WARSAW UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRONICS AND INFORMATION TECHNOLOGY

Ph.D. Thesis

Mateusz Modrzejewski, M.Sc.

Artificial Intelligence Solutions for Artistic Multimedia Musical Content Creation Support

Supervisor
Prof. dr hab. Inż. Przemysław Rokita

WARSAW 2021

Streszczenie

Algorytmy sztucznej inteligencji wspierające procesy tworzenia multimedialnych muzycznych treści artystycznych

Celem niniejszej rozprawy było przeprowadzenie badań w zakresie zastosowań sztucznej inteligencji do wspierania procesu wytwarzania artystycznych multimedialnych treści muzycznych oraz analizy tych treści. W literaturze szeroko opisane są przełomowe osiągnięcia sztucznych sieci neuronowych, zastosowanych do danych ustrukturyzowanych, tekstu, obrazów i materiałów audio. Jest to bardzo dynamicznie rozwijająca się dziedzina badawcza współczesnej informatyki. Niniejsza rozprawa ma na celu wypełnienie wybranych luk we wspomnianych zastosowaniach w kontekście muzyki. Zauważalny w ostatnich latach rozwój wielu nowych architektur głębokich sieci neuronowych był główną inspiracją dla wszystkich prezentowanych eksperymentów.

Założone cele zostały osiągnięte poprzez porównanie i zastosowanie różnych architektur sieci neuronowych w kluczowych obszarach tworzenia i analizy treści muzycznych. Wykorzystane zostały również różne reprezentacje treści muzycznych, w tym reprezentacje graficzne oraz MIDI. Zaproponowane zostały dwa modele generatywne, jeden do wytwarzania nowych fraz muzycznych i jeden do transferu stylu muzycznego, jak również rozwiązanie z dziedziny klasyfikacji gatunku muzycznego wraz z rzadko spotykaną w literaturze analizą uzyskanych wyników od strony muzycznej. Proponowane metody zostały porównane z innymi, dostępnymi i opisanymi w literaturze rozwiązaniami.

Zaprezentowane w rozprawie rozwiązania zostały zaprojektowane z myślą o niższych wymogach obliczeniowych, niż jest to spotykane chociażby przy ogromnych modelach sieci neuronowych opisywanych w pokrewnych dziedzinach. Zaproponowane rozwiązania mają realny potencjał wdrożeniowy w omawianej dziedzinie i mogą być stosowane do wzbogacenia procesu twórczego w zagadnieniach kompozycji, produkcji i analizy muzyki.

Słowa kluczowe: *sztuczna inteligencja, sieci neuronowe, muzyka, artystyczne treści multimedialne*

Abstract

Artificial Intelligence Solutions for Artistic Multimedia Musical Content Creation Support

The goal of this work was to research and propose new solutions in the field of applications of artificial intelligence algorithms for musical content creation and analysis. Breakthrough achievements of neural networks, as applied to structured data, text, images and various forms of audio, have already been previously described in literature and are a very active area of modern computer science. This thesis aims to fulfill chosen gaps of such applications to music. The rapid development of new deep neural network architectures in the recent years has served as the base inspiration for all presented experiments.

In order to achieve the objective of this thesis, several architectures have been used and compared in key areas of musical content creation and analysis, incorporating also the use of various representations of musical content, including graphical representations and MIDI data. Two generative models are proposed, one for content generation and one for musical style transfer, along with a musically informed and explained approach to the issue of music genre classification. The obtained results are compared to ones achieved by existing solutions described in literature.

The presented solutions were designed to present a light-weight approach when compared with contemporary huge models, seen in similar issues. The solutions proposed in this thesis are applicable for deployment and enhancement of creative processes in real-world musical content creation.

Key words: *artificial intelligence, neural networks, music, artistic multimedia content*

Contents

Chapter 1

Introduction

1.1 Motivation

For centuries people have dreamt about automating certain tasks, motivated by efficiency, difficulty, danger or convenience. Ideas about this kind of automation have evolved from inventing better, more technologically advanced tools, through human-steered machines and early ideas about robots, up to modern visions of technologies and devices possessing a certain degree of intelligence and sentience. Artificial intelligence has certainly gone a very long way since its beginnings. Although we are still quite some steps apart from obtaining general artificial intelligence, our perception of the usage of intelligent systems has greatly increased in the last few years. Ideas considered in the past as theoretical models have since gone far out of the drawing board, into production and deployment, and now seem to contribute more and more to many aspects of our daily lives. After many "winters" of artificial intelligence [Duan et al., 2019], we seem to be living in the era of the hottest expansion of this domain in history, with very tangible AI-powered enhancements to many domains of our lives.

This would have not been possible without a vast amount of development in many areas, from theoretical modelling, through data collection and hardware improvements, up to concrete, applied business endeavors, incremental refinement of implemented solutions and extended interpretation of the collected information.

First of all, the theoretical models for much of what we consider today as machine learning and artificial intelligence have been created a long time ago, with many of the algorithms and ideas originating from mathematics and statistics [Wengert, 1964] [Rumelhart et al., 1986] [Rosenblatt, 1958] [Cox, 1972] . However, at the time computational resources sufficient for real-world implementations and further refinement were not yet available [Baydin et al., 2018].

The rise of digitalization of information has paved the way for new approaches of data storage, processing, interpretation and utilization. Computers vastly outperform humans in storing and batch processing of massive amounts of data. The amounts of data collected in online resources seem to be almost endless, with global data centers taking areas of millions of square feet and and overall of around 175 zettabytes of data expected by 2025 [Rydning et al., 2018]. The key difficulty, however, is the knowledge and insight that can be extracted from all this data and further applied in meaningful ways.

Deep architectures of convolutional neural networks applied to images have been a true breakthrough for artificial intelligence, with a significant increase in the interest in this field after the AlexNet model [Krizhevsky et al., 2012] submitted for the ImageNet Large Scale Visual Recognition Challenge [Deng et al., 2009] [Russakovsky et al., 2015]. Deep learning has since achieved great results in certain domains, in particular in image and natural language processing, where subsequent models have been reported to outperform humans in tasks like image recognition. However, an incredibly important aspect of the evolution of artificial intelligence is the computing power needed to train the newly developed models. The usage of graphics processing units (GPU) in the early 2010's has enabled the emergence of implementations and development of artificial intelligence as we know it today, as training on GPUs is significantly faster. While the convolutional neural network presented in [Ciresan et al., 2011] or the aforementioned AlexNet model does not differ much from the networks proposed years before by Yann LeCun [LeCun et al., 1989] [LeCun et al., 1998], the computing power needed to bring it up to scale was finally in reach with the usage of GPUs.

It has been only around a decade since training artificial intelligence models (neural networks in particular) on GPU's has become the standard procedure, and the current perspective is much different, with emerging efforts of creating AI-dedicated hardware, like Google's TPU (*tensor processing unit*) [Jouppi et al., 2017] [Jouppi et al., 2018], Graphcore's IPU (*intelligence processing unit*) [Jia et al., 2019] and the recently announced Tesla D1 chip built for the Dojo Supercomputer. Some of the efforts seem to be somewhat consequences of recent chip shortages and issues with rare earth mining, with additional concerns about the carbon footprint of extremely heavy compute being raised.

As deep learning gets deeper, bigger neural network architectures are being developed, with many sophisticated layers and huge amounts of parameters. Here it is hard to overlook the financial side of AI research: for instance, the cost of training GPT-3, a state of the art language transformer model with a staggering amount of 175 billion parameters, is estimated to be around 12 million US dollars, with the authors implicitly mentioning that retraining the model after finding a (somewhat minor) bug was simply not feasible [Brown et al., 2020]. Subsequently, the model has been licensed exclusively to Microsoft. One of the descendants of GPT-3 is the OpenAI Codex, which powers Microsoft's Github Copilot, a tool able to generate working code in a multiple programming languages, when given a text description. Although still in early stages of development, it has been both an exceptionally spectacular and controversial demonstration of the capabilities of such models.

The understanding of image and text data seems to make a lot of sense from a business perspective, hence somewhat explaining the hype and financing behind this kind of research. Automated voice assistants like Amazon Alexa or the Google Assistant based on natural language understanding have already become a commodity, as have artificial intelligence-powered enhancements of camera arrays installed in modern smartphones. Other applications of understanding image and text data include automatic visual quality control, aided medical image analysis, sentiment analysis, recommendation systems and business operation optimization. Autonomous vehicle technology is also reportedly getting better and more applicable, with promising and impressive development from companies like Tesla with their *Hydra Net* approach (with RegNets [Xu et al., 2021] and BiFPN - Bidirectional Feature Pyramid Networks [Tan et al., 2020] underneath) and comma.ai's *openpilot*.

Neural networks have become a viable choice when the objective is to outperform humans in challenging tasks of classification or regression. Their increasing capabilities of analysis and understanding of multimedia content, along with the massive interest in the subject, has propelled the application of neural networks to a broader range of use cases - in particular, somewhat of a generative and artistic nature.

Google's *deep dream* [Mordvintsev et al., 2015] method for creating dreamy (and sometimes quite nightmarish) images via superimposing surreal, fractal-like elements has been created as a by-product of visualization of the learning process of convolutional neural networks. Although similar ideas for visualization were also previously presented (for instance in works like [Erhan et al., 2009] and [Simonyan et al., 2014]), Google's contribution of open-source code and online tools for generating *deep dreams* out of your own images have gained significant attention. Another application with artistic quality is neural style transfer, introduced in [Gatys et al., 2015b] - algorithmic imposition of a different style to a given artwork, for instance imposing the style of a painter to a photograph. The method has since been adopted as an artistic image filtering method in several mobile applications. Furthermore, a whole family of algorithms for generating new, previously unseen content has been built in recent years, with generative-adversarial networks [Goodfellow et al., 2014] being an especially well-known example.

An equally interesting domain is artificial intelligence in audio processing. The domain, although less mainstream than the aforementioned natural language and image domains, tackles many fascinating problems, like speech recognition and sound classification. Audio processing algorithms are increasingly finding their way into deployment, from urban sound detection and understanding systems up to listening enhancements, like active noise cancellation and personalized music equalization.

This brings us to the general issue of applying artificial intelligence to music. Music is a particularly difficult form of data for AI for a number of reasons: music has a sequential structure, often additionally convoluted through parts of different instruments; it also involves the usage of lyrics and human vocals. The contents of music have a high level of abstraction, as the qualitative and quantitative values are often deeply based in music theory or recording, mixing, mastering and production details, and, oftentimes, the way we subjectively perceive music based on our cultural heritage, listening experience and even personal taste.

The scope of applications of artificial intelligence to music span from generating new musical content, through AI-aided processing of music, up to analysis of existing music within music information retrieval systems [Casey et al., 2008]. Artificial intelligence may also be used to enhance human creativity and intelligence with new tools and solutions. This particular field is equally concerned with creating algorithms capable of some degree of human-like creativity [Pasquier et al., 2017], as it is with better understanding our own creativity and formulating a perspective on our behavior [Pearce et al., 2002].

Music is a deeply human phenomenon and, along with dance, has evolved in virtually every culture throughout human history [Dunbar, 2012]. Having its beginnings in spiritual and tribal experiences, it has gained a social and political significance in modern times, as described in [Gilbert and Pearson, 2002]. The changes in musical style and form have went pair-in-pair with political and cultural shifts in history. It's hard to overlook the role of technology in the evolution of music: inventions of new instruments and tools of expression gave birth to new compositional techniques and were in many cases the catalysts of musical progress. From the invention of the piano or the saxophone, up to the popularization of digital processing and synthesizers [Dahlhaus et al., 1983] [Burgess, 2014], music has been a lasting mark of the technological achievements of times. With the ideas of computational creativity and algorithmic composition, artificial intelligence aided software seems to provide a newly emerging type of musical instrument, slowly finding its way into the creative process of professional composers and amateur music creators alike [Fernández and Vico, 2013].

Another breakthrough has come with the idea of recorded music, which has forever changed the way how we perceive and consume music [Chanan, 1995], turning it into a previously unavailable, much more personal experience. This is obviously further reflected in today's usage of artificial intelligence-powered recommendation algorithms in streaming services like Spotify or Apple Music.

Looking at the state of the art in artificial intelligence and deep learning, the aforementioned issue of the availability of data comes back. Publicly available musical data for training and development of artificial intelligence solutions is much sparser than in the case of images. Furthermore, from an algorithmic standpoint, music can be represented in many ways: it can be processed in a similar manner to other audio signals, it can be understood as a type of rich sequential data, compressed into a series of pitches organized in a particular rhythm, it can also be described via representation learning or laboriously handcrafted meta data. Evaluation of the obtained results is also a difficult task, as human intervention and musical expertise is required in mosts cases [Lopez-Rincon et al., 2018], with a degree of fuzziness and uncertainty arising from personal taste and lack of formalisms in stylistic musical differences. As there has never been a consensus between composers and listeners about the qualities of music, a consensus about its appropriate representations and methods of evaluation for algorithmic purposes has yet to be found by researchers.

This brings us to many unanswered questions about the application of artificial intelligence to artistic multimedia musical content creation and analysis: *how can we create highly applicable AI solutions that would benefit composers? How can we support composition, production and music processing with AI methods? How can we analyze music with artificial intelligence and not lose the original, musical context? Is the progress of the domain dependent on huge models and massive computing power, as seen in other domains, or are there other ways of obtaining musical computational creativity enhancement?* This work explores some of these questions within key issues of the field, such as composition, sound quality production and information retrieval. The solutions presented in the following chapters attempt to fill the gaps in the domain with propositions of highly accessible, lightweight methods, while maintaining a musically informed and insightful point of view, which still is rarely seen in musical research in artificial intelligence [Choi et al., 2017a] [Sturm, 2013b]. The first area chosen in particular is the relatively new and uncharted timbral musical style transfer, with a concrete proposition of a novel solution. The second one is the area of original musical composition, with a method for generating new, previously unheard musical phrases intended for direct enhancement of the creative process of a human composer. The third area is music information retrieval with experiments on music genre classification, providing new answers to the problem of such classification with musical analysis and insight into the analyzed dataset and trained models, which is still a rarity in the field.

1.2 Goal of thesis

Upon the described motivation, particular scientific aims have been formulated. The thesis aims to propose solutions providing new experiences for music creators, also useful in further creation of novel tools and applications, as well as enhancing the capabilities of music analysis and music information retrieval.

The main goals of the thesis are therefore as follows:

1. Propose and evaluate a new solution for musical timbral style transfer, in particular without the usage of heavy computation as seen in raw audio models.

2. Investigate the usefulness of recurrent neural networks and autoencoder structures for the issue of lightweight musical timbral style transfer.

3. Propose and evaluate a new solution for original music composition and validate the usefulness of graphical representations of music for this purpose.

4. Provide a broader, musically informed perspective in the issue of musical genre classification upon investigation on available benchmark data and convolutional and recurrent models.

5. Provide original musical insight, analysis and context aware conclusions in all of the experiments.

1.3 Document structure

The structure of this thesis is as follows:

- Chapter 2 contains an overview of musical terms and concepts, like rhythm, harmony, musical structure in terms of phrases and forms, necessary for further analysis and discussion. It also describes various representations of music, including ones suitable for use with machine learning algorithms, providing context for all further chapters.

- Chapter 3 describes results of work on the issue of musical timbral style transfer. An efficient neural network architecture consisting of an encoder-decoder with LSTM layers is presented, along with a method for generating datasets programatically using MIDI data [Modrzejewski et al., 2021]. The chapter also provides an analysis of the obtained results and details on the proposed architecture.

- Chapter 4 describes results of work on the issue of generating short, usable musical phrases. A method utilizing deep convolutional generative-adversarial networks with a compressed graphical piano roll representation is presented [Modrzejewski et al., 2019]. The chapter also describes the contribution of an actual mini-album of music created for the needs of evaluation of the proposed approach.

- Chapter 5 describes results of work on the issue of music classification using graphical representations in a setting where the experiments have been carried out on a bigger dataset than reference literature. Musical explanation into sonic and compositional differences between genres is provided for the results, along with critical analysis of the benchmarks in musical genre classification and discussion on the performance of the trained models [Modrzejewski et al., 2020].

- Chapter 6 presents a summary of the thesis along with conclusions, discussion and ideas for further research.

- Appendix A contains a list of the author's peer-reviewed publications.

- Appendix B contains a brief summary of the author's artistic background, which was also a key factor in the motivation behind the presented research.

- Appendix C presents the current legal state on the non-obvious and important issue of the copyright of musical content created with the aid of artificial intelligence. Chosen examples of actual modern legal cases are also described and discussed in the Appendix.

Chapter 2

Musical Overview

The following chapter describes the necessary musical terms which are the base of the subsequent discussion and experiments with artificial intelligence applications. Standard concepts which constitute the musical background are explained, along with notes on musical notation, audio signal processing and the usage of dedicated multimedia formats suitable for processing by algorithms of artificial intelligence. However, it is worth noting that musical theory is a separate field of research, with a very rich history and many more concepts - the terms presented below serve only as an introduction, with no aesthetic evaluation whatsoever.

According to [Knees and Schedl, 2016], features of music suitable for algorithmic processing can be categorized with respect to various dimensions, like:

- level of abstraction,
- temporal scope,
- signal domain,
- musical aspect.

Besides the work by Knees, [Lerch, 2012], [Eyben, 2015] and [Virtanen et al., 2018] provide a detailed description of several features used for audio processing and music information retrieval.

Level of abstraction

Features may be divided according to their level of abstraction (low, mid or high), in direct proportion of their semantic meaning for the user and inverse proportion to their closeness to a raw audio signal. Therefore, low level features are computed directly from the waveform and use methods of digital signal processing. Examples of such features are the signal's energy, zero crossing rate and spectral centroid. Mid level features combine multiple low-level features in a meaningful way or apply certain psychoacoustic models. Examples include the MFCCs (*Mel frequency cepstral coefficients*), note onsets and fluctuation patterns. High level features will describe music in terms corresponding to human perception. These features will encompass, for instance, song lyrics, rhythm, melody, instrumentation and oftentimes subjective emotional qualities or natural language descriptions, as perceived by listeners.

Temporal scope

Features may be divided according to their temporal scope: we can distinguish instantaneous, segment-level and global features. Instantaneous features cover at most a few tens of milliseconds, due to the temporal resolution of human ear. Global features will encompass the entire music item (full musical piece or given audio fragment).

Signal domain

Musical features may be computed in the time or frequency domain. The time domain represents the amplitude of the signal in each time the signal was observed, while the frequency domain describes the magnitude of the signal at various frequencies [Knees and Schedl, 2016]. The frequency domain representation is usually obtained via a discrete Fourier transform, with the related Gabor transform [vd Boogaart and Lienhart, 2009], constant Q transform (CQT) [Wülfing and Riedmiller, 2012] [Schörkhuber and Klapuri, 2010] and Gammatone filtering also in use for musical data.

Musical aspect

Music audio features may be associated with a concrete musical aspect they describe, like rhythm, melody, harmony and compositional structure. These features, necessary for understanding and evaluation of computational applications of music, are described thoroughly in the following section.

2.1 Base concepts

Music, in its most general definition, can be seen as sound organized with a degree of intention. For the needs of this work, we consider primarily classical [Kallen, 2013] and modern, mostly western music [Nicholls, 1998]. The building blocks of a musical piece can be enumerated in terms of *rhythm*, *melody*, *harmony* and the underlying *structure* of the piece. These qualities may also serve as high-level features, suitable for machine learning algorithms and musical content analysis.

2.1.1 Rhythm

Musical notes, in most cases, have a fixed, finite duration. Rhythm is a general term which describes how music is organized in time.

- beat - the smallest unit of time when conceptualizing rhythm.
- rest - silence of a given time.
- bar - a unit of musical time consisting of a fixed number of notes.
- tempo - the general pace of the musical piece. It may be fixed or fluent (*rubato, a tempo*). Fixed tempos historically were described by Italian terms like *allegro*, *adagio*, but today it is more common to describe the tempo by BPM (*beats per minute* count). Tempos around 100BPM [London, 2012] and 120BPM [Moelants, 2002] are commonly heard in today's popular music, as it has been found this is a very natural tempo for human listening, as well as dance, walking and other activities.
- meter - the organization and measurement of notes inside a bar, which can be conceptualized as the rhythmic capacity of a bar. A common meter in which most popular music is written is $\frac{4}{4}$. Another common meter is the waltz meter, $\frac{3}{4}$. Other descriptions of meter include the common time 𝄴, or the *alla breve* 𝄵.
- duration, value - musical notes may be named by their duration. A bar of $\frac{4}{4}$ consists of four *quarter notes* (or any sum their rhythmic equivalents or subdivisions), noted as ♩. A whole note 𝅝 takes up a whole bar of $\frac{4}{4}$. Standard division of notes cuts the duration in half, with a subsequent division of the whole note into half notes 𝅗𝅥, quarter notes ♩, eight notes ♪, and so on. Dividing into triplet or other odd subdivisions is another possibility.
- polymeter - two or more different meters occurring simultaneously.
- polyrhythm - two or more different rhythms occurring simultaneously. A special case of polyrhythms occurs when played by a single musician, often a sign of high instrumental proficiency.
- syncopation - a rhythmic shift where the accent does not coincide with the natural metric accent. Syncopation is the base for much of modern popular music.

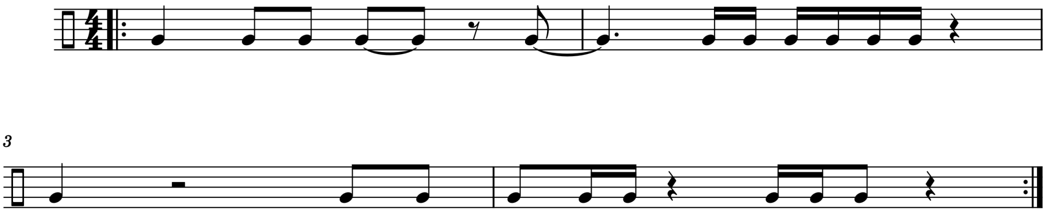

Figure 2.1: Example of rhythmic musical notation.

Figure 2.1 depicts four *bars* in $\frac{4}{4}$ meter with various note durations, rests, syncopation and note duration modifications via ties and dots. All of the note durations inside of the bars add up to a whole note, as assured by the meter.

2.1.2 Melody

Melody is the general term for organizing notes with various pitches in a sequential manner, forming successions that can be perceived as a single entity.

- pitch - a subjective psychoacoustical quality of sound, a stable vibration of a given frequency, which makes it possible to determine notes as lower or higher to one another. Pitch allows to quantize notes with names (C, D, E, F, G, A, B), with *accidentals* like the sharp ♯ or flat ♭, when needed.

- interval - difference in pitch between two musical sounds. Each interval has a distinctive sound to the human ear, regardless of the base pitch. Certain intervals may sound *consonant* or *dissonant* to the human ear, as in possessing a feeling of "pleasantness" or "harshness".

- octave - the interval between a musical pitch and another pitch, which has double the frequency. The keys on a standard 88 key grand piano form $7\frac{1}{4}$ octaves.

- contour - the direction of the melody: subsequent notes may move higher, lower or stay the same.

- scale - a series of organized intervals used to build melodies. Certain scales have evolved naturally during the development of human cultures, like the ancient Greek musical modes [Monro, 1894] [Winnington-Ingram, 2015].

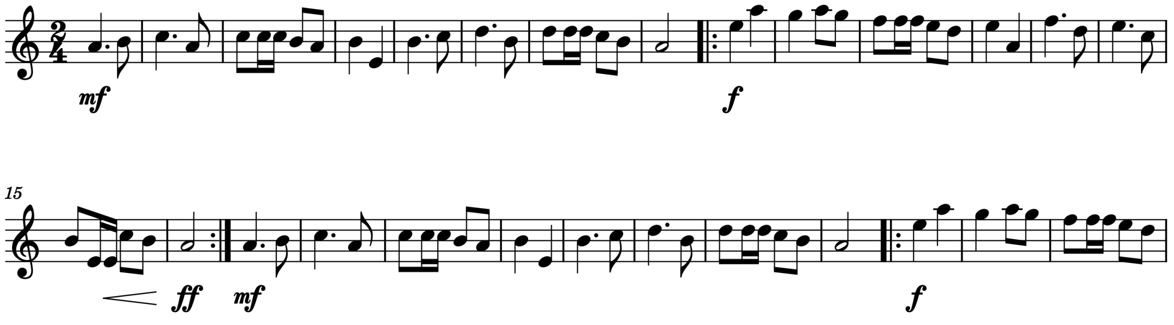

Figure 2.2: Example melody.

Figure 2.2 depicts a traditional folk song melody line with ascending and descending pitches, based on a *minor scale*.

2.1.3 Harmony

Harmony, in general, is a term for organizing the superposition of sounds in a meaningful way in order to give the music a sense of direction, tension and expression through consonance and dissonance.

- chord - three or more simultaneous pitches. The most common chords are triads consisting of the root note, a minor or major third and a perfect fifth - these are the building blocks of the most common types of chords, that is *major* and *minor* chords. Other types of chords include diminished, augmented, suspended, seventh and ninth chords, among many others [Benward, 2014].

- inversion - the root note of a chord may or may not be played as the lowest note of the chord. A chord inversion occurs when other notes than the root note are played as the bass note.

- tonality - the arrangement of pitches and chords around a certain hierarchy of relations and directionality. Tonality organizes harmonic idioms and their function in music and is the base of concepts as the *key* of a piece is written in [Shepherd et al., 2003]. Historically, because of different tunings of instruments and different measurements of pitch, musical keys have tended to widely differ in their perceived sound [Schubart, 1839].

- cadence - a progression of chords with a clear sense of finality, or otherwise punctuation of the flow of music.

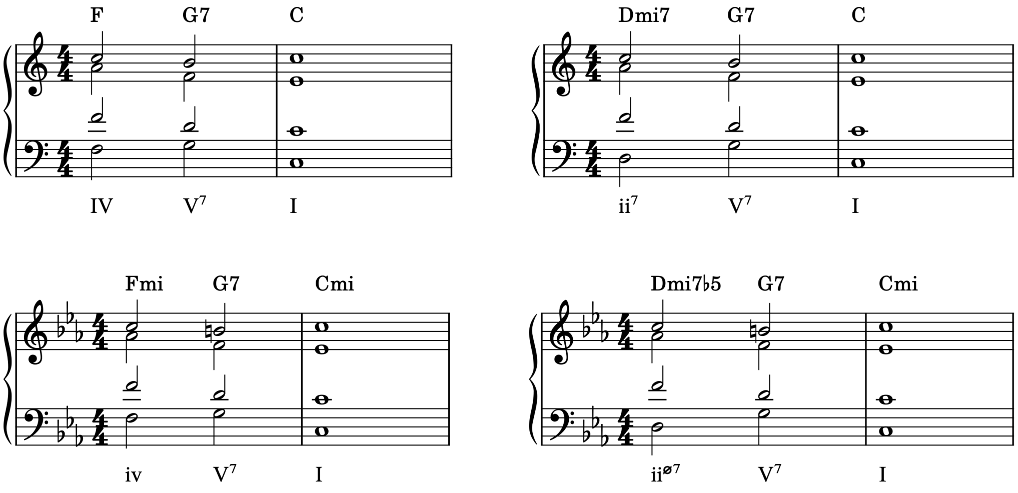

Figure 2.3: Example cadences. Source: *Marc Sabatella* [Sabatella, 2021]

Figure 2.3 depicts examples of cadences of chords, which give a sense of musical resolution and direction.

A particular disctinction in terms of harmony has to be given to jazz music and jazz harmony. Jazz utilizes a plethora of very advanced harmonic concepts, with complex chords reaching far beyond the basic triads, non-typical scales and frequent key changes, among other [Levine, 2011], on top of a framework of improvisation and very high rhythmic variety. The contributions of such artists like John Coltrane, Miles Davis, Thelonious Monk or Bill Evans (just to name a few) have deeply pushed the boundaries of music artistry and are a subject of deepened studies up to this day.

Besides what can be found in jazz music, other high-level harmonic concepts, like modulation, polytonality, atonality and more advanced harmonic concepts, although fascinating and powerful in terms of musical expression, are left to the reader for further inspection [Laitz, 2008], [Shepherd, 2003], [Mark and Gary, 2007], [Russo, 1997], [Benward, 2009].

2.1.4 Structure

The structure of a musical piece is a higher level concept which encapsulates the way the whole piece is organized in terms of rhythm, melody, harmony, as well as production and orchestration. This also includes the idea of repetition in music.

- motive, figure - succession of notes, smallest structural units of music possessing thematic identity [White, 1994], often repeatable and recurring.
- phrase - a self-contained musical thought, having its own musical sense. Often composed

of motives and figures. Phrases may span multiple bars.

- sections - a complete musical idea and a major structural unit of music. Sections, unlike whole pieces, are not independent and usually many sections constitute a whole musical piece. In orchestral music, sections of a bigger piece are usually called movements. In popular music, the typical sections of a song are the verse and chorus, often with the use of a bridge, instrumental interlude or instrumental solo.

Two of the most common structures in popular music include the *twelve bar blues* and the *AABA song structure*. Certain musical genres, like electronic dance music, often display a different approach to sections: with little harmonic variance, the music is divided mostly by rising and falling dynamics, with the most exciting section usually called the *drop*.

2.2 Music representations

Throughout history, there has been a number of ways of representing and storing music. Written sheet music and tabulature representations have evolved naturally in order to pass music down to next generations, as well as instruct performers on how a piece should be sung or played. Sheet notation includes elaborate markings and a variety of symbols to depict musical features. The possibilities of recording and processing audio, along with the rise of electronic music instruments and computers, have since brought new, digital notations into practice, while digital signal processing methods have brought many handcrafted, computed features, suitable for tasks of music information retrieval and applications of artificial intelligence.

2.2.1 Musical notations

Sheet music

In ancient times music was passed down mostly orally, although certain notions of ancient written notations exist [1]. A standardized notation, however, was introduced around the XI century by a Benedictine monk named Guido d'Arezzo. The notation was composed of what can be considered as the prototype of the modern staff notation, with multiple staff lines representing the relationship between pitches. Square neumes were used to indicate the general rhythm and shape of the notes to be sung.

[1] The presented discussion is concerned mostly with the development of Western, mostly European music, and does not take into consideration the musical development in Asian, African and other cultures and their history, which although rich and interesting, constitutes a different, self-contained research area.

From there, the standard staff notation has evolved. During the Renaissance, polyphonic forms (and music in general) have become more popular, giving also start to some music print. A tabulature notation was also sometimes utilized. By the XVIII century, the music notation that we know today and that is currently widely used by musicians throughout the world, has already been quite well established [Gould, 2016].

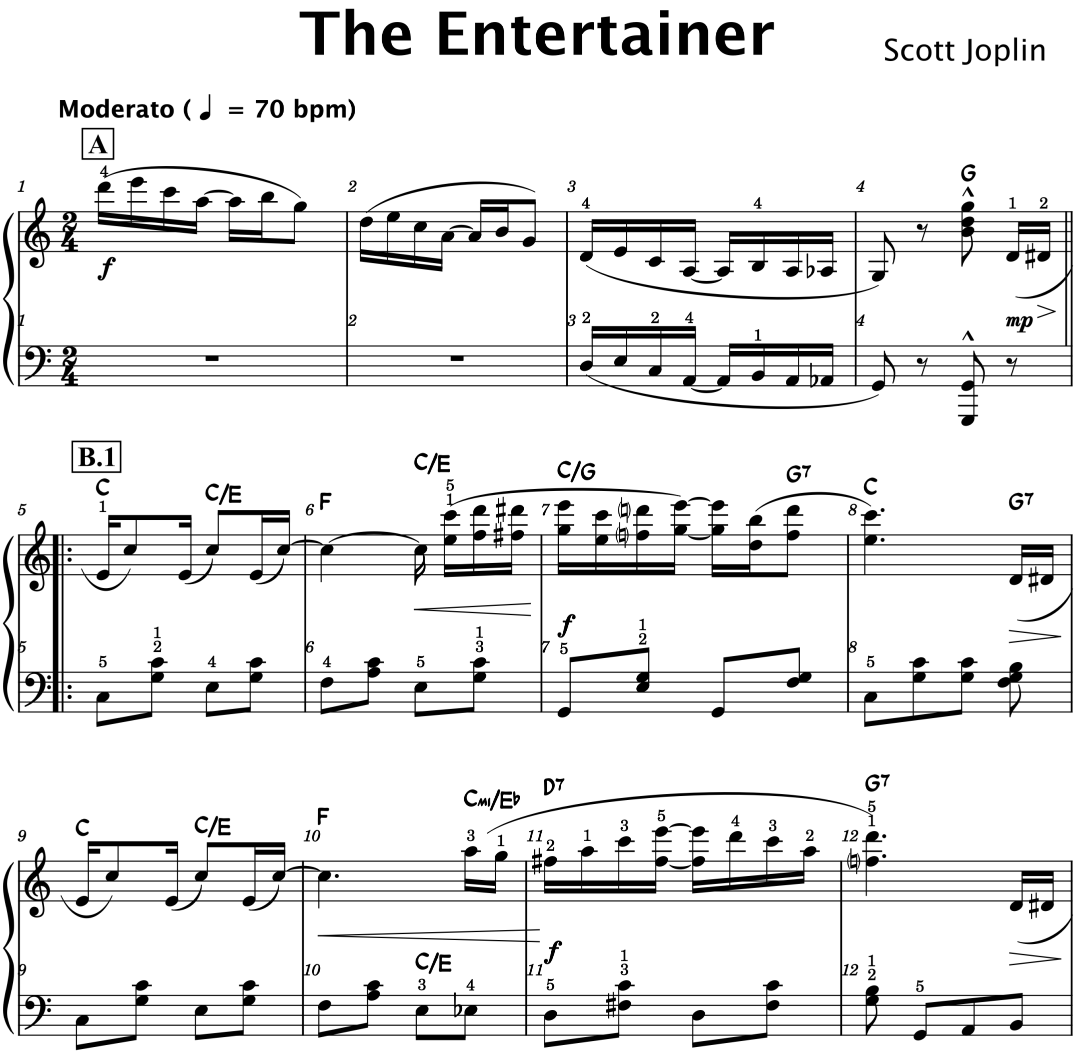

Figure 2.4: Modern sheet music, *The Entertainer* by Scott Joplin

Figures 2.4 and 2.5 depict examples of sheet music with the standard modern notation. These examples encompass a variety of musical features: time-pitch relationships, rhythm and dynamics, key, tempo and other indications on how the music should be performed.

