

# Diplomarbeit

Kerstin Dosenbach

Klassifikation von Audiosignalen mit  
Support Vector Machines

Kerstin Dosenbach  
Klassifikation von Audiosignalen mit  
Support Vector Machines

Diplomarbeit eingereicht im Rahmen der Diplomprüfung  
im Studiengang Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Andreas Meisel  
Zweitgutachter : Prof. Dr. rer. nat. Wolfgang Fohl

Abgegeben am 30. August 2007

# **Kerstin Dosenbach**

## **Thema der Diplomarbeit**

Klassifikation von Audiosignalen mit Support Vector Machines

## **Stichworte**

Maschinelles Lernen, Support Vector Machines, Audioklassifikation, Instrumentenklassifikation

## **Kurzzusammenfassung**

Diese Diplomarbeit beschäftigt sich mit dem Klassifizieren von Audiosignalen mit Hilfe von Support Vector Machines. Beispielhaft werden drei Gitarren verschiedener Gitarrenbauer klassifiziert. Der Klang der Gitarre wird analysiert und die Merkmale, die den Klang der Gitarre beschreiben, extrahiert. Um die Merkmale auf ihre Verwendbarkeit zu überprüfen, werden verschiedene Experimente mit unterschiedlichen Merkmalen durchgeführt. Die Ergebnisse der einzelnen Experimente werden diskutiert.

# **Kerstin Dosenbach**

## **Title of the paper**

Audio Classification using Support Vector Machines

## **Keywords**

Machine Learning, Support Vector Machines, Audio-Classification, Musical Instrument Classification

## **Abstract**

This thesis introduces a system for acoustic guitar recognition using Support Vector Machines. The guitar sound is analysed and different features are extracted to represent the timbre of the instruments. To verify the usability of the features, different experiments with diverse features are accomplished. The results of these experiments are discussed.

## **Danksagung**

Ich möchte mich bei Prof. Meisel und Prof. Fohl für die gute Betreuung dieser Diplomarbeit bedanken.

Mein besonderer Dank gilt Marcus Dramburg, der mir in dieser Zeit den Rücken freigehalten und mich, wann immer es sein musste, aufgebaut hat und Korrektur las.

Des Weiteren möchte ich mich bei meinen Eltern und meinem Bruder bedanken, die mir durch ihre Unterstützung das Studium erst ermöglicht haben.

Vielen Dank auch an Nina Hagemann, Lukas Johner und Rainer Sodtke für das Korrekturlesen.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>
<b>1. Einleitung</b>	<b>9</b>
1.1. Ähnliche Arbeiten . . . . .	9
1.2. Problemstellung . . . . .	10
1.3. Gliederung dieser Arbeit . . . . .	10
<b>2. Aufbau des Gitarrenklangs</b>	<b>12</b>
2.1. Aufbau eines Klangs aus Grundton und Obertönen . . . . .	12
2.2. Klangfarbe . . . . .	13
2.3. Resonanzverhalten der Gitarre . . . . .	14
<b>3. Merkmalsextraktion</b>	<b>16</b>
3.1. Bestimmung der Grundfrequenz . . . . .	16
3.2. Zeitlicher Verlauf der Harmonischen . . . . .	18
3.3. Bestimmung des nontonalen Spektrums . . . . .	21
3.4. Mel Frequency Cepstral Coefficients . . . . .	23
3.4.1. Mel . . . . .	23
3.4.2. Berechnung der Mel Frequency Cepstral Coefficients . . . . .	24
<b>4. Support Vector Machines</b>	<b>26</b>
4.1. Lineare SVMs . . . . .	26
4.2. Nichtlineare SVMs . . . . .	29
4.3. Multi-Klassen-Probleme . . . . .	31
4.3.1. One-Versus-One . . . . .	31
4.3.2. One-Versus-Rest . . . . .	32
<b>5. Realisierung</b>	<b>33</b>
5.1. Verwendete Software . . . . .	33
5.2. Aufbau der Merkmalsextraktion . . . . .	34
5.3. Aufbau der Trainings- und Testphase . . . . .	37
5.3.1. Trainingsphase . . . . .	37

---

5.3.2. Testphase . . . . .	39
<b>6. Ergebnisse</b>	<b>41</b>
6.1. Experiment 1 . . . . .	42
6.2. Experiment 2 . . . . .	44
6.3. Experiment 3 . . . . .	46
6.4. Experiment 4 . . . . .	48
6.5. Experiment 5 . . . . .	50
6.6. Experiment 6 . . . . .	52
6.7. Experiment 7 . . . . .	54
6.8. Diskussion der Ergebnisse . . . . .	56
<b>7. Fazit und Ausblick</b>	<b>58</b>
<b>Literaturverzeichnis</b>	<b>59</b>
<b>A. Verwendete und entwickelte Programme</b>	<b>61</b>
A.1. Programm zur Merkmalsextraktion . . . . .	61
A.2. Programm zum Trainieren der SVMs . . . . .	64
A.3. Programm zum Testen . . . . .	65
<b>B. Inhalt der DVD</b>	<b>66</b>
<b>C. Verwendete Mathematische Notation</b>	<b>67</b>
<b>Glossar</b>	<b>68</b>

# Tabellenverzeichnis

6.1. Aufbau Merkmalsvektor Experiment 1 . . . . .	42
6.2. Verwechslungsmatrix Experiment 1 . . . . .	42
6.3. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 1 . . . . .	43
6.4. Aufbau Merkmalsvektor Experiment 2 . . . . .	44
6.5. Verwechslungsmatrix Experiment 2 . . . . .	44
6.6. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 2 . . . . .	45
6.7. Aufbau Merkmalsvektor Experiment 3 . . . . .	46
6.8. Verwechslungsmatrix Experiment 3 . . . . .	46
6.9. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 3 . . . . .	47
6.10. Aufbau Merkmalsvektor Experiment 4 . . . . .	48
6.11. Verwechslungsmatrix Experiment 4 . . . . .	48
6.12. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 4 . . . . .	49
6.13. Aufbau Merkmalsvektor Experiment 5 . . . . .	50
6.14. Verwechslungsmatrix Experiment 5 . . . . .	50
6.15. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 5 . . . . .	51
6.16. Aufbau Merkmalsvektor Experiment 6 . . . . .	52
6.17. Verwechslungsmatrix Experiment 6 . . . . .	52
6.18. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 6 . . . . .	53
6.19. Aufbau Merkmalsvektor Experiment 7 . . . . .	54
6.20. Verwechslungsmatrix Experiment 7 . . . . .	54
6.21. Klassifikationsergebnisse einzelner Knoten des DAG Experiment 7 . . . . .	55

# Abbildungsverzeichnis

2.1. Spektrum eines Gitarrenklangs . . . . .	13
2.2. Abhängigkeit des Klangspektrums von der Anschlagposition . . . . .	14
2.3. Resonanzbereiche der Gitarre . . . . .	15
3.1. Ablauf der Cepstrumberechnung . . . . .	17
3.2. Berechnung des zeitlichen Verlaufs der Harmonischen . . . . .	19
3.3. Zeitlicher Verlauf der Harmonischen . . . . .	20
3.4. Berechnung der Nontonalen Merkmale . . . . .	21
3.5. Nontonale Spektren der Gitarren Hense, Marin und Wichmann . . . . .	22
3.6. Zusammenhang zwischen Frequenz und wahrgenommener Tonheit . . . . .	23
3.7. Prozess der MFCC-Berechnung . . . . .	24
3.8. Filterbank zur Berechnung des Mel-Spektrums . . . . .	24
3.9. MFCCs . . . . .	25
4.1. Abbildung zweier möglicher Hyperebenen . . . . .	27
4.2. Optimal trennende Hyperebene . . . . .	27
4.3. Projektion in höherdimensionalen Raum . . . . .	30
4.4. DAG zur Klassifikation . . . . .	32
5.1. Ablauf Merkmalsberechnung . . . . .	35
5.2. Merkmalsextraktion . . . . .	37
5.3. Trainingsprozess . . . . .	38
5.4. Aufbau DAG zur Testphase . . . . .	39
6.1. Erkennungsraten Experiment 1 . . . . .	42
6.2. Erkennungsraten Experiment 2 . . . . .	44
6.3. Erkennungsraten Experiment 3 . . . . .	46
6.4. Erkennungsraten Experiment 4 . . . . .	48
6.5. Erkennungsraten Experiment 5 . . . . .	50
6.6. Erkennungsraten Experiment 6 . . . . .	52
6.7. Erkennungsraten Experiment 7 . . . . .	54
6.8. Gesamterkennungsraten der einzelnen Experimente . . . . .	56
A.1. Screenshot Octavestart . . . . .	62
A.2. Screenshot Programmstart . . . . .	63



# 1. Einleitung

Der Hörsinn ist der differenzierteste Sinn des Menschen. Das gesunde menschliche Gehör kann Frequenzen im Bereich von 16 Hz bis 20 kHz hören und zwischen zehn Oktaven unterscheiden. Daher fällt es einem Menschen nicht schwer verschiedene Instrumente anhand ihres Klangs zu unterscheiden. Ein geübter Hörer kann dies sogar bei Instrumenten desselben Instrumententyps. Selbst wenn viele Instrumente in einem Orchester zusammenspielen, ist es möglich die einzelnen Instrumente bzw. Instrumententypen zu erkennen.

Die Klassifizierung von Musikinstrumenten mit Hilfe eines Computers stellt eine ungleich größere Herausforderung dar. Es ist für einen Computer nahezu unmöglich Instrumente zu unterscheiden, solange nicht durch Vorverarbeitung der Audiodateien Merkmale gefunden werden, die den Klang eines Instruments eindeutig beschreiben. Daher ist die Extraktion der geeigneten Merkmale der wichtigste Schritt bei der Audioklassifikation.

Diese Arbeit beschäftigt sich mit der Klassifikation von Audiosignalen, am Beispiel dreier Akustikgitarren. Da es sich um Instrumente des gleichen Typs handelt, müssen Merkmale gefunden werden, die das Instrument an sich beschreiben und unabhängig von der Spielweise des jeweiligen Musikers sind. Zusätzlich dazu muss die Gitarre auch bei veränderten Aufnahmebedingungen erkannt werden.

Als Klassifikationsalgorithmus werden Support Vector Machines (SVMs) eingesetzt, da diese bereits in vielen Arbeiten benutzt wurden und gute Ergebnisse geliefert haben. Sie gehören dank zahlreicher Weiterentwicklungen zu einem viel genutzten Werkzeug im Bereich des maschinellen Lernens.

## 1.1. Ähnliche Arbeiten

Es gibt bereits eine Reihe von Arbeiten, die sich mit der Audioklassifikation, insbesondere mit der Instrumentenklassifikation beschäftigen. So haben Marques und Moreno [MM99] acht unterschiedliche Instrumente (Dudelsack, Klarinette, Flöte, Cembalo, Orgel, Klavier, Posaune und Oboe) klassifiziert. Sie verglichen zwei verschiedene Klassifikationsalgorithmen (Gaussian Mixture Modell und SVMs) und drei Merkmalsextraktionsverfahren miteinander. Mit Hilfe der Support Vector Machines erreichten sie dabei eine Erkennungsrate von 70%.

Auch Deng u. a. [DSC06] setzten verschiedene Klassifikationsverfahren, unter anderem auch Support Vector Machines, ein, um verschiedene Instrumententypen zu klassifizieren. Die Instrumente wurden in vier Instrumententypen (Blechblas-, Holzblas-, Saiteninstrumente und Klavier) aufgeteilt und klassifiziert. Auch hierbei zeigte sich, dass SVMs sehr gute Klassifikationsergebnisse (Erkennungsraten von 97%) lieferten.

Steelant u. a. [STD<sup>+</sup>04] setzten SVMs ein, um Schlaginstrumente (Snare-Drum, Bass-Drum, Hihat, Becken und Tom-Tom) zu klassifizieren. Sie stellten Untersuchungen mit verschiedenen SVMs, d.h. zwei verschiedenen Kernen<sup>1</sup> an und erzielten durchgehend Erkennungsraten von über 75%.

## 1.2. Problemstellung

Ziel der vorliegenden Arbeit ist es drei Akustikgitarren von verschiedenen Gitarrenbauern (Hense, Marin und Wichmann) zu klassifizieren.

Die zur Klassifikation verwendeten Audiodateien lagen bereits vor, es handelt sich hierbei um einzeln gespielte Noten. Die Noten wurden dabei auf vorgegebenen Saiten und Lagen der Gitarre gespielt. Jede der drei Gitarren wurde jeweils von vier Gitarristen gespielt. Die Musiker hatten sich dabei an Klangvorgaben ('sonor', 'spitz' und 'warm') zu halten, die ihnen auf dem Notenblatt vorgegeben wurden. Es sollte zusätzlich zu den Klangvorgaben keine weitere Spielart, z.B. Vibrato, benutzt werden. So soll gewährleistet werden, dass ein möglichst großes Spektrum an Klängen vorhanden ist, ohne dass die angespielten Noten zu stark verändert werden. Jeder Ton wurde dabei viermal gespielt, so dass eine ausreichend große Trainings- und Testmenge vorhanden ist.

Da jeder Musiker eine individuelle Spielweise hat, wird die Trainingsmenge so aufgebaut, dass von jedem Spieler Klangdateien vorhanden sind. Damit wird gewährleistet, dass die Gitarre und nicht die Spielweise des Musikers klassifiziert wird.

## 1.3. Gliederung dieser Arbeit

Kapitel 2 beschäftigt sich mit dem Aufbau des Klangs einer Gitarre. Die aufgenommenen Klänge werden nach unterschiedlichen Gesichtspunkten auf Merkmale hin untersucht, die für die Klassifikation in Frage kommen. Dabei werden verschiedene mathematische Verfahren benutzt, um die Merkmale zu extrahieren. In Kapitel 3 werden diese Merkmalsextraktionsverfahren erklärt. Um einen guten Überblick zu bekommen, welche Merkmale für die

---

<sup>1</sup>Kernel werden in Kapitel 4 Support Vector Machines näher beschrieben.

Klassifikation geeignet sind, werden mehrere Testläufe mit verschiedenen Merkmalsvektoren vorgenommen. Die Ergebnisse der einzelnen Testläufe werden in Kapitel 6 aufgezeigt und diskutiert.

Kapitel 4 gibt eine kurze Einführung in die Support Vector Machines. Kapitel 5 zeigt die Realisierung der Merkmalsextraktionsverfahren, sowie den Aufbau der Trainings- und Testphase auf.

## 2. Aufbau des Gitarrenklangs

Um geeignete Merkmale für die Klassifikation bestimmen zu können, ist es notwendig den Klang einer Gitarre zu analysieren. Dazu wird in diesem Teil der Arbeit der Aufbau des Gitarrenklangs beschrieben und erörtert, welchen Einfluss der Spieler und die Bauweise des Instruments auf den Klang haben. Es wird dargestellt, was den Klang einer Gitarre einzigartig macht und von Klängen anderer Gitarren abgrenzt.

### 2.1. Aufbau eines Klangs aus Grundton und Obertönen

In physikalischer Hinsicht ist ein Ton eine einzelne Sinusschwingung, die durch die Parameter Frequenz (Schwingungen pro Sekunde) und Amplitude beschrieben werden kann. Die Frequenz gibt dabei die Tonhöhe, die Amplitude die Lautstärke an.

Ein Klang dagegen ist eine Überlagerung mehrerer Sinusschwingungen, d. h. mehrerer Töne mit verschiedenen Frequenzen und Amplituden. Die Schwingung mit der niedrigsten Frequenz bezeichnet man dabei als Grundfrequenz oder Grundton. Die Schwingungen, die ganzzahlige Vielfache der Grundfrequenz sind, bezeichnet man als Obertöne. Grundton und Obertöne zusammen werden auch als Harmonische bzw. als Teiltöne bezeichnet, wobei der Grundton als 1. Harmonische bzw. 1. Teilton bezeichnet wird. Im Folgenden wird von Harmonischen gesprochen, wenn Grundton und Obertöne gemeinsam gemeint sind, ansonsten von Grundton und Obertönen.

Abbildung 2.1 zeigt das Spektrum eines Gitarrenklangs. Hervorgehoben sind dabei die Harmonischen.

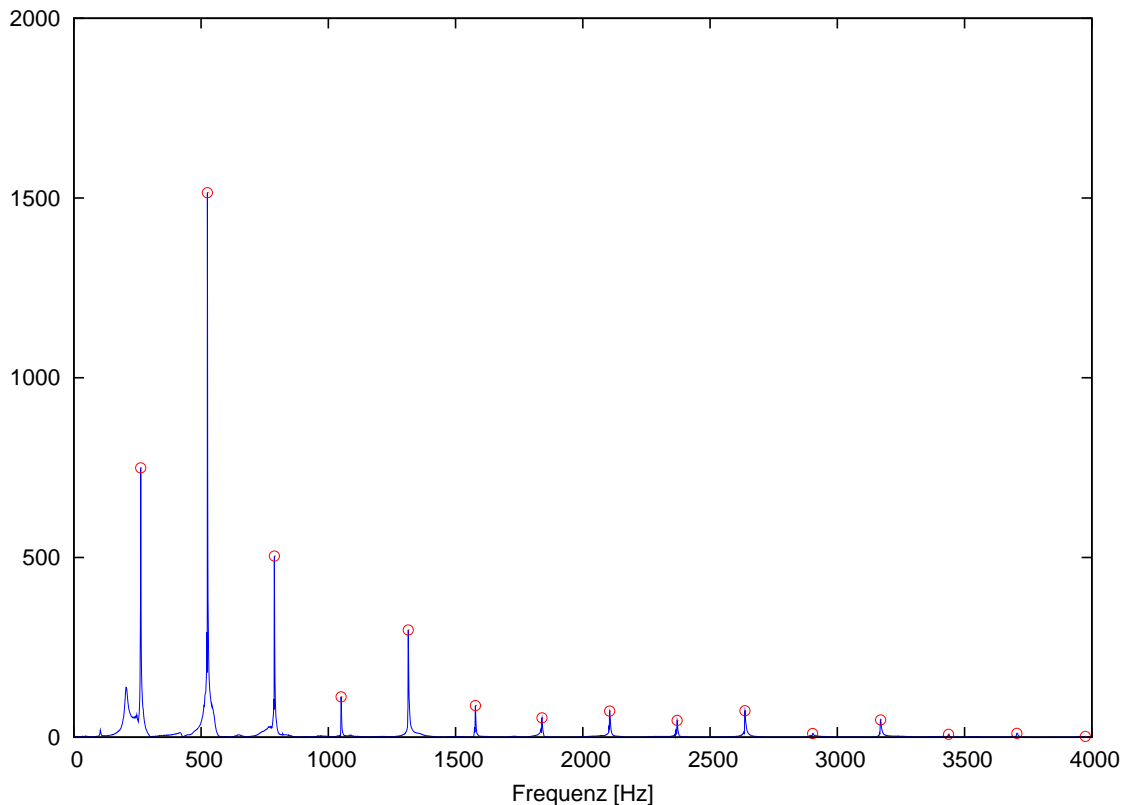


Abbildung 2.1.: Spektrum eines Gitarrenklangs: Grundfrequenz und Obertöne sind hervorgehoben. Grundfrequenz: 263 Hz

Zu sehen ist in Abbildung 2.1, dass der Klang nicht nur aus Grundton und Obertönen besteht, sondern darüber hinaus noch Schwingungen vorhanden sind, die keine ganzzahligen Vielfachen der Grundfrequenz sind. Handelt es sich hierbei um periodische Schwingungen, spricht man von unharmonischen Obertönen [Mey85].

