. In Figure 2.1, the time signature is $\frac{4}{4}$. The numerator indicates that there are 4 beats in one measure, and the denominator indicates that a $\frac{1}{4}$ note is equivalent to one beat.

The *tempo* of a song is measured in the number of beats per minute (bpm), e.g. a tempo of 60 bpm indicates that there is one beat per second.

Music structure

Music varies greatly in structure, but some common themes exist. Popular music most often contains one or more verses and choruses/themes which repeat, and build and release tension throughout the music. Repetition and variation upon familiar themes is an integral part of musical structure, and is often what sounds familiar with music one has heard before. A widely known example, from the song *Seven Nation Army* by The White Stripes, can be found in Figure 2.2. This two-measure figure is repeated throughout the song using different instruments and fortitude, adding structure and coherence to the music.

2.2. Technical Background Theory

For the task of classification within Artificial intelligence (AI), the number of approaches that can be taken is almost unlimited. This section introduces some of the most used distinct architectural approaches.

2.2.1. Artificial Neutral Networks

Artificial neural network (ANN) architectures are networks of nodes and weights between them, architecturally inspired by the brain (McCulloch and Pitts, 1943). The network is designed to "learn" some task given training with data examples. Three main components are defined, namely an input layer, a hidden layer and an output layer, which interact in determined ways so that "patterns" in the network form during training based on correcting the errors made (Dreyfus, 1973), allowing the network to continue performing

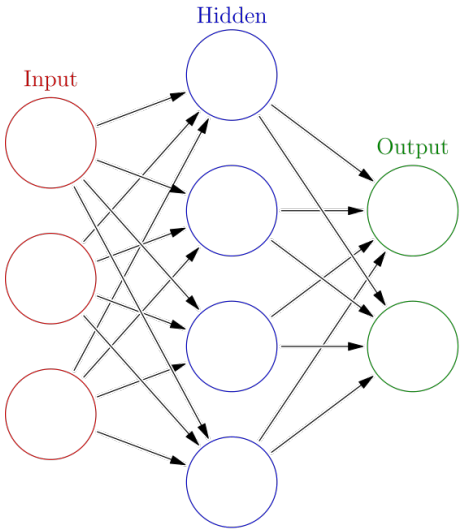

Figure 2.3.: A simple ANN with input, hidden and output nodes.

the task on its own when training is complete. Many different ANN designs exist, suited to different tasks such as image (LeCun et al., 1989) or audio recognition (Sak et al., 2014), classification (Sengupta et al., 2016) or generative models (Goodfellow et al., 2014).

The nodes and weights in an ANN are designed as a way to form "paths" in the network for certain inputs (Minsky and Papert, 2017). In some ways, the design can resemble a decision tree, or a directed and weighted graph (Zell et al., 1994), where one chooses branches along the tree depending on certain features of the input. However, there are two distinct differences. First and foremost, there is no direct representation of what a single weight in the network tells you – meaning that at first, the network will make arbitrary decisions. However, and secondly, the weights change importance as more information is learned. If a single "path" in the network is always correct, it will learn to almost always go that way (Zell et al., 1994). When the network makes a wrong decision in training, the weight of the path taken is reduced according to the error by backpropagation (Goodfellow et al., 2016), making it more likely to make another decision the next time. The goal is to ensure that the network can correctly differentiate different features of the input, but also that it can handle variations in novel input data by making more general assumptions where possible (i.e., avoiding overfitting the model to the training data). During training, the network essentially guesses predictions in the beginning. The accumulated error, or *loss*, is used as an indicator as to "how wrong" the network predicts overall (Dreyfus, 1973). This error can be seen as a point on a field of *gradient descent*, as visualized in Figure 2.4.[1] The goal for an ANN is to move to a global "error minimum", where weights are configured just right so that error is as low as possible.

[1]Source: https://en.wikipedia.org/wiki/Gradient_descent

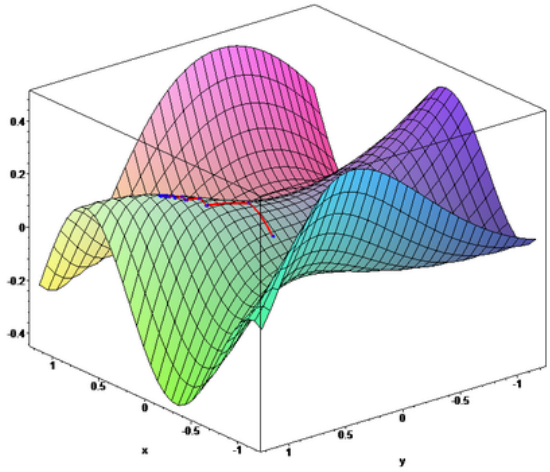

Figure 2.4.: Example of gradient descent.

Optimizing an ANN: Adam Optimizer

For an ANN, many different parameters are adjustable to create a network that is well-suited for the problem at hand. One crucial parameter in this regard is the learning rate. For the gradient descent plane in Figure 2.4, the learning rate indicates how large "jumps" should be made in the plane (Li et al., 2009), i.e. how much weights in the network should change given the error. If the learning rate is very high, one risks that a global minimum is skipped. If the learning rate is too low, one might get stuck in local minima, and never move far enough to discover the global minima. Thus, finding the optimal learning rate is a difficult task.

The Adam Optimizer (Kingma and Ba, 2015) is a different approach to the optimization of network weights. Instead of the gradient descent approach with a single learning rate, the Adam algorithm maintains separate learning rates for each network weight (parameter), which are adjusted individually as training progresses. The adaptation is not only done based on overall error, but also the momentum of which the gradients for one parameter has been changing recently. Bias-correcting is performed automatically by using both the first moment (the mean) and the second moment (the uncentered variance) of the gradients separately.

In practice, Adam is a highly effective optimization algorithm which adapts the learning rate to the problem at hand directly. It is used in the system described in Chapter 4.

2.2.2. Deep Learning

Deep learning is a term used for networks designed for representation learning; that is, learning not only what some input *is*, but higher-level features such as what it *represents* (Bengio et al., 2013). The motivation comes from the massive amounts of information in our world that is naturally unstructured, often lacking specific, discrete or measurable features, and therefore not suited for simpler, more straight-forward networks.

As an architecture, deep learning builds on the ANN architecture, but is distinguished in the use of multiple layers in the network (Schmidhuber, 2015). The process of extracting higher-level features is done by using these multiple layers with the purpose to serve many different tasks. As an example, with image processing, one layer may identify edges, while another identifies basic shapes, and yet another may identify faces. These long chains of processing between the input and output are what constitutes the "deep" property.

2.2.3. Support Vector Machines

Support Vector Machine (SVM) architectures are supervised discriminatory learning models used for classification and regression analysis (Cortes and Vapnik, 1995). In the SVM model, input samples are represented as a point in a space, mapped so that separate labels or categories are as far away from each other as possible. When a new sample is tested, it is placed in the space based on similarity to other samples, and classified to the category to which it is the closest.

2.2.4. Fuzzy Logic

Fuzzy logic, or fuzzy sets, is a term used for classification where truth values of a variable may be any real number between 0 and 1, inclusive (Zadeh, 1965). It handles the notion of partial truth, and for classification, this implies that a sample can (partially) belong to several classes.

For music emotion classification, this could mean that a sample can be classified both as, e.g. sad and tired, or energetic and joyful.

2.2.5. Evolutionary Algorithms

Evolutionary algorithms (EA) is an optimization approach which mimics evolution observed in the living world (Vikhar, 2016). When considering a problem, a *population* of solutions is produced. The solutions are ranked against a given fitness function, and the N least fit solutions are discarded. This is done repeatedly for many *generations*, often incorporating various sorts of *mutation*. Mutations could be allowing some weak solutions to continue to the next generation, or by direct mutations on the solutions such as *gene swapping* or *crossover*, namely scrambling parts of the solution either within one solution or between different solutions.

2.2.6. Evaluating a Classification Model

When developing a model for classification, several measures can be used to determine the quality of the model. This section presents a selection of benchmark measures used both in related work, and to evaluate the systems developed for this thesis.

Loss

The loss, simply put, is a function which describes "how wrong" the model is in its predictions (Dreyfus, 1973). Many different loss functions exist, while some are more

common. The loss function is used throughout the training. The most straightforward loss functions add up all errors (i.e., how far away the prediction was from the actual label) and take the average of them. Another approach is the Mean Squared Error (MSE), which squares the errors before taking the mean, punishing severe error far more (Lehmann and Casella, 2006). A third approach is the Cross-Entropy function, which measures the difference between two probability distributions: the true distribution and the predicted distribution.

The goal in a loss function is, naturally, to minimize loss. A loss of `0` would indicate that the model predicts every case perfectly. However, a loss of `0` is mostly not desirable, as it would indicate that the network is tuned specifically to the input it gets, with no mutations or exceptions. In turn, this may indicate that the model has not developed the ability to generalize. Thus, the loss can not be viewed as a success measure alone.

Testing set accuracy

When training a model on a data set, a segment of the data should be kept out from the training progress, constituting a *testing set*. When the model has trained, its predictions on the testing set is a test of whether the model has developed knowledge applicable to other data than its own training set. For accurate results, it is crucial that the model does not get to train on the testing set, so that the testing set serves its purpose for all epochs of training.

Determining testing set accuracy can be done in multiple different ways. If the model is trained using a set of discrete labels, one can simply measure how many predictions were correct out of the total. For continuous or more complex labels, quality can be measured more similarly to loss computation.

F-measure

The F-measure is a weighted combination of two values of classification performance (Derczynski, 2016), namely:

1. *Precision*: how many of the classifications of class X, actually belonged to class X.
2. *Recall*: how many of the samples belonging to class X were classified as X.

The F-measure is then expressed as the harmonic mean of precision and recall:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In total, this measure measures both how many samples were accurately classified, and how many samples should have been returned for each class.

Confusion Matrix

A confusion matrix, or an error matrix (Stehman, 1997), is a table which can be used to identify which labels are often mislabeled by the model, or "confused", in a visual manner.

In the confusion matrix in Figure 2.1, 5 cats were predicted to be cats, while 3 cats were predicted to be dogs. 1 dog was predicted to be a cat, while 4 dogs were correctly predicted to be dogs. In total, there were 8 cats and 5 dogs. From this matrix, it can seem that our classifier does quite well at recognizing dogs, but struggles to determine whether a cat is a dog or a cat.

		Actual class	
		Cat	Dog
Predicted	Cat	5	1
	Dog	3	4

Table 2.1.: Example of a confusion matrix.

2.2.7. Tools and Frameworks for Machine Learning

Machine learning involves computationally heavy tasks, with the use of mathematical and statistical models which can be separated from the user interface. This is done by using frameworks, which to a larger or smaller extent, simplify the task of setting up a network or defining training procedures. Many such frameworks exist, and this subsection presents a few.

PyTorch

PyTorch is a machine learning framework based on the framework Torch (Ketkar, 2017a), with a Python interface (C++ also available). Tensor computing, which is a large part of the work done in an ANN, can be accelerated by performing calculation on GPU devices. PyTorch provides a very easy-to-use interface to enable such acceleration. Another feature is its use of automatic differentiation (Paszke et al., 2017), meaning that operations performed are recorded, and subsequently replayed backwards to compute weights. This is a powerful and time-saving feature in neural networks, as differentiation of the parameters is a computation-intensive task performed repeatedly in training.

TensorFlow

TensorFlow is a machine learning framework for numerical computation, built by Google (Abadi et al., 2016). TensorFlow uses static data flow graphs for computations, and is slightly nearer to the machine than PyTorch, by features such as sessions and placeholders which more directly allocate resources to computation. It provides functions for several different layers of mathematical abstraction, where the highest level of abstraction is the Keras API.

Keras

Keras is a machine learning API with a high level of abstraction of tasks, building directly on TensorFlow, with a Python interface (Ketkar, 2017b). Its main feature is its low entry level, with many high-level built-in functions requiring little knowledge for anyone to experiment. However, TensorFlow features can also be utilized directly.

Scikit-learn

Scikit-learn is a machine learning and data analysis framework for Python (Pedregosa et al., 2011). It provides built-in tools for data processing, model selection and many more detailed features. Functions from Scikit-learn can be used in combination with other machine learning frameworks, or combined to create models directly, building directly on scientific computing packages such as NumPy, SciPy and Matplotlib.

2.2.8. Musical Composition

In music composition today, digital tools are virtually inevitable. Many instruments are fundamentally digital, such as digital synthesizers, where the audio output can be manipulated entirely. Many other acoustical instruments have digital counterparts, such as electrical drum kits, where sound is produced digitally based on physical input.

In order to record, process and produce music digitally, a Digital Audio Workstation (DAW) is used (Leider, 2004). Typically, many different audio tracks are combined in the DAW and edited to fit together. Waveforms can be manipulated directly, or a variety of tools can be used to achieve the desired sound.

Generative Modeling

In machine learning, two main approaches are discriminative (such as SVM or perceptrons) and generative models (Jebara, 2012). Generative models work on joint probability distributions on an observable variable X and a target variable Y. The generative model views the conditional probability of the observable variable X, given the target variable Y, $P(X|Y = y)$.

This means that the generative model can "generate" new instances of X in relation to the target variable Y (Jebara, 2012). This is exploited in architectures such as the Generative Adversarial Network (GAN), where instances of output variables are generated in a way which has no apparent relationship to probability distributions over potential samples of input variables. The generative model learns from mapping a *latent space*, i.e. variables that are not directly observed, but rather inferred from other observed variables, to a desired data distribution, e.g. an input dataset.

Another generative model is the Variational Autoencoder (VAE) (An and Cho, 2015; Khobahi and Soltanalian, 2019). An autoencoder is in fact two connected networks, an encoder and a decoder. The encoder network takes an input sample, encoding it to a smaller, dense representation. The decoder network followingly takes the representation, decoding it back to the original input. The encoder and decoder are trained in pairs, being evaluated on the ability to reconstruct the original input flawlessly. This teaches the decoder to keep important information and discard less important information.

The variational autoencoder uses continuous latent spaces to create outputs that are different from the original input. The VAE uses the means and standard deviations of the input, along with random sampling in the continuous latent space, in order to generate results that interpolate between different classes of inputs.

Generative models can be trained by discriminative models, where the generative model attempts to create new input that appears indistinguishable from the remaining possible inputs, while the discriminative model evaluates the attempts (Kingma and Welling, 2019).

Generative models are used for many purposes today, such as generating images or audio based on existing data. Related work for this is presented in Section 3.3.5. This is the basis for the different music composition methods addressed in this thesis.

2.2.9. File Formats

For digital music composition and editing, many different file formats exist. Most formats are used to encode audio data, where all instruments in an ensemble are stored on the same file. These formats can be uncompressed, with the most common format being WAV (Pan, 1993). These retain high quality, but naturally come with a large file size. Due to this, compressed file formats are commonly used, with some quality being compromised towards smaller file sizes. MP3 is an example of a file format using "lossy" compression.

The Musical Instrument Digital Interface (MIDI) is an industry standard defining notes in a musical system. It does not store audio data directly, but rather the information needed to play the audio. The MIDI format is widely used in all digital music composition, because information can be transmitted between instruments and music composition software, preserving all information (Cataltepe et al., 2007). Many different instruments can play in the same MIDI file, and they are entirely distinguishable, making it suitable for editing in a DAW.

3. Related Work

In exploring the state-of-the-art within musical computational creativity, a Structured literature review (SLR) was conducted. This chapter documents the research method applied and the following results. Section 3.1 introduces the methods applied in researching related work. Section 3.2 presents the conduction of the SLR method. Section 3.3 presents the results of the literature review, with subsections ordered by topic relevant to the different research questions.

3.1. Introduction

The SLR method is applied to ensure rigour and thoroughness in researching literature in a given field. The main motivations for applying such a rigorous method within the literature review are assisting in identifying existing solutions, helping identify bias and avoiding duplicating efforts made elsewhere in the academic community. It also helps the reader reproduce the steps taken to discover the exact set of papers used in this thesis.

The concrete process of conducting the SLR performed in this thesis is based on the guide from Kofod-Petersen (2018). However, not all articles used in this thesis are discovered by the SLR method, but rather they were provided by the project supervisor and others at the IDI institute. This set of papers will be described as the Starting set. There are also a number of other articles discovered by the "snowballing" method, i.e. starting with a given set of papers and following their listed sources to learn more on a topic. This method, as presented by Wohlin (2014), is not opposing to the SLR method, but rather one of many tools used to perform a systematic literature review while mitigating the drawbacks of using only one search string which does not guarantee finding all relevant articles within a topic. The complete review protocol is found in Appendix A.

The SLR was conducted as an introductory method for the exploration of RQ1, meaning that the search revolved around the emotion classification task. For related work regarding RQ2 and RQ3, such a rigorous process was not documented. For RQ2, articles were mainly provided by the project supervisor, and the snowballing method proved very useful in exploring different and opposing opinions. Moreover, many articles on emotion classification explore this subject themselves. For RQ3, articles were also mainly provided by the project supervisor. Also, a suiting search string could not be determined, as many existing music composition tools use a product name, making it challenging to develop a search discovering a significant amount of the progress made. Thus, the snowballing method was also applied for this research question.

3.2. Review Method

In this section, the process of the Conducting phase of the SLR review protocol, described by Kofod-Petersen (2018), is presented. The Planning and Reporting phases can be seen in Appendix A.

1. Identification of research.

A set of search terms is defined to correspond to the research questions (see Section 1.2). The development process for the search terms can be seen in Appendix A.

The search string used was:
`("Music" OR "Musical") AND ("Mood" OR "Ambiance" OR "Emotion") AND ("Classification" OR "Detection") AND ("Artificial Intelligence" OR "Machine Learning")`.

Google Scholar was chosen as the preferred search engine. Google Scholar aggregates results from several sources, many corresponding to recommended reading from the project supervisor.

The search was performed with a temporal limit of only viewing articles written in 2016 or later. This choice was made to reduce the number of resulting articles and prevent outdated results. However, articles coming from the Starting set or any snowballing "children" were not limited by publishing date, as a way to explore some of the fundamental milestones in the field's years of existence.

The search resulted in a staggering 16,000 results, indicating the variety of work already done within this field. As this amount was too large for a scope such as a Master's Thesis, the screening process used the first 50 articles from each search ranked "most relevant" by Google Scholar, ignoring duplicates within the search. The requirements for the "most relevant" ranking is not clearly presented, but some indicators are the number of citations and the search terms' frequent use within an article. Many of the displayed articles were non-technical but rather within fields such as psychology or medicine, so a large number was filtered out.

2. Selection of primary studies.

A set of primary and secondary inclusion criteria is formulated. The primary criteria focused on the studies' title and abstract and their relations to the goals of this thesis, ensuring the main concern is musical computational creativity, its relation to mood or emotion, and presenting empirical results.

The studies passing these criteria were evaluated using the secondary inclusion criteria, evaluating the entire text on aspects such as disregarding lyrics processing, and that the study should discuss the implementation of an application. The criteria definitions are documented in Appendix A. In the starting set, 15 articles passed primary inclusion criteria and 12 the secondary inclusion criteria. In the SLR set, 22 articles passed primary inclusion criteria and 12 the secondary inclusion criteria.

3. Study quality assessment.

When primary studies are selected, a set of inclusion criteria developed by Kofod-Petersen (2018) were used to assess the quality of each paper. For each criterion, each paper is ranked by whether it met the criteria; Yes (1 point), Partly (0.5 points) or No (0

points). The papers with a sum of less than 6 points were considered of not sufficiently high quality, leaving a total of 20 relevant papers for data extraction, monitoring and synthesis.

4. Data extraction.

In order to ensure extraction of similar types of information from all studies, a set of data points was defined for manual extraction from each study. The data from all articles are then assembled in a table, with each data point in one column, and data points for one study within one row. The data points used for this extraction are documented in Appendix A.

5. Data synthesis.

From the extracted data, different approaches and solutions are compared. Another point of study is what different approaches use as performance markers, in order to ensure that this project uses some performance indicator relevant for comparison and recognised in the community.

3.3. Results

This section presents the results of the performed literature review. Section 3.3.1 presents the selected studies, and an overview of their main findings, while the remaining sections present results sorted by topic. Even though the search focused on emotion classification, many articles touched on topics relevant to all research questions.

Section 3.3.2 presents the results for methods of emotion classification, addressing RQ2, and establishes a vocabulary used in the following sections. Section 3.3.3 presents how different architectures represent musical data. Section 3.3.4 presents different architectures used for emotion classification, directly related to RQ1 and used as a foundation for the architecture presented in Chapter 4. Section 3.3.5 presents different architectures used for music composition, including some commercial products that, while showing impressive results, unfortunately are not transparent with regards to the system architecture. Section 3.3.6 describes and compares a variety of musical datasets used in training, laying the foundation for selecting a dataset for training in Chapter 4.

3.3.1. Selected Studies

The selected studies which met the Quality Assessment (QA) criteria cover a range of different topics. Most studies found in the SLR search present architectures used for the MER task.

The selected studies meeting the QA criteria are listed in a table format. The format is in accordance with the data points listed for data extraction in Appendix A. The selected articles are presented in two tables; Table 3.1 for the starting set of articles and Table 3.2 for articles from the SLR. The QA accumulated score is also listed. Studies that did not meet the required QA score are omitted, explaining some “holes” in the ID numbering.

The studies selected in this literature review are presented and grouped by topic in the following subsections.

ID	Authors	Title	Year	Algorithm	Dataset	Findings and conclusions	QA
SS1	Li, Ogihara	Detecting Emotion in Music	2003	Multi-label SVM	499 30-sec segments	50% precision, many borderline cases	7
SS2	Yang, Liu, Chen	Music emotion classification: A fuzzy approach	2006	Fuzzy nearest-mean classifier	243 25-sec segments	78% precision, track variation throughout a song	7
SS3	Roberts, Engel, Raffel, Hawthorne, Eck	A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music	2018	VAE	1.5M MIDI files; Lakh MIDI Dataset	Successful long-term structure in music	9
SS4	Boenn, Brain, de Vos	Computational Music Theory	2012	ANTON: Answer-Set Programming	Musical rules	Successful with local structure	7
SS5	Bozhanov	Computoser – rule-based, probability-driven algorithmic music composition	2014	Rule-based/probability-based hybrid	500+ pieces	Loosely defined musical rules by composers give good results. Manual feedback	7.5
SS7	Huang, Vaswani, Uszkoreit, Shazeer, Simon, Hawthorne, Dai, Hoffman, Dinculescu, Eck	Music Transformer: Generating Music with Long-Term Structure	2018	Autoregressive model	MAESTRO Dataset	Self-attention and relative timing are crucial factors to long-term coherence	9
SS9	Engel, Hoffman, Roberts	Latent Constraints: Learning to Generate Conditionally from Unconditional Generative Models	2018	VAE, compares to GAN	Audio and image samples	Can do conditional sampling from an unconditional model, i.e. eliminating need to re-train for a query	8.5

ID	Authors	Title	Year	Algorithm	Dataset	Findings and conclusions	QA
SS10	Fox, Khan	Artificial Intelligence Approaches to Music Composition	2013	MAGMA: Markov chains + routine planning + genetic algorithms	MIDI pop songs	Stochastic algorithms do not follow music theory, sounds "overly random" at times. Imposed verse/chorus structure can improve on this	6
SS11	Freitas, Guimarães	Melody Harmonization in Evolutionary Music Using Multiobjective Genetic Algorithms	2011	Genetic algorithm	Single melody	Highly explicit musical rules and preferences, provide many feasible suggestions. Two styles: Simplicity and dissonance	7
SS12	Roberts, Engel, Mann, Gillick, Kayacik, Nørly, Dinculescu, Radebaugh, Hawthorne, Eck	Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live	2019	VAE	MIDI files or user input	User-ready plugins to utilize in music software to enable creativity	8.5

Table 3.1.: Data extraction and QA score, Snowballing Starting Set of articles.

ID	Authors	Title	Year	Algorithm	Dataset	Findings and conclusions	QA
S6	Chen, Zhao, Xin, Qiang, Zhang, Li	A Scheme of MIDI Music Emotion Classification Based on Fuzzy Theme Extraction and Neural Network	2016	Fuzzy pattern matching, supervised ANN	180 MIDI pieces	78% precision, genre-independent. Can find "theme" of a song	6
S9	Seo, Huh	Automatic Emotion-Based Music Classification for Supporting Intelligent IoT Applications	2019	SVM	MP3 files	77% precision. Sound quality has large impact. Good survey and experiment structure	7.5
S10	Bai et al	Music emotions recognition by cognitive classification methodologies	2017	Compares SVM, KNN, NFNC, FKNN, Bayes, LDA	MediaEval	SVM, FKNN and LDA perform well. Good feature extraction	7
S16	Liu, Chen, Wu, Liu, Liu	CNN based music emotion classification	2017	Deep Convolutional neural network	CAL500, CAL500exp	Uses only audio spectrogram	8.5
S21	Lin, Liu, Hsiung, Jhang	Music emotion recognition based on two-level support vector classification	2016	SVM	300 songs	Two level classification: Genre and mood. Attribute estimators by ReliefF	6
S23	Mo, Niu	A Novel Method Based on OMPGW Method for Feature Extraction in Automatic Music Mood Classification	2017	OMPGW, SVM, BLSTM-RNN	Soundtracks, MIREX-T, MTV, MediaEval 2015	Thorough feature extraction is useful for higher resolution and accuracy	8.5
S24	Zhang, Meng, Li	Emotion extraction and recognition from music	2016	Random forest classifier, decision tree	APM database	Stereo-level feature analysis, adding EEG data slightly improves accuracy	8

ID	Authors	Title	Year	Algorithm	Dataset	Findings and conclusions	QA
S27	Rosli, Rajaee, Bong	Non Negative Matrix Factorization for Music Emotion Classification	2016	Non-negative matrix factorization (NMF), ANN	500 samples	Separate music into instrumental and vocal components: Vocal timbre is more effective for distinguishing emotion	6.5
S28	Kartikay, Ganesan, Ladwani	Classification of Music into moods using musical features	2016	Compares Naïve Bayes, LDA, decision trees, multi-class SVM	1000 samples	FMA Linear classification, bad results with SVM.	6
S30	Patel, Chouhan, Niyogi	Using Crowd Sourced Data for Music Mood Classification	2018	ANN, SVM, decision trees	16527 songs	ANN outperforms SVM and decision trees. Using crowd-sourced labels reduces bias.	8.5
S46	Shahmansouri, Zhang	An empirical study on mood classification in music through computational approaches	2016	Compares Bayes, Multilayer Perceptron, Decision Tree and more	Million Song Dataset, Last.FM dataset	Current classification algorithms are lacking in performance, very precise feature extraction is needed	7.5

Table 3.2.: Data extraction and QA score, SLR Articles.