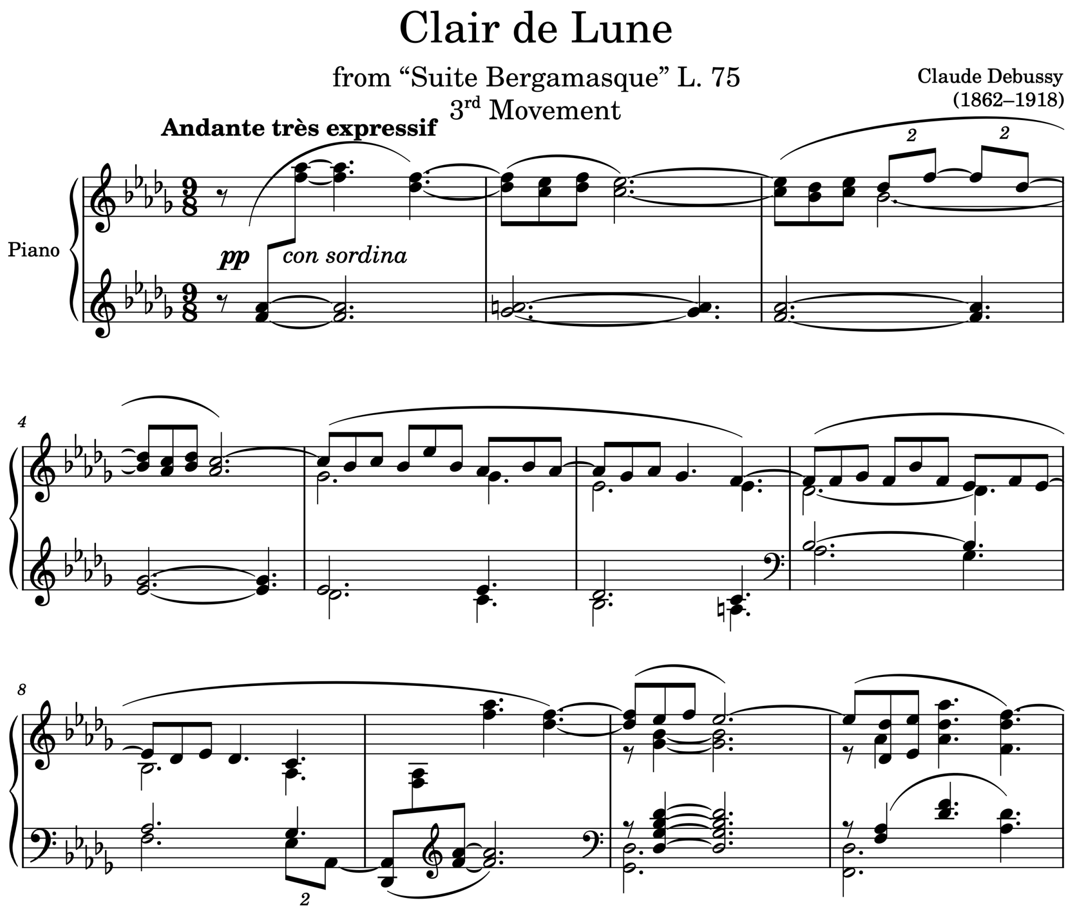

Figure 2.5: Modern sheet music, *Clair de Lune* by Claude Debussy

MIDI data

MIDI (*Musical Instrument Digital Interface*) is a standard describing a binary communication protocol and necessary digital and hardware interfaces for communication of musical instruments, computers, sequencers, synthesizers and other audio devices. It has been developed in the early and mid 80's by a panel of musical technology companies and instrument manufacturers, including Roland, Yamaha, Sequential Circuits and Korg, among others [Smith and Wood, 1981] [Moog, 1986].

MIDI carries up to sixteen channels of information with event messages. Each of the channels carries separate steering messages and can be routed to different devices. The messages are processed in a sequential manner and contain information about the channel, time of occurrence and event type. Channel voice messages are used for performance data and can describe when a note has started (*note-on message*) or ended (*note-off message*), along with the note's pitch and velocity. Note messages do not however contain information about the type of musical instrument used - the instruments are bound to particular channels and can be changed with a *program change* message. For further modification of an instrument's parameters, a control change message can be used.

MIDI also has its associated file format, used to store MIDI data in a lightweight, universal form. Music stored in the MIDI format is deplete of sound, as it contains recorded messages (as opposed to recording a particular audio signal), but can be processed by hardware instruments and digital audio workstations. MIDI files may also contain certain metadata about the piece, like the title, author or genre.

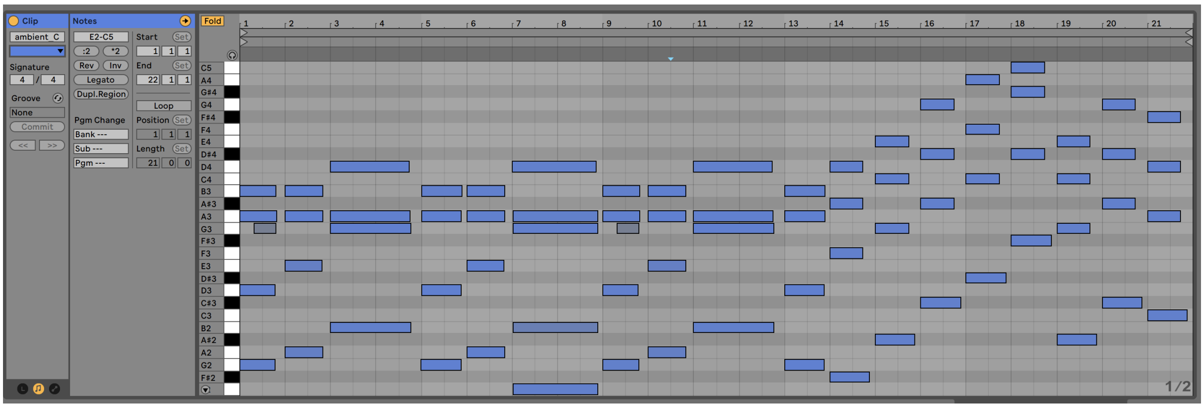

Figure 2.6: MIDI file visualized as a piano roll in Ableton Live software.

Figure 2.6 presents a MIDI file read into Ableton Live, a popular modern digital audio workstation and music production software. The MIDI is visualized using a folded piano roll representation.

ABC notation

ABC notation is a text format developed by Chris Walshaw [Walshaw, 2021]. It is used mostly for Western European traditional folk music and is easy to convert into MIDI or sound files with dedicated software. The author also maintains a collection of around 640,000 tunes in ABC notation on his website, available for free download. Files in the ABC notation contain the music, as well as metadata such as the title, meter, default note length, type of tune and key. An example song in ABC notation in shown in Figure 2.7, with the output converted into sheet music shown in Figure 2.8.

ABC notation has also found some usage in artificial intelligence applications, for instance it has been used in the work [Sturm et al., 2016] together with LSTM recurrent neural networks for transcription and generation of Celtic and Morris folk music.

```
X: 1
T: Old Town
R: reel
M: 4/4
L: 1/8
K: Gmaj
:DG GG|F/G/A/B/c2|BG GB|A/G/F/G/A2|DG GG|F/G/A/B/c2|BG A/G/F|G4:|]
:dB dB|c/B/A/B/c2|BG GB|A/G/F/G/A2|dB dB|c/B/A/B/c2|BG A/G/F|G4:|]
```

Figure 2.7: Example of music written in ABC notation [Walshaw, 2021]

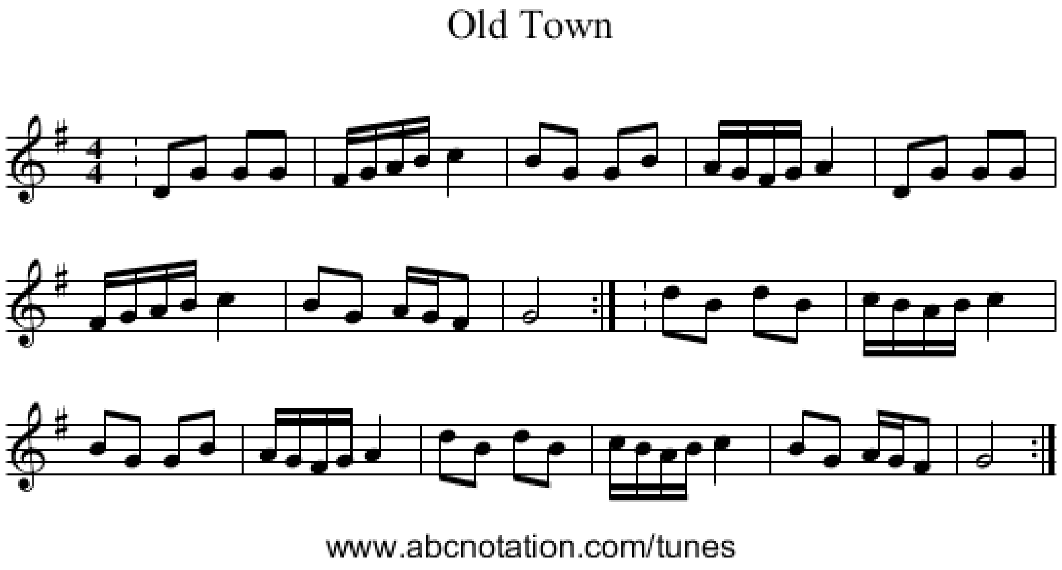

Figure 2.8: Music written in ABC notation (from Figure 2.7) compiled down to sheet music [Walshaw, 2021]

Other

Other notations include approaches such as WEDELMUSIC [Bellini and Nesi, 2001] and MusicXML [Good, 2001], based on XML and used internally by selected music software. Another notation is the text-based GUIDO notation, which is designed for readability and has also been used in some research efforts of music information retrieval [Hoos et al., 2001].

2.2.2 Computed features

Low level, computed musical features correspond closely to the analysis of the raw audio signal and involve digital signal processing (DSP) methods, abstracting a section of raw sampled audio into attributes with meaningful values. These attributes may then be used as representations for the needs of machine learning algorithms. The usage of such compressed representations reduces much of the compute associated with processing raw audio with standard sample rates of 44.1kHz or 48kHz.

Works like [Lerch, 2012], [Eyben, 2015] and [Knees and Schedl, 2016] provide in-depth descriptions of many of these attributes in a music processing context, often with particular use cases in music information retrieval tasks. Furthermore, [Murauer and Specht, 2018] show that many of these computed features are still relevant in the era of deep learning.

In addition to the raw computed values, visualizations of low level musical features enable the usage of deep learning methods which were previously found to work well in image processing tasks. Key low level musical features and representations are therefore presented in the following section, along with their applications.

Amplitude envelope

The amplitude envelope describes the max amplitude value (noted as $s(k)$) for of all the samples in a given frame t for a frame of size K and can be defined as [Knees and Schedl, 2016]:

$$AE_t = \max_{k=tK}^{(t+1)\cdot K-1} s(k) \tag{2.1}$$

The amplitude envelope informs us about general loudness and may be used for simple onset detection, but is sensitive to outliers.

Root-mean square energy

The root mean square energy of an audio signal relates to the perceived sound intensity and is less sensitive to outliers than the amplitude envelope. It has been used for instance for audio segmentation [Caetano et al., 2010] and can be defined as [Knees and Schedl, 2016]:

$$RMSE_t = \sqrt{\frac{1}{K} \sum_{k=tK}^{(t+1)\cdot K-1} s(k)^2} \tag{2.2}$$

Zero crossing rate

The zero crossing rate describes the number of times the amplitude value will change its sign in the frame t of size K:

$$ZCR_t = \frac{1}{2} \sum_{k=tK}^{(t+1)\cdot K-1} |sgn(s(k)) - sgn(s(k+1))| \tag{2.3}$$

In analogy, a *mean crossing rate* can be defined as a rate of changes through the mean of the signal. Zero and mean crossing rates are used in speech recognition and music information retrieval for detecting percussive sounds. It can also be used to estimate pitch for simple monophonic music signals [Knees and Schedl, 2016] and to discriminate periodic signals from noise [Virtanen et al., 2018].

Band energy ratio

The band energy ratio is a feature computed in the frequency domain. It measures how dominant low frequencies are in the signal. Given the split frequency band F and magnitudes $m_t(n)$ of the signal in the frequency domain at frame t in band n, the band energy ratio can be defined as:

$$BER_t = \frac{\sum_{n=1}^{F-1} m_t(n)^2}{\sum_{n=F}^{N} m_t(n)^2} \tag{2.4}$$

It has been used for music genre classification [McKinney and Breebaart, 2003] and speech versus music discrimination [Lavner and Ruinskiy, 2009].

Spectral centroid

The spectral centroid represents the center of gravity of the magnitude spectrum and is used to determine the brightness of a sound [Zwicker and Fastl, 2013]. It can be therefore related to the musical timbre and used for instrument classification. It can be defined as:

$$SC_t = \frac{\sum_{n=1}^{F-1} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)} \tag{2.5}$$

Spectral spread

Spectral spread describes the spectral range around the centroid. It can be interpreted as variance from the mean frequency in the signal [Knees and Schedl, 2016] and defined as:

$$BW_t = \frac{\sum_{n=1}^{N} |n - SC_t| \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)} \tag{2.6}$$

Spectral spread can be used for descriptions of musical timbre [Lerch, 2012].

Spectral flux

Spectral flux describes the change in power spectrum between consecutive frames. It has been used for musical onset detection [Dixon, 2006] and given the normalized frequency distribution D_t in frame t, can be defined as [Knees and Schedl, 2016]:

$$SF_t = \sum_{n=1}^{N} (D_t(n) - D_{t-1}(n))^2 \tag{2.7}$$

Different normalization coefficients may be used for computing D_t. An unnormalized version of spectral flux can be also computed [Eyben, 2015].

Note onsets

The note onset is the beginning of the musical note. Onsets may be slow (as in bow instruments played legato) or fast (sharp percussive sounds - in this case the sounds will also have a clearly defined transient) [Virtanen et al., 2018]. Onsets are related with the logarithm of the attack time [Peeters et al., 2011] and can be used as a feature to determine the rhythmic density, character and style of the music.

The issue of onset detection is an open problem in artificial intelligence applied to music. Since deep learning has begun gaining in popularity, convolutional neural networks have successfully been used for onset detection [Schlüter and Böck, 2013] [Schlüter and Böck, 2014], including onset-based automatic transcription efforts by Google Magenta with the additional usage of two stacks of networks and LSTM recurrent networks [Hawthorne et al., 2017].

Spectrograms

Spectrograms are a visualization of the spectrum of frequency of a signal in time and are represented as heatmaps. In order to obtain a spectrogram, the short-time Fourier transform (STFT) is computed [Allen and Rabiner, 1977]. The STFT is a discrete Fourier transform applied to subsequent time windows of a signal and given a windowing function $w[k]$ (ie. Hann, Blackmann, rectangular), for a hop equal to the length of the frame N and frame t, can be defined as: [Virtanen et al., 2018]:

$$STFT[t,f] = \sum_{k=0}^{N-1} w[k]x[tN+k]e^{\frac{-i2\pi kf}{N}} \tag{2.8}$$

It is also common to introduce overlap by choosing a smaller hop size. An inverse of the STFT also exists and is used to restore the frames of the signal.

The spectrogram is usually computed for a decibel logarithmic scale and can be defined as:

$$spect[t,f] = |STFT[t,f]|^2 \tag{2.9}$$

The values of the STFT matrix may be adjusted for the human hearing perception. Several methods of such adjustment exist, for instance using the mel scale, constant Q transform

[Brown, 1991] [Schörkhuber and Klapuri, 2010] or Gammatone filters. Figure 2.9 presents a mel-frequency spectrogram extracted using the *librosa* library.

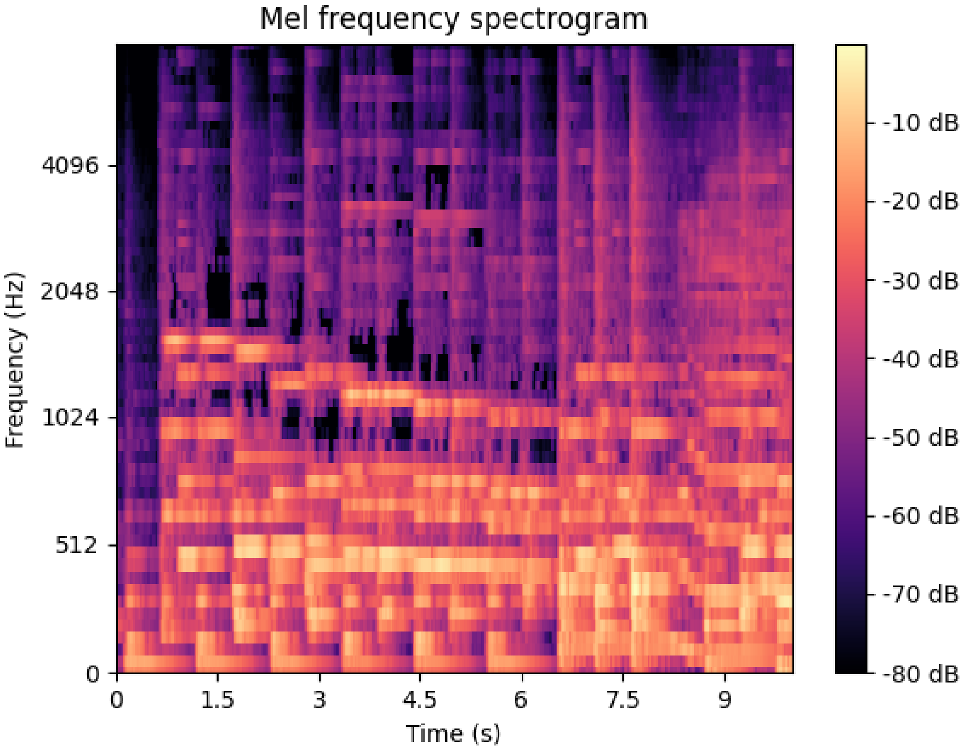

Figure 2.9: Spectrogram.

Chromagrams

Chromagrams are a visualization of the chroma feature, which is a twelve-element vector describing the energy carried by each pitch (C, C#, D, D#, E, F, F#, G, G#, A, A#, B). For the purpose of this representation, pitches from different octaves are binned together, as the octave interval bears the most similarity as perceived by the human ear [Ellis, 2007a]. Figure 2.10 presents an example of a chromagram.

The chromagram visually bears significant resemblance of the sheet music of the corresponding audio excerpt [Felix Weninger, 2012]. This characteristic has been used, for instance, for cross correlation according to beats in order to identify cover versions of the same songs [Ellis and Poliner, 2007].

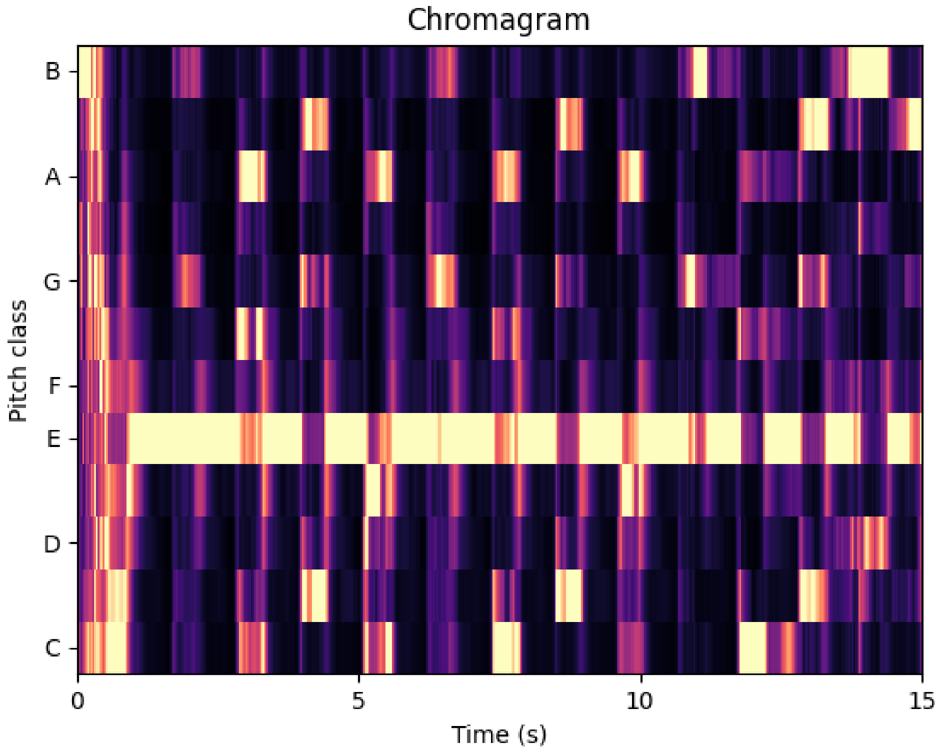

Figure 2.10: Chromagram.

Tonal centroids (Tonnetz)

Tonnetz is a pitch space defined by a network of relationships between musical pitches in just intonation proposed by Euler [Euler, 1739]. Close harmonic relations (like major and minor thirds and perfect fifths) are short distances on an infinite Euclidean plane. Chords therefore become geometric structures on the plane. With enharmonic and octave equivalence, the Tonnetz plane can be wrapped into a hypertorus, where chords can be described by their 6-dimensional centroids, which has been used for automatic chord detection [Harte et al., 2006].

Figure 2.11 presents a visualization of tonal centroids for a musical audio excerpt of an orchestral piece extracted using the *librosa* library. The x axis represents time (in seconds), while each of the steps on the y axis represents one of the six dimensions: the chroma feature projected onto a basis of two-dimensional coordinates for minor and major thirds and the perfect fifth [Harte et al., 2006], with heatmap coloring of intensity. Tonnetz features have also been used for modelling and generation of music with the usage of deep neural networks [Chuan and Herremans, 2018].

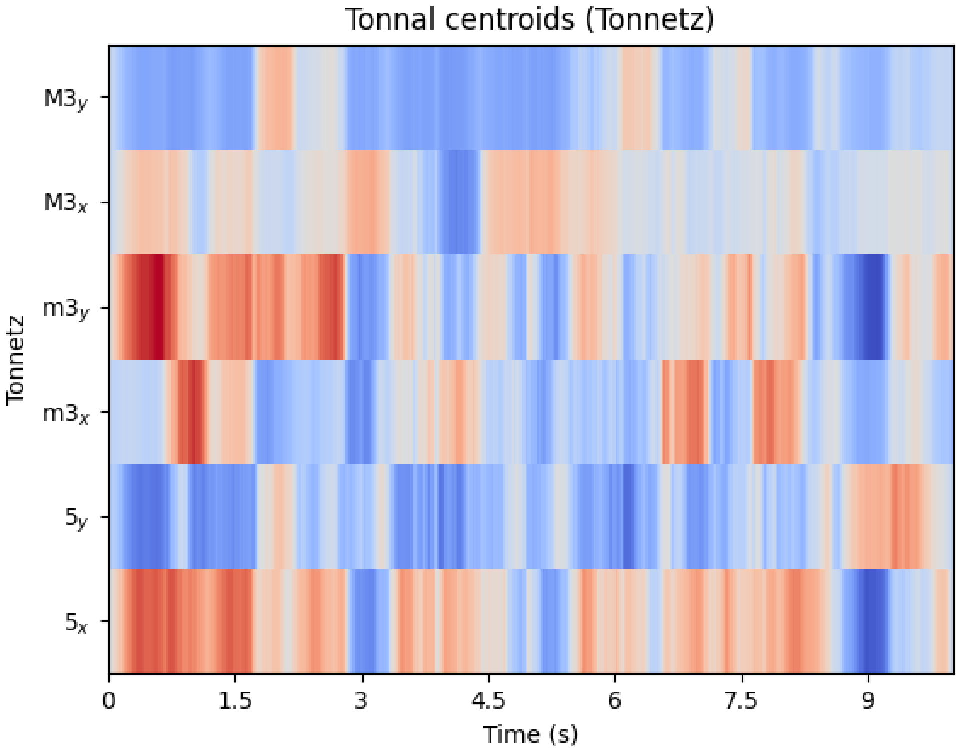

Figure 2.11: Tonnetz.

MFCC

Cepstral analysis has been proposed in order to analyze echos and reflections in the signal and has since found many applications in audio analysis. The cepstrum can be defined as the result of computing the inverse Fourier transform of the logarithm of the estimated signal spectrum [Childers et al., 1977]:

$$CEP = DFT^{-1} log\{|DFT(t)|^2\} \tag{2.10}$$

When computing MFCCs (mel frequency cepstral coefficients), the Fourier transform of the windowed signal is computed and the powers of the spectrum are mapped onto the mel scale, which takes human psychoacoustics into consideration. The logarithms of the powers at each of the mel frequency bands are then processed with a discrete cosine transform. The amplitudes of the resulting spectrum are the MFCCs, which summed up give the mel-frequency cepstrum of the signal.

Figure 2.12 presents a sample visualization of MFCCs for an audio signal, extracted using the *librosa* library. The x axis represents time (in seconds), while the y axis visualizes subsequent cepstral coefficients with heatmap coloring of intensity.

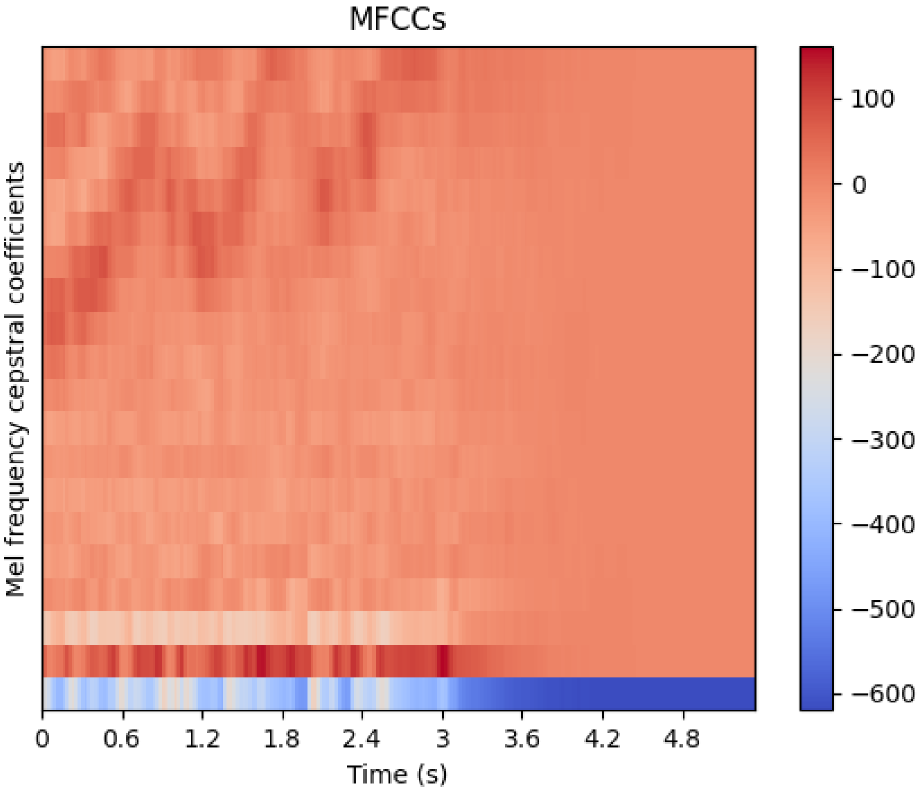

Figure 2.12: Visualization of first 20 Mel-frequency cepstral coefficients for an audio sample of a trumpet.

In case of human speech, the first MFCCs carry most of the information about the spectral envelope (and in turn - the formants), while the further ones model the glottal pulse. Typically up to 13 MFCCs are used. MFCCs have become a very popular feature for usage in speech recognition [Ganchev et al., 2005].

In the case of music, MFCCs may be used to describe the general characteristic of the timbre of a musical instrument or whole recording. However, there is no lossless inverse operation to the computation of MFCCs, therefore they are not the most suitable feature for signal reconstruction and generation.

Summary

As is clearly visible, there is no single, agreed upon representation of music suitable for the needs of artificial intelligence. Various representations of music have been used in different stages of research described in the following chapters.

Spectrogram and chromagram graphical representations were found to be useful in particular together with convolutional neural networks for music classification, as described in

Chapter 5. Digital signal processing, with the particular usage of the STFT, has been used for style transfer in order to maintain information about the input signal, as described in Chapter 3. Furthermore, compressed MIDI visualizations have been used for musical phrase generation, as described in Chapter 4. A strong background of basic musical concepts was also necessary for musically aware analysis, with sheet music and MIDI representations of the obtained results presented where applicable.

Chapter 3

Style transfer

Style transfer is the idea of superimposing the style of a particular piece of art onto another one. Example questions style transfer tries to answer may sound *"how would Mona Lisa look if it was painted by Kandinsky?"* or *"how would Mozart's 'Rondo alla Turca' sound if it was composed by Debussy?"*. It has been an active research area in artificial intelligence, especially after introducing the usage of neural networks for graphical style transfer in 2015 [Gatys et al., 2015b]. The following chapter describes the background of this issue and presents experimental work conducted in musical timbral style transfer, along with a proposition of a new solution and experimental results.

3.1 Style transfer in graphics

The usage of convolutional neural networks for style transfer in graphics has been proposed by Gatys in [Gatys et al., 2015b] and subsequently improved in the works [Gatys et al., 2015a] and [Gatys et al., 2016]. There are several difficulties to consider and the main one is to distinguish the content of the image from its style. In the domain of images this task is quite easy for the human perception, as contents are naturally perceived as shapes and objects presented in the artwork, while the style is the total of artistic and creative techniques used to depict the content.

In the domain of neural networks, feature maps of convolutional neural networks provide a good representation of the features of the input image, which can be understood as the content of the image. Convolutional networks learn the image features in a hierarchical manner - while initial layers operate on pixels, the deeper layers will learn more abstract spatial information. A pre-trained VGG [Simonyan and Zisserman, 2014] architecture was initially proposed for the extraction of content, with a selection of layers for content reconstruction. A mean-squared error may then be used to determine the difference between an input image and the representation of its content. Activations sampled mostly from initial and middle layers of the convolutional

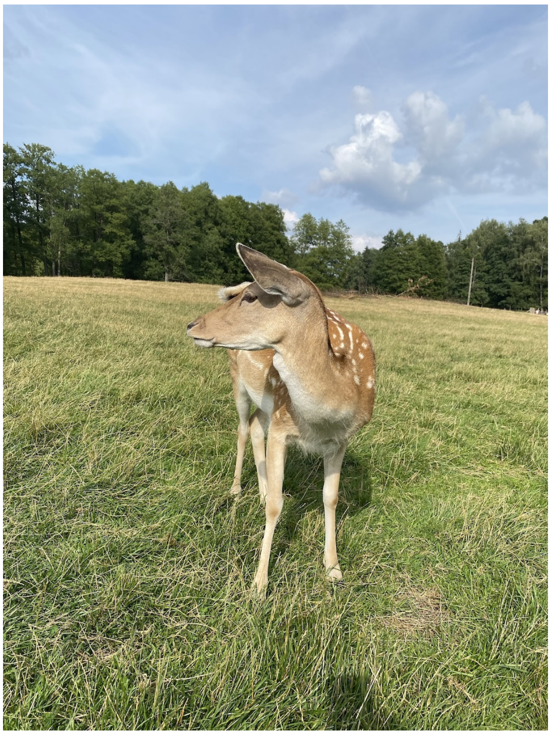

Figure 3.1: Original photograph for style transfer, taken from private photo collection of the author.