## 2.2. Klangfarbe

Die Klangfarbe eines Instruments, also der charakteristische Klang, ist durch das Verhältnis der Stärke der einzelnen Harmonischen zueinander bestimmt [Mey85]. Wichtig ist dabei die Zusammensetzung der Harmonischen und deren Intensität. Dieses Verhältnis wird zum Teil durch das Instrument selbst (Bauweise und Material) bestimmt, ist aber auch vom Spieler abhängig. So kann der Musiker durch die Anschlagposition der Saite den Klang maßgeblich beeinflussen. Schlägt der Spieler z.B. die Saite bei  $\frac{1}{4}$  der Saitenlänge an, so werden alle Obertöne deren Ordnungszahl durch 4 teilbar ist unterdrückt.

Klänge mit der Klangvorgabe 'warm' werden in einer Spielposition erzeugt, die zwischen Schallloch und Hals der Gitarre liegt. Klänge, die mit 'spitz' charakterisiert werden, werden hingegen in Stegnähe erzeugt [Voi07]. Die Abbildung 2.2 zeigt die Klangspektren der Gitarre Marin für die Klangvorgaben 'warm' und 'spitz'. Dabei ist zu sehen, dass bei der Klangvorgabe 'spitz' die höheren Frequenzen eine höhere Intensität haben, als bei der Klangvorgabe 'warm'. Bei dieser sind Frequenzen ab 1000 Hz nur noch mit einer sehr kleinen Amplitude vorhanden.

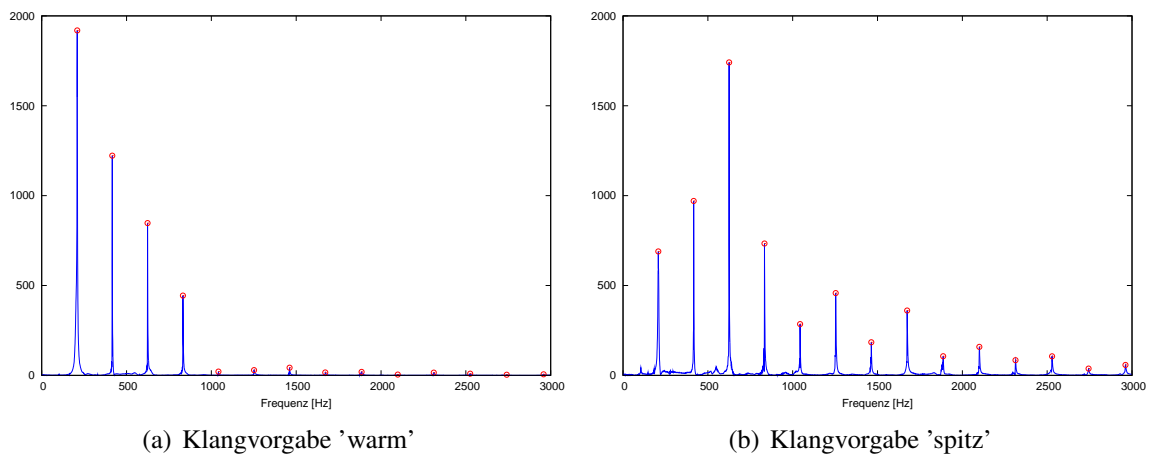


Abbildung 2.2.: Abhängigkeit des Klangspektrums von der Anschlagposition. Grundfrequenz: 207 Hz

Auch Formanten haben einen wesentlichen Einfluss auf die Klangfarbe der Gitarre. Formanten sind besonders ausgeprägte Frequenzbereiche, die im Wesentlichen von der Bauform und somit vom Resonanzverhalten der Gitarre abhängen.

Des Weiteren hängen die Ein- und Ausschwingzeiten der Harmonischen mit dem Resonanzverhalten des Gitarrenkorpus zusammen. Somit haben auch die zeitlichen Verläufe der Harmonischen einen Einfluss auf die Klangfarbe der Gitarre.

## 2.3. Resonanzverhalten der Gitarre

Das Resonanzverhalten der meisten Gitarren wird im unteren Frequenzbereich durch zwei ausgeprägte Einzelresonanzen bestimmt. Diese Einzelresonanzen haben in der Regel einen Frequenzabstand von einer Oktave. Dabei liegt die untere Resonanz im Bereich von 100 bis 140 Hz und die obere Resonanz zwischen 200 und 280 Hz [Mey85]. Aber auch in den höheren Frequenzlagen weist der Gitarrenkorpus Resonanzen auf. Für die Klangfarbe

des Instruments spielen diese Resonanzen eine bedeutende Rolle. So hängen die Intensität und die zeitlichen Verläufe der Harmonischen von ihrer Lage zu den Resonanzen ab. Je näher die Frequenzen der Harmonischen an den Resonanzspitzen liegen, desto höher ist der abgestrahlte Schallpegel und desto kürzer ist die Ausklingzeit. Dies rührt daher, dass der Saite die Schwingungsenergie besonders schnell entzogen wird.

Das Resonanzverhalten der Gitarre wird durch die Hohlraumresonanz und die Eigenresonanzen der Gitarrendecke und des Gitarrenbodens bestimmt. In Abbildung 2.3 sind, als Beispiel, die verschiedenen Resonanzbereiche der Gitarrendecke bei verschiedenen Frequenzen zu sehen.

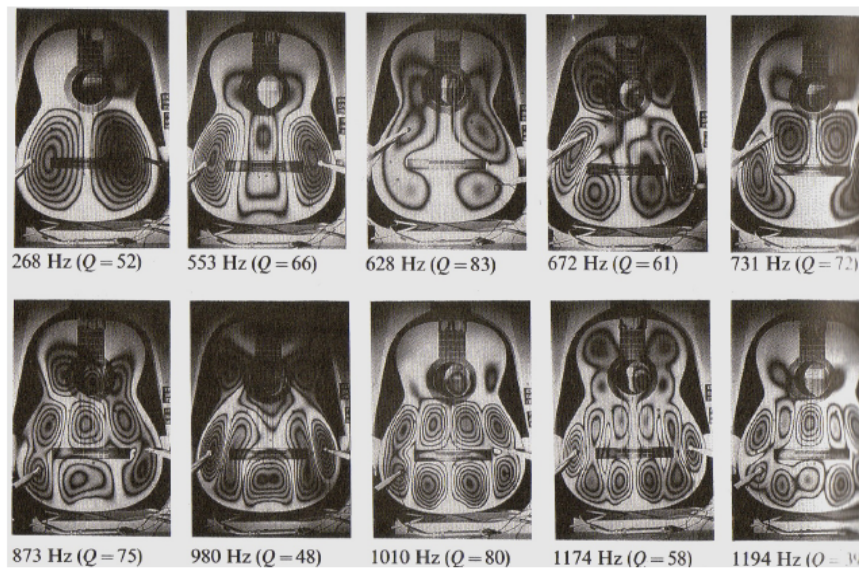


Abbildung 2.3.: Resonanzbereiche der Gitarren bei verschiedenen Frequenzen. Abbildung aus [FR98].

# 3. Merkmalsextraktion

## 3.1. Bestimmung der Grundfrequenz

Die Grundfrequenz wird mit Hilfe der Cepstrum-Analyse bestimmt. Der Begriff Cepstrum wird durch Vertauschen der ersten vier Buchstaben des Worts Spectrum gebildet. Das Cepstrum  $C(q)$  wird durch die inverse Fouriertransformation des logarithmierten Spektrums eines Signals  $s(t)$  ermittelt

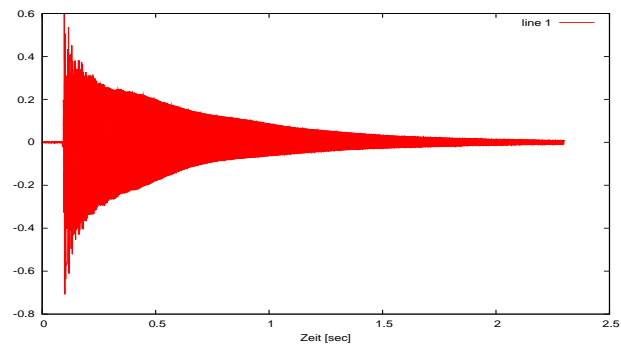
$$C(q) = IFFT(\log|FFT(s(t))|). \quad (3.1)$$

Die Logarithmierung des Spektrums hat zur Folge, dass auch relativ kleine Amplituden deutlich erkennbar sind. Die periodische Gestalt des logarithmierten Spektrums zeigt sich im Cepstrum durch ein ausgeprägtes Maximum. Dieses Maximum spiegelt die Grundfrequenz wieder.

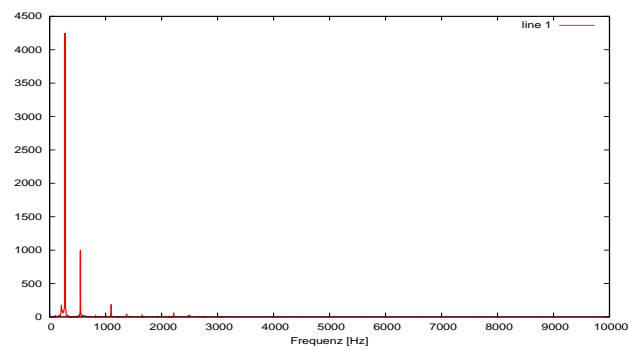
Die Ursache für die Entstehung dieses Maximums liegt darin, dass nur der Betrag des Spektrums in der Verarbeitung verwendet wird. Das heißt, bei der Rücktransformation in den Zeitbereich werden die Phaseninformationen unterdrückt. Die Phase ist jedoch für die Positionierung der einzelnen Teilschwingungen im Zeitsignal verantwortlich. Ist das ursprüngliche Signal eine Folge äquidistanter Impulse, so stellt auch das Cepstrum wieder eine Impulsfolge dar. Darüber hinaus werden die Abstände der aufeinanderfolgenden Impulse im Cepstrum mit denen im ursprünglichen Zeitsignal übereinstimmen. Da jedoch die Phaseninformationen unterdrückt wurden, liegt der erste Impuls des Cepstrums immer an der Stelle  $d = 0$ , mit  $d$  : Verzögerungszeit. Somit muss der nächste Impuls an der Stelle  $d = T_0$  liegen [Hes06]. Mit Hilfe dieser Verzögerungszeit kann die Grundfrequenz berechnet werden.

Abbildung 3.1 zeigt die einzelnen Schritte der Cepstrumberechnung am Beispiel eines Klangs. Dabei ist in Abbildung 3.1(d) nur der für die Grundfrequenzbestimmung relevante Ausschnitt des errechneten Cepstrums zu sehen. Das markante Maximum zur Grundfrequenzberechnung ist hervorgehoben. Es liegt bei einem Indexwert von 160, was bei einer Samplerate von 44100 Hz einer Verzögerungszeit von ca. 3,628 ms entspricht. Damit kann man die Grundfrequenz bestimmen, die in diesem Fall bei 275,62 Hz liegt.

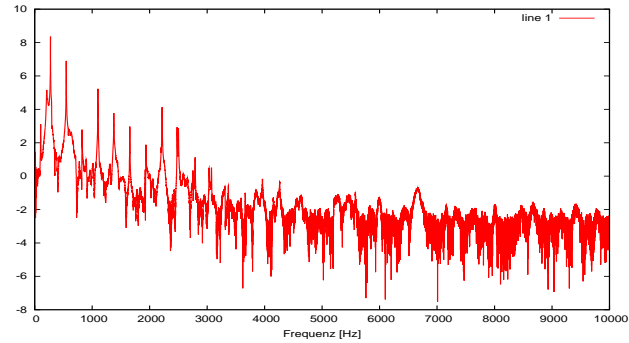




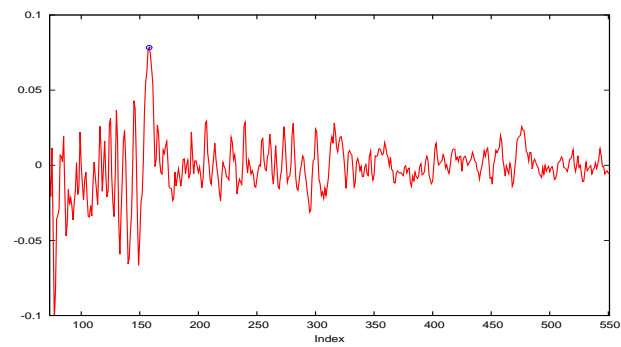
(a) Zeitlicher Verlauf des Klangs



(b) Spektrum des Klangs



(c) Logarithmiertes Spektrum des Klangs



(d) Cepstrum des Klangs

Abbildung 3.1.: Einzelne Schritte der Cepstrumberechnung.

Die Grundfrequenz wird als zusätzliches Merkmal in den Merkmalsvektor aufgenommen. Sie dient dabei nicht der Unterscheidung der einzelnen Gitarren, sondern hilft die nachfolgenden Merkmale den einzelnen Tonbereichen zuordnen zu können. Da es durchaus möglich ist, dass zwei verschiedene Klänge zweier Gitarren z. B. einen ähnlichen Verlauf der Harmonischen haben, kann die Grundfrequenz als zusätzliches Entscheidungsmerkmal dienen.

## 3.2. Zeitlicher Verlauf der Harmonischen

Während die Amplituden der Grundfrequenz und der Obertöne von der Spielweise des Musikers, d. h. von der Anschlagposition auf der Saite, abhängen, ist der zeitliche Verlauf der Harmonischen maßgeblich vom Resonanzverhalten des Korpus abhängig. Dieses wiederum hängt von der Bauweise der Gitarre und von den verwendeten Hölzern ab.

Abbildung 3.2 zeigt den Ablauf zur Bestimmung des zeitlichen Verlaufs der Harmonischen. Das Signal wird dazu in Blöcke der Länge 4096 Samples unterteilt, was in etwa 100 ms entspricht. Jeder dieser Blöcke wird mit einem Blackman-Fenster<sup>1</sup> multipliziert und auf das gefenstertere Signal die diskrete Fourier-Transformation (DFT) angewendet.

Innerhalb eines Toleranzbereichs um die erwarteten Indizes der Harmonischen wird nach dem Maximum gesucht, um so die maximale Auslenkung der Peaks zu ermitteln. Der so bestimmte Wert wird in eine Matrix geschrieben. Diese hält die Werte der ermittelten Harmonischen über den zeitlichen Verlauf des Signals.

---

<sup>1</sup>Siehe Glossar: Fensterfunktion.

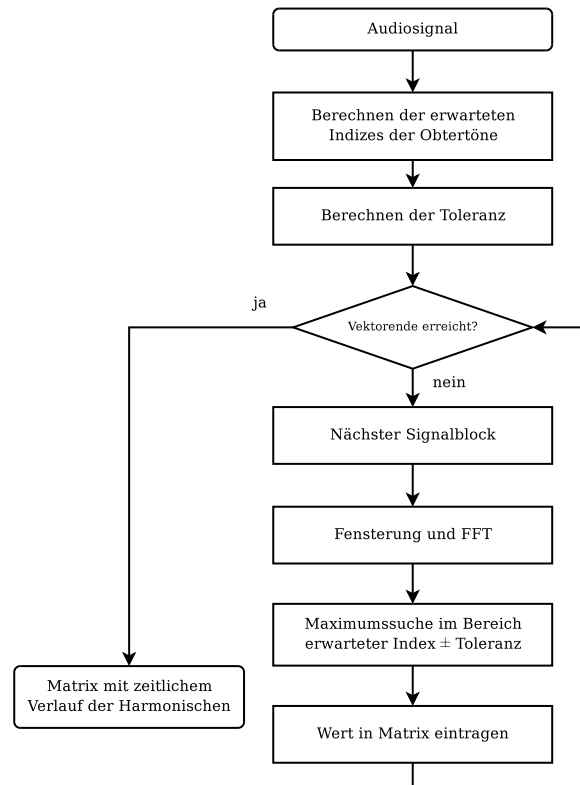
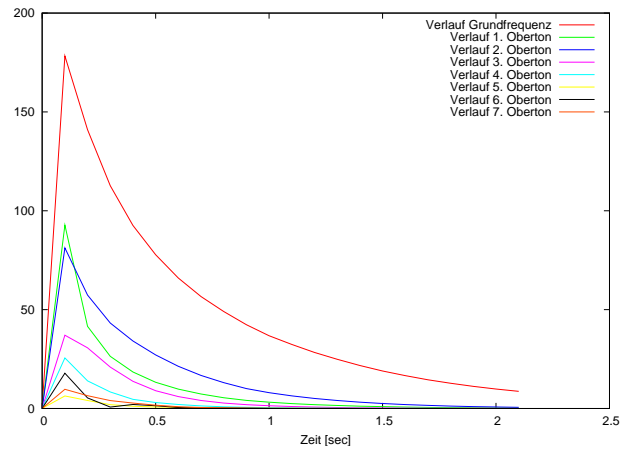
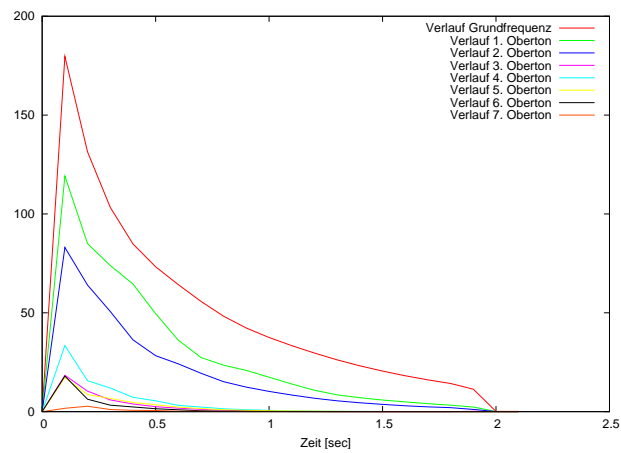


Abbildung 3.2.: Flussdiagramm zur Berechnung des zeitlichen Verlaufs der Harmonischen.

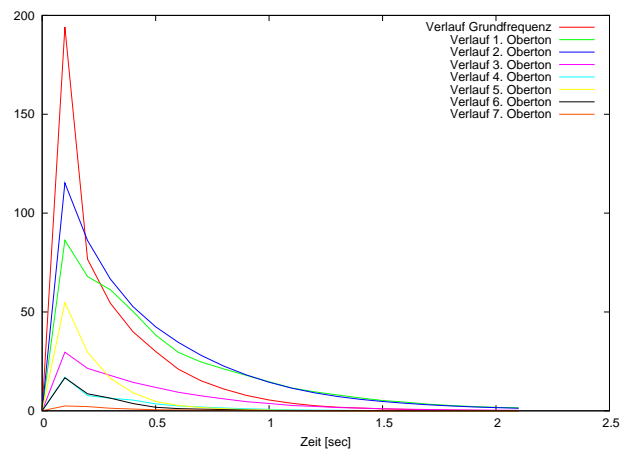
Abbildung 3.3 zeigt als Beispiel den zeitlichen Verlauf der Harmonischen für die gespielte Note f' (Klangvorgabe 'sonor'). Wie zu sehen ist, klingt der Grundton bei der Gitarre Wichmann sehr viel schneller ab als bei den Gitarren Hense und Marin. Des Weiteren ist zu erkennen, dass der erste Oberton der Gitarre Marin eine höhere Amplitude aufweist als bei den beiden anderen Gitarren. Da die zeitlichen Verläufe der Harmonischen unterschiedliche Merkmale aufweisen, können sie als Merkmale in die Klassifikation der drei Gitarren eingehen.



