Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Bachelorarbeit (Studiengang Informatik)

Talkalyzer: Neue Algorithmen für automatische Sprechererkennung

Autor	Jan Stampfli
Hauptbetreuung	Dr. Thilo Stadelmann Dr. Mark Cieliebak
Datum	02.06.2014

Erklärung betreffend das selbständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

……

Unterschriften:

……...

……

……

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Abstract bzw. dem Management Summary mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Zusammenfassung

Die heute gängigen Verfahren der automatischen Sprechererkennung befinden sich auf einem Stand, der den Einsatz in kommerziellen Bereichen, wie z.B. der Stimmerkennung in Sicherheitssystemen, ermöglicht. Werden aber die Bedingungen, beispielsweise durch fehlendes Wissen über die Anzahl und Identität vorhandener Sprecher, zu komplex, bietet der heutige Stand der Verfahren keine praktikable Verwendung. Dieser Arbeit vorangegangene Forschung zeigt, dass im zeitlichen Verlauf von Audiosignalen wichtige sprecherspezifische Information enthalten ist, die aber in den gängigen Systemen vernachlässigt wird.
In der vorliegenden Arbeit werden verschiedene Ansätze hinsichtlich der Integration zeitlicher Aspekte von Audiosignalen untersucht, mit dem Ziel, die bestehenden Verfahren der automatischen Sprechererkennung zu verbessern. Die Erkennungsleistung des aus diesen Ansätzen erarbeiteten Konzepts wird anhand von Clustering-Experimenten verifiziert. Clustering beinhaltet in Bezug auf die automatische Sprechererkennung das Clustern einer unbekannten Anzahl Sprecher, so dass sich im besten Fall nur Daten eines einzelnen Sprechers in einem Cluster befinden und pro Sprecher nur ein einziger Cluster existiert. Den Experimenten liegen Audiodaten aus dem TIMIT-Datensatz zugrunde, der von 630 englischsprachigen Sprechern in Studioqualität aufgezeichnete Aussagen enthält. Daraus werden zwei Sprechersets verwendet, einmal mit 20 und einmal mit 40 unterschiedlichen Sprechern.
Das erarbeitete Konzept sieht vor, dass die zeitliche Information von Audiosignalen anhand eines kleinen Sets von 30 Filtern aus Spektrogrammen extrahiert wird. Eine entscheidende Rolle spielt dabei die Selektion dieses Sets von Filtern, wobei auf eine Vorgehensweise aus der automatischen Objekterkennung, einem Teilbereich der *Computer Vision*, zurückgegriffen wird. Grundsätzlich werden aus einem grossen Set von potentiellen Filtern mit Hilfe des *AdaBoost*-Algorithmus die entscheidenden Filter selektiert. Nach der Selektion der Filter besteht die Durchführung des Clusterings zunächst aus dem Laden aller Audiodaten eines Sprechersets und dem anschliessenden Erstellen von Spektrogrammen. Weiter werden mit dem selektierten Filterset die Sprechermerkmale aus den Spektrogrammen extrahiert, die auch den mittellangen zeitlichen Verlauf des Audiosignals beinhalten. Anhand der Sprechermerkmale werden probabilistische Modelle erzeugt, welche die Wahrscheinlichkeitsverteilung der Merkmale wiedergeben. Abschliessend werden die Distanzen zwischen den Modellen berechnet und anhand dieser Distanzen zu Clustern zusammengefasst.
Die Ergebnisse wurden verglichen mit denen eines Baseline-Ansatzes, der die gängigsten Verfahren der automatischen Sprechererkennung beinhaltet. Die besten Resultate wurden beim Clustering von 40 unterschiedlichen Sprechern erzielt, wobei eine Präzision von 87.37% und eine Ausbeute von 83.05% gegenüber dem Baseline-Ansatz mit 89.10% Präzision und 87.24% Ausbeute resultierte. Obwohl die erzielten Ergebnisse zeigen, dass keine Verbesserung der bestehenden Verfahren erzielt werden konnte, so bietet diese Arbeit dennoch eine Grundlage für weiterführende Forschung und zeigt, welche Schritte des erarbeiteten Konzepts Anpassung benötigen, um die Erkennungsleistung entscheidend zu steigern.

