



Eine Analyse der Relevanz und Anwendbarkeit  
aktueller Künstlicher Intelligenz in der  
Generierung von Audiosamples für Dubstep  
Produzenten

Fakultät Electronic Media

Audiovisuelle Medien

Hochschule der Medien

**Bachelorarbeit**

Vorgelegt von

**Björn Pietschke**

geboren 05.12.1993 in Ulm

Im Februar 2024

Erstprüfer: Prof. Oliver Curdt

Zweitprüfer: Prof. Dr. Andreas Koch

## Zusammenfassung

Diese Arbeit untersucht die Anwendung von Künstlicher Intelligenz in der Musikproduktion. Insbesondere wird die Generierung von Audiosamples untersucht, um festzustellen, inwieweit KI-Tools bereits jetzt modernen Dubstep Produzenten nützlich sein können. Audiosamples und deren Entstehen sind ein wichtiger Bestandteil von der Musikproduktion. Diese Aufgabenstellung, übernommen von KI, könnte erhebliche Zeiteinsparnisse in der modernen Musikproduktion bedeuten, speziell im Electronic Dance Music Genre: Dubstep.

Es werden zwei aktuelle KI-Modelle analysiert hinsichtlich ihrer technischen Leistungsfähigkeit und Benutzerfreundlichkeit. Dance Diffusion von HarmonAI und AudioGen von Meta. Diese Arbeit zeigt, dass diese Modelle zwar innovative Ansätze bieten, jedoch in Bezug auf Benutzerfreundlichkeit und Audioqualität eingeschränkt sind. Die Ergebnisse zeigen, dass sich KI-basierte Audiogenerierungstools noch in einem eher experimentellen Stadium befinden und daher momentan noch nicht nützlich sind. Sie müssen noch weiterentwickelt werden, um den Anforderungen von Musikproduzenten gerecht zu werden.

Diese Arbeit bietet einen Ausblick auf zukünftige Entwicklungen und zeigt, dass weitere Forschung und Entwicklung zwar nötig ist, aber die Landschaft der Sachen Musikproduktion grundlegend verändern kann.

## **Abstract**

This thesis explores the application of Artificial Intelligence in music production, focusing particularly on the generation of audio samples to determine to what extent AI tools are currently useful for modern dubstep producers. Audio samples and their creation are a crucial part of music production. The task of generating these samples using AI could mean significant time savings in modern music production, especially in the Electronic Dance Music genre: Dubstep.

Two current AI models, Dance Diffusion by HarmonAI and AudioGen by Meta, have been analyzed for their technical performance and user-friendliness. This study reveals that while these models offer innovative approaches, they are limited in terms of user-friendliness and audio quality. The results indicate that AI-based audio generation tools are still in a rather experimental stage and are not yet useful. Further development is necessary to meet the requirements of music producers.

This work provides an outlook on future developments and indicates that further research and development are necessary, but could fundamentally change the landscape of music production

## Ehrenwörtliche Erklärung

„Hiermit versichere ich, Björn Pietschke, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

„Eine Analyse der Relevanz und Anwendbarkeit aktueller Künstlicher Intelligenz in der Generierung von Audiosamples für Dubstep Produzenten“,

selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“

Datum: 25.02.2024 Unterschrift: B. Pietschke

## **Abkürzungsverzeichnis**

AI Artificial Intelligence

KI Künstliche Intelligenz

# Inhaltsverzeichnis

<b>Zusammenfassung .....</b>	<b>I</b>
<b>Abstract.....</b>	<b>II</b>
<b>Ehrenwörtliche Erklärung.....</b>	<b>III</b>
<b>Abkürzungsverzeichnis .....</b>	<b>IV</b>
<b>Inhaltsverzeichnis .....</b>	<b>V</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Problemstellung und Zielsetzung .....	1
1.2 Vorgehensweise.....	2
<b>2 Grundlagen .....</b>	<b>3</b>
2.1 Definitionen.....	3
2.1.1 Was ist Künstliche Intelligenz? .....	3
2.1.2 Was ist Musikproduktion .....	4
2.2 Was ist Electronic Dance Music, speziell: Dubstep? .....	5
2.2.1 Growls .....	6
2.2.2 Dubstep Snares .....	7
2.3 Musikproduktion .....	7
2.3.1 Grundlagen Musikproduktion .....	7
2.3.2 Relevante Begriffe.....	8
2.3.3 Stand der Technik Künstliche Intelligenz .....	11
<b>3 Audiogenerierung durch Künstliche Intelligenz .....</b>	<b>17</b>
3.1 Symbolische Audiogenerierung .....	19
3.1.1 Markov Ketten.....	19
3.1.2 Evolutionäre Algorithmen.....	20
3.1.3 Feedforward Netzwerke .....	20
3.1.4 Recurrent Neural Networks.....	21
3.1.5 Generative adversarial networks .....	21
3.1.6 Variational autoencoders .....	22
3.1.7 Transformer Netzwerke.....	22
3.1.8 Sprache und Künstliche Intelligenz.....	23
3.2 Nicht Symbolische Audiogenerierung .....	24
3.2.1 WaveNet.....	26
3.2.2 GANSynth .....	26
3.2.3 SampleRNN.....	27

---

<b>4</b>	<b>Generative AI und Anwendungen in der Musikproduktion .....</b>	<b>30</b>
4.1	Dance Diffusion von HarmonAI .....	30
4.1.1	Fazit .....	36
4.2	Audiocraft - AUDIOGEN .....	37
4.2.1	Prompts .....	42
4.2.2	Bedingungen .....	42
4.2.3	Untersuchung Snare .....	43
4.2.4	Untersuchung Grownl .....	48
4.2.5	Fazit zu AudioGen .....	49
<b>5</b>	<b>Schlussbetrachtung .....</b>	<b>52</b>
5.1	Zusammenfassung und Fazit .....	52
5.2	Kritische Bewertung des Vorgehens und der Ergebnisse .....	53
5.3	Ausblick .....	53
	<b>Abbildungsverzeichnis .....</b>	<b>55</b>
	<b>Literaturverzeichnis .....</b>	<b>56</b>

# 1 Einleitung

Künstliche Intelligenz eröffnet neue Horizonte in sehr vielen Bereichen, so ist es bereits möglich, Text, Video, Sprache und Musik kreativ zu generieren. Mit der rasanten Entwicklung von Künstlicher Intelligenz stellt sich die Frage, inwiefern Tools effektiv und zugänglich in verschiedenen Tätigkeitsfeldern genutzt werden können, um dem Menschen zu helfen. Insbesondere hat Künstliche Intelligenz das Potential, die moderne Musikproduktion, speziell in Richtung Electronic Dance Music und Dubstep, grundlegend zu prägen. Diese Arbeit untersucht die Rolle von KI bei der Generierung von Audiosamples, einem Schlüsselaspekt der modernen Musikproduktion. In einer Branche, welche sich ständig weiterentwickelt, könnten KI-basierte Sample-Generierungs-Tools ein wichtiger Faktor für die Musikproduktion sein. Diese Arbeit erkundet die bereits zum aktuellen Stand der Forschung nützlichen Möglichkeiten und Grenzen dieser Technologie und bewertet, inwieweit sie die Arbeit von Musikproduzenten beeinflussen.

## 1.1 Problemstellung und Zielsetzung

Moderne Musikproduktion steht durch den Einsatz von Künstlicher Intelligenz vor einer technologischen Revolution. Ein wichtiger Bestandteil der elektronischen Musikproduktion ist es die richtigen Audiosamples für den Song zu finden oder mit Hilfe von Plugins, Aufnahmen und Synthesizern selbst zu erstellen. Diese Aufgabe nimmt oft sehr viel Zeit und Mühe in Anspruch.

Das Ziel dieser Arbeit ist es, herauszufinden ob KI-Tools bereits dazu in der Lage sind, dem Musikproduzenten passende Audiodateien zur Weiterverwertung individuell, kreativ und nützlich zu generieren.

Künstliche Intelligenz in der Musikproduktion ist ein aufstrebendes Feld, das die Art und Weise wie Musik produziert wird, grundlegend verändern kann. Dieses Feld ist in viele verschiedene Bereiche unterteilt, wie MIDI-Generierung, Sprachgenerierung, automatisiertes Mastering, Audioanalyse und viele mehr. Es gilt allerdings in dieser Arbeit herauszufinden wie weit die Forschung in Sachen Audiogenerierung ist und inwiefern Audiogenerierung-Tools, bereits jetzt, nützlich sind.



## 1.2 Vorgehensweise

In dieser Arbeit wird eine systematische Analyse der Einsatzmöglichkeiten von Künstlicher Intelligenz in der Musikproduktion, speziell in der Generierung von Audiosamples, durchgeführt. Der Fokus liegt hier auf Samplegeneration für die Musikrichtung Dubstep. Genauer werden die Generierungen von zwei essentiellen Bestandteilen von Dubstep. Growls und Snares.

Zunächst wird ein Überblick über relevante Literatur und Forschung geboten. Dieser bezieht sich auf Modelle von symbolischer und nicht symbolischer Audiogeneration. Symbolische Audiogeneration bezieht sich auf die Generation von MIDI-Files, während die Nicht-Symbolische Audiogeneration auf Audiogenerierung bezieht.

Anschließend erfolgt eine detaillierte Untersuchung der Ausgewählten Modelle Dance Diffusion von HarmonAI und AudioGen von Meta. Dabei werden die technischen Fähigkeiten und die Benutzerfreundlichkeit analysiert. Die Untersuchung besteht aus praktischen Tests und Durchläufen der beiden Modelle. Anschließend werden die Ergebnisse diskutiert. Hierbei wird darauf geachtet, wie viel Zeit aufgewendet werden muss, um die Modelle zu nutzen und wie einfach sie zu installieren und benutzen sind. Es wird geprüft, wie kreativ, eigenständig und zuverlässig die Modelle handeln und wie gut die Qualität der Samples ist. Abschließend gibt es einen Ausblick auf zukünftige Entwicklungen.

## 2 Grundlagen

### 2.1 Definitionen

#### 2.1.1 Was ist Künstliche Intelligenz?

Intelligenz im Allgemeinen setzt sich aus mehreren Faktoren zusammen. Hierbei spricht man vom multiplen Intelligenzansatz, welcher Intelligenz in verschiedenen Bereichen wie sprachliche Intelligenz, musikalische Intelligenz, intrapersonale und interpersonale Intelligenz, aber auch schöpferische Intelligenz aufteilt.<sup>1</sup>

Da Intelligenz auf einer Vielzahl von Komponenten basiert, gibt es auch kein singuläres Merkmal für hohe oder niedrige Intelligenz. Häufig wird behauptet, ein intelligenter Mensch sei positiv, produktiv in sozialen oder beruflichen Situationen. Begriffe wie „klug“, „begabt“ oder „talentiert“ werden verwendet. Anders als beim Alter, der menschlichen Größe oder Geschlecht gibt es hier mehrere Bestimmungsfaktoren. Bis heute lässt sich für Intelligenz oder gar Künstliche Intelligenz nur schwierig, konkret definieren. Auch wie Intelligenz erworben wird, stellt Forscher vor eine komplizierte Frage.<sup>2</sup>

Aus dem Duden geht folgendes hervor:

„Fähigkeit [des Menschen], abstrakt und vernünftig zu denken und daraus zweckvolles Handeln abzuleiten.“<sup>3</sup>

Intelligenz wird außerdem im Kontext des Menschen verstanden. Wenn Lebensformen mit Ihrer Intelligenz gegenübergestellt werden, so gibt es immer den Vergleich zur kognitiven Funktion des Menschen.

---

<sup>1</sup>Vgl. R.T. Kreuzer und M. Sirrenberg, Künstliche Intelligenz verstehen - Grundlagen – Use-Cases – unternehmenseigene KI-Journey, Wiesbaden: Springer Gabler, 2019,

<https://link.springer.com/book/10.1007/978-3-658-25561-9> (abgerufen am 1. Dezember 2023), S. 2

<sup>2</sup>Vgl. Joachim Funke und Bianca Vaterrodt, Was ist Intelligenz?, 3. Aufl., München: C.H. Beck Verlag, [https://books.google.de/books?id=G\\_H3sl4fVTEC&lpg=PA9&ots=I0g1NIwqoQ&dq=was%20ist%20intelligenz%20pdf&lr&hl=de&pg=PA11#v=onepage&q&f=false..](https://books.google.de/books?id=G_H3sl4fVTEC&lpg=PA9&ots=I0g1NIwqoQ&dq=was%20ist%20intelligenz%20pdf&lr&hl=de&pg=PA11#v=onepage&q&f=false..) (abgerufen am 1. Dezember 2023), S. 9-11.

<sup>3</sup>Dudenredaktion(o.J.), in: Dudenonline, <https://www.duden.de/node/71635/revision/1349018> (abgerufen am: 01.10.2023)

Künstliche Intelligenz kann laut Rich so definiert werden, dass es bei der Erstellung Künstlicher Intelligenz darum geht, dem Computer beizubringen, Aufgaben zu lösen, welche ein Mensch zu diesem Zeitpunkt noch besser kann.<sup>4</sup>

Generell wird Künstliche Intelligenz mit der Fähigkeit kognitive Aufgaben zu lösen verbunden. Hierbei werden Fähigkeiten zur Wahrnehmung, Argumentation oder selbständiges Lernen aufgezählt. Damit zusammenhängend ist die Möglichkeit eigenständig Aufgaben zu lösen. Das Stichwort ist hier Autonomie.<sup>5</sup>

Künstliche Intelligenz kann auch Algorithmen sein, welche Schritt für Schritt, regelbasiert fortschreiten und reagieren, basierend darauf, welche Parameter der Mensch gesetzt hat. Diese Art von Künstlicher Intelligenz nennt man symbolische Künstliche Intelligenz oder regelbasierte Künstliche Intelligenz. Symbolische Künstliche Intelligenz wird vor allem in Umgebungen verwendet, in welchen strikte Regeln vorherrschen, um auf bestimmte Muster und Regeln zu reagieren. Das heißt Algorithmen haben bestimmte Parameter, damit sie ein Problem erkennen, aber auf eine bestimmte Weise darauf reagieren.

Eine weitere Art von Künstlicher Intelligenz ist datengeleitete Künstliche Intelligenz. Dabei steht der Ansatz im Mittelpunkt, Probleme zu lösen, basierend auf einen großen Datensatz. Dieser Datensatz steht zur Ressource zur Verfügung. Der Aufbau datengeleiteter Künstlicher Intelligenz gleich dem Menschlichen Gehirn. Eingangssignale treffen auf künstliche Neuronen welche wiederum ein Ausgangssignal generieren. Die Ausgangssignale werden daraufhin analysiert und verwertet. Je mehr Daten zur Verfügung stehen, desto mehr ist Leistung und Genauigkeit gegeben. Mehr Neuronen bedeuten hier auch mehr Leistungsfähigkeit.<sup>6</sup>

### 2.1.2 Was ist Musik Produktion

Allgemein kann Musik Produktion als kreative und artistische Entwicklung von Musik bezeichnet werden. Der Begriff beschreibt den Vorgang, ein Musikalisches Stück durch mehrere kreative, technische, musikalische Prozesse zu erschaffen.<sup>7</sup>

---

<sup>4</sup>Vgl. Elaine A. Rich: Artificial Intelligence, in: Computers and The humanities, Bd. 19, Nr. 2, 1985, doi:10.1007/bf02259633 (abgerufen am 05.12.2023), S. 117-122.

<sup>5</sup> Vgl. Ralf T. Kreutzer & Marie Sirrenberg, 2019, S. 3.

<sup>6</sup> Vgl. Boucher, Philip: Artificial intelligence: How Does it Work, why Does it Matter, and what Can We Do about It?, in: European Parliamentary Research Service, 2020, S. 1-5.

<sup>7</sup> Vgl. Hepworth-Sawyer, Russ: What is music production?: A Producer's Guide: The Role, the People, the Process, Focal Press, 2017, [https://books.google.de/books?hl=de&lr=&id=UJoC\\_eibCzkC&oi=fnd&pg=PP2&dq=](https://books.google.de/books?hl=de&lr=&id=UJoC_eibCzkC&oi=fnd&pg=PP2&dq=)

Musikproduktion hat technische, aber auch soziale Komponenten. So ist die Rolle des Musikproduzenten technisch, doch sie hat mittlerweile auch eine künstlerische Komponente. Im Laufe der Zeit wandelte sich der Aufgabenbereich, so ist der Produzent nun ebenso verantwortlich, den aufgenommenen Sound dem Künstler anzupassen. Hierbei ist Kreativität gefragt. Dieser als Wertschöpfungsprozess bezeichnete Vorgang ist ein ganzheitlicher. Die Prozesse der Musikproduktion sind über verschiedene Gewerke übergreifend.<sup>8</sup>

Die Rolle des Musikproduzenten ist heute so vielschichtig und facettenreich wie noch nie. Wo früher noch in großen Studios mit Teams, bestehend aus mehreren Personen produziert wurde, gehört Musikproduktion beim Produzenten Zuhause, ohne großem Team, zum Standard. Natürlich sind große Studios mit aufwendigen Produktionen noch immer nicht wegzudenken, doch brauchen Musikproduzenten immer weniger Hardware, was das Produzieren zuhause deutlich erleichtert.<sup>9</sup>

Musikproduzenten sind die Schnittstelle zwischen Klangeigenschaften, Komposition, Arrangement, Performance und Eigenheiten des Musikers und dem klanglichen Ganzen, welches über ein Medium dem Hörer vorgelegt wird.

## 2.2 Was ist Electronic Dance Music, speziell: Dubstep?

Electronic Dance Music (EDM) ist ein Begriff, welcher einige Musikgenres beinhaltet. Er beinhaltet Untergenres wie: Techno, Drum and Bass, Dubstep und Trance. In dieser Arbeit wird hauptsächlich das Subgenre Dubstep beleuchtet. EDM wird hauptsächlich mit elektronischen Geräten produziert, meist mit Synthesizern. Sounds wie Bässe, Schlagzeug oder Leadsynth's werden hauptsächlich mit Synthesizern und in den letzten Jahren häufig nur in der Digital Audio Workstation produziert. EDM fokussiert sich nicht unbedingt auf Gesang, sondern Tanzbarkeit. Die Musik ist drauf ausgelegt, von Disk Jockeys in einem Live Setting gespielt zu werden. Mit dem Wachstum des Internets

---

music+production+pdf&ots=K2xFxRzXpc&sig=9WVkr6Rx7SSs6hOL1sUhVfNrQ0U#v=onepage&q&f=false (abgerufen am: 05.10.23), S. 17.

<sup>8</sup> Vgl. Steinhardt, Sebastian: Musikproduktion der Zukunft: Eine empirische Studie über neue Möglichkeiten für Musiker und Produzenten, Hamburg: Diplomica Verlag GmbH, 2013, [https://books.google.de/books?id=hAEnJYReJXAC&dq=Geschichte+Musikproduktion&lr=&hl=de&source=gbs\\_navlinks\\_s](https://books.google.de/books?id=hAEnJYReJXAC&dq=Geschichte+Musikproduktion&lr=&hl=de&source=gbs_navlinks_s) (abgerufen am: 05.12.2023), S. 1-3

<sup>9</sup> Vgl. Burgess, Richard James: The art of music production: the theory and practice, in: Oxford University Press, <https://books.google.de/books?hl=de&lr=&id=lWEUAAAAQBAJ&oi=fnd&pg=PP1&dq=music+production+pdf&ots=Q4RORacNLM&sig=JhfjaoOtxoXzpSUyhhQ4OHe7Gqk#v=onepage&q=music%20production%20pdf&f=false> (abgerufen am: 05.12.23), S. 2.

begann in der zweiten Hälfte der 2000er eine Welle der Popularität und hat sich Heute als festen Bestandteil der Musikindustrie etabliert. Gerade basslastige Musik wie Dubstep oder Drum and Bass sind sehr populär. Electronic Dance Music Produzenten haben von dem Aufstieg des Internets profitiert. Es wurden Songs online verbreitet, ohne dafür kostspielig in Recording Studios Instrumente aufzunehmen, da EDM und vor allem Dubstep, hauptsächlich in der DAW produziert wird.<sup>10</sup> Dubstep ist ein Genre, das seine Ursprünge im Südlondon der späten 1990er Jahre hat. Es kombinierte Einflüsse aus Drum and bass, 2-Step, Grime und oft Reggae. Charakteristisch für Dubstep sind besonders die schweren Basslinien und die betonten Sub-Bass-Frequenzen. Ein gängiges Merkmal von Dubstep sind die „Growls“ und „Wobble“ Bässe. Heutzutage sind Subgenres wie Brostep, wodurch sich beispielsweise der Künstler Skrillex auszeichnete, Riddim und Melodic Dubstep. Die Klänge sind hier eher aggressiv, erinnern öfter an Metal-E-Gitarren und sind meist durch Soundmanipulation produziert. Dubstep hat sich von einem Undergroundgenre zu einem weit etablierten und beliebten Musikgenre entwickelt. Das Genre ist sehr dynamisch, doch gibt es einige Merkmale, auf die wir uns in dieser Arbeit fokussieren.<sup>11</sup>

### 2.2.1 Growls

Ein Growl wird im Dubstep oft als tiefes, tierisches Geräusch beschrieben, das verwendet wird, um Übergänge zwischen Songabschnitten einzuleiten oder um Aufregung während eines Drops zu erzeugen. Diese Sounds sind hauptsächlich stark verzerrte Bässe, haben kratzige Höhen und im Kontrast dazu, tiefe Subfrequenzen. Um diese Sounds herzustellen, werden meist diverse Plugins und Techniken verwendet. Ein Beispiel ist Frequenzmodulationssynthese. Mehrere Oszillatoren und ein separater Motor werden verwendet, um Harmonien zu den Wellenformen hinzuzufügen oder entfernen. Weiter verzerren Produzenten den Sound durch eine Vielzahl an Techniken und Plugins wie Verzerrung, Kompression oder Samplebearbeitung. Die Entstehung von Growls bedarf viel Zeit und Mühe.<sup>12</sup> Könnte eine Künstliche Intelligenz diese Aufgabe übernehmen, ohne dass dabei die Individualität und Kreativität des Musikproduzenten verloren geht,

---

<sup>10</sup>Vgl. Michaelangelo, Matos: electronic dance music, in: Encyclopedia Britannica, 2023, <https://www.britannica.com/art/electronic-dance-music>, (abgerufen am 01.01.2024).