(a) Gitarre: Hense



(b) Gitarre: Marin



(c) Gitarre: Wichmann

Abbildung 3.3.: Zeitlicher Verlauf der Harmonischen der drei Gitarren. Grundfrequenz: 349 Hz.

### 3.3. Bestimmung des nontonalen Spektrums

Da nicht nur die Harmonischen einen Einfluss auf den Klang einer Gitarre haben, sondern auch das Resonanzverhalten des Korpus, wird zusätzlich das nontonale<sup>2</sup> Spektrum als Merkmal aufgenommen. Fragoulis u. a. [FPE<sup>+</sup>06] haben in ihrer Arbeit über die Klassifikation von Gitarre und Klavier verschiedene Versuche mit den Frequenzspektren von Gitarren- und Klavierklängen vorgenommen. Sie stellten dabei fest, dass der typische Klangcharakter eines Instruments verloren geht, wenn man die nontonalen Frequenzkomponenten des Klangspektrums entfernt und das so ermittelte tonale Spektrum in den Zeitbereich zurücktransformiert. Sie stellten dadurch die Vermutung an, dass dieses nontonale Spektrum maßgeblich für den Klangcharakter eines Instruments verantwortlich ist.

Der Ablauf der Berechnung der nontonalen Merkmale ist in Abbildungen 3.4 dargestellt.

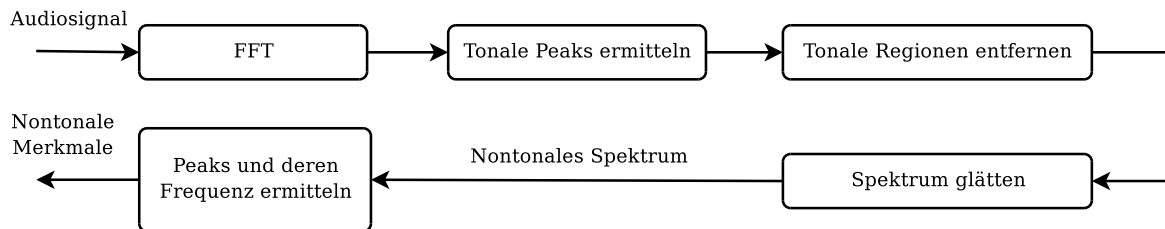
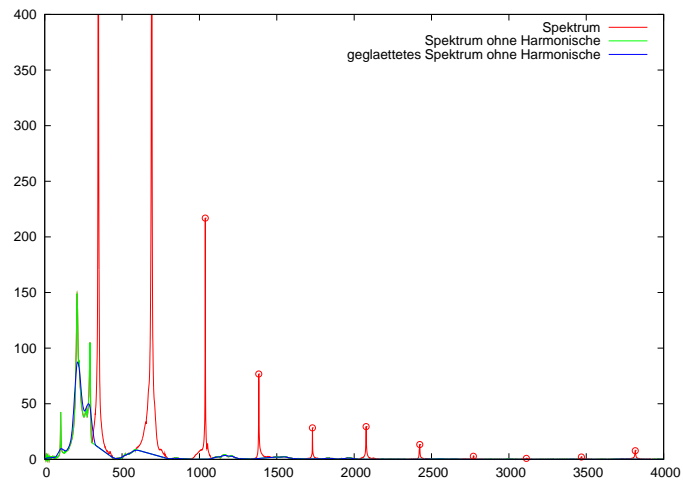


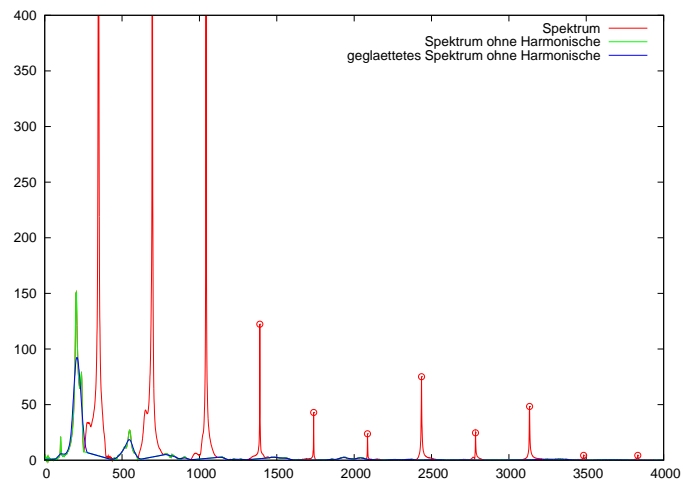
Abbildung 3.4.: Ablauf zur Berechnung der nontonalen Merkmale.

Zur Bestimmung des nontonalen Spektrums werden anhand der Grundfrequenz die Frequenzen der Obertöne berechnet. Der zu entfernende Bereich um die Peaks der Harmonischen wird in Abhängigkeit zur Grundfrequenz bestimmt und anschließend die Harmonischen entfernt, d.h. die Werte des Vektors werden auf Null gesetzt. Um die Peaks des nontonalen Spektrums ermitteln zu können, wird das so erhaltene nontonale Spektrum geglättet. Als nontonale Merkmale werden nun die Amplituden der Peaks und die zugehörigen Frequenzen der Peaks des nontonalen Spektrums ermittelt. Abbildung 3.5 zeigt als Beispiel die nontonalen Spektren im Bereich von 0 bis 4000 Hz der drei Gitarren für die gespielte Note f'. Wie man sehen kann, weisen die Gitarren gerade im unteren Frequenzbereich voneinander abweichende nontonale Spektren auf.

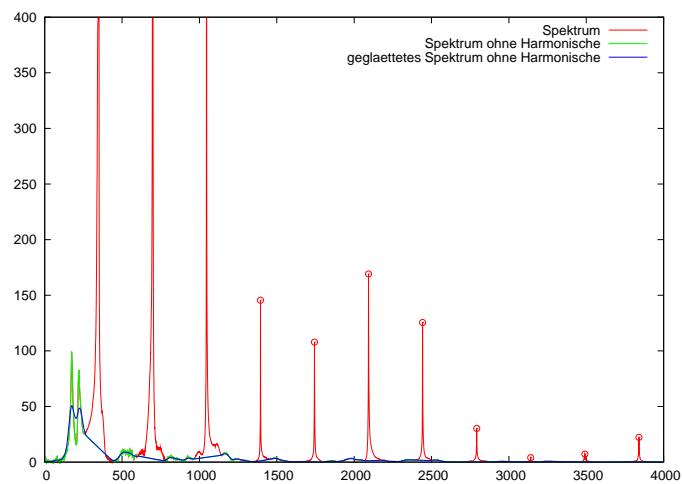
<sup>2</sup>Der Begriff 'nontonal' wird im Folgenden benutzt, wenn von den Frequenzkomponenten gesprochen wird, die zwischen den Peaks der Harmonischen hervortreten.



(a) Gitarre: Hense



(b) Gitarre: Marin



(c) Gitarre: Wichmann

Abbildung 3.5.: Nontonale Spektren der Gitarren Hense, Marin und Wichmann. Grundfrequenz: 349 Hz.

## 3.4. Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients (MFCCs) wurden ursprünglich in der Spracherkennung eingesetzt, sind jedoch auch geeignete Merkmale zur Audioklassifikation. So haben Marques und Moreno [MM99] bei der Instrumentenklassifikation die besten Erkennungsraten mit MFCCs als Merkmale erzielt. MFCCs sind Kennzahlen, die etwas über die Klangfarbe des Audiosignals zu einem bestimmten Zeitpunkt aussagen.

### 3.4.1. Mel

Mel, mit dem Formelzeichen  $Z$ , ist die Einheit für die psychoakustische Größe der Tonheit<sup>3</sup> und beschreibt die vom Menschen wahrgenommene Tonhöhe. Basis für die Mel-Skala ist dabei eine Frequenz von 1000 Hz, der die Tonheit  $Z = 1000$  mel zugeordnet wird. Die Skala ist dabei so aufgebaut, dass einem doppelt so hoch wahrgenommenen Ton ein doppelter mel-Wert zugewiesen wird. Abbildung 3.6 zeigt den Zusammenhang zwischen Frequenz und wahrgenommener Tonheit auf.

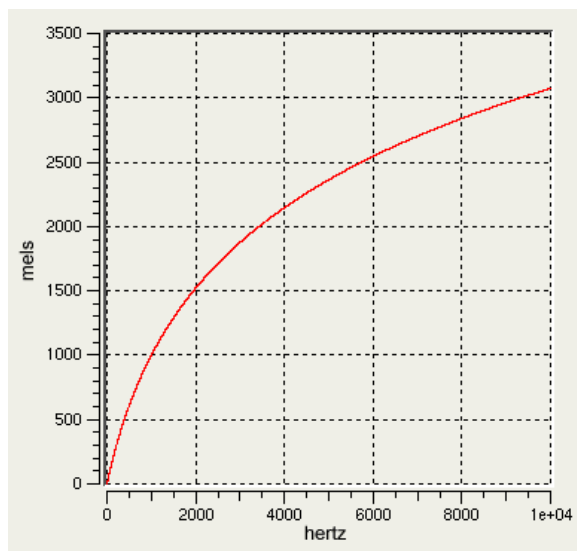


Abbildung 3.6.: Zusammenhang zwischen Frequenz und wahrgenommener Tonheit. Abbildung aus [Wik].

Wie in der Abbildung zu sehen ist, stehen ab einer Frequenz von 1000 Hz Frequenz und mel nicht mehr in linearem Zusammenhang. Dies rührt daher, dass das menschliche Gehör hohe Frequenzen schlechter auflösen kann als niedrige.

<sup>3</sup>Zum hier verwendeten Begriff der Tonheit siehe Glossar.

### 3.4.2. Berechnung der Mel Frequency Cepstral Coefficients

Um die MFCCs berechnen zu können (Abbildung 3.7 zeigt den Ablauf der Berechnung), wird zuerst das Eingangssignal in Segmente von ca. 20 ms unterteilt und mit einer geeigneten Fensterfunktion (i. d. R. wird das Hamming-Fenster benutzt) multipliziert. Auf jedes dieser Segmente wird die DFT angewendet. Um das Mel-Spektrum zu errechnen, wird auf das fouriertransformierte Signal der Mel-Filter angewendet. Dieser modelliert das menschliche Gehör. Der Mel-Filter ist eine Filterbank, bestehend aus sich überlappenden Dreiecksfiltern, die für niedrigere Frequenzen eine geringere Breite besitzen als für höhere (Abbildung 3.8).

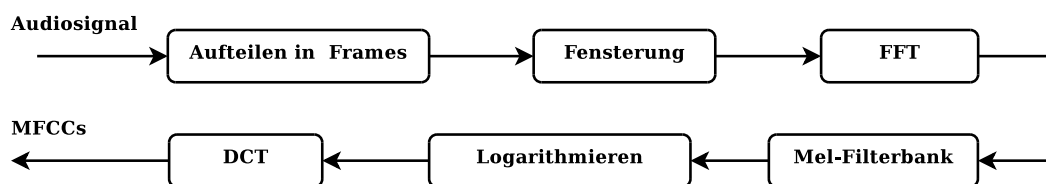


Abbildung 3.7.: Prozess der MFCC-Berechnung.

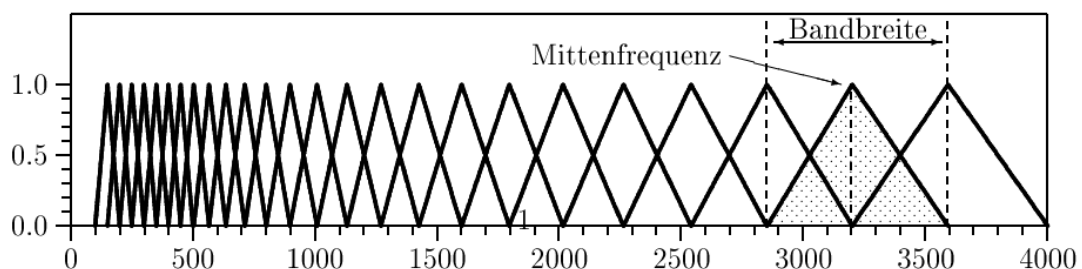


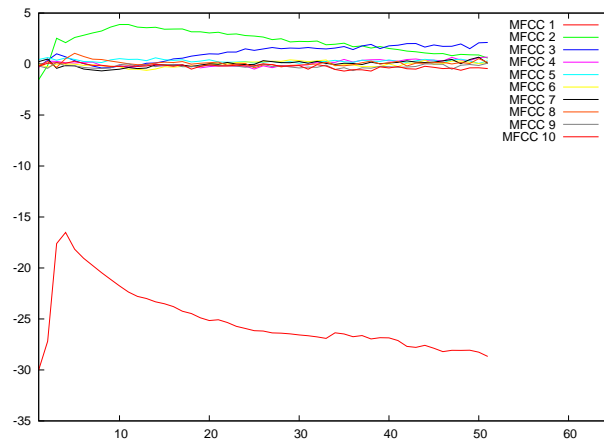
Abbildung 3.8.: Filterbank zur Berechnung des Mel-Spektrums. Abbildung aus [Ste03].

Das so erhaltene Mel-Spektrum wird logarithmiert und anschließend einer diskreten Cosinus-Transformation unterworfen. Dies führt zu einer Dekorrelation der Koeffizienten. Das Ergebnis dieser Berechnungen sind die Mel Frequency Cepstral Coefficients.

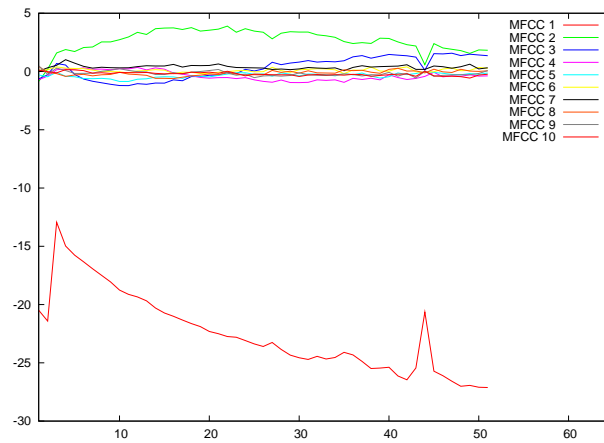
Dabei stellt der erste Koeffizient die durchschnittliche Energie des Spektrums zu einem Zeitpunkt dar. Die höheren Koeffizienten repräsentieren die Details der spektralen Form.

Abbildung 3.9 zeigt die ersten zehn MFCCs der drei Gitarren. Wie zu sehen ist, weisen die Gitarren gerade bei den ersten drei MFCCs erhebliche Unterschiede auf, aber auch die oberen MFCCs unterscheiden sich über den Verlauf des Signals so, dass sie als Merkmale aufgenommen werden.

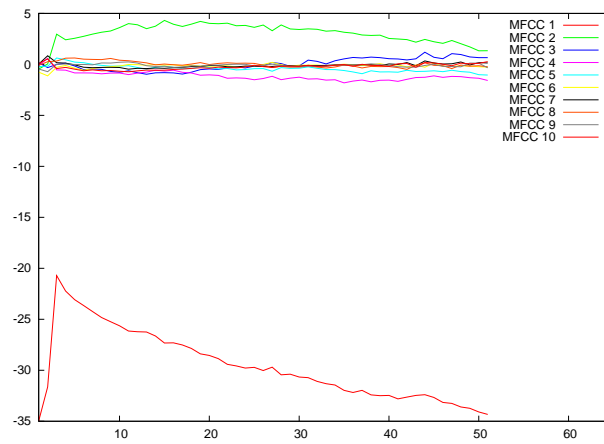




(a) Gitarre: Hense



(b) Gitarre: Marin



(c) Gitarre: Wichmann

Abbildung 3.9.: Die ersten zehn MFCCs der drei Gitarren.

# 4. Support Vector Machines

Dieses Kapitel gibt eine kurze Einführung in Support Vector Machines (SVMs). Einen detaillierteren Einblick in dieses Klassifikationsverfahren bieten Bernhard Schölkopf und Alexander J. Smola [SS02], sowie Nello Cristianini und John Shawe-Taylor [CST00].

Support Vector Machines gehören zu den statistischen Lernverfahren und sind maßgeblich von Vladimir N. Vapnik [Vap98, Vap00] entwickelt worden. Die Grundidee der Support Vector Machines liegt darin eine optimale Trennebene (die Hyperebene) zwischen den zu trennenden Klassen zu finden. Sind die Daten linear trennbar, ist dies einfach. Sind die Daten so ineinander verschränkt, dass eine lineare Trennung nicht möglich ist, wird auf den sogenannten Kernel-Trick zurückgegriffen, der es ermöglicht auch hier die optimale Trennebene zu ermitteln. Im nachfolgenden Teil wird zuerst das Verfahren für linear trennbare Daten beschrieben, der zweite Teil beschreibt das Verfahren für den nicht linear trennbaren Fall.

## 4.1. Lineare SVMs

Gegeben ist eine Menge von Merkmalsvektoren  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in X$  mit den zugehörigen Klassenlabeln  $y_1, y_2, \dots, y_n$  mit  $y_i \in \{+1, -1\}$ . Jede Hyperebene, die diese zwei Klassen voneinander trennt, kann durch

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0, \quad \mathbf{x}, \mathbf{w} \in X, b \in \mathbb{R} \quad (4.1)$$

mit  $\mathbf{w}$ : Normalenvektor der Ebene,

$b$ : konstanter Term, der den Abstand der Hyperebene zum Ursprung bezeichnet,

beschrieben werden.

Da es viele mögliche Trennebenen gibt (Abb. 4.1), welche die Klassen voneinander trennen, wird auf beiden Seiten der Ebene ein Rand, der sogenannte Margin, eingeführt und dieser solange verbreitert, bis er Merkmalsvektoren der beiden Klassen berührt. Diese Merkmalsvektoren werden Supportvektoren genannt. Die beste Hyperebene ist nun diejenige, die den breitesten Rand besitzt (Abb. 4.2).

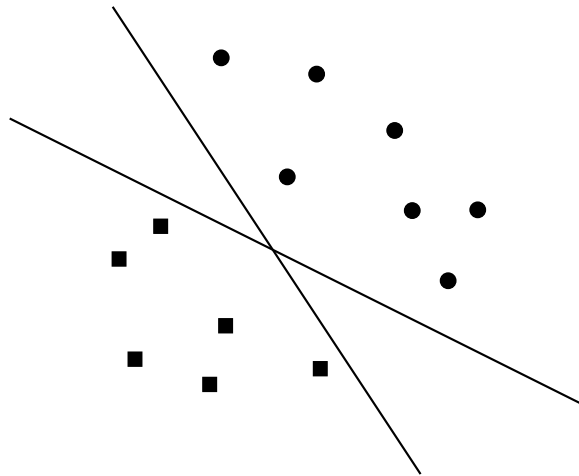


Abbildung 4.1.: Abbildung zweier möglicher Hyperebenen.