3.3.2. Emotion Categorization

Music Emotion Recognition (MER), or the understanding of mood or emotion in general, can intuitively be perceived as a highly subjective topic. Sources of contextual variation are plentiful, such as cultural or national belonging, personality, musical preferences or even time of day or the weather outside. Even though there are many ways to extract features from music and assigning them emotional meaning, the results will surely be victim to some bias. The reason for this is that in one way or another, a human notion of emotion or mood will be inserted into the system. This could be in, e.g., the naming of features, the labelling procedure, or in evaluating results. This subsection presents some efforts to model emotion in comprehensive ways.

In the psychology of emotion, an important issue is understanding whether and how emotions can be considered a universal concept. Ekman (2016) presents a view originating from Darwin, namely that emotions are indeed discrete and distinguishable, as well as being a universal human trait. This was explored by studying remote tribes excluded from the rest of the world – their smile represented happiness just as much as one's own smile does. Ekman, Darwin and many others thus view emotions more or less as a universal concept, and that the emotions' triggers and expressions are universal as well.

To further explore subgroups of emotions, Ekman presents the *Atlas of Emotions*. The atlas presents Ekman's hypothesized main emotion categories, namely enjoyment, sadness, anger, disgust and fear. Some of the emotions share some overlap, such as anger and disgust, or disgust and fear.

Within each emotion, many subgroups are presented. One example is the feeling of enjoyment. The subgroups are ordered by intensity, where enjoyment such as sensory pleasure is the least intense, and excitement and ecstasy are the most intense. The atlas in a visual manner can be found with the Paul Ekman Institute.[1]

In relation to machine learning, this system can seem to fit into a system of *fuzzy logic*. That is, categories of emotions can overlap, and a human can feel more than one feeling at once. In the classification problem, this can mean that one music sample can be classified into more than one class, and that estimated classifications can be considered partially correct.

Thayer (1990) presents the *Model of Mood*, another approach to mapping emotions, here in a two-dimensional plane. The two axes are those of *valence* and *arousal*, i.e., the positive or negative nature of the emotion, and its energy or intensity, as seen in Figure 3.2 (Yang et al., 2006). The two-dimensional plane has been used in music classification studies such as by Bai et al. (2017), Kartikay et al. (2016) and Mo and Niu (2019). The plane can also be viewed as a set of four *quadrants*, for each corner of the plane, as used by Panda et al. (2018).

Palmer et al. (2013), and later Whiteford et al. (2018), suggest that for music, people mostly agree on the understanding of what the music is trying to express. This is found through experiments of colour, by having survey participants assign a colour to music samples. The labelling from Whiteford et al. (2018) can be seen in Figure 3.1. Colour

[1] `http://atlasofemotions.org`

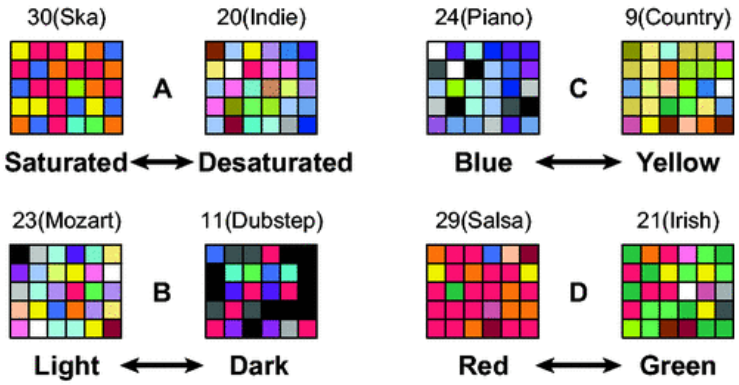

Figure 3.1.: Survey participants' colour assignment by genre. Reproduced with permission.

associations were also studied in relation to underlying musical features such as loudness, harmony or distortion. Whiteford et al. (2018) argue that the measure of Valence and Arousal (VA) suffice to express the emotions in the music, and also that the music-colour associations have a strong correlation with related emotions, e.g. faster music in the major mode would produce more saturated, lighter colours. In contrast, slower music in the minor mode would produce darker, bluer colour choices.

One advantage of using Thayer's model of mood is that the dimension of energy or arousal is quite simple to estimate by amplitude measures. Liu et al. (2003) exploit this advantage, and present a hierarchical framework for understanding acoustic data, arguing that a hierarchical system is required to reduce ambiguity in relevant categories. In the first level of the hierarchy, energy levels are distinguished into high or low. The second level distinguishes high and low valence, in total producing four emotion categories.

If the amplitude and therefore, energy, is low, the data is classified into group 1 (Contentment *or* Depression). Then, other features categorize the sample into one of the two subcategories. An advantage in favour of the hierarchical approach is that results more often can be achieved with sparse data sets (McCallum et al., 1998). A significant merit of this model is that the X/Y axes are simple to express on a computer, and all emotions fit into this continuous plane. However, in some cases, it may be an over-simplified model. For example, viewing the model as four emotional quadrants, anxiousness and anger would be placed in the same quadrant (high arousal, low valence). Clearly, the two emotions are very different. In conclusion, the VA plane and the quadrant system serve a very useful purpose, especially in the context of computer-based understanding, but should not be used without considering the dangers of over-generalization. Due to this taxonomy's ease of expression in a digital sense, and its ability to view emotions as concrete points in a two-dimensional space, this taxonomy is used in the architecture presented in Chapter 4.

One natural feature of emotion classification is that music can fit within several labels (e.g., a song can be both happy and relaxed, or both happy and upbeat), as seen in

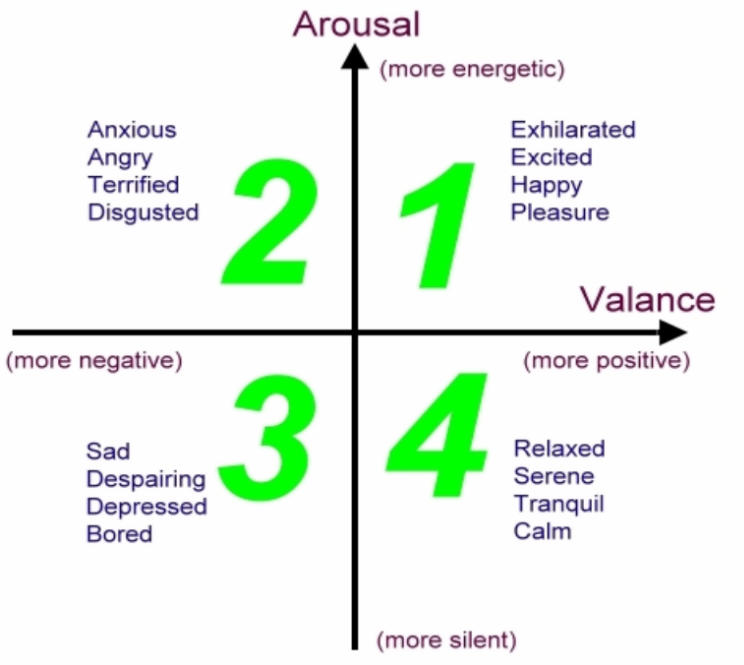

Figure 3.2.: Valence-Arousal (VA) plane. Reproduced with permission.

the Atlas of Emotion (Ekman, 2016). As such, mood and emotion classification may be perceived as a multi-label classification problem. Li and Ogihara (2003) worked on music in such a regard, treating the multi-label classification problem as a set of binary classification problems using Support Vector Machines. Categories of emotion originated from Farnsworth (1958), with some additions, see Table 3.3.

For music, Cowen et al. (2020) argue that music makes us feel 13 distinct emotions, with a large survey with participants from the United States and China. Participants reported both on specific emotions (e.g., "angry" or "dreamy") and in the valence-arousal plane. The results show that specific emotions are better preserved across the two cultures than levels of valence and arousal. Cowen et al. state: "People from different cultures can agree that a song is angry but can differ on whether that feeling is positive or negative". The results converged on 13 distinct emotions, as visualized in Figure 3.3.[2]

3.3.3. Digital Music Representation

There are many ways to "listen" to music, and which methods we choose can affect what we understand from the music. In computing, Celma et al. (2006) describe the *Music Semantic Gap*, i.e. the problem of understanding both the low-level audio signals and the higher-level features of music. From audio, the computer can understand signal features such as loudness, contrasts and pitch, as well as "content objects" such as

[2] `https://www.ocf.berkeley.edu/~acowen/music.html`

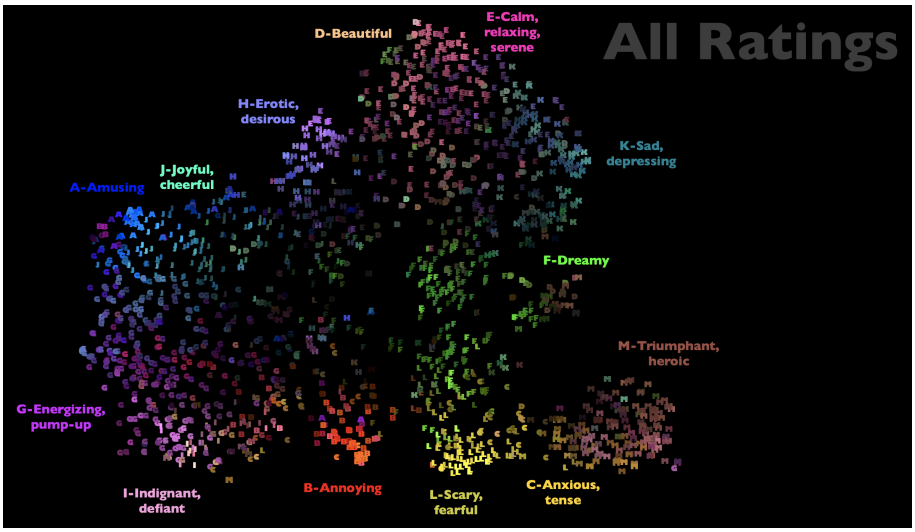

Figure 3.3.: Cowen's map of musical emotions. Reproduced with permission.

A: cheerful,gay,happy	H: dramatic, empathic
B: fanciful,light	I: agitated, exciting
C: delicate, graceful	J: frustrated
D: dreamy,leisurely	K mysterious,spooky
E: longing, pathetic	L: passionate
F: dark,depressing	M: bluesy
G: sacred, spiritual	

Table 3.3.: Adjective groups (Li and Ogihara, 2003) in describing musical emotion.

harmony, rhythm and even genre. The "semantic gap" is the road from there to the human understanding of music, relating to individual emotions, opinions and memories. Machine learning is described as one fundamental way to bridging the gap, combined with many other elements such as text understanding, music theory and computational neuroscience.

In the discovered related work, music is represented in a large variety of ways. Some use only raw, acoustic data (formats such as MP3 or WAV). This has been presented as a suitable method for deep convolutional neural networks, as a way to analyze higher-level musical features, as by Dai et al. (2017) within the recognition of urban sounds. Others use the MIDI file format to access musical content from various instruments more accurately, although this may limit the amount of available data as music seldom is released in such a format. Music can also be mapped visually using spectrograms, as with Liu et al. (2017) and seen in Figure 3.4. No single optimal method for music representation is established in the reviewed literature. However, there is often a number of explicit representations of musical features such as rhythm, mode, tonality and dynamics, extracted in advance with dedicated functions for each type of feature, on which an algorithm trains explicitly.

3.3.4. Emotion Classification Algorithms

Fuzzy Logic Classifiers

As music can often express a variety of emotions, there is a natural "borderline" nature to the mood classification problem. Yang et al. (2006) addressed the issue using *fuzzy logic* classifiers, utilizing Thayer's model of Mood as a two-dimensional emotion space (2DES), to classify the mood of waltzes into clusters. With such an approach, one can not only indicate belonging to more than one emotional category; the strength of each emotion can also be indicated. Bai et al. (2017) use the Fuzzy K Nearest Neighbour (KNN) method with accuracy as high as 83% with a similar 2DES and Gaussian function as the fuzzy function. Chen et al. (2016) use fuzzy pattern matching from musical temporal features to extract a theme from a song, and an Artificial neural network (ANN) to identify emotion within the music.

In the composition of music based on emotions, the indication of several measures of mood could be to an advantage, rather than having only one classification for each music sample. This way, musical data may be used for music in different emotions, if the fuzzy score is high for more than one emotion.

Evolutionary methods

As a way of optimizing for some novel optimization task where both the experimental and the familiar are important, evolutionary or genetic algorithms can be a suiting choice. Freitas and Guimarães (2011a) use genetic algorithms to find a melody harmonization, given an input melody. Genetic operators are concrete musical operations, such as pitch mutation, musical crossover and measure swapping. The evaluation is performed by comparing experimental results to human judgment, and a small tendency was found

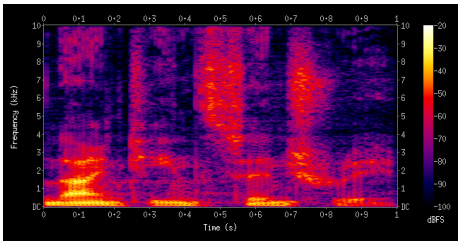

Figure 3.4.: Spectrogram of a male voice saying “nineteenth century”.

between some evaluation measures and the human input.

Support Vector Machines (SVM)

Seo and Huh (2019) apply the SVM architecture to create areas in a vector plane used for the classification of emotions in music. The evaluation was performed towards survey participants. However, for results to represent the individual variations, classification match rates were divided into “All”, “Most” and “Least” cases, where “All” cases were songs where all of the survey participants agreed on the classification. For “Most” cases, the classification hit the most popular response from (internally divided) participants, and for “Least”, at least one participant had given the response for one emotion. Overall, precision reached 73.96%, outperforming deep neural network, random forest and K-nearest neighbour classifiers.

Lin et al. (2016) use acoustic features of sound directly with a two-level SVM classification system, analyzing both music genre and music features. The algorithm uses automatic tools for feature weighting, so that the weighting impact both analysis level and their likelihood for different classifications.

Artificial Neural Networks (ANN)

In terms of music, ANNs (and many other approaches) have the disadvantage that the input data must be uniform, which music is not in the sense that it varies in length. Liu et al. (2017) propose a deep Convolutional Neural Network (CNN) on music spectrograms, transforming music into a “heat map” of the audio signal in the axes of time and intensity. See an example of a spectrogram in Figure 3.4. This approach avoids much of the complex feature extraction which is often needed with music, and the convolution can operate on different temporal locations as one would in a song to classify emotion.

Patel et al. (2018) propose an ANN where a song’s metadata come from crowd sourced annotations from the music website Last.fm[3] as well as the online music streaming service Spotify. [4] Here, music is “tagged” by users on measures such as valence, instrumentalness, danceability, mood and much more. The algorithm fetches “top tracks” from the Last.FM API for mood terms, and then fetches track metadata from Spotify resources. As such,

[3] `https://last.fm`
[4] `https://spotify.com`

a wide range of metadata tags can be of help in the classification task, but should also be noted as a possible source of bias in the unregulated nature of crowd sourced contributions. The algorithm employs a Multi-Layer Perceptron with backpropagation, with a sigmoid activation function for all layers, reaching an F-measure of 86%.

Rosli et al. (2016) utilize non-negative matrix factorization in an effort to distinguish vocals from instruments in music. The results indicate that voice is a better indicator of mood than instruments, but voice and instruments combined within the classification proves even more useful.

Hybrid Systems

Some hybrid systems have the goal of comparing performance between different approaches. Kartikay et al. (2016) compares Support Vector Machines, Naïve Bayes, Linear Discriminant Analysis and Decision Trees, rating LDA as the overall must accurate classifier. However, their results display that accuracy varies for different classes; the SVM implementation has a 87.5% classification accuracy for happy songs, and only 22.6% for peaceful songs. There can be many reasons for this, such as the method for feature extraction, or how the classes were initially defined from song metadata.

3.3.5. Music Composition Systems

Generative Modeling

In order to effectively label and model data, and retrieving from the trained model, Engel et al. (2018) propose the learning of latent constraints. On a trained model, latent constraints are value functions which identify areas in the trained latent space with desired attributes. The advantage of this is that an *unconditionally* trained model does not need re-training when performing *conditional* sampling, such as generating music in accordance with some user-specified input. A Variational Autoencoder (VAE) is trained, with a focus on being able to reconstruct its encoded and decoded input. An actor-critic pair, similarly to the role of the encoder and decoder, is used to discriminate between encodings of actual data, latent vectors and transformed samples. A "realism" constraint is imposed to address the trade-off between the quality of reconstructed vectors and sample quality. The actor-critic pair in latent space can generate samples that satisfy user-specified constraints with a high reconstruction quality.

A known weakness of many compositional systems is the lack of long-term coherence and structure in the produced music. Roberts et al. (2018) propose the aforementioned Latent Vector Model for the issue of long-term structure. Huang et al. (2019) present the Music Transformer, and suggest that temporal self-attention could be an essential aspect of achieving the essential quality of great timing. An approach to representing relative positional information is presented, as opposed to previous solutions using absolute or pairwise distance, as well as quadratically increasing sequence length. Using an autoregressive generative model, self-attention is incorporated into each layer before a feed-forward sublayer. The both local and global attention of the algorithm allows for

creating continuations with repeated motifs and variations on a given input, or merely the existing model.

The Music Transformer original source code, while seeming a suitable architecture for the purpose of automated music composition, is not fully available. The authors have released a platform for working with the project as a Colab Notebook.[5] However, on this platform, some "model checkpoints and auxiliary data" are not included with the code, only copied from Google Storage. As this data is not available for extraction, the system is difficult to verify and reproduce. However, while not a scalable solution, the Colab Notebook accepts one MIDI file as melody conditioning, which could serve the simple purpose of the composition of music based on *one* sample annotated with mood.

Several efforts have been made to reproduce the Music Transformer in reproducibility challenges. Koh et al. (2019) implement the proposed memory-efficient relative attention transformer. However, results could not be reproduced, namely in the form of a higher loss, as well as the evident lack of long term structure in produced results. The proposed architecture varied only from the original in that PyTorch was used as a framework, rather than Google Brain's own Tensor2Tensor. The results could be verified as complete code is available online.[6] As of yet, no other viable, complete, and freely available implementation producing similar results has been discovered throughout this project.

Building on the work of Huang et al. (2019), Huang and Yang (2020) present the Pop Music Transformer. This work perceives musical scores as sequences of Revamped MIDI-Derived Events (REMI), providing a more explicit metric structure for rhythmic and harmonic structure. Also, the model facilitates several instruments, such as the piano, bass and drums. The goal is both to compose music based on both conditioned and unconditioned models, and to represent music in a more similar fashion to how music is read and understood by humans. Furthermore, the Pop Music Transformer can fine-tune an existing model on a particular subset of data, making it highly suitable for the mood-adjustment purposes desired in this thesis.

Another multi-instrument composer, also building on the Transformer architecture, is the LakhNES, presented by Donahue et al. (2019), which composes music based on 8-bit chiptune music designed for the Nintendo Entertainment System (NES). The model trains on both the Lakh MIDI dataset (also used by Engel et al., 2018) and the NES four-instrument dataset (Donahue et al., 2018), by adapting the samples from the Lakh MIDI Dataset to being "performed" by the available instruments from the NES. Although the two datasets differ in content and genre, transfer learning between the two improves results. Benefits are drawn from the Lakh MIDI Dataset's large size (175,000 samples), and the NES Music Database's structural homogeneity.

Hybrid Systems

Hybrid systems can be used as an attempt to mitigate disadvantages from one or more systems by combining their features. One such project is the Computoser, presented by

[5] https://magenta.tensorflow.org/piano-transformer
[6] https://github.com/COMP6248-Reproducability-Challenge/music-transformer-comp6248

Bozhanov (2014), a hybrid between a rule-based and probability-based algorithm. The algorithm analyses both defined rules on terms such as structure, rhythm and repetition as well as sample data, producing new songs. The evaluation is also in a hybrid form, by machine detection of discrepancies from rules, and also by an audience as the system's output is released online.[7]

Fox and Khan (2013) introduce the Multi-AlGorithmic Music Arranger MAGMA, a knowledge-based system using Markov chains, routine planning, and genetic algorithms to compose music. A user inputs desired values for five different preferences: Transition, repetition, variety, range and mood. First, the song structure is generated (i.e., verses, choruses, bridges and outros). Then, the measure structure is determined, followed by generating a chord sequence and at last, the melody. Each of the three algorithms used performs all four steps and produces a result for the user to compare.

Commercial products

In the development of systems related to AI and music, much progress has been made outside of the strictly academic context. Systems for composition, adjustment and recommendation can rapidly become commercial products. One significant contribution is AIVA, or the Artificial Intelligence Virtual Artist, composing soundtracks based on emotion, genre or other parameters (AIVA, 2019). The architecture is only explained briefly, but consists of a combination of genetic algorithms and deep neural networks (AIVA Technologies, 2018). This has resulted in several products, one of them being a Music Engine, composing to a variety of genres and moods, with the stated goal to assist human composers in "the cases where human creativity doesn't scale". This could be relevant for use in movies, video games, or other areas where the need for musical content is large, but where the demand for variation and originality is also present. AIVA has been registered as a composer in an author's rights society, SACEM, making it the first software recognized for its creative capabilities (AIVA, 2019).

Other commercial products also exist in the sphere of AI music composition. Melodrive[8] is a company which has created Melodrive Indie, which composes an infinite stream of music adjusting to user input, primarily intended for video games (Melodrive, 2019). Another is Popgun [9] where a tool has been created for automatic mastering of musical tracks to improve audio quality in addition to working with musical composition. However, very little is published about the architecture.

3.3.6. Musical Datasets

In order to train a model for use with music, a suited dataset of sufficient size and quality is necessary. However, the access or construction of a suitable dataset can be difficult, and even more so with music due to copyright and usage restrictions. In this section, some dataset options are presented, and the general conclusion can be seen in that there

[7] `http://computoser.com`. Accessed November 9, 2019
[8] `https://melodrive.com`. Accessed November 12, 2019
[9] `https://popgun.ai`. Accessed November 12, 2019

often is a clear trade-off between dataset size, data availability and data or annotation quality. As a large amount of data is paramount to the successful usage of neural network and deep learning methods, the demands for a suitable dataset are high.

The 2007-2017 MIREX Audio Mood Classification tasks annually compare MER algorithms with large popular music datasets. However, due to proprietary restrictions, datasets are not directly accessible to participants.

Panda et al. (2018) present a dataset extracted from the openly available AllMusic API[10], querying songs for their title, artist, genre and emotion "tags". AllMusic's existing tags do not spring from any known taxonomy, but the dataset maps them as accurately as possible onto a VA plane. For the emotion recognition task, SVM is used. This dataset provided music information directly, as well as emotion tags corresponding to the valence/arousal taxonomy presented by Thayer (1990). Thus, this was a fitting choice for the network presented in Chapter 4.

Ferreira and Whitehead (2019) present the VGMIDI dataset, consisting of 95 labelled and 728 non-labelled piano pieces stemming from video game soundtracks. The dataset is annotated in the two-dimensional VA plane. Due to its limited size, the dataset is primarily designed for training a generative model composing new music.

Soleymani et al. (2017) present The Database for Emotional Analysis of Music (DEAM) dataset, consisting of 1802 royalty-free 45-second song excerpts. Although this dataset excels in size and in the fact that audio samples are available, data quality is not consistent. In some excerpts, noise, claps or silence are highly present. This is due to the excerpts being chosen from a random point in the song, and makes its results difficult to trust and use for machine learning. Furthermore, other researches have performed further validation on the annotation methods, and there is a significant discrepancy between the annotations from the original subjects and the final quadrant annotation, only agreeing on 47% of samples (Vale, 2017). Also, the dataset is quite unbalanced, with as much as 681 samples for Q3 and only 200 for Q2. Thus, the dataset is not considered to be of sufficient quality for use in its current state.

Turnbull et al. (2007) present the Computer Audition Lab 500 (CAL500) dataset. The dataset consists of 500 popular music songs, each labelled with multiple adjectives by at least three subjects. There are 174 labels, categorized within emotion, genre, instrument, song, usage and vocal, where 18 of the labels are related to emotion. The labelling process was performed by 66 paid students in a controlled laboratory environment, under "controlled experimental conditions", and should, therefore, be of sufficient quality. The original full dataset is unavailable. However, the features are available online [11].

Donahue et al. (2018) present the NES Music Database, a corpus of 46 hours of video game soundtracks created for the Nintendo Entertainment System (NES). As the console was released in 1983, music was limited to its 8-bit architecture and four monophonic instruments. As the music is composed for video games over a limited time, it is remarkably stylistically cohesive, making it suitable for training in a machine

[10] `http://developer.rovicorp.com/docs`. Accessed November 10, 2019
[11] `https://github.com/yzhaobk/CAL500`

learning context. The dataset is openly available online[12].

One effort to enable MIR research for machine learning in terms of dataset size is the Million Song Dataset. It is a very large and freely available dataset consisting of audio features and metadata for one million popular music tracks (Bertin-Mahieux et al., 2011). Its data is provided by The Echo Nest [13], a music intelligence platform owned by Spotify. No audio data is available directly, but rather features extracted from the music. These features are both baseline features such as tempo, key, loudness, as well as community-based tags such as crowd-sourced Last.FM metadata. The quality of the metadata is not guaranteed, and therefore this cannot be considered a valid source of information without manual validation.

Spotify Developers Platform Another service, which is not quite a dataset but can provide a significant amount of data, is the Spotify Developers platform. For any song in the Spotify roster, information can be fetched on various attributes. Some baseline attributes include song duration, key, mode and tempo, while more high-level attributes include acousticness, danceability, loudness, speechiness and valence. Spotify also provides their own recommendation API, where a song, artist or genre is entered as a "seed", and a number of other tracks are returned based on the seed. In theory, the information on loudness and valence could be combined to produce an emotion quadrant for each song. However, how these attributes are determined on Spotify's side is nontransparent, and therefore is difficult to trust and build upon. With Spotify's APIs, music can, to some extent, be accessed, but is limited to registered users, and often paying users. Also, downloading samples is not available, but the music can be played and used within Spotify's own web interface.