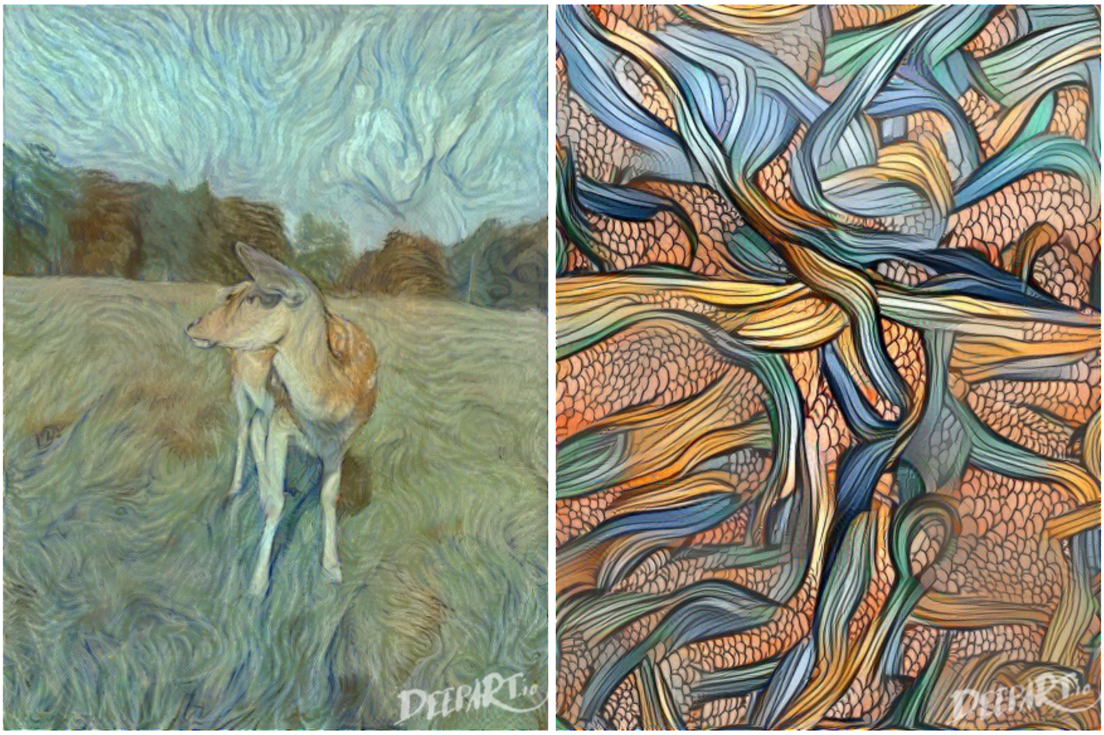

Figure 3.2: Examples of different styles superimposed on the original photograph using the free API available at `https://deepart.io`, provided by Gatys et al.

network are then encoded into a Gram matrix of flattened feature vectors to extract the *texture* of the image, which is the perceived style. Feature maps from several conovlutional layers may be used for extracting and encoding the style information. Finally, a weighted loss of the content and style reconstruction losses is computed in order to parametrize the smoothing effect of the style transfer [Gatys et al., 2015b]. For superimposing the style of an artwork $\vec{a}$ over a photograph $\vec{p}$, the minimized loss function with smoothing weights α and β is:

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x}) \tag{3.1}$$

The crucial conclusion of the initial works on image style transfer was the fact that the representations of content and style in convolutional neural networks are separable. This, along with very convincing images presented by the authors, has paved the way for further research into the domain.

3.2 Style transfer in music

Neural style transfer in music is quite a new idea, gaining interest in recent years thanks to the advances in image style transfer. Due to music having a strong sequential relationship and multiple possible representations carrying different amounts and qualities of descriptive information, there have been attempts of providing a conceptual framework for the requirements of music style transfer. [Dai et al., 2018] propose the following summary of representations:

Table 3.1: Music representations according to [Dai et al., 2018]

	sensory system	unique features	scale of measure	type of data
score (top)	visual	structure and symbolic abstraction	all	discrete
control (middle)	motor	expressive timing and dynamics	interval and ratio	mixed
sound (low)	auditory	acoustic details	ratio	continuous

Worth noting is the fact that MIDI [MIDIAssociation, 1999] is given as an example of performance control. They also formulate that the content of music is an obtained via abstraction from a lower level to a higher one, while style is created via realization from a higher level to a lower one.

Furthermore, [Dai et al., 2018] decompose musical style transfer into subproblems:

- **timbral** style transfer applies to the low level representation of sound, with modification of the timbre in a meaningful way with preservation of performance control
- **performance** style transfer applies to the middle representation of performance control with preservation of the implicit score. A performance control encodes an interpretation of the cor- responding score, rely on which a performer turns the score into performance motions. The main character of performance control is the nuanced detailing of timing and dynamics (as in MIDI), with flattening and implicit treatment of structural information when compared to the score representation.
- **composition** style transfer applies to the top score representation, with a meaningful modification of the general score information with preservation of the melody contour and underlying characteristic of the harmony.

3.2.1 Related works in timbral style transfer

Pioneering studies

The area of musical timbral style transfer has already gained some interest, although the number of related works is currently small when compared to image style transfer. The authors of [Dai et al., 2018] cite [Verma and Smith, 2018] and [Engel et al., 2017] as the pioneering studies on the subject.

The former is a short paper proposing a the usage of a modified convolutional AlexNet [Krizhevsky et al., 2012]) model, with smaller receptive fields and additional loss terms. The authors use spectrograms as their music representation and train the network to transfer the timbre of a music fork onto a harp and a singing voice onto a violin. Audio samples are not provided, although the authors display animations of their solution performing bandwith compression and expansion of the signals.

The latter, on the other hand, is a proposition by Google and DeepMind. The authors base their work on WaveNet [van den Oord et al., 2016], a very robust convolutional, probabilistic and autoregressive model with dilated convolutions created with a focus on state of the art speech generation and synthesis, working on raw audio signals. At each step, the next sample of raw audio is predicted from a fixed-size input of prior sample values. The probability of generated audio x is expressed as a product of conditional probabilities:

$$p(x) = \prod_{i=1}^{N} p(x_i|x_1,...,x_{N-1}) \tag{3.2}$$

The authors of [Engel et al., 2017] propose a WaveNet autoencoder for raw waveforms, which conditions an autoregressive decoder on temporal codes. The bottleneck hidden layer of the autoencoder provides a timbre representation. A heuristic loss of weighted MSE is used, with high weights for the lower frequencies and a weight of 1 above 4kHz. They also propose NSynth, a large dataset of single, four-second notes with different timbres of harmonic instruments. The limitation of the temporal context due to the chunk size of the used training audio is also noted. WaveNets have achieved state of the art performance in tasks like speech generation, as the usage of raw audio signal allows to maintain full information about the signal, but in turn this approach requires massive amounts of computational power to process.

MoVE

Another approach to the discussed issue is the application of VAEs (*variational autoencoders*) like MoVE: modulated variational autoencoders in a one-to-one and many-to-many setting, described in [Bitton et al., 2018]. MoVE consists of convolutional, dense and *FiLM* layers [Perez et al., 2018] in an encoder-decoder architecture and attempts training on a dataset consisting of multiple instruments. As an input, audio embeddings are given along with the instrument class. The Non-Stationary Gabor Transform [Balazs et al., 2011] is used to obtain the time-frequency representation of the audio signal. However, the method introduces some loss of information, making it difficult to reconstruct the original signal.

Universal Music Translation Network

[Mor et al., 2018] by Facebook AI Research proposes the Universal Music Translation Network, a network capable of style transfer between instruments and classical composers. The authors use raw audio signals as the chosen representation of music and augment them by randomly de-tuning short segments of less than half a second by up to half of a semitone. A WaveNet-like convolution encoder is used along with multiple WaveNet decoders, conditioned on the latent representation produced by the encoder. The network was trained for 6 days on 8 Tesla V100 GPUs.

TimbreTron

[Huang et al., 2019] propose a *"WaveNet(CycleGan(CQT(audio)))"* model and use it for timbral style transfer between piano, flute, harpsichord and violin. The model is called TimbreTron. The authors apply image-like style transfer to a time-frequency representation of audio using *rainbowgrams* [Engel et al., 2017] produced using a constant Q transform. An example open source implementation of rainbowgram extraction written in Python is available at [tarepan, 2018]. This representation was chosen because of the pitch equivariance and high frequency resolution at low frequencies of constant Q transform, which outperformed STFT when used with the CycleGAN. A conditional WaveNet is used to reconstruct waveforms from the constant Q transform representation. Validated with a human survey using Amazon Mechanical Turk, TimbreTron was found to be able to perform recognizable transfer of timbre with preservation of the musical content for monophonic and polyphonic signals.

Other approaches

Another approach to the style transfer is the usage of relativistic-average generative adversarial networks [Jolicoeur-Martineau, 2018] with a complex, compound representation of music using Mel-spectrograms, MFCCs, spectral difference, and spectral envelope. In [Lu et al., 2018], the authors use this representation to perform multi-modal one-to-many style transfer. The work [Brunner et al., 2018] on the other hand proposes the use of symbolic MIDI data representation and a variational autoencoder in an attempt to transfer the dynamics and instrumentation of music.

The related works in style transfer often cite obtaining a good quality of the reconstruction of signal as a difficult task. [Mital, 2017] states that using only the modulus of the STFT introduces additional problems with signal reconstruction, as phase information is lost. A network deplete of such acoustic data will not be able to model, for instance, a vibrato effect, natural for many instruments. The phase may be reconstructed upon partial data with a trade-off of additional noise. The authors propose to use direct outputs of the Fourier transform and conclude that using the real and imaginary part works better than using the modulus and argument for style transfer.

[Dai et al., 2018] conclude that most of the existing efforts in disentanglement of timbre and performance control have not yet been very successful, especially for bigger lengths of the audio excerpts (like full musical phrases and pieces with a defined structure). The music created by the provided early studies is still quite immature when compared to music created, performed, produced and processed by humans. The research area is clearly open for more ideas, especially in terms of the used data, chosen representations and efficiency of the developed solutions.

3.3 Proposed solution

The following section describes the proposition of a new solution for timbral style transfer and expands on the results presented in [Modrzejewski et al., 2021]. The proposed recurrent autoencoder network performs style transfer between pairs of instruments, preserving the content of the provided input music and modifying the timbre. For the purpose of the performed experiments, piano to guitar style transfer has been performed and tested, due to relatively high availability of pieces performed originally on piano or guitar among publicly available data. The proposed method can however be used for timbral style transfer between any pair of instruments. The method preserves the content of the music, therefore is suitable for creating new sound effects, for instance via the transfer of vibrato from one instrument to another. The proposed method successfully performs timbral style transfer between the chosen instruments, is efficient and requires much less computational power than WaveNet based approaches mentioned in the previous section.

3.3.1 Data

[Engel et al., 2017], among others, highlight the ongoing lack of datasets and sparsity of existing ones, suitable for experimentation and benchmarking artificial intelligence solutions applied to music. In order to tackle this problem, a streamlined method of generating paired training datasets by synthesizing MIDI data into sounds is proposed. It is very close to the way MIDI data would be used by musicians in a professional music production setting, thus bringing the context of the experiments closer to the function of the MIDI data. Also, sufficient pure MIDI datasets already exist and have been made public.

The following two MIDI datasets have been used for the experiments:

- LMD-matched subset of the *Lakh MIDI Dataset v0.1* [Raffel, 2016], consisting of 45129 samples of modern songs, including mostly western rock, pop and electronic music. The pieces in the subset have been verified and paired with the Million Song Dataset [Bertin-Mahieux et al., 2011], ensuring that the MIDI files are not randomly generated, but arranged from the list of a million of actual modern songs.

- 130,000 MIDI File Collection [midi_man, 2019], consisting of 130,000 MIDI songs of various genres and performed on various instruments. The collection also includes soundtracks from films and video games.

Exploratory data analysis and pre-processing

A very small percentage of the data files were found to be corrupt MIDI - these were removed from further processing. For the purpose of the experiments, all of the remaining MIDI files were compared using their MD5 hash in order to eliminate direct duplicates. This allowed to remove exact copies of the same song, but still the same tune may be included more than once in the final dataset, as different versions of the tune in terms of key or arrangement may exist. The presence of such versions will also be beneficial to the network's capabilities for generalization.

Furthermore, a subgroup of songs performed in at least 95% on a single instrument was selected. This was possible by reading all of the MIDI messages in the songs, measuring the time of all of the note messages and computing the share of each instrument, according to instruments included in the General MIDI [MIDIAssociation, 1999] standard. Two possible erroneous edge cases regarding the *note-off* message have been identified:

- the *note-off* message may never occur for a particular note,
- the *note-off* message may occur before the *note-on* message.

In the former instance the notes have been assumed to end with the end of the whole tune. In the latter instance, the messages were discarded. In order to prepare paired datasets, *program change* MIDI messages were extracted and modified to represent the output instrument. In cases where no *program change* message was provided originally in a valid MIDI file, an adequate message would have been inserted as the first message.

Further MIDI processing

The collected dataset consists of 2000 MIDI songs. The MIDI was synthesized using *FluidSynth*, a state of the art programmatic synthesizer, which allowed to incorporate a workflow similar to one used in actual music production with VST plug-ins. *FluidSynth* uses SoundFont files as the base for synthesis - these files contain multiple samples of physical or digital musical instruments. Fluid Release 3 General MIDI Soundfont by Frank Wen has been used as the base SoundFont for the purposes of synthesis in the described experiments. It contains high fidelity samples of all of the instruments in the General MIDI standard.

After synthesis, the selected 2000 songs have translated to roughly 130 hours of music in an uncompressed WAV linear pulse code modulation format. The high quality audio is further converted into a time-frequency representation using the Short Term Fourier Transform (STFT) with a Hann windowing function [Harris, 1978]. In order to keep full information about the signal (including its phase), both the real and imaginary parts of the STFT are used for the purposes of the subsequent experiments. The resulting dataset was divided into train and test sets, with the test dataset representing 10% of the volume of the original set. Additional spectrogram representations, as presented in Figure 3.3, were also used for analysis and comparison purposes.

The desired output of the experiments was audio - in particular, lossless WAV format output. The STFT is invertible, therefore the signal could be recreated from the transform by using the

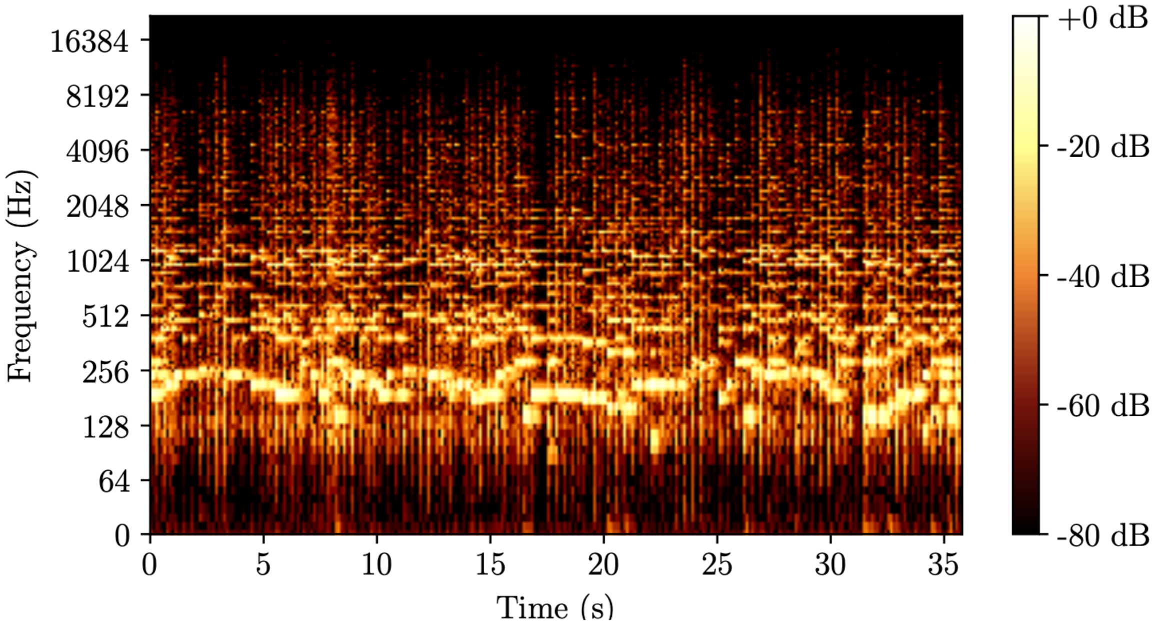

Figure 3.3: Example STFT spectrogram visualization of the input extracted using the *librosa* Python library.

inverse STFT, which was used to recover playable WAV files from the outputs of the networks in subsequent experiments. Although certain solutions for transcribing audio back into the initial considered format of MIDI already exist, this step of processing had no significance to the conducted experiments and therefore has not been performed.

3.3.2 Method

The proposed method is timbral musical style transfer with the usage of recurrent autoencoder neural networks. For this purpose, several models have been prepared, starting with a baseline autoencoder and improving it with subsequent experiments. The process of modelling the contents of the music is described along with reconstruction of the desired timbre. In order to train the networks on music of varying length, the tunes were divided into fixed packets, with only the endings of the tunes artificially filled with silence in cases where the length of the tune was not a multiplicity of the packet length. The networks were then trained on mini-batches [Masters and Luschi, 2018] of the preprocessed data.

Baseline autodencoder T0

The first issue when performing timbral musical style transfer is the preservation of the musical contents. We may assume that the input sample X and output sample Y come from different distributions sharing high-level characteristics. The common part of the distributions is the content of the music. When dealing with recorded music, the parameters of the audio signals separating contents from timbre are difficult to precisely indicate. The proposed baseline for further experiments is an encoder-decoder model reducing the attributes of the music to a latent space. The common part of X and Y, representing the content, is encoded by the encoder network, while the decoder network restores the desired timbre of the of the output instrument.

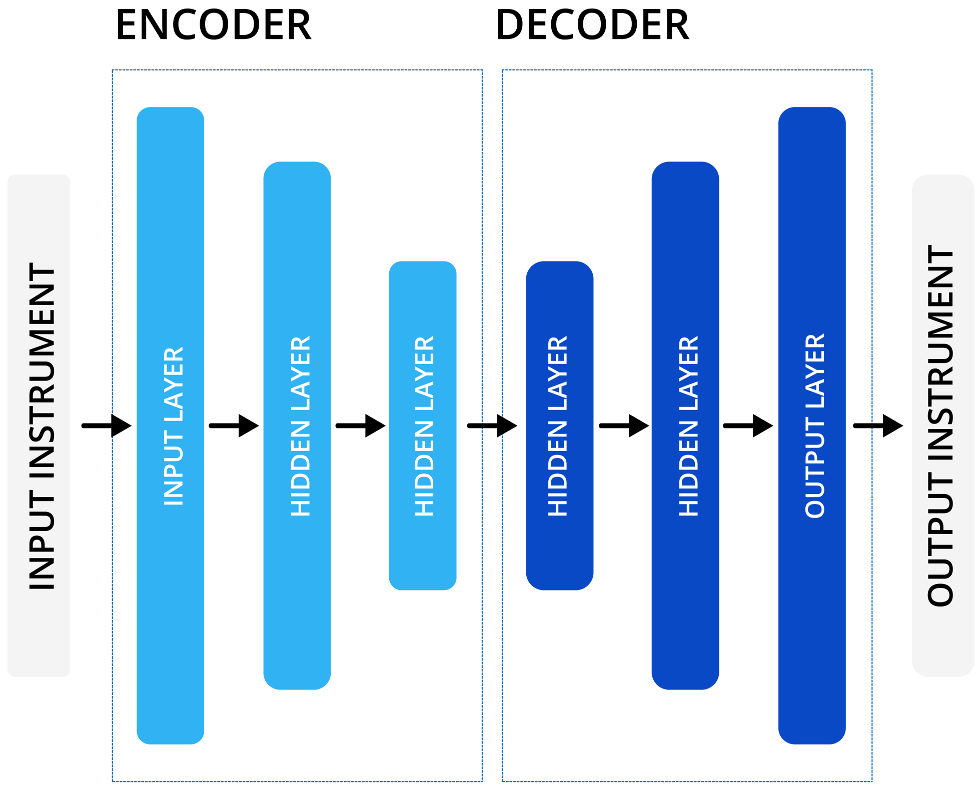

Figure 3.4: Baseline autoencoder $T0$. The input audio of is processed by the network to create samples corresponding to a different instrument.

The proposed baseline model is a symmetrical encoder-decoder, as conceptualized and presented in Figure 3.4. For the purpose of scoring and comparison, the model will be called $T0$. The model consists of fully connected layers with ReLU activations, a fixed-size bottleneck in the middle and projection layers corresponding to frequency bins of the STFT. The bottleneck stores data needed to transform samples between the selected domains. The used optimizer was ADAM [Kingma and Ba, 2014] with a learning rate of $1e-5$ and a β_1 of 0.9.

The sizes of the layers were as follows:

- Encoder:
  - 1024
  - 512
  - 256
- Decoder:
  - 256
  - 512
  - 1024
- projection layer: 2049
- projection layer: 2049

Recurrent autoencoder TLSTM1

Having established a baseline, all further experiments were conducted with utilization of recurrent layers. Figure 3.5 conceptualizes and presents the first proposed model of such type, called $TLSTM1$ for the purpose of scoring and further comparisons.

The conceptual base of LSTM (long short term memory) architectures has been proposed in [Hochreiter and Schmidhuber, 1997] and has since been developed and used with great success, for instance, in a similar encoder-decoder fashion for NLP problems [Wu et al., 2016]. Classic recurrent neural networks tend to introduce problems with vanishing and exploding gradients, where during backpropagation the gradients become near-nullable or grow to infinity, thus making training very difficult. LSTM cells are a solution in particular to the vanishing gradient problem, as they keep an additional cell state vector along with "forget" and "update" gates (as presented in Figure 3.6) allowing to model a wider time window and thus operate on longer context inputs.

The TLSTM1 model uses two initial dense layers of a fixed size of 4098, which is double the frequency bins of the STFT, connected to a recurrent encoder layer with 512 LSTM cells. The inputs encoded by the dense layers combined with the final cell states of the encoder are passed to the decoder. The final cell states of the LSTM encoder are therefore used as the initial cell states of the decoder. This bottleneck is therefore considered a timbre representation, as described for WaveNet autoencoders by [Dai et al., 2018]. The decoder's input is then split

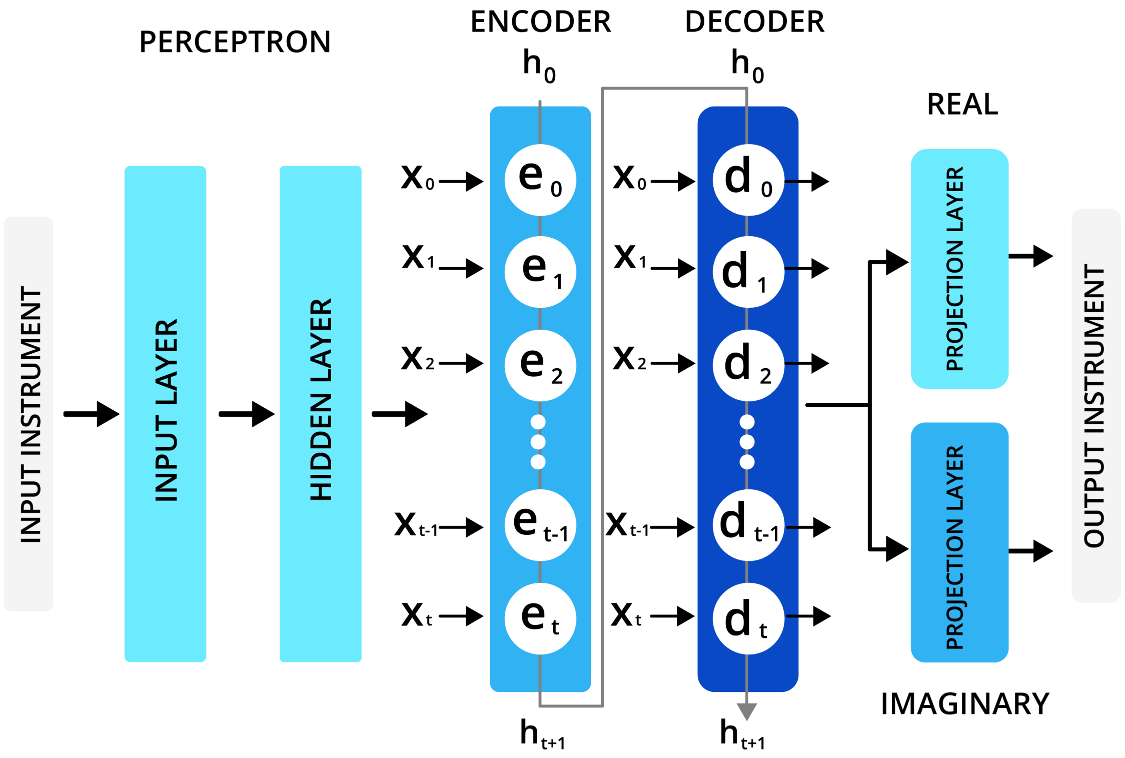

Figure 3.5: Recurrent encoder-decoder model $TLSTM1$ for musical timbral style transfer, utilizing initial dense layers.

into time steps and decoded according to the current state vector. The output is then passed into two projection layers of size 2049, representing the real and imaginary parts of the output. Mean squared error was used as the reconstruction loss function - the model processes real and imaginary parts separately, therefore both components are summed and included in the final optimization criterion. The used optimizer was ADAM with a learning rate of $1e-5$ and a β_1 of 0.9.

Worth noting is that networks with LSTM cells have previously been shown to work well with moderate datasets in audio context [Ezen-Can, 2020] [Ding et al., 2016], additionally supporting the proposed approach to data processing presented in this work. The described recurrent model allowed for significant improvements over the baseline model, as shown in Section 3.4.

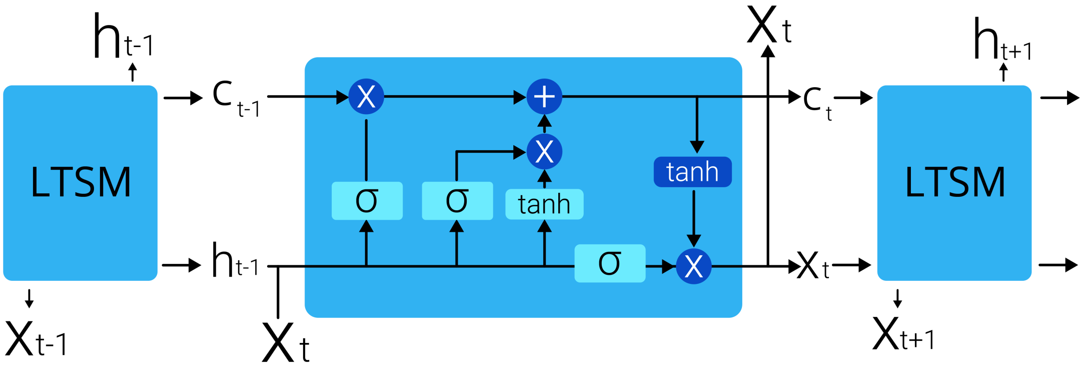

Figure 3.6: Conceptual view of a LSTM cell with additional cell state and forget, update and output gates.

Recurrent autoencoders TLSTM2, TLSTM3 and TLSTM4

A modified version of the previous model has also been prepared. The layers of this model are similar, but arranged in a different way. The model, codenamed $TLSTM2$, is conceptualized and presented in Figure 3.7. The input is passed directly into the LSTM encoder.

The recurrent encoder takes one time step of the STFT signal x_t as an input and combines it with the context vector h_{t-1}^{enc} to produce a compressed representation of the LSTM content vector c_t. The information this vector retains is not limited to the currently processed section of the song, but also contains a cumulative summary of previous parts. This allows for continuous processing of an arbitrarily long piece of music without artificial slicing. The last state vector h_T^{enc} produced by the encoder summarizes the whole sequence. It then populates the initial context vector h_0^{dec} of the decoder along with the data of the input instrument audio signal.

The outputs of the decoder are passed to two connected dense layers. As presented in Section 3.4, this model has improved the standard deviation and error of the results. Two further experiments were also conducted with the same architecture of the model, but with LSTM layers consisting of 1024 and 2048 cells, called $TLSTM3$ and $TLSTM4$, respectively. The used optimizer was ADAM with a learning rate of $1e-5$.

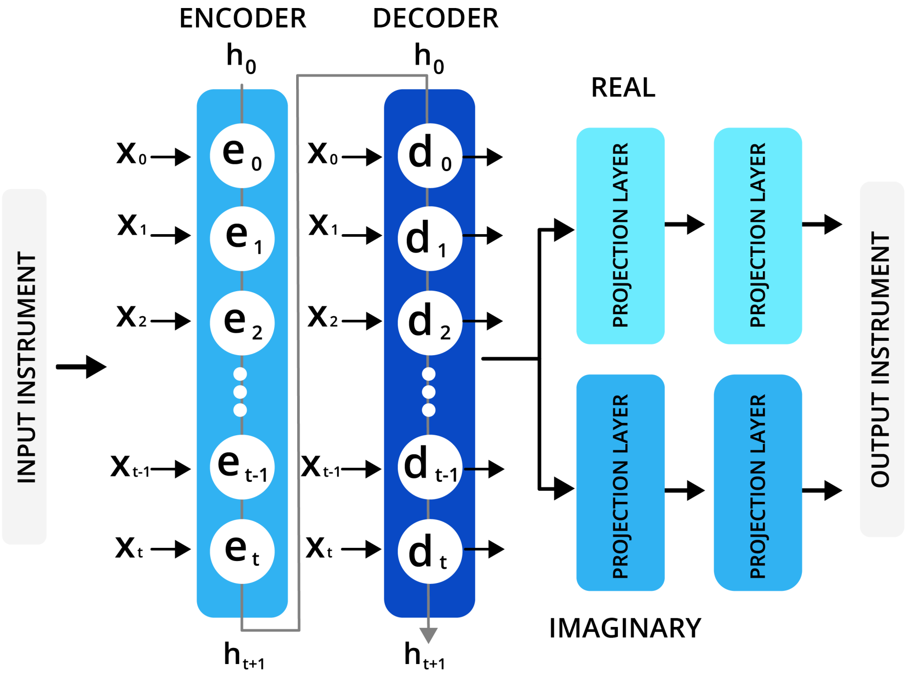

Figure 3.7: LSTM encoder-decoder model for timbral musical style transfer with inputs passed directly into recurrent layers. Architecture used for models TLSTM2, TLSTM3 and TLSTM4.

Recurrent autoencoder TLSTM5

The final experiment was conducted with a similar architecture and LSTM layers with 2048 cells, but a different connections graph: instead of using the input audio signal data along with the final cell state, the decoder's recurrent cells inputs are connected with the encoder's corresponding outputs. The decoder uses encoded portions of the current signal and its hidden state, to step-by step produce a consistent next sequence prediction, containing the features of the song in the domain of the target instrument. The used optimizer was, again, ADAM with a learning rate of $1e-5$ and a β_1 of 0.9.

The modification of the connections graph instead of the size of the layers was caused by the fact that larger models have become not feasible to train within the possessed computational resources (the model exceeded the memory of a single Nvidia Tesla GPU). The performed modifications have allowed to further improve the model's results, both in term of error, standard deviation and audible quality of the obtained results. The model is conceptualized and presented in Figure 3.8.