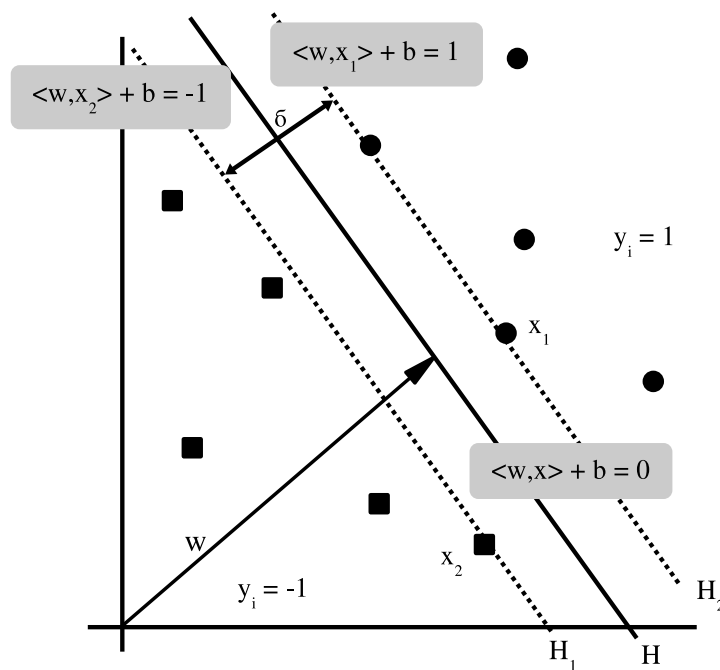


Abbildung 4.2.: Optimal trennende Hyperebene.

Man kann  $w$  und  $b$  so skalieren, dass für die Supportvektoren gilt

$$|\langle w, x \rangle + b| = 1. \quad (4.2)$$

Die Distanz  $\delta$  zwischen den Ebenen  $H_1$  und  $H_2$  ist gegeben durch

$$\begin{aligned}\langle \mathbf{w}, \mathbf{x}_1 \rangle + b &= 1 \\ \langle \mathbf{w}, \mathbf{x}_2 \rangle + b &= -1 \\ \Rightarrow \langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle &= 2 \\ \Rightarrow \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle &= \frac{2}{\|\mathbf{w}\|}\end{aligned}$$

Um die optimale Hyperebene bestimmen zu können, müssen folgende Bedingungen erfüllt sein:

- Es gilt die Distanz  $\delta$  zwischen den Ebenen  $H_1$  und  $H_2$  zu maximieren, was gleichbedeutend ist mit einer Minimierung von  $\|\mathbf{w}\|$
- Zwischen den Ebenen  $H_1$  und  $H_2$  befinden sich keine Merkmalsvektoren

$$\begin{aligned}\langle \mathbf{w}, \mathbf{x}_i \rangle + b &\geq 1 && \text{für alle } y_i = 1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b &\leq -1 && \text{für alle } y_i = -1\end{aligned}$$

Diese beiden Bedingungen lassen sich zu einer zusammenfassen

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, \dots, n$$

Aus den genannten Bedingungen lässt sich folgendes Optimierungsproblem<sup>1</sup> ableiten

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \tag{4.3}$$

unter der Nebenbedingung

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, \dots, n. \tag{4.4}$$

Dieses Optimierungsproblem lässt sich mit dem Lagrange-Ansatz lösen

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^m \alpha_i (y_i(\langle \mathbf{w}, \vec{x}_i \rangle + b) - 1) \tag{4.5}$$

mit den Lagrange'schen Multiplikatoren  $\alpha_i \geq 0$ .

<sup>1</sup>Aufgrund des Monotonieverhaltens kann statt  $\|\mathbf{w}\|$  auch  $\|\mathbf{w}\|^2$  minimiert werden [Zit06].

Daraus ergibt sich, um ein unbekanntes Datum  $\mathbf{x}_{\text{neu}}$  zu klassifizieren, folgende Entscheidungsfunktion

$$f(\mathbf{x}_{\text{neu}}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}_{\text{neu}} \rangle + b\right) \quad (4.6)$$

mit  $n$ : Anzahl der Supportvektoren  
 $\mathbf{x}_i$ : Supportvektor  
 $\alpha_i$ : Lagrange-Multiplikator  
 $y_i$ : Klassenlabel des Supportvektors  $\mathbf{x}_i$

Dabei ist die Lösung nur von den Supportvektoren abhängig. Für diese gilt  $0 < \alpha_i < \infty$ , für alle anderen Merkmalsvektoren  $\alpha_i = 0$ .

Um auch Daten klassifizieren zu können, die einen gewissen Rauschpegel aufweisen, d.h. Daten die geringfügig ineinander verschränkt sind, wird ein Parameter  $C$  eingeführt. Dadurch gilt für die Lagrange'schen Multiplikatoren  $0 \leq \alpha_i \leq C$ . Dieser Parameter lockert die Bedingung, dass sich keine Merkmalsvektoren zwischen den Ebenen  $H_1$  und  $H_2$  befinden dürfen und erlaubt das Abwägen zwischen Randmaximierung und Fehlerminimierung. Je größer  $C$  ist, desto härter werden Fehler 'bestraft'.

## 4.2. Nichtlineare SVMs

Die bisher vorgestellten Support Vector Machines eignen sich nur für Daten die geringfügig ineinander verschränkt sind. Da dies in der Praxis aber kaum anzutreffen ist, muss ein Weg gefunden werden die Daten dennoch linear zu separieren. Die Lösung dieses Problems liegt in der Annahme, dass in einem höherdimensionalen Raum eine lineare Trennebene existiert. Die Merkmalsvektoren  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  werden mit Hilfe einer nichtlinearen Abbildung

$$\begin{aligned} \Phi : X &\rightarrow F \\ \mathbf{x} &\rightarrow \Phi(\mathbf{x}) \end{aligned}$$

in einen höherdimensionalen Raum, den sogenannten Featurespace  $F$ , projiziert, um so die lineare Trennbarkeit zu realisieren (Abb. 4.3).

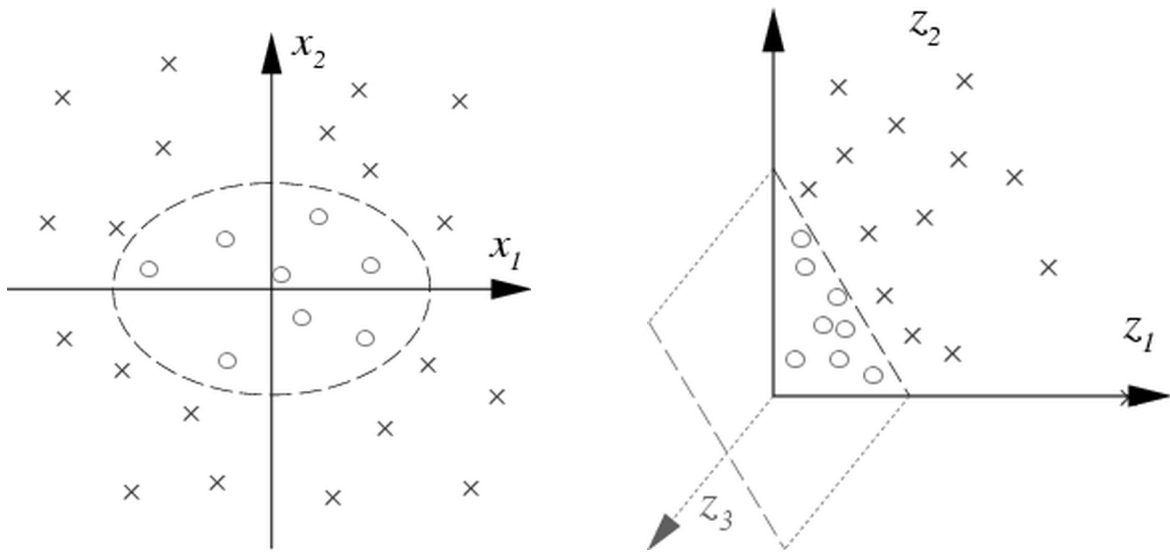


Abbildung 4.3.: Projektion der Daten in einen höherdimensionalen Raum, um eine lineare Trennung zu ermöglichen. Abbildung aus [SS02].

Die Projektion in einen höherdimensionalen Raum ist allerdings sehr kostspielig, d.h. rechenintensiv. Dieses Problem lässt sich dadurch umgehen, dass man das Skalarprodukt  $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_{\text{neu}}) \rangle$  nicht explizit berechnet, sondern stattdessen eine sogenannte Kernel-Funktion  $k$  benutzt, so dass gilt

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_{\text{neu}}) \rangle \quad (4.7)$$

Die Entscheidungsfunktion lautet somit

$$f(\mathbf{x}_{\text{neu}}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) + b\right) \quad (4.8)$$

Es gibt eine Vielzahl von Kernel-Funktionen. Die am häufigsten verwendeten sind

- Linear-Kernel

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = \langle \mathbf{x}_i, \mathbf{x}_{\text{neu}} \rangle \quad (4.9)$$

- Polynomial-Kernel

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = \langle \mathbf{x}_i, \mathbf{x}_{\text{neu}} \rangle^d, \quad d \in \mathbb{N} \quad (4.10)$$

- Inhomogener Polynomial-Kernel

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = (\langle \mathbf{x}_i, \mathbf{x}_{\text{neu}} \rangle + c)^d, \quad d \in \mathbb{N}, c \geq 0 \quad (4.11)$$

- Gauß-Kernel oder RBF-Kernel (Radiale-Basis-Funktion-Kernel)

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_{\text{neu}}\|^2}{2\sigma^2}\right), \quad \sigma > 0 \quad (4.12)$$

- Sigmoid-Kernel

$$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}}) = \tanh(\kappa \langle \mathbf{x}_i, \mathbf{x}_{\text{neu}} \rangle + \vartheta), \quad \kappa > 0, \vartheta < 0 \quad (4.13)$$

Der Linear-Kernel kann nur eingesetzt werden, wenn eine lineare Trennbarkeit zu erwarten ist. Im eigentlichen Sinn ist der Linear-Kernel keine Kernel-Funktion, da das Skalarprodukt verwendet wird. Sind die Daten komplexer, lassen sich in vielen Fällen mit dem RBF- und dem Polynomial-Kernel gute Ergebnisse erzielen.

## 4.3. Multi-Klassen-Probleme

Da sich Support Vector Machines nur für binäre Klassifikationsprobleme eignen, müssen Verfahren entwickelt werden, um SVMs auch für Mehrklassen-Probleme einsetzen zu können. Die zwei am häufigsten verwendeten Verfahren werden hier kurz vorgestellt.

### 4.3.1. One-Versus-One

Bei diesem Verfahren wird jede Klasse gegen jede der anderen Klassen trainiert. Es werden also insgesamt für  $m$  Klassen  $\frac{(m-1)m}{2}$  Support Vector Machines benötigt. Um zu entscheiden, zu welcher Klasse ein unbekanntes Datum gehört, kann ein gerichteter, azyklischer Graph (engl.: Directed Acyclic Graph (DAG)) aufgebaut werden. Abbildung 4.4 zeigt als Beispiel einen Graphen für ein 4-Klassen-Problem.

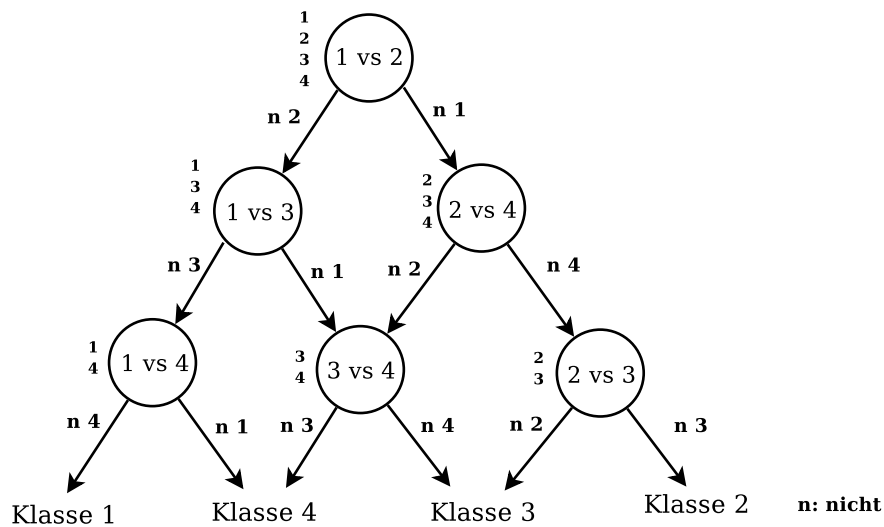


Abbildung 4.4.: DAG zur Klassifikation beim One-Vs-One-Verfahren. Abbildung nach [PCST00].

Diese Verfahren ist vielfach beim Training schneller, da die einzelnen Klassen nur kleine Teilmengen der gesamten Menge repräsentieren. Des Weiteren können, im Gegensatz zum One-Versus-Rest-Verfahren, meist Linear-Kernel eingesetzt werden.

### 4.3.2. One-Versus-Rest

Bei diesem Verfahren wird jede Klasse gegen den Rest, d.h.  $m - 1$  Klassen, trainiert. Es werden also  $m$  SVMs erzeugt. Um zu entscheiden, zu welcher Klasse ein unbekanntes Datum gehört, werden  $m$  Klassifikationen durchgeführt. Als Klasse wird nun diejenige ausgewählt, für welche die Entscheidungsfunktion den maximalen Wert liefert<sup>2</sup>.

<sup>2</sup>Je weiter der Merkmalsvektor vom Margin entfernt liegt, desto größer ist der Betrag des Werts, den die Entscheidungsfunktion liefert.



# 5. Realisierung

## 5.1. Verwendete Software

Für die Merkmalsextraktion wurde Octave<sup>1</sup>, in der Version 2.1, verwendet. Octave ist, wie auch Matlab, eine Skriptsprache zur Berechnung von numerischen Problemen. Octave ist dabei ein unter GPL stehender Matlab-Clone und als solcher weitgehend mit Matlab kompatibel, verfügt jedoch nicht über den gleichen Funktionsumfang.

Für Octave stehen bereits umfangreiche Funktionsbibliotheken zur Signalverarbeitung und -analyse, z. B. Filter-Funktionen, Funktionen zur Berechnung der Fouriertransformation etc., bereit. Somit können relativ einfach Algorithmen zur Merkmalsextraktion entworfen werden.

### Verwendete SVM-Implementierung

Es wurden zwei verschiedene SVM-Implementierungen getestet. Zum einen die Implementierung von Thorsten Joachims [Joa99], SVM<sup>light</sup> in der Version 6.01 und zum anderen die Implementierung der RWTH Aachen<sup>2</sup>. Die Umsetzung der SVM der RWTH Aachen ist dabei in der C++-Bibliothek ltilib<sup>3</sup> vorhanden. Diese stellt Algorithmen und Datenstrukturen zur Verfügung, die hauptsächlich in der Bildmusterverarbeitung Verwendung finden.

Die Entscheidung fiel auf die SVM-Implementierung von Joachims, da diese einige Merkmale aufweist, welche die Verwendung besonders vorteilhaft macht. So wird ausgegeben wie lange die Trainingsphase gedauert hat, sowohl in Sekunden, als auch die Anzahl benötigter Iterationen zur Berechnung der optimalen Hyperebene. Ein weiterer Vorteil ist, dass zusätzlich zur Laufzeit ausgegeben wird, wieviele Supportvektoren berechnet wurden. Des Weiteren wird als erster Anhaltspunkt, wie gut die berechnete Hyperebene die Klassen trennt, ausgegeben, wieviele Vektoren innerhalb des Margins und wieviele auf der falschen

---

<sup>1</sup>Download unter: <http://www.gnu.org/software/octave/>.

Download von Zusatzpaketen unter: <http://octave.sourceforge.net/>.

Download der Version für Windows: [http://sourceforge.net/project/showfiles.php?group\\_id=2888](http://sourceforge.net/project/showfiles.php?group_id=2888)

<sup>2</sup><http://www.rwth-aachen.de/go/id/hi/>

<sup>3</sup>Download unter: <http://ltilib.sourceforge.net/doc/homepage/index.shtml>

Seite des Margins liegen. Darüber hinaus ist die SVM<sup>light</sup> nicht innerhalb einer Bibliothek vorhanden, es muss also kein Softwarepaket installiert werden, sondern die Implementierung besteht nur aus den zwei C-Programmen `svm_learn` und `svm_classify`.

Die Trainingsdaten werden der verwendeten SVM-Implementierung in Form einer Textdatei übergeben. Es wird folgendes Format erwartet:

```
<Klasse> <Merkmalsnr.>:<Wert> ... <Merkmalsnr.>:<Wert>
```

wobei:

<Klasse> den Wert +1 oder -1 annehmen kann und die Klasse des Merkmalsvektors repräsentiert.

<Merkmalsnr.> die Nummer des Merkmals angibt. Merkmale deren Wert Null ist, können weggelassen werden.

<Wert> den Wert des Merkmals angibt.

Jede Zeile in der Textdatei stellt dabei einen Merkmalsvektor dar.

Die Testdaten, bzw. die Daten zur Klassifikation, werden nach dem gleichen Schema wie die Trainingsdaten erwartet. Der einzige Unterschied besteht darin, dass keine Klasse angegeben werden muss, d. h. der Wert kann auf Null gesetzt werden.

Bei der Klassifikation der Daten wird das Ergebnis der Klassifikationen in eine Datei geschrieben, wobei hier jede Zeile den Wert der Entscheidungsfunktion für den zugehörigen Merkmalsvektor hält. Das Vorzeichen des Werts gibt dabei die ermittelte Klassenzugehörigkeit an.

Zusätzlich wurde das von Anton Schwaighofer entwickelte Matlab-Interface<sup>4</sup> zur SVM<sup>light</sup> in der Version 0.92 benutzt, das auch innerhalb von Octave eingesetzt werden kann.

## 5.2. Aufbau der Merkmalsextraktion

Abbildung 5.1 zeigt den Ablauf der Merkmalsberechnung.

Die Audiodateien werden mit Hilfe der Funktion `wavRead.m` eingelesen. Diese Funktion<sup>5</sup> wurde von Doug Stewart geschrieben und ermöglicht es auch 24-bit kodierte wav-Dateien einzulesen<sup>6</sup>.

Die eingelesenen Audiosignale werden normalisiert, so dass die maximale Lautstärke -3 dB ist. Die Normalisierung dient dazu, alle Audiosignale auf ein einheitliches Lautstärkeniveau

<sup>4</sup>Download unter: <http://ida.first.fraunhofer.de/~anton/software.html>

<sup>5</sup>Download unter: <http://dougs.homeip.net/octave/wavread.m>

<sup>6</sup>Die Datei `wavread.m` wurde in `wavRead.m` umbenannt, um nicht die in Octave vorhandene Funktion aufzurufen.