<sup>11</sup>Vgl. Kerlinger Charlie: A Brief History Of Dubstep: From Its Underground Origins To Worldwide Popularity, in : benvaughn, 2022, <https://www.benvaughn.com/a-brief-history-of-dubstep-from-its-underground-origins-to-worldwide-popularity/>, (abgerufen am 04.01.2024).

<sup>12</sup>Vgl. Kerlinger Charlie: How To Create A Dubstep Growl, benvaughn, 2022, <https://www.benvaughn.com/how-to-create-a-dubstep-growl/> (abgerufen am 04.01.2023)

wäre dies von größtem Vorteil. Voraussetzung hierfür wäre, dass man der Künstlichen Intelligenz einen spezifischen Befehl gäbe und dieser umgesetzt würde.

### 2.2.2 Dubstep Snares

Dubstep Snares haben einige hervorstechende Charakteristika. Es müssen beispielsweise die Transienten klar zu hören sein.<sup>13</sup> Ein Transient ist der erste Spitzenwert eines Audiosignals. Bei der Snare ist es der erste Schlag und damit der erste Ton, welchen wir hören. Dieser Schlag ist meist ein hochfrequenter.<sup>14</sup> Außerdem spielt der sogenannte „body“ eine wichtige Rolle, indem er in der 200-400 Hz Gegend zusammen mit dem Transient klingt, welcher sich wiederum in der 2 kHz Gegend befindet. Auf dem Boden der Snare befinden sich die Drähte, welche der Snare den Charakter geben.<sup>15</sup>

## 2.3 Musikproduktion

### 2.3.1 Grundlagen Musikproduktion

Um verstehen zu können, wie Künstliche Intelligenz Musik oder Audio generiert, müssen wir einige Begriffe und Parameter erklären.

Um Audio generieren zu können müssen wir verstehen, was Sound oder Schall eigentlich ist. In seinem Buch Handbuch der Tonstudioteknik (2014) definiert Michael Dickreiter Schall als die durch mechanische Schwingungen eines Mediums hervorgerufenen Druckschwankungen. Der Begriff Ton beschreibt eine sinusförmige Schallschwingung im Hörbereich. In der Musik wird ein Ton als eine einzelne Note definiert. In der Akustik wäre eine einzelne Note der Klang.

Schall ist eine Form von mechanischer Energie, die sich durch ein Medium ausbreitet. Die Schwingungen, die Schall verursachen, können von verschiedenen Quellen ausgehen, z. B. von Musikinstrumenten, Stimmen, Maschinen oder natürlichen Phänomenen wie Wind oder Regen. Die Schwingungen können sich durch verschiedene Medien

---

<sup>13</sup> Vgl. Sturgis, Joey: How To Get A Great Snare Sound In Any Mix, in: Joey Sturgis Tones, 2023, <https://joeysturgistones.com/blogs/learn/how-to-get-a-great-snare-sound-in-any-mix> (abgerufen am: 22.12.2023).

<sup>14</sup> Vgl. Miraglia, Dusti: What Are Transients? Amplify Your Mixes with Electrifying Dynamics and Unbelievable Control, in: Unison Audio, 2023, <https://unison.audio/what-are-transients/>, (abgerufen am: 22.12.2023).

<sup>15</sup> Vgl. Monahan Max, The Engineer's Guide to the Perfect Snare Sound, in: Sonicbids 2016, <https://blog.sonicbids.com/the-engineers-guide-to-the-perfect-snare-sound> (abgerufen am: 22.12.2023).

ausbreiten, z. B. durch Luft, Wasser oder feste Stoffe, da die Ausbreitung von Schall in Form von Wellen erfolgt.<sup>16</sup>

Die Wellenlänge ist der Abstand zwischen zwei aufeinanderfolgenden Maxima oder Minima. Die Frequenz ist die Anzahl der Wellen, die pro Sekunde durch einen bestimmten Punkt hindurchgehen. Die Frequenz des Schalls bestimmt seine Tonhöhe. Hohe Töne haben eine hohe Frequenz, tiefe Töne eine niedrige Frequenz. Die Lautstärke des Schalls wird durch die Amplitude der Schwingung bestimmt. Eine große Amplitude bedeutet eine hohe Lautstärke, eine kleine Amplitude eine geringe Lautstärke. Die Schallgeschwindigkeit ist die Geschwindigkeit, mit der sich Schall durch ein Medium ausbreitet. Schall kann durch verschiedene Medien absorbiert, reflektiert oder gestreut werden. Die Absorption von Schall durch ein Medium verringert die Lautstärke des Schalls. Die Reflexion desselben durch ein Medium erzeugt Echos. Die Streuung von Schall durch ein Medium verändert die Richtung der Schallausbreitung.

Schall ist ein wichtiger Bestandteil unseres Lebens. Wir verwenden Schall, um zu kommunizieren, Musik zu hören und unsere Umgebung wahrzunehmen.

In Bezug auf die Musikproduktion ist Schall die Grundlage für alle akustischen Signale. Die Tonhöhe, Lautstärke und Klangfarbe eines Musikinstruments werden durch die Schwingungseigenschaften des Instruments bestimmt. Die Tonhöhe eines Musikstücks wird durch die Frequenz des Schalls bestimmt, die Lautstärke durch die Amplitude und die Klangfarbe durch die Form der Schwingung.

Genau diese physikalischen Eigenschaften von Schall sind es, welche Audiogenerierung durch Künstliche Intelligenz komplex machen.<sup>17</sup>

### **2.3.2 Relevante Begriffe**

Für das Thema Künstliche Intelligenz in Verbindung mit Musikproduktion oder Soundgenerierung gibt es einige Begriffe, welche relevant sind.

Eine grundlegende Einheit in der Musikproduktion ist der Ton. Ein Klang hat eine bestimmte Tonhöhe, welche durch seine Frequenz, Amplitude und Klangfarbe charakterisiert wird.

---

<sup>16</sup> Vgl. Dickreiter, Michael/Dittel, Volker: Handbuch der Tonstudioteknik, Walter de Gruyter GmbH & Co. KG.: 2023, S 1-2.

<sup>17</sup> Vgl. Dickreiter/Dittel, 2023, Seiten 4-12

Die Tonhöhe, auch als Tonart bekannt, bezieht sich auf die wahrgenommene Frequenz eines Klangs. Sie wird durch die Anzahl an Schwingungen pro Sekunde bestimmt und kann als hoch oder tief klassifiziert werden.

Harmonie ist eine Kombination aus verschiedener Tonhöhen und Tönen. Wenn verschiedene Tonhöhen auf angenehme Weise miteinander kombiniert werden, entsteht ein Gefühl der Einheit und Kohärenz. Harmonie ist wichtig, um Akkorde zu bilden.

Akkorde sind die Grundlage der Harmonie in der Musik. Sie werden durch das gleichzeitige Spielen von zwei oder mehr Noten gebildet. Wenn Akkorde gespielt werden, erzeugen sie ein Gefühl von Spannung und Entspannung, welches wiederum bestimmte Gefühle beim Hörer auslöst.

Das Tempo ist eine weitere wichtige Einheit, welche meist in Schlägen pro Minute gemessen wird. Das ist auch bekannt als BPM. Das Tempo kann die Stimmung und die emotionale Wirkung eines Stücks erheblich beeinflussen.

Lautstärke bezieht sich auf die wahrgenommene Lautstärke eines Klangs. Sie steht in engem Zusammenhang mit seiner Amplitude oder Intensität. Die Lautstärke wird in Dezibel gemessen und kann von sehr leise bis sehr laut wahrgenommen werden.

Der Stil umfasst die charakteristischen Merkmale und Techniken eines Stückes, welche vom Komponisten benutzt wurden. Er ist die einzigartige Identität musikalischer Kreationen.

Polyphone Musik ist Musik, die aus mehreren unabhängigen Melodielinien besteht, welche gleichzeitig gespielt oder gesungen werden. Diese Melodielinien interagieren miteinander, um Harmonien, Kontrapunkte und Strukturen zu schaffen.

MIDI ist ein Standardprotokoll für die Kommunikation zwischen elektronischen Musikinstrumenten, Computern und anderen digitalen Geräten. Es ermöglicht den Austausch von musikalischen Informationen wie Noten, Anschlagstärken und Steuermeldungen zwischen verschiedenen Geräten und Softwareanwendungen. MIDI ist essentiell wenn es um symbolische Musikgenerierung geht.

Die Anschlagsgeschwindigkeit oder im Englischen Key Velocity genannt ist ein Maß dafür, wie stark eine Taste auf einem MIDI-Keyboard oder einem anderen MIDI-Instrument gedrückt wird. Zahlen um diesen Wert zu beschreiben befinden sich typischerweise zwischen 0 und 127. 0 bedeutet hier, dass die Taste überhaupt nicht gedrückt wurde und 127, dass die Taste mit maximaler Kraft gedrückt wurde.



ABC-Notation ist ein Kurzschrift-Notationssystem zum Schreiben von Musik. Es wird oft in der keltischen oder Volksmusiktradition verwendet, um traditionelle Musik weiterzugeben und zu verbreiten. Es werden Buchstaben und Symbole verwendet um Noten, Rhythmen und andere Elemente darzustellen. Die Notation ist daher wichtig, da sie einfach von Algorithmen zu lesen ist.

Als Pianoroll wird eine Schnittstelle in Digital Audio Workstations bezeichnet, die die Bearbeitung von MIDI Daten ermöglicht. Dabei wird ein Raster verwendet, bei dem die X-Achse die Zeit und die Y-Achse die Tonhöhe darstellt. Die Dauer und die Intensität der Noten sind einstellbar, was sie zu einem integralen Bestandteil der Struktur von Musikkompositionen macht.

Das Chronogramm ist eine Visualisierung, die die Intensität verschiedener Tonhöhen über eine bestimmte Zeit sichtbar macht.<sup>18</sup>

---

<sup>18</sup>Zhu, Yueyue, et al.: A Survey of AI Music Generation Tools and Models, in: arXiv.org: 2023, <https://arxiv.org/abs/2308.12982> (abgerufen am: 05.12.2023), Seiten 2-3

## 3 Künstliche Intelligenz

### 3.1.1 Stand der Technik

In den letzten Jahren ist Künstliche Intelligenz schlauer und menschenähnlicher geworden. Künstliche Intelligenz ist auf den Gebieten Bilderkennung, Sprachtranskribierung oder Übersetzung sehr gut, wenn nicht sogar besser geworden als der Mensch. Diese Systeme sind in der Lage dazu, Texte zu analysieren und spezifische Antworten dazu zu geben, zu fahren, Gesichter zu erkennen, Emotionen aus Fotos und Video zu erkennen, Drehbücher zu schreiben, Kochrezepte erstellen und Poesie zu schreiben. Hierbei hat sich in der Geschichte der Künstlichen Intelligenz echter Nutzen für eine Breite Masse von Menschen herausgetan. Sogar bei monotoner Arbeit wie Spam Filter bei E-Mails ist Künstliche Intelligenz einsetzbar und durchaus nützlich. Die Anwendungsmöglichkeiten von Künstlicher Intelligenz sind vielfältig. Hierbei können wir die Anwendungsmöglichkeiten in Schlüsseltechnologien und Kategorien der Anwendung aufteilen.

Schlüsseltechnologien umfassen mehrere Kategorien. Dazu gehören Deep Learning, Maschinenlernen, Verarbeitung natürlicher Sprache, virtuelle persönliche Assistenten und viele mehr.

KI-Anwendungen sind in der Regel in 3 Kategorien einteilbar. Allen voran geht es hier um Informationsverarbeitung, Integrierung und Analysis. Dazu gehören Suchmaschinen, Meinungsabbau, Sprach und Handschrifterkennung, Aktienmarktanalyse und vieles mehr.

Zweiten gibt es KI-Tools. Diese werden unter anderem für Gesichtserkennung, Spamfilter und Testen von Software verwendet. Abschließend beinhalten KI-gestützte Dienstleistungen gezielte Werbung und Kundensegmentierung, die Klassifizierung von DNA-Sequenzen, Objekterkennung durch Computervision, Bioinformatik und chemische Analyse sowie Rechtsfallrecherche.<sup>19</sup>

---

<sup>19</sup>Vgl. De Spiegeleire, Stephan., et al.: AI – TODAY AND TOMORROW. In ARTIFICIAL INTELLIGENCE AND THE FUTURE OF DEFENSE: STRATEGIC IMPLICATIONS FOR SMALL- AND MEDIUM-SIZED FORCE PROVIDERS, in: Hague Centre for Strategic Studies: 2017, <http://www.jstor.org/stable/resrep12564.8> (abgerufen am: 10.10.2023), S. 44-46.

Mit dieser breitgefächerten Masse an Anwendungsoptionen, welche durchaus nützlich sind, gibt es allen Grund anzunehmen, dass Künstliche Intelligenz Arbeit in vielen Bereichen einfacher gestaltet oder sogar ganz ersetzt. Dies bringt Vorteile aber auch Nachteile mit sich, zu welchen ich in dieser Thesis mit Bezug auf die Musikproduktion später näher eingehe.

Allgemein kann man aber davon ausgehen, dass Künstliche Intelligenz viele einfache, monotonen Aufgaben in Zukunft übernehmen wird. Das bringt den Vorteil mit sich, dass der Arbeiter, in welchem Arbeitsfeld auch immer, sich den anspruchsvollen, wertvollen und interessanten Aufgaben widmen kann.<sup>20</sup>

Oben erwähnte Tätigkeiten wie Spracherkennung, Deep Learning oder Objekterkennung durch Computervision wird durch „Big Data“ Möglich gemacht. Big Data ist ein Begriff, welcher große Datenmengen umschreibt, die nicht mit konventionellen Methoden verarbeitet werden können. Es braucht hier Künstliche Intelligenz um diese Datensätze zu analysieren und daraus Eindrücke und Werte zu generieren. Große Datensätze zu analysieren und zu verwerten ist eine Fähigkeit, welche sich besonders produktiv auf verschiedene Geschäftsmodelle und Forschungskapazitäten auswirkt. Gerade in nachhaltigen Geschäftsmodellen wird Künstliche Intelligenz schon vermehrt eingesetzt. Ein Beispiel ist die Agrarwirtschaft, um Versorgungsketten zu überprüfen, Nahrungssicherheit zu gewährleisten und Hygiene Standards aufrecht zu erhalten. Durch diese disruptive Entwicklung von Künstlicher Intelligenz im Zusammenhang mit Entwicklung effektiverer Geschäftsmodelle wird der Konkurrenzkampf verstärkt. Der Druck ist hoch, Performance aufrecht zu erhalten und voranzutreiben, da mehr und mehr Unternehmen den technologischen Vorteil in einer technologisch motivierten Welt zu erlangen.

In der letzten Dekade wurden eine große Vielfalt an Datensätzen generiert, welche zu sehr guten KI-gesteuerten Datenverarbeitungstools führten. Dementsprechend wuchs der Bereich Big Data Analytics. Firmen wie Netflix, Google, Amazon, Airbnb setzen vermehrt auf Big Data Analytics in großem Rahmen, wodurch strukturelle Veränderungen in diesen Firmen entstehen. Beispielsweise schauen Investoren vermehrt auf eher alternative Daten, welche nicht in die Bereiche Wertpapierpreise,

---

<sup>20</sup>Vgl. James Hayton et al.: What drives UK firms to adopt AI and robotics, and what are the consequences for jobs?, in: Institute for the Future of Work: London: 2023, [https://global-uploads.webflow.com/64d5f73a7fc5e8a240310c4d/650a05c1b2daf9e31b0ae741\\_FINAL%20WP%20-%20Adoption%20of%20Automation%20and%20AI%20in%20the%20UK.pdf](https://global-uploads.webflow.com/64d5f73a7fc5e8a240310c4d/650a05c1b2daf9e31b0ae741_FINAL%20WP%20-%20Adoption%20of%20Automation%20and%20AI%20in%20the%20UK.pdf) (abgerufen am 10.10.2023), Seiten 4-5.

Unternehmensgrundlagen oder Makroökonomische Indikatoren fallen. Beispielsweise werden hier Daten ausgewertet, welche Informationen über Umwelt, Soziales und Unternehmensführung beinhalten. Datensätze hierfür sind Audioaufnahmen, Artikel oder Beiträge in den Sozialen Medien. Diese Daten haben einen großen Einfluss auf Investitionsentscheidungen.<sup>21</sup>

Big Data Analysis und Generative AI sind beides Arten von Künstlicher Intelligenz, die von großen Datenmengen abhängig sind. Wo Big Data Analysis den Vorteil mitbringt, große Datenmengen verarbeiten zu können ist auf der anderen Seite Generative AI ein fortschrittlicher Bereich des maschinellen Lernens, welcher die Fähigkeit hat, Daten zu generieren.

Generative Künstliche Intelligenz hat in den letzten Jahren sehr viel Aufmerksamkeit auf sich gezogen. Besonders einzelne Projekte und Programme, wie Chat GPT von der Firma Open AI. 80% der Fortune 500 Unternehmen haben bereits Mitarbeiter, die ChatGPT für ihre Arbeit nutzen heißt es auf der Website von Open AI. Chat GPT ist eine Variante von Open AI's Generative Pre-trained Transformer Modellen, die speziell für Konversationen und Chats optimiert ist. Die Software ist ein Modell für Maschinelles Lernen, das darauf spezialisiert ist, menschenähnliche Texte basieren auf Daten zu generieren, mit welchen es trainiert wurde. Dieses GPT-Modell kann in einer Vielzahl von Anwendungen eingesetzt werden, von konversationsbasiert bis hin zur Textgenerierung.<sup>22</sup>

Der Schlüsselfaktor für eine Generative AI wie ChatGPT ist es eine enorme Datenmenge als Grundlage. Es werden hier Datensätze von Wikipedia, Github, Soziale Medien eingesetzt um die Künstliche Intelligenz zu trainieren.

Chat GPT ist ein Large Language Model. Es besteht aus einem neuronalen Netzwerk, welches wiederum aus kleinen Mathematischen Formeln besteht, auch Neuronen genannt. Jedes Neuron ist mit einem anderen verbunden. Die Stärke oder Qualität der Verbindung wird durch eine Nummer zugeteilt. Sie legen fest, inwiefern die Ausgabe eines Neurons als Eingabe für ein nachfolgendes Neuron berücksichtigt wird. Ein Neuronales Netzwerk kann sehr klein sein, beispielsweise sechs Neuronen mit 8

---

<sup>21</sup>Vgl. P. V. Thayyib et al.: State-of-the-Art of Artificial Intelligence and Big Data Analytics Reviews in Five Different Domains: A Bibliometric Summary, in: Multidisciplinary Digital Publishing Institute, bd. 15, Nr. 5, 2023, <https://www.mdpi.com/2071-1050/15/5/4026> (abgerufen am: 11.10.2023), Seiten 2-3.

<sup>22</sup> Vgl. Open AI, What is ChatGPT, in: Open AI, 2023, <https://help.openai.com/en/articles/6783457-what-is-chatgpt>, (abgerufen am: 11.10.2023).

Verbindungen zwischen Ihnen. Chat GPT ist ein Large Language Model, welches Millionen von Neuronen hat und Milliarden von Verbindungen zwischen ihnen.<sup>23</sup>

Der Überbegriff, welchen man für diese Art von Neuronaler Netzwerke verwendet heißt Machine Learning. Diese Technologie treibt viele Aspekte der modernen Gesellschaft an und gleicht dem Menschlichen Hirn. Von Websuchen über Content-Filterung in Sozialen Netzwerken aber auch Empfehlungen auf E-Commerce-Websites. Machine Learning Systeme können Objekte identifizieren und Sprache in Text umwandeln. Hierbei wird die Technik Namens Deep Learning verwendet. Deep Learning Netze haben mehrere Schichten von Neuronen, wobei jede einzelne Schicht eine neue Repräsentation der Eingabedaten lernt. Die erste Schicht an Neuronen lernt grundlegende Merkmale wie Farben oder Strukturen. Die zweite Schicht lernt komplexere Daten, welche wiederum aus den Daten der ersten Schicht abgeleitet werden. Dies wird fortgezogen, bis die oberste Schicht erreicht ist und die gewünschte Aufgabe erledigt. Genauer wird hier ein Algorithmus Namens Stochastic Gradient Descent verwendet. Er funktioniert, indem er iterativ die Gewichte der Neuronen im Netzwerk aktualisiert, um Unterschiede zwischen den vorhergesagten und den gewünschten Ausgaben zu minimieren. Er wählt zufällig eine kleine Menge an Beispielen aus einem Trainingsdatensatz aus. Der Algorithmus berechnet nun die Vorhersagen des Modells für diese Beispiele. Dann wird berechnet, ob es Fehler zwischen Vorhersagen und den gewünschten Ausgaben gibt. Die Neuronen werden nun gewichtet, um Fehler zu minimieren. Daraufhin wird der Prozess wiederholt, bis der Fehler nicht mehr weiter abnimmt. Er wird stochastisch genannt, da er eine Schätzung über den Durchschnittsgradienten über alle Beispiele hinweg abgibt.<sup>24</sup>

Chat GPT ist in einfachen Worten ein Computerprogramm, welches eine Eingabe braucht um ein großes Volumen an Daten zu überprüfen. Im Falle von einem Large Language Model wird so viel grammatikalischer Text wie möglich eingeführt und davon das Modell trainiert. Anfangs wird das Language Modell Unsinn generieren. Wenn man allerdings dem Programm sagt, es solle den Output mit dem Input vergleichen, kann es Verbesserungen vornehmen. Mit genug Zeit und Ressourcen lernt das Modell, Text zu produzieren, welches einem Menschen gleicht.<sup>25</sup>

---

<sup>23</sup>Vgl. Muehmel Kurt: What Is a Large Language Model, the Tech Behind ChatGPT?, Blogbeitrag von Dataiku, 2023, <https://blog.dataiku.com/large-language-model-chatgpt> (abgerufen am: 12.10.2023).