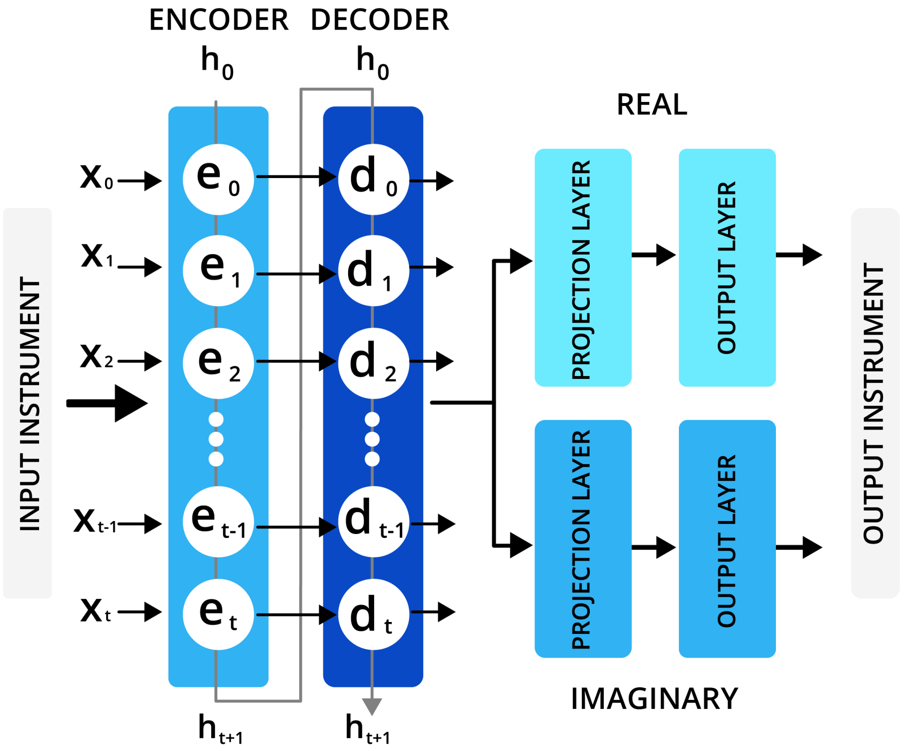

Figure 3.8: LSTM encoder-decoder model $TLSTM5$ for timbral musical style transfer with inputs passed directly into recurrent layers and connections between recurrent cells.

3.4 Results

The following section describes the results of performing timbral musical style transfer between piano and guitar musical pieces. The error and standard deviation of the produced samples were computed and compared for all of the proposed models, along with further musical analysis and explanation. The models taken into consideration in the performed experiments can be summarized as follows:

- $T0$ - baseline, non-recurrent autoencoder,
- $TLSTM1$ - recurrent model with initial dense layers,
- $TLSTM2$ - recurrent model with input passed directly into LSTM layers with 512 cells,
- $TLSTM3$ - like TLSTM2, but with LSTM layers with 1024 cells,
- $TLSTM4$ - like TLSTM2, but with LSTM layers with 2048 cells,

- $TLSTM5$ - TLSTM4 with direct connections between LSTM cells of encoder and decoder.

Metrics

The difference between the input X and the expected output Y was used as a relative benchmark of models' performance. In the case of the performed experiments, X is synthesized piano music and Y is synthesized guitar music.

To establish a reference point for the performed comparisons, the loss value of the test dataset was calculated, ie. how much the original inputs differ from the outputs in terms of their STFT representation. Results below the raw data score should be treated as contributions of the particular model.

Table 3.2: Mean squared error and standard deviation of different models for piano to guitar timbral style transfer task on the test dataset in the described approach.

Model	Real	Imaginary	Average	Avg. std. deviation
Raw data (no model)	0.7328	0.7308	0.7318	1.6281
Baseline (T0)	0.3484	0.3481	0.3483	0.8192
TLSTM1 - initial dense layers	0.2135	0.2130	0.2133	0.6041
TLSTM2 - initial 512 LSTM	0.1903	0.1899	0.1901	0.4804
TLSTM3 - initial 1024 LSTM	0.1577	0.1579	0.1578	0.3779
TLSTM4 - initial 2048 LSTM	0.1421	0.1423	0.1422	0.3252
TLSTM5 - final model	**0.1139**	**0.1137**	**0.1138**	**0.2582**

As presented in Table 3.2, subsequent models have been able to steadily decrease the error and standard deviation values of the outputs. The baseline model with Hann windowing $T0$ has already been able to significantly lower the values and thus perform a naive version of timbral style transfer, however within further experiments it was not able to improve upon the presented values despite using longer training times or different learning rates.

The main disadvantage of the $T0$ model was its limited context. As a data type, music is sequential in its nature and should be processed with respect to its temporal dependencies. The baseline $T0$ model, however, forces long tracks to be processed in independent parts, which results in poor audible qualities of the produced samples. In certain parts, context continuity is not preserved, which results in audible artifacts between boundaries of consecutive blocks.

The usage of recurrent layers in $TLSTM1$ allowed for significant improvement both in terms of audible quality and the collected metrics. Variations in the order of the layers introduced by $TLSTM2$ have further improved the results, although mostly in terms of the standard deviation, thus suggesting that the model produces less outlier values and is more consistent in the quality of the performed style transfer. This has lead to experiments with larger LSTM layers, represented by the models $TSLTM3$ and $TLSTM4$. The final model, $TLSTM5$ obtains the loss of 0.1138 on average and produces the best quality samples of the proposed models.

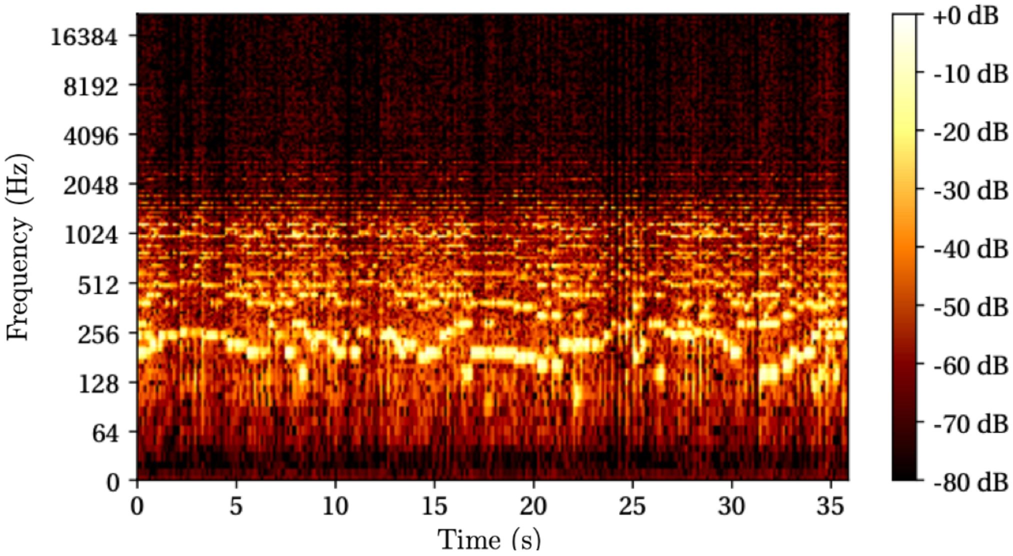

Figure 3.9: Example output of the final model, $TLSTM5$, visualized as a spectrogram.

In the case of recurrent models, the compositional details of the music (like the rhythm, melody, particular notes) are not shifted or distorted, which means that the network was able to successfully preserve the musical content. The output samples produced by the models, in particular the final model $TLSTM5$, have audible characteristics of the desired output instrument, which in the performed experiments was a guitar. Being a plucked string instrument, the guitar's timbre is, in general, brighter and more "metallic" than a piano. It also has a different, well pronounced attack envelope and middle range of frequencies. The audible quality of the achieved timbral style transfer is high, with minimal sonic artifacts introduced in the produced samples, mostly in the form of lightly audible artificial noise, corresponding to the small error values and present mostly in the higher frequencies. A sample output of $TLSTM5$ is presented in Figure 3.9. A selection of samples to listen to is also available at https://bit.ly/3lVWbwQ.

The final model was fast to converge, as it took around 4-5 hours of training time on a single Nvidia Tesla card. This is much faster than the solutions referenced in Section 3.2.1, most of which utilize WaveNet-based approaches, trained for days on numerous high-end GPUs.

3.5 Summary

The method proposed in this chapter is an original method for performing timbral musical style transfer between tracks played on different instruments. Although piano to guitar style transfer has been chosen for the conducted experiments, the network is instrument-agnostic when encoding and decoding the representation. The method can be therefore trained for style transfer between any pair of instruments. The samples produced using the presented method have audible sonic qualities of the desired output instrument. The method also preserves musical context in terms of melody, rhythm and structure of the original samples, as well as preserving effects such as the vibrato or ones achievable by using outboard effect units.

The proposed method utilizes an encoder-decoder network with recurrent LSTM cells. For the purposes of training, MIDI datasets are synthesized into audible files and their STFT is computed for a time-frequency representation retaining information about the phase of the audio signal. The proposed models are fast to converge and offer a much lighter approach when compared to solutions described in Section 3.2.1 (for instance WaveNet-based approaches), trained on numerous GPUs for several days.

Due to the lightweight character, good quality of produced samples and the preservation of context and musical effects, the method is highly applicable in actual music production context and may be used, among other ideas, for:

- building new creative tools for musicians (via, for instance deployment in virtual studio technology instrument),
- transferring signal modulations from one synthesizer to another and comparison of capabilities of such modulations,
- creating new, interesting sounds via, for instance, transferring a brass instrument's vibrato to a keyboard instrument.

Chapter 4

Music generation

Generating new content using deep learning solutions is a very hot issue in artificial intelligence, as seen in numerous examples from the graphics and text domains [Brown et al., 2020] [Goodfellow et al., 2014] [Dosovitskiy and Brox, 2016] . Many efforts have also been put into the automatization of generating music and enhancing the human composer's intelligence with AI. The authors of the comprehensive survey on deep music generation [Ji et al., 2020] decompose the problem into an array of subproblems, like particular tasks of generating chord sequences, melodies, bass lines, full music of narrowed down to one particular genre, polyphonic versus monophonic music and voice synthesis, among others. The authors also highlight the multi-level and multi-modal character of music and its various representations.

A very important issue when considering musical content generated or enhanced by artificial intelligence are the copyrights of the music. The current legal status of such music and its ambiguities are discussed and elaborated on in Appendix C.

The following chapter also presents experimental work conducted in original music generation, along with a proposition of a new solution and experimental results.

The chapter also describes the background of music generation using artificial intelligence solutions, both in terms of neural networks and more traditional machine learning approaches. The overall lack of benchmark datasets (with many undisclosed ones), high variety of proposed methods and sometimes undisclosed audio results makes it somewhat difficult to clearly evaluate and categorize the proposed approaches, therefore only a certain selection has been described. Some of the works also provide selections of generated audio to listen to. The generated audio, although often immature when compared to music composed by humans, often has a distinct character and very particular musical qualities. This allows to attribute the solutions as representants of a newly emerging class of musical instruments - for instance, to broadly talk about "how does this neural network sound when compared to a different one". This is a stream of thought that the author is in particular a supporter of, one that the author considers important in the spirit of computational creativity and content creation and one that allows to consider the algorithms in terms of computational artistry.

4.1 Related works for music generation

Classic machine learning approaches

The work of Pinkerton [Pinkerton, 1956] may be the first attempt of using computational technology and information theory to generate new music. Computing the entropy per note for various melodies, the author was able to conclude that a certain amount of redundancy is needed in order to produce tuneful melodies. The following work [Brooks et al., 1957] proposed a Markov transformation model to generate simple melodies with the usage of a small musical corpus.

Since then, many approaches have been proposed for various subproblems of music generation. In further efforts, Markov chains [Hiller and Isaacson, 1979], probabilistic generative grammars [García Salas et al., 2011], and various combinations of thereof [Cope, 2000] have been used for semi-automatic generation of musical excerpts.

[Lavrenko and Pickens, 2003] propose statistical modelling of polyphonic music using random fields and show their approach outperforms Markov chains on four different musical collections. Genetic algorithms with a human mentor have been used in [Biles et al., 1994] in order to emulate a novice jazz musician learning to play improvised solos over a given harmony. Genetic algorithms have also been used along with a bracketed L-system by [Fox, 2006] for automatic composition and interpolation between musical structures.

Certain effort has also been put in emulating human performance and interpretation. The authors of [Grindlay and Helmbold, 2006] propose a hidden Markov model for the modelling of expressive piano performance in order to emulate variations employed by actual piano players, as opposed to literal synthesis of notes from a given score. [Gu and Raphael, 2012] propose the usage of switched Kalman filters for the same issue of modelling expressive, dynamic performance.

While considerable success in various subproblems of music generation has been achieved by some of the mentioned approaches, [Ji et al., 2020] state many of these models only reproduce subsequences existing in the original data, use abstract and hard to follow representations, suffer from lack of memory or are limited by the need of human supervision.

Deep learning

As with the domain of images and text, a paradigm shift in applications of artificial intelligence to music has come with the era of deep learning, although the idea of using neural networks for music generation can be traced back to 1989, where sequential, recurrent neural networks were proposed for generating melodies similar to provided training examples [Todd, 1989].

[Eck and Schmidhuber, 2002] propose the usage of LSTM recurrent neural networks for generating short, cohesive excerpts of blues music. [Boulanger-Lewandowski et al., 2012] propose a recurrent neural network and restricted Boltzmann machine model for learning harmonic and rhythmic probabilistic rules from polyphonic music scores. The authors state that their model outperforms many previous approaches, but they still find long term structure and musical meter as "elusive".

[Dong et al., 2018] propose the usage of generative adversarial networks for generating multi-track polyphonic music and improve the chaotic character of their outputs in their subsequent work [Dong and Yang, 2018]. Other approaches also include applications of the previously mentioned WaveNet [van den Oord et al., 2016] models by DeepMind, trained on massive sets of musical raw audio signals.

Google Magenta

Magenta is a subdivision of Google AI focused on creative applications of artificial intelligence, especially in the context of deep learning for musical purposes. Magenta has put out several interesting contributions in the domain of music generation, such as:

- MusicVAE [Roberts et al., 2019], a hierarchical variational autoencoder able of interpolating between musical samples,
- the music transformer [Huang et al., 2018] with relative self-attention, capable of generating long, polyphonic musical phrases,
- the performance RNN [Simon and Oore, 2017] based on LSTM layers, capable of generating short phrases with locally coherent, expressive patterns,
- Wave2Midi2Wave, a cooperative method combining various models trained on their MAESTRO dataset [Hawthorne et al., 2018], usable for transcribing and creating new musical excerpts.

OpenAI: MuseNet and Jukebox

The artificial intelligence powerhouse OpenAI known, among others, for the GPT-3 transformer has also contributed research to music generation. Their most interesting contributions to the topic of music generation are MuseNet [Payne and OpenAI, 2019] and Jukebox [Dhariwal et al., 2020]. MuseNet is based on GPT-2, a powerful unsupervised large-scale (1.5 billion parameters) transformer model used for predicting the next token in a given sequence. MuseNet uses a 72 layer sparse transformer network [Child et al., 2019] with 24 attention heads and is able to produce samples with a selection of instruments when given a musical prompt consisting of only a few notes.

Jukebox takes the genre, artist and lyrics as the input and generates audio output. The network is trained on a huge dataset of over a million songs in raw audio format. Multi-level vector-quantised variational autoencoders [van den Oord et al., 2017] are used to compress audio to a latent space. 72-layer sparse transformer models are then used to generate novel samples in the compressed spaces, with additional artist, genre, and lyrics conditioning.

Both of these solutions are able to produce very interesting and high quality results, but require massive computational resources and huge amounts of data for training.

DeepJazz

DeepJazz [Kim, 2016] uses a two layer LSTM network trained on MIDI files and has been built in a few days as a submission to a hackathon. The author has trained the network to reproduce music in the style of Pat Metheny. The solution, although clearly overfit and built for a high "wow" effect, is able to produce very good MIDI samples sounding like actual Pat Metheny compositions. DeepJazz has been featured in The Guardian, Aeon Magazine and the front page of HackerNews. The project is no longer maintained or developed.

Amper Music, Endel and other commercial solutions

Currently some companies and emerging startups are offering music generated (to an unknown extent) by artificial intelligence as a commercial service. Examples of such businesses are [Amper, 2014], which is able to produce musical excerpts of varying length, genre and instrumentation and [Endel, 2021], providing ambient soundscapes for focus and relaxation. These services use undisclosed algorithms, post-processing and training data. The quality of the output music provided by such companies is mostly very high.

4.2 Proposed solution

The following section describes the proposition of a new solution for music generation and expands on the experimental results presented in [Modrzejewski et al., 2019]. A deep convolutional generative adversarial network is used to create new, previously unheard music. Such networks have been shown to perform well with images, therefore a graphical representation of music has been used for the purpose of training, utilizing a piano roll format with color-encoding in order to compress more musical information into a single training sample. The piano roll has the advantage of encapsulating rhythm, melody and harmony information in an explicit way. It is also a natural way of visualization for digital music, closely related to actual sheet music. Due to relatively good availability and the expressiveness of MIDI [MIDIAssociation, 1999] datasets, MIDI data has been used along with a pre-processing pipeline for creating the color-encoded piano rolls. During research presented in this thesis, the method has been proven successful at generating original musical phrases upon musical ideas learned from four different training datasets. Along with the method, a musical analysis of the obtained results and an actual album of music created with the proposed method are presented.

4.2.1 Data

The motivation for using MIDI data for the purposes of generating new musical samples using neural networks is similar to the one presented in Section 3.3.1. Music datasets for the purposes of training neural networks often lack in terms of their quality and quantity, as previously stated by [Engel et al., 2017] and [Sturm, 2012b], among others. However, at the time of the presented research, some sufficient MIDI datasets have already existed. The fact that MIDI contains information about the performance control, rhythm, melody and harmony makes it a natural choice for the issue of generating new musical content.

The following MIDI datasets have been used for the experiments:

- LMD-matched [Raffel, 2016]
- 130,000 MIDI file collection [midi_man, 2019]
- MAESTRO dataset [Hawthorne et al., 2018]
- Doug McKenzie Jazz Piano MIDI dataset [McKenzie, 2012]

Exploratory data analysis

The music contained in the datasets was analyzed for its character and presence of particular musical features in the context of composition and arrangement. The selected datasets contain a broad range of means of musical expression:

- MAESTRO [Hawthorne et al., 2018]
  - dominant genre: classical,
  - used length of music: around 170 hours of music,
  - technical difficulty: very high, virtuoso-level classical music piano performances collected from a piano competition in Minneapolis, USA,
  - harmony: ordered, in most cases obedient to rules of classical music characteristic for corresponding musical epochs,
  - rhythm: non-repetitive with frequent tempo and phrasing changes.
- Doug McKenzie Jazz Piano [McKenzie, 2012]
  - dominant genre: jazz,
  - used length of music: around 20 hours of music,
  - technical difficulty: very high, professional-grade jazz piano performances,
  - harmony: very rich, with complex jazz harmonies utilizing a variety of chords and scales, also including improvised phrases and known jazz chord progressions,
  - rhythm: varying, often in a swing triplet phrasing, typical for jazz music.
- LMD-matched [Raffel, 2016]
  - dominant genre: pop, rock, electronic,
  - used length of music: subset of around 60 hours of music,
  - technical difficulty: varying from low to high,
  - harmony: mostly ordered, with typical song structures containing popular chord progressions, in some cases simple and easy-listening,
  - rhythm: mostly ordered, with many rigid and symmetrical structures typical for pop, rock and electronic music

- 130,000 MIDI file collection [midi_man, 2019]
  - dominant genre: pop, classical, electronic, film and game scores,
  - used length of music: subset of around 70 hours of music,
  - technical difficulty: varying from low to high,
  - harmony: varying, mostly ordered, in some cases simple and easy-listening,
  - rhythm: varying depending on particular genre,

Data pre-processing

The MIDI standard, as stated in Section 2.2.1, defines a binary communications protocol. For the purposes of the presented approach, binary MIDI files are first converted into a text form in order to extract the raw information about the particular notes, represented as subsequent lines of text. Having obtained the text representation, redundant information like MIDI comments, *program change* messages, time signature and tempo information etc. is dropped. The velocity of each note is set to the maximum as a method of initial reduction of dimensionality, although additional experiments with full velocity range have also been performed and are described in further sections. Furthermore, the samples are quantized to the value of 30 milliseconds, which is the length of a 16th note in 120BPM tempo.

The considered data represents piano music with a full scale, 88 key keyboard. This range is cropped to 64 by transposing the extreme low and high notes respectively up and down by an octave. Although this introduces some bias due to the loss of information, in the considered datasets the extreme ranges of the piano contribute a small amount of the content. The overall character of the samples has been therefore left unmodified.

Error handling in the datasets was as follows:

- Instances of unclosed sustain pedal events along the datasets (MAESTRO and LMD-matched in particular) were handled by introducing a modification in the MIDI files. Sustain pedal events were shut after 3 seconds, which is an arbitrarily selected value corresponding to actual musical context - in most pieces of music, a sustain pedal for a certain set of notes would not be held for much more.

- Instances of corrupt MIDI data along the datasets, whenever encountered, were dropped entirely.

- Instances of *note-off* messages occurring before a corresponding *note-on* message were ignored.

The MIDI is then further compressed into 64x64 pixel images. In order to fit more musical information into a single image, the rhythmic structure is coded using RGB channels of each pixel to represent additional sixteenth notes, resulting in red, green, blue, yellow, cyan and magenta pixels. Notes longer than six sixteenth notes (the equivalent of three eight notes) are marked as white. If no note occurs, the pixel stays black. This transformation allows to encode 20 seconds of music in a single 64x64 image. An example of such a training image is presented in Figure 4.1 - the x axis represents time, while the y axis represents the 64 considered pitches.

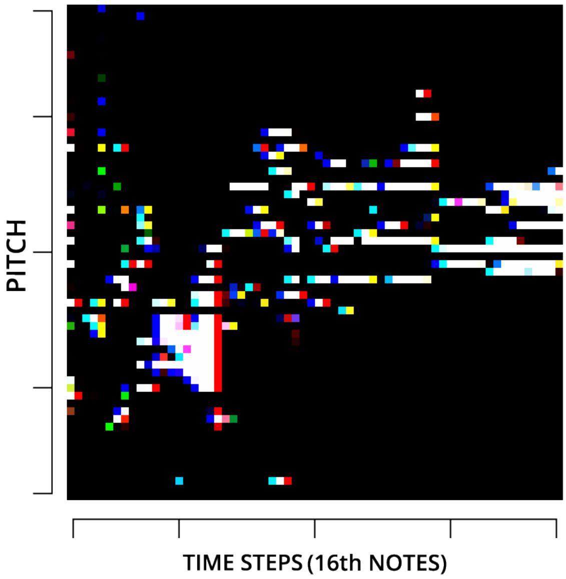

Figure 4.1: Example of an input image for the network with time steps on the x axis and pitch on the y axis.

4.2.2 Method

The model used for generating musical content is a deep convolutional generative adversarial network. GANs have been proposed in [Goodfellow et al., 2014] and have since become incredibly popular for image generation tasks. A GAN consists of two networks, a generator G and a discriminator D. The generator's task is to produce "fake" images from a given domain, while the discriminator's task is to decide whether an image is a "real" example from the training data or a "fake" one from the generator. This idea conceptualized and presented in Figure 4.2.

Given some data x and a latent space vector z, $G(z)$ is the generator's function mapping z to the space of x. The generator tries to estimate the original distribution $p_{data(x)}$ of the data in order to produce "fake" samples from it. $D(G(z))$ will therefore be the probability that the

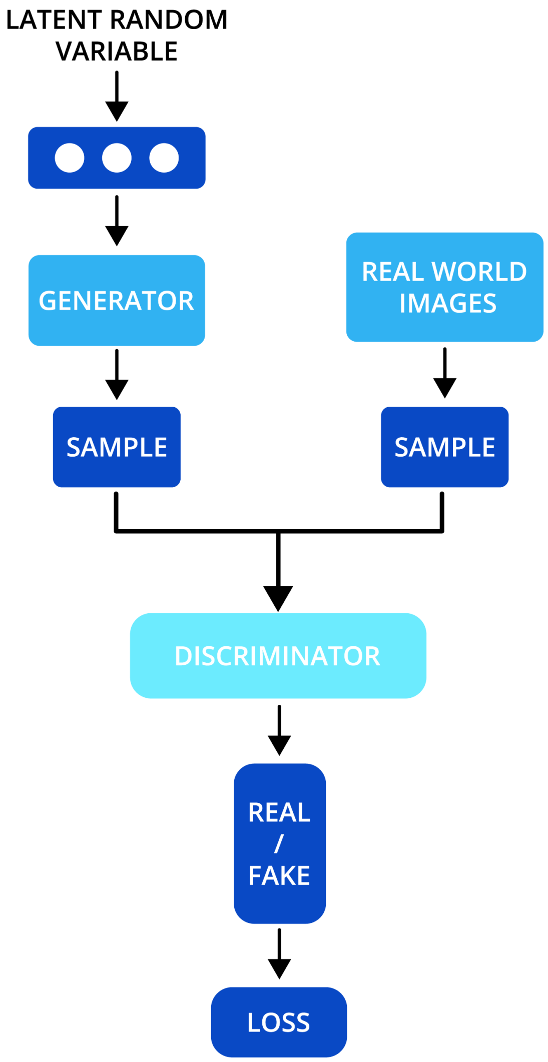

Figure 4.2: Conceptual schema of a generative adversarial network.

output is a real image. [Goodfellow et al., 2014] define the general loss function of the GAN as a minimax game between the generator and discriminator as in Equation 4.1:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data(x)}}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1-D(G(z)))] \tag{4.1}$$

The idea of generative adversarial networks has since been extended in numerous ways, as the networks have proved to pose quite some difficulties in training in practice. In particular, a *model collapse* may occur when the generator starts outputting the same sample every time when it finds a sample that is able to perfectly fool the discriminator every time. [Radford et al., 2015] have proposed a number of guidelines for training of a GAN with convolutional layers, which improve both the stability of training and quality of the outputs. These guidelines include the usage of ReLU for the generator and LeakyReLU for the discriminator, strided convolutions in the discriminator and fractional-strided convolutions in the discriminator. The generator's final layer uses tanh function. Batchnorm [Ioffe and Szegedy, 2015] is used for for both networks. Schematic views of the generator and discriminator are presented in Figure 4.3.

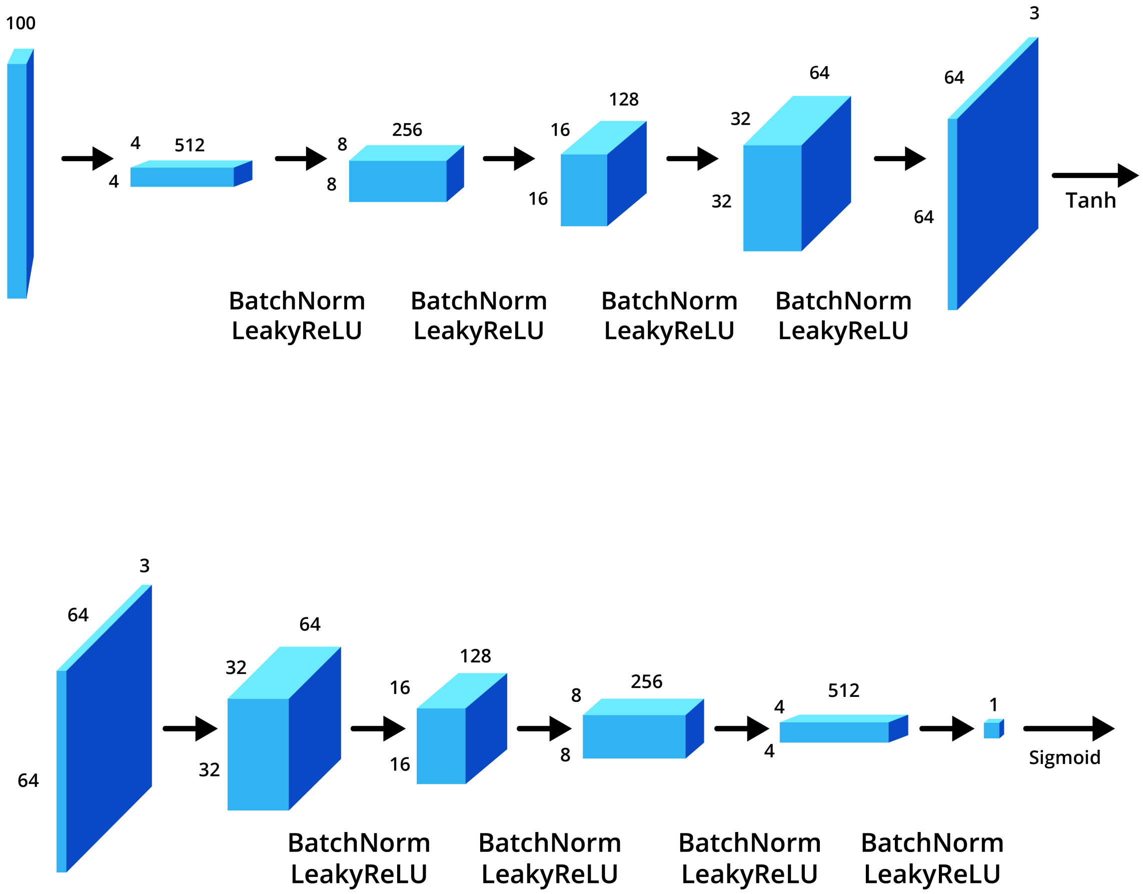

Figure 4.3: Schematic structure of the generator (top) and discriminator (bottom).

4.2.3 Experiments

The network was trained for each of the datasets described in Section 4.2.1 for around 12 hours on a single Nvidia Tesla GPU. The network was trained for around 50,000 iterations on MAESTRO, LMD-matched and 130,000 MIDI file collection datasets and around 120,000 iterations on the smaller Doug McKenzie dataset. Upon experiments with the datasets and the preprocessing pipeline described in Section 4.2.1, a few additional experiments were performed with preservation of the original dynamic range with varying velocity of the music, as opposed to the flat dynamics set at the maximum value as in the main part of the experiments. This has been done in order to verify whether the dynamic spectrum influences the training process.

The networks were trained using the ADAM [Kingma and Ba, 2014] optimizer with a learning rate of $2e-4$ and momentum term β_1 of 0.5.

Training the network on the whole collected dataset would have been a very interesting experiment, unfortunately at the time of research it was out of reach for computational reasons. Also, training on smaller, better-defined datasets allow for a more in-depth musical analysis of the obtained results and throughout verification of the networks performance in terms of recreating musical features of the training datasets.

4.3 Results

Each of the networks has learned to recreate the provided input images. The images were then converted back to MIDI form and listened to for analysis purposes. Increasing the training time over 50,000 iterations for the three large datasets did not bring any audible improvements to the produced samples. At that point the loss of the discriminator has oscillated around 0, meaning that it is capable of distinguishing the real images from the fake ones. When the training was stopped, the cost of the generator was oscillating around the value of 5. In case of the Doug McKenzie dataset, the model has collapsed when over 120,000 iterations, therefore results were collected from one of the final iterations before the model collapse.

The experiments with a full spectrum of dynamic have not improved the results of any of the experiments, as the loss of the generator fell to 0, meaning it was able to fool the discriminator every time. This in turn has resulted in a lower quality of the produced samples both in terms of harmony and, in particular, chaotic rhythmic structure. These samples were discarded and not included in the subsequent analysis.

The results of the experiments were evaluated according to the network's performance and the quality and usability of the produced samples. The samples produced by the network are evaluated in terms of their visual similarity to the input images and the musical qualities extracted and recreated from the test datasets.

In order to verify the usefulness of the method for actual content creation support, a mini-album of original music was created for the purposes of this work. The album is described and analyzed in section 4.3.3. To the best of the knowledge of the author, this is one of the first cases of where a method for musical generation using deep learning is backed by an actual, self-contained album of music created with cooperation between human and artificial intelligence.

4.3.1 Result analysis

Visual results

The images generated on all of the datasets are indistinguishable for the human eye from the real images, as presented in Figure 4.4. This is in particular a result of the fact that the presented method uses a compressed, latent representation of music. This specific color-encoded piano roll representation holds some visual information readable by humans, as it presents the relative pitch of the notes and their position in the whole musical sequence, although the RGB compression partially distorts the visible musical structure. Nevertheless, certain rhythmic patterns are still visible, as in the case of MAESTRO and Doug McKenzie datasets the produced samples have a much more chaotic structure than in the case of LMD-matched and 130,000 Midi File Collection. This is an indication that the network has learned to reproduce the rhythmic patterns of the input music, as virtuoso classical piano and jazz music indeed will tend to have more intricate rhythmic structure than most pop, rock and electronic music.