[12] `https://github.com/chrisdonahue/nesmdb`
[13] `http://the.echonest.com`

4. System Architecture

As part of the work on this thesis, a system was implemented for mood classification, and music composition based on the mood-classified data. This chapter describes the architecture of the system. Section 4.1 describes the overall architecture of the emotion classification system. Section 4.2 describes the classification network structure in detail. Section 4.3 describes the architecture used for music composition. Finally, section 4.4 describes the hardware and software requirements for running the system.

4.1. Overall Architecture

The system predicts the mood of a given music sample within the four quadrants of the Valence and Arousal (VA) plane, as initially described in 3.3.2 on page 23. The VA plane and quadrant model was chosen as it seemed widely used in existing work on Music Emotion Recognition (MER), and it has a simple, yet comprehensive structure argued to give a sufficient understanding of emotions. Music data is processed as tensors of their raw audio waveforms. Following this, a deep neural network trains a model on the annotated music data, classifying each sample based on the audio data only. The mood predicted by the model after the training phase is stored along with the sample. Finally, the annotated samples are used to compose new music conforming to a desired mood. The overall architecture can be seen in Figure 4.1. The following subsections explain each aspect of the system more thoroughly.

4.1.1. Dataset

The dataset, visualized as the "Music Database" element in Figure 4.1, used with the system was constructed by Panda et al. (2018). The music samples and related metadata in this dataset are gathered from the AllMusic API. The AllMusic API categorizes songs belonging to one or more of 289 distinct categories, not explicitly related to a known emotion categorization taxonomy. Thus, Panda et al. semi-manually mapped the categories to fit within the four **quadrants** of the VA plane, as displayed in Figure 3.2 on page 25. This includes the removal of songs where a dominant emotion quadrant could not be established given the category data, songs without genre information and songs without sufficient category information. After filtering, annotation validation was performed manually.

Table 4.1 describes the features provided in the dataset from Panda et al., and gives an example for each feature. The highlighted feature, Quadrant, indicates the sample's quadrant position in the VA plane, and is the feature used as the target in model training.

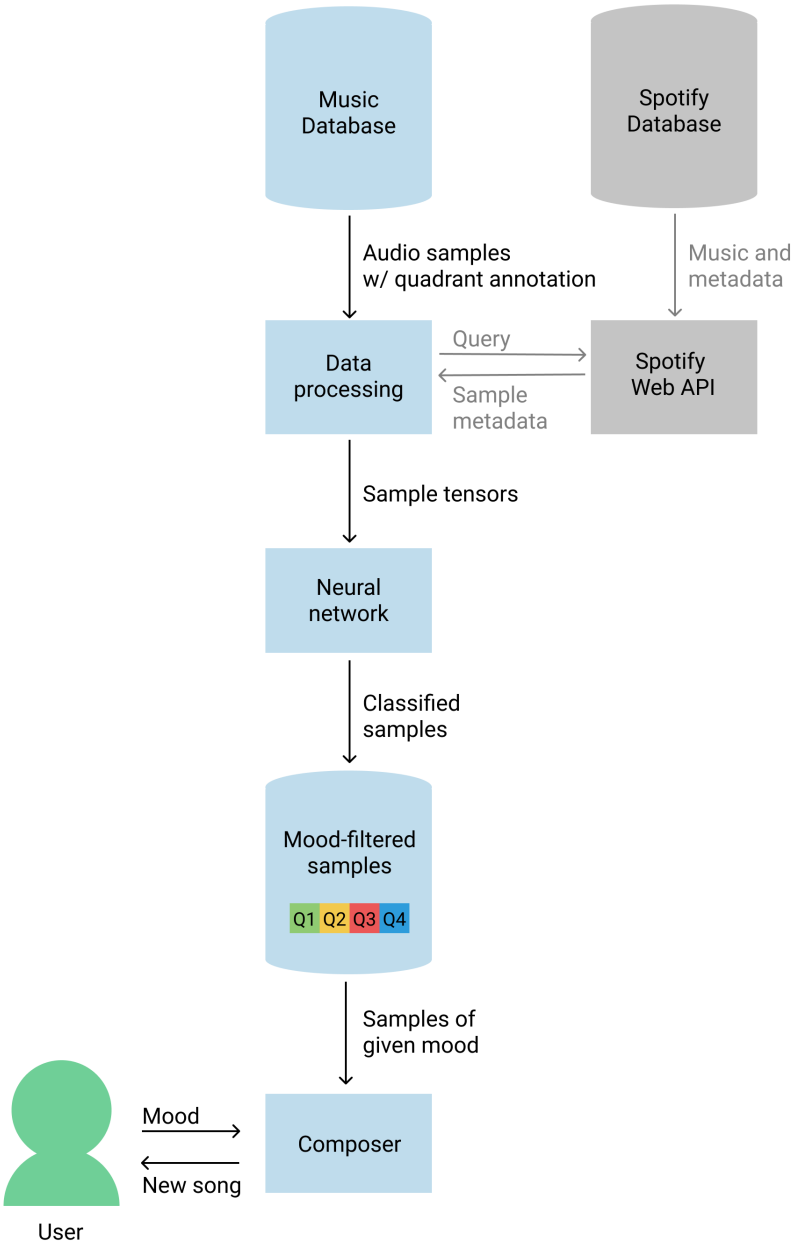

Figure 4.1.: Overall system architecture.

Feature name	Feature description	Example
Song	Song unique ID	MT0000044741
Artist	Artist name	Gipsy Kings
Title	Song title	Flamencos en el Aire
Quadrant	**VA-plane quadrant (Q)**	**Q4**
PQuad	Level of certainty in classification	0.75
MoodsTotal	No. of moods found in total	4
Moods	No. of moods corresponding to a Q	3
MoodsFoundStr	Moods corresponding to a Q	fiery, sexy, spicy
MoodsStr	All moods	Cathartic, Fiery, Sexy, Spicy
Genres	No. of genres	2
GenresStr	Genres	International, Jazz

Table 4.1.: Metadata given for each song in dataset from Panda et al. (2018)

Input format

The input data consists of MP3 music files in samples with a length of 30 seconds. As argued by Dai et al. (2017), deep convolutional networks should be able to construct their own understanding of musical features without specific functions to extract them. Thus, this was selected as the starting point for the algorithm, excluding hand-tuned features, metadata and spectrograms.

The labels given from the dataset were related to the four emotion quadrants: `Q1, Q2, Q3, Q4`. This should be noted as a possible point of improvement, as it does not position each sample within a quadrant. As an example, if a sample belongs to `Q3`, but in the XY-axis was placed very close to `Q2`, classifying the sample as `Q2` will be considered "equally wrong" as choosing `Q4`. A more accurate label would improve the system's ability to accurately indicate "how wrong" a prediction is. This issue is discussed in Section 6.2.5.

4.1.2. Data Processing

In order for a sample to be usable in a neural network, it must be processed into a tensor. This is the procedure executed in the Data processing step of Figure 4.1. The data loads from a CSV file containing all file names, as well as quadrant information. This section describes this procedure for one sample.

1. The MP3 sample is retrieved and loaded using the Torchaudio library[1], a library implemented for PyTorch for loading sound as a tensor. Data is normalized for each entry so that values are between `[-1, 1]`.

2. The two audio channels produced (because the sound is stereophonic) are averaged for each signal in each channel, to convert to mono sound on one channel only.

[1] https://pytorch.org/audio/

3. A tensor containing zeros of shape `[160000, 1]` is created, and then filled with the data from the audio sample. Samples with less than 160,000 signals are filled with zeros at the end, while longer samples are cut, so that all samples are of equal length.

4. The sample is reduced in quality to around 8 kHz, in order to reduce dataset size and training time. This is done by taking every fifth signal of data, in a new tensor of shape `[32000, 1]`.

With this procedure applied to all samples, all sound data is homogenous, normalized, equal in length and generally suitable for training. It should be noted that this procedure does reduce the audio quality quite significantly by using mono channel music and reducing its size by 80%. However, this kind of compression would still result in a perfectly recognizable sample if loaded back into sound, so it was considered a reasonable trade-off.

Training/testing data split

Many data sets include a predefined training/testing split, in which the testing split is constructed to ensure even representation of all classes and inputs. In the used dataset, no such splits were indicated, so PyTorch built-in functions for dataset splitting were used to construct a test dataset consisting of 20% of the samples chosen at random. Thus, it should be noted that a stratified distribution in the testing set is not guaranteed.

4.1.3. Data Querying with the Spotify Web API

As seen in the Spotify Web API element in Figure 4.1, a connection to the Spotify Web API can be made in the data processing, which returns data from the Spotify Database, containing songs and metadata for millions of songs. This is a procedure which can be run in the event that a quadrant is not provided for the sample. This is the case for the MAESTRO dataset, where a CSV file provides information on the composer and track name, but no emotion quadrant has been annotated.

This procedure works as follows:

1. An access token is established by using Spotify's Authorization Code flow and a private Spotify account.[2] An access token and a refresh token is granted, which is later used to send requests.

2. The metadata of artist and song name is extracted from the CSV, along with the file name used to load the correct MP3 data.

3. The artist and song names are processed so that spaces are replaced with `+`, and they are concatenated. Other signs such as `.` or `'` are not removed, as the search

[2] https://developer.spotify.com/documentation/general/guides/authorization-guide/#authorization-code-flow

Artist name	Song name	Query
Paul Gilbert	Let The Computer Decide	paul+gilbert+let+the+computer+decide
Dr. John	I Don't Wanna Know	dr.+john+i+don't+wanna+know

Table 4.2.: Search query processing for the Spotify Web API.

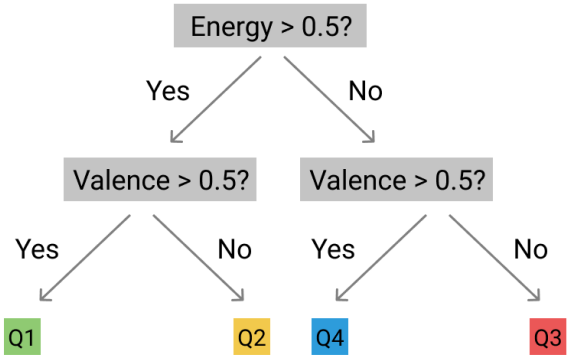

Figure 4.2.: Flowchart of quadrant classification using valence and energy measures.

engine can handle them as part of the search string. Examples can be seen in Table 4.2.

4. The Spotify Web API is queried using an HTTP GET query on Spotify's `audio-features` API endpoint. The parameters passed are the search string created, and the access token granted.

5. A JSON object containing information on zero, one or more songs given the search string. If results are found, each song comes with an `audio-features` JSON object containing audio features such as valence and energy (arousal). All features available can be seen in Table 5.4.

6. The Valence and Energy measures are combined to produce a sample quadrant, where 0.5 constitutes a threshold for classification on high or low energy and valence. A classification flowchart can be seen in Figure 4.2.

7. The determined quadrant is stored with the sample and can be used for training in the same manner as discussed above.

4.2. Classification Network Structure

This section describes the structure of the classification network, visualized as the Neural network element in Figure 4.1. The initial network structure itself was inspired by Dai et al. (2017), as an efficient and comprehensive solution to classification based on raw

audio data. However, the solution has been substantially modified to adapt to the emotion classification problem with a musical dataset, whereas Dai et al. worked on recognition of urban sounds such as car horns, barking or children playing. The network structure can be seen in Figure 4.3. For simplicity, the several layers following the routine of convolution/max pooling/batch normalization have been truncated in the figure.

Detailed layer information for the first three layers can be seen in Table 4.3. The most important information here is the growth of the different numbers. The receptive field is large for the input layer, but very small in the following convolutional layers. The number of feature maps doubles for each layer, while the input length decreases because of the max pooling. The reasons for this are explained in this section.

Figure 4.3.: Classification network structure, with a truncated view of the convolutional, max pooling and batch normalization layers.

Layer	Receptive Field	Feature maps	Length
Convolutional Layer 1	80	128	8000
Max Pooling Layer 1		4	2000
Convolutional Layer 2	3	256	2000
Max Pooling Layer 2		4	500
Convolutional Layer 3	3	512	500
Max Pooling Layer 3		4	125

Table 4.3.: Layer specifications in classification network, first three layers.

4.2.1. Fully Convolutional Network Design

The network constructed is a *fully convolutional* design. Many deep convolutional networks use several fully connected high-dimensional layers for discriminative modelling. However, this method gives a very high number of parameters in the network, leading to high computational costs. With a fully convolutional design, no fully connected layers are used, which is hypothesized to force the network to learn a fitting representation in convolutional layers and develop better generalization abilities (Dai et al., 2017; He et al., 2016; Long et al., 2015). The fully convolutional architecture, as described by Long et al. (2015), shares similar traits with the architecture used in this thesis, in the sense that the input layer has a large receptive field, whereas following layers have narrow receptive fields, essentially reducing how much of the input is considered at once. For our purpose, this translates to that the beginning layers consider higher-level features, while deeper layers consider lower-level features. Combined, a fully convolutional design proves powerful in building models for inputs with hierarchies of features.

4.2.2. Input Layer

The input audio is represented by a single input tensor, visualized in Figure 4.3 as the Input vector element. Therefore, the stereo sound provided in the dataset is processed as discussed in Section 4.1.2, in order to ensure homogeneity and equal quality for all samples. The receptive field for the input layer is very large compared to the receptive fields in the following layers. This is done in order to enable learning for high-level music features in the early layers. This is an attempt at mimicking the behaviour of a band-pass filter, i.e. passing frequencies in a particular range and rejecting others, used in many kinds of audio processing. A receptive field of size 80 covers around 10 milliseconds of one sample, which is argued to produce the best possible performance by Dai et al. (2017).

4.2.3. Convolutional Layers

The samples pass through several convolutional layers, which is visualized as the Convolution element in Figure 4.3. Dai et al., and originally Simonyan and Zisserman (2014), suggest that very small receptive fields for each layer (except the first input layer) allow for a reduction in the number of parameters for each layer, and thus restrain overall

model size when working with many layers. This is used in this architecture as well. Between each layer, max pooling with large strides is also used to substancially reduce computational costs deeper in the network.

In each layer, a feature map encodes activity level of the associated convolutional kernel. The number of feature maps doubles as temporal resolution decreases in each max pooling layer, visualized as the Max Pooling element in Figure 4.3. This trade-off between the number of feature maps and temporal resolution throughout the network, as reflected in Table 4.3, allows different levels of specialization. Higher layers focus on overall structure, and lower layers focus on more basic musical features.

After each layer, batch normalization, visualized as the Batch normalization element in Figure 4.3, is applied in order to reduce *internal covariate shift*, i.e. the problem of network gradients changing too much, too soon, caused by the amplification of small changes in the first layer being amplified as network depth increases (Ioffe and Szegedy, 2015). After batch normalization, the ReLU activation function is applied.

4.2.4. Output Layer

In the final layer, a global average pooling is applied directly following the final max pooling. In the end, a softmax function is applied to determine the most likely emotion quadrant.

4.3. Music Composition

This section describes the Composer element in Figure 4.1, which takes the element called Mood-filtered samples as its input. The input format is a CSV file containing file names, used to fetch music data, and the annotated quadrants determined by the network.

For music composition, an existing architecture was used, building on the Music Transformer principles, namely, the Pop Music Transformer as presented by Huang and Yang (2020) and introduced in Section 3.3.5. The Pop Music Transformer was used as is, with modifications so that training is performed on the MAESTRO dataset, and that fine-tuning is performed on the mood-annotated data from the classification model. This section describes the Pop Music Transformer architecture and the modifications done to adapt the system to the use in this thesis.

4.3.1. Pop Music Transformer

The Pop Music Transformer, by Huang and Yang (2020), presented in Section 3.3.5, is a tool for automatic music composition, which serializes a musical score in a sequence of *events*, similar to the MIDI format. This event set, called Revamped MIDI-Derived Events (REMI), is used for sequence modelling rhythmic and melodic patterns of the music. The architecture is based on the Music Transformer Architecture (Huang et al., 2019).

The typical event in the MIDI format is the `NOTE_ON` or `NOTE_OFF` event for any note entered by an instrument. The events are based on discrete time ticks, making a metrical

structure such as bars and beats, as well as natural variations in tempo, something that a sequence model has to reconstruct, often with difficulty.

The REMI structure introduces the `BAR` event, representing one bar, and a `POSITION` event to indicate the placement of a note in the bar, which is split into 16 discrete points. The two combined create a grid structure, more similar to the way humans write musical scores.

Within this structure, other musical tokens are added, such as `TEMPO`, allowing for local changes in tempo without interfering with the notes' positioning. `CHORD` events are introduced to state harmonic structure in the music explicitly.

The musical components in the Pop Music Transformer aim to express music more similarly to how humans compose music. Building on the attention-based model of the Music Transformer, listening tests prove this system preferred to the Music Transformer (Huang and Yang, 2020).

4.3.2. Composition with Mood-Annotated Music

In order to bridge the gap between the annotated data and the Pop Music Transformer, the existing fine-tuning mechanism of the system was utilized. To use this, a model either needs to be pre-trained, or trained as the first step of running the code.

When an initial model is defined, a set of MIDI files is input as the new data corpus on which to fine-tune. These music samples are annotated with a quadrant, and only the files belonging to the input quadrant are used. Then, the system trains for 200 epochs on this new data corpus, on top of the existing model. This way, some common information will be dispersed no matter which mood is input, ensuring that the model is built on a sufficient amount of data to produce meaningful results.

4.4. System Requirements

The system can be run on any computer using Python version 3.7 and the ability to download packages for Python.

As training is a computing-intensive task, it is recommended to perform this task on a computer with access to GPU resources. The training procedures used in order to train the models described in this thesis was performed on the NTNU IDUN high-performance cluster.[3] The resources accessed were NVIDIA GPU resources. As the IDUN Cluster is a distributed system, exactly which resources are used can vary between the following: NVIDIA Tesla P100, NVIDIA Tesla V100 16GB and NVIDIA Tesla V100 32GB. No changes are needed to run the code on CPU resources, as the code detects this automatically. However, depending on the size of the dataset used, this is a very time-consuming task.

[3] https://www.hpc.ntnu.no/idun

5. Experiments

The experiments conducted in this thesis aim to address different aspects of the research questions RQ1 and RQ3, as presented in Section 1.2. Namely, they involve experimentation within the architecture described in Chapter 4, as to what configurations are efficient for classifying mood, and for composing new music.

As the music composer used is an existing system, a smaller focus has been set on this in the experiments. The neural network used for emotion classification is the main focus, and several aspects of the network and its input data have been addressed in different experiments.

Section 5.1 describes the experimental plan for five different experiments. Sections 5.2-5.7 describe the setup and execution of each of the described experiments. Section 5.8 describes a survey designed and conducted for evaluation of the music produced.

5.1. Experimental Plan

When training an ANN, a large array of parameters and aspects are subject to change, which in turn all may affect performance in some way. The goal is, of course, to achieve the "perfect combination" of all possible parameters. However, this would require a very laborious testing procedure, where many combinations may not be relevant to each other and have small effects on performance.

Due to this, and in order to reduce the experimentation scope of this thesis, the most promising results for each experiment are used in the system throughout each next experiment. If several results seem similar in terms of performance, one is selected and explicitly stated at the start of the next experiment. This is a somewhat naïve approach, as it neglects that a seemingly optimal result of the first experiments may prove sub-optimal in combination with some other set of parameters later on. However, this approach was deemed feasible given the time constraints on the thesis work.

The experiments were conducted in the following order:

Experiment 1: Learning rate adaptation. This includes determining the initial learning rate, using an optimization algorithm to dynamically adjust the learning rate, and at which rate the learning rate should be adjusted.

Experiment 2: Network depth. Comparison of loss and accuracy using between four and eight convolutional layers.

Experiment 3: Expanding the training dataset. An expanded version of the dataset from Panda et al. (2018) was acquired, and used as input instead of the dataset

described in the original paper.

Experiment 4: Metadata incorporation. In the original dataset, several metadata attributes were available. This experiment performs simple tests on using some metadata as an explicit part of the network's input.

Experiment 5: Classification using Spotify track features. The Spotify Web API is an openly available tool that, among other features, provides Valence and Arousal (energy) values for any song in the Spotify database. The dataset is classified using this service and compared to the annotation given by Panda et al.

5.2. Learning Rate Adaptation

To dynamically adjust the learning rate, the Adam optimizer was utilized. Adam, the name derived from *adaptive moment estimation*, is an optimization algorithm that updates network weights for each parameter individually as learning progresses (Kingma and Ba, 2015). An initial learning rate is supplied, as well as a weight decay rate, indicating at which rate learning should decrease within epochs.

The loss function used for all experiments is the negative log likelihood loss. The function is built into Pytorch.[1]

Table 5.1 describes several different implementations of the Adam algorithm and their effect on accuracy on the testing accuracy after 300 epochs of training. Also, each iteration runs classification on the Maestro dataset after training. The distribution of quadrants (Q1/Q2/Q3/Q4) is also indicated in the table.

5.3. Network Depth

For the exploration of network depth, the learning rate setup used was the Adam algorithm with an initial learning rate of 0.01 and a weight decay of 0.001.

Increasing depth in the network implies a doubling of feature maps, as explained in Section 4.2. For music, this means that more resources are allocated to identifying low-level features.

For each iteration, training was performed over 300 epochs. In Table 5.2, various network depths are compared. Results are measured using the loss in the final training epoch and the accuracy on the testing set after the final training epoch.

5.4. Expanded Dataset

The original dataset presented by Panda et al. (2018) contains 900 songs. For deep learning purposes, the dataset should be larger. This is especially because the system in this thesis works on raw audio data, not any other derived features or metadata, other than what the network itself can deduce from the audio data. In their article, Panda

[1] https://pytorch.org/docs/stable/nn.html#torch.nn.NLLLoss

Initial learning rate	Weight decay	Testing accuracy	Maestro distribution			
			Q1	Q2	Q3	Q4
0.1	0.0001	49%	0	0	8	1176
0.1	0.001	49%	0	0	493	691
0.1	0.01	48%	0	0	46	1138
0.1	0.1	20%	0	0	0	1184
0.05	0.0001	46%	0	0	0	1184
0.05	0.001	50%	0	0	11	1173
0.05	0.01	52%	0	0	0	1184
0.05	0.1	24%	1184	0	0	0
0.01	0.00001	42%	0	0	163	1021
0.01	0.0001	46%	5	1	353	825
0.01	0.001	49%	2	0	819	363
0.01	0.01	44%	2	0	1133	49
0.01	0.1	41%	0	0	1184	0

Table 5.1.: Classification accuracy for various learning rate setups

Number of convolutional layers	Loss	Testing accuracy	Maestro distribution			
			Q1	Q2	Q3	Q4
4	0.030592	47%	1	0	793	390
5	0.110965	54%	5	1	353	825
6	0.042910	51%	4	0	824	356
7	0.005466	52%	8	0	679	497
8	0.117877	49%	49	0	900	235

Table 5.2.: Loss and testing accuracy for different network depths.

Dataset	Loss	Testing accuracy	Maestro distribution			
			Q1	Q2	Q3	Q4
Panda et al.	0.110965	54%	5	1	353	825
Panda et al. expanded	0.035813	52%	2	0	0	1182

Table 5.3.: Loss and testing accuracy for the original and expanded dataset.

et al. (2018) state that larger versions of the dataset exist and that the size of the dataset has been reduced for several reasons. One is that the manual validation and annotation is labour-intensive. Another is that much of the data was sparsely annotated in terms of metadata, possibly making the automatic part of the processing inaccurate. A third reason is that the data was unbalanced in terms of the number annotations for each emotion quadrant and genre. In an effort to acquire a larger usable dataset, contact was made with the authors of the original dataset. Luckily, they showed great interest in the project and were able to contribute. A new, much larger dataset was provided.

The expanded dataset contains 51,781 samples, with the attributes SongID, artist, title, moods and genre extracted from the aforementioned AllMusic API. Each sample is also annotated with a Quadrant, computed by Panda et al., making the metadata match the original dataset, as described in Figure 4.1. However, in the expanded dataset, most of the songs were not used in the original dataset of 900 songs because the computed quadrant probability was below 0.5. 686 of the songs in the expanded dataset were not associated with any quadrant at all `(i.e., Q0, PQuad=0)`. 32,138 of the songs have a computed quadrant probability of less than 0.5.

Table 5.3 displays the difference in loss and testing accuracy, as well as the distribution over quadrants for the MAESTRO dataset after 300 epochs of training. Both used five convolutional layers and an initial learning rate of 0.01 and a weight decay of 0.001. The loss and accuracy measures are averaged over 10 runs.

5.5. Metadata Incorporation

This experiment involves the exploitation of metadata in the working dataset, such as information on genre, artist or song names, in order to improve accuracy.

With using music data only, the input vector of one music sample is of the shape `[32000, 1]`. However, the dataset used also provides metadata with each sample. Figure 4.1 describes each metadata feature and gives an example.

In the classification process, the Quadrant column is used as the baseline for the classification, using only the music content as input. This experiment tests the network using metadata in various ways to support network decisions.

Using metadata as a part of the network's input

A natural starting point for using metadata is incorporating it into the network as a part of the input. This would mean that the input for `n` samples, originally at size `n x 1`

`x 32000`, would be expanded to size `n x 1 x 3200+m` where m is the dimension of the input metadata.

The initial experiment introduces the metadata feature *GenresStr*, where the set of genres found in a song is encoded as a m-length "one-hot" vector (although a song can belong to $n \ll m$ genres), where m is the total number of genres found in the dataset.

The feature *GenresStr* was chosen because of its somewhat indicative nature on music, although it has no linear connection to the labelling. Features such as song or artist name were not used because they would need further grouping or semantic encoding in order to make sense in a classification problem. *MoodsFoundStr* could have been used, but was not chosen in the first round due to its more direct relation to the annotated quadrant.