<sup>24</sup> LeCun, Yann & Bengio, Y. & Hinton, Geoffrey: Deep Learning, in: Nature, Bd. 521, Nr. 7553, 2015, [https://www.researchgate.net/publication/277411157\\_Deep\\_Learning](https://www.researchgate.net/publication/277411157_Deep_Learning) (abgerufen am: 12.10.2023), S. 436-437.

<sup>25</sup> Vgl. Kurt Muehmel, 2023.

Neben Deep Learning, Machine Learning und Natural Language Processing gibt es außerdem noch Robotics, Fuzzy Logic und Expert Systems.

Der Bereich AI Robotics fokussiert sich auf Anwendungen im realen Leben, welche reale Auswirkungen haben. Auch Chat GPT wird hier in der Rolle der AI Robotics angesehen. Die Grenzen der Kategorien sind also allgemein als fließend anzusehen, da alle Bereiche miteinander arbeiten. Aber auch Versorgungsrobotics im Zusammenhang mit Bedürftigen Verpflegung fällt unter den Begriff.

Fuzzy Logic betrifft Anwendungen unter anderem im medizinischen Feld und ist involviert in Problemlösung, aber auch Entscheidungsverfahren. Beispielsweise wird Fuzzy Logic bei Erkennung von periodontale Krankheiten oder Infektionskrankheiten eingesetzt.

Expert Systems sind Funktionen, welche Entscheidungsverhalten von Experten lernt und anstelle dieser ausführt. Sie funktionieren klassischerweise auf der „Wenn, dann“ Logik um komplexe Probleme zu lösen. Beispielsweise sind Exptert Systems in den Bereichen wie Schadsoftwareerkennung oder der Erstellung von medizinischen Aufzeichnungen.<sup>26</sup>

Die Anzahl an Forschungen über das Thema Künstliche Intelligenz ist in den letzten Jahren stark angestiegen. Dies ist in der nachfolgenden Abbildung zu sehen.

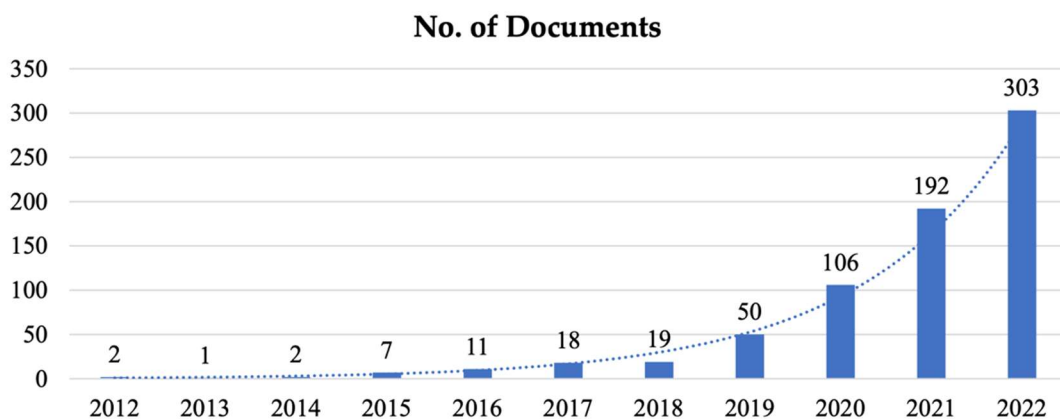


Figure 4. AI and BDA Bibliometric Review Growth.

Abbildung 1: Number of Documents (P. V. Thayyib et al, Sustainability, 2023)

Die Abbildung veranschaulicht die Häufigkeitsverteilung des Wachstums der Veröffentlichungen ab 2012. Von 2012 bis 2018 zeigt sich, dass die Produktion an Publikationen relativ gleichmäßig anstieg. Ab 2019 wuchs die Anzahl jedoch rapide. Die Studie bezieht sich auf einen Datensatz von 711 bibliometrischen Artikeln in 433 verschiedenen Fachzeitschriften. Diese wurden zwischen den Jahren 2012 und 2022

<sup>26</sup> Vgl. P. V. Thayyib et al., 2023, S. 5-6.

veröffentlicht. Die Anzahl an Zitationen betrug 17,9 pro Dokument. Die Anzahl an Autoren der Artikel betrug 2334. Zusätzlich haben die Autoren in ihren Artikeln 47722 Quellen zitiert. Die Daten in der Studie wurden für das Jahr 2022, den 23. Oktober abgerufen.

Es ist davon auszugehen, dass die Zahl an Publikationen zum Thema Künstliche Intelligenz in den nächsten Jahren weiter ansteigt. Besonders Länder wie China und Indien haben nun begonnen in den Bereichen Künstliche Intelligenz und Big Data Analysis zu forschen.<sup>27</sup>

---

<sup>27</sup> Vgl. P. V. Thayyib et al., 2023, S. 11-12

## 4 Audiogenerierung durch Künstliche Intelligenz

Musik ist ein wichtiger Bestandteil der menschlichen Kultur und hat im Laufe der Jahrhunderte viele Entwicklungen durch Faktoren wie: Kultur, Technologie und individueller Stil durchgemacht. Besonders in der Kategorie der Musikgenerierung hat Künstliche Intelligenz einen Paradigmenwechsel hervorgebracht. Das Aufkommen von tiefen Neuronalen Netzwerken, auch bekannt als Deep Learning hat seit 2012 verschiedenste Informatikdisziplinen verändert. Darunter auch die KI-Musikgenerierung. Hier generieren Deep Learning Netzwerke fortlaufende Melodien, auch wenn diese nicht unbedingt einem zentralen Motiv folgen und möglicherweise menschlichen Input benötigen. Deep Learning in Musikgeneration hat immer noch seine Limitierungen. Beispielsweise fehlende Kreativität und Kontrolle.<sup>28</sup>

Kreativität ist sehr subjektiv und ist abhängig von der Beobachtung eines Betrachters und dessen Hintergrund in Kultur. Auch wird bei Bewertungen kaum der Entstehungsprozess der Sache betrachtet, sondern das Endergebnis selbst. Das heißt also ein kreativer Prozess ist anerkannt, wenn eine Gruppe von Menschen die Sache als kreativ ansehen. Hier können wir also als Gruppe von Menschen entscheiden, ob Musik von KI kreiert, als kreativ angesehen werden kann. Dies zu messen kann allerdings herausfordernd sein. Welche Qualität AI generierte Musik hat und wie kreativ das Musikstück ist, wird durch Umfragen bestimmt. Es gibt Umfragen, bei welchen getestet wird, ob der Hörer erkennen kann, ob die Musik AI-generiert ist oder nicht.<sup>29</sup>

Sprachgenerierung, auch wenn Musik nicht als Sprache anerkannt wird, so gibt es strukturelle Elemente in Sprache, welche Musik ähneln. Musikalische Elemente, wie Noten oder Tempo geben die Möglichkeit diese Eigenschaften in Worten zu beschreiben oder zu übersetzen. Indem man eine Beziehung zwischen Sprache und Musik herstellt, gibt es die Möglichkeit, bestehende Modelle der Spracherzeugung auf die Musikerzeugung anzuwenden.

Der Bereich des Gehirns, welcher dafür zuständig ist, Sprache zu verstehen ist auf der linken Hirn-Hemisphäre zu lokalisieren. Auch wird diesem Bereich zugesprochen, dass

---

<sup>28</sup> Yueyue Zhu et al., S. 3.

<sup>29</sup> Vgl. Hernandez-Olivan, Carlos et al.: A Survey on Artificial Intelligence for Music Generation: Agents, Domains and Perspectives, in: arXiv.org (Cornell University), 2022, <https://arxiv.org/pdf/2210.13944.pdf> (abgerufen am: 02.12.2023), S. 2-3.



er für die Erkennung von Rhythmen, temporalen und sequentiellen Strukturen und für Melodieerkennung zuständig ist. Außerdem sagt das Gehirn automatisch das nächste gesprochene Wort voraus, was genauso in Deep Language Autoregressive Models vorkommt. Teile des Gehirns, welche dafür zuständig sind, Text und Sprache zu verstehen sind auch aktiv, wenn es darum geht Musik zu erkennen und zu verstehen.<sup>30</sup>

Musik besteht aus fundamentalen Faktoren wie Tempo, Lautstärke oder Stil. Verbindet man nun Musikproduktion mit digitaler Technologie und Künstlicher Intelligenz, gilt es folgende Begriffe zu beachten. Das MIDI Protokoll zur Kommunikation zwischen elektronischen Musikinstrumenten und Computer und anderen Digitalen Komponenten oder Key Velocity, um zu messen wie stark eine Taste beispielsweise eines elektronischen MIDI-Pianos gedrückt wird. Die genannten Begriffe sind Parameter, welche ein KI-Programm nutzen kann um Aktionen zu erkennen und auszuführen.<sup>31</sup>

Eines haben die meisten Methoden gemeinsam. Und zwar große Datensätze, welche auf vorhandener Musik basieren. Man muss hier zwischen symbolischen und nicht-symbolischen Datensätzen unterscheiden. Auch wird symbolische und nicht-symbolische Musik generiert. Damit ist gemeint, dass im Datensatz kein Audio direkt zugrunde liegt, sondern eher ein Datensatz an MIDI-Noten, beziehungsweise Notenblätter übertragen in MIDI-Noten. Ein symbolischer Datensatz ist also von MIDI-Noten und Befehlen abhängig und nicht unbedingt von echter Musik. Bei der Generierung von Musik ist es außerdem wichtig, zu beachten, dass man dem Algorithmus ein bestimmtes Genre füttert. Laut der Studie von Miguel Civit et al. Zugänglich auf der Website von Science Direct, gibt es hier ein bestimmtes Dataset, welches weiterverbreitet ist und über Hundertfünzigtausend MIDI Daten beinhaltet. Auch gibt es Datasets, welche synchronisierte Audiofiles in Verbindung mit symbolischen Informationen benutzen. Weiter gilt es zu beachten, dass bei Symbolischer Audio Generierung zwischen Multitrack und Singletrack unterschieden wird. Multitrack heißt, dass das System mehrere MIDI-Spuren ausgibt. Bei Singletrack wird davon ausgegangen, dass die einzelne Spur etwa ein Piano oder eine Orgel darstellt.

---

<sup>30</sup>Vgl. Carlos Hernandez-Olivan et al., 2022, S. 3-4.

<sup>31</sup>Vgl. Yueyue Zhu et al., 2023, S. 3.

Zur Zeit der Veröffentlichung der Studie war es üblich, bei Musik generierenden Systemen, auf ein User-System Interface zu verzichten. Es wurde also nicht unbedingt darauf geachtet, wie bedienbar das System ist.<sup>32</sup>

Um Musik nun zu generieren, werden verschiedene Tools und Methoden verwendet. Dies wird mit Methoden, welche Neuronale Netze benutzen, ermöglicht aber auch durch Methoden, welche keine Neuronale Netze brauchen. Traditionell verwenden AI Tools eine Vielfalt an Methoden, hauptsächlich sind oder waren diese parametergetriebene Algorithmen, welche menschlichen Input brauchten. Diese sind hier nicht unbedingt Künstliche Intelligenzen, sondern vielmehr normale Algorithmen, zum Beispiel basierend auf der Markov Chain. Spätere, aktuellere Modelle verwenden eher Neuronale Netzwerke. Die Qualität der Ergebnisse ist bei Neuronalen Netzen besser als bei parametergetriebenen Algorithmen. Beispielsweise brauchen sie Parameter, wie Tempo oder Tonart. Tools in Verbindung mit Neuronalen Netzen, brauchen diese Parameter nicht unbedingt. Bei Modellen mit Neuronalen Netzen gibt es zwei Kategorien. Prompt basierte Modelle und visuell basierte Modelle. Prompt basierte Modelle benutzen Prompts, bei welchen beschrieben wird, wie sich die Musik anhören soll. Visuell basierte Modelle benutzen Bilder oder Videos um darauf Musik zu generieren.<sup>33</sup>

## 4.1 Symbolische Audiogenerierung

### 4.1.1 Markov Ketten

Wie oben schon erwähnt kommen bei Systemen, ohne Neuronale Netzwerke, sogenannte Markov Ketten zum Einsatz. Markov-Ketten sind mathematische Modelle, die zur Analyse und Vorhersage des Verhaltens von Systemen verwendet werden, die eine Eigenschaft aufweisen, die so genannte Markov-Eigenschaft. Diese Eigenschaft besagt, dass der zukünftige Zustand eines Systems nur von seinem aktuellen Zustand abhängt und nicht von seiner Vergangenheit. Im Zusammenhang mit der Musikgenerierung können Markov-Ketten verwendet werden, um die Wahrscheinlichkeit des Übergangs von einer musikalischen Note oder einem Ereignis zu einem anderen zu modellieren. Dies ermöglicht die Erzeugung neuer Melodien und Harmonien, die reibungslos und natürlich

---

<sup>32</sup>Vgl. Miguel Civit et al., A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends, Expert Systems with Applications, Expert Systems with Applications: Volume 209, 2022, <https://www.sciencedirect.com/science/article/pii/S0957417422013537?via%3Dihub> (abgerufen am 22.12.2023)

<sup>33</sup>Vgl. Yueyue Zhu et al., 2023, Seite 6.

ablaufen.. Eine andere Gruppe von Arbeiten konzentriert sich auf die regelbasierte Musikgenerierung, bei der vordefinierte Regeln verwendet werden, um Musikdaten zu erstellen, die bestimmten Mustern oder Stilen folgen. Ein Beispiel für diesen Ansatz liefert Spangler, der ein System für die Echtzeitbegleitung vorstellte. Dieses System extrahiert und verwendet einen Satz harmonischer Regeln aus musikalischen Beispielen, um als Reaktion auf eine Eingabemelodie neue Harmonien zu erzeugen, und demonstriert damit die Anwendung deterministischer Regelsätze bei der Musikerzeugung.<sup>34</sup>

#### **4.1.2 Evolutionäre Algorithmen**

Weiter gibt es Evolutionäre Algorithmen. Sie sind eine Art von Algorithmen der künstlichen Intelligenz, welche den Prozess der Evolution nachahmen um neue Lösungen für Probleme zu finden. Im Zusammenhang mit der Erzeugung von Musik können evolutionäre Algorithmen dazu verwendet werden, neue Melodien oder Musikstücke zu erzeugen, die einem gewünschten Stil entsprechen. Um einen evolutionären Algorithmus für die Musikgenerierung zu verwenden, beginnt man mit einer Reihe von Ausgangsmelodien, die als Chromosomen dargestellt werden. Jedes Chromosom besteht aus Genen, die Noten, Akkorde, Takte oder Gruppen von Takten verwenden. Nun werden die Chromosomen mit der Mutationsfunktion zufällig verändert. Um dabei nicht gegen die Regeln der Musiktheorie zu verstoßen, basiert diese Aktion auf harmonischen Regeln. Nach dieser Mutation werden nun Gene aus zwei verschiedenen Chromosomen kombiniert. Dann wird überprüft, ob sich das Chromosom an die vorgegebenen Regeln des Musikstils hält. Der Algorithmus gibt dann sogenannte Fitnesswerte für die Chromosomen aus, womit das System nun entscheiden kann, welche Chromosomen erhalten bleiben, um somit die Basis für die nächsten Chromosomen zu bilden. Dieser Prozess wird so lange wiederholt, bis die Qualität der erzeugten Musik zufriedenstellend ist.<sup>35</sup>

#### **4.1.3 Feedforward Netzwerke**

Ein neuronales Feedforward Netzwerk ist ein Netzwerk, bei welchem die Zusammenschlüsse der Knotenpunkte keinen Kreis formen. Meistens werden diese Algorithmen in Bildverarbeitungs Applikationen verwendet. Diese Verarbeitungstechnik

---

<sup>34</sup> Vgl. Yueyue Zhu et al., 2023, Seite 9

<sup>35</sup> Vgl. Miguel Civit et al., 2022

kann verwendet werden, wenn auch die Datensätze Fehler beinhalten oder lückenhaft sind.<sup>36</sup>

#### 4.1.4 Recurrent Neural Networks

Viele Aufgaben von Algorithmen erfordern den Umgang mit sequentiellen Daten. Sprachsynthese oder Musikgenerierung erfordern, dass die Modelle sequenziell strukturierte Daten aufnehmen als auch ausgeben. Bei Rekurrenten neuronaler Netze wird die Ausgabe vorheriger Schritte wieder in das neuronale Netz eingespeist. Dabei wird der Zustand des Netzes nach jeder Eingabe aktualisiert, sodass es sich an die vorherigen Eingaben erinnern kann. Rekurrente neuronale Netzwerke verfügen also über einen internen Speicher, der es ermöglicht, die vorherigen Eingaben, die nachfolgenden Vorhersagen zu beeinflussen. In Bezug auf Sprachsynthese ist es viel einfacher das nächste Wort in einem Satz mit größerer Genauigkeit vorherzusagen, wenn man weiß, was die vorherigen Wörter waren. Das trifft natürlich auf sequentielle Notenfinden zu, wie es bei Musikgeneratoren der Fall ist.<sup>37</sup>

#### 4.1.5 Generative adversarial networks

Das Kernkonzept von GAN's ist es, zwei Netzwerke aufzubauen und somit ein kontradiktorisches Lernen zu erreichen. Es gibt hier das Generator Netzwerk und das Diskriminator Netzwerk. Hier gibt der Generator ein Signal aus. Der Diskriminator wird trainiert, reale Daten von denen zu unterscheiden, die vom Generator erzeugt wurden, während der Generator trainiert wird, den Diskriminator zu täuschen. Die beiden Netzwerke werden simultan trainiert. Über eine gewisse Zeit lernt der Generator, bessere, realistischere Daten zu produzieren und der Diskriminator lernt, zwischen generierten und realen Daten besser zu unterscheiden. In den letzten Jahren sind die Anzahl an Studien der GAN basierten Architekturen deutlich angestiegen.<sup>38</sup>

---

<sup>36</sup>Vgl. Tsantekidis, Avraam, et al.: Chapter 5 - Recurrent neural networks, in: Academic Press, 2022, <https://www.sciencedirect.com/science/article/abs/pii/B9780323857871000105> (abgerufen am: 07.12.2023), S. 101-115.

<sup>37</sup>Vgl. Aston Zhang, Zack C. Lipton et al.: Dive into Deep Learning, in: Cambridge University Press, 2023, [https://d2l.ai/chapter\\_recurrent-neural-networks/index.html](https://d2l.ai/chapter_recurrent-neural-networks/index.html) (abgerufen am: 07.12.2023).

<sup>38</sup>Vgl. Hao-Wen Dong/Wen-Yi Hsiao, et al.: useGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment., in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, <https://doi.org/10.1609/aaai.v32i1.11312> (abgerufen am: 07.12.2023), S. 35

#### 4.1.6 Variational autoencoders

Variationale Atukodierer sind so konzipiert, dass sie Eingangsinformationen kodieren und versuchen sie so genau wie möglich zu dekodieren, also rekonstruieren. Diese Art von Encoder kann die grundlegenden Muster in einem Trainingsdatensatz lernen und unwahrscheinliche Möglichkeiten ausschließen. Dadurch werden die Informationen über ein Musikstück in eine kleinere Menge an Informationen komprimiert, die im latenten Raum dargestellt werden. Der latente Raum ist ein abstrakter Raum, in dem die Informationen über ein Musikstück unabhängig von der Zeit dargestellt werden können. Der Encoder lernt also, welche Kombinationen von Noten, Rhythmen und Harmonien unwahrscheinlich sind oder nicht zum jeweiligen Musikstil passen.<sup>39</sup>

#### 4.1.7 Transformer Netzwerke

Auch Transformer Netzwerke sind dafür da, sequenzielle Daten wie Text, Audio und Video zu verarbeiten. Auch hier gibt es wieder einen Encoder und einen Decoder. Der Encoder nimmt eine Eingabesequenz und erzeugt eine Repräsentation dieser Sequenz. Der Decoder nimmt diese Repräsentation und erzeugt eine Ausgabesequenz. Der Encoder besteht aus einigen sogenannten Multi-Head-Aufmerksamkeitsschichten. Jede Aufmerksamkeitsschicht verwendet eine Aufmerksamkeitsfunktion. Hier werden die Beziehungen zwischen den Elementen der Eingabesequenz berücksichtigt. Die Aufmerksamkeitsfunktion berechnet eine Gewichtung für jedes Element der Eingabesequenz, die angibt, wie wichtig dieses Element ist.

Der Decoder funktioniert nach dem gleichen Prinzip. Das Transformer Netzwerk wird mit einem Trainingsdatensatz von Musikstücken trainiert. Nun wird das Netzwerk mit einer Startsequenz oder einem Anfangsmotiv gefüttert. Das Transformernetzwerk erzeugt nun eine Sequenz von Noten. Die Aufmerksamkeitsschichten dienen hierbei zur Gewichtung, welche Elemente der Eingabesequenz wichtig für die Ausgabe sind und welche nicht. Mit dieser Fähigkeit ist es Transformer Netzwerken möglich, Stilmerkmale eines Musikstils zu erkennen.<sup>40</sup>

---

<sup>39</sup> Vgl. Miguel Civit et al., 2022

<sup>40</sup> Vgl. Vaswani, Ashish/Shazeer, Noam M., et al.: Attention is All you Need, In: semanticscholar.org, 2017, <https://www.semanticscholar.org/paper/Attention-is-All-you-Need-Vaswani-Shazeer/204e3073870fae3d05bcbc2f6a8e263d9b72e776> (abgerufen am: 07.12.2023), S.2-3

### 4.1.8 Sprache und Künstliche Intelligenz

Genau, wie bei Musik und Signalerkennung, beziehungsweise Signalgenerierung gibt es Eigenschaften, die im Programm zu Parametern werden, welche der Algorithmus lesen und benutzen kann. Dies gibt einen Einblick, wie generierende und lesende Algorithmen funktionieren. Hier gibt es 2 Hauptkategorien, die es zu beachten gilt und welche als Parameter verwendet werden können.