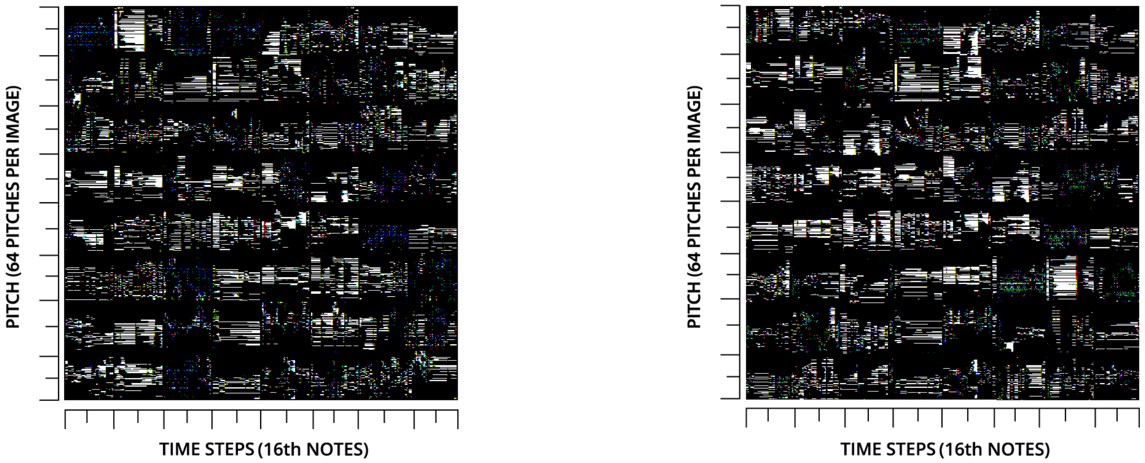

Figure 4.4: 8x8 matrices of real training images (left) vs fake images generated by the network (right). The x axis represents time, while the y axis represents pitches.

Harmonic analysis

In order to verify the cohesiveness of the generated results with the overall harmonic character of the music in the datasets, musical analysis has been performed on a selection of result MIDI files from each training. A batch of 10 files per dataset was chosen at random and listened to in order to count selected musical terms and investigate their cohesiveness with the music in the input datasets, which is a non-trivial task [Benward, 2014] [Laitz, 2008]. The following Tables 4.1, 4.3 and 4.2 present the results of the analysis, with amounts of selected, usable musical terms found in batches of randomly chosen files. As the overall character of LMD-matched and 130,000 MIDI File Collections datasets is somewhat similar, the results of those experiments have been aggregated into a single table.

Table 4.1: Musical terms found in a sample of 10 output files of the network trained on MAESTRO dataset.

#	musical term	occurrences	comments
1	chords (triads)	32	major chords - 18, minor chords - 14
2	chords (suspended)	10	-
3	chords (complex)	4	altered
4	self-contained chord progressions	12	-
5	cadences	10	-
6	self-contained melody lines	16	-
7	self-contained bass lines	20	-
8	self-contained phrases	18	-
9	phrases based on scales	20	harmonic, diminished, mixolydian, dorian, ionnian, eolian

The music generated by the network has similar harmony to the training examples. On the MAESTRO dataset, containing virtuoso classical piano music, mostly minor and major chords were found. The harmony generated by the deep convolutional generative adversarial network also contained typical classical music voicings and chord inversions. A moderate amount of suspended chords with fourths were also found. The samples contained pronounced bass lines and arpeggios, as well as resolutions from a dominant chord to a tonic chord, which is one of the most important and basic resolution in music.

The harmonic content generated upon training on the Doug McKenzie dataset was very rich, as expected when training on a jazz dataset. On top of the musical terms found in the samples trained on MAESTRO, a batch of output files from this training contained complex voicings and chords, with seventh, ninth and altered chords, among others. The phrases were also built on a larger variety of scales, which is typical for jazz music. The network was also able to recreate typical jazz chord progressions, including the ii-V-I turnaround, a staple in jazz music.

The harmonic content for the two remaining datasets was somewhat similar, as both of the datasets contain pop and rock music, therefore the results were aggregated into a single

Table 4.2: Musical terms found in a sample of 10 output files of the network trained on the Doug McKenzie dataset.

#	musical term	occurrences	comments
1	chords (triads)	29	major chords - 13, minor chords - 16
2	chords (suspended)	15	-
3	chords (complex)	15	altered, seventh, ninth, sixth and others
4	self-contained chord progressions	17	-
5	cadences	14	ii-V-I turnarounds
6	self-contained melody lines	19	-
7	self-contained bass lines	18	-
8	self-contained phrases	23	-
9	phrases based on scales	24	harmonic, dorian, diminished, locrian, phrygian, mixolydian, lydian, ionnian, eolian

Table 4.3: Musical terms found in a sample of 10 output files of the network trained on the LMD-matched and 130,000 MIDI file collection dataset (5 samples each).

#	musical term	occurrences	comments
1	chords (triads)	28	major chords - 16, minor chords - 12
2	chords (suspended)	8	-
3	chords (complex)	4	seventh, altered
4	self-contained chord progressions	11	-
5	cadences	10	-
6	self-contained melody lines	21	-
7	self-contained bass lines	18	-
8	self-contained phrases	17	-
9	phrases based on scales	13	dorian, ionnian, eolian, mixolydian

table. The overall harmony was more structured and simple when compared to the previous experiments. Typical resolutions for popular music were also found in the produced samples, as well as pronounced melody and bass lines. These files were also found to be less chaotic than in the previous experiments, which also corresponds well to the differences between pop, rock, classical and jazz music.

Many of the output samples also contained dense note clusters, which would not be typically composed in actual musical settings. Many of those clusters are also impossible to perform by an actual instrumentalist. However, a significant amount of the clusters was found to be consonant, with chords spanning over several octaves. This type of cluster works very well in an ambient or electronic music production setting, allowing for the usage of slowly evolving pad synthesizers with gradual manipulation of filters and modulators.

Rhythm analysis

The music generated with the MAESTRO and DougMcKenzie datasets mostly had a chaotic, heavily syncopated rhythmic structure. This structure in many instances can be considered quite unpleasant to listen to, if considering the raw output of the network. The chaotic rhythm may be attributed to a number of causes:

- training using virtuoso performances with very fast phrases and arpeggios,
- training using overall very complex, information-rich musical classical and jazz genres,
- color-coded compression of rhythm used in the pre-processing of data, which maximizes the length of the samples with a certain rhythmic clarity trade-off,

However, the produced samples contained multiple usable musical ideas, often spanning an even number of bars and contained within symmetrical phrases. Many of these ideas have a pronounced and audibly deliberate rhythmic structure, constructed with clear phrases consisting of eight and sixteenth single notes intersected by or played above adequate chords. This provides a great feature of loops for a human composer to choose from, as demonstrated in Section 4.3.3.

As expected, the rhythmic structure of the samples produced with the use of the LMD-matched and 130,000 MIDI file collection datasets had a less chaotic rhythmic structure. Symmetric ideas spanning an equal number of bars or quarter notes were also frequent. These observations confirm that the deep convolutional generative adversarial network is able to successfully retrieve and reconstruct the rhythmic qualities of the input dataset.

4.3.2 Summary

The deep convolutional generative adversarial model was successful in generating music with the MIDI data compressed into images using the presented method. The proposed piano roll compression successfully resulted in the creation of long, varying, interesting motifs. The method allows for generating infinite amounts of musical phrases, effectively acting as a sample generator for the human composer to choose from.

The proposed approach allowed for generation of interesting, harmonically rich, usable musical ideas. Many existing solutions for similar problems of generating music create phrases that are much simpler, shorter or overfit to a particular style when compared to the proposed approach [Kim, 2016] [Amper, 2014] [Boulanger-Lewandowski et al., 2012] [Dong et al., 2018]. The advanced chord progressions and harmonically complex structures with resolutions are the main advantage of the proposed approach over other approaches. Methods such as described in Section 4.1 often have a narrower scope of operation or focus closely on certain aspects of the generated music (recreating short melodies, operating within certain scales or keys, matching a melody to given harmony etc.), while the harmonic richness of phrases generated in the proposed approach is designed to work as a mean of enhancing and inspiring the composer's creativity.

4.3.3 DROP mini-album

In order to demonstrate the usage of the proposed method and confirm its usability for actual content creation support, a short digital album of musical miniatures has been created in collaboration with renowned Polish jazz musician Michał Milczarek [Milczarek, 2021]. A batch of 50 MIDI files produced by the network, trained on the aforementioned datasets and containing various musical terms, was chosen at random by the author and given to the musician. Milczarek, as the human creator, has subsequently selected a set of files arbitrarily and contributed intentional expression, sound design, musical production and mixing to the files composed by the neural network. The output MIDI of the network was not modified in any way except for selection of full phrases of varying length. The general creation workflow of the album is presented in Figure 4.5.

The album is called *DROP* and is free to stream and listen at `https://bit.ly/3G4rkXj`. It contains 6 musical pieces co-created by human and artificial intelligence. It is highly advised to listen to the album in order to conceptualize the results of the presented method and its applications.

Figurers 4.6 - 4.11 present the MIDI used for the creation of *DROP*. The images present the raw MIDI generated by the network, retrieved from the generated piano rolls and loaded into Ableton Live digital audio workstation software. Figures 4.12 - 4.17 present the sheet music generated by the author from the selected MIDI excerpts used for the creation of *DROP*. Both of these representations are already available at the "Retrieved MIDI samples" block of Figure 4.5. Musical qualities not contained within these representations, like sound design and production, are human contribution. The sheet music visualizes the musical qualities described in section 4.3.1: the presence of chords and musical phrases based on particular scales, along with some

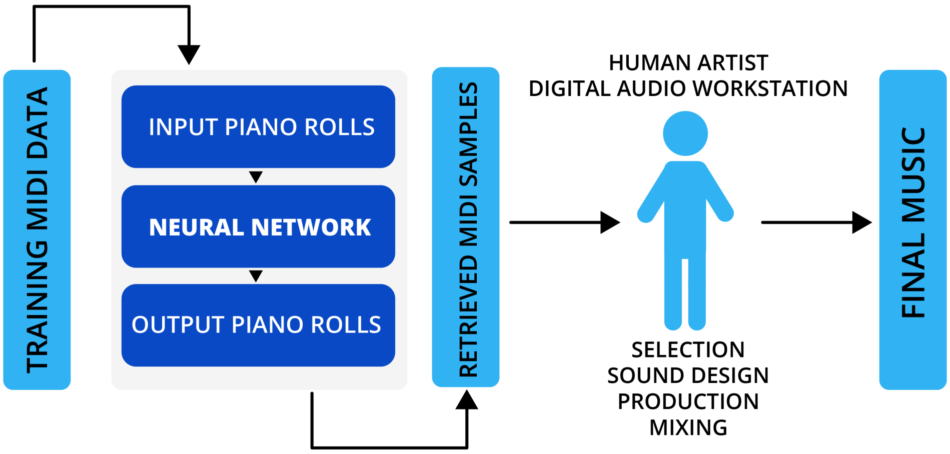

Figure 4.5: AI enhanced music creation workflow employed for *DROP*.

of the chaotic rhythmic character and the presence of dense legato clusters. These qualities were found to work very well as means of enhancing an actual modern musical creation and production workflow, clearly confirming the contribution of the proposed method to content creation support.

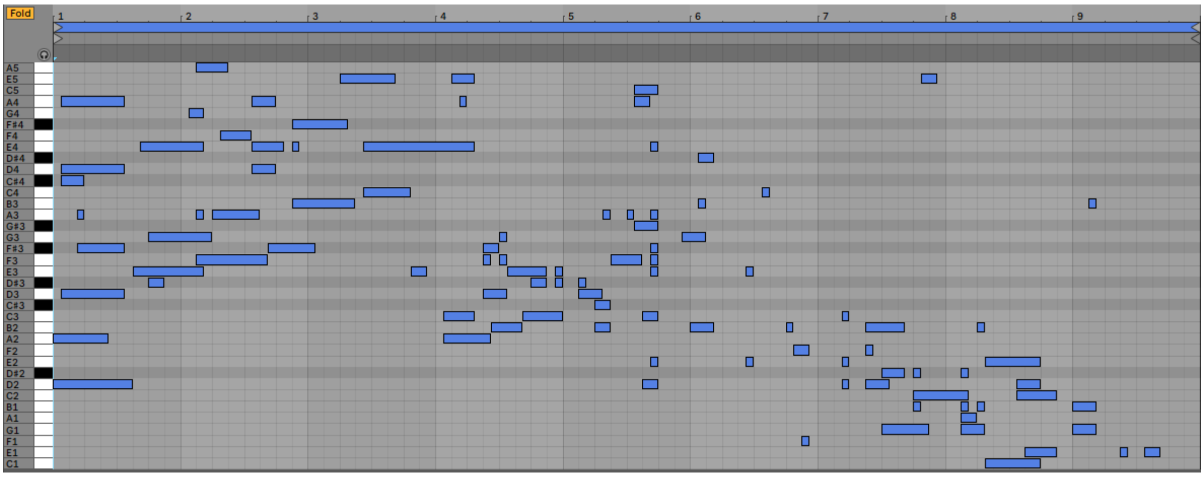

Figure 4.6: MIDI excerpt 1 used for *DROP* album.

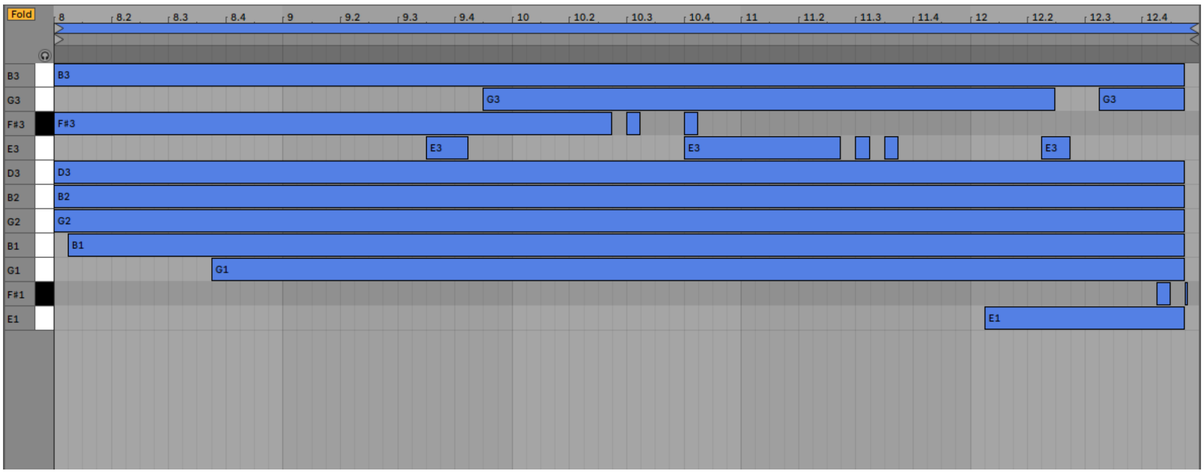

Figure 4.7: MIDI excerpt 2 used for *DROP* album.

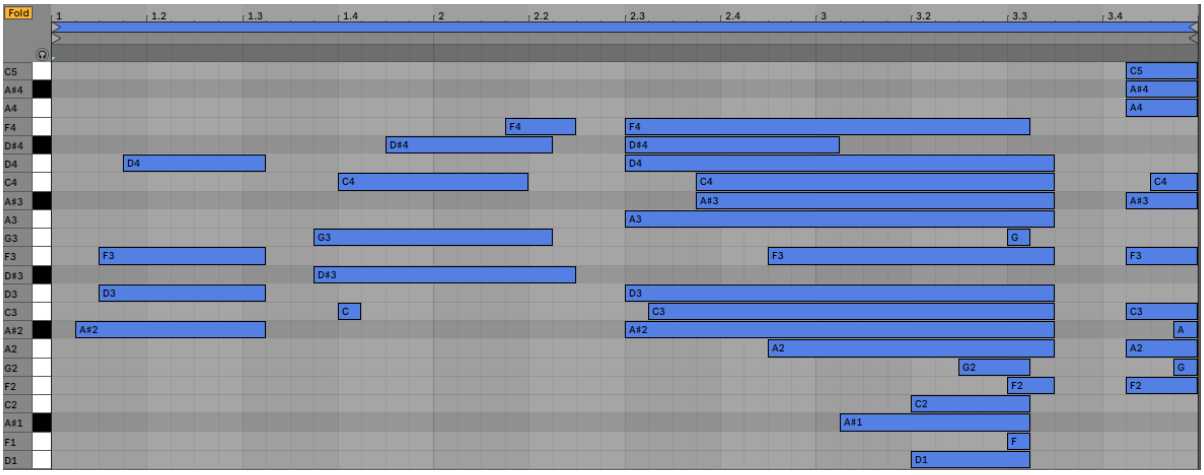

Figure 4.8: MIDI excerpt 3 used for *DROP* album.

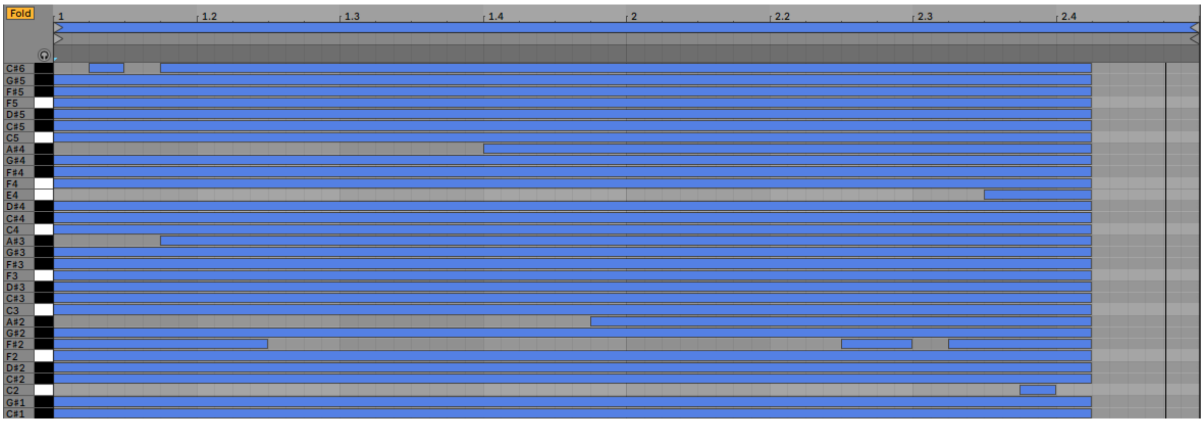

Figure 4.9: MIDI excerpt 4 used for *DROP* album.

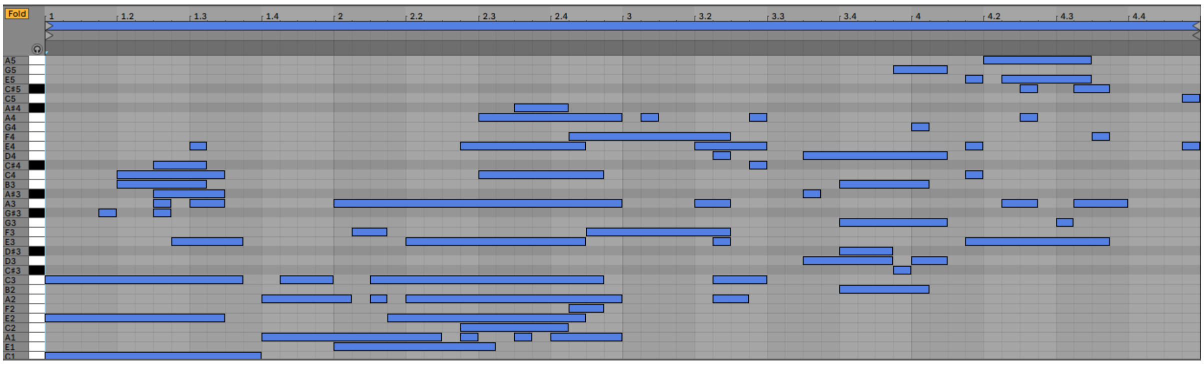

Figure 4.10: MIDI excerpt 5 used for *DROP* album.

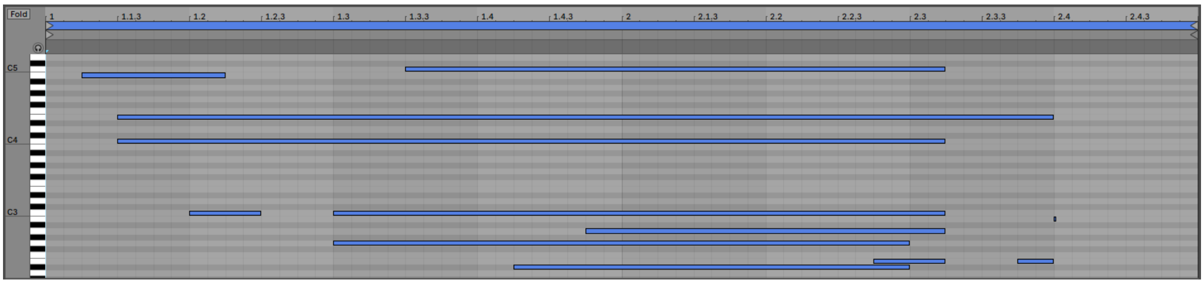

Figure 4.11: MIDI excerpt 6 used for *DROP* album.

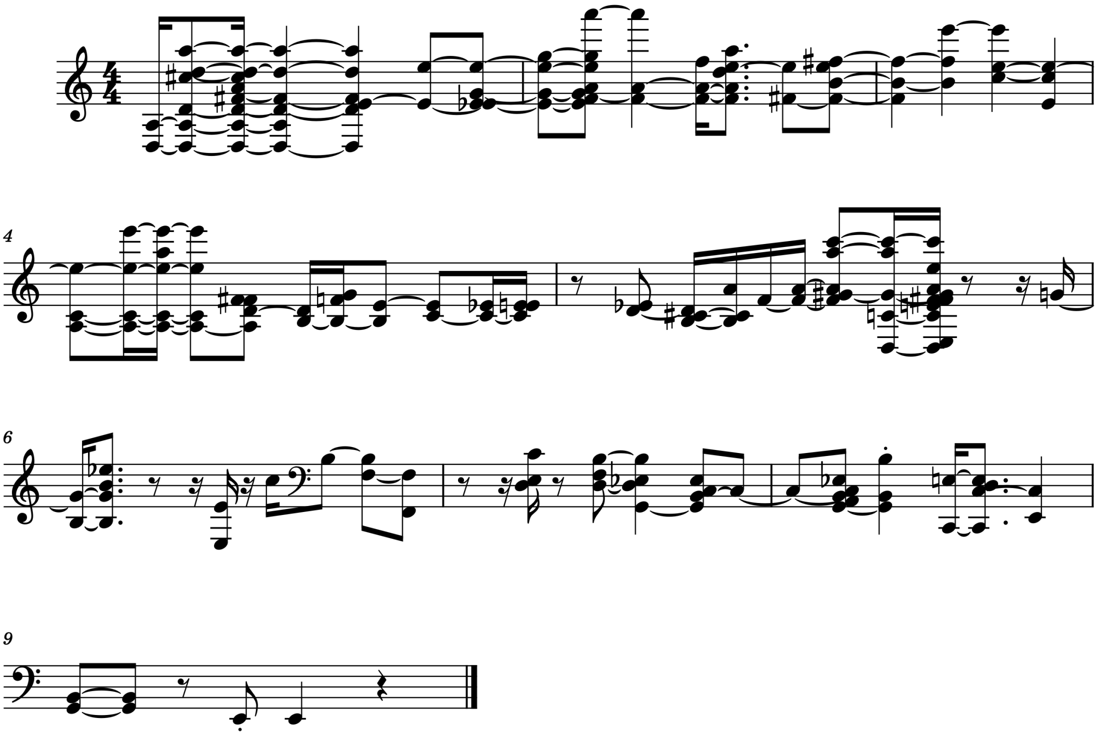

Figure 4.12: Sheet music for excerpt 1 used for *DROP*.

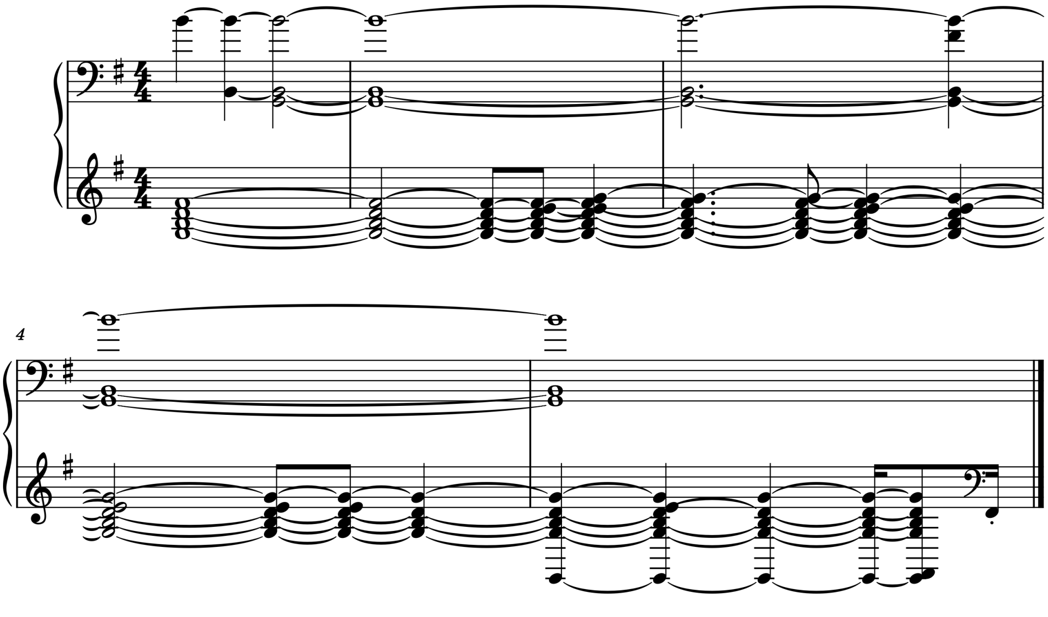

Figure 4.13: Sheet music for excerpt 2 used for *DROP*.

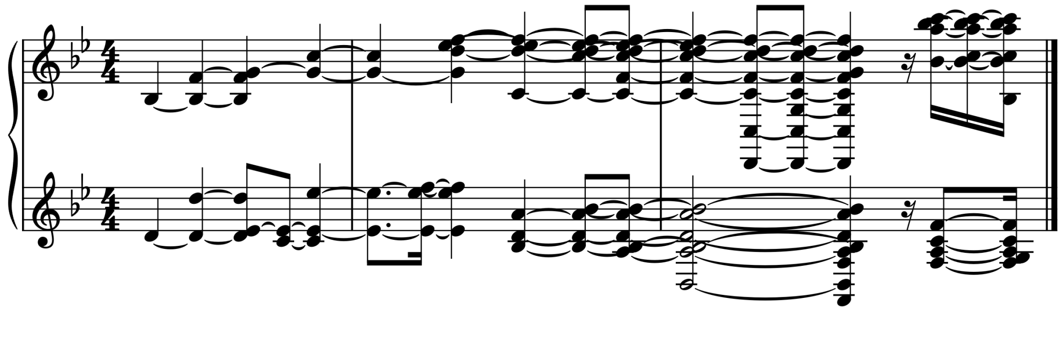

Figure 4.14: Sheet music for excerpt 3 used for *DROP*.

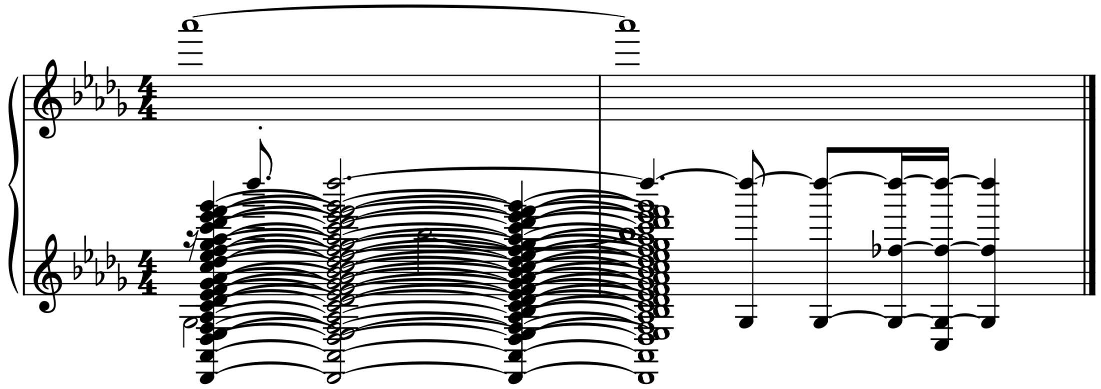

Figure 4.15: Sheet music for excerpt 4 used for *DROP*.

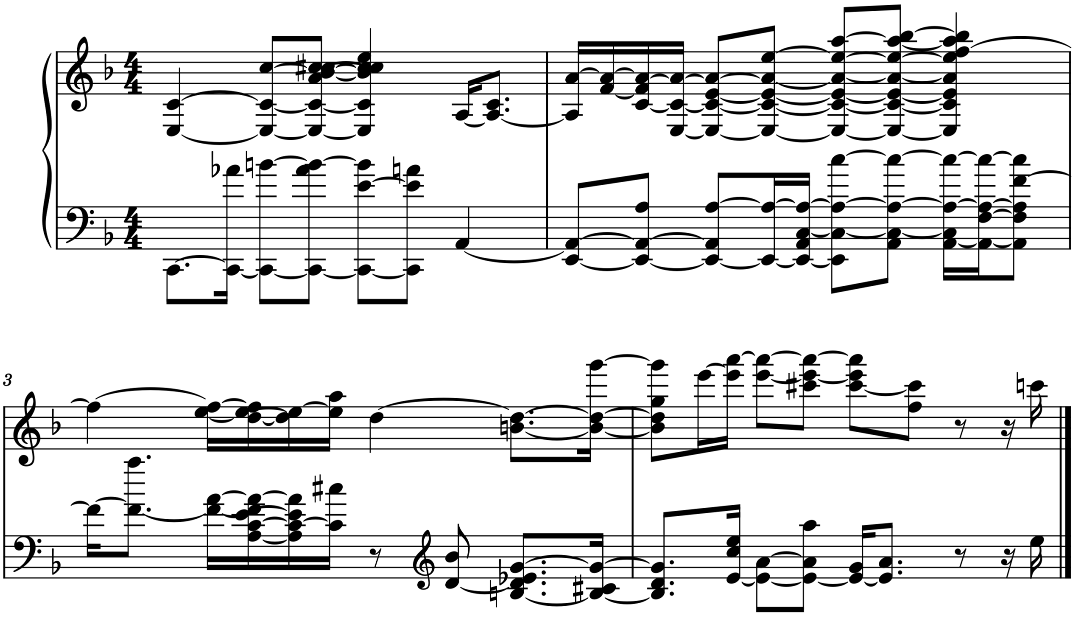

Figure 4.16: Sheet music for excerpt 5 used for *DROP*.

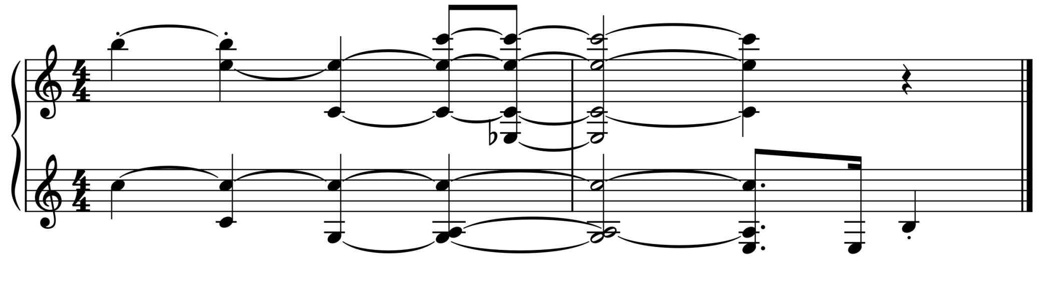

Figure 4.17: Sheet music for excerpt 6 used for *DROP*.

Chapter 5

Music genre classification

Music information retrieval (MIR) is a particular area of research on the intersections of computer science, machine learning, signal processing, musicology, psychology and psychoacoustics, among others. MIR seeks to answer questions about the perceived qualities of music, its representations suitable for various tasks of automation, the correlations between its various features and the insight emerging from those answers.