In order to produce the complete list of genres, all genres were extracted from the dataset's samples as no such list was provided directly. Genres were split on the characters `,` and `/`, and spaces, and stripped of casing to prevent duplicates. With this process, 23 genres were found, and all samples were appended with the "n-hot" vector produced for each sample.

To compare performance with and without metadata, the process was run on the exact same code, on the same computer with the same memory and processing specifications, on 100 epochs of training.

5.6. Classification using Spotify Track Features

A known factor inhibiting machine learning on music is the lack of consistent, high-quality annotation on larger datasets. This experiment explores the use of Spotify metadata, which aspects of it can be used to annotate mood and to which extent the provided data aligns with the semi-manual annotations of the dataset from Panda et al. (2018).

5.6.1. Feature selection

For any song available on Spotify, metadata and *Audio Features* can be extracted using the Spotify Web API.[2] The audio features provided can be seen in Table 5.4.

Some of these features are simple to measure, while others are more complex and require the balancing of several both high- and low-level features of the music. No further information is listed on Spotify's documentation on how these features are captured, thus no guarantee can be given as to their quality. However, these measures are widely used in Spotify's own music recommendation work. This, combined with its ease of access, makes it relevant in the process of creating annotations for larger datasets.

The two features most directly aligned with our notion of valence and arousal are *valence* and *energy*, based on the description from Spotify. However, several other measures can also prove relevant. For *valence*, *mode* can be a simple indicator (major or minor). For *arousal*, some relevant measures could be *danceability*, *loudness* and *tempo*.

[2] https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

Feature name	Value description	Range
duration_ms	The duration of the track in milliseconds	Hundred-thousands
key	The estimated overall key of the track	-1 (no result), 0-11
mode	The modality (major or minor) of the track	0 or 1
time_signature	Estimated overall number of beats in each bar	Typically 2-7
acousticness	Confidence of whether the track is acoustic	0.0 to 1.0
danceability	Describes how suitable a track is for dancing	0.0 to 1.0
energy	Describes a measure of intensity and activity	0.0 to 1.0
instrumentalness	Predicts whether a track contains no vocals	0.0 to 1.0
liveness	Detects the presence of audience in the recording	0.0 to 1.0
loudness	the overall loudness of a track in decibels (dB)	Typically -60 to 0
speechiness	Detects the presence of spoken words in a track	0.0 to 1.0
valence	Describes the musical positiveness conveyed	0.0 to 1.0
tempo	The overall estimated tempo of a track in BPM	Typically 50-200

Table 5.4.: Available Spotify Web API musical features.

Valence measure	Arousal measure	Result
Average of Valence and Mode	Average of Energy and Danceability	0.42
Valence	Average of Energy and Danceability	0.46
Valence	Energy	0.49

Table 5.5.: Different configurations of valence and arousal measures and their agreement with Panda annotations

5.6.2. Comparison of Spotify classification and Panda et al. classification

In order to learn more about Spotify's musical feature metadata, tests were performed towards the dataset and annotations of Panda et al. (2018). As Spotify provides a variety of metadata attributes which can possibly contribute in the creation of annotations, several combinations were tested. Of the possibly relevant measures, two were selected for each axis (the others were also tested, but worsened results). Results can be seen in Table 5.5.

As the usage of the Energy and Valence features initially proved most accurate, the results from this annotation process can be seen in Figure 5.1 and Figure 5.2. The plots have been split in two for ease of understanding.

As can be seen in Figure 5.1 and Figure 5.2, there is evidently some connection between Spotify's and Panda et al.'s annotations. In the figures, a dot indicates a music sample, and its position in the X/Y plane indicates its levels of arousal and valence. The coloured background indicates the four quadrants. The colour of the dots indicates the original annotation from Panda et al. This means that, as an example, a blue dot on blue background indicates agreement between the two sources on that given sample, while a blue dot on yellow background indicates that Panda suggests Q1, while Spotify's

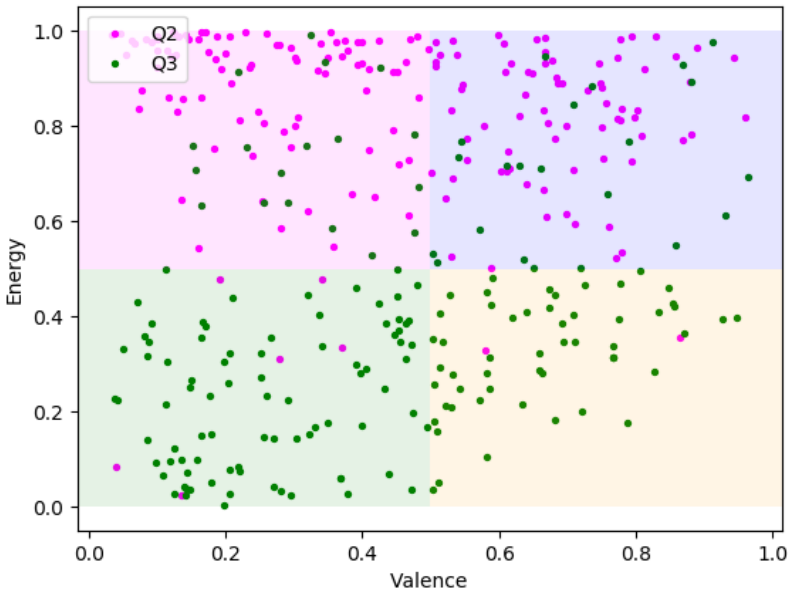

Figure 5.1.: Plot in the Valence-Energy(Arousal) plane for Q2 and Q3.

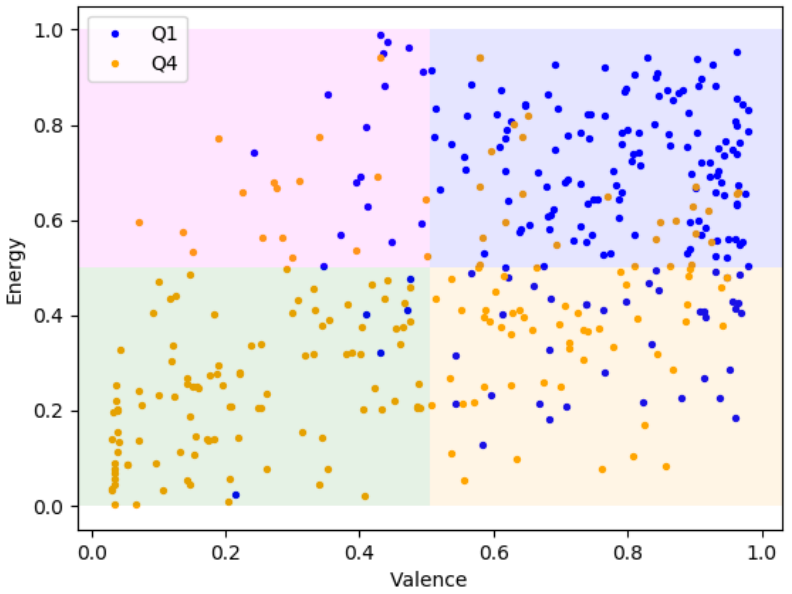

Figure 5.2.: Plot in the Valence-Energy(Arousal) plane for Q1 and Q4.

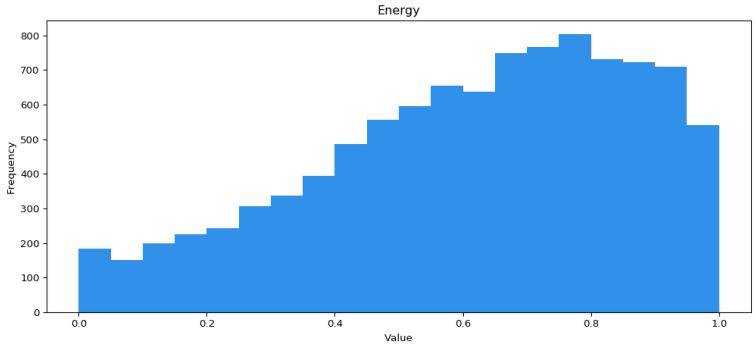

Figure 5.3.: Distribution of the *energy* measure in Spotify's data.

metadata suggests Q4.

The level of agreement from the two sources varies from quadrant to quadrant.

For Q1, we see a relatively high level of agreement, where Spotify's metadata identify 61.5% of Panda's Q1 annotated samples. We can also see that the samples with differing annotations are quite close to Panda's annotated quadrant in the X/Y plane.

For Q4, the situation is quite different. Spotify's metadata only agrees on 27% of Panda's annotations, and a large amount of samples is labelled with Q3 instead. As can be seen from Figure 5.2, there is a dense area where Spotify's metadata assigns these samples valence values close to zero, where Panda et al. disagree. The reason for this is difficult to pinpoint, but one reason may be a difficulty in distinguishing calm and relaxing music with regards to valence.

For Q2, we see a high level of agreement when it comes to energy levels, and some disagreement on the valence levels.

For Q3, Spotify's metadata gives quite widespread results. While the metadata aligns with 54% of Panda's annotated samples, the samples not agreeing with Panda's estimated quadrant seem to be spread out over the X/Y axis, more so than the remaining quadrants, with a slight overweight on Q4.

5.7. Composition Using Mood-Annotated Music

The final step in the pipeline of understanding mood is the generation of new music which uses the acquired mood understanding. This section explains how the mood-annotated data is used in order to condition a generative model to create music according to a mood.

5.7.1. Dataset Selection

For composition on mood-annotated music, the MAESTRO dataset was chosen. One great merit of this dataset is that all data is available both in MP3 and MIDI format, which is suitable because the classification process uses the MP3 format, while the

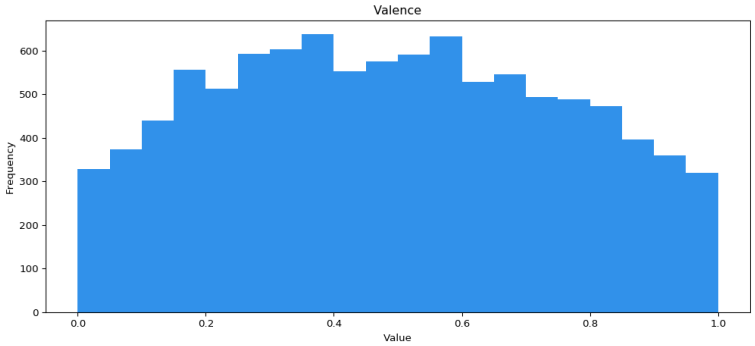

Figure 5.4.: Distribution of the *valence* measure in Spotify's data.

composer uses the MIDI format. Thus, the pipeline between the two segments can be automated as the file naming is entirely consistent between the data in the two formats.

5.7.2. Training on the Selected Dataset

The MAESTRO dataset has no emotional quadrant annotation, and performing such a task would be labour-intensive, as well as not guaranteed to align with methods of annotation performed in the dataset from Panda et al. (2018), used to train for classification. The datasets differ in that Panda et al. classify music of all genres, while the MAESTRO dataset is comprised of classical piano pieces. Thus, using the pre-trained classification model on the MAESTRO dataset could induce some bias. This is discussed further in Chapter 6.2.2. For the purpose of experimenting, however, the model trained on the dataset from Panda et al. was used for classification of the MAESTRO dataset.

5.7.3. Producing Music

For the composition of new music based on the annotated dataset, the Pop Music Transformer (Huang and Yang, 2020) architecture was used as the composer.

The training of the composer is performed in two phases:

1. **Training on the MAESTRO dataset.** The entire dataset is used in training a model. This is to ensure that enough data is available to the system to acquire a sense of its musical "universe", musical structure and other general features.

2. **Finetuning on mood-annotated data.** After the initial training, a smaller dataset consisting of mood-annotated data of one specific mood is used for further training.

After both stages of training, a *checkpoint* is stored so that the model does not need re-training for each run unless prompted, and so that finetuning is performed on the same base model for any mood.

As discussed in Section 6.1.1, the classification model was not able to reliably classify samples from the MAESTRO dataset into the two high-arousal categories, Q1 and Q2. Thus, there was no reliable corpus to use for producing music in these two quadrants. Therefore, composition and testing could only be performed on samples annotated with Q3 and Q4.

5.7.4. Network Configuration

In training the composer and producing new music, several parameters are adjustable both in training and network setup. The system is run with different configurations in order to explore their impacts on the composed samples' originality and conformity to the applied data.

Aspects studied were:

1. **Temperature:** A number ranging from 1 to 10, indicating a degree of mutation allowed in generation. A low temperature leads to less originality and more replication of the applied data, while a high temperature causes the music to sound more original, or "random". Thus, a healthy balance is important to create music that is both novel and pleasant.

2. **Emphasis on finetuning.** In a relatively uniform dataset such as the MAESTRO, finetuning can be applied rigorously in several ways. There are two main methods for this: Dataset size and the number of training epochs. A large finetuning dataset can ensure breadth in the composed results, but it can be difficult to verify that the resulting model truly has learned from the finetuning dataset. In adjusting the number of training epochs, the model increasingly focuses on the finetuning dataset and less on the base model.

5.8. Survey

In order to evaluate the results of the composition on mood, a digital survey was conducted.

The survey conducted was a quantitative study, performed online. Initially, a wish was to conduct such a survey with participants from public areas, such as on the street. However, due to COVID-19 restrictions in the spring of 2020, no such activity could be conducted.

The main goal of the survey was to uncover whether music produced in accordance with some mood appears to align with that mood in the ears of the listener.

5.8.1. Music Sample Configuration

For each combination of composer network configurations introduced in Section 5.7.3, one sample was tested in the survey. However, when running the composer, the same

Quadrant	Temp.	Finetune samples	Training epochs	ID	Included
Baseline	*1.2*	*20*	*300*	-	-
Q4	**0.8**	20	300	Q4-08-20-300	Yes
Q4	**1.2**	20	300	Q4-12-20-300	Yes
Q4	**1.6**	20	300	Q4-16-20-300	Yes
Q4	1.2	**10**	300	Q4-12-10-300	No
Q4	1.2	**40**	300	Q4-12-40-300	Yes
Q4	1.2	20	**100**	Q4-12-20-100	No
Q4	1.2	20	**500**	Q4-12-20-500	Yes
Q3	**0.8**	20	300	Q3-08-20-300	No
Q3	**1.2**	20	300	Q3-12-20-300	Yes
Q3	**1.6**	20	300	Q3-16-20-300	Yes
Q3	1.2	**10**	300	Q3-12-10-300	Yes
Q3	1.2	**40**	300	Q3-12-40-300	Yes
Q3	1.2	20	**100**	Q3-12-20-100	Yes
Q3	1.2	20	**500**	Q3-12-20-500	Yes

Table 5.6.: Experimental setup for music composition.

combination of features could result in widely different composed music. Thus, the result produced by the algorithm is not guaranteed to be representative.

Table 5.6 displays the configuration of the samples produced and used in the survey. The main parameters tested were the temperature, finetuning sample set size, and the number of training epochs. Combining all features and variations upon them gave 14 distinct samples.

Baseline measures were selected, following the base inputs used by Huang and Yang (2020). For the finetuning sample set size, no baseline input was provided, so 20 was chosen as a number which would give a substantial amount of data, while all samples could be recognized in the case that the model should purely duplicate one of the samples.

In the survey, one sample for each of the experimental setups was shown. However, three samples were omitted from the study. These were samples of "failed" training, which sounded highly random and unpleasant. The selection of samples to omit was judged on the author's own opinion. Which samples were included and omitted can be seen in the *Included* column in Table 5.6.

The sheet music for all samples used in the survey, including omitted samples, can be found in Appendix B. The samples produced for the survey can be seen in Table 6.2.

5.8.2. Survey Design

For each sample, the participant was asked to rate the song at hand on two measures: Energy and valence. The two measures were explained as follows:

- *Valence* is the measure of positive or negative emotion expressed. Feelings with positive valence could be happiness, joy and content, while feelings with negative

valence could be anger, sadness, depression or despair.

- *Arousal* expresses the energy level of the emotion expressed. A high level of arousal is found in emotions such as exhilaration, excitement, rage or shock. A low level of arousal is found in emotions such as tiredness, sleepiness, relaxation and calmness.

The participants were asked to evaluate each song for measures of valence and arousal on a scale from 1-5.

A side goal was for the participants to evaluate the music on enjoyability and quality overall. Adopting terms from another survey, namely Olseng (2016), participants were asked to evaluate the songs on three criteria: *pleasantness*, *interestingness*, and *randomness*. These criteria were not explained explicitly, but rather shown as antonyms on a scale, where *pleasant*, *interesting* and *random* were shown as antonyms of *unpleasant*, *boring* and *structured*, respectively. *Pleasantness* is a measure indicating what the individual listener finds pleasing. It is not necessarily a measure of musical quality, but rather what is pleasing to the ear. *Interestingness* means to explore whether the music has interesting features, regardless of whether they are pleasant or not. However, as introduced by Olseng, this measure can be prone to a fatigue bias. *Randomness* asks the participant to evaluate the structure of the music. While some randomness is useful for making the piece interesting, it should not appear completely random in order to be convincing. A piece with high randomness may be deemed unlikely to sound like a human composer created it. Participants were asked to rate each sample for these measures on a scale from 1-5.

For each sample, a text box was included, where the participant had the option to include comments on the sample they had just heard. Also, as a background question, participants were asked to briefly explain their relationship with music.

The participant group consisted of 101 fellow students or other people in the social network of the author's friends and family. This group is not a representative group of an entire society, as the possibility to reach out to a broader and more diverse group was reduced due to COVID-19 restrictions. This and other biases are discussed in Section 6.2.8.

6. Evaluation and Discussion

This chapter presents results for the experiments conducted and discusses the significance of these results in relation to related work. Section 6.1 presents the results and evaluation of experiments conducted on the task of emotion classification and music composition. Also, a survey is presented as a tool for evaluating the music composed. Section 6.2 presents a discussion on key topics that influence the performance of the system.

6.1. Evaluation

6.1.1. Emotion Classification

In the task of emotion classification, answering **RQ1**, the Panda et al. (2018) dataset was used for training. This dataset utilised the four emotion quadrants on the valence/arousal plane as its emotion taxonomy, making this the focal point explored in answering **RQ2**. The deep neural network implemented is able to perform classification with a testing accuracy of 54%, averaged over 20 runs on 300 epochs of training, with variations between 50 and 60% for different runs due to differing random seeds. Given that it runs on a dataset of 900 samples only, and has no explicit musical understanding, this can be seen as impressive as it indicates that the network to some extent has built its own understanding of musical features indicative to each quadrant. However, much work remains to be done before it can be put into use for this purpose.

Figure 6.1 displays the confusion matrix for the test set of 180 samples. Here, we can see that the performance of the classifier varies for each quadrant. For Q2, classification is quite accurate. However, Q1 is often mistaken for Q3, and the model struggles to distinguish the low-arousal quadrants Q3 and Q4. The latter issue is one also present for this dataset with the SVM architecture used by Panda et al. (2018).

With this system, we can to some extent conclude that the network itself develops mechanisms to distinguish these four emotion categories. While not entirely ready for use, it supports the notion that expert knowledge or hand-tuned musical features may not be needed in the future for machines to make their own understanding of music.

Emotion Classification of the MAESTRO dataset

The model produced was trained on the dataset presented by Panda et al. (2018). However, for composition, the model was used to classify emotions in the MAESTRO dataset (Hawthorne et al., 2018). This proved a very difficult transfer learning task for the model. As can be seen in Tables 5.1, 5.2 and 5.3, no setup for the system was able

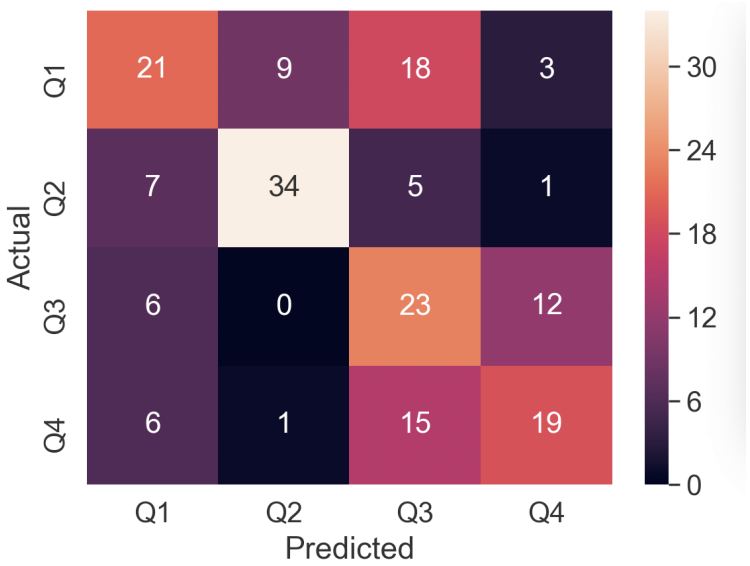

Figure 6.1.: Confusion Matrix for emotion classification, based on test set of 180 samples.

to produce results that gave even a close to even distribution of the quadrants for the MAESTRO dataset. Many setups classified all samples either to Q3 or Q4, and some setups classified a more even distribution between Q3 and Q4. However, samples rarely were classified into Q1, and virtually never into Q2.

In light of these results, a broad conclusion can be drawn that the model seems to struggle with identifying high arousal in the MAESTRO dataset. There can be many reasons for this. The main reason hypothesized is that the dataset from Panda et al. consists of music of all genres, while the MAESTRO dataset consists of classical piano music only. Naturally, classical piano music is overall a less energetic genre than genres such as rock or electronic music.

6.1.2. Music Composition

For music composition, seven different configurations were used to produce samples belonging to each quadrant. Each sample consisted of roughly 16 bars, lasting between 30 seconds and one minute, depending on the tempo of the song produced. This subsection presents a small selection of the sheet music, to provide an overview of the overall quality of the produced results.

All sheet music and music clips in .MP3 and .MIDI formats can be seen and listened to in the thesis attachments or on Google Drive.[1]

In general, the subjective quality of the music produced seems highly inconsistent. For one configuration, running the composer system many times can produce very different results. However, some tendencies were evident across many configurations, speaking to

[1] https://drive.google.com/drive/folders/1AJ9DkSOIr-2mV-Cio8W_PgBKAdvhVC09?usp=sharing

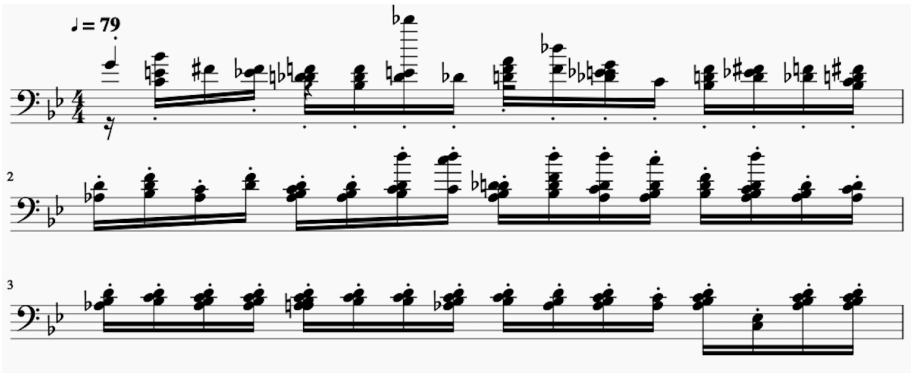

Figure 6.2.: Sheet music, sample Q4-12-10-300.

the quality of the composer overall.

Figure 6.2 shows a tendency found in many composed samples, of continuing, rapid staccato (short and decisive) notes. There is also widespread use of dissonant notes in the sense that half-note intervals are played simultaneously. This piece of sheet music is an extreme example of a recurring issue in several of the pieces produced, namely segments of very rapid note changes, difficult for the ear to keep up with.

On the other hand, many composed pieces have very pleasant qualities. Several compositions prove very successful in establishing recurring themes which create coherence and enjoyment in the music. Figure 6.3 displays a sample carrying these traits. Red markings (inserted by the author) indicate one recurring musical element, while blue markings show another. These elements are, in spite of messy-looking notes, very similar, and are recurring in different keys and variations throughout the song, creating a sense of progression and coherence.

6.1.3. Survey Results

This section presents the results from the survey conducted. First, the participant group is described. Secondly, the scores for music pleasantness, interestingness and randomness are presented. Finally, the participants' opinions on emotion conveyed, including written comments are described, and seen in relation to different qualities in the music produced. Discussion points drawn from the survey results can be found in Section 6.2.7. Strengths and weaknesses with the survey are discussed in Section 6.2.8.

Participants

A total of 101 participants contributed to the survey. A total of 1,073 music samples were evaluated.

