Inhaltsverzeichnis

[Abstract 3](#_Toc29289706)

[Danksagung 3](#_Toc29289707)

[1. Einleitung 3](#_Toc29289708)

[2. Arten von Pattern 3](#_Toc29289709)

[3. Die Fourier-Analyse 3](#_Toc29289710)

[3.1. Die diskrete Fouriertransformation 3](#_Toc29289711)

[4. Die Spektralanalyse 6](#_Toc29289712)

[5. Künstliche Neuronale Netze 7](#_Toc29289713)

[5.1. Convolutional Neural Network (CNN) 7](#_Toc29289714)

[5.1.1. Aufbau 7](#_Toc29289715)

[5.1.1.1. Convolutional Schicht 7](#_Toc29289716)

[5.1.1.2. Pooling Schicht 7](#_Toc29289717)

[5.1.1.3. Vollständig Vermaschtes Netzwerk 7](#_Toc29289718)

[5.1.2. Aktivierungsfunktionen 7](#_Toc29289719)

[5.1.2.1. Backpropagation 7](#_Toc29289720)

[5.1.2.2. ReLU Funktion 7](#_Toc29289721)

[5. Pattern Erkennung mithilfe der Fourier Transformation 7](#_Toc29289722)

[6. Pattern Erkennung mithilfe von Rhythmus und Melodie 7](#_Toc29289723)

[6.1. String basierte Pattern suche 7](#_Toc29289724)

[6.2. Geometrische Pattern suche 7](#_Toc29289725)

[6.2.1. Fünf-Dimensionales Punkt Set 7](#_Toc29289726)

[6.2.1.1. SIA 8](#_Toc29289727)

[6.2.1.2. SIATEC 8](#_Toc29289728)

[6.2.1.3. COSIATEC 8](#_Toc29289729)

[6.3. Pattern Erkennung mithilfe von Matrizen 8](#_Toc29289730)

[7. Pattern Erkennung mithilfe von Neuronalen Netzen 8](#_Toc29289731)

[8. Realisierung der Pattern Erkennung 9](#_Toc29289732)

[8.1. Realisierung des Low-/Band-/High-Pass-Filter 9](#_Toc29289733)

[Abbildungsverzeichnis 9](#_Toc29289734)

[Tabellenverzeichnis 9](#_Toc29289735)

[Literaturverzeichnis 9](#_Toc29289736)

[Eidesstattliche Erklärung 10](#_Toc29289737)

# Abstract

# Danksagung

# 1. Einleitung

# 2. Arten von Pattern

# 3. Die Fourier-Analyse

Für jedes wissenschaftliche Feld, welches mit zum Beispiel mechanischen Schwingungen, elektrischen Schwingungen oder auch mit der Bildverarbeitung in Berührung kommt, ist die Fourier-Analyse ein wichtiges Werkzeug. Diese beruht auf der Grundaussage von Jean Baptiste Joseph Fouriers Forschung aus dem Jahre 1822 welche besagt, dass jede Schwingung mithilfe von unendlich vielen Sinus- und Kosinus-Schwingungen zusammengesetzt werden kann. (Strick, 2012)

Somit kann auch jede periodische Schwingung wieder in ihre Einzelteile zerlegt werden. Dieser Vorgang wird Fourier-Analyse genannt. Dabei gibt es, je nach Eigenschaft der Funktion, vier verschiedene Arten der Fouriertransformation.

* Fourierreihe
* Kontinuierliche Fouriertransformation
* Diskrete Fouriertransformation (DFT)
* Fouriertransformation für zeitdisktrete Signale (DTFT)

In dieser Arbeit werden lediglich die DFT oder die DTFT betrachtet, da diese die einzigen Transformationen sind, die von einem Computer ausgeführt werden können, weil die DFT/DTFT diskrete Werte und eine endliche Länge besitzen (Smith, 1997).

# 3.1. Die diskrete Fouriertransformation

Wie im vorherigen Kapitel schon beschrieben, ist die diskrete Fouriertransformation die einzige Transformation, welche vom Computer berechnet werden kann. Um aber die diskrete und endliche Funktion der DFT zu verstehen, wird an dieser Stelle aufgezeigt wie sich die Formeln der DFT und der kontinuierlichen Fouriertransformation (CTFT) unterscheiden.