However, the applications of deep learning techniques to music information retrieval is still quite a new and undeveloped issue. It has been gaining attention in recent years, although the authors of the positional paper [Choi et al., 2017a] state that the majority of works still adopt and asses methods effective in other domain, such as images and text. The authors also call "a great need" of original research with a primary focus on music and much larger utilization of musical knowledge and insight.

Classification is one of the central issues of artificial intelligence. The application of neural networks to classification has gained widespread attention after outperforming many previous methods (and, in some cases, humans), especially since the significant advancements in the domain made after 2012 [Krizhevsky et al., 2012] in the ImageNet challenge.

One of the classification issues tackled within the music information retrieval domain is musical genre classification using machine learning methods. This problem is non-trivial and poses additional difficulties, as the boundaries between musical genres are often fuzzy, ambiguous and varying due to cultural definition [Scaringella et al., 2006]. However, the effects of such classification are well understood by end-users and useful in discussion of musical categories [McKay and Fujinaga, 2006]. Another problem, also discussed in this chapter, is the overall difficulty of obtaining large, easily-available and complete datasets for MIR purposes, including genre classification. Attempts of solving this problem have only recently been tackled with datasets such as the FMA dataset [Defferrard et al., 2017] and AudioSet [Gemmeke et al., 2017] in 2017, thus enabling the emergence of new solutions in MIR.

The following chapter presents a background of musical genre classification along with original experimental results performed on a substantially bigger dataset than used in much of the reference literature, like [Yang et al., 2020], [Feng et al., 2017], [Sigtia and Dixon, 2014] or [Defferrard, 2015]. Musical explanations are provided for the steps of experimentation and musically insightful conclusions are drawn.

5.1 Related works in music genre classification

A note on the GTZAN dataset

The GTZAN dataset [Tzanetakis and Cook, 2002], since its publication in 2002, has become one of the few well-known benchmark datasets for music information retrieval and, in particular, musical genre classification. It is also by far the most popular dataset in MIR. It consists of 1000 audio clips of 30s length, divided into 10 musical genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock, along with some pre-computed metadata.

Unfortunately, GTZAN also has many flaws. It has been the object of severe criticism [Sturm, 2012a] [Mulongo, 2020], as on top of being a relatively small dataset it contains mislabelled data, noisy audio, several recordings of the same artists and a controversial selection of pieces representing a particular genre. Although it may be useful for certain baseline benchmarking purposes, it is currently considered as deplete of further musical insight and not representative for real-world applications. Despite that, it is still commonly used by researchers and seen in published papers. [Sturm, 2013b] implicitly states that "few researchers have ever listened to it and critically evaluated its contents". Furthermore, out of over a hundred published papers analyzed in [Sturm, 2013b], only five indicate the authors have listened to the music in the dataset and *no* paper explicitly considers the musical content of GTZAN in the evaluation. The authors of the survey also highlight the need for more musically informed research in MIR.

Classic machine learning approaches

Traditional approaches to the issue of genre classification have been well documented and compared in surveys like [Li et al., 2003], [Schedl et al., 2014] and [Sturm, 2012b] and the content-based and real-world application focused [Bahuleyan, 2018], among others.

Some interesting works include [Ellis, 2007b], where chroma feature is used along with Gaussian models. [Mandel and Ellis, 2005] proposes the usage of support vector machines and MFCC features. [McKinney and Breebaart, 2003] uses Gaussian-based quadratic discriminant analysis along with various representations of music, including standard low level signal parameters, like zero crossing rate, band energy ratio, spectral centroid and spectral rolloff. The discrimination capability of several features for various audio signals, including music, have been also tested and benchmarked in [Li et al., 2001] using an audio processing pipeline with feature extraction and a Bayesian classifier.

An interesting recent work is [Murauer and Specht, 2018], where extreme gradient boosting (XGBoost) was used on the FMA dataset with a set of high-level features extracted using the Essentia framework. The proposed method reportedly outperformed several alternatives, including neural networks. The authors however state they used few small layers in their neural networks, possibly resulting in their poor performance when compared to ensemble methods.

Deep neural networks

Several efforts of using deep neural networks have been made in the domain of MIR and music classification. [Sigtia and Dixon, 2014] analyze the use dropout, Hessian-free optimization and the usage of sigmoids versus ReLUs for genre classification using deep neural networks and the GTZAN dataset. [Defferrard, 2015] applies autoencoders for learning musical features and audio classification, again using the GTZAN dataset. [Choi et al., 2015], further expanded in [Choi et al., 2016], proposes an auralisation mechanism for features learned by training convolutional neural networks on audio, showing that deep layers tend to capture textures rather than shapes and lines.

[Chiliguano and Fazekas, 2016] use deep convolutional networks and estimation of distribution algorithms for genre estimation and music recommendation. The approach is evaluated on a subset of the Million Song Dataset. [Park et al., 2017] propose a deep convolutional neural network and a siamese network for representation learning of artist features. [Kim et al., 2018] propose a deep convolutional neural network with label pre-processing and extraction of artist group factors as learning targets. They train a multi-task network to jointly predict the artist group and genre, proposing also very interesting insight into the noise of genre labels and the correlation of artist and genre.

[Feng et al., 2017] and [Yang et al., 2020] have used a paralleling recurrent convolutional neural network (PRCNN [Choi et al., 2017b]) for music genre classification, again using the GTZAN dataset.

5.2 Proposed solution

The following section describes results of original experimental work in music genre classification and proposition of a new solution using deep learning on visual features in graphical representations of music. The section expands on the results presented in [Modrzejewski et al., 2020]. The experiments were performed on a large, challenging, modern MIR dataset - the FMA medium dataset [Defferrard et al., 2017]. Three neural network models have been trained and tested on the dataset. In order to address the need of a large utilization of musical insight within MIR research [Choi et al., 2017a], musical analysis is provided for the performed experiments.

5.2.1 Dataset

The dataset used for the experiments with genre music classification was the FMA dataset [Defferrard et al., 2017] dataset, which contains a massive library of songs. The authors of the dataset attempt to fill the gap in benchmark datasets in the domain of music information retrieval: the music contained in the dataset has an open license, is carefully selected and contains mostly good quality audio. The music is labelled by a top-level genre with multiple subgenres. Rich metadata is also provided. The FMA provides access to four sets of songs, as shown in Table 5.1:

Table 5.1: FMA datasets sub-datasets [Defferrard et al., 2017]

name	number of songs	sample length	number of genres	size of dataset
FMA small	8000	30 sec.	8	7,2 GiB
FMA medium	25000	30 sec.	16	22 GiB
FMA large	106574	30 sec.	161	93 GiB
FMA full	106574	full length	161	879 GiB

Upon listening to the provided samples, the FMA medium dataset was chosen for experimental work, along with a subset of 8 common genres. The dropped top genres were therefore: instrumental, international, old-time/historic, country, soul-RnB, spoken, blues and easy listening. Instrumental, international and old-time genres were dropped because of their relative conceptual difference with the rest of the genres (based more on the style, like "rock" rather than a quality like "international"), while the other dropped genres contained very few examples compared to the smallest remaining genre, which was jazz. This selection is therefore based on an interesting combination of sample counts in the dataset with distinguishable qualities of the music itself, as described in further sections. The selected genres and sample count per genre is shown in Table 5.2:

Table 5.2: Number of samples per genre in the chosen subset.

genre	count
classical	447
electronic	3851
folk	1131
hip-hop	1922
jazz	286
pop	646
punk	1392
rock	2380
Σ	12055

The dataset is deliberately unbalanced in order to model real world disproportions between genre popularity and quantity - the creators of the dataset state single top genre classification on the unbalanced FMA medium subset as an implicit challenge for researchers [Defferrard et al., 2017] [Defferrard et al., 2018].

Surprisingly, efforts of oversampling the minority classes and undersampling the majority classes on the FMA dataset have already been shown to decrease the overall classification score with various methods when utilizing a spectrogram approach [Valerio et al., 2018]. The authors note an increase of the individual results for the minority classes, however, also note a decrease for the overall results. They also predict the problem may lie in the actual musical qualities of the audio excerpts, a thesis that is supported in the following sections and elaborated and expanded on.

5.2.2 Data processing and musical analysis

For the purpose of music genre classification, visual representations of chromagrams and spectrograms have been extracted from the training samples. As described in Section 2.2.2, chromagrams bear close visual resemblance to the score of a particular tune and present information about the energy carried by each of the pitches within a flattened octave context. Spectrograms, as described in Section 2.2.2, are heatmap visualizations of the time-frequency representation of a signal. The following subsections explore and analyze the data present in the FMA medium dataset in the context of their character and correlation to the images. The presented spectrograms and chromagrams for selected samples of rock, pop, jazz, folk, classical, punk and hip-hop music were extracted by Jakub Szachewicz using the *librosa* Python library, while under diploma thesis supervision of the author.

Electronic

Electronic music is a wide genre, distinguished mostly by the dominant usage of electronic instruments, such as sequencers, electronic drum machines and synthesizers (additive, subtractive, FM, sample-based and wavetable, among others). Computers, software instruments and rich usage of effects is also present. Electronic music will usually have a very stable tempo and very regular song structure. Overall, a high presence of bass will be present in the mix of the song, as seen in the spectrogram in Figure 5.1. Production values and overall sound quality will usually be very high, as a demonstration of the electronic musician's production proficiency.

Depending on the particular subgenre, the songs will have a very distinguishable rhythm, either a steady beat similar in structure to one played by a live drummer, or a driving bass drum quarter note pulse, as in house and techno subgenres. The samples in the dataset contain both of these rhythmic approaches.

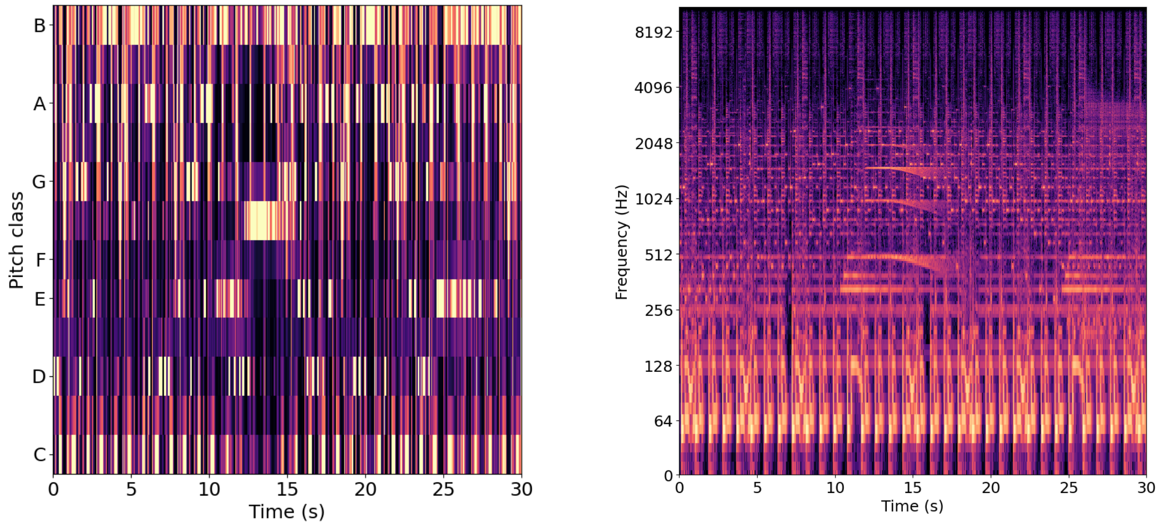

Figure 5.1: Chromagram and spectrogram for an electronic music sample. Both representations include heatmap visualization of magnitude. The axis descriptions were omitted in all subsequent images for better clarity.

Rock

Rock music is generally characterized by the pronounced usage of electric guitars, with a steady rhythm in a wide spectrum of tempos. The arrangements vary from very simple to very rich, while the song structures are mostly regular. The mixes are usually very balanced in all frequency ranges. Many rock artists utilize an aggressive, bass heavy sound. Furthermore, the vocals almost always play a significant role in the music and are very pronounced. Figure 5.2 presents a chromagram and spectrogram for a rock sample from the dataset.

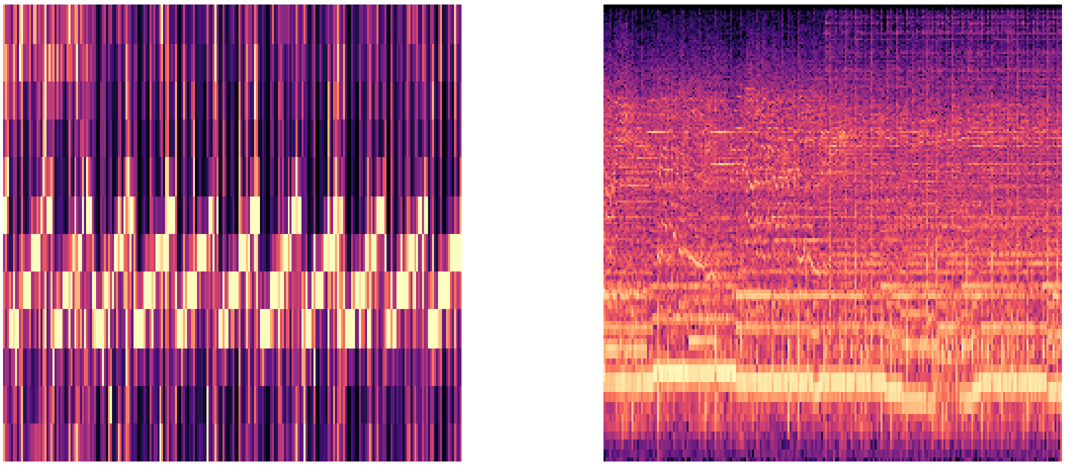

Figure 5.2: Chromagram and spectrogram for a rock music sample.

Pop

Pop music will have an overall easy-listening character with emphasis on high production values. Certain artists prefer an acoustic sound of guitars and pianos, while many utilize a more electronic approach. The overall tempo is steady, with a repetitive harmonic structure and presence of repeating, "catchy" melodies. Pop music, in most cases, has a very balanced frequency mix, as presented in 5.3. The structured harmony is also visible on the chromagram.

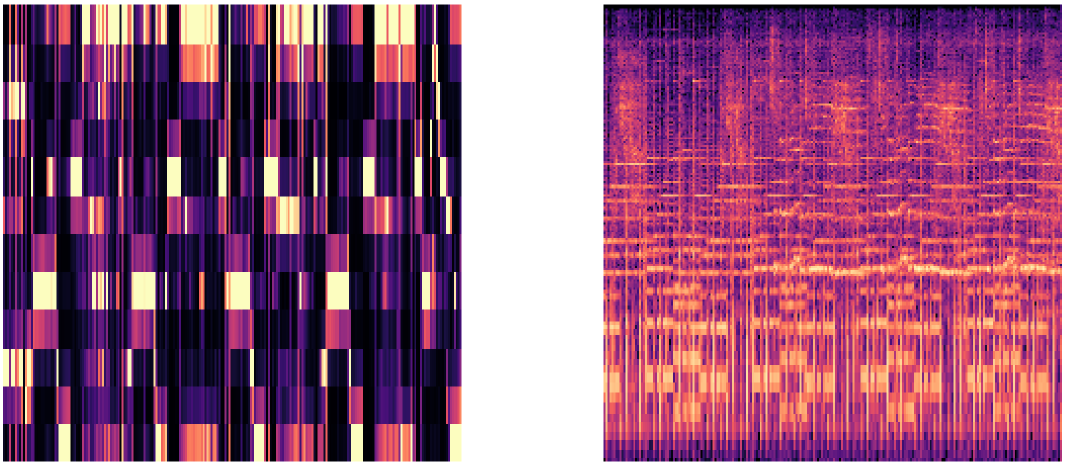

Figure 5.3: Chromagram and spectrogram for a pop music sample.

Jazz

Jazz music will often contain virtuoso level musical performance with high rhythmic and harmonic variance. The traditional jazz band will utilize mostly acoustic instruments, such as the piano, saxophone, trumpet, double bass and drums. Varieties of jazz will often encompass performances of recognizable *jazz standards*, often with vocals and a ballad setting (dramatically different in character than faster and harder *bebop* jazz subgenre).

An important feature of jazz is the quarter note pulse generated by a walking double bass line along with the ride cymbal - Figure 5.4 clearly presents an example of a jazz walking bass line found in the dataset. Jazz will often have a swing feel, as opposed to more rigid rhythmic feeling found in other genres. The recordings are oftentimes live performances. The mix will also have more middle and high frequencies than other genres of music.

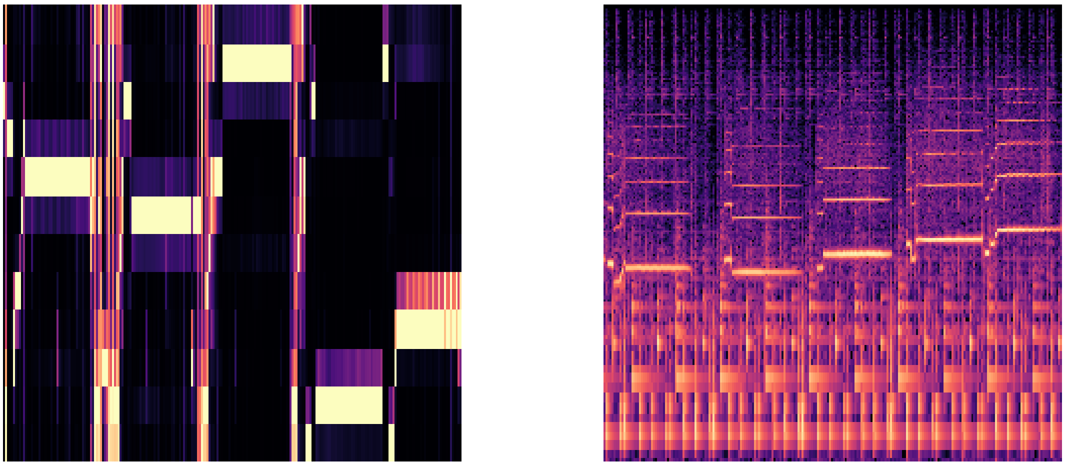

Figure 5.4: Chromagram and spectrogram for a jazz music sample.

Hip hop

Hip hop music will in most cases consist of a repeating *beat* with very rhythmic, rhymed vocals. The overall harmonic variety is mostly very low due to the consistent, repetitive structure of the accompanying music, as visible in Figure 5.5, although the beats often contain samples of jazz and soul music with complex local harmonies. The drums and bass will usually be very well pronounced in the mix.

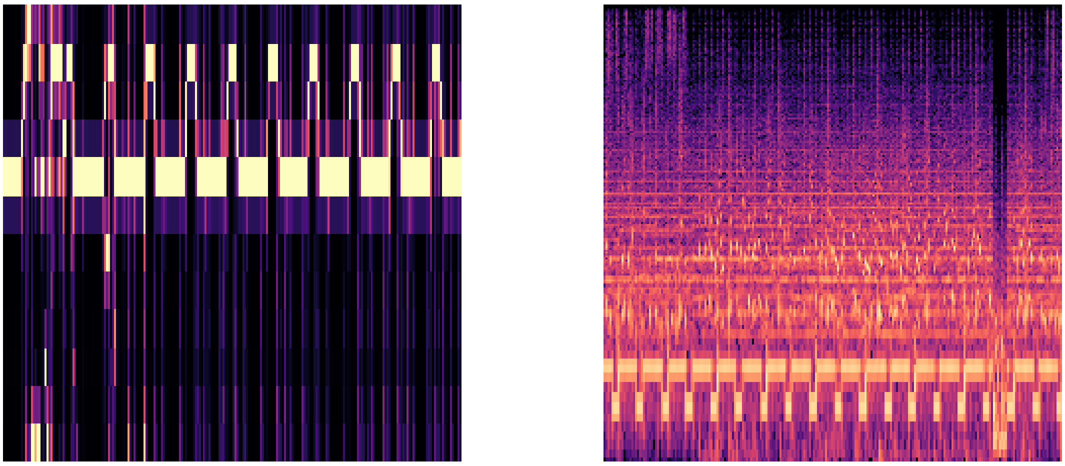

Figure 5.5: Chromagram and spectrogram for a hip-hop music sample.

Folk

Music labelled as folk in the dataset has a very high variety. Folk music has rich arrangements with instruments unlikely to be found in other genres, like accordions and various traditional woodwinds or percussion instruments. Unique, traditional rhythmic and melodic structures are also frequent, as seen in Figure 5.6.

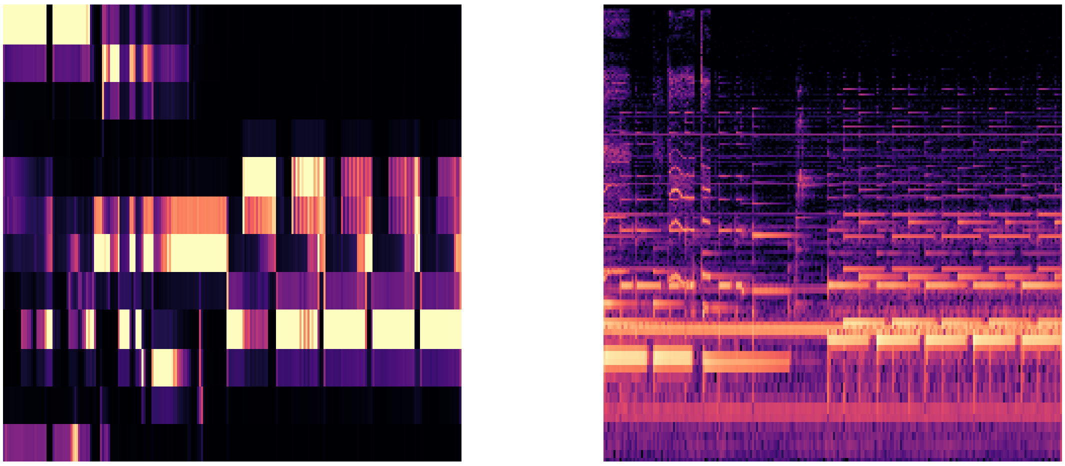

Figure 5.6: Chromagram and spectrogram for a folk music sample.

Punk

Punk music is an subtype of rock music with important historic and cultural meaning. Punk songs usually have a fast tempo and overall high volume. Most arrangements consist of vocals, drums and electric and bass guitar. The overall rhythmic structure is quite dense and general harmonic complexity is usually low - these features are pronounced and can be seen in the respective chromagrams and spectrograms in Figure 5.7.

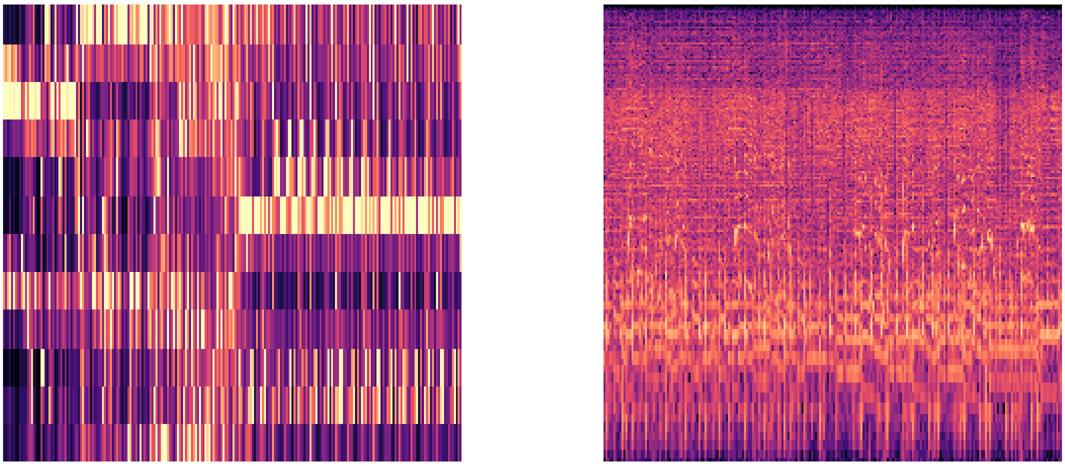

Figure 5.7: Chromagram and spectrogram for a punk music sample.

Classical

Classical music oftentimes has a very high instrumental complexity and virtuoso-level performances. Out of the genres considered in this work, classical music has the most examples of pieces performed on a single instrument, although chamber and orchestral tunes are also present. The harmony and arrangements are usually very rich. Many of the pieces have an unstable tempo with abrupt dynamic and rhythmic changes and a wide range of expressive musical terms, as presented in Figure 5.8. The overall musical qualities and compositional techniques vary depending on the epoch the particular composer has created in.

5.2.3 Models

In order to evaluate the classification capabilities of convolutional models on a bigger, musically relevant dataset with the usage of visual representations, three models were trained: two baseline convolutional models and a paralleling recurrent convolutional network (PRCNN),

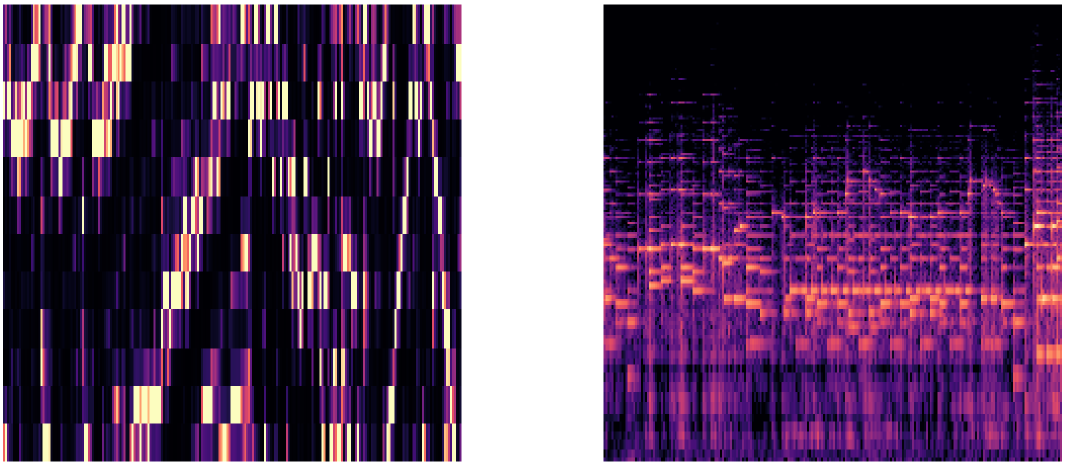

Figure 5.8: Chromagram and spectrogram for a classical music sample.

as an architecture that has been shown to perform well in a similar task on smaller datasets [Feng et al., 2017].

First baseline: convolutional network 1

The baseline simple convolutional network consists of two convolutional layers with 32 and 64 filters respectively, followed by a 2x2 max pooling layer, followed by a 0.5 dropout and a flattening layer. The flattened vectors are then fed into a dense layer, followed by another 0.5 dropout layer and the final dense layer with a softmax activation function. Other than the final layer, ReLU activations were used. The used optimizer was ADAM [Kingma and Ba, 2014] with a learning rate of $2e-4$ and β_1 of 0.9. A schematic view of the network is presented in Figure 5.9.

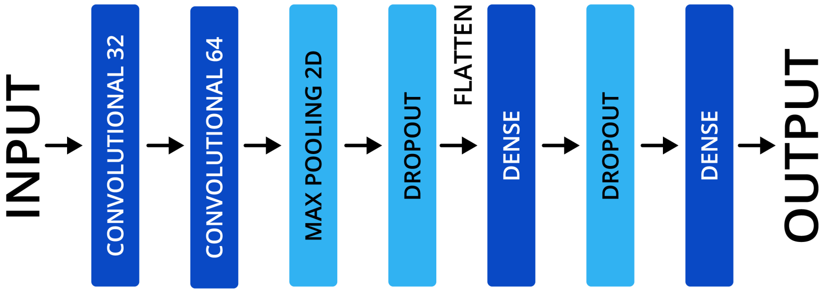

Figure 5.9: Schematic view of baseline convolutional network 1.

Second baseline: convolutional network 2

The second baseline convolutional network has a similar structure, with four convolutional layers instead of two, in order to evaluate the possibility of a gain of performance with a gradually deeper network.

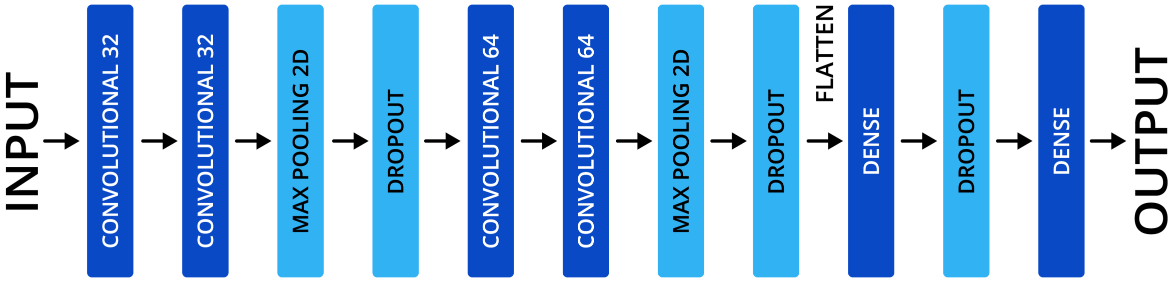

Figure 5.10: Schematic view of baseline convolutional network 2.

The layers have 32, 32, 64, 64 filters, respectively, with max pooling and dropout used in between. Similar to the previous network, ReLU activations were used in all but the final layer. The used optimizer was ADAM with a learning rate of $2e-4$ and β_1 of 0.9. A schematic view of the network is presented in Figure 5.10.

Final model: parallel recurrent convolutional neural network

Augmenting the results obtained by convolutional neural networks for various tasks of music classification by using a hybrid model with a recurrent neural network has been proposed by the prolific MIR researcher Keunwoo Choi et al. in [Choi et al., 2017b], in order to incorporate the temporal analysis capabilities of recurrent network into classification with convnets. [Feng et al., 2017] have used a paralleling recurrent convolutional neural network (PRCNN) for music genre classification and obtained promising results - unfortunately, the benchmark dataset was the aforementioned GTZAN dataset. The GTZAN dataset was also again subsequently used with a similar network architecture in the recent [Yang et al., 2020] (published the same year as [Modrzejewski et al., 2020], where a bigger, more representative dataset is used). In the work described below, a PRCNN with a modified structure (as shown in Figure 5.12) is proposed with the much bigger FMA medium dataset in order to provide musical insight into the capabilities of this type of network.

The network processes data in a parallel fashion with separated convolutional and recurrent blocks. The convolutional block consists of five convolutional layers with 16, 32, 64, 64 and 64 layers, respectively. 2x2 and 4x4 max pooling is used between the layers, and in the context of music classification the pooling layer is responsible for choosing the most prominent musical features, like the amplitude. The used activation functions are all ReLUs and the optimizer is ADAM with the same parameters as the baseline convolutional networks.

The recurrent block uses a max pooling layer and an embedding layer, followed by a BGRU - bidirectional gated recurrent unit layer. The gated recurrent unit (GRU), as proposed in [Cho et al., 2014], is somewhat a variation upon the LSTM. It combines the forget and input gate of the LSTM into a single update gate with addition of a reset gate. It also merges the cell state and hidden state. A gated recurrent unit is conceptualized and presented in Figure 5.11.

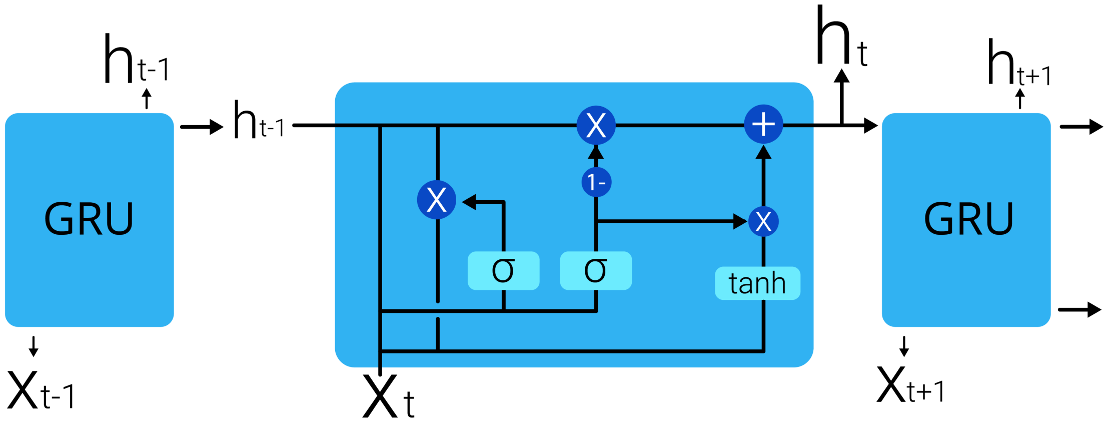

Figure 5.11: Conceptual view of a gated recurrent unit cell with update and reset gates.