Die Zeit Domäne und die Frequenz Domäne. Die Zeit Domäne beinhaltet beispielsweise den Faktor Energie. Energie ist eine messbare Eigenschaft, welche die Amplitudeneigenschaften eines Sprachsignals im Zeitverlauf beschreibt. Die Gesamtstärke und Dynamik des Signals werden erfasst und zeitliche Schwankungen der Intensität aufgezeigt. Der Energieparameter gibt Aufschluss darüber, ob es zeitliche Segmente mit höherer oder niedriger Amplitude gibt. Er hilft auch bei der Identifizierung von Ereignissen und Übergängen, die auf Veränderungen der Stimmaktivität hinweisen. Variationen in der Amplitude zu messen trägt dazu bei, Sprachsignale und deren akustischen Eigenschaften zu verstehen.

Weiter gibt es die so genannte „Zero-crossing rate“, welche beschreibt, wie oft das Sprachsignal die Null Axe in einem definierten Zeitraum überschreitet. Dies wird analysiert indem der Algorithmus oder das Programm die Anzahl an Veränderungen des Signals in einem bestimmten Zeitraum zählt.

Dazu kommt der Pitch, mit welchem erkennbar wird, welchen Grundton der Sprecher hat. Linear predictive Coding ist eine leistungsstarke Technik, die das Sprachsignal als lineare Kombination vergangener Abtastwerte darstellt und ein autoregressives Modell verwendet. Die enthaltenen Daten sind wichtig für die Verarbeitung verschiedener Sprachaufgaben.

Bei der zweiten wichtigen Domäne, der Frequenz Domäne geht es darum Daten in einem Spektrum zu lesen und zu verarbeiten. Spektrogramme sind visuelle Repräsentationen, welche Variationen im Frequenzspektrum in einem Signal über Zeit darstellen. Spektrogramme werden zur Analyse von Schallsignalen verwendet und bietet hier dem Algorithmus oder der Künstlichen Intelligenz relevante Daten.<sup>41</sup>

---

<sup>41</sup>Vgl. Mehrish Ambhuj et al.: A Review of Deep Learning Techniques for Speech Processing, in: arXiv.org, 2023, <https://arxiv.org/abs/2305.00359>, (abgerufen am: 02.12.2023), S. 3-8.

## 4.2 Nicht Symbolische Audio Generierung

Audio Signale zu generieren stellt sich als größere Herausforderung dar, als die Generierung von symbolischer Musik. Die Aufmerksamkeit lag daher bisher eher nicht bei der Generierung von Audiosignalen. Ein Musikstück besteht oft aus wiederkehrenden Elementen, welche auf verschiedenen Ebenen von Motiven über Phrasen bis hin zu Abschnitten, wie Strophe oder der Refrain zusammenarbeiten. Um hier ein fortlaufendes Stück zu generieren, muss ein System auf Elemente in der Vergangenheit zugreifen können. Während der Generierung muss wird ersichtlich, was in der Strophe passiert ist oder welche Noten im Intro gespielt wurden, um einen Song in Szene zu setzen. Das Problem bei der Generierung von Audio in Hoher Qualität ist es, dass Audio über einen sehr weiten Bereich von Zeitskalen abgebildet wird. Zum Beispiel kann eine einzelne Audiodatei eine Dauer von mehreren Sekunden haben, aber auch einzelne Audioproben von nur wenigen Millisekunden. Bildgenerierung durch Deep Learning Modelle benutzen 2 Dimensionale Datensätze. Aus diesen Bildgenerierungsmodellen stammen viele Audioerkennungs- und Generierungsalgorithmen, doch unterscheiden sich die Eigenschaften von Audio- und Bilddatensätzen fundamental. Bilddatensätze sind zweidimensional, während Audiodateien eindimensionale Zeitseriensignale sind. Audiodateien werden in zweidimensionale Repräsentationen umgewandelt, welche die Faktoren Zeit und Frequenz besitzen. Bilder sind Schnappschüsse eines Moments. Audio oder Musik hat eine zeitliche Komponente, welche es zu berücksichtigen gilt.

"In der umfassenden Studie 'Deep Learning for Audio Signal Processing' von Hendrik Purwins und Kollegen wird die fortschrittliche Anwendung von Tiefenlernverfahren in der Audio-Signalverarbeitung detailliert untersucht. Es werden Klassifizierungsmethoden in der Verarbeitung von Audiosignalen, insbesondere im Kontext des maschinellen Lernens und künstlicher Intelligenz beschrieben. Es wird beschrieben, wie Generative Modelle, verschiedene Labels benutzen, um Dinge zu klassifizieren. Zum Beispiel globale Labels, um einen ganzen Song in einem Genre zu klassifizieren, oder lokale Labels pro Zeitschritt. Letztere sind spezifischer, welche jede Sekunde eines Songs identifizieren, um festzustellen ob es sich gerade zum Beispiel um einen Drum Break oder einen Chorus handelt. Außerdem gibt es Label Sequenzen variabler Länge. In diesem Kontext bezieht es sich auf die Kennzeichnung oder Etikettierung von Audioabschnitten, wobei die Länge dieser Abschnitte nicht festgelegt ist. Das bedeutet, in einem Audiozeitraum können bestimmte Teile, wie etwa eine Gesprächspassage oder ein musikalisches Solo, mit Labels versehen werden, und diese

Teile können unterschiedlich lang sein. Diese Art der Klassifizierung ist besonders nützlich in Situationen, wo die zu kennzeichnenden Audioereignisse nicht in regelmäßigen oder vorhersehbaren Intervallen auftreten.

Labels werden in der Audioanalyse eingesetzt, um komplexe Audio-Zeitreihen in verständliche Kategorien zu unterteilen. Diese Merkmale können dann verwendet werden, um das Signal zu klassifizieren oder zu analysieren. Aber um Audio klassifizieren und erkennen zu können, braucht es bestimmte grundlegende Methoden um Datensätze, also Audio zu beobachten und verarbeiten.

Eine davon ist die Mel-Frequency Cepstral Coefficients. Sie sind eine Art von Merkmalen, um die Tonhöhe und den Klang eines Audiosignals zu beschreiben. Das Frequenzspektrum wird berechnet, indem das Magnituden Spektrum des Audiosignals in ein reduziertes Set von Frequenzbändern projiziert wird. Nachdem die Frequenzbänder berechnet wurden, werden sie in logarithmische Größenordnungen umgewandelt. Jede Intensität der Schallwellen werden in jedem Frequenzband durch eine logarithmische Funktion dargestellt.

Um das zu ermöglichen wird die Mel-Filterbank verwendet, welche nah am Gehörssystem des Menschen modelliert ist. Sie ist bestehend, auf dem Prinzip, dass das menschliche Ohr die Änderung von Tonhöhe in tiefen Frequenzen besser erkennen kann als höhere. Die Mel-Filterbank unterteilt das Frequenzspektrum in eine Serie an Frequenzbändern, welche der logarithmischen Skala angepasst sind. Das macht die Mel-Filterbank verwendbar für Transpositionen, wie z. B. die Transkription von Musik und die Spracherkennung.

Ein log-mel Spektrogramm ist eine visuelle Repräsentation von Sound über Zeit. Es zeigt den Frequenzinhalt von Sound zu bestimmten Zeitpunkten. Bei einem normalen Bild sind benachbarte Pixel miteinander verbunden, um ein ganzes Bild zu ergeben. Genauso sind auf die Frequenzbereiche in einem Spektrogramm miteinander verbunden. Jedoch gibt es bei Schall zusätzliche Beziehungen, wie zum Beispiel der Fakt, dass es Obertöne gibt, welche das Vielfache derselben Grundfrequenz sind. Audio Signale können also mithilfe von verschiedenen Arten von Spektrogrammen durch Deep Learning Algorithmen analysiert werden. Ein tiefes neuronales Netzwerk ist ein Netzwerk mit vielen aufgehäuften Schichten. Diese werden durch verschiedene Algorithmen und Deep



Learning Modellen untersucht. Wenn man Audio Generierung verstehen will, müssen diese ebenfalls betrachtet werden.<sup>42</sup>

#### 4.2.1 WaveNet

WaveNet wurde von Forschern bei DeepMind entwickelt und hat die Landschaft der Audiogenerierung und Sprachsynthese maßgeblich beeinflusst. WaveNet ist ein tiefes neuronales Netzwerk, welches darauf trainiert wurde, Roh-Audiosequenzen direkt zu generieren. Eine besondere Fähigkeit von WaveNet ist es realistische menschliche Stimmen und Musik zu erzeugen. Die Studie hebt hervor, wie WaveNet durch die Verwendung eines tiefen, dilatierten Faltungsnetzwerks in der Lage ist, eine breite Palette von Zeitskalen von Audiosignalen zu erfassen. Die Idee hinter Faltungsnetzwerken, auch kausale Faltung genannt, ist es, dass die Vorhersagen des Modelles nur von vergangenen und aktuellen, nicht aber von zukünftigen Daten abhängen. Die Modellstruktur ist so gestaltet, dass bei der Berechnung eines Ausgangswertes nur die Datenpunkte berücksichtigt werden, die bis zu diesem Zeitpunkt aufgetreten sind. Der Ausgangswert ist hier zum Beispiel ein Ton oder ein Tonsegment. Kausale Faltungen sind ein Mechanismus in Modellen, wie WaveNet, die sicherstellen, dass Vorhersagen nur auf Informationen basieren, die bis zu einem bestimmten Zeitpunkt verfügbar sind und um das Vorwärtsblicken in die Zukunft zu verhindern. Das könnte nämlich zu ungenauen Vorhersagen führen.

Wavenet kreiert Sound Sample für Sample stückweise und iterativ. Es schaut sich vorherige Samples an und berechnet, viele weitere mögliche Samples. Dann sucht sich WaveNet das nächste Sample aus diesen Samples aus. Stück für Stück wird ein Audiosample generiert, welches über längere Zeit abspielbar ist.<sup>43</sup>

#### 4.2.2 GANSynth

Die Studie von Jesse Engel und Kollegen von Google AI stellt eine Methode vor, die Generative Adversarial Networks, kurz GANs für Audio Synthese benutzt. GANSynth verwendet Generatoren und Diskriminatoren, welche gegeneinander arbeiten, um realistische Audioausgaben zu erzeugen.

---

<sup>42</sup>Vgl. Purwins Hendrik et al.: Deep Learning for Audio Signal Processing, in: IEEE Journal of Selected Topics in Signal Processing, Bd. 13, Nr. 2, 2019, doi:10.1109/jstsp.2019.2908700 (abgerufen am: 02.12.2023), S. 1-9.

<sup>43</sup>Vgl. Van den Oord, Aaron, et al.: WaveNet: A Generative Model for Raw Audio, in: arXiv.org, 2016, <https://arxiv.org/abs/1609.03499> (abgerufen am: 02.12.2023), S. 1-5.

Während Wavenet sequentiell Sample für Sample arbeitet, generiert GANSynth Audiosignale in ihrer Gesamtheit. Wavenet tendiert dazu zeitintensiver als GANSynth zu sein, während GANSynth effizienter komplexe Klanglandschaften generieren kann. GANSynth wurde ursprünglich, wie WaveNet für die Erzeugung hochauflösender Bilder entwickelt.

GANSynth verarbeitet nicht direkt, rohe Audiosignale, sondern nutzt hauptsächlich Spektrogramme. Diese zeigen wie die ein Audiosignal sich über Zeit verändert.

Bei GANSynth wird ein Generator verwendet, welcher ein Autoencoder ist. Autoencoder sind neuronale Netzwerke, die Daten in eine niedrigere Dimension und dann wieder in die ursprüngliche Dimension zurücktransformieren. Er generiert einen Eingangsvektor. Dieser wird in ein Spektrogramm umgewandelt, welches die Frequenzkomponenten des generierten Audiosignals darstellt.

Der Diskriminator hingegen erfasst die globale Struktur des Audios. Der Diskriminator basiert auf einem spektralen Konverter, welcher in der Lage ist, langfristige Abhängigkeiten in Audio zu erkennen. Er bewertet die vom Generator erzeugten Spektrogramme, indem er sie mit echten Spektrogrammen vergleicht, welche aus realen Audiodaten abgeleitet wurden. Der Diskriminator versucht nun zu beurteilen, ob das generierte Spektrogramm realistisch aussieht oder nicht.

Nachdem ein realistisches Spektrogramm erzeugt wurden, kann es zurück in ein Audiosignal umgewandelt werden. Dies geschieht durch inverse Verfahren, die das Spektrogramm in ein hörbares Signal umwandeln. Durch diesen iterativen Prozess lernt GANSynth, immer genauere und realistischere Spektrogramme zu erzeugen, was wiederum zu einer verbesserten Qualität der generierten Audiosignale führt.<sup>44</sup>

### 4.2.3 SampleRNN

Die Studie „SampleRNN: An Unconditional End-to-End Neural Audio Generation Model von Soroush Mehre et al. befasst sich mit einem neuartigen Modell, welches auf der Erzeugung von einzelnen Audio-Samples basiert. Eine gängige Methode der Audiogenerierung ist es, das Audiosignal erst in spektrale oder handgefertigte Merkmale komprimiert und dann ein generatives Modell über diese Merkmale definiert. Das

---

<sup>44</sup>Vgl. Engel, Jesse et al.: GANSynth: Adversarial Neural Audio Synthesis, in: arXiv.org, 2019, Online: <https://arxiv.org/abs/1902.08710>, (abgerufen am: 02.12.2023), S. 1-8.

Problem dabei ist es allerdings, dass beim Dekomprimieren in Audiosignale, oft die Qualität verloren geht.

SampleRNN benutzt verschiedene Module, welche zu unterschiedlichen Geschwindigkeiten arbeiten, im Gegensatz zu WaveNet, welche eine einheitliche Geschwindigkeit für alle Berechnungen verwendet. Dieses Modell kann flexibler entscheiden, welche und wie viele Ressourcen es für verschiedene Aufgaben verwendet.

SampleRNN betrachtet jedes Sample einer Audiosequenz einzeln und berücksichtigt dabei, was in vorherigen Samples passiert ist. Jedes generierte Sample basiert auf allen vorherigen Samples. Es wird die Wahrscheinlichkeit jedes Samples in der Sequenz berechnet, also zum Beispiel eine Reihe von einzelnen Tönen. Dabei berücksichtigt das Modell jedes Mal alle vorherigen Samples in der Sequenz. Dieses Vorgehen erlaubt es SampleRNN, die natürliche Abfolge und Verbindung zwischen den Samples in einem Audiosignal zu verstehen und zu modellieren. Jedes Sample wird im Kontext der vorherigen Samples gesehen

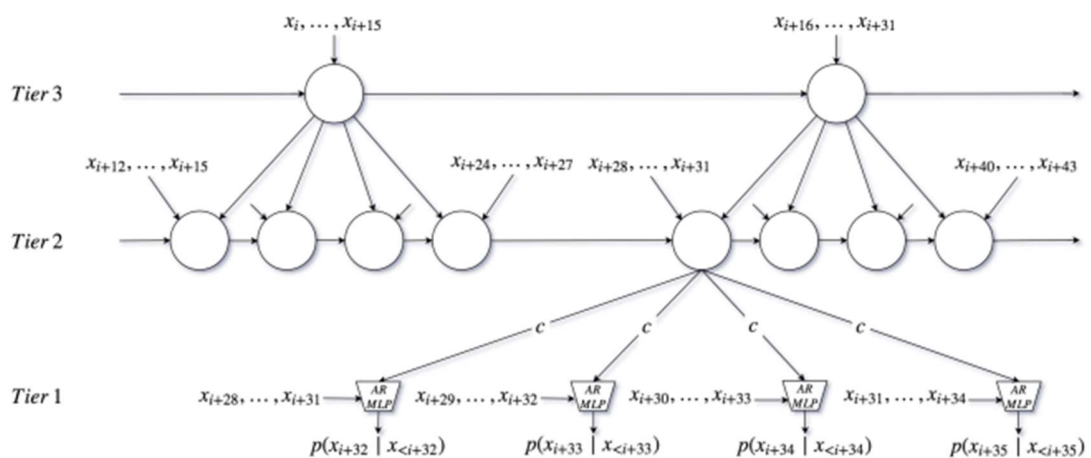


Abbildung 2: Zeigt einen Zeitschritt mit 3 Ebenen in der Hierarchie.

Oben wurden schon verschiedene Module erwähnt, welche jedes auf unterschiedlichen zeitlichen Auflösungen funktioniert und auf einem rekurrentem neuronalen Netzwerk basiert. Das unterste Modul, wie es auf der Abbildung 2 zu sehen ist, verarbeitet individuelle Samples, während höhere Module längere Zeitskalen und niedrigere zeitliche Auflösungen abdecken. Jedes Modul beeinflusst hier das darunterliegende. Das ermöglicht es dem Modell, Informationen über verschiedene zeitliche Ebenen hinweg zu integrieren. Das unterste Modul macht Vorhersagen auf der Sample Ebene und wird durch

die höherliegenden Module informiert. Alle Module lernen also gemeinsam, um die bestmögliche Vorhersage zu treffen.

Die Studie „SampleRNN“ hat einen bedeutenden Fortschritt in der Modellierung und Generierung von Audio-Wellenformen gemacht. Die hierarchischen, rekurrente Netzwerkstruktur ermöglicht eine kontextbezogene Analyse von Audiodaten. Gerade die Fähigkeit, unterschiedliche zeitliche Auflösung zu betrachten bietet einen großen Vorteil.<sup>45</sup>

Dieses Kapitel bot uns einen Einblick, in drei wegweisende Modelle, der KI-basierten Audiogenerierung. WaveNet, GANSynth und SampleRNN. Diese Modelle zeigen einen einzigartigen Ansatz zur Erzeugung von Audio durch künstliche Intelligenz und spiegeln den aktuellsten Stand der Technik in diesem Bereich wider. WaveNet, bekannt für seine Qualität in der Sprachsynthese, stellt einen Fortschritt in der Feinheit und Natürlichkeit generierter Audioinhalte dar. GANSynth wiederum basiert auf generative, gegnerische Netzwerke und zeichnet sich dadurch aus, hochwertige und vielfältige Audiodaten zu erzeugen. SampleRNN schließlich mit seinem Fokus auf rekurrenten neuronalen Netzen, bietet eine dynamische Herangehensweise für die Sequenzierung und Synthese von Audio.

Diese Modelle bilden nicht nur die Grundlage für zukünftige Entwicklungen, sie eröffnen auch neue Perspektiven für die Art und Weise, wie Interaktionen mit Sound und Musik gestalten. In den folgenden Kapiteln wird untersucht, wie diese Technologien in verschiedenen Bereichen der Musikproduktion Anwendung finden und welche kreativen und technischen Herausforderungen dabei zu bewältigen sind.

---

<sup>45</sup> Vgl. Mehri Soroush et al.: SampleRNN: An Unconditional End-to-End Neural Audio Generation Model, in: arXiv.org: 2016, Online: <https://arxiv.org/abs/1612.07837>, (abgerufen am: 02.12.2023) , S. 1-4.

## 5 Generative AI und Anwendungen in der Musikproduktion

Die Anwendungen generativer KI in der Musikproduktion werden beleuchtet. Hierbei nehmen wir funktionierende Modelle und prüfen ob sie in dem Kontext der Musikproduktion anwendbar sind und dem Produzenten in einer Weise hilfreich sein können.

Von der Erzeugung von Samples bis hin zur Unterstützung im Kompositionsprozess – die Möglichkeiten sind vielseitig-jedoch gilt es herauszufinden ob Künstliche Intelligenz jetzt schon auf einem Stand ist, um ein essenzieller Teil des Prozesses sein kann. Besonders Samplegenerierung oder Audiogenerierung allgemein sind hier interessant.

### 5.1 Dance Diffusion von HarmonAI

Wie wir bereits gelernt haben, gibt es verschiedene Modelle zur Audiogenerierung. Eines davon ist Dance Diffusion. Kreiert von einer Gemeinschaft namens HarmonAI, mit dem Ziel AI Tools zugänglich für Musikproduzenten zu machen. Diese Gemeinschaft ist Teil des AI Startups Stability AI, welches sich auf Bildproduktion fokussiert. Dance Diffusion fokussiert sich darauf zufällige Samples zu generieren, basierend auf trainierten Modelle. Diese Modelle sind entweder schon vortrainiert oder werden mit Hilfe eines Datensatzes trainiert.

Hier wird ein Diffusionsmodell verwendet, welches als erstes einen großen Datensatz braucht um Muster und Eigenschaften der Bilder oder vorliegenden Fall Sounds lernt. Dann wird ein zufälliges Rauschen zu dem Original Datensatz hinzugeneriert und mit dem Datensatz abgeglichen. So geht das Modell mehrere Iterationen durch. Das anfängliche Bild oder die Audio repräsentiert nur ein Rauschen. Mit jeder Iteration gelangt das Modell näher an den Original Datensatz indem es mit Hilfe des Abgleichs mit dem Orginal Rauschen entfernt.