Formel : Formel der CTFT (Weisstein, Discrete Fourier Transform, 2015)

Formel : Formel der DFT (Weisstein, Discrete Fourier Transform, 2015)

In Formel 2 ist N die Anzahl der Samples und *k* das *k*-te Sample Bin.

Der Unterschied der beiden Formel liegt bei dem Tausch des Integrals durch die Summe und bei dem Exponenten der e-Funktion. Das Integral in der DFT entfällt und wurde durch ein Sigma ersetzt, da die DFT mit konkreten Zahlenwerten rechnet und nicht mit Flächen. Die Exponenten unterscheiden sich darin, dass die Frequenz *F* aus der CTFT mit substituiert wurde. Aus der DFT ist *n* gleichzusetzten mit dem aus der CTFT stammende *t* (Weisstein, 2015). Die DFT-Formel kann noch vereinfacht werden indem mit und die *e*-Funktion durch die Eulersche Identitätsubstituiert wird. Dadurch erhält man folgende Formel. (Thormählen, 2018)

Formel : Umformulierte DFT-Formel

Somit müssen für die *x* nur noch die einzelnen Samplewerte eingetragen und die Formel ausgerechnet werden. Das Ergebnis aus dieser Formel besteht aus einem Real- und einen Imaginär Anteil. Die Real- und Imaginären Anteile können dann in ein Koordinatensystem eingetragen werden, wobei der Realteil die x-Achse und der imaginäre Anteil die y-Achse darstellt. Der Winkel des eingetragenen Punkt zur positiven x-Achse beschreibt die Phasenverschiebung der Schwingung und die Entfernung des eingetragenen Punktes zum Koordinatenursprung beschreibt die Amplitude der -ten Schwingung.

Formel : Berechnung des Betrages eines Vektors

Welche Frequenz die -te Schwingung hat hängt von der Abtastfrequenz und der Abtastpunkte ab. Die Frequenz des Punktes berechnet sich durch . (Hermann, 2010)

Bevor Formel 4 benutzt werden kann, um die Amplitude zu berechnen, müssen alle Samplewerte deren Index ≥ Anzahl Samples/2 sind gelöscht werden. Dieser Wert wird das Nyquist-Limit genannt. Es besagt das alle Ergebnisse über diesem Limit gelöscht werden und die Ergebnisse der Samples, die unter dem Nyquist-Limit liegen, verdoppelt werden (Weisstein, 2005). Zuletzt müssen die jetzigen Resultate durch die Anzahl der Samples *N* geteilt werden. (Hermann, 2010)

Bei der DFT können jedoch zwei Fehler geschehen. Zum einen *Leakage* und zum anderen *Aliasing*. Leakage tritt auf, wenn die Schwingung in dem betrachteten Zeitabschnitt nicht perfekt periodisch ist.

Ein Bild, das Text, Karte enthält.

Automatisch generierte Beschreibung

Abbildung : Aufzeigen des Leakage-Effektes mit einer Funktion von 2 Hz (oben) und 2.5 Hz (unten)

Wie in der vorgestellten Abbildung zu sehen, ist in dem Zeitfenster eine Sprungstelle zu sehen. Die Sprungstelle befindet sich an den Rändern des Ursprungsignals. Durch den daraus resultierenden ungleichmäßigen Kurvenverlauf kommt es zu einer „*Verschmierung*“ des Spektrums. Der *Leakage*-Effekt kann mit Fenstfunktionen abgeschwächt werden. Wie in **Abbildung 2** zu sehen ist, schwächen diese Funktionen die Ränder des Ursprungsignals so ab, dass diese gegen Null gehen. Dadurch werden bei einem wiederholen des Signals nur noch möglichst kleine Sprünge im Kurvenverlauf vorkommen (Roberts, 2017). Einige Fensterfunktionen sind unter anderem Barlett-, Gauß- oder die Hanning-Festerfunktion (Weisstein, 2015).

Ein Bild, das Himmel enthält.