The bidirectional gated recurrent unit layer uses layers of forward GRU cells and backward GRU cells, with no direct connection between the former and the latter. The output is produced by both the forward and backward layer. The output of the convolutional and recurrent blocks are then concatenated and fed into a final dense layer with softmax activation.

The structure of the implemented PRCNN is conceptualized and presented in Figure 5.12. The contributions are as follows: in comparison with the original paper [Choi et al., 2017b], an additional convolutional layer is used in order to better capture the local, instance-level acoustic characteristics by the CNN block. Previously unpublished chromagram results, along with spectrogram results, are presented. Both the data (in section 5.2.2), as well as the results are interpreted with original, musically informed insight, which is possible due to the long inputs enabled by the PRCNN. The long input sequence is summarized by the recurrent block, therefore no feature aggregation step is needed. Finally, a larger dataset is used than in [Yang et al., 2020], benchmarking a similar network on the GTZAN dataset, with no musical explanations of the presented results.

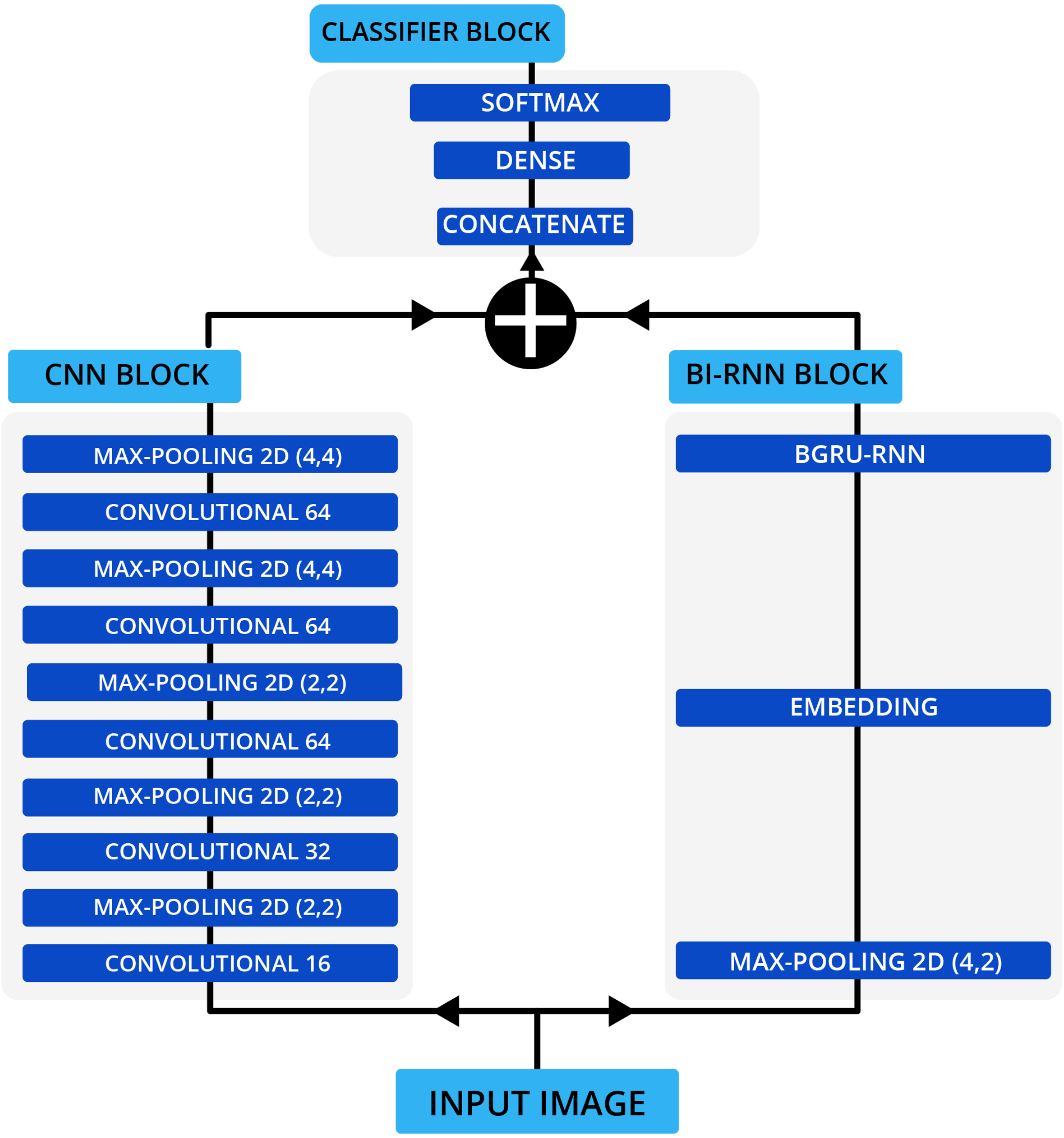

Figure 5.12: Schematic view of the parallel convolutional recurrent neural network for music genre classification.

5.3 Results

The networks were trained on the images extracted from the audio files within the FMA medium dataset. Three aforementioned architectures of neural networks have been used: two convolutional models and the final model, a parallel convolutional recurrent model. The training time for each of the networks was around 20 hours on a NVidia Tesla GPU. Upon training, in total six models have been obtained, as a product of two graphic representations used (spectrogram or chromagram) and three architectures.

The following tables present the precision, recall and F1 score metrics for the network models. The collected metrics are denoted as follows:

- CP - chromagrams precision,
- CR - chromagrams recall,
- C F1 - chromagram F1 score,
- SP - spectrograms precision,
- SR - spectrograms recall,
- S F1 - spectrograms F1 score.

In addition, Figures 5.13 and 5.14 present the obtained confusion matrices for the baseline convolutional network and the PRCNN. The confusion matrices are helpful in analysis of the mistakes made by the networks and drawing musically insightful conclusions from the classification.

Table 5.3: Metrics for the CNN

genre	CP	CR	C F1	SP	SR	S F1
classical	0.67	0.60	0.63	0.70	0.71	0.71
electronic	0.67	0.81	0.73	0.67	0.73	0.70
folk	0.64	0.57	0.61	0.61	0.69	0.65
hip-hop	0.65	0.68	0.66	0.59	0.69	0.64
jazz	0.50	0.10	0.17	0.70	0.15	0.25
pop	0.29	0.05	0.09	0.29	0.06	0.10
punk	0.45	0.40	0.43	0.43	0.39	0.40
rock	0.55	0.57	0.56	0.51	0.51	0.51
average	0.55	0.47	0.48	0.56	0.49	0.49
weighed	0.59	0.61	0.59	0.57	0.59	0.57

Table 5.4: Metrics for the deeper CNN

genre	CP	CR	C F1	SP	SR	S F1
classical	0.66	0.87	0.75	0.90	0.54	0.68
electronic	0.65	0.80	0.72	0.71	0.79	0.75
folk	0.57	0.72	0.64	0.55	0.68	0.61
hip-hop	0.80	0.49	0.61	0.65	0.66	0.66
jazz	0.80	0.07	0.12	0.83	0.10	0.18
pop	0.17	0.05	0.08	0.35	0.09	0.14
punk	0.44	0.38	0.40	0.46	0.19	0.26
rock	0.51	0.58	0.55	0.50	0.69	0.58
average	0.57	0.49	0.48	0.62	0.47	0.48
weighed	0.60	0.60	0.58	0.60	0.61	0.58

Table 5.5: Metrics for the final PRCNN model

	CP	CR	C F1	SP	SR	S F1
classical	0.45	0.73	0.56	0.79	0.78	0.78
electronic	0.63	0.63	0.63	0.80	0.74	0.77
folk	0.47	0.61	0.53	0.47	0.84	0.60
hip-hop	0.66	0.45	0.54	0.87	0.51	0.64
jazz	0.22	0.03	0.05	0.25	0.01	0.02
pop	0.18	0.02	0.03	0.18	0.06	0.09
punk	0.35	0.25	0.30	0.64	0.14	0.23
rock	0.39	0.60	0.50	0.48	0.87	0.62
average	0.42	0.41	0.40	0.56	0.50	0.47
weighed	0.50	0.50	0.49	**0.65**	**0.62**	**0.60**

5.3.1 Results analysis

The performance of the two purely convolutional networks was comparable, with a very similar distribution of errors (hence the confusion matrix for the larger CNN was omitted above). The CNN' performance was similar with both representations of music, leading to the conclusion that the spatial analysis of harmonic and compositional values of chromagrams using convolutional networks is a valid approach. Using the chromagram representation, the CNN obtained the best results in the classical, electronic, folk, rock and hip-hop genres - that is, genres with either very repetitive harmonic and melodic structure or with a highly varying and unique one, as seen in Figure 5.13. The networks had significant difficulties in the classification of punk music, often not being able to distinguish it from rock based on the chromagram representation.

Regardless of representation, pop chromagrams were mostly classified as rock or electronic music by the CNN, with a significant amount of hip-hop samples also classified as electronic,

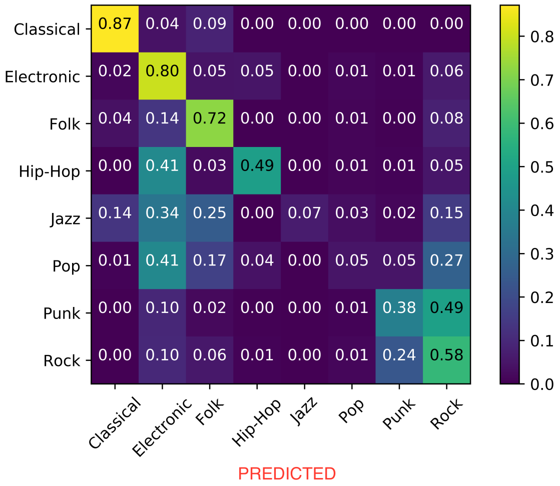

Figure 5.13: Confusion matrices for chromagram (left) and spectrogram (right) for the baseline CNN.

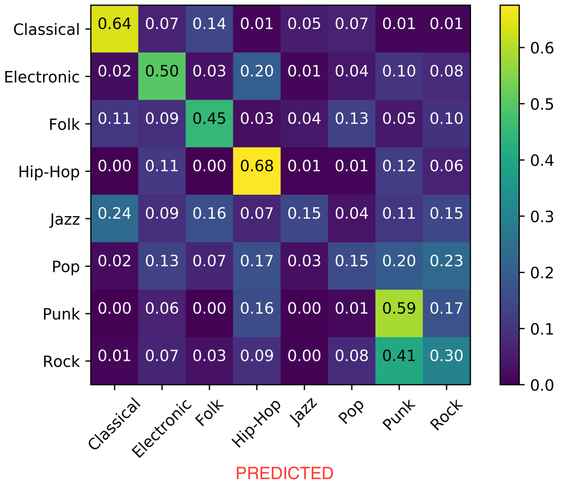

Figure 5.14: Confusion matrices for chromagram (left) and spectrogram (right) for the PRCNN.

as seen in Figure 5.13. This is easily explainable by the harmonic similarity and repetitiveness of these genres and the character of CNNs, which analyze the spatial features of images. The PRCNN, much more capable of learning temporal structures, classified pop music mostly as rock, probably as a result of the presence of similar, repeating, "catchy" melodies in both of these genres, as seen in Figure 5.14.

Classical samples, although having an even lower count in the unbalanced dataset than pop samples, were surprisingly quite easy for to classify using either of the approaches and either network architecture, as seen in Figures 5.13 and 5.14. This leads to many conclusions. First of all, the dense harmony, fluent structure, virtuoso passages and sophisticated compositional techniques of classical music are well represented in the chromagram approach, which closely resembles musical score. Indeed, a musical score for a classical piece of music would not really resemble the score of any other genre when considering actual music. The PRCNN however misclassified some of the classical pieces as jazz music - this is especially interesting, as some of the later classical pieces, like for instance the works of impressionists, were directly an inspiration for jazz composers and had heavy use of harmonic structures that would later become a staple in jazz music. Using the spectrogram approach the classification results for classical music were better in the case of the PRCNN, and worse in the case of the CNN, as seen in Tables 5.4, 5.3 and 5.5. The most often misclassification being folk music - possibly due to the sparsity in frequencies of tunes performed on a single instrument or a small ensemble.

Many interesting conclusions may be drawn for the misclassification of the jazz samples. The obtained precision was in all cases better than the recall, meaning that if the network chose jazz, it was usually right, but also it missed a lot of the jazz samples. The misclassification of jazz as classical by the PRCNN with chromagrams is the result of similar fast passages and virtuoso performances. The PRCNN also misclassified jazz as folk music - possibly due to the acoustic character and similar frequency character of the genres. The CNN, on the other hand, misclassified jazz mostly an electronic, as seen in Figure 5.13. This may be attributed to the *walking bass line*, a way for the double bass to play in a jazz ensemble. The bass plays steady quarter notes with overall little rhythmic variation (although high harmonic variation) - when translated to a graphical format, this might have been interpreted as very similar to the pulsating, regular patterns of electronic music, for instance the kick drum.

The PRCNN obtained high recall for rock music and a relatively low precision, meaning the network often defaulted to rock music. This, again, confirms the temporal capabilities of the recurrent block in the PRCNN, as similar song structures are found in rock, punk and pop music.

The best of the proposed models in terms of F1 score was the PRCNN with a spectrogram, achieving the score of 0.60. It is worth noting that in the classification challenge organized by the authors of the FMA dataset [Defferrard et al., 2018] the top obtained F1 in the leaderboard was 0.64, with most of the entries on the leaderboard obtaining a score of around 0.60. To the best of the knowledge of the author, this is the first music genre classification method with such thorough musically insightful analysis, performed on the FMA dataset and a parallel recurrent convolutional architecture. In comparison, [Yang et al., 2020], which was published the same year as [Modrzejewski et al., 2020], does not provide any musical explanation for the obtained results and benchmarks the networks on the GTZAN dataset.

The results and contributions can be summed up as follows:

- upon classification, a F1 score comparable with the FMA challenge leaderboard is achieved with a modified parallel recurrent convolutional network,
- previously untested chromagram representations (bearing close resemblance to the sheet music of a given piece) are tested, evaluated and analyzed in the described music classification context,
- parallel recurrent convolutional network architecture is benchmarked against purely convolutional neural networks on a bigger dataset than previously described in literature [Yang et al., 2020],
- original guidelines for future research in interpretability and explanation of the results of music classification methods are presented, with in-depth analysis of the correlation of the results with human-perceivable characteristics of the input music,
- misclassifications for purely convolutional networks and the PRCNN architecture are interpreted for both the chromagram and the spectrogram representations, showing and explaining the advantage of augmenting instance-level acoustic feature extraction via the CNN block by the sequence analysis capabilities of the RNN block in certain genres.

The overall conclusion drawn from the experimentation is that the issue of musical genre classification indeed should be treated not only as a matter of benchmarks and algorithmic effort, but also should be considered through the deep insight it provides into our understanding of music labels and musical perception.

Chapter 6

Conclusions

6.1 Summary

The focus of this thesis was the application of artificial intelligence methods to artistic content creation and analysis. The main focus of the conducted research was to verify the hypothesis that neural networks may be used to support and enhance music creation, in particular without the need of huge models and massive computing power, which is a trend in many state of the art solutions [Roberts et al., 2019], [Hawthorne et al., 2018], [van den Oord et al., 2016] or the GPT2-based [Payne and OpenAI, 2019]. Due to the author's background in musical creation and performance, the research has been conducted with an intention of high usability and lightweight character of the proposed solutions, which is reflected in the obtained results.

Fulfilling the aims of this thesis has been achieved through the proposition of solutions for chosen subdomains of applications of artificial intelligence to musical content: creativity augmentation and music information retrieval. Two solutions for creativity augmentation have been proposed, one for generating new musical content (in Chapter 4) and one for musical style transfer (in Chapter 3). Both solutions allow the creation of new, previously unheard music, both in terms of the notes composed and the sonic qualities achieved. At the time of publishing, to the best of the authors' knowledge, the presented methods are original propositions. The presented methods were compared with existing approaches, along with their advantages over said approaches. Also described is the scope of possible enhancements to a creative's workflow obtainable via implementing the proposed solutions.

Additionally, a solution for music classification has been presented (in Chapter 5) along with original critical analysis and discussion on the ground of music information retrieval. Results of music classification over several genres performed with different models and using different representations of music on a large dataset are presented and discussed. The conclusions of the performed experiments adhere and add to the highly critical stream of thought in machine learning in music proposed by [Sturm et al., 2019].

Additional emphasis has been put on preserving the musical context of the proposed solutions and their analysis in terms of said context. Several formats of music for the needs of artificial intelligence have also been considered and described. Although there is much less datasets used for machine learning in music and the available ones are sparser than the ones used for image tasks, the results obtained with using the selected datasets have allowed to prove the hypothesis stated in this thesis.

In conclusion, all of the assumed aims of this thesis have been achieved.

6.1.1 Original contribution

In terms of the original contribution to the state of the discipline described in this thesis, the following may be enumerated:

- proposition of a novel method for timbral style transfer in music, along with a suitable data processing pipeline. The musical representations used were MIDI files synthesized into raw audio and transformed into spectrograms. The experiments were performed for piano to guitar style transfer, but the method is suitable for style transfer between any pair of instruments. The method, published in [Modrzejewski et al., 2021] and is presented and expanded on in Chapter 3.

- proposition of a novel method for original music content creation using a generative-adversarial model and a graphical piano roll representation obtained from MIDI files. The method has been published in [Modrzejewski et al., 2019] and is presented and expanded on in Chapter 4.

- application of neural network models to the issue of music classification upon larger datasets than commonly used for state of the art comparisons, along with vital contribution to the ongoing critical discussion about the musical context of such solutions. The musical representations considered in the experiments were raw audio for analysis and spectrogram and chromagram images for training. The method has been published in [Modrzejewski et al., 2020] and is presented and expanded on in Chapter 5.

The primary contributions are reflected in three positions of peer-reviewed literature published on international conferences. Each of the contributions was also presented before a live audience at the associated conference and was met with a warm reception.

In addition, the above work has been inspired by the author's previous research and experiments with highly applicable solutions of artificial intelligence. These have also been peer-reviewed and published on international conferences as the following positions of literature:

- [Modrzejewski and Rokita, 2018b] and [Modrzejewski and Rokita, 2018a] describe and propose solutions for artificial intelligence powered conversational agents. The presented critical analysis and propositions are applicable both to text-based and audio-based systems.

- [Modrzejewski and Rokita, 2019] proposes a generic artificial intelligence solution for agent steering in computer game programming.

6.1.2 Primary results summary

The summary of the results and conclusions obtained from the experimental and analytical work described in the thesis is as follows:

- Autoencoder networks are a viable solution for timbral style transfer in music, although in a basic form they introduce the need of splitting musical samples into artificial chunks, resulting in unwanted sonic artifacts and lower continuity of the produced samples. The usage of LSTM layers allows to alleviate this problem and significantly improves content perseverance, resulting in much higher quality of style transfer. The samples produced by the approach presented in [Modrzejewski et al., 2021] persist the melody and structure of the input music and have the desired timbre and sonic qualities of the output instrument. The proposed LSTM autoencoder architecture is also fast to converge and can be used without heavy computational resources.

- Generative-adversarial networks provide a framework for creating original, harmonically rich, long musical samples, as confirmed in [Modrzejewski et al., 2019]. A significant advantage of the proposed approach is that it allows to generate many concentrated and cohesive musical ideas. The ideas often span an even number of bars, thus providing a great source of short loops for the composer to choose from. Existing solutions for a similar problem create phrases that are much simpler and shorter, with just basic harmony and certain pre-defined resolutions. Previous approaches, based on more „traditional" machine learning solutions, like genetic algorithms and Markov models (as in i.e. [Van Der Merwe and Schulze, 2010], [Li et al., 2019], [Bell, 2011], [Herremans et al., 2015]) on the other hand have a narrower scope of operation, with a focus of generation of certain aspects of music (short melodies, operating within certain scales or keys, matching a melody to given harmony etc.). The harmonic richness of phrases generated with generative-adversarial networks and a graphical representation of MIDI directly serves as a mean of enhancing and inspiring the composer's creativity.

- Generative-adversarial networks are a viable solution for learning and reproducing qualitative features of music, like harmony and rhythmic structure. Provided with sufficient training data, this network architecture allows to generate long harmonic phrases, chord progressions, arpeggios and melody and bass lines.

- The possibility of creating lightweight models for augmenting the creative intelligence of a composer with artificial intelligence has been confirmed. The augmentation may occur in direct creation of musical ideas and content, sound shaping and musical information retrieval and analysis. The focus on lightweight approaches is also well-aligned with the emerging discussion over the carbon footprint of the massive computational resources used when training machine learning algorithms.

- An ongoing key issue in applying artificial intelligence to music is the representation of music. The audio domain can be seen as somewhat suspended between the domains of images and text. Music is sequential and can be seen as a time series, like text. On the other hand, there are various graphical formats that audio can be compressed into, with a varying degree of information loss. The approaches described in this thesis focus on MIDI, raw audio and selected graphical representations, including the unorthodox piano roll images. Music stored in the MIDI format has proved to be an excellent, versatile source of training data for neural networks. Although deplete of sound itself, MIDI allows to incorporate dynamics, rhythm, pitch (with bends), control change and other musical information in a very accessible format. Since its introduction, MIDI has been a standard both in musical device communication and in musical composition, especially in the electronic realm. The usage of MIDI ensures the proposed approaches are close to the context of creativity augmentation.

- Convolutional and recurrent neural networks work well for musical genre classification experiments conducted on a large, well-described and demanding dataset. Various models can be successfully trained to distinguish the most common musical genres with visible and interpretable arising mistakes [Modrzejewski et al., 2020]. When using graphical representations to closely resemble the time-frequency and time-energy dependencies in music, the mistakes of the networks will follow closely the chosen set of musical differences than cannot be clearly represented within the said representations. Awareness of the distinct musical features is key in the analysis of the obtained classification results and is still a missing factor in many music information retrieval efforts.

- A significant amount of research in music classification is flawed by the usage of the benchmark GTZAN dataset, which is very sparse and does not reflect the actual plethora of genres found in music [Sturm, 2013a]. A similar issue occurs for other subproblems of applying machine learning to music composition with the Bach Doodle dataset, which, although rich in samples, offers a one-sided view on composition. The extremely high metrics obtained in some research on those datasets, although valuable in terms of improving the state of the art of machine learning in music, are often quite far away from the original, musical context of the research. It is safe to say that such benchmarks, with the GTZAN dataset in particular, should be treated with additional consideration.

- Artificial intelligence algorithms provide new tools for the multimedia creator and allow for development and deployment of new types of creator and user experiences, as has been presented and described in the previous works [Modrzejewski and Rokita, 2018a], [Modrzejewski and Rokita, 2018b] and [Modrzejewski and Rokita, 2019]. These algorithms, at some point, fall into the boundaries of applied programming and deployment details. The context of the practical applications will oftentimes dictate the algorithm design and particular usage.

6.2 Discussion and directions for further research

Neural networks and their applications are evolving very rapidly, thus not yet allowing for a high degree of agreed upon formalism in how to create them: implementation can still be associated with quite a high dose of trial and error due to loose principles of their design and only some practical, general rules of thumb [Karpathy, 2019]. Although the solutions discussed in this thesis were designed to be lightweight, the training of neural networks is still a time-consuming task, especially as the models grow. An interesting approach is using architectures without batch normalization [Brock et al., 2021], which instead use an adaptive gradient clipping technique. The authors were able to improve the state of the art accuracy of their model while significantly speeding up the training process. Perhaps similar ideas may be adapted for certain musical tasks.

In terms of generating previously unheard musical content, the most solutions which are currently the most promising will most certainly require huge computational resources. Generative pre-trained transformer models have shown to adapt well to generating sequential data of various kinds, including music [Dhariwal et al., 2020]. Another very promising area of research lies in multimodal representations, where for instance image data can be connected with a text description [Radford et al., 2021]. In a similar manner, music could be generated based on natural language descriptions with a high degree of sophistication, including emotional, stylistic and auditory quality specifications. This research path has not yet yielded results in the musical domain and most likely will go in pair with rethinking a restating the initial issue of musical content generation: expanding the amount of control the end-user will have over the deep models output in order to express their arbitrary creative goals.

Recent advances in generative adversarial networks have also brought the idea of music inpainting. In a recent paper a GAN has been used to restore short pieces of missing audio data, with lengths from a few milliseconds to a few seconds [Marafioti et al., 2020]. Although the authors state that their solution introduces audible artifacts rated between "not disturbing" and "mildly disturbing", they also point the solution represents a framework for further improvement.

Another interesting idea would be the usage of CycleGAN architectures for selected types of musical style transfer. CycleGAN networks are basically composed of a number of generators and a number of discriminators and make sure the generated content still falls into the initial domain. This allows for very interesting transformations on image and video content and prevents model collapse, a special case of overfitting where the generator learns to produce a single image that is able to fool the discriminator perfectly in every case, fulfilling the training goal, but rendering the model useless. In [Kaneko and Kameoka, 2018] and [Kaneko et al., 2019] CycleGAN models have already been used with promising results for voice conversion, ie. changing the voice timbre and accent of speech recordings. It is likely that modifications of this architecture, paired with appropriate data representations, would allow to obtain interesting results also in the case of musical recordings.

The issue of music classification and music information retrieval also has interesting paths for future research. First of all, taking into account the emerging need for explainable artificial intelligence applications, the author would like to express his hope that more research will provide in-depth, musically informed explanations of the obtained results.

Good results of representation learning for music (and other types of audio) have recently been documented with the introduction of L^3 embeddings [Cramer et al., 2019]. These embeddings allow to quickly create baseline models and apply, for instance, *scikit-learn* algorithms to audio in a simple way. They have also been used for certain tasks of music classification, like music emotion recognition in [Koh and Dubnov, 2021]: although in some cases numerous, hand-crafted features of a particular dataset work better for this task, the authors have achieved favorable performance over multiple datasets. Currently, the further development and application of L^3 embeddings is hard to predict and is an interesting and promising research direction.

Modern streaming services posess near limitless collections of music and, effectively, music metadata. For instance, Spotify has in 2014 acquired The Echo Nest, a company providing vast musical fingerprinting metadata. The acquisition enabled Spotify to introduce powerful features like personalized discovery playlists, with predictions of what music the user may like based on their listening history. Some analysis of the metadata claim that it also incorporates "trivial" and "random" data aggregated by various means [Eriksson, 2016] - nevertheless, the proprietary musical recognition capatilities of streaming services seem to currently push the boundaries of what is applicable with music fingerprinting and clustering. The recommendation algorithms for music discovery and infinite playlist creation have become an impressive, well-established commodity used daily by a huge userbase.

Future research in music classification seems to be in equal parts a problem of algorithmic efforts and our understanding of musical labels. As new genres and methods of expressions emerge from the creative minds of musicians, many musical and stylistic boundaries get blurred. The author predicts that a dichotomy in music classification will continue to occur, according to the size of the datasets: one path of research based in the huge-scale recommendation systems implemented by streaming services posessing near unlimited musical libraries, and the other path based in specific, well-defined environments of highly granulated collections.

A very promising area of research may also unfold with the upcoming MIDI 2.0 standard [Lehrman, 2020]. At the time of writing, the specification for MIDI 2.0 has already been made public and the standard is waiting for its launch [MIDIAssociation, 2021]. It is the first true update to the MIDI standard since its introduction in 1983 and it brings many improvements and new possibilities. One of them is MIDI Capability Inquiry, which will allow MIDI gear to communicate with other MIDI gear and find out about its functionality. Also, MIDI 2.0 controller will now offer 32 bit resolution, which is much greater than the current 7-bit resolution. This will allow for new forms of automation, and a much smoother, more "human" feel of many control effects. The new specification allows for per-note controllers, so that when you hit a key or playing surface, the controller can define how that particular note should respond. Think of this as the digital equivalent of holding a bow differently when playing different cello notes or like changing the pick angle when plucking a guitar string. Including such richness into a format that we already know serves well for machine learning purposes may unfold great improvements in the audible products of the algorithms with little to no programming effort.

Although the history of music has been molded by generations of artists, certain innovations were possible only after certain technological breakthroughs. From the invention of particular musical instruments, through the research on physical qualities of sound up to relatively new inventions like audio recording and electronic instruments, the history of music is closely connected with the history of technology. Looking at modern times, neural networks and artificial intelligence algorithms are becoming more and more present in our lives by the means of a variety of products deployed to numerous markets. The technology of creating them is slowly becoming more accessible, as is programming in general. On a finishing note, the author would like to express hope that these skills and ideas will continue to find ways into many artistic realms, augment human creativity and perhaps initiate entirely new means of human expression, as other technological advances clearly have already done throughout history.

Appendices

Appendix A

Author's publications

The results of experimental work, described thoroughly in subsequent chapters of this thesis, have been presented in the following pieces of peer-reviewed literature:

- Mateusz Modrzejewski, Konrad Bereda, Przemysław Rokita, *Efficient recurrent neural network architecture for musical style transfer*, International Conference on Artificial Intelligence and Soft Computing. Springer, 2021,

  marked as [Modrzejewski et al., 2021].

- Mateusz Modrzejewski, Mateusz Dorobek, Przemysław Rokita, *Application of deep neural networks to music composition based on MIDI datasets and graphical representation*, International Conference on Artificial Intelligence and Soft Computing. Springer, 2019,

  marked as [Modrzejewski et al., 2019].

- Mateusz Modrzejewski, Jakub Szachewicz, Przemysław Rokita, *Application of Neural Networks and Graphical Representations for Musical Genre Classification*, International Conference on Artificial Intelligence and Soft Computing. Springer, 2020,

  marked as [Modrzejewski et al., 2020].

Chapters 3, 4 and 5, respectively, contain a detailed description and discussion of the experimental work. Furthermore, the author's previous experiments with lightweight models of artificial intelligence, have been also published and presented in the following pieces of peer-reviewed literature:

- Mateusz Modrzejewski and Przemysław Rokita. *"Critical Analysis of Conversational Agent Technology for Intelligent Customer Support and Proposition of a New Solution."* International Conference on Artificial Intelligence and Soft Computing. Springer, 2018.

  marked as [Modrzejewski and Rokita, 2018a]

- Mateusz Modrzejewski, Przemysław Rokita. *"Graphical interface design for chatbots for the needs of artificial intelligence support in web and mobile applications."* International Conference on Computer Vision and Graphics. Springer, 2018

  marked as [Modrzejewski and Rokita, 2018b]

- Mateusz Modrzejewski, Przemysław Rokita. *"Implementation of generic steering algorithms for AI agents in computer games."* Intelligent Methods and Big Data in Industrial Applications. Springer, 2019.

  marked as [Modrzejewski and Rokita, 2019]

Appendix B

Author's artistic background

The author's understanding and motivation for music-centered artificial intelligence research has a background of professional-grade musical performance (mostly on drumset), recording and creation, which has been a key inspiration for the interest in the issues described in this thesis. The author would like to highlight his key achievements in the domain as:

- representing Poland at the *I European Jazz Festival* in Guangzhou, China - March 2019 (with Michał Milczarek Trio, co-organized by Jazzpopolsku and the Consulate General of Poland in Guangzhou, China),
- representing Poland at the *Hue Festival 2017* in Hue, Vietnam - May 2016 (with Michał Milczarek Trio, co-organized by Jazzpopolsku and the Embassy of Poland in Hanoi, Vietnam),
- nomination for *Fryderyk* music prize in rock category - 2021, with Majka Jeżowska, for *Live at Pol'and'Rock 2019* live album (Złoty Melon, Mystic Production),
- *Jazz Phonographic Debut of the Year* prize - 2014, with Michał Milczarek Trio for *Squirrels and Butterflies* LP, award of the National Institute of Music and Dance by the Ministry of Culture and National Heritage of Poland,
- laureate of *Kultura w Sieci* stipend of the Ministry of Culture and National Heritage of Poland, 2020
- 1st place in the drumset contest of the International Percussion Festival in Opole - 2012
- having played over 600 concerts with audiences up to 150,000 people in 8 countries around the world (with various artists).

Appendix C

Legal notice on musical copyright

The authorship of music, as a form of art and intellectual property, is protected by various forms of copyright. A crucial stream of revenue for composers and artists comes from royalties paid by institutions who manage and maintain these copyrights. But what is the current legal status of enhancing music creation with artificial intelligence?