Momentan gibt es ein paar öffentlich zugängliche Datensätze, welche zum Beispiel reine Piano Musik oder eine Kanadischen Gans repräsentieren. Die Einschränkung des Modells ist hier eben der Orginaldatensatz. Die zufällig Generierten Sounds werden immer auch

den Datensatz repräsentieren. Wenn der Datensatz also nur aus Piano Samples besteht, werden auch immer nur Piano Samples ausgegeben.<sup>46</sup>

Um Dance Diffusion nutzen zu können, muss der Nutzer einige Schritte folgen, bevor er an die zufällig generierten Samples kommt. Dance Diffusion wird über Google Colab zugänglich gemacht. Google Colab ermöglicht es dem Nutzer, Python Code über den Browser ausführen zu lassen, wofür eine Internetverbindung und ein Google Konto benötigt wird. Der Service benutzt das Jupyter Notebook Interface, welches dem Nutzer ermöglicht, Echtzeitprogrammierung auszuführen. Eines der Besonderheiten des Programms ist es, dass der Nutzer Zugriff auf eine GPU, kurz, Graphics Processing Unit und eine Tensor Processing Unit hat, welche von Google gestellt ist. Das heißt, der Nutzer ist unabhängig von seinen Hardwarebeschränkungen und standortunabhängig.<sup>47</sup>

Dance Diffusion ist mit Weights & Biases verbunden. Eine Onlineanwendung zur Hilfe, Experimente zu tracken oder Daten zu visualisieren. Es wird besonders in Verbindung mit Künstlicher Intelligenz Forschung benutzt und besitzt die Fähigkeit, Große Datenmengen zu analysieren und dem Forscher auf gewünschte Weise zu präsentieren. In unserem Fall wird es hauptsächlich dafür verwendet, die generierten Sounds zu downloaden.<sup>48</sup>

Wie oben schon erwähnt, müssen bei dem Programm Dance Diffusion mehrere Schritte getätigt werden, um Sounds zu generieren. Es kann ein vorgefertigter Datensatz genommen werden, doch in diesem Fall habe wurden Samples aus eigener Sound Bibliothek verwendet. Diese Sounds sind entweder selbst erschaffen oder aus Sample Packs. Die Sounds, welche ich für die Erstellung der Datenbank ausgewählt habe beschränken sich weitestgehend auf so genannte „Bass“ Sounds. Bassige, elektronisch erzeugte Sounds, welche ihren Hauptmerk an Charakteristik im tieferen Frequenzbereich haben. Die Höhen vieler meiner ausgewählten Sounds sind White Noise lastig und erzeugen mit dem vibrierenden Bass eine kratzige Charakteristik. Diese Mischung aus tiefen Vibrationen und kratzigen, verzerrten Obertönen und White Noise wird vom Hörer eher als aggressiv oder mächtig wahrgenommen. Die Sounds werden oft als Growls oder BassShots beschrieben. Mein Ziel war es, ebensolche Sounds zu rekreieren.

---

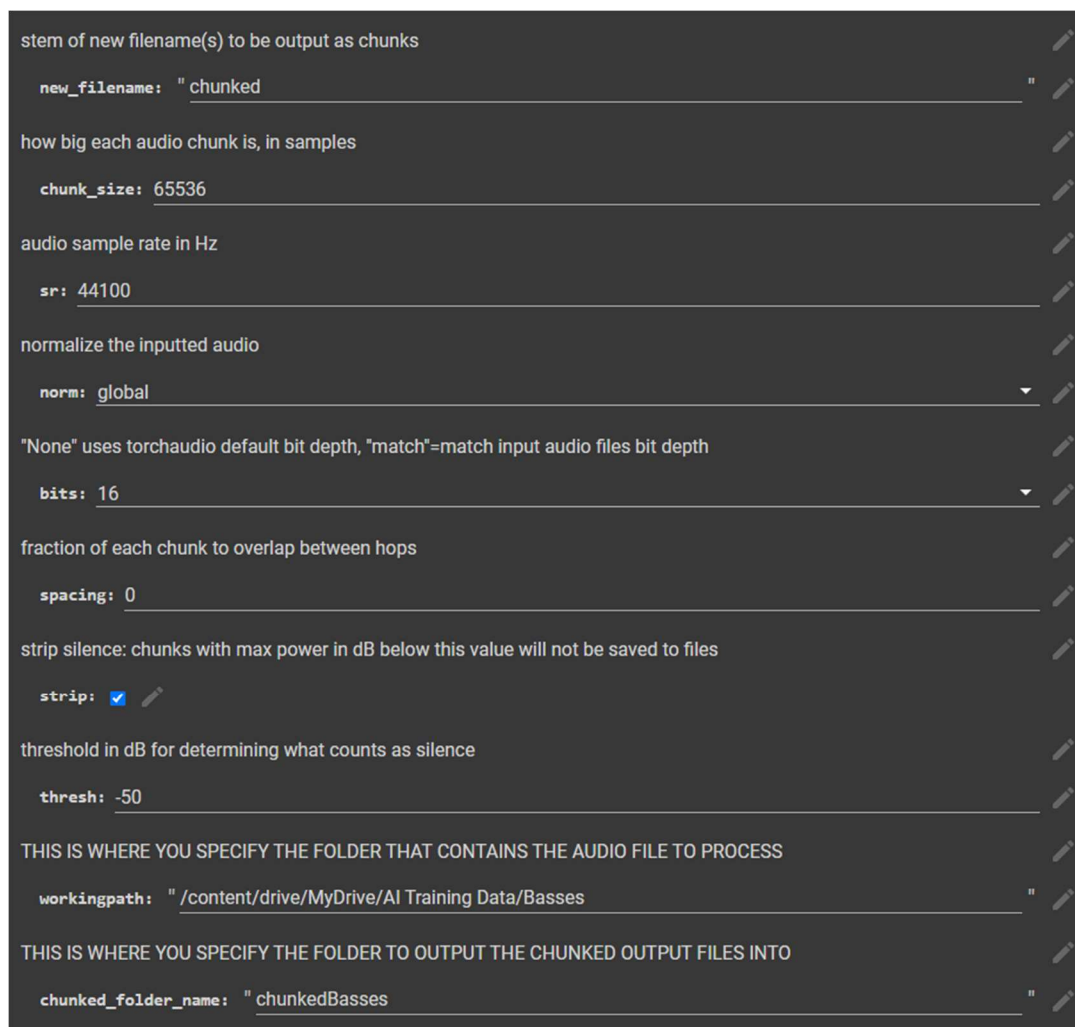
<sup>46</sup>Vgl. Pan Angelica: A Gentle Introduction to Dance in: Diffusion, Weights and Biases, 2023, [https://wandb.ai/wandb\\_gen/audio/reports/A-Gentle-Introduction-to-Dance-Diffusion--VmldzoyNjg1Mzky](https://wandb.ai/wandb_gen/audio/reports/A-Gentle-Introduction-to-Dance-Diffusion--VmldzoyNjg1Mzky), (abgerufen am: 10.12.2023).

<sup>47</sup>Vgl. Colaboratory: Willkommen bei Colab!, in: Colaboratory, 2023, [https://colab.research.google.com/#scrollTo=Nma\\_JWh-W-IF](https://colab.research.google.com/#scrollTo=Nma_JWh-W-IF) (abgerufen: 10.12.2023).

<sup>48</sup>Vgl. Weights & Biases, Weights & Biases for Enterprise, in: Weights & Biases, 2023, <https://wandb.ai/site/for-enterprise> (abgerufen am: 10.12.2023).

Der Prozess, in welchem man solche Sounds generiert, kann ein langwieriger Sound Design Prozess sein. Es wäre also wünschenswert als Produzent Zeit zu sparen und diese Aufgabe einer Künstlichen Intelligenz zu überlassen.

Der erste Schritt ist es, seine Sounds in einen Google Drive Ordner hochzuladen. Insgesamt wurden 489 Samples in den Ordner hochgeladen. Google Colab muss nun mit dem Drive Ordner verbunden werden. Ein Play-Button weist darauf hin, hier die Aktion starten zu können. Außerdem muss man auf der Chunking Seite ein Packet installieren. Dabei handelt es sich um ein Python Packet. Die im Google Drive hochgeladenen Samples sind bisher alle unterschiedlich lang. Der Dance Diffusion Fine Tuner sucht die Datenbank nach Sounds in einer bestimmten Länge ab. In unserem Fall beträgt die Sample Länge 65536 Samples. Daher müssen die Sounds erst einmal auf diese Länge gebracht werden. Dieser Prozess wird „Chunking“ genannt. Hierfür gibt es eine eigene Seite, in welcher man seinen Google Drive Ordner mit den Sounds verlinkt. Auf dieser muss man nun ein Eingabefeld ausfüllen, welches in der Abbildung 3 zu sehen ist.



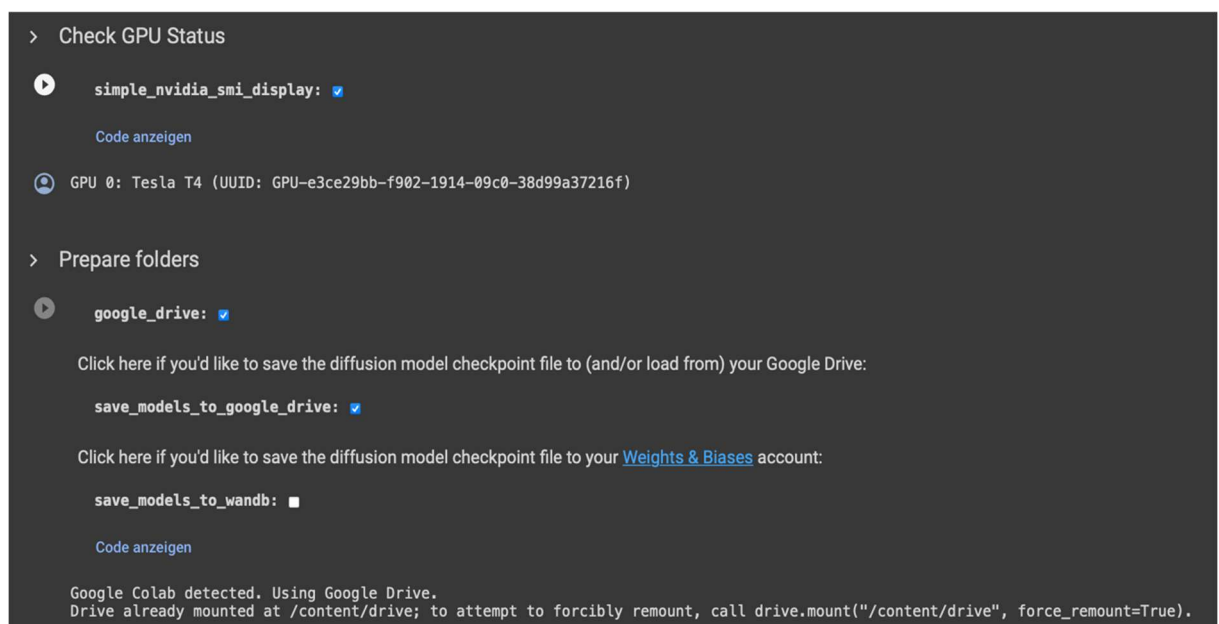
The image shows a dark-themed web form for audio chunking. It contains several input fields and dropdown menus, each with a pencil icon for editing. The fields are:

- stem of new filename(s) to be output as chunks: `new_filename: " chunked "`
- how big each audio chunk is, in samples: `chunk_size: 65536`
- audio sample rate in Hz: `sr: 44100`
- normalize the inputted audio: `norm: global` (dropdown menu)
- "None" uses torchaudio default bit depth, "match"=match input audio files bit depth: `bits: 16` (dropdown menu)
- fraction of each chunk to overlap between hops: `spacing: 0`
- strip silence: chunks with max power in dB below this value will not be saved to files: `strip:`
- threshold in dB for determining what counts as silence: `thresh: -50`
- THIS IS WHERE YOU SPECIFY THE FOLDER THAT CONTAINS THE AUDIO FILE TO PROCESS: `workingpath: "/content/drive/MyDrive/AI Training Data/Basses "`
- THIS IS WHERE YOU SPECIFY THE FOLDER TO OUTPUT THE CHUNKED OUTPUT FILES INTO: `chunked_folder_name: " chunkedBasses "`

Abbildung 3 Eingabefeld der Chunking Webpage

Hier wird der Name der ausgegebenen Datei festgelegt, die Sample Länge, Bit Tiefe, aber auch der Pfad, in welchem die Samples hinterlegt sind. Beim Chunking Prozess wird außerdem ein Ordner erstellt, indem die angepassten Sounds liegen. Hier rauf muss Dance Diffusion Zugriff haben, um den weiteren Schritt der eigentlichen Generierung vollführen zu können. Google Colab funktioniert, indem man nacheinander Befehle ausführt. Um die Dateien zu „chunken“ muss nun auf den Play-Button geklickt werden und das Programm führt den Befehl aus. Dies dauert mehrere Minuten. In unserem Fall bei mehreren verschiedenen Versuchen dauerte der Chunking Prozess meist 5-10 Minuten.<sup>49</sup>