Automatisch generierte Beschreibung

Abbildung : Anwenden der Hanning-Fensterfunktion auf eine Schwingung von 2.5 Hz

*Aliasing* tritt auf, wenn die Samplerate zu gering für hohe Frequenzen ist. Um Aliasing zu verhindern muss die Samplerate angehoben werden oder das Signal muss vorgefiltert werden, um die zu hohen Frequenzteile zu minimieren (Roberts, 2017).

# 4. Die Spektralanalyse

Die Spektralanalyse eines Musikstücks besteht aus vielen einzelnen Fouriertransformationen, die über den gesamten Verlauf des Stücks durchgeführt werden. Dabei hängt die Länge der einzelnen Fouriertransformationen und die Auflösung der Frequenzen von der Größe des betrachteten Fensters ab. Bei einer Samplerate von 44100 Samples pro Sekunde, was eine Standartgröße bei Musik auf CDs angesehen werden kann (Teufel, 2019), bedeutet dass das der Abstand zwischen 2 Samples Rund 22,67 µs beträgt. Daraus lässt sich wiederum Schlussfolgern das bei einer Fenstergröße von 32 Samples die zeitliche Genauigkeit, durch die Fenstergröße von 0,7 ms, im Mittelpunkt steht aber die Frequenz vernachlässigt wird. Die Vernachlässigung beruht auf dem in Punkt 3.1 beschrieben Nyquist-Limit, wodurch der Frequenzbereich in 15 gleichgroße Bereiche geteilt wird. Bei 44100 Samples pro Sekunde und der daraus resultierenden maximalen Frequenz von 22050 Hz Besitz jeder Bereich 1470 Frequenzen, was für eine Erkennung von Pattern unbrauchbar ist wie **Abbildung 3** aufzeigt.

Ist jedoch das Fenster groß, gibt es eine feine Unterteilung der Frequenz aber eine ungenaue zeitliche Genauigkeit. Analog zu dem obigen Rechenbeispiel aber mit einer Fenstergröße von 32768 Bins bedeutet dies, dass das Fenster circa 0,74 Sekunden groß ist und jeder Frequenzbereich ein Delta von, rein Rechnerisch, 1,345 Hz hat. Um die großen Zeitintervalle auszugleichen können Bereiche auch überlappt werden. Dadurch wird das Spektralbild, je nach Überlappungsgrad, genauer. (Cannam, Landone , & Sandler, 2010)



Abbildung : a) Spektrum mit 32 Bins berechnet  
 b) Spektrum mit 32768 Bins berechnet (herangezoomt)

# 5. Convolutional Neural Network (CNN)

Um die Spektren, welche vergleichbar mit **Abbildung 3.b** sind, automatisch verarbeiten zu können werden neuronale Netze benötigt. Dafür wird ein CNN benutzt, da diese besser geeignet sind als andere Formen von neuronalen Netzwerken. Der wichtigste Vorteil ist die Genauigkeit, mit der das Netzwerk die geforderte Klassifizierung vornimmt. In dem Bereich der Pattern Erkennung sind CNN’s seit circa 9 Jahren besser als andere Typen von neuronalen Netzen. (Imagenet, 2017)

Der zweite Vorteil ist die Effizienz. Durch die Pooling-Schichten der CNN’s wird der Rechenaufwand verringert, da bei einem *max-Pooling* nur der größte Wert aus einer vordefinierten *mxn Poolingmatrix* übernommen wird. (Karpathy, 2018)

Die in diesem Kapitel vorgestellten Netze haben auch einen anderen Aufbau im Gegensatz zu anderen neuronalen Netzen (NN). Dabei haben NN mindestens 2 Schichten. Eine eindimensionale Input-Schicht und eine eindimensionale Ausgabe-Schicht. Optional können versteckte Schichten eingefügt werden und diese vollständig oder nur teilweise mit der Vorgängerschicht verbunden werden. Durch die Eindimensionalität der Input-Schicht können Daten nur als Vektor an das NN weitergegeben werden, wodurch die Verarbeitung von Bildern nur noch eine eingeschränkte Effizienz aufweist. CNN’s haben den eben beschriebenen Aufbau und Nachteil nicht. Sie besitzen eine bis zu dreidimensionale Input-Schicht, womit RGB-Farbbilder Effizient an das Netz gegeben werden kann. Die zweite Komponente der CNN’s ist eine *Convolutional* Schicht, welche Matrizen über die Vorgängerschicht schiebt, um gewünschte Eigenschaften stärker hervor zu heben. Komponente drei ist die *Pooling* Schicht. Das Pooling bewirkt, dass gewisse Informationen, innerhalb einer mxn Matrix, verworfen werden. Der letzte Baustein ist ein voll Vermaschtes neuronales Netz. Das vollvermaschte Netz macht an dieser Stelle wieder Sinn, da die Bilder so bearbeitet wurden, dass das es nur noch relevante Informationen von den Vorgängerschichten bekommt.