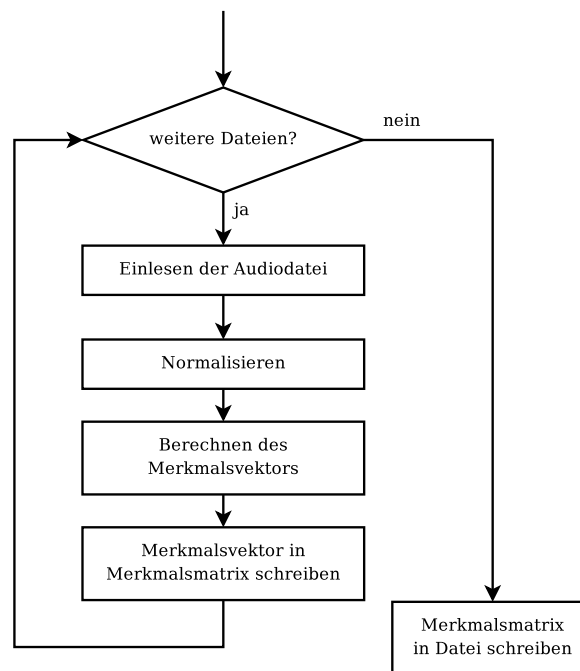


Abbildung 5.1.: Ablauf der Merkmalsberechnung.

zu heben. Somit wird verhindert, dass z. B. verschiedene Entfernungen der Musiker vom Mikrofon einen Einfluss auf die Merkmale haben. Nachdem die Audiodateien normalisiert wurden, wird die Grundfrequenz mittels der in Kapitel 3.1 vorgestellten Methode berechnet.

Das Cepstrum wird dazu mit Hilfe der in Octave vorhandenen Funktion `rceps.m` bestimmt. Innerhalb des so ermittelten Vektors kann, in dem für die Grundfrequenzbestimmung relevanten Indexbereich von  $LF = \frac{f_s}{600} = 73$  bis  $HF = \frac{f_s}{80} = 551$ , nach dem Maximum gesucht werden<sup>7</sup>. Mit  $f_s$  wird die Samplerate bezeichnet, diese gibt an, wie oft das Signal beim Digitalisieren pro Sekunde abgetastet wird. Mit dem durch die Maximumssuche ermittelten Index kann mit Hilfe der Formel

$$f_0 = \frac{f_s}{index}, \quad \text{mit } f_s: \text{ Samplerate in Hz} \quad (5.1)$$

die Grundfrequenz  $f_0$  bestimmt werden. In Abbildung 3.1(d) ist das markante Maximum bei Index 160 zu sehen, was bei einer Samplerate von 44100 Hz einer Grundfrequenz von 275,62 Hz entspricht.

Die Octave-Funktion zur Berechnung des zeitlichen Verlaufs der Harmonischen wurde von Julia Voigt [Voi07] übernommen. Das Programm wurde dabei so geändert, dass die Suche

<sup>7</sup>Der Bereich wird so eingeschränkt, da in den vorliegenden Klängen nur Grundfrequenzen von ca. 87 Hz bis ca. 590 Hz vorkommen.

nach den Maxima der Harmonischen Peaks innerhalb eines gewissen Toleranzbereichs hinzugefügt wurde. Das Ergebnis der Berechnung des zeitlichen Verlaufs der Harmonischen ist eine Matrix, diese hält in jeder Zeile einen Vektor. Dieser beschreibt den zeitlichen Verlauf einer Harmonischen. Der zeitliche Verlauf wird für die ersten 16 Harmonischen berechnet, damit ein möglichst detailliertes Abbild der Klangfarbe gebildet wird.

Zur Berechnung der MFCCs wurde die Matlab-Funktion `mfcc.m` benutzt. Diese ist Teil der von Apple entwickelten und von Malcolm Slaney [Sla98] weitergeführten Auditory Toolbox. Da Octave in den meisten Fällen kompatibel zu Matlab ist, können die Matlab-Funktionen meist ohne Probleme innerhalb von Octave-Programmen benutzt werden. Die Funktion `mfcc.m` wurde insoweit angepasst, dass die Größe der Fenster für die Analyse auf 1024 Samples geändert wurde. Dies entspricht bei einer Samplerate von 44,1 kHz ca. 23,2 ms. Des Weiteren wird die Funktion mit einer Framerate<sup>8</sup> von 25 Hz aufgerufen, was bedeutet, dass die einzelnen Segmente immer um 1764 Samples 'weitergeschoben' werden. Das heißt die MFCCs werden alle 40 ms berechnet. Das Ergebnis der MFCC-Berechnung ist eine Matrix. In dieser sind in jeder Spalte die zehn MFCCs, die zu den bestimmten Zeitpunkten (festgelegt durch die Framerate) berechnet werden, enthalten.

Alle so ermittelten Merkmale werden in Merkmalsvektoren geschrieben. Diese werden in einer Matrix, eine für jede Gitarre, zwischengespeichert, wobei jede Zeile einen Merkmalsvektor hält.

Damit eine einheitliche Größe der Merkmalsvektoren gewährleistet ist, werden die einzelnen Merkmale nach der Extraktion alle auf die gleiche Länge gebracht. Das bedeutet für den zeitlichen Verlauf der Harmonischen, dass die Spaltenanzahl der Matrix auf 40 festgesetzt wird. Ist die Spaltenzahl der ermittelten Matrix kleiner, wird mit Null aufgefüllt, ist sie größer, werden die Werte abgeschnitten. Dasselbe wird mit der Matrix gemacht, die für die MFCCs ermittelt wird. Hier wird die Spaltenanzahl auf 50 begrenzt.

Die Merkmale der drei Gitarren, insbesondere die Werte der zeitlichen Verläufe der Harmonischen, der MFCCs und der Amplituden der nontonalen Peaks, sind sich sehr ähnlich, d. h. sie unterscheiden sich nur hinter dem Komma. Um dies zu kompensieren, werden die genannten Merkmale mit 1000 multipliziert. Diese Skalierung hat zur Folge, dass die Hyperebene zwischen den einzelnen Klassen leichter berechnet werden kann. Dies ist zulässig, da die Skalierung auch auf die zu klassifizierenden Merkmale angewendet wird und somit alle Daten einheitlich verändert werden.

Da die One-Versus-One-Methode angewandt wird, wird für jede zu trainierende SVM jeweils eine Datei geschrieben. In diesen Dateien sind alle Merkmalsvektoren von jeweils zwei Gitarren für das Training enthalten. Den Gitarren werden dabei die beiden Klassen +1 und -1 zugeordnet.

---

<sup>8</sup>Gibt an wie oft die MFCC-Berechnung pro Sekunde durchgeführt wird.

Hierbei sei darauf verwiesen, dass der Gitarre Hense immer die Klasse +1 und der Gitarre Wichmann immer die Klasse -1 zugeordnet wird. Die Gitarre Marin bekommt einmal die Klasse +1 (in der Datei MarinVsWichmann.data) und einmal die Klasse -1 (in der Datei HenseVsMarin.data) zugeordnet. Darauf ist in der Klassifizierungsphase, beim Aufbau des azyklischen Graphen, zu achten.

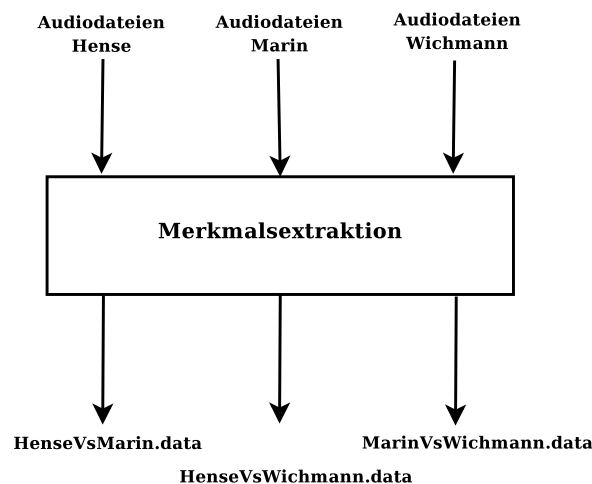


Abbildung 5.2.: Merkmalsextraktion

## 5.3. Aufbau der Trainings- und Testphase

Die vorliegenden Audiodateien wurden so aufgeteilt, dass jeweils 50 % der Klänge zum Training bzw. zum Testen verwendet wurden<sup>9</sup>. Somit liegen für die Trainingsphase insgesamt 978 und für die Testphase insgesamt 972 Klänge vor.

### 5.3.1. Trainingsphase

Da die benutzte SVM-Implementierung nur Zwei-Klassen-Probleme lösen kann, werden insgesamt drei Support Vector Machines trainiert. Der Prozess der Trainingsphase ist in Abbildung 5.3 aufgezeigt.

<sup>9</sup>Für die Gitarren Hense und Marin liegen jeweils drei Klänge mehr vor, da einige der Klänge fünfmal gespielt wurden. Diese zusätzlichen Klänge wurden in den Trainingsdatensatz aufgenommen.

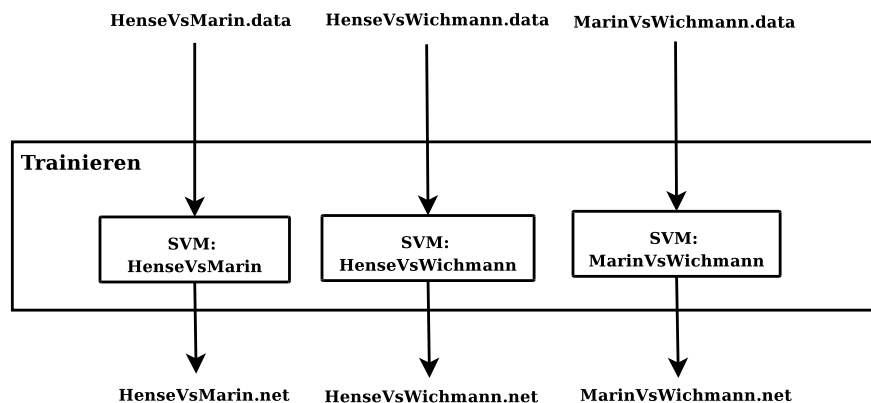


Abbildung 5.3.: Trainingsprozess

Die während der Merkmalsextraktionsphase geschriebenen Dateien werden der SVM-Implementierung übergeben. Um eine einheitliche Programmiersprache zu verwenden, wurde das Matlab-Interface zur SVM<sup>light</sup> verwendet. Mit Hilfe der Funktion `svmlopt.m` werden die Parameter der SVM festgelegt,

```
optionsLinear = svmlopt('Verbosity',1,'C',1.0,'Biased',1,'
    Kernel',0);
```

'Verbosity', 1: legt fest, wie detailliert die Trainingsausgabe sein soll.

'C', 1.0: legt den Wert für den Parameter  $C$  fest<sup>10</sup>.

'Biased', 1: legt fest, dass die Hyperebene die Form  $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$  haben soll.

'Kernel', 0: legt fest welche Kernel-Funktion benutzt werden soll.

0: Linear-Kernel, 1: Polynomial-Kernel, 2: RBF-Kernel, 3: Sigmoid-Kernel, 4: Benutzerdefinierte Kernel-Funktion

Mit Hilfe der Funktion `svm_learn.m` wird die SVM trainiert. Die Funktion wurde insofern verändert, dass ein zusätzlicher Parameter `svmOutputPath` hinzugefügt wurde.

```
svm_learn(optionsLinear, filepath, modelpath,
    svmOutputPath);
```

`optionsLinear`: Optionen der SVM.

`filepath`: Pfad zu der Datei, welche die Trainingsdaten hält.

`modelpath`: Pfad zu der Datei, in welche die Daten der trainierten SVM geschrieben werden sollen.

<sup>10</sup>siehe Seite 29.

`svmOutputPath`: Pfad zu der Datei, welche die Trainingsausgabe hält.

Der Parameter `svmOutputPath` wurde hinzugefügt, um die Ausgabe der Trainingsphase in eine Datei abspeichern zu können. Dadurch hat man die Möglichkeit, nach dem Trainieren die Qualität der trainierten SVM zu überprüfen. Diese Datei enthält unter anderem Informationen über die Dauer der Trainingsphase, zum einen in Sekunden, zum anderen in der Anzahl der benötigten Iterationen. Des Weiteren wird angegeben, wieviele Supportvektoren berechnet wurden.

Die Daten der drei trainierten SVMs werden jeweils in eine Datei, z. B. `HenseVsMarin.net`, geschrieben. In dieser steht, mit welchem Kernel trainiert wurde, sowie die Parameter des trainierten Kernels. Zusätzlich dazu werden alle Supportvektoren aufgelistet und der Wert der Variable  $b$  angegeben.

Der Vorteil, die Daten der SVMs in eine Datei zu schreiben, liegt darin, dass nicht für jede Klassifikation neu trainiert werden muss.

### 5.3.2. Testphase

Um Testen zu können, wie exakt die einzelnen trainierten SVMs klassifizieren, wurde die Testphase so konzipiert, dass mit allen zur Verfügung stehenden Testdateien getestet wurde. Dazu wurde ein gerichteter azyklischer Graph (siehe Kapitel 4.3.1) aufgebaut. Abbildung 5.4 zeigt den Aufbau des DAG.

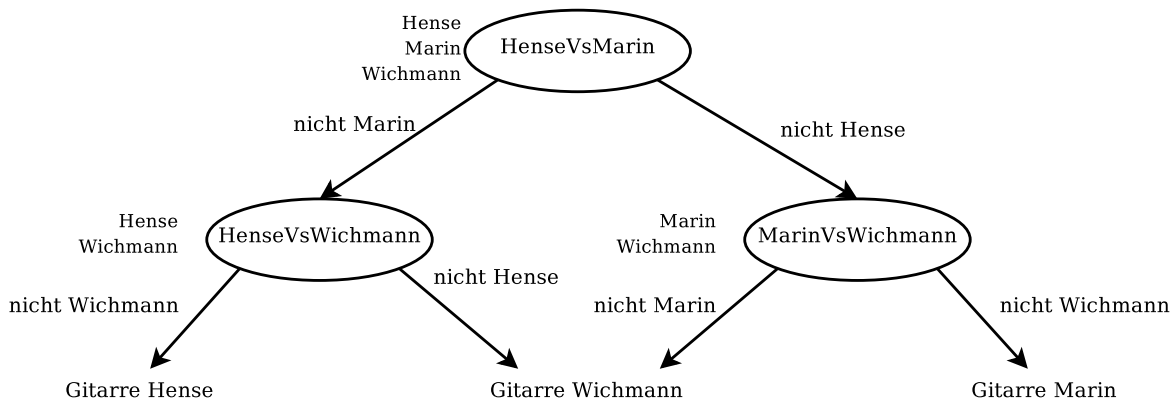


Abbildung 5.4.: Aufbau des DAG für die Testphase.

Die Ergebnisse jedes Testlaufs wurden gespeichert und anschließend ausgewertet, so dass nach Abschluss der Testphase die einzelnen Erkennungsraten ausgegeben werden können. Zusätzlich zu den Erkennungsraten wird die Anzahl, wie oft eine Gitarre als eine bestimmte

Gitarre klassifiziert wurde, angegeben. Dies bietet den Vorteil, die Testphase besser auswerten zu können.

Die Optionen werden wieder mit

```
options = svmlopt('Verbosity', 1);
```

gesetzt. Wobei die Optionen hier den Detailgrad der Ausgabe festlegen. Die Klassifizierung wird dabei mit Hilfe der Funktion `svm_classify.m` vorgenommen.

```
status = svm_classify(options, filename, model, output);
```

`options`: Optionen.

`filename`: Datei mit den zu klassifizierenden Merkmalsvektoren.

`model`: Name der Datei der trainierten SVM.

`output`: Name der Datei, in der das Klassifizierungsergebnis abgespeichert werden soll.

`status`: Gibt den Status der Klassifikation an. 0 bedeutet die Klassifizierung war erfolgreich.



## 6. Ergebnisse

In diesem Teil der Arbeit sind die Ergebnisse der einzelnen Experimente aufgelistet. Es wurden verschiedene Experimente mit unterschiedlichen Merkmalsvektoren durchgeführt, um die, für die Klassifikation der drei Gitarren am besten geeigneten Merkmale, zu ermitteln. Dabei wurden die Support Vector Machines mit insgesamt 978 Datensätzen trainiert und mit 972 Datensätzen getestet.

Die Experimente wurden dabei so durchgeführt, dass zuerst mit jeweils einem, der in Kapitel 3 vorgestellten, Merkmale getestet wurde. In den folgenden Experimenten wurde jeweils ein weiteres Merkmal aufgenommen und überprüft, ob und wie sich die Erkennungsraten verbessert haben. Im letzten Experiment wurden alle zur Verfügung stehenden Merkmale in den Merkmalsvektor aufgenommen.

Die Merkmalsvektoren setzen sich dabei so zusammen, dass die Grundfrequenz immer als Merkmal aufgenommen wird. Die Anzahl der Werte für die zeitlichen Verläufe der Harmonischen ergibt sich daraus, dass die zeitlichen Verläufe der ersten 16 Harmonischen über vier Sekunden berechnet werden. Da die Amplitudenwerte der Harmonischen alle 100 ms bestimmt werden, ergibt dies einen Vektor mit 640 Werten.

Die Größe des Vektors, der die Werte der MFCCs hält, ergibt sich daraus, dass die ersten zehn MFCCs über einen Zeitraum von zwei Sekunden bestimmt werden. Da die MFCCs mit einer Framerate von 25 Hz, also alle 40 ms, berechnet werden, ergibt sich ein Vektor mit 500 Werten.

Die Diagramme zeigen dabei die einzelnen Erkennungsraten der drei Gitarren. Um einen detaillierteren Überblick über die Klassifikation zu bekommen, werden für die einzelnen Experimente die Verwechslungsmatrizen aufgezeigt. Diese geben an in welcher Anzahl eine Gitarre als eine bestimmte Gitarre erkannt wurde. Des Weiteren wird dokumentiert, wie die Gitarren durch die einzelnen Knoten des DAG (siehe Kapitel 5.4) klassifiziert werden. Dadurch hat man die Möglichkeit, die Klassifikationsergebnisse besser zu analysieren und evtl., bei schlechter Erkennungsrate, die Parameter des Kernels oder den Kernel der betreffenden Support Vector Machine zu ändern. Bei den Tabellen (siehe z. B. Tabelle 6.3) ist zu beachten, dass die Ergebnisse verneinend dargestellt werden.

Alle Support Vector Machines wurden dabei mit einem Linear-Kernel mit dem Parameter  $C = 1$  trainiert.

## 6.1. Experiment 1

In diesem Experiment wird der zeitliche Verlauf der Harmonischen als Merkmal benutzt. Tabelle 6.1 zeigt den Aufbau des Merkmalsvektors und dessen Größe. Abbildung 6.1 gibt die Erkennungsraten, die mit allen zur Verfügung stehenden Testdaten erreicht wurden, an. Die Tabellen 6.2 und 6.3 geben an, wieviele der Datensätze richtig bzw. falsch erkannt wurden. Tabelle 6.3 führt dies in detaillierter Form für jeden Knoten des DAG auf.

Merkmale	Anzahl
Grundfrequenz	1
Zeitlicher Verlauf der Harmonischen	640
<b>Gesamt</b>	<b>641</b>

Tabelle 6.1.: Aufbau des Merkmalsvektors für Experiment 1.