The short answer is: the road is being paved as we walk on it. We have already seen legislation falling behind on improvements in artificial intelligence algorithm deployment, for instance in the case of *deep fakes*. The law does not account for the emerging generative abilities of machine learning solutions. As described by [Makhmutov et al., 2020], more work in global copyright legislature is needed when it comes to joint creations of human and artificial intelligence creativity, depending on how the AI tool was used and trained.

The issue of whether code can be the author of a musical work has been considered for over half a century in the US. In 1966, the US Copyright Office brought up this concern in the section "Problems Arising From Computer Technology" of [Copyright-Office, 1966]. Interestingly enough, the report anticipates the evolution of computational creativity and states one application for a piece of music composed by a computer has already been delivered.

The word "human" is not even mentioned in the US copyright law, and this has already provoked certain cases of unprecedented legal action, as for instance the loud case of the *monkey selfie*. In 2011, newspapers have published selfie images taken by Celebes crested macaques (a species of monkey). The wildlife photographer who arranged the whole photoshoot has in turn undertook legal action, claiming he holds the copyright to the photographs. This lead to a long court trial, in which the United States Copyright Office stated that works created by a non-human are not copyrightable and a final statement by an appeals court that animals cannot legally hold copyrights.

Another extremely interesting legal case is one of a company called Endel [Endel, 2021]. Their product is an application that generates personalized *soundscapes* - very soft ambient music meant for focus, relaxation, background noise, sleep etc. - based on the user's location, time zone and weather conditions. In musical terms, the soundscapes may be characterized as washed out and full of long notes. In 2019 Endel has signed a deal with Warner Music, effectively becoming the first ever algorithm to have had a publishing deal with a major musical label.

Endel's algorithms were used to create 600 tracks on 20 albums published on streaming services. The label is not in control of the product as Endel has retained full ownership of the master recordings, with a 50% royalty split. The company states all of the 600 tracks were made "with a click of a button", with minimal involvement of a producer or composer. The company could easily create an infinite number of albums, and with the increasing popularity of background music playlists in streaming services, is more than certain to proceed in this direction. Endel does not provide a usable tool, instead it produces the end product. The label treats the algorithmic approach exactly the same as it would treat traditional music. Endel's employees were listed as songwriters, as with the current legal status, claiming the authorship is sufficient to be considered the author. Issues occur only when there is any pushback, like in the *monkey selfie* case. Most of the credited employees have never before written a song and do not know how to write one.

An interesting case is also the one of the AI startup Boomy [Boomy, 2021]. Boomy provides its users with an "artificial intelligence-powered one button music studio", where the user is able to create a whole track within a few clicks with no prior musical knowledge or experience, choosing from several styles and mix options. They claim each use of their algorithm will produce a unique output, leaving some of the customization to the user. Their technology is said to be trained exclusively on non-copyrighted material, with a brute-force development approach and evaluation by human listeners. Tracks created using the platform can be automatically distributed to several digital vendors and streaming services. The company owns copyright to the songs created on the platform and manages the royality shares paid to the customers, although the default license allows the users to use their songs for most commercial and non-commercial uses.

Upon the discussed cases, let us consider the following statements as applied to machine learning in music:

- AI algorithms are able to endlessly generate novel, previously unheard music,
- AI algorithms are able to mimic the style of a particular artist or genre,
- AI algorithms may be trained on sets of copyright-protected music,

- AI algorithms may be deployed in end-user software, such as they are invisible to the end-user,
- AI algorithms are only tools in the hands of the end-user and cannot be credited with copyright.

In order for a copyright issue to occur, artificial intelligence would have to produce the finished product, for instance a full song that sounds like an already existing one. Marketing the output of the algorithm as sounding like a particular artist without her or his consent would also generate a persona or trademark violation. But in order for that violation to occur, the general style of a musical piece is not enough: the piece couldn't just sound like a particular artist or mimic her or his style. It would have to sound exactly like a specific song that artist has made.

Another issue is proving a particular algorithm was designed to mimic an artist. Copyright infringement cases always call for proof over plagiarism of musical work. These cases are almost never obvious, even when striking similarities in terms of musical melody, harmony or production qualities occur. Famous legal cases over blockbuster singles were held between Led Zeppelin versus Spirit (*"Stairway to Heaven"*, Robin Thicke versus Marvin Gaye (*"Blurred Lines"*) and Katy Perry versus Flame (*"Dark Horse"*). In the case of *"Dark Horse"*, the music under legal scrutiny was actually over a musical phrase comprised of only eight notes. The court's verdict on that song has actually been overturned after a year of the original rule, which ordered Perry to pay $2.8 million in damages to Flame, showcasing how vague an issue can be over a short phrase. Going back to machine learning - in order to prove an algorithm was trained on a particular song or artist, one would essentially have to reverse engineer a neural network, which is an impossible task at the time of writing and in the discussed context. The trained models would most certainly be part of a bigger piece of software, deployed and encapsulated within other, functional code. The machine learning model itself could be easily credited as a trade secret of a software company, thus taking discovery of the algorithm to another level of court action.

Training AI directly on a particular artist could lead to various legal issues. At the time of writing there is no clear answer to the question if purchasing a song is synonymous with obtaining the right to use the audio as training data. It is therefore not clear whether artificial intelligence can be legally trained on copyright protected music. The engineer in charge of the training process could violate a copyright owner's rights to create derivative works based upon the original material, if the AI is trained implicitly to sound like a particular artist.

Taking this issue even further, sampling has been used as a creative technique for years, becoming a staple of genres like hip-hop and electronic music. Sampling has its own set of legal and ethical issues. In some cases, samples are cleared under fair use laws, which grant limited use of material without permission. In other cases, artists must acquire permission from the copyright holders. Some machine learning algorithms process only short, chunks of audio, which in some cases also could fall under sampling laws, thus further adding to the vagueness of the issue.

In terms of Polish law, at the time of writing, no legislature on the issue of copyrights for music created by artificial intelligence has been passed. No official statement on the usage of such solutions has been issued neither the Association of Stage Writers and Composers (Związek Autorów i Kompozytorów Scenicznych ZAIKS) nor the Association of Performing Artists (Związek Artystów Wykonawców STOART), both of which the author is associated with, although a few public discussion panels on the increasing usage of creative artificial intelligence have been organized by ZAIKS.

The models for creativity augmentation proposed in this thesis, when deployed, would be an encapsulated part of a musical tool, for instance a VST effect plugin, software synthesizer or a bigger system for creativity augmentation. Considering the aforementioned cases, it is safe to say that at the time of writing, the music enhanced with the aid of the proposed solutions would most certainly maintain full copyright for the creator - user of the end-product software.

Chapter 7

Bibliography

[Allen and Rabiner, 1977] Allen, J. B. and Rabiner, L. R. (1977). A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564.

[Amper, 2014] Amper (2014). Amper music - create the right sound, define your narrative, drive emotion with amper ai. https://www.ampermusic.com, access September 2021.

[Bahuleyan, 2018] Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*.

[Balazs et al., 2011] Balazs, P., Dörfler, M., Jaillet, F., Holighaus, N., and Velasco, G. (2011). Theory, implementation and applications of nonstationary gabor frames. *Journal of Computational and Applied Mathematics*, 236(6):1481 – 1496.

[Baydin et al., 2018] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18.

[Bell, 2011] Bell, C. (2011). Algorithmic music composition using dynamic markov chains and genetic algorithms. *Journal of Computing Sciences in Colleges*, 27(2):99–107.

[Bellini and Nesi, 2001] Bellini, P. and Nesi, P. (2001). Wedelmusic format: An xml music notation format for emerging applications. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, pages 79–86. IEEE.

[Benward, 2009] Benward, B. (2009). *Ear training*. McGraw-Hill Higher Education.

[Benward, 2014] Benward, B. (2014). *Music in Theory and Practice Volume 1*. McGraw-Hill Higher Education.

[Bertin-Mahieux et al., 2011] Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 591—-596.

[Biles et al., 1994] Biles, J. et al. (1994). Genjam: A genetic algorithm for generating jazz solos. In *ICMC*, volume 94, pages 131–137. Ann Arbor, MI.

[Bitton et al., 2018] Bitton, A., Esling, P., and Chemla-Romeu-Santos, A. (2018). Modulated variational auto-encoders for many-to-many musical timbre transfer. *CoRR*, abs/1810.00222.

[Boomy, 2021] Boomy (2021). Make instant music, release to billions, create original songs in seconds. https://boomy.com, access September 2021.

[Boulanger-Lewandowski et al., 2012] Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*.

[Brock et al., 2021] Brock, A., De, S., Smith, S. L., and Simonyan, K. (2021). High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*.

[Brooks et al., 1957] Brooks, F. P., Hopkins, A., Neumann, P. G., and Wright, W. V. (1957). An experiment in musical composition. *IRE Transactions on Electronic Computers*, (3):175–182.

[Brown, 1991] Brown, J. C. (1991). Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[Brunner et al., 2018] Brunner, G., Konrad, A., Wang, Y., and Wattenhofer, R. (2018). Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer.

[Burgess, 2014] Burgess, R. J. (2014). *The history of music production*. Oxford University Press.

[Caetano et al., 2010] Caetano, M. F., Burred, J. J., and Rodet, X. (2010). Automatic segmentation of the temporal evolution of isolated acoustic musical instrument sounds using spectro-temporal cues. In *International Conference on Digital Audio Effects (DAFx-10)*, pages 11–21.

[Casey et al., 2008] Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. (2008). Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696.

[Chanan, 1995] Chanan, M. (1995). *Repeated takes: A short history of recording and its effects on music*. Verso.

[Child et al., 2019] Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

[Childers et al., 1977] Childers, D. G., Skinner, D. P., and Kemerait, R. C. (1977). The cepstrum: A guide to processing. *Proceedings of the IEEE*, 65(10):1428–1443.

[Chiliguano and Fazekas, 2016] Chiliguano, P. and Fazekas, G. (2016). Hybrid music recommender using content-based and social information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2618–2622. IEEE.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[Choi et al., 2017a] Choi, K., Fazekas, G., Cho, K., and Sandler, M. (2017a). A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396*.

[Choi et al., 2016] Choi, K., Fazekas, G., and Sandler, M. (2016). Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*.

[Choi et al., 2017b] Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017b). Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE.

[Choi et al., 2015] Choi, K., Fazekas, G., Sandler, M., and Kim, J. (2015). Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, pages 26–30.

[Chuan and Herremans, 2018] Chuan, C.-H. and Herremans, D. (2018). Modeling temporal tonal relations in polyphonic music through deep networks with a novel image-based representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Ciresan et al., 2011] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*.

[Cope, 2000] Cope, D. (2000). *The algorithmic composer*, volume 16. AR Editions, Inc.

[Copyright-Office, 1966] Copyright-Office (1966). Sixty-eighth annual report of the register of copyrights for the fiscal year ending june 30, 1965. *The Library Of Congress*.

[Cox, 1972] Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

[Cramer et al., 2019] Cramer, J., Wu, H.-H., Salamon, J., and Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856. IEEE.

[Dahlhaus et al., 1983] Dahlhaus, C. et al. (1983). *Foundations of music history*. Cambridge University Press.

[Dai et al., 2018] Dai, S., Zhang, Z., and Xia, G. G. (2018). Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*.

[Defferrard, 2015] Defferrard, M. (2015). Structured auto-encoder with application to music genre recognition. Technical report.

[Defferrard et al., 2017] Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. (2017). FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*.

[Defferrard et al., 2018] Defferrard, M., Mohanty, S. P., Carroll, S. F., and Salathé, M. (2018). Learning to recognize musical genre from audio. In *The 2018 Web Conference Companion*. ACM Press.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Dhariwal et al., 2020] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. (2020). Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

[Ding et al., 2016] Ding, W., Xu, M., Huang, D., Lin, W., Dong, M., Yu, X., and Li, H. (2016). Audio and face video emotion recognition in the wild using deep neural networks and small datasets. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 506–513.

[Dixon, 2006] Dixon, S. (2006). Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects*, volume 120, pages 133–137. Citeseer.

[Dong et al., 2018] Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., and Yang, Y.-H. (2018). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Dong and Yang, 2018] Dong, H.-W. and Yang, Y.-H. (2018). Convolutional generative adversarial networks with binary neurons for polyphonic music generation. *arXiv preprint arXiv:1804.09399*.

[Dosovitskiy and Brox, 2016] Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29:658–666.

[Duan et al., 2019] Duan, Y., Edwards, J. S., and Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. *International Journal of Information Management*, 48:63–71.

[Dunbar, 2012] Dunbar, R. I. (2012). On the evolutionary function of song and dance. *Music, language, and human evolution*, pages 201–14.

[Eck and Schmidhuber, 2002] Eck, D. and Schmidhuber, J. (2002). Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In *Proceedings of the 12th IEEE workshop on neural networks for signal processing*, pages 747–756. IEEE.

[Ellis, 2007a] Ellis, D. (Columbia University, 2007a). *Chroma feature analysis and synthesis*.

[Ellis, 2007b] Ellis, D. P. (2007b). Classifying music audio with timbral and chroma features.

[Ellis and Poliner, 2007] Ellis, D. P. and Poliner, G. E. (2007). Identifyingcover songs' with chroma features and dynamic programming beat tracking. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1429. IEEE.

[Endel, 2021] Endel (2021). Personalized soundscapes to help you focus, relax and sleep. backed by neuroscience. https://endel.io, access September 2021.

[Engel et al., 2017] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., and Norouzi, M. (2017). Neural audio synthesis of musical notes with wavenet autoencoders. In *ICML*.

[Erhan et al., 2009] Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1.

[Eriksson, 2016] Eriksson, M. (2016). Close reading big data: The echo nest and the production of (rotten) music metadata. *First Monday*.

[Euler, 1739] Euler, L. (1739). *Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae*. Ex typographia Academiae scientiarum.

[Eyben, 2015] Eyben, F. (2015). *Real-time speech and music classification by large audio feature space extraction*. Springer.

[Ezen-Can, 2020] Ezen-Can, A. (2020). A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.

[Felix Weninger, 2012] Felix Weninger, B. S. (2012). *Music Information Retrieval: An Inspirational Guide to Transfer from Related Disciplines*.

[Feng et al., 2017] Feng, L., Liu, S., and Yao, J. (2017). Music genre classification with paralleling recurrent convolutional neural network. *arXiv preprint arXiv:1712.08370*.

[Fernández and Vico, 2013] Fernández, J. D. and Vico, F. (2013). Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582.

[Fox, 2006] Fox, C. (2006). Genetic hierarchical music structures. In *FLAIRS Conference*, pages 243–247.

[Ganchev et al., 2005] Ganchev, T., Fakotakis, N., and Kokkinakis, G. (2005). Comparative evaluation of various mfcc implementations on the speaker verification task. In *Proceedings of the SPECOM*, volume 1, pages 191–194.

[García Salas et al., 2011] García Salas, H. A., Gelbukh, A., Calvo, H., and Galindo Soria, F. (2011). Automatic music composition with simple probabilistic generative grammars. *Polibits*, (44):59–65.

[Gatys et al., 2015a] Gatys, L., Ecker, A. S., and Bethge, M. (2015a). Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28:262–270.

[Gatys et al., 2015b] Gatys, L. A., Ecker, A. S., and Bethge, M. (2015b). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

[Gatys et al., 2016] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

[Gemmeke et al., 2017] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE.

[Gilbert and Pearson, 2002] Gilbert, J. and Pearson, E. (2002). *Discographies: Dance, Music, Culture and the Politics of Sound*. Routledge.

[Good, 2001] Good, M. (2001). Musicxml for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12(113-124):160.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[Gould, 2016] Gould, E. (2016). *Behind bars: the definitive guide to music notation*. Faber Music Ltd.

[Grindlay and Helmbold, 2006] Grindlay, G. and Helmbold, D. (2006). Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine learning*, 65(2):361–387.

[Gu and Raphael, 2012] Gu, Y. and Raphael, C. (2012). Modeling piano interpretation using switching kalman filter. In *ISMIR*, pages 145–150.

[Harris, 1978] Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83.

[Harte et al., 2006] Harte, C., Sandler, M., and Gasser, M. (2006). Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 21–26.

[Hawthorne et al., 2017] Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S., and Eck, D. (2017). Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*.

[Hawthorne et al., 2018] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., and Eck, D. (2018). Enabling factorized piano music modeling and generation with the maestro dataset. *arXiv preprint arXiv:1810.12247*.

[Herremans et al., 2015] Herremans, D., Weisser, S., Sörensen, K., and Conklin, D. (2015). Generating structured music for bagana using quality metrics based on markov models. *Expert Systems with Applications*, 42(21):7424–7435.

[Hiller and Isaacson, 1979] Hiller, L. A. and Isaacson, L. M. (1979). *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hoos et al., 2001] Hoos, H. H., Renz, K., and Görg, M. (2001). Guido/mir-an experimental musical information retrieval system based on guido music notation. In *ISMIR*, pages 41–50. Citeseer.

[Huang et al., 2018] Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. (2018). Music transformer. *arXiv preprint arXiv:1809.04281*.

[Huang et al., 2019] Huang, S., Li, Q., Anil, C., Bao, X., Oore, S., and Grosse, R. B. (2019). Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer. In *International Conference on Learning Representations*.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[Ji et al., 2020] Ji, S., Luo, J., and Yang, X. (2020). A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*.

[Jia et al., 2019] Jia, Z., Tillman, B., Maggioni, M., and Scarpazza, D. P. (2019). Dissecting the graphcore ipu architecture via microbenchmarking. *arXiv preprint arXiv:1912.03413*.

[Jolicoeur-Martineau, 2018] Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard gan.

[Jouppi et al., 2018] Jouppi, N., Young, C., Patil, N., and Patterson, D. (2018). Motivation for and evaluation of the first tensor processing unit. *IEEE Micro*, 38(3):10–19.

[Jouppi et al., 2017] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12.

[Kallen, 2013] Kallen, S. A. (2013). *The history of classical music*. Greenhaven Publishing LLC.

[Kaneko and Kameoka, 2018] Kaneko, T. and Kameoka, H. (2018). Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104. IEEE.

[Kaneko et al., 2019] Kaneko, T., Kameoka, H., Tanaka, K., and Hojo, N. (2019). Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824. IEEE.

[Karpathy, 2019] Karpathy, A. (2019). *A Recipe for Training Neural Networks, `http://karpathy.github.io/2019/04/25/recipe/`, access September 2021.*

[Kim et al., 2018] Kim, J., Won, M., Serra, X., and Liem, C. C. (2018). Transfer learning of artist group factors to musical genre classification. In *Companion Proceedings of the The Web Conference 2018*, pages 1929–1934.

[Kim, 2016] Kim, J.-S. (2016). *DeepJazz, `https://deepjazz.io`, access September 2021.*

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Knees and Schedl, 2016] Knees, P. and Schedl, M. (2016). *Music similarity and retrieval: an introduction to audio-and web-based strategies*, volume 9. Springer.

[Koh and Dubnov, 2021] Koh, E. and Dubnov, S. (2021). Comparison and analysis of deep audio embeddings for music emotion recognition. *arXiv preprint arXiv:2104.06517*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

[Laitz, 2008] Laitz, S. G. (2008). *The complete musician: An integrated approach to tonal theory, analysis, and listening*, volume 1. Oxford University Press, USA.

[Lavner and Ruinskiy, 2009] Lavner, Y. and Ruinskiy, D. (2009). A decision-tree-based algorithm for speech/music classification and segmentation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:1–14.

[Lavrenko and Pickens, 2003] Lavrenko, V. and Pickens, J. (2003). Polyphonic music modeling with random fields. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 120–129.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lehrman, 2020] Lehrman, P. D. (2020). Midi 2.0: Promises and challenges. In *Music Encoding Conference 2020*.

[Lerch, 2012] Lerch, A. (2012). *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press.

[Levine, 2011] Levine, M. (2011). *The jazz theory book*. " O'Reilly Media, Inc.".

[Li et al., 2001] Li, D., Sethi, I. K., Dimitrova, N., and McGee, T. (2001). Classification of general audio data for content-based retrieval. *Pattern recognition letters*, 22(5):533–544.

[Li et al., 2019] Li, T., Choi, M., Fu, K., and Lin, L. (2019). Music sequence prediction with mixture hidden markov models. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 6128–6132. IEEE.

[Li et al., 2003] Li, T., Ogihara, M., and Li, Q. (2003). A comparative study on content-based music genre classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289.

[London, 2012] London, J. (2012). *Hearing in time: Psychological aspects of musical meter*. Oxford University Press.

[Lopez-Rincon et al., 2018] Lopez-Rincon, O., Starostenko, O., and Ayala-San Martín, G. (2018). Algoritmic music composition based on artificial intelligence: A survey. In *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 187–193. IEEE.

[Lu et al., 2018] Lu, C.-Y., Xue, M.-X., Chang, C.-C., Lee, C.-R., and Su, L. (2018). Play as you like: Timbre-enhanced multi-modal music style transfer.

[Makhmutov et al., 2020] Makhmutov, M., Varouqa, S., and Brow, J. A. (2020). Survey on copyright laws about music generated by artificial intelligence. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 3003–3009. IEEE.

[Mandel and Ellis, 2005] Mandel, M. I. and Ellis, D. P. (2005). Song-level features and support vector machines for music classification.

[Marafioti et al., 2020] Marafioti, A., Majdak, P., Holighaus, N., and Perraudin, N. (2020). Gacela: A generative adversarial context encoder for long audio inpainting of music. *IEEE Journal of Selected Topics in Signal Processing*, 15(1):120–131.

[Mark and Gary, 2007] Mark, M. and Gary, C. L. (2007). *A history of American music education*. ERIC.

[Masters and Luschi, 2018] Masters, D. and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.

[McKay and Fujinaga, 2006] McKay, C. and Fujinaga, I. (2006). Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106. Citeseer.

[McKenzie, 2012] McKenzie, D. (2012). *Jazz Piano MIDI Dataset,* `bushgrafts.com/midi/`*, access October 2021.*

[McKinney and Breebaart, 2003] McKinney, M. and Breebaart, J. (2003). Features for audio and music classification. *Proceedings of the 4th International Conference on Music Information Retrieval*.

[MIDIAssociation, 1999] MIDIAssociation (1999). *General MIDI standard specification, https://www.midi.org/specifications, access September 2021.*

[MIDIAssociation, 2021] MIDIAssociation (2021). *MIDI 2.0 Specification, https://www.midi.org/midi/specifications/midi-2-0-specifications, access September 2021.*

[midi_man, 2019] midi_man, U. (2019). The largest midi collection on the internet, collected and sorted diligently by yours truly. https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/.

[Milczarek, 2021] Milczarek, M. (2021). *DROP, https://bit.ly/3G4rkXj, access October 2021.*

[Mital, 2017] Mital, P. K. (2017). Time domain neural audio style transfer. *arXiv preprint arXiv:1711.11160.*

[Modrzejewski et al., 2021] Modrzejewski, M., Bereda, K., and Rokita, P. (2021). Efficient recurrent neural network architecture for musical style transfer. In *Artificial Intelligence and Soft Computing*, pages 124–132. Springer International Publishing.

[Modrzejewski et al., 2019] Modrzejewski, M., Dorobek, M., and Rokita, P. (2019). Application of deep neural networks to music composition based on midi datasets and graphical representation. In *Artificial Intelligence and Soft Computing*, pages 143–152. Springer International Publishing.

[Modrzejewski and Rokita, 2018a] Modrzejewski, M. and Rokita, P. (2018a). Critical analysis of conversational agent technology for intelligent customer support and proposition of a new solution. In *Artificial Intelligence and Soft Computing*, pages 723–733. Springer International Publishing.

[Modrzejewski and Rokita, 2018b] Modrzejewski, M. and Rokita, P. (2018b). Graphical interface design for chatbots for the needs of artificial intelligence support in web and mobile applications. In *Computer Vision and Graphics*, pages 48–56. Springer International Publishing.

[Modrzejewski and Rokita, 2019] Modrzejewski, M. and Rokita, P. (2019). Implementation of generic steering algorithms for ai agents in computer games. In *Intelligent Methods and Big Data in Industrial Applications*, pages 15–27. Springer International Publishing.

[Modrzejewski et al., 2020] Modrzejewski, M., Szachewicz, J., and Rokita, P. (2020). Application of neural networks and graphical representations for musical genre classification. In *International Conference on Artificial Intelligence and Soft Computing*, pages 193–202. Springer.

[Moelants, 2002] Moelants, D. (2002). Preferred tempo reconsidered. In *Proceedings of the 7th international conference on music perception and cognition*, volume 2002, pages 1–4. Citeseer.

[Monro, 1894] Monro, D. B. (1894). *The modes of ancient Greek music*. Clarendon Press.

[Moog, 1986] Moog, R. A. (1986). Midi: Musical instrument digital interface. *Journal of the Audio Engineering Society*, 34(5):394–404.

[Mor et al., 2018] Mor, N., Wolf, L., Polyak, A., and Taigman, Y. (2018). A universal music translation network. *CoRR*, abs/1805.07848.

[Mordvintsev et al., 2015] Mordvintsev, A., Olah, C., and Tyka, M. (2015). Inceptionism: Going deeper into neural networks.

[Mulongo, 2020] Mulongo, B. (2020). Analyzing music genre classification using item response theory: A case study of the gtzan data.

[Murauer and Specht, 2018] Murauer, B. and Specht, G. (2018). Detecting music genre using extreme gradient boosting. In *Companion proceedings of the the web conference 2018*, pages 1923–1927.

[Nicholls, 1998] Nicholls, D. (1998). *The Cambridge history of American music*. Cambridge University Press.

[Park et al., 2017] Park, J., Lee, J., Park, J., Ha, J.-W., and Nam, J. (2017). Representation learning of music using artist labels. *arXiv preprint arXiv:1710.06648*.

[Pasquier et al., 2017] Pasquier, P., Eigenfeldt, A., Bown, O., and Dubnov, S. (2017). An introduction to musical metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14.

[Payne and OpenAI, 2019] Payne, C. and OpenAI (2019). *MuseNet* openai.com/blog/musenet.

[Pearce et al., 2002] Pearce, M., Meredith, D., and Wiggins, G. (2002). Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, 6(2):119–147.

[Peeters et al., 2011] Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., and McAdams, S. (2011). The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916.

[Perez et al., 2018] Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Pinkerton, 1956] Pinkerton, R. C. (1956). Information theory and melody. *Scientific American*, 194(2):77–87.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[Raffel, 2016] Raffel, C. (2016). *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. PhD thesis, Columbia University.

[Roberts et al., 2019] Roberts, A., Engel, J., Raffel, C., Hawthorne, C., and Eck, D. (2019). A hierarchical latent vector model for learning long-term structure in music.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[Russo, 1997] Russo, W. (1997). *Jazz composition and orchestration*. University of Chicago Press.

[Rydning et al., 2018] Rydning, J., Reinsel, D., and Gantz, J. (2018). The digitization of the world from edge to core. *Framingham: International Data Corporation*, page 16.

[Sabatella, 2021] Sabatella, M. (2021). *MusceScore, https://musescore.com/marcsabatella, access September 2021.*

[Scaringella et al., 2006] Scaringella, N., Zoia, G., and Mlynek, D. (2006). Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141.

[Schedl et al., 2014] Schedl, M., Gómez Gutiérrez, E., and Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval. 2014 Sept 12; 8 (2-3): 127-261.*

[Schlüter and Böck, 2013] Schlüter, J. and Böck, S. (2013). Musical onset detection with convolutional neural networks. In *6th international workshop on machine learning and music (MML), Prague, Czech Republic*. sn.

[Schlüter and Böck, 2014] Schlüter, J. and Böck, S. (2014). Improved musical onset detection with convolutional neural networks. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 6979–6983. IEEE.

[Schörkhuber and Klapuri, 2010] Schörkhuber, C. and Klapuri, A. (2010). Constant-q transform toolbox for music processing. In *7th sound and music computing conference, Barcelona, Spain*, pages 3–64.

[Schubart, 1839] Schubart, C. F. D. (1839). *Gesammelte Schriften und Schicksale: Ideen zu einer Aesthetik der Tonkunst*, volume 5. J. Scheible.

[Shepherd, 2003] Shepherd, J. (2003). *Continuum Encyclopedia of Popular Music of the World: Performance and production. Volume II*, volume 1. A&C Black.

[Shepherd et al., 2003] Shepherd, J., Horn, D., Laing, D., Oliver, P., and Wicke, P. (2003). *Continuum Encyclopedia of Popular Music of the World, Volume 1: Media, Industry, Society*, volume 1. A&C Black.

[Sigtia and Dixon, 2014] Sigtia, S. and Dixon, S. (2014). Improved music feature learning with deep neural networks. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6959–6963. IEEE.

[Simon and Oore, 2017] Simon, I. and Oore, S. (2017). Performance rnn: Generating music with expressive timing and dynamics. https://magenta.tensorflow.org/performance-rnn.

[Simonyan et al., 2014] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Smith and Wood, 1981] Smith, D. and Wood, C. (1981). The'usi', or universal synthesizer interface. In *Audio Engineering Society Convention 70*. Audio Engineering Society.

[Sturm, 2012a] Sturm, B. (2012a). An analysis of the gtzanmusic genre dataset. In *"Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies (Vol. 2012, pp. 7- 12). Association for Computing Machinery. ACM Multimedia"*.

[Sturm, 2012b] Sturm, B. (2012b). A survey of evaluation in music genre recognition.

[Sturm, 2013a] Sturm, B. (arXiv preprint, arXiv:1306.1461, 2013a). “the gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use".

[Sturm, 2013b] Sturm, B. L. (2013b). The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*.

[Sturm et al., 2019] Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., Hadjeres, G., Deruty, E., and Pachet, F. (2019). Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, 48(1):36–55.

[Sturm et al., 2016] Sturm, B. L., Santos, J. F., Ben-Tal, O., and Korshunova, I. (2016). Music transcription modelling and composition using deep learning. *arXiv preprint arXiv:1604.08723*.

[Tan et al., 2020] Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.

[tarepan, 2018] tarepan (2018). *Rainbowgram exctraction Python package, https://github.com/tarepan/rainbowgram, access September 2021.*

[Todd, 1989] Todd, P. M. (1989). A connectionist approach to algorithmic composition. *Computer Music Journal*, 13(4):27–43.

[Tzanetakis and Cook, 2002] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293 – 302.

[Valerio et al., 2018] Valerio, V. D., Pereira, R. M., Costa, Y. M., Bertoini, D., and Silla Jr, C. N. (2018). A resampling approach for imbalanceness on music genre classification using spectrograms. In *The Thirty-First International Flairs Conference*.

[van den Oord et al., 2016] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499.

[van den Oord et al., 2017] van den Oord, A., Vinyals, O., and kavukcuoglu, k. (2017). Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Van Der Merwe and Schulze, 2010] Van Der Merwe, A. and Schulze, W. (2010). Music generation with markov models. *IEEE MultiMedia*, 18(3):78–85.

[vd Boogaart and Lienhart, 2009] vd Boogaart, C. G. and Lienhart, R. (2009). Note onset detection for the transcription of polyphonic piano music. In *2009 IEEE International Conference on Multimedia and Expo*, pages 446–449. IEEE.

[Verma and Smith, 2018] Verma, P. and Smith, J. O. (2018). Neural style transfer for audio spectograms.

[Virtanen et al., 2018] Virtanen, T., Plumbley, M. D., and Ellis, D. (2018). *Computational analysis of sound scenes and events*. Springer.

[Walshaw, 2021] Walshaw, C. (1995-2021). *ABC Notation, `https://abcnotation.com`, access September 2021.*

[Wengert, 1964] Wengert, R. E. (1964). A simple automatic derivative evaluation program. *Communications of the ACM*, 7(8):463–464.

[White, 1994] White, J. D. (1994). *Comprehensive musical analysis*. Scarecrow Press.

[Winnington-Ingram, 2015] Winnington-Ingram, R. P. (2015). *Mode in ancient Greek music*. Cambridge University Press.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[Wülfing and Riedmiller, 2012] Wülfing, J. and Riedmiller, M. A. (2012). Unsupervised learning of local features for music classification. In *ISMIR*, pages 139–144.

[Xu et al., 2021] Xu, J., Pan, Y., Pan, X., Hoi, S., Yi, Z., and Xu, Z. (2021). Regnet: Self-regulated network for image classification. *arXiv preprint arXiv:2101.00590*.

[Yang et al., 2020] Yang, R., Feng, L., Wang, H., Yao, J., and Luo, S. (2020). Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices. *IEEE Access*, 8:19629–19637.

[Zwicker and Fastl, 2013] Zwicker, E. and Fastl, H. (2013). *Psychoacoustics: Facts and models*, volume 22. Springer Science & Business Media.

List of Figures

List of Tables