Die eben beschriebenen Bausteine müssen nicht in der vorgestellten Reihenfolge vorkommen. Es können mehrere Convolutional Schichten realisiert werden bevor eine oder mehrere Pooling Schichten implementiert wurden. Diese Strukturen können sich dabei in ein und demselben Netz auch wiederholen. Die einzige Strukturen welche einen festen Platz in der Topologie des neuronalen Netzes besitzen ist die Input-Schicht, welche am Anfang sein muss, und das voll vermaschte neuronale Netz mit der dazugehörigen Output-Schicht, welches das letzte Segment des CNN’s darstellt.

# 5.1. Funktionsweise der einzelnen Segmente

# 5.1.1. Convolutional Schicht

# 5.1.2. Pooling Schicht

# 5.1.3. Vollständig Vermaschtes Netzwerk

# 5.2. Aktivierungsfunktionen

# 5.2.1. Backpropagation

# 5.2.2. ReLU Funktion

# 5. Pattern Erkennung mithilfe der Fourier Transformation

# 6. Pattern Erkennung mithilfe von Rhythmus und Melodie

# 6.1. String basierte Pattern suche

# 6.2. Geometrische Pattern suche

# 6.2.1. Fünf-Dimensionales Punkt Set

(Erst ab drittens Interessant!! Zweidimensional Aufbereiten <http://drops.dagstuhl.de/opus/volltexte/2006/652/pdf/06171.MeredithDavid.Paper.652.pdf> Plus Quellen, Chromatische Scala für Musik(Wert 2 der Fünf Dimensionales Punktset)

Algorithmen dafür unter anderen: SIA (MTPs), SIATEC, COSIATEC)

# 6.2.1.1. SIA

# 6.2.1.2. SIATEC

# 6.2.1.3. COSIATEC

# 6.3. Pattern Erkennung mithilfe von Matrizen

<http://hss.ulb.uni-bonn.de/2013/3271/3271.pdf> (darunter alle von dem Link ab 6.2 interessant)

Pattern können auch in Subpattern unterteilt sein

Pattern ranking function?

Ganze Musikstück muss nicht unbedingt ein Pattern sein es können auch „sinnlose“ Filler dabei sein

Arten von Pattern (Seite 98)

Methode: Melody extraction (nimmt an das Melodie in der höchsten Note enthalten ist)

Stringbasierte Erkennung und geometrische Erkennung (Seite 101 Quellen anschauen) und beides zusammen (102 oben)

<https://www.researchgate.net/profile/Jia_Lien_Hsu/publication/221615538_Efficient_Repeating_Pattern_Finding_in_Music_Databases/links/54e614280cf2bff5a4f29302.pdf>

correlative Matrix Verfahren

<http://pdfs.semanticscholar.org/f6ce/8e49f1987d927be91c99e82458c415266c89.pdf>

Baumstruktur (suffix tree) beim auswerten (auch schon in anderen Werken benutzt)

# 7. Pattern Erkennung mithilfe von Neuronalen Netzen

Mit Bildoperation aus Computergrafik bearbeiten <- vllt nicht nötig da Einstellungen im Sonic Visualizer die obsolet machen

Motif Viewer mal anschauen

# 8. Realisierung der Pattern Erkennung

Nach dem in den vorherigen Kapiteln Grundlagen der einzelnen Grundbausteine der Pattern Erkennung erklärt wurden, geht es in diesem Abschnitt um die Realisierung der einzelnen Zwischenaufgaben. Die Umsetzung des Projektes wird in Python geschehen.