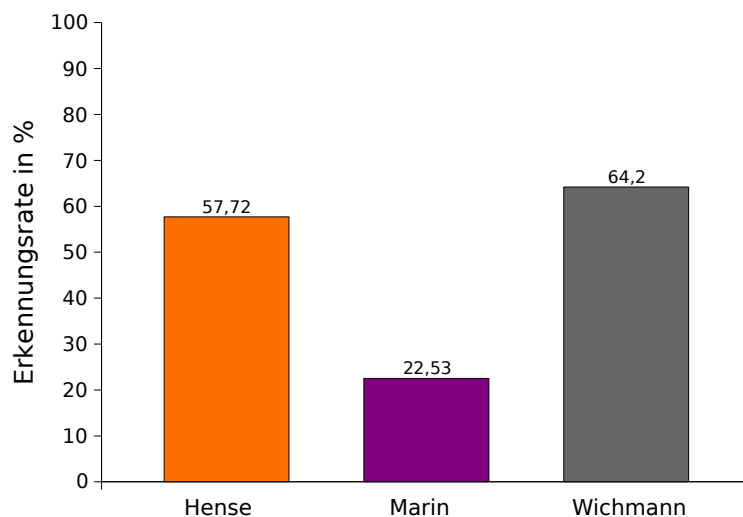


Abbildung 6.1.: Erkennungsraten der drei Gitarren für Experiment 1.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	<b>187</b>	34	103
Marin	127	<b>73</b>	124
Wichmann	92	24	<b>208</b>

Tabelle 6.2.: Verwechslungsmatrix Experiment 1.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
<b>Hense</b>	66	258
<b>Marin</b>	131	193
<b>Wichmann</b>	116	208

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
<b>Hense</b>	71	<b>187</b>
<b>Marin</b>	66	127
<b>Wichmann</b>	<b>116</b>	92

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
<b>Hense</b>	34	32
<b>Marin</b>	58	<b>73</b>
<b>Wichmann</b>	<b>92</b>	24

Tabelle 6.3.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 1. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Wie die Abbildung 6.1 zeigt, liefert der zeitliche Verlauf der Harmonischen als Merkmal gute Ergebnisse für die Gitarren Hense und Wichmann. Die Gitarre Marin wird allerdings fast zu gleichen Teilen als Hense bzw. als Wichmann erkannt. Das spricht dafür, dass die zeitlichen Verläufe der Klänge für die Gitarre Marin, sowohl denen der Gitarre Hense, als auch denen der Gitarre Wichmann sehr ähnlich sind.

Eine detaillierte Analyse der Klassifizierung der einzelnen Knoten des DAG ergab, dass schon im ersten Knoten (siehe Tabelle 6.2(a)) 193 Datensätze als 'nicht Marin', also falsch, klassifiziert wurden. Im zweiten Knoten (siehe Tabelle 6.2(b)) wurden 127 Klänge der Gitarre Marin als Gitarre Hense eingeordnet. Aber auch im Knoten 3 wird die Gitarre Marin schlecht erkannt, so werden nur 73 von 131 Datensätze richtig klassifiziert.

Die Ergebnisse der Gitarren Hense und Wichmann zeigen zufriedenstellendere Werte auf, wobei das Instrument Wichmann mit 64,2% eine bessere Erkennungsrate aufweist. Die Gitarre Hense wird dabei im ersten Knoten ('HenseVsMarin') mit 258 Datensätzen als 'nicht Marin', also richtig, erkannt. Beim zweiten Knoten ('HenseVsWichmann') werden allerdings nur 187 Dateien als Hense erkannt.

## 6.2. Experiment 2

In diesem Versuch werden die Amplitudenwerte der ersten 15 Peaks des nontonalen Spektrums und die Frequenzen bei denen die Peaks liegen als Merkmale verwendet. Tabelle 6.4 zeigt den Inhalt und die Größe des Merkmalsvektors auf.

Merkmale	Anzahl
Grundfrequenz	1
Amplituden und Frequenzen der ersten 15 Peaks des nontonalen Spektrums	30
<b>Gesamt</b>	<b>31</b>

Tabelle 6.4.: Aufbau des Merkmalsvektors für Experiment 2.

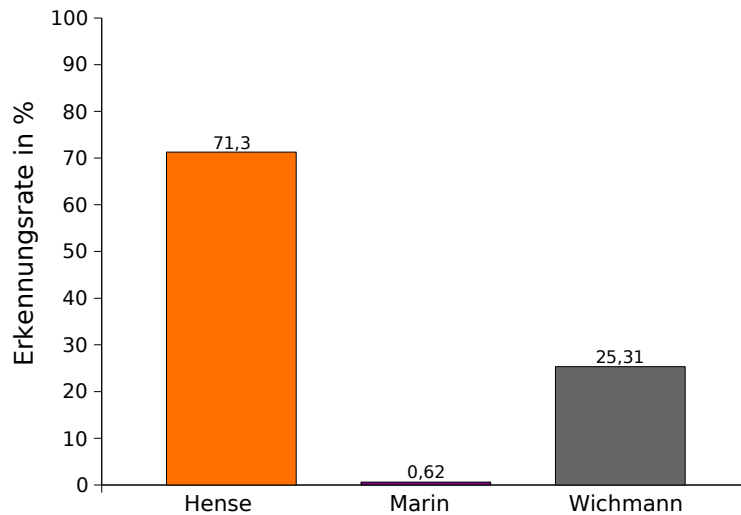


Abbildung 6.2.: Erkennungsraten der drei Gitarren für Experiment 2.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	<b>231</b>	4	89
Marin	243	<b>2</b>	79
Wichmann	242	0	<b>82</b>

Tabelle 6.5.: Verwechslungsmatrix Experiment 2.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
<b>Hense</b>	75	249
<b>Marin</b>	69	255
<b>Wichmann</b>	74	250

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
<b>Hense</b>	18	<b>231</b>
<b>Marin</b>	12	243
<b>Wichmann</b>	<b>8</b>	242

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
<b>Hense</b>	71	4
<b>Marin</b>	67	<b>2</b>
<b>Wichmann</b>	<b>74</b>	0

Tabelle 6.6.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 2. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Dieses Experiment weist besonders für die Gitarre Marin sehr schlechte Erkennungsraten auf. Auch die Gitarre Wichmann wird schlechter erkannt als im vorhergehenden Experiment. Dabei ist auffällig, dass beide Gitarren, Marin und Wichmann, vornehmlich als Hense erkannt wurden. Eine Analyse der Klassifizierungsergebnisse der einzelnen Knoten (siehe Tabelle 6.6) ergab, dass die Datensätze der zwei Gitarren schon im ersten Knoten ('HenseVsMarin') stark zur Gitarre Hense tendierten. Dies ist besonders für die richtige Klassifikation des Instruments Marin von Nachteil. Mit nur zwei richtigen Klassifizierungen hat diese Gitarre eine sehr schlechte Erkennungsrate.

Dies spricht dafür, dass die Peaks des nontonalen Spektrums der Gitarre Marin stark der Gitarre Hense gleichen und es somit schon während der Trainingsphase zu viele Fehlklassifikationen<sup>1</sup> gab.

Auch die Gitarre Wichmann wird im zweiten Knoten ('HenseVsWichmann') mit nur acht korrekt klassifizierten Klängen sehr schlecht erkannt. Hier kann man versuchen durch Änderung der Parameter und des Kernels die Ergebnisse zu verbessern. Dagegen wurden die 74 Klänge, die im ersten Knoten als 'nicht Hense' erkannt wurden, alle im dritten Knoten ('MarinVsWichmann') als Wichmann eingestuft.

<sup>1</sup>Als Fehlklassifikationen während der Trainingsphasen werden hier die Trainingsdatensätze bezeichnet, die einen größeren Fehler aufweisen als den zugelassenen und somit auf der falschen Seite des Margins liegen.

### 6.3. Experiment 3

In diesem Versuch werden die ersten 10 MFCCs als Merkmale benutzt. Abbildung 6.3 zeigt die Erkennungsraten der drei Gitarren auf. Die Tabellen 6.8 und 6.9 geben die Anzahl der richtig bzw. falsch klassifizierten Datensätze an.

Merkmale	Anzahl
Grundfrequenz	1
Die ersten 10 MFCCs bei einer Framerate von 25 Hz	500
<b>Gesamt</b>	<b>501</b>

Tabelle 6.7.: Aufbau des Merkmalsvektors für Experiment 3.

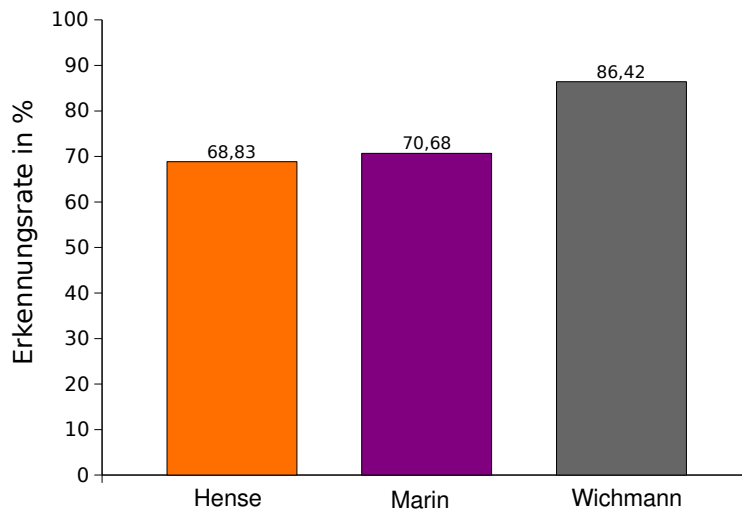


Abbildung 6.3.: Erkennungsraten der drei Gitarren für Experiment 3.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	223	55	46
Marin	52	229	43
Wichmann	28	16	280

Tabelle 6.8.: Verwechslungsmatrix Experiment 3.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
<b>Hense</b>	72	252
<b>Marin</b>	256	68
<b>Wichmann</b>	122	202

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
<b>Hense</b>	29	<b>223</b>
<b>Marin</b>	16	52
<b>Wichmann</b>	<b>174</b>	28

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
<b>Hense</b>	17	55
<b>Marin</b>	27	<b>229</b>
<b>Wichmann</b>	<b>106</b>	16

Tabelle 6.9.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 3. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Die Erkennungsraten der drei Gitarren bei diesem Experiment weisen zufriedenstellende Ergebnisse auf, wobei die Gitarre Wichmann das beste Ergebnis aufzeigt. Die beiden anderen Gitarren werden jedoch mit über 60 % richtiger Klassifikation deutlich besser erkannt, als in den vorhergehenden Experimenten.

Da sowohl die Merkmale aus Experiment 1, als auch diejenigen, die in diesem Experiment verwendet wurden, die Klangfarbe eines Instruments beschreiben, waren die besseren Klassifikationsergebnisse nicht vorhersehbar. Ein möglicher Grund für dieses Ergebnis ist, dass die MFCCs die Klangfarbe eines Instruments in einer detaillierteren Weise darstellen, als der zeitliche Verlauf der Harmonischen.

Besonders die Erkennungsrate der Gitarre Marin ist deutlich besser im Vergleich zu Experiment 1. Es fällt auf, dass gerade die Fehlklassifikationen im ersten Knoten aus Experiment 1 für Gitarre Marin merklich abnehmen. Dies spricht dafür, dass mit Hilfe der MFCCs die klanglichen Eigenschaften der Gitarren Hense und Marin besser beschrieben werden, als mit den zeitlichen Verläufen der Harmonischen.

## 6.4. Experiment 4

Dieses Experiment kombiniert die Peaks des nontonalen Spektrums mit dem zeitlichen Verlauf der Harmonischen. Der Aufbau des Merkmalsvektors ist in Tabelle 6.10 aufgezeigt. Die Erkennungsraten der drei Gitarren werden in Abbildung 6.4 dargestellt.

Merkmale	Anzahl
Grundfrequenz	1
Amplituden und Frequenzen der ersten 15 Peaks des nontonalen Spektrums	30
Zeitlicher Verlauf der Harmonischen	640
<b>Gesamt</b>	<b>671</b>

Tabelle 6.10.: Aufbau des Merkmalsvektors für Experiment 4.

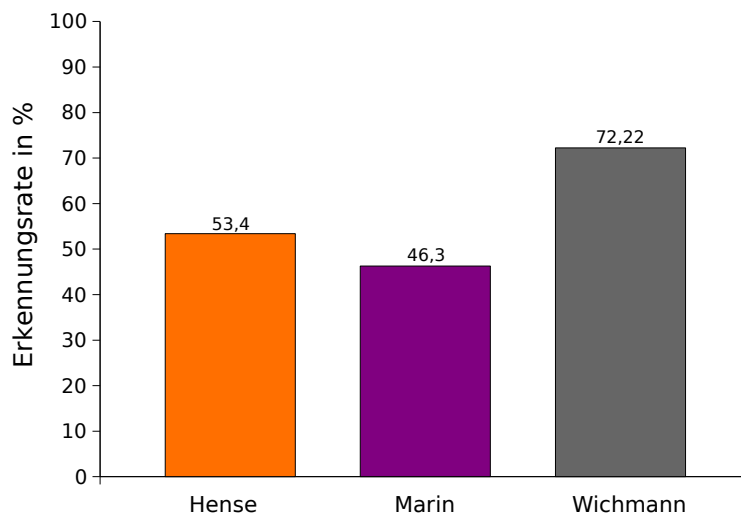


Abbildung 6.4.: Erkennungsraten der drei Gitarren für Experiment 4.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	173	83	68
Marin	105	150	69
Wichmann	49	41	234

Tabelle 6.11.: Verwechslungsmatrix Experiment 4.



(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
<b>Hense</b>	114	210
<b>Marin</b>	179	145
<b>Wichmann</b>	152	172

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
<b>Hense</b>	37	<b>173</b>
<b>Marin</b>	40	105
<b>Wichmann</b>	<b>123</b>	49

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
<b>Hense</b>	31	83
<b>Marin</b>	29	<b>150</b>
<b>Wichmann</b>	<b>111</b>	41

Tabelle 6.12.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 4. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Obwohl die Gitarre Marin sowohl in Experiment 1, als auch in Experiment 2 sehr schlechte Erkennungsraten aufweist, zeigt sich, dass die Zusammenführung beider Merkmale, dem zeitlichen Verlauf der Harmonischen und den Peaks des nontonalen Spektrums, eine deutlich bessere Klassifizierung mit sich bringt.

So ist auffällig, dass gerade im ersten Knoten (siehe Tabelle 6.11(a)) die Zahl der Fehlklassifikationen sinkt. Aber auch der dritte Knoten (Tabelle 6.11(c)) weist für die Gitarre Marin eine geringere Fehlklassifikation auf. Dies zeigt, dass die Gitarre Marin durch die Kombination der beiden Merkmale deutlich besser beschrieben werden kann, als mit der einzelnen Verwendung der Merkmale.

Ebenso wird das Instrument Wichmann besser erkannt als in den Experimenten 1 und 2, in denen nur ein Merkmal verwendet wurde. So ist in Tabelle 6.11 zu sehen, dass die Fehlklassifikation in Richtung Hense aus Experiment 2 merklich abnimmt.

## 6.5. Experiment 5

Dieses Experiment verbindet die MFCCs mit den zeitlichen Verläufen der Harmonischen. Tabelle 6.13 zeigt den Aufbau des Merkmalsvektors und gibt dessen Größe an. Wie die Tabelle zeigt, ist der Merkmalsvektor, mit 1141 Merkmalen, sehr groß. Abbildung 6.5 stellt die Erkennungsraten der drei Gitarren dar.

Merkmale	Anzahl
Grundfrequenz	1
Die ersten 10 MFCCs bei einer Framerate von 25 Hz	500
Zeitlicher Verlauf der Harmonischen	640
<b>Gesamt</b>	<b>1141</b>

Tabelle 6.13.: Aufbau des Merkmalsvektors für Experiment 5.

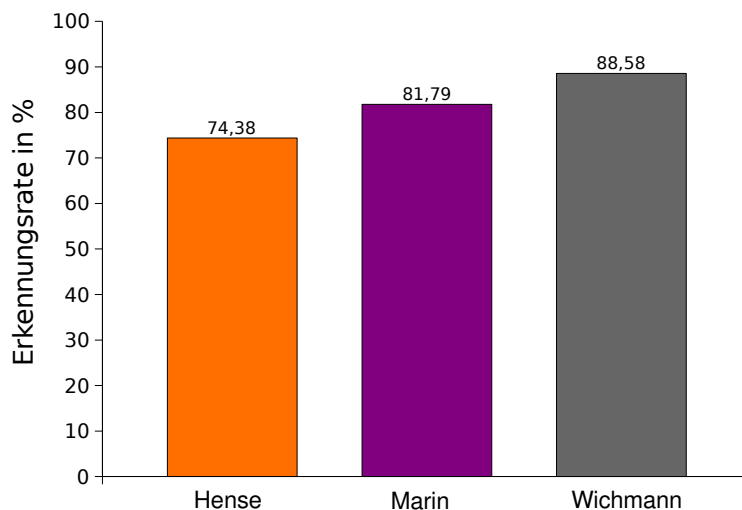


Abbildung 6.5.: Erkennungsraten der drei Gitarren für Experiment 5.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	<b>241</b>	43	40
Marin	32	<b>265</b>	27
Wichmann	17	20	<b>287</b>

Tabelle 6.14.: Verwechslungsmatrix Experiment 5.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
Hense	60	264
Marin	282	42
Wichmann	100	224

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
Hense	23	<b>241</b>
Marin	10	32
Wichmann	<b>207</b>	17

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
Hense	17	43
Marin	17	<b>265</b>
Wichmann	<b>80</b>	20

Tabelle 6.15.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 5. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Die Verwendung der zeitlichen Verläufe der Harmonischen zusammen mit den MFCCs bringt eine bezeichnende Verbesserung der Erkennungsraten, gegenüber denen des Experiments 3. So wird die Erkennungsrate der Gitarre Marin um zehn Prozent gesteigert, aber auch bei den beiden anderen Gitarren ist ein Anstieg zu erkennen.

Die Klassifikationsergebnisse der einzelnen Knoten, besonders die des ersten Knotens (Tabelle 6.14(a)) sind für die Gitarre Marin besser. Aber auch für die SVM 'MarinVsWichmann' (Tabelle 6.14(c)) sind die Erkennungsraten gestiegen.

Dieses gute Klassifikationsergebnis zeigt, dass die Mel Frequency Cepstral Coefficients in Verbindung mit dem zeitlichen Verlauf der Harmonischen die klanglichen Eigenschaften der Instrumente sehr gut beschreiben.

## 6.6. Experiment 6

In diesem Test werden die MFCCs mit den Peaks des nontonalen Spektrums als Merkmale verwendet. Tabelle 6.16 zeigt den Aufbau und die Größe des benutzten Merkmalsvektors.

Merkmale	Anzahl
Grundfrequenz	1
Amplituden und Frequenzen der ersten 15 Peaks des nontonalen Spektrums	30
Die ersten 10 MFCCs bei einer Framerate von 25 Hz	500
<b>Gesamt</b>	<b>531</b>

Tabelle 6.16.: Aufbau des Merkmalsvektors für Experiment 6.