The survey participants freely described their relationship with music. Of the 101 participants, 100 answered this question, and all 100 described themselves as consumers (assuming that a composer or musician is also a consumer, even if it is not mentioned

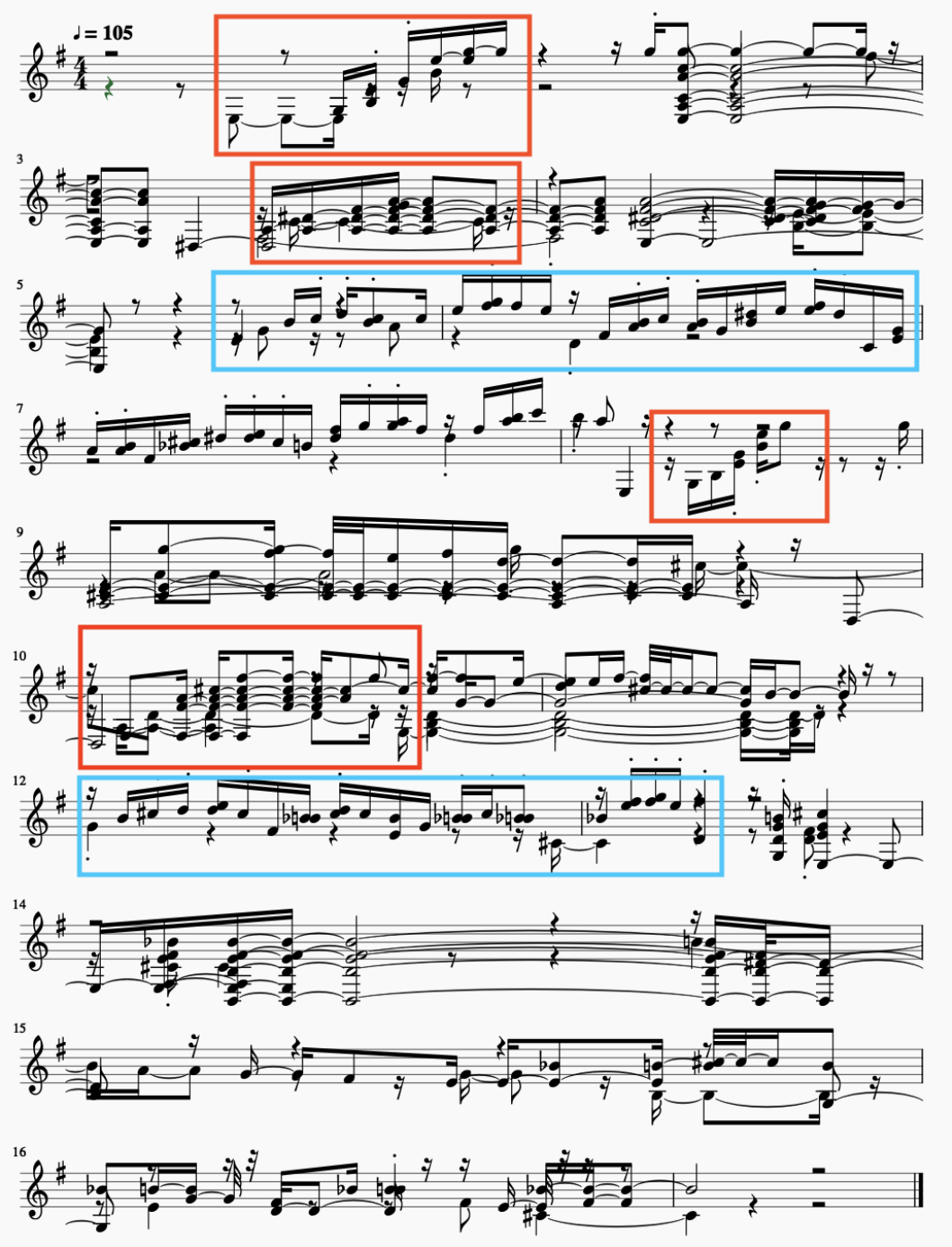

Figure 6.3.: Sheet music with recurring themes highlighted, sample Q4-12-40-300.

	Pleasantness	Interestingness	Randomness
Average	3.02	3.12	3.11
Standard Deviation	0.51	0.32	0.37
Median	3.09	3.08	3.10

Table 6.1.: Aggregate scores for Pleasantness, Interestingness and Randomness metrics.

explicitly). Nine also described themselves as composers or songwriters in a hobby or professional manner. Fifty-nine identified themselves within playing one or more instruments.

Pleasantness, Interestingness and Randomness Scores Table 6.1 presents the average scores, the standard deviation and median scores for all samples for the measures of Pleasantness, Interestingness and Randomness.

The sheet music for all samples used in the experiment, including the omitted ones, can be seen in Appendix B. The survey results for each music sample can be seen in Table B.1.

Participant Sentiment and Qualitative Observations

The survey feedback places pleasantness, interestingness and randomness slightly higher than the mean value, 3. The feedback varies with each sample. All sentiments can be found in Appendix B.

Many of the sentiments included give feedback that the music sounds "Mechanic" or "stops abruptly", and that this at times feels unpredictable in a song. One participant comments, for sample Q4-12-20-300: "It's periodically fine, however some notes are clearly off, and it changes suddenly in a non-positive way". Another, for sample Q4-08-20-300, states: "Randomness undermines valence, pleasentness and interest". This statement can mean that when the music sounds too random, it is impossible to evaluate other measures at all. For sample Q3-12-10-300, a participant states: "Unplayable for humans, a machine out of control". This sample was played in a very high tempo, which makes it clear that humans did not compose it, because playing it is not humanly possible.

One common thread in several of the samples is the abrupt endings. Examples of these can be seen in the sheet music in Appendix B; many samples do not have "logical" endings.

Some of the samples perform much better than others, and are described by participants as more convincing, even reminiscent of human composers such as Bach or Debussy. One participant states that "Obvious conotations to classical music makes it more interesting and a little bit more valencing", i.e., connotations to a familiar genre helps the listener to connect more with the music. In sample Q3-12-20-300, a participant says that the sound is like "Debussy in the land of jazz ballads?", possibly implying that the music sounds like a fusion of familiar genres.

In all samples, a common tendency is that there are seemingly random notes inserted here and there, disturbing the flow of the music. For sample Q3-12-20-500, a participant states: "Pretty, but a bit messy at times. The melody is nice, but it disappears a bit in the composure."

Some participants found the valence and arousal measures challenging to use in order to classify all emotions. One participant, for sample Q4-08-20-300, states: "Difficult to evaluate valence for a bittersweet feeling, e.g. melancholy".

Valence and Arousal Scores

Table 6.2 on page 63 displays the participants' average values and standard deviations for valence and arousal. Values marked in bold are the ones which indicate belonging to the intended quadrant, i.e. high valence and low arousal for Q4 and low valence and low energy for Q3 (threshold 2.5).

The valence and arousal average scores can also be seen as a scatterplot in Figure 6.4 on page 63. The background colours of the figure indicate the four emotional quadrants, and the colour of the dots indicate the quadrant to which they were finetuned. The placement of the dot on the X/Y axes indicate the average valence and arousal values, respectively, from participants.

From the scatterplot, it can be seen that only two of the samples ended up as annotated with what would correspond to their original, intended quadrant. However, given that the classification system was only able to distinguish quadrants Q3 and Q4, this is not surprising in terms of the arousal (energy) axis. The composer had no direct input on what constituted as high or low energy in the music. However, there seems to be some correlation between low arousal and low energy, but this is not easy to prove due to sparse data.

6.2. Discussion

6.2.1. Reproducibility In Related Work

RQ1, as presented in Section 1.2, asks: "What are suitable methods for computer-based classification of emotion in music?" This implies a need for benchmark measures that can be used across multiple studies, for comparison in determining suitability of different methods.

While papers and workshops related to musical computational creativity are plentiful, their results and conclusions are often difficult to verify. There are many reasons for this. One is that the concrete models and representations often are left out of a paper, due to various reasons of copyright and privacy.

Another is the inherent non-deterministic nature of evaluating the results of creative work. An AI system may produce different results each time it is run unless explicit preventative steps such as using determined random seeds. The evaluation methods applied may also vary. As seen in the related work discussed in this thesis, the evaluating

Sample	Valence	SD(Valence)	Arousal	SD(Arousal)
Q4-12-20-300	**3.22**	0.971	4.34	0.790
Q4-08-20-300	**3.49**	0.853	**2.75**	0.699
Q4-16-20-300	**3.92**	0.898	4.61	0.616
Q4-12-10-300	Omitted	-	-	-
Q4-12-40-300	**2.77**	1.043	3.48	0.729
Q4-12-20-100	Omitted	-	-	-
Q4-12-20-500	**3.86**	1.105	4.79	0.454
Q3-08-20-300	Omitted	-	-	-
Q3-12-20-300	3.24	0.757	**2.73**	0.807
Q3-16-20-300	**2.49**	0.959	4.23	0.654
Q3-12-10-300	**2.94**	1.109	4.79	0.503
Q3-12-40-300	**2.74**	0.988	**2.13**	0.755
Q3-12-20-100	3.24	0.867	4.06	0.709
Q3-12-20-500	3.53	0.915	3.67	0.828
All Q4 samples	3.45	0.475	3.99	0.861
All Q3 samples	3.03	0.380	3.77	1.013

Table 6.2.: Average scores and standard deviation for valence and arousal for each sample

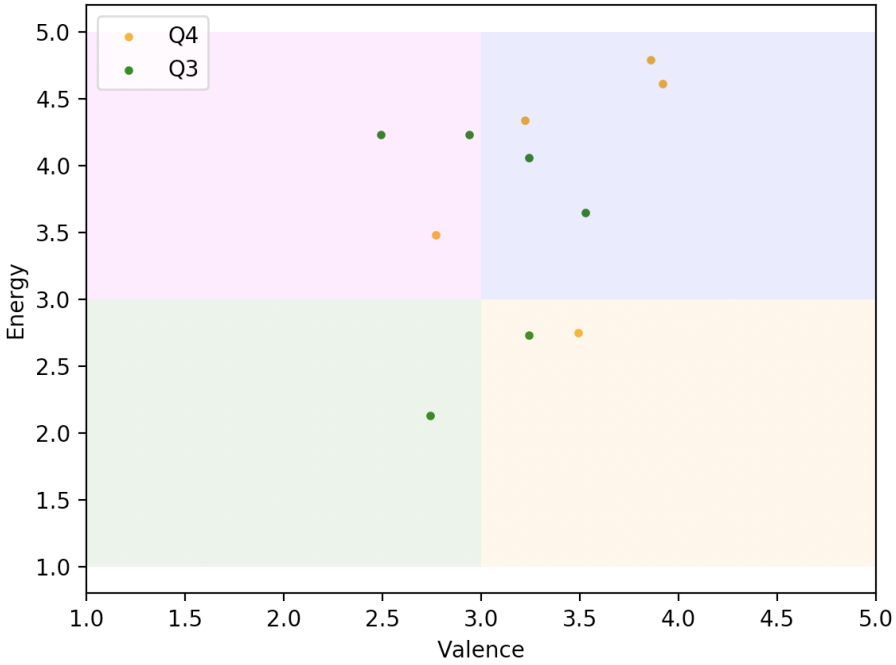

Figure 6.4.: Scatterplot for average values from survey response of valence and arousal scores.

mechanisms almost always contain some domain knowledge provided by the system developers. This domain knowledge may be prone to bias, and therefore the results may not make sense to another reader. Especially in a field as subjective as evaluating a mood or ambience, complete objectivity will be impossible.

Given this, exploring methods used in related work is highly relevant, but cannot be expected to yield equivalent results. There is also a significant lack of benchmarking which methods are genuinely the most effective, as there is an absence of meaningful and measurable empirical results from other systems to which we can compare.

In conclusion, RQ1 becomes difficult to answer due to the fact that results are difficult to verify and reproduce. Also, there is no established "common ground" dataset or task within MER which allows for fair comparison between performance of different systems.

6.2.2. Dataset Comparison for Classification and Composition

RQ3, as presented in Section 1.2, asks: "What are relevant and efficient methods for creating emotion-based computer-generated music, and evaluating it?" In this task, an understanding of emotion in some music dataset must be established. Also, a music dataset is needed for composition of the emotion-based music. This music needs to be written in a format which can be decomposed and used in new combinations for new music, such as the MIDI format.

In the tasks of classification and composition, finding a dataset suited for both missions was unsuccessful. For classification, the dataset needs sufficient size and quality, and high-quality mood annotations. For composition, the data needs to be separable, so that the building blocks of the music can be reused in novel ways to compose new music. As no discovered dataset met these rigorous requirements, two different datasets were used for the two tasks.

The nature of the two datasets varied, in that the dataset used for classification consisted of popular music of all genres, and that the MAESTRO dataset consisted of classical piano music. Therefore, it could be seen that the classification model struggled to predict high energy levels in the MAESTRO dataset, classifying virtually all samples into Q3 or Q4. This tendency is a clear sign of a lack of familiarity with the classical piano music genre and the ability to distinguish music within the dataset. This is not highly surprising, as other genres present in the classification dataset have objectively much higher energy levels.

With this limitation, it is evident that the model cannot produce music which distinguishes lower and higher-energy emotions. However, the model *can* distinguish between quadrants Q3 and Q4, making it possible to produce music differing in the two moods.

6.2.3. Expanded Dataset

A concern in the work within this topic was that available data sets were often too small. While the dataset from Panda et al. (2018) was of high quality, its size was hypothesized too small for the neural network to be able to extract emotion from its audio content. Thus, a larger dataset was acquired, with over 50,000 entries. As seen in the experiment

in Table 5.3 on page 48, the expanded dataset proved inefficient in improving testing accuracy – in fact, accuracy *decreased* from 54% to 52% using the expanded datasets. The loss was much lower, going from 0.11 to 0.036. This probably shows a case of overfitting, where the model has capacities to "memorize" the data without developing abilities to generalize any more than what was achieved with the original dataset.

Using the expanded dataset came with a series of known limitations, as presented in Section 5.4. For one, the dataset was very uneven in its quadrant distribution, whereas the original dataset was evenly distributed. Also, annotations in the dataset were not validated, only automatically calculated. A large portion of the dataset had a computed quadrant probability of less than 0.5. Finally, a fifth quadrant was present in the dataset, Q0. The intended use for the quadrant Q0 was unclear, but may be an expression for when no quadrant was found.

With all these limitations listed, it is evident that the dataset did not have the sufficient qualities required for successful training. It is possible that a differing network configuration, more resilient toward the dataset's imbalances, may have mitigated this, but this was not explored further in this thesis. Overall, it seems that a large dataset of lower quality does not yield higher performance than a smaller dataset of higher quality.

6.2.4. Metadata Incorporation

Experiment 5.5 incorporated genre metadata as part of the network's input, along with the sound data. Figure 6.5 displays the development of loss over 100 epochs of training. The figure shows that the loss over the training datasets develops similarly for the first roughly 20 epochs. However, following the blue line, loss proves more volatile and eventually comes to rest at a much higher rate than without the metadata as part of the input. The loss for the original method, without metadata, converges at around 0.9, and loss for the network using metadata converges at around 2.0, averaged over 10 runs.

Figure 6.6 displays accuracy for the testing set for networks with and without using metadata as input. The development is somewhat similar to the loss function. Accuracy is low for the first 10-20 epochs, but eventually stabilizes after roughly 40-60 epochs. Eventually, the model using metadata as input stabilizes on a testing accuracy of around 45%, while the network not using metadata stabilizes on roughly 50%. Overall, it can be concluded that the inclusion of one-hot-vector encoded genre metadata as a part of the network's input does not improve performance.

The central aspect to note with regards to the model's lower performance is that the direct inclusion of metadata causes the final 23 input tensors to represent something entirely different than the first 32,000, which represent audio signals. This input implies that we expect the network to discover that this is an entirely different representation on its own.

Another important aspect of this inclusion of metadata is that the distribution of values is very different from those of audio signals. Genre information is represented by 0 or 1 in a one-hot vector, whereas audio signals are floating-point numbers virtually never 0 (except when no signal is received) or 1. Thus, the genre information values deviate significantly from the values used in the rest of the input.

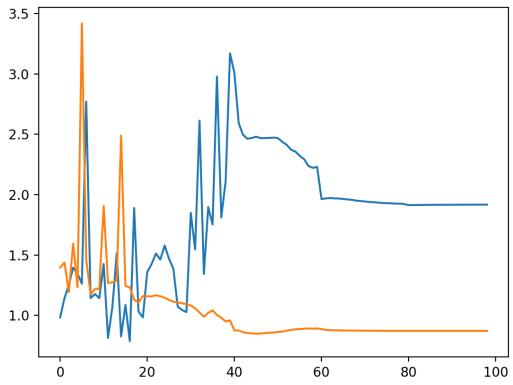

Figure 6.5.: Loss function, using (blue line) and not using (orange line) metadata.

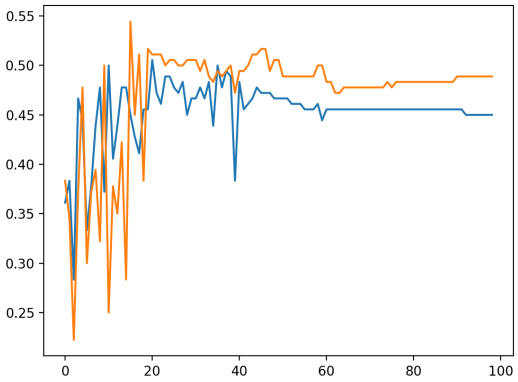

Figure 6.6.: Test accuracy, using (blue line) and not using (orange line) metadata.

While in theory a deep neural network can have the ability to learn its own notion of what its input represents, this may be too little information to enable this kind of learning. Moreover, the direct inclusion of metadata into the network along with audio signals may overall be an unsuited way of including metadata.

6.2.5. Indicating Degree of Classification Correctness

RQ2, as presented in Section 1.2, asks: "What are sets of emotion categories that are comprehensible and effective for machine learning use?". In the experiments performed, only the four emotion quadrants Q1-Q4 are used. However, a song's position within a quadrant implies that it is positioned somewhere in the X/Y plane. In the dataset used for classification, this X/Y position was not known. Contact was made with the authors of the dataset (Panda et al., 2018), and the information did exist, but the authors could not find the time to provide the data. This means that many samples may have been very close to an adjacent quadrant, but there was no way to express this. In an optimal system, if that adjacent quadrant is predicted, the error should be considered less severe than other predictions. In fact, if the quadrant predicted was correct, but the X/Y measures were far away from each other although in the same quadrant, perhaps some error should be enforced as well.

One possible solution may have been viable in the effort towards finding partial correctness in samples. The Spotify Web API for music feature metadata does provide this information on valence and arousal. As Section 5.6.1 concluded, the features *energy* and *valence* proved most indicative of position in the VA plane. However, as these two data sources often indicate different quadrants for the same sample, it is an imperfect solution.

The following setup could have been used for indicating "partial correctness" in a classification:

```
if sample.classified_quadrant == sample.annotated_quadrant:
    correct = 1
else:
    close_enough_energy =
          sample.energy < annotated_quadrant.max_energy*1.1
       or sample.energy > annotated_quadrant.min_energy*0.9
    close_enough_valence =
          sample.valence < annotated_quadrant.max_valence*1.1
       or sample.valence > annotated_quadrant.min_valence*0.9

    if close_enough_energy and close_enough_valence:
        correct = 0.5
    else:
        correct = 0
```

Using this setup, the annotation is rewarded `0.5` correctness if both energy and valence are within 10% of the maximum values of valence and arousal belonging to the correct

quadrant. The goal of such a setup would be to use this partial correctness in order to punish those classifications less in the training process, causing the model to stay closer to its partially correct predictions in future iterations. Also, tests should be performed regarding which measures should be used as acceptable distance from the annotated quadrant, as well as measures for partial correctness reward.

As a conclusion, the quadrant annotation, hypothesized to be an effective taxonomy for classification, may give too little information on the emotional content of the music. Moreover, the lack of X/Y position information in the valence/arousal plane inhibits the opportunity to reward partial correctness.

6.2.6. Discussions with Spotify engineers

In an effort to acquire more information on emotions conveyed in music, the Spotify feature analysis tool was used in the experiment presented in Section 5.6. The feature analysis tool provided, among other data points, valence and energy measures which together could form an X/Y position in the valence/arousal plane.

An issue with the Spotify feature analysis is its lack of documentation on how the features are produced. The quality of the feature analysis appears high and coherent on samples tested, but its use is difficult to verify and justify when the system is non-transparent. Fortunately, contact was established with Jussi Karlgren, adjunct professor at Kungliga Tekniska Högskolan (KTH) and researcher at Spotify. The conversations that followed were not structured interviews, but rather a dialogue on Music Emotion Recognition (MER) in general and at Spotify, the ways such systems are built at Spotify and what difficulties they have met in this development.

Karlgren describes music recommendation work at Spotify as split mainly into two segments. One is the social aspect of listening, namely that when you have similar listening habits to another user, you may get recommendations based on other songs that the other user listens to, but you may not have discovered yet. This method is highly efficient in that it exploits the users' own listening habits for recommendation, providing the "human touch" that recommender algorithms often may lack. The other aspect of music recommendation is what is called the **cold start problem**. In Spotify's roster of millions of songs, many of them have almost never been listened to and therefore cannot be used in social recommendation. A famous artist may have an advantage when publishing new music, given that the artist is already familiar. Thus, that information can be used for recommendation to users deemed likely to enjoy that artist's music. However, small or independent artists are more likely to go unnoticed. Here, Spotify depends on systems which classify songs based purely on audio content, a similar topic to the one addressed in this thesis. It is the cold start problem, and the analysis tools developed for this purpose at Spotify, which were the main topics of discussion with Karlgren.

With regards to the question on documentation of the analysis tools for the cold start problem, information on how features such as valence and energy are computed was superficial or lacking. In discussion with Karlgren, three main hypotheses were presented and later researched for evidence. The following section describes these hypotheses.

- **Analysis tools are valuable to the business.** Spotify is not just a music listening service. It is also widely known for its high-quality recommendation algorithms. If Spotify's methods of music classification and recommendation were publicly available, it might jeopardize Spotify's competitive advantage.
- **Analysis tools may not have been documented upon creation.** There are many possible reasons for this. First, many systems from Spotify are developed using test-driven development, which creates a form of implicit documentation which would be difficult to publish. Second, the people who were most familiar with the systems may not be working at Spotify anymore, as the company has seen great growth in the last years. Spotify is probably not alone in being reliant on their workforce being their direct source of documentation, which then can become a problem upon workforce turnover through the years.
- **Documentation may not be necessary when you feel certain that it works.** As a commercial company, the quality of a certain tool within feature analysis can, in some regards, be measured in its success when meeting the users. When the recommendation system is effective in getting the user to listen to more, and more diverse, music, it can be considered a success. On the other hand, if an aspect of the system proved ineffective in reaching such user-related goals, it would have been discarded or improved upon. Thus, its very existence and availability may serve as evidence that it should be considered accurate in its predictions.

After discussing these hypotheses, contact was made with Simon Durand and Ching Sung, research scientist and product manager, respectively, at Spotify working in the team building audio analysis features. Discussion with them provided some more insight into how the features were developed.

One main aspect to note is that the audio features addressed in this chapter were designed by a music intelligence company called The Echo Nest, which was acquired by Spotify in 2014. According to Durand, these inner workings of the features were not documented clearly. However, some more high-level explanations were available. Here follows the direct explanation from Durand for the energy and valence features.

- "**Energy** tries to convey how energetic a song is. It is a linear combination of 20 features extracted from the overall analysis of a song, like average note duration. It is trained on human judgement to rank 5 songs (among about 1000) from more energetic to less energetic.
- **Valence** is the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). It is a feed-forward Neural Network applied on the 2d fast Fourier transform of short audio windows, then averaged. It was trained on human judgements."

The experimental method for extracting Valence and Energy measures on the basis of human judgment are described as follows:

- "A dataset of 1000 songs was selected (the 1000 songs were probably randomly drawn among the Echo Nest database but unfortunately I don't have more information about the dataset composition)
- 5 examples are drawn randomly among those 1000 songs and constitutes a set. This is done many times.
- Each of these sets of 5 examples is being shown to human participants who are asked to ranked the 5 songs according to an audio attribute (eg: rank from highest to lowest energy)
- The audio attribute model is also shown those sets of 5 examples and is trained to output a measure of an attributes that can rank them such that the Kendall's tau coefficient between the prediction and the human judgement is minimized. The Kendall's tau coefficient measures the similarity of two rankings by counting how many swaps you have to do to align one list to the other"

While this sheds important light on the testing methods, several aspects of the testing remain unclear. The most significant issue is that the dataset composition is unclear, meaning that it may be biased towards specific kinds of music. If the music was drawn randomly from a large database, it might be biased towards whether the music database is distributed evenly within all kinds of music. The ranking method used is an efficient way to rank several samples against each other. However, it naturally takes on the assumption that all songs can be ranked towards each other. For example, if a test is between four very calm songs and one very energetic, they would still be ranked in order where all samples are seen to be equidistant to one another. To mitigate this flaw, one method is to ensure a high enough number of these rankings for each song. However, to what extent this is done is not clear.

In the end, it is difficult to conclude whether this method is strong enough to use. However, it is clear that human judgment is deemed necessary in the development of these features, and this method may be a reasonable choice in the trade-off with resource demand. It seems that Spotify developers comfortably rely on the fact that their recommendation algorithms, which utilize these features, are popular and considered accurate by their user base. In such a regard, the features, in combination with Spotify's other recommendation systems, should be considered as successful, although no scientific conclusion can be made to why this is, in this thesis.

6.2.7. Composition Quality and Recommended Composition Setup

The goal for this thesis overall, as presented in Section 1.2, was to "Classify emotion in music and use the classifications in automatic music composition". The survey conducted gives an evaluation of whether the music composed does indeed convey the emotions intended by the system, by giving valence and arousal values for music samples. This implicitly results in a quadrant, to which we can compare with the intended quadrant used in composition.

The music composed with a given emotion as input, presented in Section 6.1.3, was not recognised by survey participants as consistent in adhering to the intended emotion quadrant. However, there are differences in the composition results that may indicate that the classification was effective to some extent. On average, the music composed based on Q3 (low arousal, low energy) annotations are rated by survey participants with lower valence and arousal values than the Q4 (high arousal, low energy) compositions. This can be seen most clearly by the visualized survey results in Figure 6.4 on page 63. Therefore, a conclusion here is that the music produced with the intent on low arousal is on average scored with lower arousal than the music produced with the intent of high arousal. However, the results are not consistent, and further testing should be performed in order to increase the certainty that this is indeed the case.

Another interesting matter here is a strange phenomenon best seen when studying the sheet music. Overall, there is an over-representation of very short notes and pauses. Even when a longer note is heard, it is often the joining of several short notes in the sheet music. As the system used is an existing architecture, it is difficult to understand why this is. However, it is natural that participants connect short, rapidly changing notes as "high-energy". This may be part of the explanation as to why the energy is mostly rated as very high for many samples.

RQ3 asks for relevant and efficient methods for creating emotion-based computer-generated music. In using the Pop Music Transformer, many different experimental setups were used, in an attempt to understand which parameters had an impact on both music quality and whether the music conveyed the intended emotion.

With regards to what is the most suitable experimentation setup, no clear tendency was found as to which configuration of the composer created the "best" result. However, some conclusions can be made from the results produced.

First and foremost, too little finetuning data (10 samples) or too few epochs of training (100 epochs) most often led to very unpleasant music samples. Sample Q4-12-10-300 (using 10 finetuning samples) and Q4-12-20-100 (training for 100 epochs) were discarded due to their very low quality and coherence. The corresponding configurations for Q3, namely Q3-12-10-300, was rated with one of the lowest scores for pleasantness and high randomness, as well as being described by participants as too fast and incoherent. Q3-12-20-100 also scored below average for pleasantness, but was overall better received. Thus, a conclusion here is that for future experimentation, a small amount of finetuning data is not recommended.