Die nächsten Schritte sind ähnlich, dem Chunking Prozess. Auch hier müssen wir mit Hilfe von Google Colab Schritte nacheinander durchgehen um Daten zu generieren. Es geht nun weiter mit der Seite: „Finetune\_Dance\_Diffusion.ipynb“. Auf dieser müssen nun die Google GPU verbunden werden, wie in der Abbildung 4 dargestellt ist. Die anderen Schritte sehen ähnlich aus. Ist das getan erfolgt ein weiteres Mal die Verbindung mit Google Drive und dann wieder die Installation zweier Python-Pakete, welche für den Audiogenerierungsprozess vonnöten sind. Ist das getan muss sich Google Colab nun mit dem Weights & Biases Konto verbinden.



```
> Check GPU Status

▶ simple_nvidia_smi_display: ✓
  Code anzeigen

GPU 0: Tesla T4 (UUID: GPU-e3ce29bb-f902-1914-09c0-38d99a37216f)

> Prepare folders

▶ google_drive: ✓

Click here if you'd like to save the diffusion model checkpoint file to (and/or load from) your Google Drive:

save_models_to_google_drive: ✓

Click here if you'd like to save the diffusion model checkpoint file to your Weights & Biases account:

save_models_to_wandb: ■

  Code anzeigen

Google Colab detected. Using Google Drive.
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Abbildung 4 Schritte Check GPU Status und Prepare Folders auf der „Dance Diffusion finetune“ Seite

Der nächste Schritt ist ähnlich dem des letzten vom Chunking Prozess. Auch hier müssen wir wieder verschiedene Angaben machen, wie es in der Abbildung 5 zu sehen ist. Es wird der Name eingegeben, den Google Drive Pfad zur Trainings directory und der

<sup>49</sup>Vgl. Chunking, in: Google Colab, 2023, <https://colab.research.google.com/drive/1wCjjFir4vkWSTU3DwVgCvi5qGtaNnGb> (abgerufen am: 15.12.2023)



Checkpoint Pfad. Dieser dient als Grundlage für das Modell. In der Checkpoint Datei ist ein schon fertiger Datensatz. Mit diesem wird der vorliegende Datensatz abgeglichen und so entsteht in mehreren Iterationen die Datei. Außerdem ist in dem Feld definiert nach wie vielen Trainings Steps Demos vollführt werden. Es wird festgelegt, nach wie vielen Schritten Zwischenergebnisse gespeichert werden. Diese Zwischenergebnisse sind auf der Website von Weights & Biases abzurufen. In unserem Fall sind es 250 Stufen oder Schritte. Der Trainingsprozess dauert viele Minuten beziehungsweise auch Stunden.

Name for the finetune project, used as the W&B project name, as well as the directory for the saved checkpoints

NAME: `dd-drums-finetune`

Path to the directory of audio data to use for fine-tuning

TRAINING\_DIR: `/content/drive/MyDrive/Audio/Drums`

Path to the checkpoint to fine-tune

CKPT\_PATH: `/content/drive/MyDrive/AI/models/jmann-small-190k.ckpt`

Directory path for saving the fine-tuned outputs

OUTPUT\_DIR: `/content/drive/MyDrive/AI/models/DanceDiffusion/finetune`

Number of training steps between demos

DEMO\_EVERY: `250`

Number of training steps between saving model checkpoints

CHECKPOINT\_EVERY: `500`

Sample rate to train at

SAMPLE\_RATE: `48000`

Number of audio samples per training sample

SAMPLE\_SIZE: `65536`

If true, the audio samples provided will be randomly cropped to SAMPLE\_SIZE samples  
Turn off if you want to ensure the training data always starts at the beginning of the audio files (good for things like drum one-shots)

RANDOM\_CROP:

Batch size to fine-tune (make it as high as it can go for your GPU)

BATCH\_SIZE: `2`

Accumulate gradients over n batches, useful for training on one GPU.  
Effective batch size is BATCH\_SIZE \* ACCUM\_BATCHES.  
Also increases the time between demos and saved checkpoints

ACCUM\_BATCHES: `4`

[Code anzeigen](#)

Abbildung 5 Auszufüllendes Feld für Trainings Schritt

Auf der Weights & Biases Website können wir nun live verfolgen was das Modell bereits geschafft hat. Alle Durchgänge die bereits vollführt wurden, sind hier nachvollziehbar. Es gibt ein Fenster, bei welchen die Zwischenergebnisse hinterlegt sind. Mit einem Regler scrollt man nun die Ergebnisse und durch. In unserem Fall stellen wir fest, dass anfängliche Iterationen noch sehr Wirr sind. Je nach dem mit welchem Modell unser Datensatz trainiert wurde, klingen zumindest die Anfänge anders. Man hört zum Beispiel bei anfänglichen Iterationen noch das Grund Modell heraus. Zum Beispiel konnte man bei einem Modell Pianoklänge ausmachen, da dieses eben nur Pianoklänge beinhaltete.

Oder Stücke von anderen Songs, da dieses Modell nur Songs beinhaltet. Je nach dem welches Modell wir verwendeten, klang auch unser Endergebnis anders.

Das Ergebnis ist ein 24 sekundiger Audioclip, welcher sich in Weights & Biases abspielen lässt. Diesen kann man zudem downloaden und wird als .wav Datei gespeichert. Zu hören sind unterschiedlich lange Töne, gemischt mit ruhigeren Passagen, welche fast wie Nachhall klingen. Die Töne spiegeln teilweise Sounds aus unserer zusammengestellten Bibliothek wieder. Der Charakter der Sounds ähnelt sich, man erkennt sie wieder, wenn unsere Bibliothek relativ ähnliche Sounds beinhaltet. Hier muss erwähnt werden, dass die ausgegebenen Sounds qualitativ hochwertig sind und in Tiefen sowie den Höhen den Charakter der von uns vorgegebenen Sounds widerspiegeln. Allerdings gibt es einige Passagen, bei welchen die ausgegebenen Sounds recht kurz und charakterlos sind. Auch wäre es wäre wünschenswert, hätten die Sounds den Ablauf, unserer Sounds, welche wir vorgaben.<sup>50</sup>

Beispielsweise haben wir viele bassige Sounds verwendet, auch Sounds, die so genannte Brass Shots widerspiegeln. Diese werden in Filmmusik zum Beispiel viel verwendet, um mächtig wirkende Stimmung einzuleiten. Sie bestehen aus kurzen, kraftvollen Klängen wie Trompeten, Posaunen und Hörnern und erzeugen einen aufmerksamkeiterregenden Effekt.<sup>51</sup>

Es sind also Teilelemente dieser Sounds zu hören, der Ablauf der Sounds ist allerdings nur wenig oft zu hören. Ob dieses Ergebnis nun zufriedenstellend ist, liegt im Auge des Erzeugers und demjenigen, der die Sounds im Verlauf der Musikproduktion verwendet. Das Ergebnis, welches die Künstliche Intelligenz erzeugt gleicht einer Sound Design Session, bei welcher ein Synthesizer oder Sample als Grundlage gewählt wird und im Lauf der Session durch Effekte und Parameterregelung verändert wird. Hier ist es üblich die Sitzung komplett aufzunehmen, sodass man Sounds, welche eventuell zufällig, während des Prozesses entstehen nicht verliert. Das heißt, es gibt hier viele einzelne Sounds in einer Langen Aufnahme, bei welcher der Großteil der Sounds unbrauchbar sind. Genauso lässt sich das Ergebnis unserer Generierung beschreiben. Einzelne Sounds im 24 Sekunden Clip lassen sich durchaus verwenden. Besonders kann man die entstandenen Sounds bei weiteren Sound Design Sitzungen verwenden. Da die Sounds

---

<sup>50</sup> Vgl. Weights and Biases: in wandDB.ai, 2023, <https://wandb.ai/home> (abgerufen am: 15.12.2023).

<sup>51</sup>Vgl. Daub Adrian: "BRAAAM!": The Sound that Invaded the Hollywood Soundtrack, in Longreads, 2016, <https://longreads.com/2016/12/08/braaam-inception-hollywood-soundtracks/> (abgerufen am: 16.12.2023)

und Samples zufällig generiert werden, gibt es durchaus interessante Ergebnisse. Diese lassen sich gut weiterverwenden.

### 5.1.1 Fazit

Ist die Dance Diffusion Anwendung von OpenAI nun nutzbar in einem Musikproduktionskontext? Es kommt hier immer drauf an, was genau das Ziel ist, mit der Generierung. Will man fertige Sounds, welche man einfach in seine Digital Audio Workstation lädt und dort direkt weitermachen kann, so ist das Modell eher weniger nützlich. Für Inspiration und weitere Sound Design Sessions sind die ausgegebenen Audiofiles allerdings sehr nützlich. Der Charakter der Sounds, welche anfänglich in der Trainings-Library definiert wurden, ist eindeutig wiederzuerkennen. Das ist ein großer Vorteil, denn so kann das Modell seine eigene Soundlibrary an Growls liefern und das Ergebnis weicht nicht unbedingt vom Stil des Musikproduzenten ab. Dies ist ein wichtiges Kriterium, denn jeder Dubstep Produzent hat seinen eigenen Sound. Der große Nachteil dieses Modells ist allerdings die Bedienung durch Google Colab. Auch wenn die Schritte alle nachvollziehbar sind, so dauert das Setup des Ganzen recht lange. Bei den ersten Malen dauerte das Setup ca. eine Stunde und das Generieren der Sounds 2 bis 3 Stunden. Außerdem werden getroffene Einstellungen, also zum Beispiel angegebene Ordner Pfade unseres Google Drive's nicht gespeichert. Es muss jedenr Schritt erneut gemacht werden, was viel Zeit kostet. Dieses Modell ist also interessant, weil es gute Ergebnisse erzielt. In einem Produktionskontext ist es allerdings unbrauchbar, da die Prozedur zu lange dauert. Hier kann ein Musikproduzent genauso die Sounds selber herstellen, bei welchem das Ergebnis außerdem noch genauer ist. Dies bietet mehr Kontrolle und braucht so lange Zeit, bis der gewünschte Sound vorhanden ist. Wäre die Benutzer Oberfläche einfach gestaltet, beispielsweise in einem VST, bei welchem nicht von der Chunking Seite auf die Dance Diffusion Seite gewechselt werden muss und es die Einstellungen speichert, welche vorher bereits vorgenommen wurden, dann wäre das Dance Diffusion Modell durchaus nützlich.

## 5.2 Audiocraft - AUDIOGEN

AudioCraft ist eine PyTorch-Bibliothek, die speziell für die Forschung im Bereich der Audioerzeugung mittels Deep Learning entwickelt wurde. Eine PyTorch-Bibliothek ist eine Sammlung von Modulen und Funktionen, die speziell für die Verwendung mit PyTorch, einer Open-Source-Maschinlernbibliothek, entwickelt wurden. Es handelt sich um ein Open-Source-Projekt von Meta AI, das sowohl für Musik- als auch für Soundeffektgenerierung eingesetzt werden kann.<sup>52</sup> Dieses Tool ermöglicht es über einfache Texteingaben Musik und Soundeffekte zu generieren. Es wird speziell das AudioGen Modell beleuchtet, welches sich auf die Erzeugung von Audiomaterial konzentriert. Es soll also Samples generieren, welche für die Musikproduktion verwendet werden kann.<sup>53</sup>

AUDIOGEN ist ein autoregressives Audio Generations Tool mit Sprachmodel-Decoder. Die Besonderheit ist, dass das Modell auf Textprompts reagiert und dementsprechend Audio generiert. Das Modell basiert auch nicht unbedingt auf einer Datenbank, sondern einem Modell, welches mit großen Datenmengen trainiert wird. Hier wird das autoregressive Modell auf Muster und Beziehungen in den Daten trainiert. Da es ein textbasiertes Modell ist, werden hier auch Beschreibungen zu den Audiofiles mittrainiert. Es kreiert Audio dynamisch und in Echtzeit, basierend auf dem im Training gelernten Muster. Das Modell nutzt beim Trainieren drei Komponenten. Ein Encoder Netzwerk, welches das Audiosignal für ein Netzwerk verständlich macht. Es wandelt das Audiosignal also in Sprache um, welche die KI beziehungsweise das Netzwerk verstehen und auswerten kann. Diese Darstellung ist für den Menschen nicht mehr interpretierbar. Dann wird das ursprüngliche Audiosignal durch die Quantisierungsschicht weiter komprimiert. Das Decoder Netzwerk, also der dritte Schritt, stellt aus der komprimierten Datei wieder ein Zeitsignal her. Das System wird so trainiert, dass es das Originalsignal bei der Rekonstruktion eines Audiosignals nicht verliert. Wenn jemand in die Eingabe „Pistolenschuss“ schreibt, muss auch ein Pistolenschuss zu hören sein und keine ferne Repräsentation eines Pistolenschusses.

---

<sup>52</sup> Vgl. Audio Craft: Projekt-Beschreibung AudioCraft, in Python Package Index, 2023, <https://pypi.org/project/audiocraft/> (abgerufen am: 18.12.2023).

<sup>53</sup> Vgl. Chappel Roger: How to Install Audiocraft Locally – Meta’s FREE and Open AI Music Gen, in AITooltip, 2023, <https://aitooltip.com/how-to-install-audiocraft-locally-metas-free-and-open-ai-music-gen/#:~:text=1%20September%202023%20,for%20both%20Musicgen%20and%20Audiogen> (abgerufen am: 18.12.2023)

Ähnlich, wie beim Audio, wandelt das Programm Text in eine von Künstlicher Intelligenz verständliche Sprache um.

AUDIOGEN ist ein autoregressives texttextbasiertes Audio Generations Tool. Der Nutzer gibt Dinge wie Art des Klangs, Stimmungen oder spezifische Geräusche ein. Das System verwendet dann diese Textinformationen, um die Generierung oder Modifikation des Audios zu steuern. In einem Modell wie AUDIOGEN wird der Text verwendet, um das neuronale Netzwerk anzuleiten, wie es das Audio basierend auf den eingegebenen Textanweisungen erzeugen soll. Während des Trainings hat man dem Modell beigebracht, die Audiobeschreibungen mit dem Audio zu assoziieren. Beziehungsweise gab es einige Sound Datenbanken, bei welchen die Beschreibungen schon hinterlegt waren. Stimmungen, Muster und Eigenschaften aus anderen eventuell ähnlichen Audios und Datenbanken können miteinander assoziiert werden, da die Beschreibung zueinander passt. Sie werden miteinander durch die textbasierte Beschreibung assoziiert und kombiniert. Diese Merkmale oder Eigenschaften können Tonhöhe, Tempo, Rhythmus oder Harmonie sein.

Das Audio, welches aus dem Prozess entsteht, soll so gut wie möglich der Texteingabe entsprechen.<sup>54</sup>

Um dieses Programm nutzbar zu machen, müssen einige Umwege gegangen werden. Ich bin einen einfacheren Weg gegangen, bei welchem man ein Hilfsprogramm installiert, um AudioCraft dann mit Oberfläche nutzen zu können. Dieses Programm heißt Pinokio. Es handelt sich um ein Programm, welches AI Applikationen installiert und nutzen lässt. Oft muss man nämlich AI fokussierte Programme selbst installieren, was für einen normalen Nutzer oft sehr umständlich und schlechter erreichbar sein kann. Audiocraft wird zum Beispiel nutzbar gemacht, indem man eine Reihe an Python Bibliotheken installieren muss und selber an bestimmten Stellen den Code manipulieren muss. Dort stoßen Nutzer, welche nichts mit Programmieren zu tun haben schnell an ihre Grenzen. Man lädt nun Pionokio auf der Website runter und installiert das Programm. Man landet nun auf der Hauptseite von Pinokio. Dort gibt es die Möglichkeiten, wie in der Abbildung 6 zu sehen ist, aus mehreren AI-Anwendungen, welche sich durch Pinokio installieren lassen.

---

<sup>54</sup>Kreuk Felix, et al.: AudioGen: Textually Guided Audio Generation, in: arxiv.org, 2023, <https://arxiv.org/abs/2209.15352> (abgerufen am: 18.12.2023), S. 3-5.

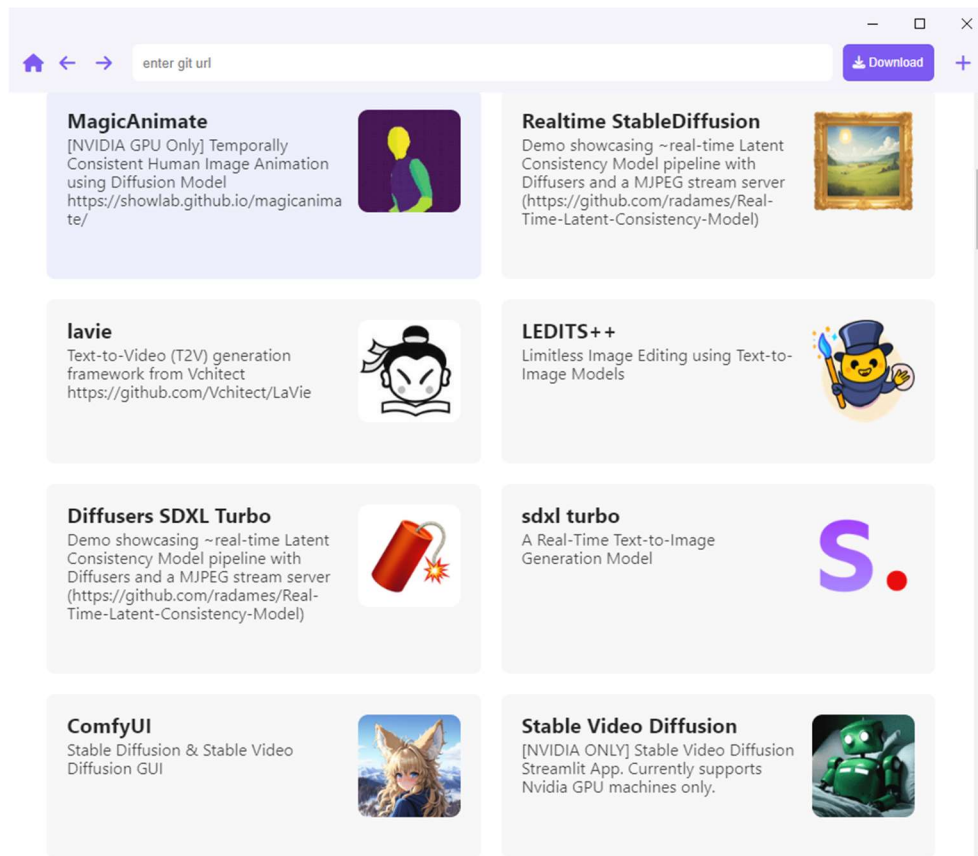


Abbildung 6: Pinokio Discover Page

Würde AudioCraft manuell installiert, wären folgende Schritte zu befolgen. Zuerst muss sichergestellt werden, dass Python auf dem Computer installiert ist. AudioCraft erfordert Python 1.9. Da AudioCraft wie oben erwähnt, eine PyTorch-Bibliothek ist, muss nun PyTorch von der offiziellen Website heruntergeladen und installiert werden. Nun muss mit Windows Power Shell der pip-Befehl ausgeführt werden, um AudioCraft zu installieren. Der Befehl lautet: „python -m pip install -U audiocraft“. Alle Befehle und eine genaue Anleitung der Schritte befinden sich auf der Github Seite von AudioCraft.<sup>55</sup>

Danach wird das Multimedia Framework namens FFmpeg installiert. FFmpeg ist ein Multimedia-Softwareprojekt, welches freie Computerprogramme und Programmbibliotheken, die digitales Video- und Audiomaterial aufnehmen, konvertieren, senden und weitere Funktionen.<sup>56</sup> All diese Schritte erledigt Pinokio für uns durch einen Mausklick. Zuerst wählen wir die Seite AudioGradio auf der Discover Page aus und klickt den Download Button. Ist der Download abgeschlossen klicken wir nun auf „Install“ und

<sup>55</sup>Audiocraft: AudioCraft, in: Meta Platforms, 2023, <https://github.com/facebookresearch/audiocraft> (abgerufen am: 18.12.2023).

<sup>56</sup>Vgl. About FFmpeg, in: ffmpeg.org, 2023, <https://ffmpeg.org/about.html> (abgerufen am: 20.12.2023).

ein neues Fenster öffnet sich, wie es auf der Abbildung 8 zu sehen ist.

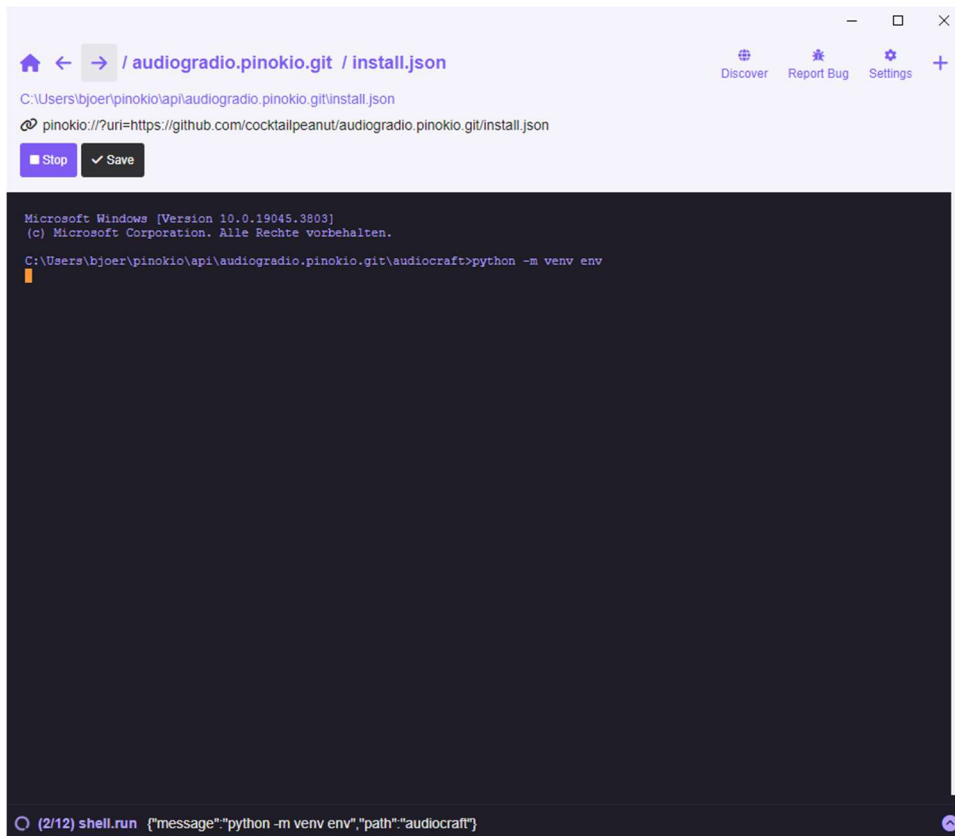


Abbildung 7: Pinokio Installationsvorgang

Hier geht Pinokio alle einzelnen Schritte, wie sie auf der Github Seite zu finden sind eigens durch und installiert das Programm. Wenn die Installation abgeschlossen ist, gibt es nun die Wahl zwischen AudioGen und MusicGen. Beides sind Audiocraft Programme. In dieser Bachelorarbeit wird allerdings Sound Design behandelt, daher wählen wir nun AudioGen aus und starten das Programm. Es öffnet sich nun ein Fenster in dem Standard Internet Browser. AudioGen ist jetzt mit einer Oberfläche nutzbar. Manuell installiert, kann der Parameter nur im Code selbst geändert werden. Hier gibt es Buttons, Regler und Eingabefelder wie auf Abbildung 8 zu sehen ist.

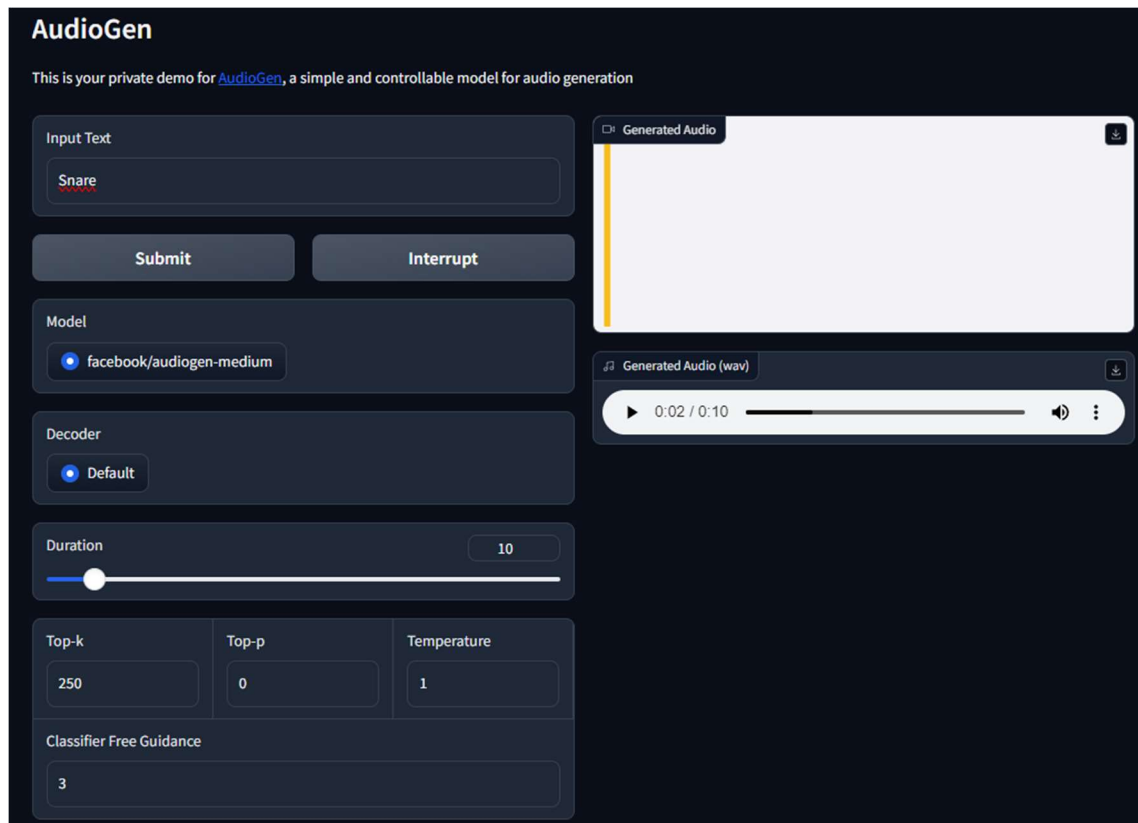


Abbildung 8 AudioGen Benutzeroberfläche

Im Eingabefeld namens „Input Text“ beschreiben wir den Sound, welchen wir generieren wollen. Mit Schieberegler kann ausgewählt werden, wie lang der Sound sein soll. „Temperature“ meint Variabilität. Je höher die „Temperature“, desto variabler und einzigartiger sind die Sounds. Das AI-Modell sagt die Wahrscheinlichkeiten voraus, welches Element am wahrscheinlichsten als nächstes kommt. Um einzustellen, wie viele der möglichen Elemente von Modell als nächstes beachtet werden, gibt es die Möglichkeit, bei „Top-k“ eine Zahl einzutragen. Setzen wir zum Beispiel die Zahl auf 10 im Feld, so beachtet das Modell, je Schicht immer nur die 10 wahrscheinlichen Elemente. Das Modell entscheidet sich dann zufällig für einen dieser 10 Top-Kandidaten. Ein kleinerer „k“-Wert führt zu einem recht absehbaren Ergebnis. Ist der „k“-Wert größer, so wird das Ergebnis variabler.

Anders als bei „Top - k“ die wahrscheinlichsten Elemente auszuwählen, betrachtet „Top-p“ Sampling die gesamte Liste möglicher nächster Elemente und summiert die Wahrscheinlichkeiten. Der Schwellenwert „p“ ist eine Wahrscheinlichkeit, wie zum Beispiel 0,9% oder 90%. Das Modell addiert nun die Wahrscheinlichkeiten der wahrscheinlichsten Elemente, bis die kumulative Wahrscheinlichkeit diesen Schwellenwert überschreitet. Heißt, ein festgelegter Schwellenwert wird erreicht und nun werden Elemente ausgewählt, deren kumulierte Wahrscheinlichkeit zusammen den



Schwellenwert „p“ überschreiten. „Top-p“ Sampling sucht also die besten Elemente raus, bis ein bestimmter Schwellenwert erreicht ist. Der Topf an Elementen ist nun gut genug, andere werden rausgeworfen.

### 5.2.1 Prompts

Da Audiocraft den Input durch Text braucht, müssen wir eine Art System finden, welche Prompts nun benutzt werden können, zu sehen, wie sich die Ergebnisse der Generierung verändern.

Bei den Tests mit der Künstlichen Intelligenz wurden hauptsächlich Snares erstellt. Snares zu finden und zu erstellen und verändern, ist ein essenzieller Teil der Musikproduktion, speziell im Subgenre Dubstep. Könnte man hier einfach eingeben, wie man die Snare oder den Grownl denn haben will, wäre das ein großer Vorteil. Snares zu finden ist ein zeitaufwändiger und manchmal auch ein kostspieliger Vorgang, da oftmals Samplepacks gekauft werden müssen, um qualitative Snares zu finden. Eine schnelle, qualitativ hochwertige, präzise Generierung auf Knopfdruck wäre also zeit- und kosteneinsparend.

Folgende Struktur eines idealen Prompts wurde gewählt:

Instrument oder Art des Sounds, Eigenschaft, Musikgenre, Note.

Beispiel:

Snare, heavy, Dubstep, Key in D