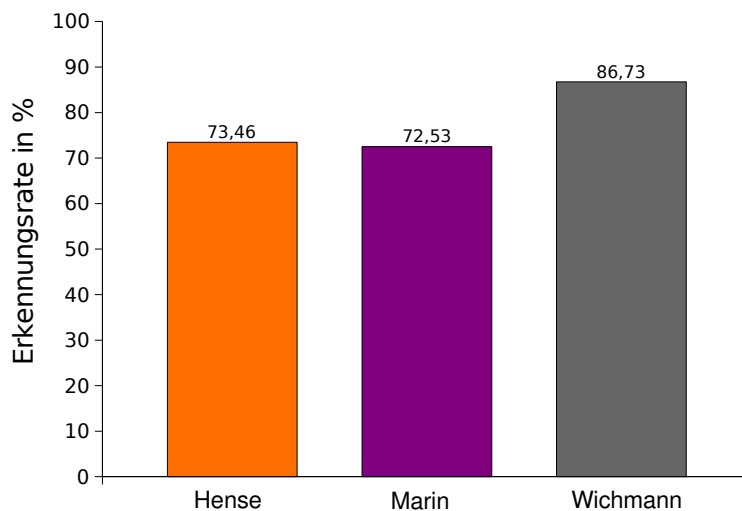


Abbildung 6.6.: Erkennungsraten der drei Gitarren für Experiment 6.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	238	50	36
Marin	46	235	43
Wichmann	24	19	281

Tabelle 6.17.: Verwechslungsmatrix Experiment 6.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
Hense	59	265
Marin	264	60
Wichmann	118	206

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
Hense	27	<b>238</b>
Marin	14	46
Wichmann	<b>182</b>	24

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
Hense	9	50
Marin	29	<b>235</b>
Wichmann	<b>99</b>	19

Tabelle 6.18.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 6. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Werden die nontonalen Peaks und die MFCCs zusammen als Merkmale eingesetzt, zeigt sich gerade gegenüber Experiment 2 eine deutliche Verbesserung der Klassifikationsergebnisse für die Gitarren Marin und Wichmann. Aber auch gegenüber Experiment 3 steigen die Erkennungsraten.

Was auffällt ist, dass sich, obwohl Gitarre Marin in Experiment 2 sehr schlechte Erkennungsraten aufweist, die Klassifikationsergebnisse bei Verwendung beider Merkmale leicht verbessern. Dies läßt darauf schließen, dass die klanglichen Eigenschaften der Gitarre Marin durch Kombination beider Merkmale besser beschrieben werden.

Das sich die Anzahl der richtig erkannten Klänge der Gitarre Hense gegenüber Experiment 3 steigert, hängt offensichtlich mit der Verwendung der nontonalen Peaks und deren Frequenzen ab, da in Experiment 2 Gitarre Hense die höchste Erkennungsrate aufweist.

Durch die MFCCs nimmt für die Gitarren Marin und Wichmann die Verschiebung in Richtung Gitarre Hense aus Experiment 2 merklich ab, was darauf schließen läßt, dass die MFCCs gerade die Gitarren Marin und Wichmann besser beschreiben.

## 6.7. Experiment 7

In diesem Versuch werden alle zur Verfügung stehenden Merkmale verwendet. Tabelle 6.19 zeigt den Aufbau des Merkmalsvektors und die Anzahl der verwendeten Merkmale auf.

Merkmale	Anzahl
Grundfrequenz	1
Amplituden und Frequenzen der ersten 15 Peaks des nontonalen Spektrums	30
Die ersten 10 MFCCs bei einer Framerate von 25 Hz	500
Zeitlicher Verlauf der Harmonischen	640
<b>Gesamt</b>	<b>1171</b>

Tabelle 6.19.: Aufbau des Merkmalsvektors für Experiment 7.

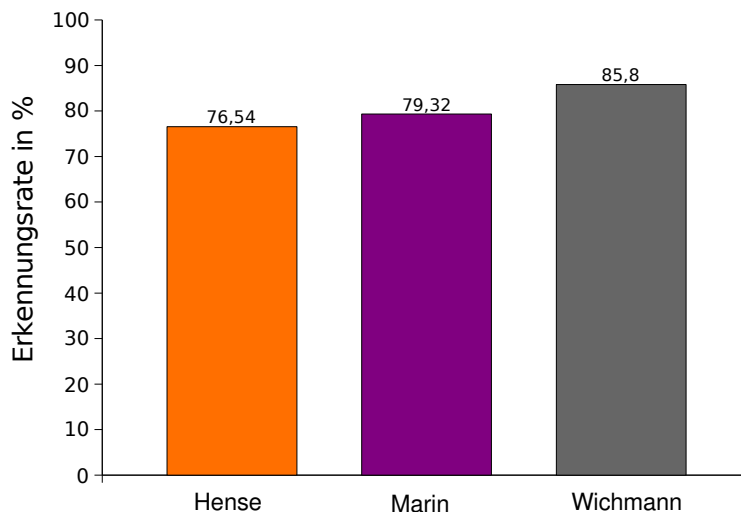


Abbildung 6.7.: Erkennungsraten der drei Gitarren für Experiment 7.

	klassifiziert als Gitarre		
	Hense	Marin	Wichmann
Hense	<b>248</b>	38	38
Marin	38	<b>257</b>	29
Wichmann	21	25	<b>278</b>

Tabelle 6.20.: Verwechslungsmatrix Experiment 7.

(a) Knoten 1: HenseVsMarin		
	als nicht Hense	als nicht Marin
Hense	54	270
Marin	277	47
Wichmann	116	208

(b) Knoten 2: HenseVsWichmann		
	als nicht Hense	als nicht Wichmann
Hense	22	<b>248</b>
Marin	9	38
Wichmann	<b>187</b>	21

(c) Knoten 3: MarinVsWichmann		
	als nicht Marin	als nicht Wichmann
Hense	16	38
Marin	20	<b>257</b>
Wichmann	<b>91</b>	25

Tabelle 6.21.: Klassifikationsergebnisse der einzelnen Knoten des DAG für Experiment 7. Die Anzahl der richtig klassifizierten Ergebnisse ist dabei hervorgehoben.

Die Gitarre Hense weist die beste Erkennungsrate aller Experimente, bei der Verwendung aller zur Verfügung stehenden Merkmale, auf. Die Erkennungsraten der Gitarren Marin und Wichmann sinken jedoch, gegenüber den besten Klassifizierungsergebnissen dieser beiden Gitarren. So erreichen diese die besten Ergebnisse in Experiment 5 mit 81,79 % bzw. 88,58 %.

Die geringere Fehlerrate der Gitarre Hense hängt offensichtlich mit der Verwendung der Peaks des nontonalen Spektrums zusammen, da auch in Experiment 6 die Verwendung dieser Peaks ein besseres Klassifikationsergebnis für Gitarre Hense, gegenüber Experiment 3 bewirkt.

Das die Erkennungsraten der beiden anderen Gitarren gegenüber Experiment 5 sinken, hängt offensichtlich auch mit der Verwendung der Peaks des nontonalen Spektrums zusammen.

## 6.8. Diskussion der Ergebnisse

Abbildung 6.8 zeigt die Gesamterkennungsraten der einzelnen Experimente auf.

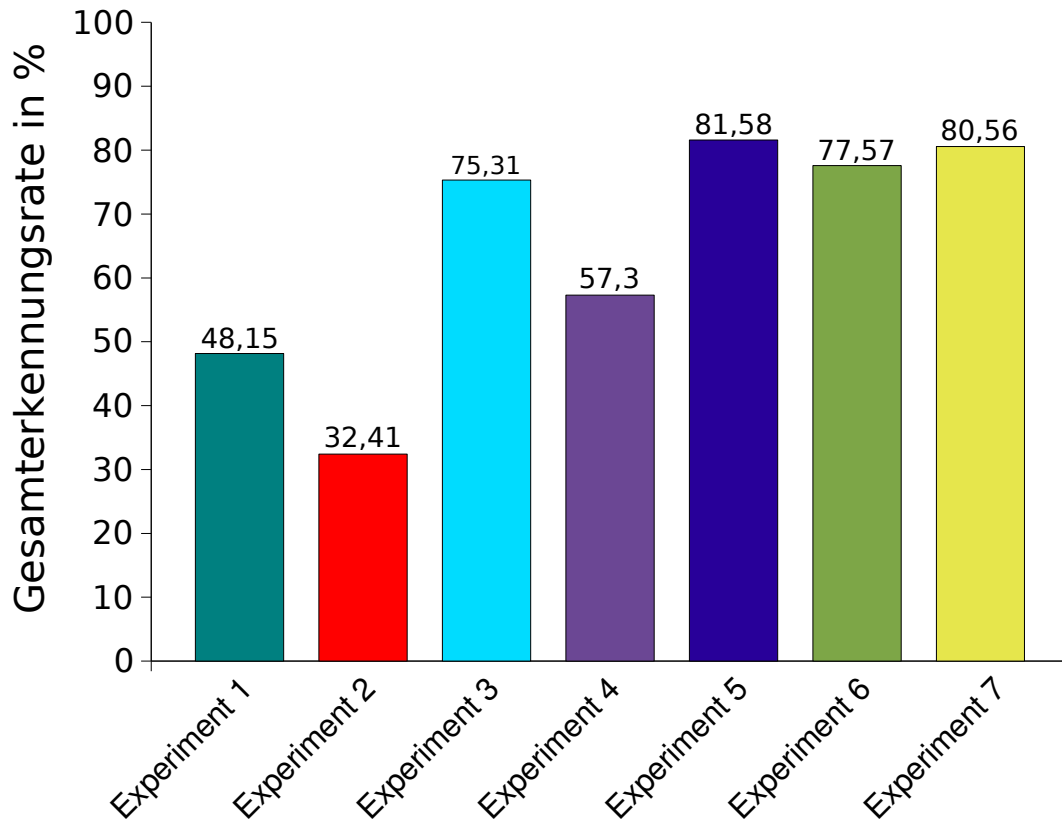


Abbildung 6.8.: Gesamterkennungsraten der einzelnen Experimente.

Wie sich zeigt, wird das beste Ergebnis in Experiment 5 erreicht. Hier werden die zeitlichen Verläufe der Harmonischen und die MFCCs als Merkmale eingesetzt. Die Gesamterkennungsraten des Experiments 7, in dem alle zur Verfügung stehenden Merkmale verwendet werden, sinkt gegenüber Experiment 5 um 1,02 %. Dieses schlechtere Ergebnis hängt damit zusammen, dass die Anzahl der richtig klassifizierten Klänge der Gitarren Marin und Wichmann, aus Experiment 7 gegenüber Experiment 5, sinkt. Die Erkennungsrate der Gitarre Hense steigt jedoch.

Dies zeigt, dass die Peaks des nontonalen Spektrums und deren Frequenzen auf die Gesamterkennungsraten einen sehr geringen Einfluss haben. Es zeigt sich jedoch auch, dass mit Hilfe dieser Merkmale, die Gitarre Hense die höchste Erkennungsraten aufweist, daher ist es



sinnvoll diese als Merkmale beizubehalten. Es sollten jedoch nicht die Amplituden und Frequenzen der Peaks als Merkmal benutzt werden, da ein Teil der Peaks durch das Wegschneiden der Harmonischen entsteht (siehe Abbildung 3.5). Ein mögliches Merkmal wäre z. B. das Verhältnis der Fläche des nontonalen Spektrums zur Fläche des gesamten Spektrums. Dies würde aufzeigen, welchen Anteil das nontonale Spektrum am gesamten Spektrum hat. Man kann, um dieses Verhältnis feiner darzustellen, das Spektrum in einzelne Frequenzabschnitte aufteilen und für diese Abschnitte das Verhältnis berechnen. Somit könnte das nontonale Spektrum auch die Gitarren Marin und Wichmann besser beschreiben.

Des Weiteren zeigen die Erkennungsraten, dass die Verwendung der MFCCs zusammen mit dem zeitlichen Verlauf der Harmonischen, gegenüber den einzelnen Verwendungen dieser Merkmale einen wesentlich geringere Fehlerrate produziert. Dies zeigt, dass die charakteristischen Klangeigenschaften der einzelnen Gitarren, sehr gut mit Hilfe dieser beiden Merkmale beschrieben werden können.

Auch wird deutlich, dass die Experimente, in denen die MFCCs als Merkmale Verwendung finden, die besten Klassifikationsergebnisse liefern. Somit ist ersichtlich, dass die charakteristischen Eigenschaften der Gitarren besonders gut mit Hilfe der MFCCs beschrieben werden können.

Zu untersuchen wäre, wie sich eine Erhöhung der MFCC-Anzahl auf die Erkennungsraten der Gitarren auswirkt. Durch eine größere Anzahl an MFCCs könnten die spektralen Formen der einzelnen Klänge in einer noch detaillierteren Weise beschrieben werden. Denkbar wäre auch, die Framerate bei der MFCC-Berechnung von 25 Hz auf 50 Hz zu erhöhen. Dadurch würden die MFCCs nicht nur alle 40 ms berechnet werden, sondern alle 20 ms. Dies hätte allerdings zur Folge, dass der Merkmalsvektor stark vergrößert wird. Eine Verdoppelung der Framerate verursacht eine Verdoppelung der MFCC-Merkmale.

Man kann zusätzlich dazu die Auflösung des zeitlichen Verlaufs der Harmonischen, von 100 ms auf z. B. 50 ms, feiner gestalten. Dies ließe sich dadurch erreichen, dass die Fensterbreite beibehalten wird, das Fenster jedoch statt um eine gesamte Fensterbreite, nur um die halbe Fensterbreite verschoben wird. Jedoch würde sich die Größe des Merkmalsvektors auch hier erhöhen.

In den vorliegenden Experimenten wurde, für alle Support Vector Machines, der Linear-Kernel mit dem Wert 1 für den Parameter  $C$  verwendet. Tests mit einem höheren Wert für  $C$  ergaben keine Verbesserungen der Ergebnisse.

Versuche mit anderen Kernen lieferten keine zufriedenstellenden Ergebnisse. Es sei jedoch darauf hingewiesen, dass für alle Knoten immer der Linear-Kernel verwendet wurde. Daher ist zu testen, ob die Verwendung einer anderen Kernel-Funktion, für die Knoten die schlecht klassifiziert haben, eine Verbesserung der Erkennungsraten mit sich bringt. Dies ist möglich, da die Trainingsdaten für diese Knoten so ineinander verschränkt sein können, dass eine lineare Trennung nicht möglich ist. Für diesen Fall wäre z. B. die Verwendung des Polynomial-Kernels, ab dem Grad 2, oder des RBF-Kernels sinnvoller.

## 7. Fazit und Ausblick

Diese Arbeit zeigt, dass es mit genügend großen Merkmalsvektoren möglich ist, auch Instrumente desselben Instrumententyps zu unterscheiden. Es sollten jedoch noch verschiedene Testläufe mit anderen Kernen und Kernparametern absolviert werden, um eventuell bessere Ergebnisse zu erzielen.

Auch kann analysiert werden, wie groß der Einfluss eines Musikers ist. Dazu kann man ermitteln, inwiefern sich die Klassifikationsergebnisse ändern, wenn mit Klängen trainiert wird, die alle von einem Musiker gespielt werden und mit Klängen anderer Musiker getestet wird. Dies gibt einen Anhaltspunkt, wie stark der Klang einer Gitarre von der Spielweise eines Musikers abhängt.

Weiterhin kann untersucht werden, wieviele Klänge benötigt werden, um eine Gitarre zuverlässig einer Klasse zuzuordnen zu können. Die vorliegenden Testergebnisse wurden mit allen zur Verfügung stehenden Klängen durchgeführt. Daher ist es sinnvoll zu untersuchen, inwieweit sich die Erkennungsraten ändern, wenn z. B. nur die Klänge, die auf der ersten Saite gespielt wurden, klassifiziert werden. Werden verschiedene Experimente mit, auf unterschiedlichen Saiten und Bündeln gespielten Klängen, durchgeführt, so könnte man feststellen, ob bestimmte Noten besonders schlecht oder gut erkannt werden. Ergäbe sich aus den Erkennungsraten solcher Experimente eine Systematik, so könnte man bestimmen welche Klänge notwendig sind, um eine Gitarre zuverlässig einer Klasse zuzuordnen. Dies hätte den Vorteil, dass in einer praktischen Anwendung nur einzelne Klänge gespielt werden müssten und nicht der gesamte hier verwendete Notenumfang.

Um die entwickelte Software praktisch einsetzen zu können, bedarf es noch einiger Änderungen. So sollte bei der Klassifikation ausgegeben werden, zu welchem Prozentsatz sich die Gitarren den Klassen zuordnen lassen. Diesen Prozentsatz kann man mit Hilfe des Klassifikationsergebnisses errechnen. Der Betrag des Wertes der Entscheidungsfunktion dient dabei als Ähnlichkeitsmaß. Je größer der Betrag des Ergebnis, desto wahrscheinlicher gehört die Gitarre zu der klassifizierten Klasse, d. h. je größer der Wert ist, desto weiter liegt der Merkmalsvektor von dem Margin entfernt.

Zusätzlich wäre eine Änderung der verwendeten SVM-Implementierung sinnvoll, so dass die Klassifikationsergebnisse nicht in einer Datei gespeichert, sondern als Ergebnis der Klassifizierungsfunktion zurückgegeben werden. Dadurch kann die Klassifizierungszeit verkürzt werden, da laufzeitintensive Dateioperationen vermieden werden können.

# Literaturverzeichnis

- [CST00] Cristianini, Nello und Shawe-Taylor, John. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. ISBN 0521780195.
- [DSC06] Deng, Da, Simmermacher, Christian und Cranefield, Stephen. *Finding the Right Features for Instrument Classification of Classical Music*. In *International Workshop on Integrating AI and Data Mining, 2006. AIDM '06*, Seiten 34–41, 2006.
- [FPE<sup>+</sup>06] Fragoulis, Dimitrios, Papaodysseus, Constantin u. a.. *Automated classification of piano-guitar notes*. *IEEE Transactions on Speech and Audio Processing*, Band 14(3):Seiten 1040– 1050, 2006.
- [FR98] Fletcher, N. H. und Rossing, T. D. *The Physics of Musical Instruments*. Springer Verlag, New York, 1998.
- [Hes06] Hess, Wolfgang. *Grundlagen der Sprachsignalverarbeitung, Kapitel 4: Vom Spektrum zum Spektrogramm*, 2006. URL [http://www.ikp.uni-bonn.de/dt/lehre/materialien/grundl\\_ssv/gsv\\_4b.pdf](http://www.ikp.uni-bonn.de/dt/lehre/materialien/grundl_ssv/gsv_4b.pdf). – Zugriffsdatum: Juli 2007.
- [Joa99] Joachims, Thorsten. *Making large-Scale SVM Learning Practical*. In B. Schölkopf, C. Burges und A. Smola (Herausgeber), *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999. URL <http://svmlight.joachims.org/>.
- [Mey85] Meyer, Jürgen. *Akustik der Gitarre in Einzeldarstellungen*. Das Musikinstrument, Band 42. Verlag Erwin Bochinsky, 1985. ISBN 3-923639-66-X.
- [MM99] Marques, Janet und Moreno, Pedro J. *A study of musical instrument classification using gaussian mixture models and support vector machines*. *Technischer Bericht*, Cambridge Research Laboratory, 1999. URL <http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-99-4.pdf>. – Zugriffsdatum: Mai 2007.
- [PCST00] Platt, J., Cristianini, N. und Shawe-Taylor, J. *Large Margin DAGS for Multiclass Classification*. In S.A. Solla, T.K. Leen und K.-R. Mueller (Herausgeber), *Advances in Neural Information Processing Systems 12*, Seiten 547–553, 2000.