Zu aller erst müssen auf das gewünschte Musikstück verschiedene Filter gelegt werden. Die benötigten Filter sind ein Low-Pass-Filter, ein Band-Pass-Filter und ein High-Pass-Filter. Als nächstes werden die Patternstreams der Beats per minute (BPM) und des Rhythmus des Musikstücks extrahiert. Danach werden die Spektrumbilder des Low- und High-Pass-Filter und die Patternstreams der BPM und des Rhythmus an ein neuronales Netz gegeben, um die Pattern des Basses, der Drum und der Clap zu erkennen.

Der darauffolgende Schritt besteht darin, dass die Audiodatei des Band-Pass-Filters so manipuliert wird, sodass die Konvertierung der .wav Datei in ein MIDI-File möglich ist. Wenn dies geschehen ist werden die im Kapitel [am Schluss aktualisieren] verschiedene Algorithmen zur Pattern Erkennung implementiert und bewertet.

Zum Schluss müssen die erkannten Pattern angezeigt werden.

# 8.1. Realisierung des Low-/Band-/High-Pass-Filter

Für die Umsetzung der drei Filter wird das aus der Pythonbibliothek stammende Packet aubio verwendet. Dieses ist zum einlesen der .wav zuständig. Da es in diesem Packet keine Funktionen für einen Low-/Band-/High-Pass-Filter gibt, muss für dessen Umsetzung ein Biquad-Filter benutzt werden.

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Literaturverzeichnis

Cannam, C., Landone , C., & Sandler, M. (2010). *A Brief Reference*. Abgerufen am 2. Januar 2020 von Sonic Visualiser: https://www.sonicvisualiser.org/doc/reference/1.3/en/

Hermann, P. D. (2010). *Die Disktrete Fouriertransformation (DFT)*. Abgerufen am 19. Dezember 2019 von Technische Universität Wien: https://ti.tuwien.ac.at/cps/teaching/courses/dspv/files/DFT-FFT.pdf

Imagenet. (26. July 2017). *Large Scale Visual Recognition Challenge 2017 (ILSVRC2017)*. Abgerufen am 7. Januar 2020 von Imagenet: http://image-net.org/challenges/LSVRC/2017/results

Karpathy, D. A. (17. Oktober 2018). *CS231n Convolutional Neural Networks for Visual Recognition*. Abgerufen am 8. Januar 2020 von Github: http://cs231n.github.io/convolutional-networks/#add

Roberts, P. S. (2017). *University of Oxford.* Abgerufen am 31. Dezember 2019 von Lecture 7 - The Discrete Fourier: http://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf

Smith, S. W. (1997). *The Scientist & Engineer's Guide to Digital Signal Processing.* USA, Kalifornien: California Technical Pub.

Strick, H. K. (1. Juli 2012). *Joseph Fourier (1768–1830)*. Abgerufen am 18. Dezember 2019 von Spektrum: https://www.spektrum.de/wissen/joseph-fourier-1768-1830/1156113

Teufel. (9. Juli 2019). *Die Abtastrate – Tastend nach dem besten Sound*. Abgerufen am 2. Januar 2020 von Teufel: https://blog.teufel.de/abtastrate/#chapter2

Thormählen, P. D. (23. April 2018). *Multimediale Signalverarbeitung Frequenztransformation*. Abgerufen am 19. Dezember 2019 von Philipps Universität Marburg: https://www.mathematik.uni-marburg.de/~thormae/lectures/mmk/mmk\_3\_2\_ger\_web.html#1

Weisstein, E. W. (15. April 2005). *Nyquist Frequency*. Abgerufen am 19. Dezember 2019 von Wolfram Math World: view-source:http://mathworld.wolfram.com/NyquistFrequency.html

Weisstein, E. W. (2. Februar 2015). *Apodization Function*. Abgerufen am 31. Dezember 2019 von Wolfram Math World: http://mathworld.wolfram.com/ApodizationFunction.html

Weisstein, E. W. (2. Februar 2015). *Discrete Fourier Transform*. Abgerufen am 19. Dezember 2019 von Wolfram Math World: http://mathworld.wolfram.com/DiscreteFourierTransform.html

# Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Leipzig, den