With regards to the experiments on including a more substantial amount of training data or number of training epochs, no direct conclusion can be drawn as to what facilitates a clearly better result. The samples rated as most pleasant were Q4-08-20-300, Q4-12-40-300 and Q3-12-40-300. One indication may be that a larger amount of finetuning samples lead to a more pleasant result. However, more samples would need testing in order to verify this.

6.2.8. Survey Strengths and Weaknesses

The conducting of a survey to evaluate the results in this thesis has some strength and weaknesses, discussed in this section. A primary strength of the survey is its ability to gather information on people's subjective experiences in a structured manner. It makes the participants adhere to the same type of language in expressing opinions on the produced music. Also, the participants can make comments if they wish to express themselves outside of the structured language. Combined, this gives a combination of quantitative and qualitative data which is suited for the preliminary understanding of the quality of the system produced.

There are, however, several weaknesses to which the survey is prone. First and foremost, the survey was conducted digitally and online. This method means that the participants have no way of asking questions if anything is unclear. No assurance can be made that the participants understand the language used in the survey in the same way.

Another weakness is found in the group of the participants and their relationship with the author. Due to the COVID-19 outbreak, which took place in the spring of 2020, at the time of writing, no physical experiments could be conducted. This also severely limited the people that could be reached with other methods of contact, such as asking people on the street or at the school campus. Thus, the participants asked directly to participate were all in a familiar, friendly or professional relationship to the author. This can introduce several forms of bias. One is that the participants give overly positive answers (on factors such as pleasantness), to show their support of the work. On the other hand, this may be a factor that is known to the participants, for which they may try to overcompensate. This issue may introduce another bias, i.e. the participants giving overly negative answers, trying to avoid the original bias.

Attempts were made to mitigate these biases by encouraging participants to ask friends or colleagues of their own to participate in the survey. This removes the direct relationship with the author, but may still be a source of similar familiarity biases.

In the participant group, over half of the participants stated that they play one or more instruments, indicating that this group overall has a strong relationship to and understanding of music. This number is a clear over-representation as to what could be expected of a more diverse group, as it reflects the author's personal network where there is a large number of musicians. It can also be a sign that people who have a personal interest in music felt more compelled to answer the survey than the more casual consumer.

A third possible weakness is in the fact that only two of the four quadrants were used in the survey, because the classifier only could predict songs from the MAESTRO dataset into quadrants Q3 and Q4. A possible bias here is that the participants may expect that all quadrants should be evenly present in the survey, and answers will be adjusted so that all quadrants are present there as well.

The main attempt to mitigate this weakness was to ask the participants to rank the music on scales of valence and energy, instead of asking the participants to place each sample in one of the quadrants. This creates more options for the participants than just the four categories, possibly reducing the participants' bias towards giving an even

distribution of answers. However, this cannot be validated, as no direct conversation was had with the participants.

In presenting the survey, samples were not presented to the participants in a random order, due to technical difficulties with the online survey tool. This lead to survey responses which did not evaluate all samples, causing the final samples to be evaluated fewer times. 19 participants skipped the final sample in the survey. This can be a sign of fatigue bias with the participants (i.e. the deteriorating effort put in by participants in later sections of the survey).

6.2.9. Composition: Should all results be good?

In the produced music, quality and perceived coherence vary greatly. Some pieces are enjoyable, while others feel "random" or unpleasant in other ways.

In a sense, this is not so different from the work of a musician, as no musician is expected to produce exclusively great compositions, or even compositions liked by all people. Ideas, themes and melodies must be hand-picked and combined in order to produce pleasant results, often to some defined audience. Thus, one use of the system created in this thesis is just that: the facilitation of ideas and creativity. This could either be in the input of a mood, and receiving a new piece from scratch upon which to build, or inputting one's own melody and creating a continuation.

7. Conclusion and Future Work

From Chapter 1, the goal of this thesis was to classify emotion in music and use the classifications in automatic music composition. This goal has been addressed by the exploration and answering of three main research questions. This section summarizes contributions made as introduced in Section 1.4 on page 3, and finally, to which extent the goal has been met.

In this Master's thesis, a structured literary review has been performed, reviewing the state-of-the-art within musical computational creativity. From this, experiments have been conducted in order to find relevant machine learning methods and data sources for the understanding of mood in a popular music data set.

As presented in Section 4.1, a neural network architecture has been produced, in a fully functional pipeline between the classification of mood in music, and the composition of new and novel music based on mood-annotated music data. Test accuracy in classification lies around the 50% mark, whereas random guesses would perform at around 25%. This performance could have been improved by two main factors: A larger dataset, as the system only had 900 samples to work on, and a better indication of each sample's position in the emotional X/Y plane.

Answering **Research Question 1 (RQ1)**, it seems that a deep neural network classifying on raw musical data is a viable method for classifying mood in music data. This method has the significant advantage that little musical expert knowledge is required, and that the network itself is given the task of distinguishing both high- and low-level features.

RQ2 asks what suitable sets of emotion/moods categories are. In this thesis, I conclude that the X/Y axis is a sufficient method for understanding mood in music in such a way that it can be used to compose new music. However, it has also been made clear that this choice cannot be used universally on all genres on music. The classical piano genre can be categorized on such a set of axes, but if it is combined with a much louder genre such as rock, no classical piano music would ever be considered energetic. This indicates that universal measures for all music may not fit in such a taxonomy, but they can be very useful within a genre.

RQ3 asks for methods for the production and elevation of mood-based computer-generated music. In this thesis, several methods of compositions have been compared, with a focus on instrumental music with a limited number of instruments. The most significant find is that the element of relative attention is essential to composing music that feels both novel and "human". Recurrence of themes and melodies contribute to a song feeling complete.

The element of relative attention has been used in the composer used in this thesis.

The compositions were produced with emotion-annotated data as a "finetuning" input, training specifically on the emotion-annotated samples in order to learn elements which distinguish that data from the entire data corpus. The compositions produced vary in quality. Some are pleasant and seem to cohere to the input mood, while others feel random. However, there seems to be evidence of structure and repetition within the music, and thus the goal of self-attention is to some extent achieved.

A survey was performed in order to test whether the emotional "intent" was indeed conveyed in the produced music. Results from the survey indicated that high-arousal emotion was generally well-reflected in the composed music, and low-arousal emotion was on average rated as lower-arousal, however not within the desired thresholds required in order to successfully place the composed music in the originally desired emotion quadrant.

7.1. Future Work

Expanding datasets containing raw musical data.

An important step in the progress of music emotion recognition would be to develop larger, yet still publicly available, datasets containing the actual music. In this sense, the most significant part of the problem is probably the issue of intellectual property and licensing. However, expanded availability on even short samples of songs could be of great help in understanding the relationship between music and emotion.

Data for both classification and composition.

The first inhibiting factor of this work has been the accessibility of data suited for both the purpose of composition and of classification. A significant step towards smarter music composition would be to develop a dataset which both carries mood annotations and the music in a file format which can be deconstructed and reconstructed for composition.

Measures for partial correctness.

With the strict limits of the emotional quadrants, a measure for partial correctness should be developed in order to avoid suppressing "almost correct" solutions just as much as the completely incorrect solutions. This means working on annotations in the X/Y plane, and developing a more "fuzzy" loss function.

Separate networks for audio and metadata.

A measure that may improve classification accuracy is a system with the ability to combine audio data and metadata. A small experiment was conducted in this thesis, combining the two types of data in the same network. However, as the data types are very different, a possibility for the processing of metadata could be employing it in a different network structure entirely and combining the results to ensure that it is not treated as the same type of information as audio signals.

Further testing for significant survey results.

In the survey conducted for this thesis, 101 participants evaluated 11 samples, one for

each training configuration. However, one training configuration could produce a massive variety of different samples. Thus, further testing on the model using a larger amount of samples for each training configuration would give a more representative image of what the impact of each training configuration truly is. Moreover, similar surveys should be performed on more diverse participant groups in order to reduce familiarity biases.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, nov 2016. USENIX Association. ISBN 978-1-931971-33-1. URL `https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi`.

AIVA. About AIVA. `https://www.aiva.ai/about#about`, 2019. [Online; accessed 11-November-2019].

AIVA Technologies. How we used our Music Engine to create the first AI-generated album of Chinese Music. `https://medium.com/@aivatech/how-we-used-our-music-engine-to-create-the-first-ai-generated-album-of-chinese-music-9d6fa984b4e8`, 2018. [Online; accessed 11-November-2019].

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

J. Bai, K. Luo, J. Peng, J. Shi, Y. Wu, L. Feng, J. Li, and Y. Wang. Music emotions recognition by cognitive classification methodologies. In *2017 IEEE 16th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 121–129, July 2017. doi: 10.1109/ICCI-CC.2017.8109740.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Thierry Bertin-Mahieux, Daniel Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 591–596, 01 2011.

Margaret A. Boden. Creativity and artificial intelligence. *Artificial Intelligence*, 103(1-2): 347–356, August 1998. ISSN 0004-3702. doi: 10.1016/S0004-3702(98)00055-1. URL `http://dx.doi.org/10.1016/S0004-3702(98)00055-1`.

Andrew Botros, John Smith, and Joe Wolfe. The Virtual Boehm Flute – A web service that predicts multiphonics, microtones and alternative fingerings. *Acoustics Australia*, 30(2):61–66, 2002.

Bozhidar Bozhanov. Computoser - rule-based, probability-driven algorithmic music composition. *Computing Research Repository (CoRR)*, abs/1412.3079, 2014. URL `http://arxiv.org/abs/1412.3079`.

Catherine B Bruch. Assessment of creativity in culturally different children. *Gifted Child Quarterly*, 19(2):164–174, 1975.

Zehra Cataltepe, Yusuf Yaslan, and Abdullah Sonmez. Music genre classification using midi and audio features. *EURASIP Journal on Advances in Signal Processing*, 2007: 1–8, 01 2007. doi: 10.1155/2007/36409.

Òscar Celma, Perfecto Herrera, and Xavier Serra. Bridging the music semantic gap. *Extended Semantic Web Conference (ESWC) 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation*, 2006.

P. Chen, L. Zhao, Z. Xin, Y. Qiang, M. Zhang, and T. Li. A scheme of midi music emotion classification based on fuzzy theme extraction and neural network. In *2016 12th International Conference on Computational Intelligence and Security (CIS)*, pages 323–326, Dec 2016. doi: 10.1109/CIS.2016.0079.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995. Springer.

Alan S. Cowen, Xia Fang, Disa Sauter, and Dacher Keltner. *What music makes us feel: At least 13 dimensions organize subjective experiences associated with music across different cultures.* National Academy of Sciences, 2020. doi: 10.1073/pnas.1910704117. URL `https://www.pnas.org/content/early/2020/01/01/1910704117`.

Darren W Dahl and Page Moreau. The influence and value of analogical thinking during new product ideation. *Journal of marketing research*, 39(1):47–60, 2002.

Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425. IEEE, 2017.

Roger Dannenberg. Computer Models of Musical Creativity, MIT Press (2005). *Artificial Intelligence*, 170:1218–1221, 12 2006. doi: 10.1016/j.artint.2006.10.004.

Leon Derczynski. Complementarity, f-score, and NLP evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

Chris Donahue, Huanru Henry Mao, and Julian McAuley. The NES Music Database: A multi-instrumental dataset with expressive performance attributes. In *Proceedings of the International Society for Music Information Retrieval (ISMIR) 2018*, pages 475–482, 2018.

Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W Cottrell, and Julian McAuley. LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. *arXiv preprint arXiv:1907.04868*, 2019. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR 2019)*.

Stuart Dreyfus. The computational solution of optimal control problems with time lag. *IEEE Transactions on Automatic Control*, 18(4):383–385, 1973.

Paul Ekman. What scientists who study emotion agree about. *Perspectives on Psychological Science*, 11(1):31–34, 2016. doi: 10.1177/1745691615596992. URL `https://doi.org/10.1177/1745691615596992`. PMID: 26817724.

Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sy8XvGb0-`.

Paul R. Farnsworth. The social psychology of music. *Journal of Aesthetics and Art Criticism*, 17(1):133–133, 1958. doi: 10.2307/428031.

Lucas N. Ferreira and Jim Whitehead. Learning to generate music with sentiment. In *Proceedings of the Conference of the International Society for Music Information Retrieval*, ISMIR 2019, 2019.

R. Fox and Adil Haleem Khan. Artificial intelligence approaches to music composition. In *The International Conference on Artificial Intelligence (ICAI) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).*, Northern Kentucky University, 2013.

A. Freitas and Frederico Guimarães. Melody harmonization in evolutionary music using multiobjective genetic algorithms. In *Proceedings of the Sound and Music Computing Conference*, January 2011a.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. Pages 200–220.

Kazjon Grace and Mary Lou Maher. Expectation-based models of novelty for evaluating computational creativity. In Tony Veale and F. Amílcar Cardoso, editors, *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems*, pages 195–209. Springer International Publishing, Cham, 2019. ISBN 978-3-319-43610-4. doi: 10.1007/978-3-319-43610-4_9. URL `https://doi.org/10.1007/978-3-319-43610-4_9`.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation, 2019.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. *arXiv preprint arXiv:1810.12247*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

L. A. Hiller, Jr. and L. M. Isaacson. Musical composition with a high-speed digital computer. *Journal of the Audio Engineering Society*, 6(3):154–160, 1958. URL `http://www.aes.org/e-lib/browse.cfm?elib=231`.

John C Houtz, Edwin Selby, Giselle B Esquivel, Ruth A Okoye, Kristen M Peters, and Donald J Treffinger. Creativity styles and personal type. *Creativity Research Journal*, 15(4):321–330, 2003.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew Dai, Matt Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations (ICLR)*, 2019. URL `https://arxiv.org/abs/1809.04281`.

Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Generating music with rhythm and harmony, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 2015.

Tony Jebara. *Machine learning: discriminative and generative*, volume 755. Springer Science & Business Media, 2012.

Sean F Johnston. A cultural history of the hologram. *Leonardo*, 41(3):223–229, 2008.

A. Kartikay, H. Ganesan, and V. M. Ladwani. Classification of music into moods using musical features. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, volume 3, pages 1–5, Coimbatore, India, Aug 2016. doi: 10.1109/INVENTIVE.2016.7830197.

Nikhil Ketkar. *Introduction to PyTorch*, pages 195–208. Apress, Berkeley, CA, 2017a. ISBN 978-1-4842-2766-4. doi: 10.1007/978-1-4842-2766-4_12. URL `https://doi.org/10.1007/978-1-4842-2766-4_12`.

Nikhil Ketkar. *Introduction to Keras*, pages 97–111. Apress, Berkeley, CA, 2017b. ISBN 978-1-4842-2766-4. doi: 10.1007/978-1-4842-2766-4_7. URL https://doi.org/10.1007/978-1-4842-2766-4_7.

Shahin Khobahi and Mojtaba Soltanalian. Model-Aware Deep Architectures for One-Bit Compressive Variational Autoencoding. *arXiv preprint arXiv:1911.12410*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. An Introduction to Variational Autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

Anders Kofod-Petersen. How to do a Structured Literature Review in Computer Science. https://research.idi.ntnu.no/aimasters/files/SLR_HowTo2018.pdf, 2018. [Online; accessed 24-November-2019].

Zheng Cong Koh, Jin Hong Yong, and Jin Yi Yong. Re-implementation of music transformer: Generating music with long term structure, 2019. URL https://github.com/COMP6248-Reproducability-Challenge/music-transformer-comp6248. [Online; accessed 24-November-2019].

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation Applied to Handwritten ZIP Code Recognition. *Neural computation*, 1(4):541–551, 1989.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006. Pages 50–51.

Colby N Leider. *Digital audio workstation*. McGraw-Hill, Inc., 2004.

Tao Li and Mitsunori Ogihara. Detecting emotion in music. *Proceedings of the International Society for Music Information Retrieval (ISMIR) 2003; 4th Int. Symp. Music Information Retrieval*, 2003:1–2, 11 2003.

Yong Li, Yang Fu, Hui Li, and Si-Wen Zhang. The improved training algorithm of back propagation neural network with self-adaptive learning rate. In *2009 International Conference on Computational Intelligence and Natural Computing*, volume 1, pages 73–76. IEEE, 2009.

C. Lin, M. Liu, W. Hsiung, and J. Jhang. Music emotion recognition based on two-level support vector classification. In *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 375–389, July 2016. doi: 10.1109/ICMLC.2016.7860930.

Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *Proc. ISMIR 2003; 4th Int. Symp. Music Information Retrieval*, January 2003.

Xin Liu, Qingcai Chen, Xiangping Wu, Yan Liu, and Yang Liu. CNN Based Music Emotion Classification. *arXiv preprint arXiv:1704.05665*, April 2017.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

Todd I Lubart. Models of the creative process: Past, present and future. *Creativity research journal*, 13(3-4):295–308, 2001.

Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving Text Classification by Shrinkage in a Hierarchy of Classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL `http://dl.acm.org/citation.cfm?id=645527.657461`.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Melodrive. About melodrive. `http://melodrive.com/index.php#about`, 2019. [Online; accessed 11-November-2019].

Marvin Minsky and Seymour A Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT press, 2017. Pages 227–246.

S. Mo and J. Niu. A Novel Method Based on OMPGW Method for Feature Extraction in Automatic Music Mood Classification. *IEEE Transactions on Affective Computing*, 10(3):313–324, July 2019. ISSN 2371-9850. doi: 10.1109/TAFFC.2017.2724515.

Olav Andreas E Olseng. An application of evolutionary algorithms to music: - co-evolving melodies and harmonization. *Master's Thesis, Norwegian University of Science and Technology, Faculty of Information Technology and Electrical Engineering, Department of Computer Science*, page 47, June 2016.

Stephen E. Palmer, Karen B. Schloss, Zoe Xu, and Lilia R. Prado-León. Music–color associations are mediated by emotion. *Proceedings of the National Academy of Sciences*, 110(22):8836–8841, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1212562110. URL `https://www.pnas.org/content/110/22/8836`.

Davis Yen Pan. Digital audio compression. *Digital Technical Journal*, 5(2):28–40, 1993.

Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 9:240–254, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch. *Proceedings of Neural Information Processing Systems*, pages 1–4, 2017.

Ashish Kumar Patel, Satyendra Singh Chouhan, and Rajdeep Niyogi. Using crowd sourced data for music mood classification. In Anirban Mondal, Himanshu Gupta, Jaideep Srivastava, P. Krishna Reddy, and D.V.L.N. Somayajulu, editors, *Big Data Analytics*, pages 363–375, Cham, 2018. Springer International Publishing. ISBN 978-3-030-04780-1.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

Trevor Pinch and Frank Trocco. The Social Construction of the Early Electronic Music Synthesizer. *Icon*, 4:9–31, 1998. ISSN 13618113. URL `http://www.jstor.org/stable/23785956`.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*, 2018.

Nurlaila Rosli, Nordiana Rajaee, and David Bong. Non Negative Matrix Factorization for Music Emotion Classification. In Ping Jack Soh, Wai Lok Woo, Hamzah Asyrani Sulaiman, Mohd Azlishah Othman, and Mohd Shakir Saat, editors, *Advances in Machine Learning and Signal Processing*, pages 175–185, Cham, 2016. Springer International Publishing. ISBN 978-3-319-32213-1.

Jon Rowe and Derek Partridge. Creativity: A survey of AI approaches. *Artificial Intelligence Review*, 7(1):43–70, 1993.

Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *arXiv preprint arXiv:1402.1128*, 2014.

Rob Saunders. Towards Autonomous Creative Systems: A Computational Approach. *Cognitive Computation*, 4(3):216–225, 2012.

Jürgen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural networks*, 61:85–117, 2015.

Nandini Sengupta, Md Sahidullah, and Goutam Saha. Lung sound classification using cepstral-based statistical features. *Computers in biology and medicine*, 75:118–129, 2016.

Yeong-Seok Seo and Jun-Ho Huh. Automatic Emotion-Based Music Classification for Supporting Intelligent IoT Applications. *Electronics*, 8:164, 02 2019. doi: 10.3390/electronics8020164.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint 1409.1556*, September 2014.

M Soleymani, A Aljanaki, and YH Yang. DEAM: MediaEval Database for Emotional Analysis in Music. *PloS One*, 12(3), 2017.

Stephen V Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89, 1997.

Hope R Strayer. From neumes to notes: The evolution of music notation. *Musical Offerings*, 4(1), 2013.

Robert E. Thayer. *The Biopsychology of Mood and Arousal*. Oxford University Press USA, New York, 1990. Pages 15–20.

Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards Musical Query-by-Semantic-Description Using the CAL500 Data Set. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 439–446, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277817. URL `https://doi.org/10.1145/1277741.1277817`.

Pedro F. Vale. The role of artist and genre on music emotion recognition. *Master's thesis, Repositório da Universidade Nova de Lisboa, NIMS - Dissertações de Mestrado em Gestão da Informação*, 2017.

Pradnya A Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE, 2016.

Kelly L. Whiteford, Karen B. Schloss, Nathaniel E. Helwig, and Stephen E. Palmer. Color, music, and emotion: Bach to the blues. *i-Perception*, 9(6):2041669518808535, 2018. doi: 10.1177/2041669518808535. URL `https://doi.org/10.1177/2041669518808535`.

Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.709.9164&rep=rep1&type=pdf`, 2014. [Online; accessed 24-November-2019].

Uwe Wolfradt and Jean E Pretz. Individual differences in creativity: Personality, story writing, and hobbies. *European Journal of Personality*, 15(4):297–310, 2001.

Yi-Hsuan Yang, Chia-Chu Liu, and Homer H. Chen. Music emotion classification: A fuzzy approach. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, page 81–84, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595934472. doi: 10.1145/1180639.1180665. URL `https://doi.org/10.1145/1180639.1180665`.

Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

Andreas Zell, Niels Mache, Ralf Huebner, Günter Mamier, Michael Vogt, Michael Schmalzl, and Kai-Uwe Herrmann. SNNS (Stuttgart Neural Network Simulator). In *Neural Network Simulation Environments*, pages 165–186. Springer, 1994.

A. Structured Literature Review (SLR) Protocol

A.1. Introduction

This SLR Protocol was developed during the fall of 2019 as a part of the Master's Thesis spanning from the fall of 2019 to the spring of 2020.

This Master's Thesis revolves around the field of computational musical creativity, and in particular the understanding and synthesizing music in specific moods and genres. The purpose of the SLR is to discover existing related work, to uncover performance of various solutions, and to identify points of further work where this Master's Thesis can make a contribution.

A.2. Research Questions

The research questions can be found in Section 1.2.

A.3. Search Strategy

The search engine used for the review is Google Scholar. This tool aggregates results from other domains, lists these domains clearly and also has useful functions for searching for synonyms or filtering by citations. Other domains considered were SpringerLink, ACM, IEEE, and ResearchGate.

The set of search terms is defined with regards to **RQ1**. Some terms are split into groups where terms in one group are synonyms or have similar semantic meaning. The search is conducted with boolean notation, using **OR** notation for synonyms and **AND** notation to concatenate the different search term groups. The search terms are found in table A.1.

The search string is represented as follows, where G indicates the row and T indicates the column:

([G1, T1 OR G1, T2] AND [G2,T1 OR G2, T2 OR G2, T3]
AND [G3, T1 OR G3, T2] AND [G4, T1 OR G4, T2])

All results produced by using the search string are collected and reduced by removing duplicate papers or papers published from multiple sources.

Table A.1.: Search terms and groups

	Term 1	Term 2	Term 3
Group 1	Music	Musical	
Group 2	Mood	Ambiance	Emotion
Group 3	Classification	Detection	
Group 4	Artificial Intelligence	Machine Learning	

Table A.2.: Number of search results for each publishing year 2010-2019

Year	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
# results	1580	1780	2870	2640	3000	3420	3960	4780	5560	3870

A.3.1. Search term limitations

In researching a evolving and growing field, it is natural to impose a temporal limit in order to examine as new and relevant studies as possible. With the selected search term, the number of papers published has increased every year (see table A.2). In Google Scholar, for this search term, 16,500 papers have in total been published in or before 2015, and 18,170 have been published in 2016 or later. Thus, the limit for the search was set to articles written in 2016 or later in order to roughly halve the amount of search results. However, older articles could be included if they were a part of the Starting set.

Naturally, exploration of thousands of articles is impossible within the scope of a Master's Thesis. The top 100 articles ranked "most relevant" on Google Scholar were used as the foundation for the SLR quality assessments, in addition to the given articles from the Starting set. Note that Google Scholar does not provide documentation for its ranking algorithm. This may be a cause of possible bias and oversight of important articles. The inclusion of a Starting set of articles, and use of the Snowballing method within these, is one attempt to mitigate such effects.

A.4. Selection of Primary Studies

To reduce the number of studies evaluated, three different selection processes are applied. The first two are primary and secondary screening, used to filter out non-thematically relevant studies. The primary inclusion criteria uses meta data such as the title and abstract, while the secondary screening uses the entire study text. The third step, quality assessment, is also applied to the entire text.

A.4.1. Primary Inclusion Criteria

IC1 The study's main concern is musical computational creativity.

IC2 The study is a primary study presenting empirical results.

IC3 The study concerns the computational understanding of music's mood or ambiance.

A.4.2. Secondary Inclusion Criteria

IC4 The study concerns music without lyrics.

IC5 The study describes the implementation of an application.

All studies passing the primary and secondary inclusion criteria will continue to quality assessment.

A.5. Study Quality Assessment

To further assess the quality of studies passing the primary and secondary inclusion criteria, a set of 10 quality criteria is used (provided by Kofod-Petersen (2018)). Each study is ranked on each of the quality criteria. The goal of this quality assessment is to evaluate the article with regards to **RQ3** (*What is the strength of the evidence supporting the various conclusions presented?*).
The possible outcomes for each quality criteria are:

- Yes. 1 point
- Partly. 0,5 points
- No. 0 points

QC1 Is there a clear statement of the aim of the research?

QC2 Is the study put into context of other studies and research?

QC3 Are system or algorithmic design decisions justified?