Ob diese Art von Prompt bei dem System nun aufgenommen wird, stellt sich aufgrund des folgenden Tests heraus. Hierfür Testen wird erstmal getestet, ob das Programm eine normale Snare generieren kann. Es wird fortgefahren, indem immer mehr Beschreibungen, den Prompts hinzugefügt werden. Der erste Prompt wird lauten: „Snare“, dann „Snare, heavy“, dann „Snare, heavy, Dubstep“ und so weiter, um zu testen, ob das Modell überhaupt auf unsere Angaben reagiert.

### 5.2.2 Bedingungen

Zuerst muss klargestellt werden, welche Ergebnisse bei unserem Test, als gut oder schlecht bewertet werden können. Hierfür müssen wir die grundlegenden Eigenschaften einer Snare betrachten. Perkussive Instrumente, wie eine Snare bieten dem Hörer regelmäßige, klare Sounds, um dem Rhythmus eines Songs voran zu treiben. Sind Body, Drähte und Transient eindeutig zu hören, ist das schonmal ein Grundgerüst, welches für

Musikproduktion Anwendung hat. Natürlich klingen Snares in den verschiedenen Musikrichtungen anders. Es wird das Grundgerüst der generierten Snare betrachtet, um festzustellen, ob die Künstliche Intelligenz überhaupt in der Lage ist, grundlegend Snares zu erzeugen oder ob das Modell allein schon damit Probleme hat. Daraufhin gilt es zu testen, ob das Modell Genregerecht Snares erstellen kann.

Außerdem wird untersucht, ob das Modell in der Lage dazu ist, Dubstep Growls zu erzeugen. Diese müssen bass-lastig, vokal und aggressiv klingen. Der Sub Bass muss zu hören sein, der Sound muss sich über Zeit bewegen, er muss einen bestimmten Charakter haben, die Höhen müssen klar zu hören sein und denn Bass unterstützen.

Das Modell kann erfolgreich zur Hilfe des Musikproduzenten genutzt werden, wenn es spezifische, den Wünschen des Produzenten gerechte Samples generieren kann. Außerdem ist Benutzerfreundlichkeit hier wichtig. Musikproduktion ist zwar etwas kompliziertes, doch muss effizient gearbeitet werden können. Ist das nicht der Fall, so ist das Modell nicht nützlich für den Produzenten.

### 5.2.3 Untersuchung Snare

Um zu testen, wie sich das Modell verhält, wenn man einen direkten Befehl gibt, tippen wir nun „Snare“ in das Feld ein unter in der Abbildung 9 gezeigten Einstellungen, welchen den Einstellungen entsprechen die von vornherein gegeben sind.

Später wird untersucht, wie sich das Ergebnis unter anderen Einstellungen verändert.

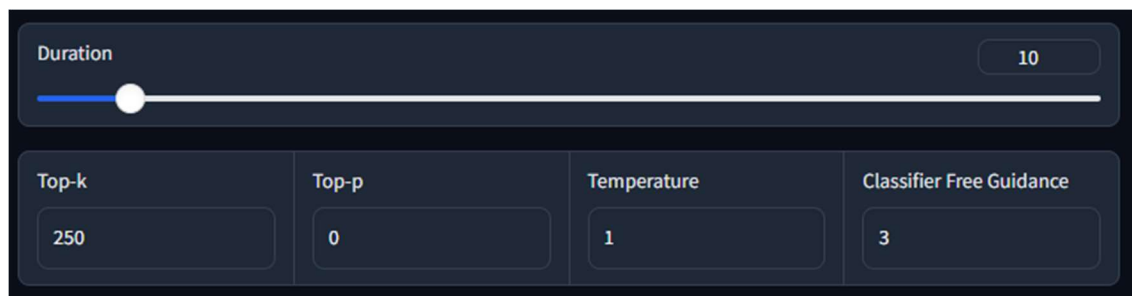


Abbildung 9 AudioGen Default Einstellungen

Der allererste Test gibt uns bei einer Samplelänge von 10 Sekunden 2 Snare aus. Diese sind ca. eine Sekunde voneinander getrennt doch klingen beide sehr unterschiedlich. Im ersten Sound, welcher eine Snare repräsentieren soll, ist ein tonaler transient zu hören und danach ein Nachhall einer, oder dieser Snare. Der Sound ist sehr kurz und hat nicht viel Höhen. Es sind außerdem kaum die an der Snare unten angebrachten Drähte zu hören. Der Snare Sound ist also nicht komplett. Der zweite Sound hat ein wenig mehr Audioinformationen. Er hört sich mehr an wie eine Snare als der erste Sound. Er hat

keinen Nachhall, wie es beim ersten Sound der Fall ist. Der Transient ist wenig klar zu hören, und ist damit als Snare Sample nicht sehr effektiv, doch sind die Drähte und der Body gut zu hören, was dem Sound seine Hauptcharakteristika einer Snare gibt. Beide Sounds können also als Snare identifiziert werden. Sie haben klanglich die Charakteristika einer Snare, doch scheint es auch, als seien beide Sounds unvollständig. Diese Beobachtung führte zur Überlegung, die beiden Snares übereinanderzulegen, um zu schauen, ob ein Fehler in der Generierung vorlag. Der Fehler wäre in dem Fall, dass die Künstliche Intelligenz zwei zusammengehörende Teile einer einzelnen Snare, unabhängig voneinander rausgerendert hätte. Tatsächlich war das Ergebnis des Übereinanderlegens positiv. Die beiden Snares haben die gleiche Tonhöhe, was durch einen EQ überprüft wurde. Dort, wo normalerweise tonal die Tonhöhe einer Snare bestimmt wird, also im Bereich des Body's, wurden uns durch einen parametrischen Equalizer ein Peak bei 230 Hz angezeigt, was ungefähr der Tonhöhe eines A#'s entspricht. Man kann also die Snares übereinanderlegen, ohne dass sie die Tonhöhen unterscheiden und eine Dissonanz entsteht. Tatsächlich werden die Snares hiermit brauchbarer, wenn auch nicht professionell brauchbar. Denn beide Snares weisen eine gewisse Undeutlichkeit auf. Man vernimmt eine Art Rauschen, spielt man den Sound ab.

Es gab noch Versuche, in welchen nur das Wort „Snare“ in das „Input Text“ Textfeld eingegeben wurde. Die Generierungen entsprachen weiterhin den Default Vorgaben, wie oben erwähnt. Die Ergebnisse waren doch sehr unterschiedlich, fast zufällig. Allerdings kann man sagen, dass jede Generierung eine Snare hervorbrachte. Daraus können wir zumindest schließen, dass das Programm das Wort kennt und die grundlegendsten Charakteristika einer Snare wiedergeben kann. Dies ist allerdings noch nicht nützlich für den Musikproduzenten. Das Modell wird erst verwendbar, wenn der Künstlichen Intelligenz beschrieben werden kann, was gefordert ist und diese dann ein Ergebnis liefert, welches dem Input Text entspricht.

Um zu testen, wie AudioGen auf weitere Beschreibung einer Snare reagiert, geben wir nun „Snare, snappy and short“ in das Input Textfeld ein. Der erste Versuch gab uns eine Audiodatei, in welcher knapp eine Sekunde lang, verschiedene Geräusche zu hören sind. Diese repräsentierten keine Snare sondern eine Art mechanisches Rascheln. Das Wort Snare hat mehrere Bedeutungen. Es wird natürlich für die Schlagzeugtrommel verwendet, aber auch für eine Falle beim Jagen von Tieren. Es ist also möglich, dass das Modell, diese Snare meint und eine Snare in Bezug auf Jagd darzustellen versucht. Dabei scheint das Modell diese Eingabe durch mehrere metallene Gegenstände, welche

aneinanderstoßen, zu interpretieren. Bei fünf weiteren Generierungen war dies der gleiche Fall. Mit dem Prompt konnte die Künstliche Intelligenz also nicht identifizieren, ob es sich um eine musikalische Snare oder eine Falle für die Jagd handelt. Der Befehl „Snare, short and snappy“ war also nicht verwendbar für eine musikalische Snare.

Daher lautete der nächste Prompt nur: „Snare, short“. Die Ergebnisse hier waren besser. Bis auf ein Sample von fünf waren deutlich musikalische Snares zu hören. Was allerdings auffällt ist, dass die Länge der Snares nicht deutlich anders war, als bei unseren ersten Generierungen, bei welchen wir nur den „Snare“ Prompt benutzten.

Es wird nun der „Duration“ Regler nun auf eine Sekunde gestellt. Dies führte allerdings dazu, dass das Ergebnis der nächsten fünf Generierungen sehr unterschiedlich war. Bei nur einer Generierung war auch eine Snare zu hören. Bei den anderen vier waren nur unterschiedlichste Geräusche zu vernehmen. Beispielsweise ein Rauschen, ein Schlag, welcher sich anhörte, wie ein Tennisschlägerschlag, ein kurzer Laut, welcher sich anhörte als würde man etwas mit Luftdruck durch ein Plastikrohr schießen. Keines dieser Laute kam von den Charakteristika an eine Snare ran. Das heißt also, dass wenn man den Duration Regler auf bestimmte Sekundenlängen stellt, sich das Ergebnis grundlegend verändert. Dies überprüften wir, indem wir den Regler auf 2 und auf 3 Sekunden stellten, wo man feststellte, dass das Modell bessere Ergebnisse lieferte. Bei der Grundeinstellung von 10 Sekunden waren aus den Tests, die meisten Snares entstanden, woraufhin für den Rest des Experiments dieser Regler nicht mehr verändert wurde.

Wie man beim „Snare, short“ Prompt feststellte, veränderte sich die Länge zum „Snare“ Prompt kaum oder nicht erfassbar. Daraufhin wurde der Prompt „Snare, very short“ ausprobiert. Um herauszufinden, ob man mit der Länge der Generierten Snare spielen kann und tatsächlich wurden hier nun kürzere Snares generiert, wenn auch nicht bedeutend. Weiterhin wurde getestet, ob die eingegebenen Längenangaben auch wirklich einen Effekt auf das Endergebnis haben. Daher wechselt man zum gegenteiligen Prompt: „Snare, very long“. Es gab keine sinnvollen Auswirkungen auf das Ergebnis. Eher wurde das Ergebnis unberechenbarer. Mal generierte das Modell kurze Snares, mal Drumrolls, mal Snares in normaler Länge. Der Grund hierfür ist wahrscheinlich, dass dem Modell Trainingsdaten zur Verfügung stand, welche Snares nur mit den Wörtern „Snare“ oder „short“ beschrieben. Snares werden nicht unbedingt als lang bezeichnet. Einzelne Snare Sounds sind nun mal relativ kurz. Eine lange Snare kann nur generiert werden, wenn wir die top-k Einstellung verändern. Zu diesem Zeitpunkt wusste man das allerdings noch

nicht. Im Laufe des Experiments wird klar, dass die Grundeinstellungen nicht unbedingt dem entsprechen, was man als Musikproduzent braucht.

Kurze Snares zu generieren geht also, wenn auch unzuverlässig unter den vorhandenen Einstellungen. Immer wieder passiert es, dass das Modell Snares generiert, welche normal langezogen sind. Ein kleiner Teil davon ist tatsächlich kurz. Von 20 generierten Snares, waren 16 tatsächlich klar zu hörende Snares. Der Rest waren zufällige Geräusche, welche nicht als Snare identifiziert werden konnten. Von diesen 16 Snares konnten 10 als kurz bezeichnet werden. Hierbei war das Kriterium, dass die Snare einen kurzen Body und auffälligen Transienten haben musste. Dies war bei 10 Snares der Fall.

Nach diesem Test galt es, herauszufinden, welche Prompts, die Länge der Samples genauer beeinflussen. Der nächste Prompt hieß: „Snare, 100 milliseconds“. Die Einstellungen war ein wenig anders, wie bei Abbildung 10 zu sehen ist. Die größte Änderung war es, top-k auf 10 zu stellen. Die Reduzierung von top-k auf 10 beeinflusst die Vielfalt und Kreativität der Ergebnisse. Damit wird das Modell auch genauer, wie in unserem Test zu sehen war. Mit diesen Einstellungen haben wir wieder 20 Snares generiert. 18 davon waren eindeutig Samples, welche sich nach einer normalen Snare anhörten und 2 davon waren leer, beziehungsweise war leichtes Rauschen zu vernehmen. 8 von den 20 Snares waren tatsächlich um die 100 Millisekunden lang. Sie waren sie nicht länger als 100 Millisekunden. Manche waren minimal kürzer. Die meisten Samples aber, waren keine langen Samples. Das Modell hat versucht die Snares kurz zu halten. Es hat also unseren Prompt verstanden und umgesetzt. Auch wurde getestet ob sich top-k auf unsere vorherigen Prompts auswirkt. Das war nicht unbedingt der Fall. Fehler, wie leere Dateien oder Samples in welchen andere Sounds zu hören waren, wurden reduziert. Die Länge der Samples jedoch wurde kaum beeinflusst. Das Modell kann also mit genaueren Angaben arbeiten. Nun galt es zu überprüfen, ob man die Snare kreativ mit anderen Prompts beeinflussen kann. Beispielsweise mit einer Musikrichtung. Der nächste Prompt lautete: „Snare, 150 milliseconds, Dubstep“, um zu überprüfen, wie nützlich die Künstliche Intelligenz einem Musikproduzenten sein kann, welcher nach spezifischen Samples sucht. Wieder gab es 20 Generierungen. Von diesen 20 Generierungen waren 14 Samples als Snares zu erkennen. 3 der nicht zu verwendenden Samples waren Kick Drums anstelle von Snares und ein Sample war ein sich wiederholender Bass Sound,

welcher anschwellt und wieder abschwelt. Er wurde also periodisch lauter und leiser. Ein solcher Bass wird oftmals für klassischen Dubstep verwendet.

Dubstep Snares zeichnen sich vor allen dafür aus, nicht unbedingt wie eine reale Snare zu klingen. Sie haben die typischen Charakteristika, sind meist aber keine echten Aufnahmen von Snares, sondern Nachbildungen mit Hilfe von Synthesizern. Außerdem mischt man Dubstep Snares meist mit einer Clap, also einem Klatschen. Dies sorgt vor allem für mehr Fülle in den hohen Frequenzen.<sup>57</sup> Eine Dubstep Snare hat also einen gewissen Charakter, welchen es nachzustellen gilt. Sie klingt meist sehr punchig, knackig und klar.

Diese Aufgabe erfüllte AudioGen ganz gut, wenn auch nicht überzeugend. Die Ergebnisse sind sehr unterschiedlich, auch variieren die Längen der Snares. Transienten sind oft zu schwach und dementsprechend fehlen den Samples der nötige Punch, um in einem dichten Klangteppich wie es bei Dubstep der Fall ist herauszustechen. Alle Snares klingen nicht wie eine aufgenommene Snare, sondern wie eine typische elektronisch erzeugte Snare.



Abbildung 10 AudioGen veränderte Einstellungen

Natürlich muss hier überprüft werden, ob AudioGen auch andere Musikgenres erkennt. Die Ergebnisse hierzu waren eher schlechter. Gibt man als Prompt „Snare, 150 miliseconds, heavy metal“ ein, so wurden oftmals Oneshots aus E-Gitarrenakkorden, Becken und Snares generiert. Falls dann doch Snares generiert wurden, war der Unterschied zu allen anderen Generierten Snares nicht unbedingt groß. Der Großteil der generierten Snares klang ähnlich zu den Snares, welche mit dem Prompt „Snare“ erstellt wurden.

Der nächste Versuch, den generierten Snares Charakter zu geben, erzielte nüchterne Ergebnisse. Es wurden hier mehrere Varianten der Reihenfolge der Prompts versucht. Zuerst natürlich verwendeten wir die geübte Promptstruktur mit folgendem Prompt: „Snare, 150 miliseconds, Dubstep, punchy“. Punchy in dem Fall steht für eine Snare mit

---

<sup>57</sup>Vgl. Jenkins Peter, *Dubstep Basics – An Introduction To Dubstep Production*, in: *Sound on Sound*, 2010, <https://www.soundonsound.com/techniques/dubstep-basics> (abgerufen am: 30.12.2023).

klar zu hörendem Body und knalligem Transienten. Die Snare soll voll wirken. Die Snares klangen wie die bisher generierten Snares. Es war kein deutlicher Unterschied zu hören.

#### 5.2.4 Untersuchung Growl

Da nun die Erfahrungen aus dem vorherigen Versuch vorliegen, können wir die Prompts dementsprechend anpassen und genauer gestalten. Außerdem werden die Einstellungen dementsprechend angepasst. Top-K steht nun auf 10, wie schon beim Snare Versuch. Die Länge des Samples bleibt fürs Erste bei 10 Sekunden und der Erste Prompt wird „Growl“ sein um, zu überprüfen, was das Modell als Growl anerkennt. Zu erwarten sind erstmal Tiergeräusche, da wir noch keinen weiteren Kontext gegeben haben.

Tatsächlich war dies der Fall. Wir haben 10 Samples mit dem Prompt „Growl“ generiert, welche allesamt Tiergeräusche beinhalteten. Hauptsächlich erinnerten die Growls an Bärengebrüll. Bei einem Sample war nur ein Rauschen zu hören. Nun wurde der Prompt „Dubstep Growl“ ausgeführt, Einstellungen gleichbleibend.

Nun waren Dubstep typische Beats zu hören. Oftmals mit Sub Bass und Melodie im Hintergrund. Dies war bei allen 10 Generierungen der Fall. Das Modell generierte nun also keine einzelnen Samples, sondern fortlaufende Beats. Diese Beats waren allerdings als Dubstep zu erkennen. Wir müssen also unsere Prompts und Einstellungen verändern, um ein Ergebnis zu erzielen.

Die nächste Iteration der Versuchs veränderte den Prompt zu: „Dubstep Growl OneShot“ und änderte nur die Samplelänge auf 1 Sekunde. Einen OneShot bezeichnet man bei Samplepacks als einzelne Sounds.

Die Ergebnisse waren nun deutlich genauer. Insgesamt wurden 20 Audiodateien generiert. 11 von diesen Dateien waren als Growls zu erkennen. 5 davon waren wieder Ausschnitte aus Beats in welchen Drum, Snare aber auch Growls zu hören waren. Der Rest waren kurze Geräusche, welche nicht als Growls zu identifizieren waren.

Als Nächstes galt zu überprüfen, ob das Modell auf Genretypische Begriffe reagiert. Begriffe, wie „Screechy“ oder „Aggressive“ sind bei Samplespacks oft gesucht. Screechy beschreibt einen hoch gepitchten, langgezogenen aggressiven Growl. Er erinnert an ein Tier, welches hochfrequent Schreit. Als „aggressive“ werden Bässe mit vielen Hochfrequenten Anteilen bezeichnet. Es wird unter anderen White Noise und viel Distortion verwendet um die Sounds zu verzerren. Unter dem Prompt „aggressive“ wurden nur 10 Versuche gemacht, da schnell klar wurde, dass dieser Prompt nicht effektiv

war. Es schien als würde er die Ergebnisse zufälliger erzeugen. Hier waren keiner der generierten Sounds als Growls zu erkennen. Es waren beispielsweise einige Shouts mit dabei oder zumindest kurze Sounds die wie Shouts klangen. Growls wurden hier nicht generiert.

Für den Prompt „Dubstep Growl Oneshot Screechy“ wurden 20 Sounds generiert. Hiervon waren 8 als ein Schrei oder Kreischen zu erkennen und einer davon konnte als Screechy durchgehen. Die restlichen Dateien waren Tiergeräusche, Dubstep Beats, Growls und Sub Bässe. Das Modell konnte unseren Befehl nicht umsetzen. Dadurch entstand die Frage, ob wenn man top-k Verändern würde, das Ergebnis ertragreicher wird. Denn anders, wie bei den Snares kann es hier nun sein, dass die niedrige Zahl von top-k einschränkend wirkt. Das Modell könnte weniger Spielraum haben. Es ist gezwungen die Vielfalt der Sounds einzuschränken. Daher wird das Experiment mit dem gleichen Prompt aber anderer top-k Einstellung wiederholt.

Top-k wurde nun auf 100 gestellt. Bei diesem Durchgang wurden wieder 20 Samples generiert. Es wurde während der Generierungen schnell klar, dass auch diese Einstellung keine genauen Ergebnisse erzielt. Von 20 generierten Sounds war nur ein Sound als Screechy zu erkennen. Andere Sounds waren Growls, Beats, Down- und Upriser. Das Ergebnis war also ungenügend.

Es hat also den Anschein, dass das Modell zwar Dubstep Growls und Oneshots erkennt, nicht aber spezifischere Angaben wie, „Agressive“ oder „Screechy“.

### 5.2.5 Fazit zu AudioGen

AudioGen ist ein generatives Tool, welches mit Hilfe von Textprompts Audio generiert. Diese Grundaufgabe erfüllt AudioGen. Erhält das Programm das Wort „Snare“ wird eine Snare generiert. Gibt man dem Programm als Prompt das Wort „Pistolenschuss“ wird tatsächlich ein Pistolenschuss generiert. Diese Grundfunktionen sind gegeben. Das größte Problem bei dem Modell unter der Ausführung mit dem Programm Pinokio, war es herauszufinden welche Prompts das Programm erkennt und verwertet. Am Anfang des Versuches war der Plan, die Prompt Struktur durchzutesten, welche anfangs festgelegt wurde. Als dann aber die Ergebnisse von den Erwartungen abwichen, musste die Teststruktur angepasst werden.

Da das Ziel dieses Experiments war zu testen, wie nützlich die Künstliche Intelligenz, dem Musikproduzenten jetzt schon sein könnte, wurden anfangs nur die schon



vorhandenen Grundeinstellungen verwendet, um Samples zu generieren. Der Grund hierfür ist, dass ein Musikproduzent nicht noch lange Zeit dafür aufwenden kann, ihm nicht bekannte Einstellungen wie top-k zu verändern. Ein KI-Programm, welches dem Produzenten helfen soll, muss von Grund auf einfach und benutzerfreundlich zu bedienen sein. Ohne top-k zu verändern, waren die Ergebnisse sehr ernüchternd. Das Programm reagierte nicht unbedingt auf Prompteingaben, welche genauere Informationen zur gewünschten Snare lieferten. Als man die Einstellung top-k von 250 auf 10 änderte, lieferte das Programm bessere Ergebnisse. So konnte nun eine Länge bestimmt werden. Auch konnte das Programm zumindest das Genre Dubstep erkennen und dem Genre gerechte Snare generieren. Das war allerdings nicht der Fall, bei eingegebenen Genres wie „Heavy Metal“, „Hip-Hop“ oder „Rock“. Im Normalfall sollten deutliche Charakterunterschiede zwischen den Genres festzustellen sein. Dies war nicht der Fall. Es gilt außerdem zu erwähnen, dass die Qualität der generierten Samples nicht den Standards entsprachen, die man zur professionellen Musikproduktion bräuchte. Das Grundgerüst einer Snare war zu hören, die Samples waren eindeutig Snares. Doch oftmals waren die Samples sehr undeutlich. Es war bei fast allen Samples eine Art Rauschen zu hören. Dieses Rauschen machte die Samples zu undeutlich um wirklich verwendbar in der Musikproduktion zu sein. Das Rauschen maskierte die Samples.