- [Sla98] Slaney, Malcolm. *Auditory Toolbox. A MATLAB Toolbox for Auditory Modeling Work. Version 2*, 1998. URL <http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/>. – Zugriffsdatum: Juni 2007.
- [SS02] Schölkopf, Bernhard und Smola, Alexander J. *Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002. ISBN 0-262-19475-9.
- [STD<sup>+</sup>04] Steelant, D. Van, Tanghe, K. u. a.. *Classification of Percussive Sounds using Support Vector Machines*. In *Proceedings of the annual machine learning conference of Belgium and The Netherlands, Brussels, Belgium, January 8-9, 2004*. URL [http://www.ipem.ugent.be/MAMI/Public/Papers/ClassificationPercussionSVM\\_Benelearn2004.pdf](http://www.ipem.ugent.be/MAMI/Public/Papers/ClassificationPercussionSVM_Benelearn2004.pdf).
- [Ste03] Stemmer, Georg. *Algorithmen der Mustererkennung und ihre Realisierung 2, Merkmalberechnung, Cepstrum, Mel-Cepstrum*, 2003. URL <http://www5.informatik.uni-erlangen.de/Lehre/SS05/amer2-v/>. – Zugriffsdatum: 17 Juli 2007.
- [Vap98] Vapnik, Vladimir N. *Statistical Learning Theory*. Adaptive and learning Systems for signal processing, communications and control. A Wiley-Interscience publication. Wiley, 1998.
- [Vap00] Vapnik, Vladimir N. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer-Verlag, 2000.
- [Voi07] Voigt, Julia. *Klanguntersuchungen an Konzertgitarren: Korrelation zwischen physikalischen Signalmerkmalen und psychoakustischen Attributen*. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2007. URL [http://www.informatik.haw-hamburg.de/fileadmin/Homepages/ProfFohl/Abschlussarbeiten/DA\\_Voigt.pdf](http://www.informatik.haw-hamburg.de/fileadmin/Homepages/ProfFohl/Abschlussarbeiten/DA_Voigt.pdf).
- [Wik] Wikipedia:Mel\_Scale. *Wikipedia: the free Encyclopedia*. URL [http://en.wikipedia.org/wiki/Mel\\_scale](http://en.wikipedia.org/wiki/Mel_scale). – Zugriffsdatum: Juli 2007.
- [Zit06] Zitzelsberger, Michael. *Mustererkennung unter Verwendung der Stützvektormethode*. Diplomarbeit, Fachhochschule Regensburg, 2006. URL <http://www.opus-bayern.de/fh-regensburg/volltexte/2006/24/>.

# A. Verwendete und entwickelte Programme

In diesem Kapitel wird beschrieben, wie die entwickelten Programme zu benutzen sind. Dabei wird zuerst auf die Merkmalsextraktion eingegangen, danach auf das Trainieren und Klassifizieren. Alle verwendeten Octave- und C-Programme, sowie die benutzten Audiodateien, sind auf der, zur Diplomarbeit gehörenden, DVD vorhanden (siehe Anhang B). Alle Programme wurden unter Ubuntu, einer auf Debian basierenden Linux Distribution, entwickelt und getestet. Sie sollten aber auch unter Windows einsetzbar sein. Zum Starten der entwickelten Programme müssen die Programme Octave und SVM<sup>light</sup> installiert sein.

## A.1. Programm zur Merkmalsextraktion

Die Merkmalsextraktion wird über die Funktion `start.m` gestartet. Hier werden die notwendigen Variablen, wie z.B. der Verzeichnisname zu den Trainings- und Testdaten<sup>1</sup> gesetzt. Des Weiteren wird mit Hilfe der Variablen `calcNontonal`, `calcMFCCs` und `calcTemporal` angegeben welche Merkmale berechnet werden sollen. Die Grundfrequenz wird immer berechnet, da sie für die Berechnung der nontonalen Peaks und des zeitlichen Verlaufs benötigt wird und darüber hinaus auch in den Merkmalsvektor aufgenommen wird.

Listing A.1 zeigt einen Ausschnitt der Datei `start.m` auf.

```
#!/usr/bin/octave --silent
# -----
# Liest die Ordner die die Test- und Trainingsdaten
# halten aus und schreibt alles in Dateien um trainieren
# und testen zu koennen
# -----
```

---

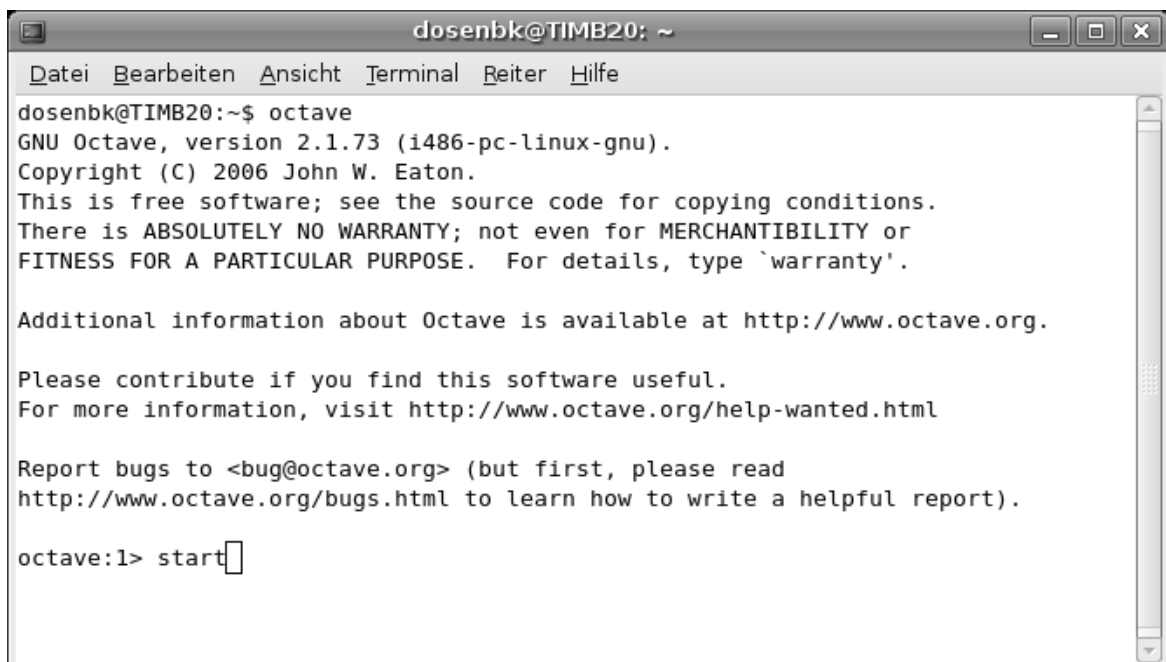
<sup>1</sup>In der vorliegenden Version des Programm werden sowohl die Trainings-, als auch die Testdaten ausgelesen und die Merkmalsextraktionsverfahren darauf angewendet. In nachfolgenden Versionen wäre es sinnvoll dies zu trennen, so dass z.B. die Merkmalsextraktion nur auf die Testdaten angewendet wird, um sie anschließend gleich zu klassifizieren.

```
# Verzeichnis in dem die Audiodateien liegen
AUDIODIR = '/home/dosenbk/Audio/';
# Verzeichnisname der Audio-Test- bzw. Trainingsdaten
TrainAudio = 'Train';
TestAudio = 'Test';
# Verzeichnis in das die extrahierten Trainingsdaten geschrieben werden
  sollen
TRAINDATA = '/home/dosenbk/svmOutput/traindata/';
# Verzeichnis in das die extrahierten Testdaten geschrieben werden
  sollen
TESTDATA = '/home/dosenbk/svmOutput/testdata/';

# Mindestens 1 Variable muss auf 1 gesetzt werden
# 1: Merkmal berechnen, 0: Merkmal nicht berechnen
calcNontonal = 1; # Die Peaks des nontonalen Spektrums.
calcMFCCs = 1; # MFCCs
calcTemporal = 1; # Zeitl. Verlauf der Harmonischen
```

Listing A.1: Ausschnitt aus der Datei start.m.

Die Funktion wird entweder direkt aus Octave oder von der Kommandozeile aufgerufen. Octave wird dabei mit Hilfe des Befehls `octave` gestartet. Danach kann die Funktion mit `start` aufgerufen werden.



```
dosenbk@TIMB20: ~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
dosenbk@TIMB20:~$ octave
GNU Octave, version 2.1.73 (i486-pc-linux-gnu).
Copyright (C) 2006 John W. Eaton.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

octave:1> start
```

Abbildung A.1.: Start des Programmes aus Octave.

Um das Programm unter Linux von der Kommandozeile aus zu starten, muss die erste Zeile

(siehe Listing A.1) eingefügt werden, dadurch wird der Shell der zu verwendende Interpreter mitgeteilt. Die Option `silent` unterdrückt die Octave-Begrüßung. Nach dem das Skript mit Hilfe des Kommandos `chmod +x start.m` ausführbar gemacht wurde, kann es an der Kommandozeile gestartet werden. Unter Windows ist das Einfügen der ersten Zeile nicht notwendig, da die Version für Windows eine Kommandozeile beinhaltet, aus der die Funktionen aufgerufen werden können.

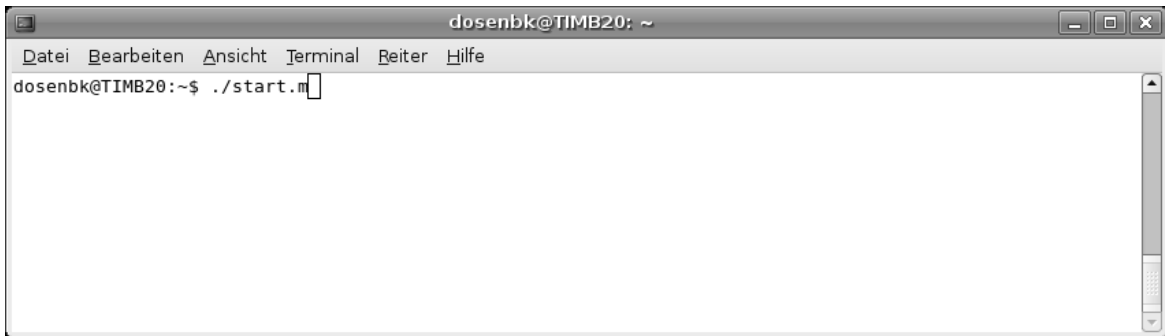


Abbildung A.2.: Start des Programmes von der Kommandozeile.

Im Folgenden sind alle Octave-Funktionen aufgelistet, die zur Merkmalsextraktion benötigt werden:

**start.m** : Startet die Merkmalsextraktion.

**getFeatures.m** : Steuert die Merkmalsextraktion und gibt eine Matrix mit allen Merkmalsvektoren einer Gitarre zurück.

**wavRead.m** : Liest die Audiodateien ein. Entwickelt von Doug Stewart (siehe Kapitel 5.2).

**getF0.m** : Berechnet die Grundfrequenz.

**getNontonalSpektrum.m** : Berechnet das nontonale Spektrum. Geschrieben von Prof. Fohl<sup>2</sup>.

**peakdetect.m** : Berechnet die Peaks des nontonalen Spektrums. Entwickelt von Eli Billauer<sup>3</sup>.

**mfcc.m** : Berechnet die MFCCs. Programmiert von Malcolm Slaney<sup>4</sup>.

<sup>2</sup>Homepage: <http://www.informatik.haw-hamburg.de/fohl.html>

<sup>3</sup>Homepage: <http://www.billauer.co.il/peakdet.html>

<sup>4</sup><http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/>

**getTemporalCourseHarmonics.m** : Berechnet den zeitlichen Verlauf der Harmonischen. Entwickelt von Julia Voigt<sup>5</sup>.

**writeToFileOneVsOne.m** : Schreibt die Dateien für das Training.

**writeTestFile** : Schreibt die Dateien für die Testfälle.

## A.2. Programm zum Trainieren der SVMs

Das Training der SVMs wird über die Funktion `learn.m` gestartet. Die Funktion kann dabei, wie auch bei der Merkmalsextraktion, aus Octave oder von Kommandozeile gestartet werden.

In Listing A.2 ist die Funktion `learn.m` aufgeführt.

```
#!/usr/bin/octave --silent

SVMOUTPUT = '/home/dosenbk/svmOutput/';
# Verzeichnis der Trainingsdaten
TRAINDATA = '/home/dosenbk/svmOutput/traindata/';
# Verzeichnis der trainierten SVMs
MODELS = '/home/dosenbk/svmOutput/models/';
#

# Optionen, siehe svmlopt fuer Argumente
# Optionen fuer LinearKernel
optionsLinear = svmlopt('Verbosity', 1,
                       'C', 1.0, 'Biased', 1, 'Kernel', 0);

trainDataList = dir(TRAINDATA);
# Solange Trainingsdaten vorhanden sind
for l=1:length(trainDataList)
    if trainDataList(l).isdir != 1
        filename = trainDataList(l).name;
        filepath = strcat(TRAINDATA, filename);
        idx = strfind(filename, '.');
# Trainierte SVM in diese Datei schreiben
        modelname = strcat(filename(1:idx-1), '.net');
        modelpath = strcat(MODELS, modelname);
# Ausgabe des Trainings in diese Datei schreiben
        svmOutput = strcat(filename(1:idx), 'linear');
        svmOutputPath = strcat(SVMOUTPUT, svmOutput);
# Trainieren
        svm_learn(optionsLinear, filepath, modelpath, svmOutputPath);
```

<sup>5</sup>[http://www.informatik.haw-hamburg.de/fileadmin/Homepages/ProfFohl/Abschlussarbeiten/DA\\_Voigt.pdf](http://www.informatik.haw-hamburg.de/fileadmin/Homepages/ProfFohl/Abschlussarbeiten/DA_Voigt.pdf)



```
end %if  
end %for
```

Listing A.2: Die Datei learn.m.

Im Folgenden sind alle Programme, die zum Training der SVMs benötigt werden, aufgelistet:

**learn.m** : Startet das Training.

**svm\_learn.m** : Matlab-Interface zum Programm svm\_learn. Geschrieben von Anton Schwaighofer<sup>6</sup>.

**svmlopt.m** : Mit Hilfe dieser Funktion werden die Parameter für die zu trainierende SVM festgelegt. Geschrieben von Anton Schwaighofer.

**svm\_learn** : Trainingsfunktion der SVM<sup>light</sup>. Entwickelt von Thorsten Joachims<sup>7</sup>.

### A.3. Programm zum Testen

Um zu überprüfen wie die trainierten Support Vector Machines klassifizieren, wurde die Funktion `testdagsvm.m` geschrieben. Sie erwartet alle zum Test verwendeten Merkmale jeweils in einer Datei. Diese werden einzeln ausgelesen und mit Hilfe eine DAG klassifiziert.

Um zu überprüfen, wie die einzelnen SVMs klassifizieren, wird die Ausgabe der Funktion in eine Datei geschrieben. Diese beinhaltet unter anderem die Gesamterkennungsrate, aber auch die Ergebnisse der einzelnen Knoten des DAG.

Im Folgenden sind alle Programme, die zum Testen der SVMs benötigt werden, aufgelistet:

**testdagsvm.m** : Funktion um die trainierten SVMs zu testen. Enthält den DAG.

**svmlopt.m** : Legt den Ausgabelevel des Trainings fest. Geschrieben von Anton Schwaighofer.

**svm\_classify** : Matlab-Interface zum Programm svm\_classify. Entwickelt von Anton Schwaighofer.

**svm\_classify** : Klassifizierungsfunktion der SVM<sup>light</sup>. Entwickelt von Thorsten Joachims.

<sup>6</sup>Download unter: <http://ida.first.fraunhofer.de/~anton/software.html>

<sup>7</sup>Download unter: <http://svmlight.joachims.org/>

## B. Inhalt der DVD

**Audio/** Beinhaltet die zum Training und Testen verwendeten Audiodateien. In dieser Struktur werden die Audiodateien von dem Merkmalsextraktionsprogramm erwartet.

**Train/** Trainingsdaten

**Hense/**

**Marin/**

**Wichmann/**

**Test/** Testdaten

**Hense/**

**Marin/**

**Wichmann/**

**Experimente/** Enthält die einzelnen Experimente

**ExperimentX/** Verzeichnisstruktur der Experimente, mit  $X = 1-7$

**ExtrahierteTestdaten/** Beinhaltet die zum Testen verwendeten Dateien

**ExtrahierteTrainingsdaten/** Beinhaltet die zum Training verwendeten Dateien

**SVM/** Beinhaltet die Ausgaben der SVMs

**TrainierteSVMs/** Beinhaltet die drei trainierten SVMs

**Trainingsoutput/** Beinhaltet die Trainingsausgaben der drei SVMs

**Testoutput/** Beinhaltet die Ergebnisse des Testlaufs

**Software/** Entwickelte Programme

**Merkmalsextraktion/** Programme zur Merkmalsextraktion

**Trainieren/** Programme zum Trainieren der SVMs

**Testen/** Programme zum Testen der trainierten SVMs

## C. Verwendete Mathematische Notation

$\mathbb{R}$	Die Menge der reellen Zahlen
$X$	Wertebereich der Merkmalsvektoren
$\mathbf{x}_i$	Merkmalsvektor
$y_i$	Klassenlabel des Merkmalsvektors $\mathbf{x}_i$ , $y_i \in \{+1, -1\}$
$\mathbf{w}$	Normalenvektor der Hyperebene
$b$	Konstanter Term, der den Abstand der Hyperebene zum Ursprung bezeichnet.
$\langle \mathbf{x}, \mathbf{x}' \rangle$	Skalarprodukt zwischen den Vektoren $\mathbf{x}$ und $\mathbf{x}'$
$\ \mathbf{w}\ $	Norm des Vektors $\mathbf{w}$
$\alpha_i$	Lagrange Multiplikator
$\mathbf{x}_{\text{neu}}$	Der zu klassifizierende Merkmalsvektor
$f(\mathbf{x}_{\text{neu}})$	Entscheidungsfunktion um $\mathbf{x}_{\text{neu}}$ einer Klasse zuzuordnen
$\text{sgn}$	Signumsfunktion
$\Phi$	Nichtlineare Abbildung, $\Phi : X \rightarrow F$
$F$	Featurespace
$k(\mathbf{x}_i, \mathbf{x}_{\text{neu}})$	Kernelfunktion

# Glossar

**DFT** Diskrete Fourier-Transformation.

**Fensterfunktion** In der Regel werden bei der Fouriertransformation nur einzelne Signalausschnitte verwendet. Ist die Länge dieses Signalausschnitts nicht ein Vielfaches der Periodenlänge, wird das Spektrum verfälscht. Um dies zu verhindern, wird der Signalausschnitt mit einer Fensterfunktion gewichtet. Das hat zur Folge, dass die Randbereiche des Signalblocks abgeschwächt werden, was zu einer künstlichen Periodisierung führt.

**FFT** Fast Fourier Transform (dt. Schnelle Fourier-Transformation).

**Formanten** Besonders ausgeprägte Frequenzbereiche. Formanten sind abhängig vom Resonanzverhalten der Instruments. Jeder Formant hat einen bestimmten Frequenzbereich.

**Framerate** Gibt bei der Berechnung der MFCCs an, wie oft die MFCC-Berechnung pro Sekunde durchgeführt wird.

**IFFT** Inverse Fast Fourier Transform (dt. Inverse Schnelle Fourier-Transformation).

**MFCC** Mel Frequency Cepstral Coefficients.

**Sample** Abtastwert. Bezeichnet den diskreten Wert einer Messung.

**Samplerate** Die Samplerate gibt die Häufigkeit an, mit der das Signal beim Digitalisieren pro Sekunde abgetastet wird.

**SVMs** Support Vector Machines

**Tonheit** Die vom Menschen wahrgenommene Tonhöhe.

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. August 2007

---

Ort, Datum

---

Unterschrift