QC4 Is the test data set reproducible?

QC5 Is the study algorithm reproducible?

QC6 Is the experimental procedure thoroughly explained and reproducible?

QC7 Is it clearly stated in the study which other algorithms the study's algorithm(s) have been compared to?

QC8 Are the performance metrics used in the study explained and justified?

QC9 Are the test results thoroughly analysed?

QC10 Does the test evidence support the findings presented?

A.6. Data Extraction

The following data will be extracted from each study to perform the SLR:

- Name of author(s)
- Title
- Year of publication
- Name of proposed system
- Type of proposed algorithm
- Data set source
- Algorithm source (if available)
- Findings and conclusions

The data will be presented in a table, with each data point in one column, and data points for each study within one row.

B. Sheet music and survey results

This section lists all sheet music produced, and their according survey results. The survey results are presented as the average answer from the metrics asked. All metrics were ranked on a scale from 1-5.

Table B.1 presents survey results for each music sample.

Sample ID	Valence	Energy	Pleasantness	Interestingness	Randomness	Sentiments
Q4-12-20-300	3.22	4.34	2.46	3.21	3.84	Sample A
Q4-08-20-300	3.49	2.75	3.65	3.39	2.88	Sample B
Q4-16-20-300	3.92	4.61	3.09	3.18	3.13	Sample C
Q4-12-10-300	-	-	-	-	-	Omitted from survey
Q4-12-40-300	2.77	3.48	3.77	3.73	2.36	Sample E
Q4-12-20-100	-	-	-	-	-	Omitted from survey
Q4-12-20-500	3.86	4.79	2.69	2.80	3.10	Sample G
Q3-08-20-300	-	-	-	-	-	Omitted from survey
Q3-12-20-300	3.24	2.73	3.35	3.04	3.01	Sample H
Q3-16-20-300	2.49	4.23	2.65	3.46	3.39	Sample I
Q3-12-10-300	2.94	4.79	2.14	2.79	3.40	Sample J
Q3-12-40-300	2.74	2.13	3.40	2.88	3.02	Sample K
Q3-12-20-100	3.24	4.06	2.80	2.74	2.95	Sample L
Q3-12-20-500	3.53	3.67	3.19	3.08	3.10	Sample M

Table B.1.: Survey results for each music sample.

Sentiments for **Q4-12-20-300**:

- This would be the soundtrack in a movie where you're being chased by an axe murderer but then you stop to buy ice cream on the way
- This one was fascinating. It actually developed from a very pleasant melody to a much more energetic and random style.
- Sounds computerized
- Mechanical and rigid. But could work as a small part of a film score.
- It's periodically fine, however some notes are clearly off, and it changes suddenly in a non-positive way.

Sentiments for **Q4-08-20-300**:

- Almost sounds like Jazz
- Reminds me of Zelda: Breath of the Wild
- Atmospheric
- two diffrent pieces from before and after 27 secs and out. got a whole diffrent feeling after that
- Good start, towards the ending it's maybe a bit sour, and it stops abruptly. That can be good if done correctly, but maybe not in this manner. It was more structured, which was nice.
- Randomness undermines valence, pleasentness and interest.
- A tonal tendency limits the randomness, the form maintains it.
- Like someone who can play is Just testing a piano

Sentiments for **Q4-16-20-300**:

- It was very random at the end.
- Mechanical
- Uncomfortable
- Like sample A (*Q4-12-20-300*), a few hints of ragtime connotations makes it a little bit more interesting.
- A tonal and structured tendency dissolves a bit at the end.

Sentiments for **Q4-12-40-300**:

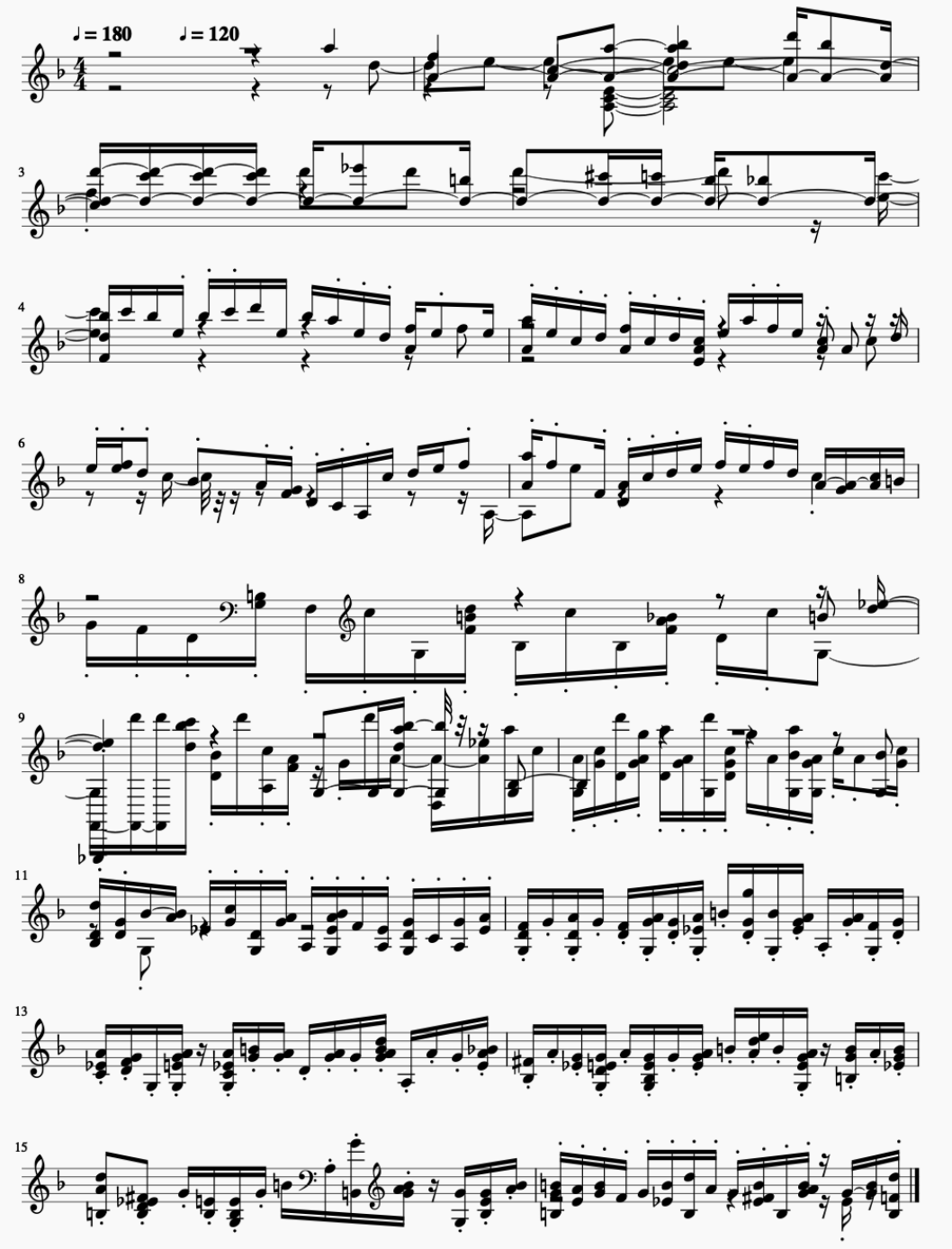

Figure B.1.: Sheet music for sample Q4-12-20-300.

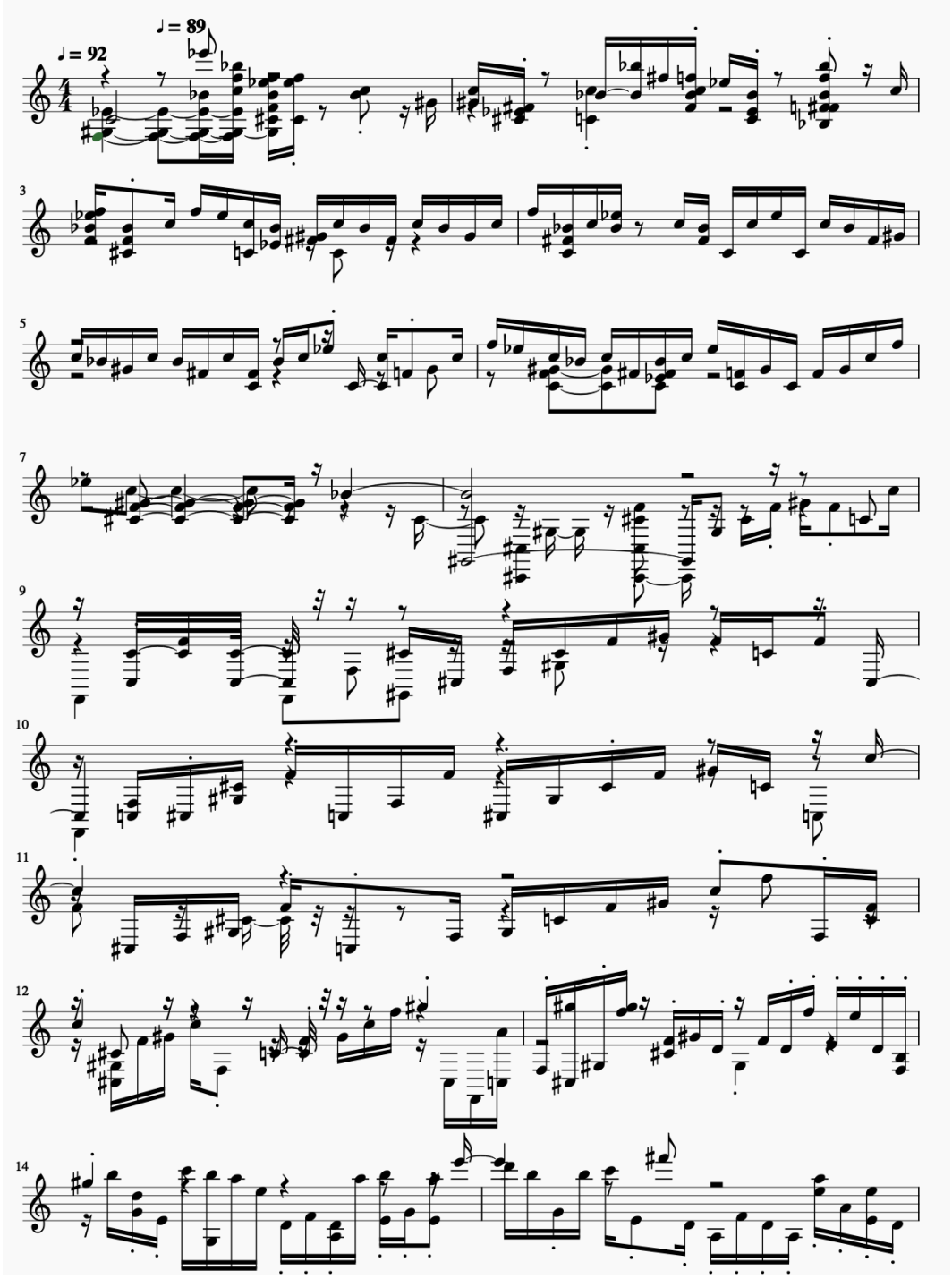

Figure B.2.: Sheet music for sample Q4-08-20-300.

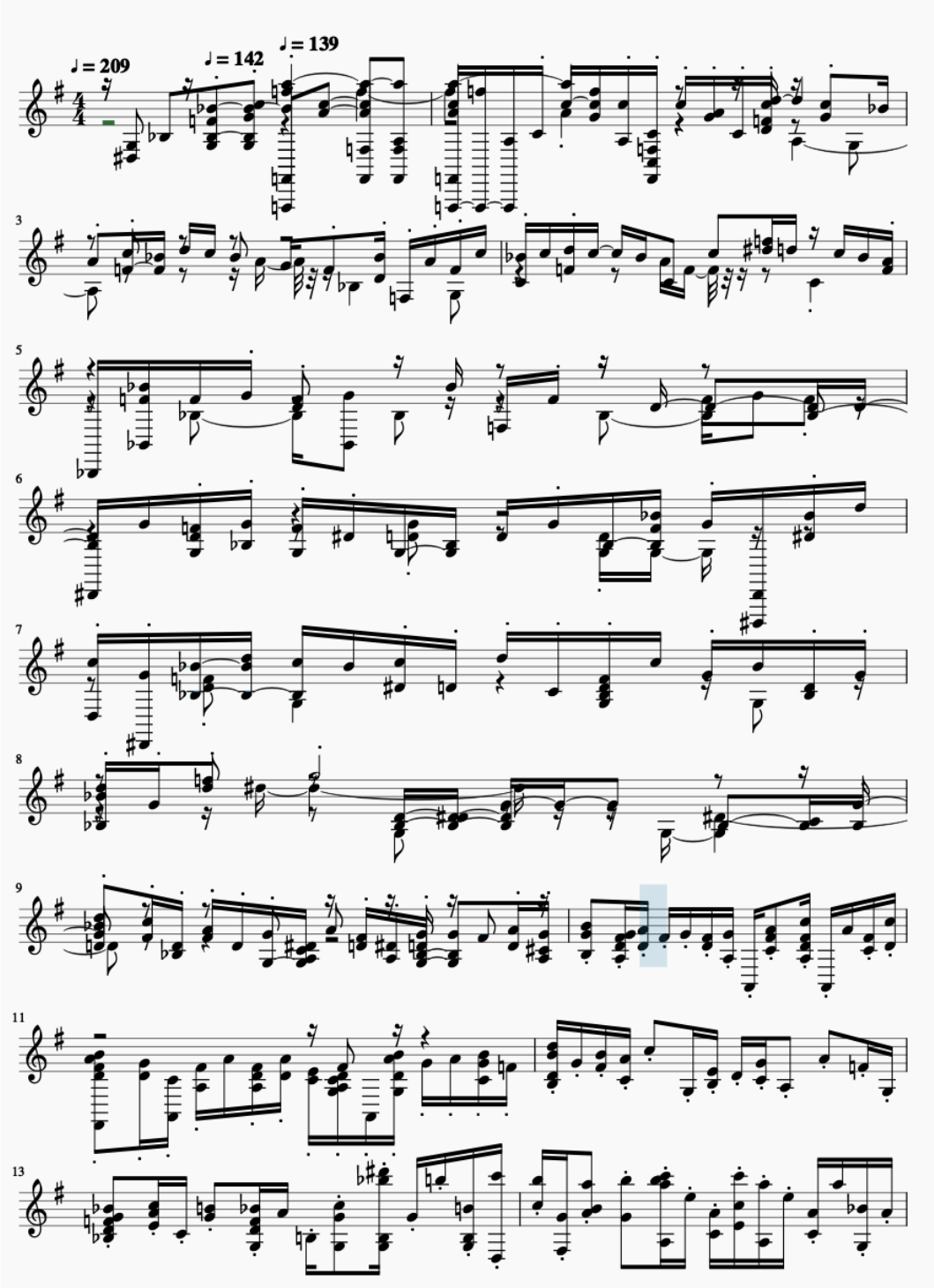

Figure B.3.: Sheet music for sample Q4-16-20-300.

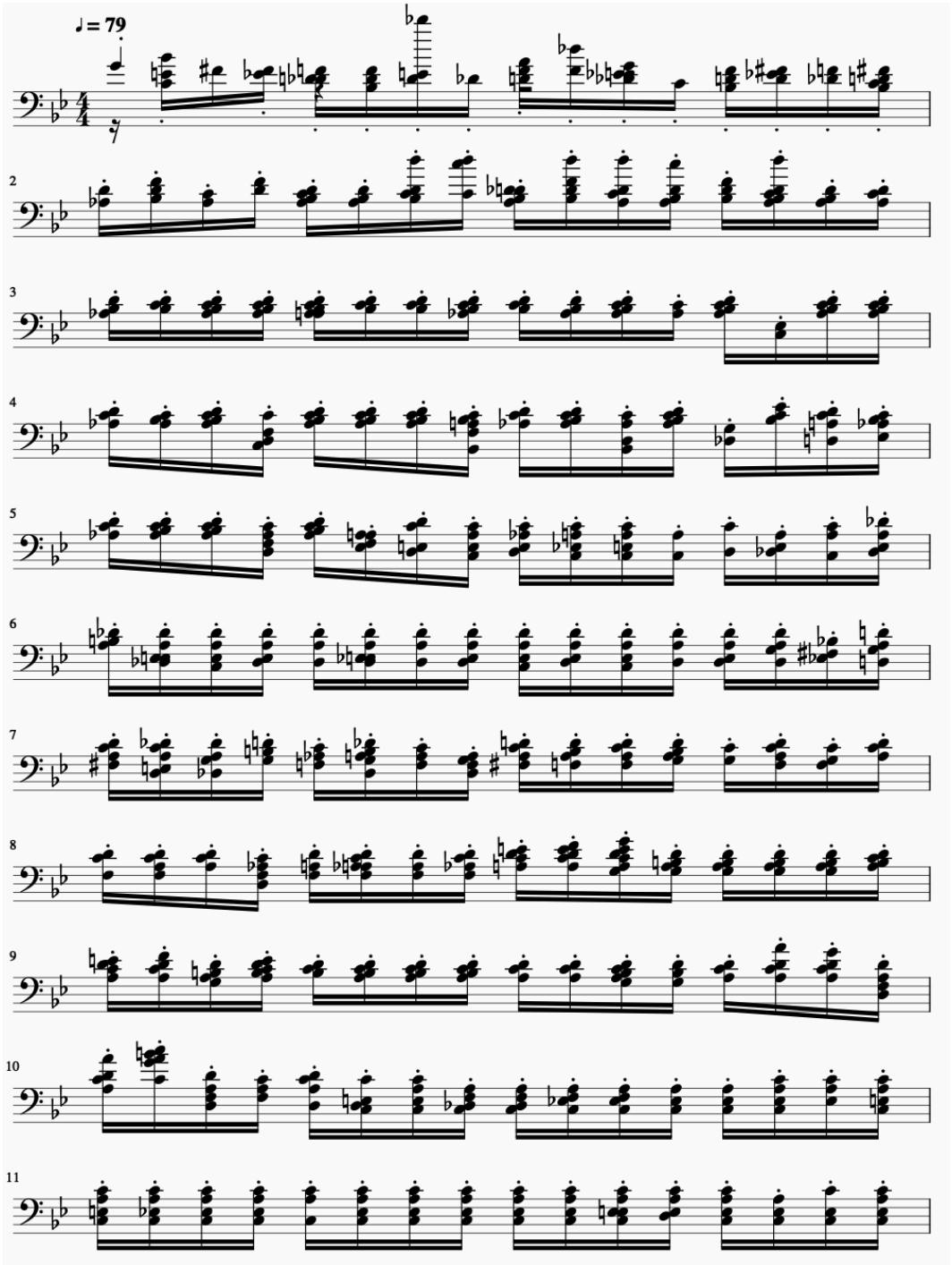

Figure B.4.: Sheet music for sample Q4-12-10-300.

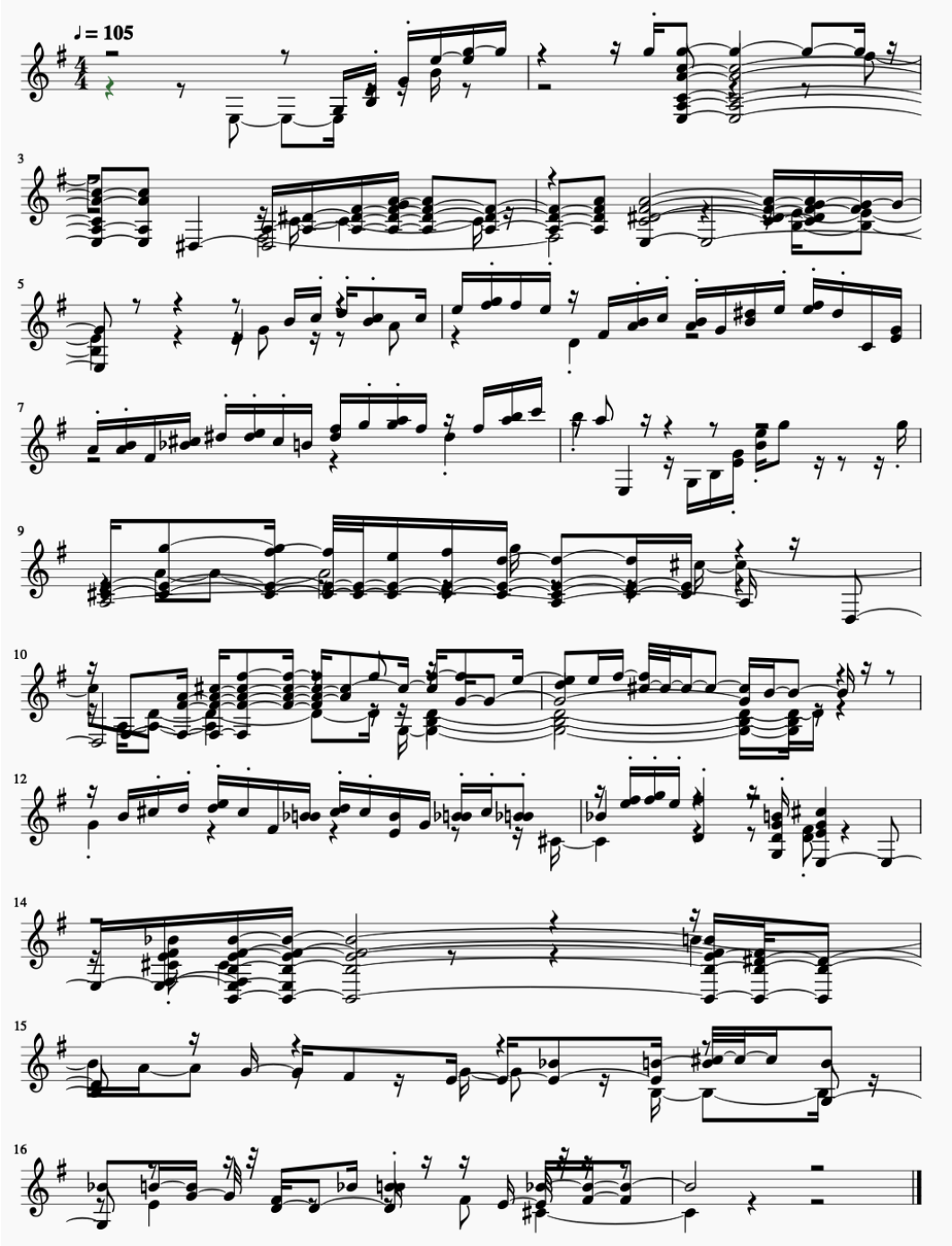

Figure B.5.: Sheet music for sample Q4-12-40-300.

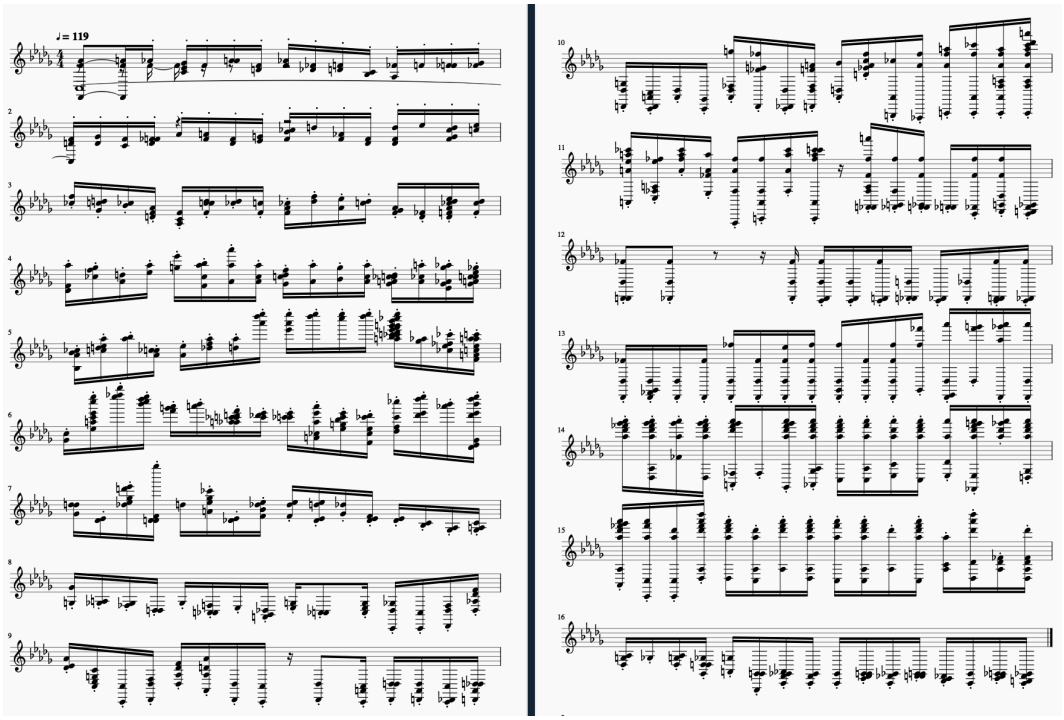

Figure B.6.: Sheet music for sample Q4-12-20-100.