Auch bei den so genannten „Growls“ waren die Ergebnisse eher ernüchternd. Das Modell kennt zwar die Begriffe und kann durchaus Growls generieren. Will man diese jedoch etwas genauer beschreiben, so generiert das Modell sehr ungenau, wenn nicht sogar zufällig.

Einer der Hauptkritikpunkte für nützliche Künstliche Intelligenz in der Musikproduktion ist die Benutzerfreundlichkeit. Für einen Musikproduzenten, welcher sich nur bedingt mit Informatik auskennt, kann AudioGen überwältigend sein. Auf manuellem Wege das Modell auf dem eigenen Rechner zum Laufen zu bringen, stellt dem Laien eine große Herausforderung dar. Programme, wie Pinokio, welche wir auch in diesem Experiment verwendeten, helfen dem Nutzer natürlich, doch dies ist nur ein Umweg. Es hätte gut sein können, dass es Pinokio eben nicht gibt und der Musikproduzent nicht auf AudioGen hätte zugreifen können.

Zu diesem Zeitpunkt ist AudioCraft also noch nicht nützlich für einen Musikproduzenten. Hier gilt es zu beachten, dass das auch nicht unbedingt das Ziel von AudioCraft sein soll. In Wirklichkeit ist das Ziel von dem Meta betriebenen Projekt, Entwicklung interaktiver

KI-Systeme voranzutreiben.<sup>58</sup> Es bietet Entwicklern ein Framework, mit den Generierungs-Modellen zu interagieren, auch wenn es noch nicht dafür ausgelegt ist, dem Musikproduzenten nützlich zu sein. Dieser Fakt ist auch bedeutend für die aktuelle Landschaft in Sachen Generierungstools, welche dem Produzenten hilfreich sein sollen. Momentan ist das Angebot noch recht spärlich.

---

<sup>58</sup>Vgl. AudioCraft, AudioCraft: generating high-quality audio and music from text, in: Meta AI, <https://ai.meta.com/resources/models-and-libraries/audiocraft/> (abgerufen am: 30.12.2023).

## 6 Schlussbetrachtung

### 6.1 Zusammenfassung und Fazit

In dieser Arbeit wurden die Fähigkeiten und Grenzen von Künstlicher Intelligenz in Verbindung mit Sample Generierung, zur Unterstützung von Musikproduzenten untersucht. Dabei stand die Frage im Vordergrund, inwiefern Künstliche Intelligenz jetzt schon nützlich sein kann, indem sie dem Dubstep-Produzenten auf Anfrage Audio Samples generiert. Das war die Voraussetzung für ein positives Ergebnis der Untersuchungen. Hierfür recherchierten wir verschiedene AI Tools, mit der Feststellung, dass die meisten Modelle dieser Aufgabenstellung nicht gerecht wurden. Viele AI-Tools beschäftigen sich mit der Generierung von Melodien, Akkorden, oder ganzen Musikstücken. Beispielsweise hat Google's Magenta Studio bereits eine Integrierung für die Digital Audio Workstation Ableton Live, welche neue Melodien und Rythmen generieren kann, um diese dann mit MIDI-Files in das Projekt einzubinden.<sup>59</sup> Auch wenn solche Tools, so wie aktuelle Mastering Tools, schon mit Künstlicher Intelligenz zusammenarbeiten und auch nützlich sind, so war die Fragestellung eine andere. Also testeten wir 2 aktuelle generative Modelle um festzustellen, ob diese dem Musikproduzenten helfen. Dabei konnten wir feststellen, dass Dance Diffusion von HarmonAI und Meta's AudioGen noch signifikante technische Limitationen aufweisen. Während die Audioqualität der generierten Samples von Dance Diffusion überraschte, so enttäuschte das Modell mit der fehlenden Benutzerfreundlichkeit. Das Modell kann zwar für Sample Generierung verwendet werden, doch braucht es sehr lange bis man an die Samples kommt, um damit kreativ weiterzuarbeiten. Es entsteht hier kein kohärenter Workflow. Schließt man das Modell einmal, so muss jede einzelne Seite, jeder einzelne Schritt erneut gestartet werden. Das Setup, hat man das Programm noch nie bedient, dauert mehrere Stunden. Die Sounds letztendlich zu generieren dauert ebenfalls mehrere Stunden. Der Musikproduzent, gerade im Elektronischen Bereich braucht generierte Samples relativ schnell und einfach. Das war hier nicht der Fall. Das Potential des Modell ist allerdings massiv, denn Samples zu generieren, auf der eigenen Soundpalette basierend, wäre für einen Dubstep Produzenten äußerst wichtig.

---

<sup>59</sup>Vgl. Magenta Studio - Ableton Live Plugin, in: Magenta, 2023, <https://magenta.tensorflow.org/studio/> (abgerufen am 03.01.2024).

Auch ließ AudioCraft von Meta's AudioGen bei der Benutzerfreundlichkeit zu wünschen übrig. Hier gab es die Wahl, AudioGen manuell über Python laufen zu lassen, oder durch Pinokio, einem Hilfsprogramm, welches diesen Schritt übersprang. Auch waren die Benutzeroberfläche und die Einstellungen von AudioGen für einen Laien nicht zu verstehen. Es musste erstmal recherchiert werden, was Einstellungsmöglichkeiten überhaupt bedeuteten. Erst dann konnte mit den Einstellungen gespielt werden und damit konnten Prompts, eingegeben werden konnten, überhaupt zielführend umgesetzt werden. Die Grundeinstellungen waren ungenau. Auch war das Ergebnis, der einzelnen Generierungen eher schlecht. Zwar waren den Prompts entsprechenden Samples zu hören, doch konnten diese nur bedingt zur Musikproduktion beitragen, da die Qualität der Samples nicht den Ansprüchen gerecht wurden. Es war oft eine Art Rauschen zu vernehmen, welche die Sounds maskierte. Dadurch ging die Klarheit der Samples verloren. Die Modelle befinden sich also noch in einem eher experimentellem Stadium, es bedarf noch Verbesserung hinsichtlich Genauigkeit und Benutzerfreundlichkeit.

## **6.2 Kritische Bewertung des Vorgehens und der Ergebnisse**

Es wurden hier nun zwei eher unterschiedliche Modelle analysiert. Die Vorgehensweise der Analyse war bei beiden Modellen unterschiedlich. Während der Tests für AudioGen wurden mehrere Dinge festgestellt, welche dazu führten vom vorgeschlagenen Muster abzuweichen. So mussten Einstellungen getroffen werden, um bessere Ergebnisse oder überhaupt Ergebnisse zu erzielen. Ursprünglich war vorgesehen, die Einstellungen des Modells zu belassen, wie in den Grundeinstellungen vorgesehen war. Dies führte zu nicht zufriedenstellenden Ergebnissen. Doch hat sich unter anderem aus diesem Grund die Feststellung ergeben, dass das Modell eben nicht benutzerfreundlich nutzbar ist und dementsprechend unsere Bedingungen nicht erfüllte. Außerdem wurde während dieser Arbeit festgestellt, dass es noch wenig Sample Generierungsmodelle gibt, welche wirklich nutzbar sind.

## **6.3 Ausblick**

Auch wenn KI-gesteuerte Tools wie AudioGen und Dance Diffusion innovative Ansätze bieten, sind sie noch weiter von einer idealen Benutzerfreundlichkeit und Qualität entfernt. Die Zukunft in diesem Bereich ist jedoch sehr vielversprechend. Angesichts der schnellen Entwicklung von Technologien im Bereich Künstlicher Intelligenz, ist zu

erwarten, dass zukünftige Modelle die Art und Weise wie Musik elektronisch produziert wird, grundlegend verändern. Ein wichtiger Aspekt in der elektronischen Musikproduktion ist es, die richtigen Samples für den richtigen Song zu finden oder selbst mit Hilfe von Plugins und Synthesizern zu produzieren. Diese Aufgabe nimmt oft sehr viel Zeit in Anspruch. Gäbe es eine Künstliche Intelligenz, welche beispielsweise den Song im Kontext analysiert und auf Anfrage mit Hilfe eines spezifischen Prompts ein passendes Sample generiert, würde das dem Produzenten sehr viel Zeit und Mühe einsparen.

Die Fortschreitende Integration von KI in der Musikproduktion könnte zu tiefgreifenden Veränderungen in der Musikindustrie führen. Es ist wahrscheinlich, dass KI-Tools zunehmend als kreative Partner und nicht nur als Werkzeuge betrachtet werden. Das zukünftige Potential ist enorm.

## Abbildungsverzeichnis

Abbildung 1: Number of Documents (P. V. Thayyib et al, Sustainability, 2023).....	15
Abbildung 2: Zeigt einen Zeitschritt mit 3 Ebenen in der Hierarchie. ....	28
Abbildung 3: Eingabefeld der Chunking Webpage .....	32
Abbildung 4: Schritte Check GPU Status und Prepare Folders auf der „Dance Diffusion finetune“ Seite.....	33
Abbildung 5 Auszufüllendes Feld für Trainings Schritt .....	34
Abbildung 6: Pinokio Discover Page.....	39
Abbildung 7: Pinokio Installationsvorgang .....	40
Abbildung 8 AudioGen Benutzeroberfläche.....	41
Abbildung 9 AudioGen Default Einstellungen.....	43
Abbildung 10 AudioGen veränderte Einstellungen .....	47

## Literaturverzeichnis

About FFmpeg, in: ffmpeg.org, 2023,

<https://ffmpeg.org/about.html> (abgerufen am 20.12.2023).

AudioCraft, in Python Package Index, 2023,

<https://pypi.org/project/audiocraft/#:~:text=Project%20description%20AudioCraft%20AudioCraft%20is> (abgerufen am 18.12.2023).

Audiocraft: AudioCraft, in: Meta Platforms, 2023,

<https://github.com/facebookresearch/audiocraft> (abgerufen am 18.12.2023).

AudioCraft, AudioCraft: generating high-quality audio and music from text, in: Meta AI,

<https://ai.meta.com/audio/audio-craft/> (abgerufen am: 30.12.2023).

Burgess, Richard James: The art of music production: the theory and practice, Verlag:

Oxford University Press,

<https://books.google.de/books?hl=de&lr=&id=IWEUAAAAQBAJ&oi=fnd&pg=PP1&dq=music+production+pdf&ots=Q4RORacNLM&sig=JhfjaoOtxoXzpS>

[UyhhQ4OHe7Gqk#v=onepage&q=music%20production%20pdf&f=false](https://books.google.de/books?hl=de&lr=&id=IWEUAAAAQBAJ&oi=fnd&pg=PP1&dq=music+production+pdf&ots=Q4RORacNLM&sig=JhfjaoOtxoXzpS)

(abgerufen am 05.12.23), S. 2.

Boucher, Philip: Artificial intelligence: How Does it Work, why Does it Matter, and what Can We Do about It?, in: Think Tank European Parliament, 2020

[https://www.europarl.europa.eu/thinktank/en/document/EPRS\\_STU\(2020\)641547](https://www.europarl.europa.eu/thinktank/en/document/EPRS_STU(2020)641547) (abgerufen am: 01.01.2020).

Chappel Roger: How to Install Audiocraft Locally – Meta’s FREE and Open AI Music Gen, in AITooltip, 2023,

<https://aitooltip.com/how-to-install-audiocraft-locally-metas-free-and-open-ai-musicgen/#:~:text=1%20September%202023%20,for%20both%20Musicgen%20and%20Audiogen> (abgerufen am 18.12.2023)

Colaboratory: Willkommen bei Colab!, in: Colaboratory, 2023,

[https://colab.research.google.com/#scrollTo=Nma\\_JWh-W-IF](https://colab.research.google.com/#scrollTo=Nma_JWh-W-IF) (abgerufen am 10.12.2023).

Chunking, in: Google Colab, 2023,

[https://colab.research.google.com/drive/1wCjjFir4vkWSTU3DwVgCvi5qGtaNnGb\\_](https://colab.research.google.com/drive/1wCjjFir4vkWSTU3DwVgCvi5qGtaNnGb_) (abgerufen am 15.12.2023)

- Daub Adrian: “BRAAAM!”: The Sound that Invaded the Hollywood Soundtrack, in Longreads, 2016,  
<https://longreads.com/2016/12/08/braaam-inception-hollywood-soundtracks/>  
(abgerufen am 16.12.2023)
- De Spiegeleire, Stephan., et al.: AI – Today and Tomorrow. In Artificial Intelligence and the Future of Defense: Strategic Implications for Small- and Medium-Sized Force Providers, in: Hague Centre for Strategic Studies: 2017,  
<http://www.jstor.org/stable/resrep12564.8> (abgerufen am 10.10.2023), S. 44-46.
- Dudenredaktion(o.J.), in: Dudenonline,  
<https://www.duden.de/node/71635/revision/1349018> (abgerufen am 01.10.2023).
- Dickreiter, Michael & Dittel, Volker: Handbuch der Tonstudioteknik, Walter de Gruyter GmbH & Co. KG.: 2023, S 1-2.
- Elaine A. Rich: Artificial Intelligence and the humanities, in: Computers and The Humanities, Bd, 19, Nr. 2,  
doi:10.1007/bf02259633, S. 117–122.
- Engel, Jesse et al.: GANSynth: Adversarial Neural Audio Synthesis, in: arXiv.org, 2019,  
<https://arxiv.org/abs/1902.08710>, (abgerufen am 02.12.2023), S. 1-8.
- Funke, Joachim/Vaterrodt, Bianca: Was ist Intelligenz?, 3. Aufl., München: C.H. Beck Verlag, 2004,  
[https://books.google.de/books?id=G\\_H3sl4fVTEC&lpg=PA9&ots=I0g1NIwqoQ&dq=was%20ist%20intelligenz%20pdf&lr&hl=de&pg=PA11#v=onepage&q&f=false..](https://books.google.de/books?id=G_H3sl4fVTEC&lpg=PA9&ots=I0g1NIwqoQ&dq=was%20ist%20intelligenz%20pdf&lr&hl=de&pg=PA11#v=onepage&q&f=false..) (abgerufen am 1. Dezember 2023), S. 9-11.
- Hao-Wen Dong/Wen-Yi Hsiao, et al.: useGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment., in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018,  
<https://doi.org/10.1609/aaai.v32i1.11312> (abgerufen am 07.12.2023), S. 35.
- Hepworth-Sawyer, Russ: What is music production?: A Producer’s Guide: The Role, the People, the Process, Taylor & Francis, 2017,  
[https://books.google.de/books?hl=de&lr=&id=UJoC\\_eibCzkC&oi=fnd&pg=PP2&dq=music+production+pdf&ots=K2xFxRzXpc&sig=9WVkr6Rx7SSs6hOL1sUhVfNrQ0U#v=onepage&q&f=false](https://books.google.de/books?hl=de&lr=&id=UJoC_eibCzkC&oi=fnd&pg=PP2&dq=music+production+pdf&ots=K2xFxRzXpc&sig=9WVkr6Rx7SSs6hOL1sUhVfNrQ0U#v=onepage&q&f=false) (abgerufen am 05.10.23), S. 17.



- Hernandez-Olivan, Carlos et al.: A Survey on Artificial Intelligence for Music Generation: Agents, Domains and Perspectives, in: arXiv.org (Cornell University), 2022, <https://arxiv.org/pdf/2210.13944.pdf> (abgerufen am 02.12.2023), S. 2-3.
- James Hayton et al.: What drives UK firms to adopt AI and robotics, and what are the consequences for jobs?, in: Institute for the Future of Work: London: 2023, [https://global-uploads.webflow.com/64d5f73a7fc5e8a240310c4d/650a05c1b2daf9e31b0ae741\\_FINAL%20WP%20-%20Adoption%20of%20Automation%20and%20AI%20in%20the%20UK.pdf](https://global-uploads.webflow.com/64d5f73a7fc5e8a240310c4d/650a05c1b2daf9e31b0ae741_FINAL%20WP%20-%20Adoption%20of%20Automation%20and%20AI%20in%20the%20UK.pdf) (abgerufen am 10.10.2023), Seiten 4-5.
- Jenkins Peter, Dubstep Basics – An Introduction To Dubstep Production, in: Sound on Sound, 2010, <https://www.soundonsound.com/techniques/dubstep-basics> (abgerufen: 30.12.2023).
- Kerlinger Charlie: How To Create A Dubstep Growl, benvaughn, 2022, <https://www.benvaughn.com/how-to-create-a-dubstep-growl/> (abgerufen am 04.01.2023)
- Kerlinger Charlie: A Brief History Of Dubstep: From Its Underground Origins To Worldwide Popularity, in: benvaughn, 2022, <https://www.benvaughn.com/a-brief-history-of-dubstep-from-its-underground-origins-to-worldwide-popularity/>, (abgerufen am 04.01.2024).
- Kreuk Felix, et al.: AudioGen: Textually Guided Audio Generation, in: arxiv.org, 2023, <https://arxiv.org/abs/2209.15352> (abgerufen am 18.12.2023), S. 3-5.
- Kreutzer R.T. / Sirrenberg M., Künstliche Intelligenz verstehen - Grundlagen – Use-Cases – unternehmenseigene KI-Journey, Wiesbaden: Springer Gabler, 2019, <https://link.springer.com/book/10.1007/978-3-658-25561-9> (abgerufen am 1. Dezember 2023), S. 2.
- Magenta Studio - Ableton Live Plugin, in: Magenta, 2023, <https://magenta.tensorflow.org/studio/> (abgerufen am: 03.01.2024).

- Mehri Soroush et al.: SampleRNN: An Unconditional End-to-End Neural Audio Generation Model, in: arXiv.org: 2016, Online:  
<https://arxiv.org/abs/1612.07837>, (online abgerufen am 02.12.2023) , S. 1-4.
- Mehrish Ambhuj et al.: A Review of Deep Learning Techniques for Speech Processing, in: arXiv.org, 2023,  
<https://arxiv.org/abs/2305.00359>, (online abgerufen am 02.12.2023), S. 3-8.
- Michaelangelo, Matos: electronic dance music, in: Encyclopedia Britannica, 2023,  
<https://www.britannica.com/art/electronic-dance-music>, (abgerufen am 01.01.2024).
- Miguel Civit et al., A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends, Expert Systems with Applications, Expert Systems with Applications: Volume 209, 2022,  
<https://www.sciencedirect.com/science/article/pii/S0957417422013537?via%3DiHub>
- Miraglia, Dusti: What Are Transients? Amplify Your Mixes with Electrifying Dynamics and Unbelievable Control, in: Unison Audio, 2023,  
<https://unison.audio/what-are-transients/>, (abgerufen: 22.12.2023).
- Monahan Max, The Engineer's Guide to the Perfect Snare Sound, in: Sonicbids 2016,  
<https://blog.sonicbids.com/the-engineers-guide-to-the-perfect-snare-sound>  
(abgerufen: 22.12.2023).
- Muehmel Kurt: What Is a Large Language Model, the Tech Behind ChatGPT?,  
Blogeintrag von Dataiku, 2023,  
<https://blog.dataiku.com/large-language-model-chatgpt> (abgerufen am 12.10.2023).
- LeCun, Yann & Bengio, Y. & Hinton, Geoffrey: Deep Learning, in: Nature, Bd. 521, Nr. 7553, 2015,  
[https://www.researchgate.net/publication/277411157\\_Deep\\_Learning](https://www.researchgate.net/publication/277411157_Deep_Learning) (Online abgerufen 12.10.2023), S. 436-437.
- Open AI, What is ChatGPT, in: Open AI, 2023,  
<https://help.openai.com/en/articles/6783457-what-is-chatgpt>, (abgerufen am 11.10.2023).

- Pan Angelica: A Gentle Introduction to Dance in: Diffusion, Weights and Biases, 2023, [https://wandb.ai/wandb\\_gen/audio/reports/A-Gentle-Introduction-to-Dance-Diffusion--VmlldzoyNjg1Mzky](https://wandb.ai/wandb_gen/audio/reports/A-Gentle-Introduction-to-Dance-Diffusion--VmlldzoyNjg1Mzky), (abgerufen: 10.12.2023).
- Purwins Hendrik et al.: Deep Learning for Audio Signal Processing, in: IEEE Journal of Selected Topics in Signal Processing, Bd. 13, Nr. 2, 2019, doi:10.1109/jstsp.2019.2908700 (abgerufen am 02.12.2023), S. 1-9.
- P. V. Thayyib et al.: State-of-the-Art of Artificial Intelligence and Big Data Analytics Reviews in Five Different Domains: A Bibliometric Summary, in: Multidisciplinary Digital Publishing Institute, bd. 15, Nr. 5, 2023, <https://www.mdpi.com/2071-1050/15/5/4026> (abgerufen am 11.10.2023), Seiten 2-3.
- Steinhardt, Sebastian: Musikproduktion der Zukunft: Eine empirische Studie über neue Möglichkeiten für Musiker und Produzenten, Hamburg, Deutschland: Diplomica Verlag GmbH, 2013, [https://books.google.de/books?id=hAEnJYReJXAC&dq=Geschichte+Musikproduktion&lr=&hl=de&source=gbs\\_navlinks\\_s](https://books.google.de/books?id=hAEnJYReJXAC&dq=Geschichte+Musikproduktion&lr=&hl=de&source=gbs_navlinks_s) (abgerufen am 05.12.2023), S. 1-3.
- Sturgis, Joey: How To Get A Great Snare Sound In Any Mix, in: Joey Sturgis Tones, 2023, <https://joeysturgistones.com/blogs/learn/how-to-get-a-great-snare-sound-in-any-mix> (abgerufen: 22.12.2023).
- Tsantekidis, Avraam, et al.: Chapter 5 - Recurrent neural networks, in: Academic Press, 2022, <https://www.sciencedirect.com/science/article/abs/pii/B9780323857871000105> (online abgerufen am 07.12.2023), S. 101-115.
- Van den Oord, Aaron, et al.: WaveNet: A Generative Model for Raw Audio, in: arXiv.org, 2016, <https://arxiv.org/abs/1609.03499> (abgerufen am 02.12.2023), S. 1-5.
- Vaswani, Ashish/Shazeer, Noam M., et al.: Attention is All you Need, In: semanticscholar.org, 2017, <https://www.semanticscholar.org/paper/Attention-is-All-you-Need-Vaswani-Shazeer/204e3073870fae3d05bcbc2f6a8e263d9b72e776> (abgerufen am 07.12.2023), S.2-3

Weights & Biases, Weights & Biases for Enterprise, in: Weights & Biases, 2023,  
<https://wandb.ai/site/for-enterprise> (abgerufen: 10.12.2023).

Weights and Biases: in wandDB.ai, 2023,  
<https://wandb.ai/home> (abgerufen: 15.12.2023).

Zhang Aston/ Lipton Zack C. et al.: Dive into Deep Learning, in: Cambridge University  
Press, 2023,  
[https://d2l.ai/chapter\\_recurrent-neural-networks/index.html](https://d2l.ai/chapter_recurrent-neural-networks/index.html) (Online abgerufen  
am 07.12.2023).

Zhu, Yueyue, et al.: A Survey of AI Music Generation Tools and Models, in: arXiv.org:  
2023,  
<https://arxiv.org/abs/2308.12982> (abgerufen am 05.12.2023), Seiten 2-3.