- Destroyed baroch
- Nice! It's a bit off on some of the notes, which kinda always ruins a piece of music, but I can see what it's trying to do, it's pleasant to follow and at times I find it almost beautiful.
- Bach-ish :-)
- Obvious conotations to classical music makes it more interesting and a little bit more valencing.
- Moments of Bach or Beethoven, but hitting random additional keys most of the time.
- Perpetum mobile

Sentiments for **Q4-12-20-500**:

- think I've heard the first four bars before
- It's mostly just fast
- Ragtime on speed

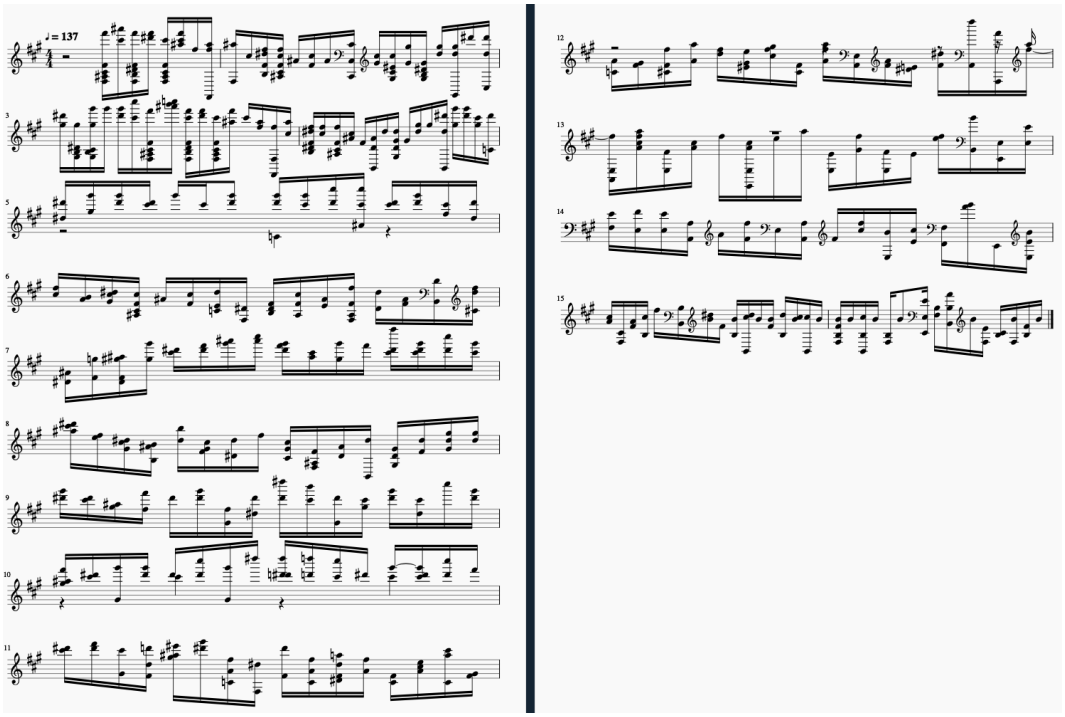

Figure B.7.: Sheet music for sample Q4-12-20-500.

- Late 19th to early 20th century pastisch (or perhaps something between Schubert and Winifred Atwell).
- Stumfilmfeeling
- The First one I think could be made of a human

Sentiments for **Q3-12-20-300**:

- It's pleasant, but then towards the end it kinda loses it. Pleasant melody, but a bit random.
- More chord progressions with jazz conotations in the beginning, less afterwards
- Debussy in the land of jazz ballads?

Sentiments for **Q3-16-20-300**:

- Quite close to contemporary music composed by humans
- Debussy after a visit to Strawinsky's?

Sentiments for **Q3-12-10-300**:

- Also just fast

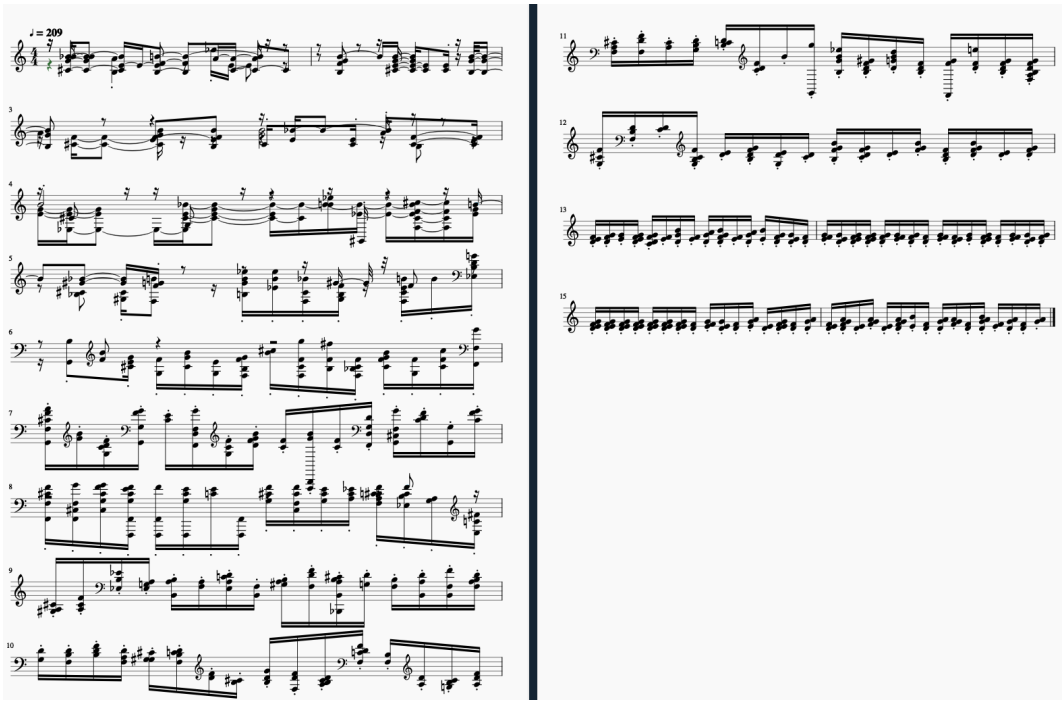

Figure B.8.: Sheet music for sample Q3-08-20-300.

- Unplayable for humans, a machine out of control
- This track made me want to hear more, it felt like I was interested in a longer version of the track.
- Too many fingers and a high level of stress?
- Noisy

Sentiments for **Q3-12-40-300**:

- the syncopation just seems random
- jazz standard gone ai
- The notes are good together, but timing and length of the need adjustment.
- Intended sadness is traseable in classical references
- A Chopin/Grieg-like probing of harmony.
- Halting

Sentiments for **Q3-12-20-100**:

- Pretty!

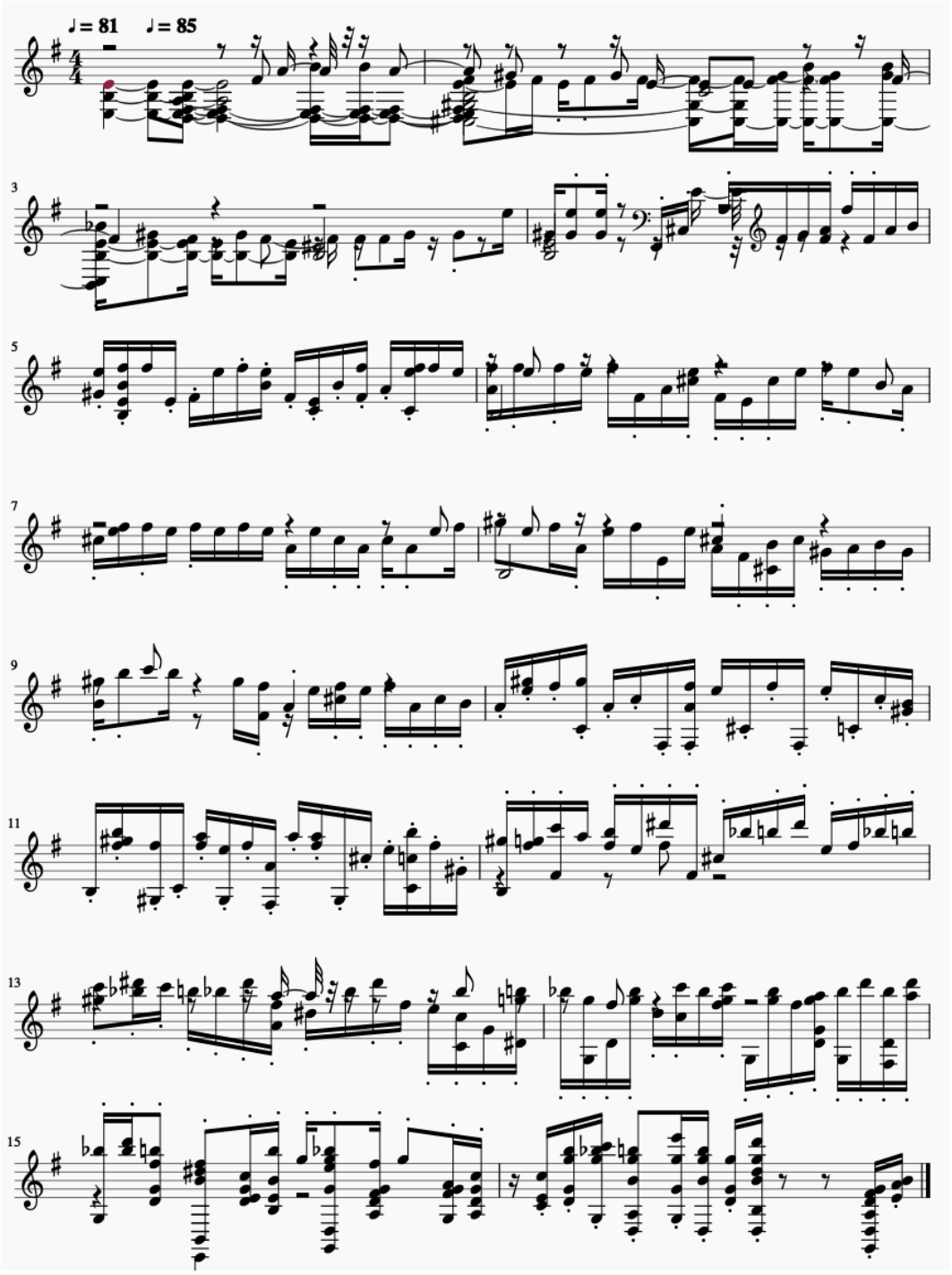

Figure B.9.: Sheet music for sample Q3-12-20-300.

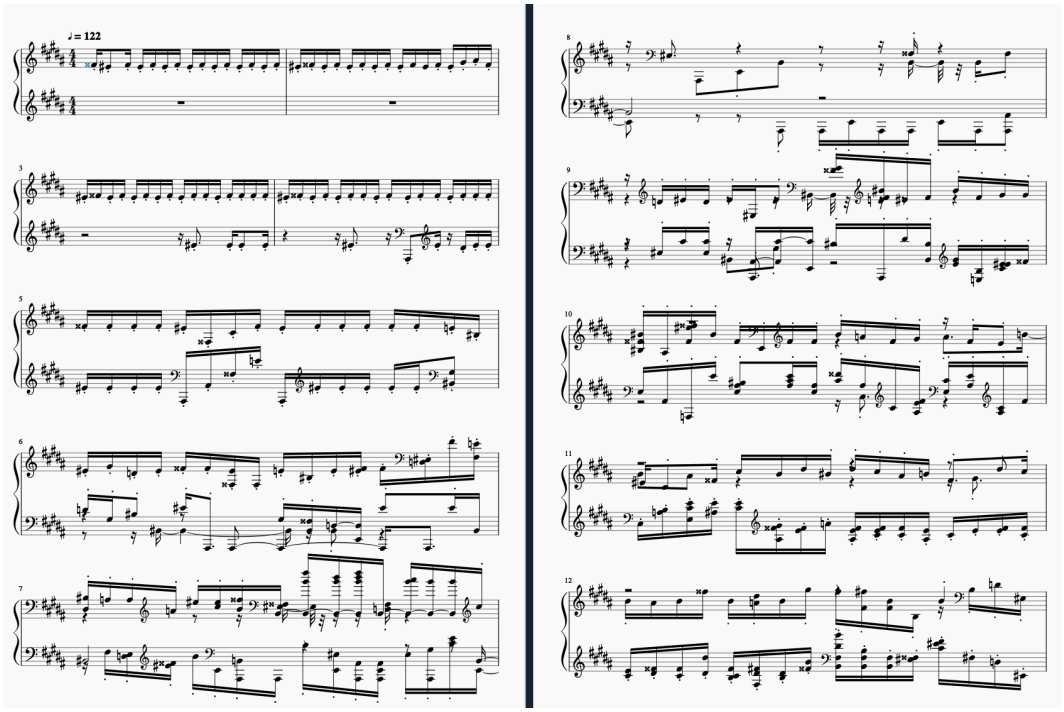

Figure B.10.: Sheet music for sample Q3-16-20-300.

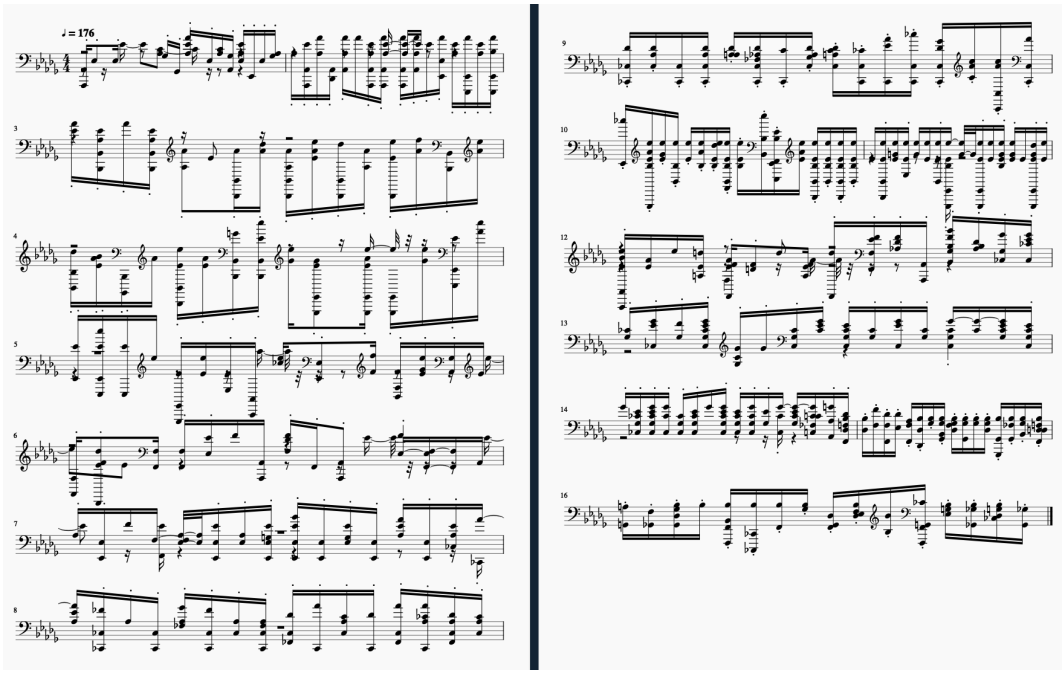

Figure B.11.: Sheet music for sample Q3-12-10-300.

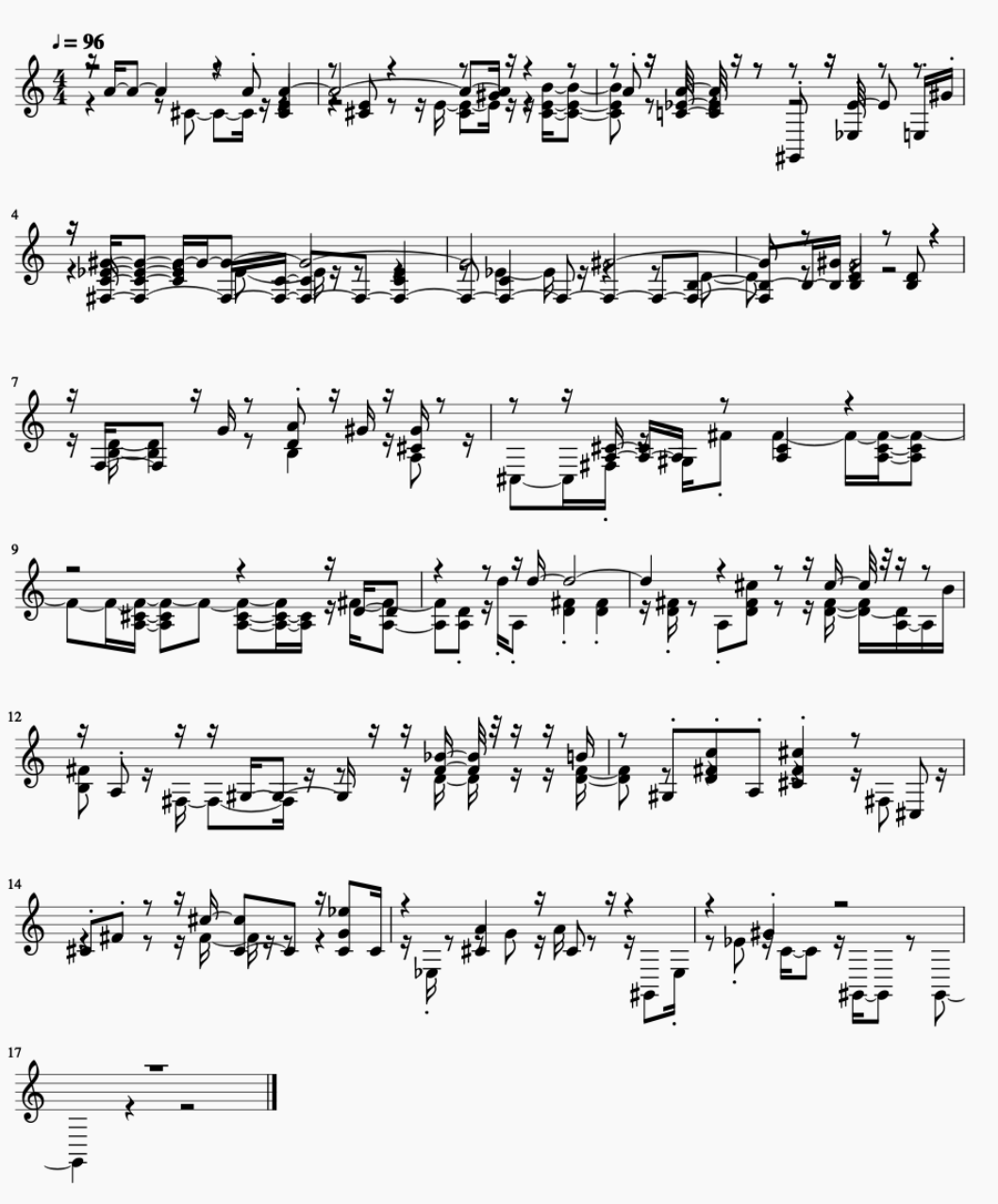

Figure B.12.: Sheet music for sample Q3-12-40-300.

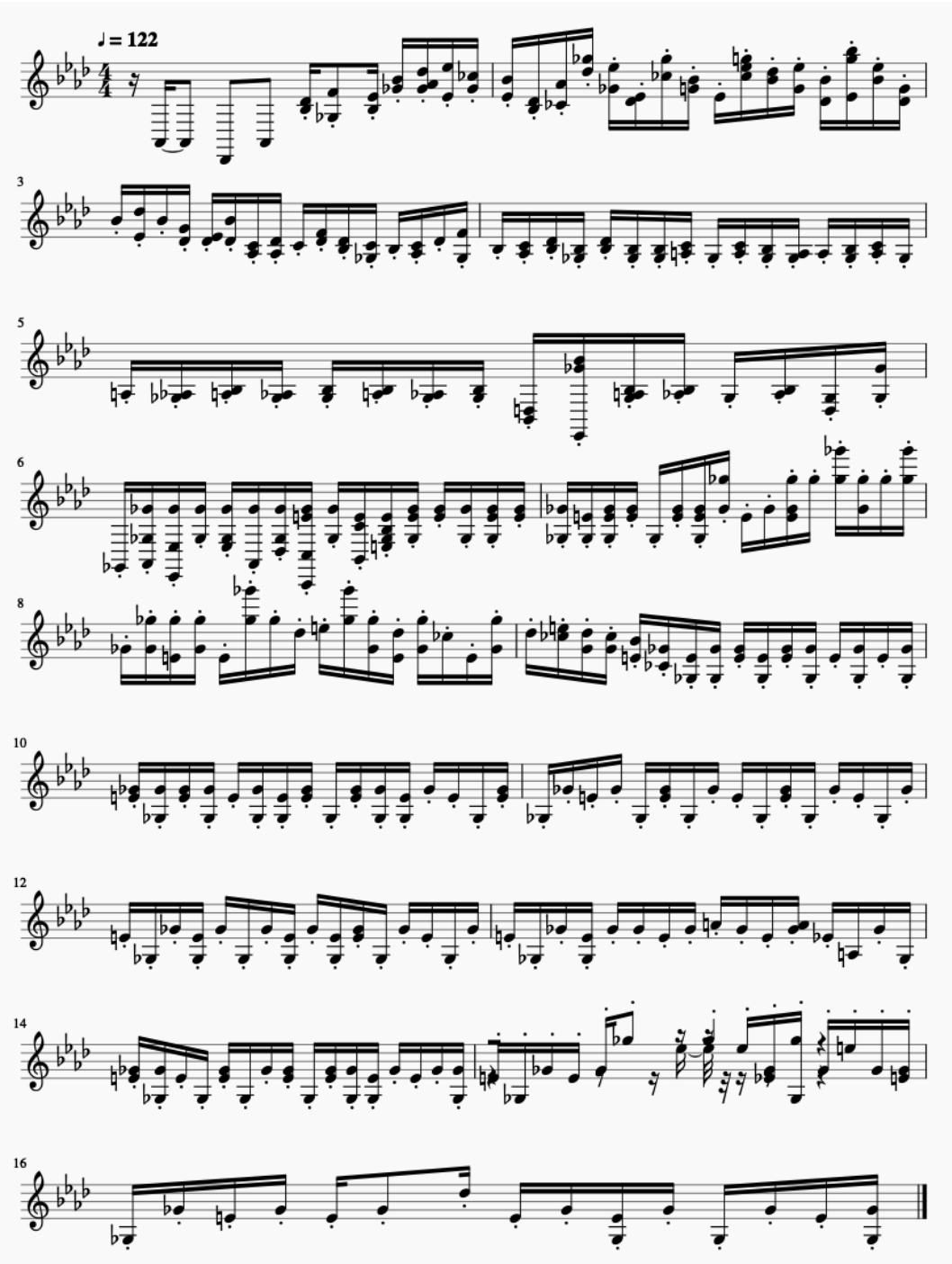

Figure B.13.: Sheet music for sample Q3-12-20-100.

- Semistructures here and there, but not really interesting
- Debussy after a visit to Sæverud's.

Sentiments for **Q3-12-20-500**:

- Pretty, but a bit messy at times. The melody is nice, but it disappears a bit in the composure.
- This high energy need to say something empty (witout content) makes me sad.
- Oh, I discovered the A, how nice - but I miss it every now and then...

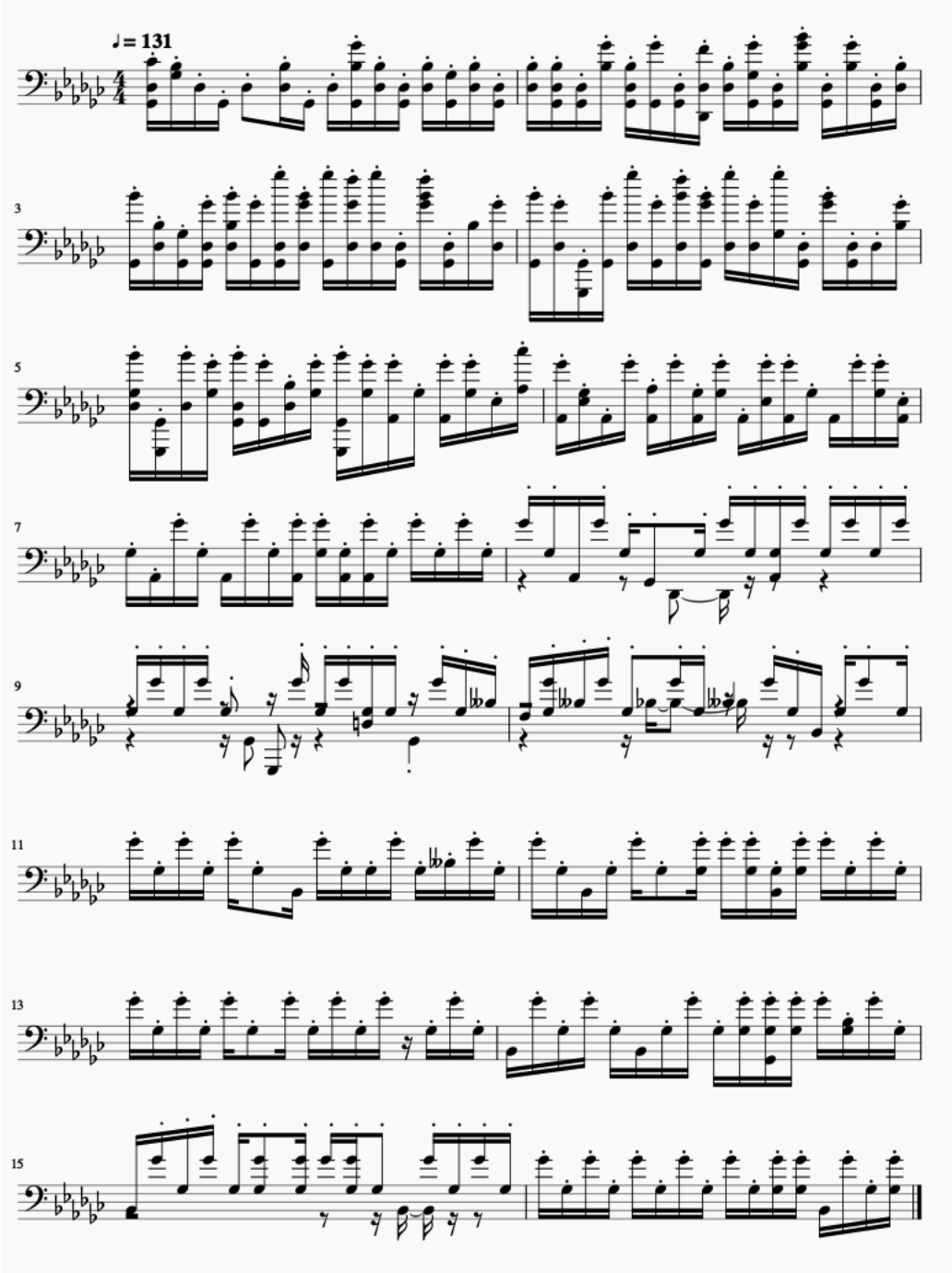

Figure B.14.: Sheet music for sample Q3-12-20-500.

Agnete Djupvik

Music That Feels Just Right

NTNU
Kunnskap for en bedre verden