Abstract

The level of today's prevalent methods of automatic speaker recognition allows the use in commercial areas, such as voice recognition in security systems. However, if the conditions become too complex, for example by lack of knowledge about the number and identity of existing speakers, the methods at their current level are of no practical use. Previous research shows that important speaker-specific information is contained in the temporal course of audio signals. However, this has been neglected in the current systems.
In the present paper, different approaches to the integration of temporal aspects of audio signals are investigated, with the aim to improve the existing methods of automatic speaker recognition. The recognition performance of the concept developed from these approaches is verified by clustering experiments. Clustering, in terms of automatic speaker recognition, means clustering an unknown number of speakers, in order to receive clusters with data from only one single speaker at a time and to receive only one single cluster for each speaker. The experiments are based on audio data from the TIMIT data set containing the recorded statements of 630 English speaking people in studio quality. From this data set two speaker sets are used, once with 20 and once with 40 different speakers.
The developed concept allows temporal information of audio signals to be extracted from spectrograms with a small set of 30 filters. The selection of this set of filters plays a crucial role and is reached by a procedure known from the automatic object recognition, a subdomain of *Computer Vision*. Basically, the key filters are selected from a large set of potential filters using the *AdaBoost* algorithm. After selecting the filters, implementing the clustering process initially consists of loading all audio data of a speaker set and subsequently creating spectrograms. On the basis of the selected filter set, speaker characteristics are extracted from the spectrograms, which also include the medium-term time course of the audio signal. Probabilistic models are generated from the speaker characteristics, which reflect the probability distribution of the characteristics. Finally, the distances between the models are calculated and, based on these distances, summarized into clusters.
The results were compared with those of a baseline approach, which includes the most common methods of automatic speaker recognition. The best results were achieved when clustering 40 different speakers. Precision amounted to 87.37% and recall to 83.05% compared with the baseline approach where precision amounted to 89.10% and recall to 87.24%. Although the results show that no improvement in the existing process could be achieved, this paper nevertheless offers a basis for further research and names the steps of the developed concept which need to be adapted to increase the recognition performance significantly.

Inhaltsverzeichnis

1. Einleitung

1.1. Motivation

Seit jeher ist die Stimme ein wesentliches Merkmal, woran sich Menschen erkennen. Bereits im Mutterleib reagieren Kinder auf die Stimmen ihrer Eltern und prägen die Fähigkeit, Stimmen zu unterscheiden, nach der Geburt rasch weiter aus. Das menschliche Gehör ist in der Lage, feinste Unterschiede von Stimmen wahrzunehmen und so zur Erkennung und Unterscheidung von Personen einzusetzen und ist bis heute im Bereich der Sprechererkennung unübertroffen. Aus diesem Grund versuchen viele der heute gängigen Systeme, die Vorgänge im menschlichen Gehörapparat zu imitieren, um dadurch zumindest annähernd dieselbe Erkennungsleistung zu erreichen.
Diese Arbeit ist motiviert durch das Projekt *Talkalyzer*, ins Leben gerufen durch das Institut für angewandte Informationstechnologie (InIT) der Zürcher Hochschule für Angewandte Wissenschaften (ZHAW). Das Ziel des Projekts ist die Entwicklung einer Smartphone-Applikation, die zur Erkennung von Sprechern eingesetzt werden kann. Für den Erfolg einer solchen Applikation bedarf es jedoch weiterer Forschung, um die Verlässlichkeit der bestehenden Verfahren zu erhöhen.
Auch in anderen Bereichen ist eine verlässliche Sprechererkennung von grosser Bedeutung. Die Sicherheitssysteme von VoiceTrust[28] beispielsweise schützen sensible Daten durch Verifikation der Stimme und versprechen dadurch die Verhinderung von Missbrauch durch unautorisierte Personen. Die *International Association for Forensic Phonetics and Acoustics*[24] ist eine Plattform zum Austausch von Wissen und Ideen im Bereich der forensischen Sprechererkennung und ist bestrebt, Standards für die praktische Anwendung in der Strafverfolung festzulegen. Auch die immer schneller wachsende Menge elektronischer Daten, und damit auch Audiodaten, hat ein Interesse an der Analyse und Kategorisierung eben dieser geschaffen. Z.B. enthält das Internet in seiner heutigen Form eine Unmenge an Information, die, richtig aufbereitet, auch für kommerzielle Zwecke genutzt werden kann. Diese nicht abschliessende Liste von Anwendungen der Sprechererkennung trägt zusätzlich zur Motivation für diese Arbeit bei.

1.2. Zielsetzung

Das grundlegende Ziel dieser Arbeit ist es, bestehende Verfahren im Bereich der automatischen Sprechererkennung, genauer im Bereich des Clusterings (vgl. 2.1), anhand neuer Ansätze zu verbessern. Es wird nach einem Weg gesucht, sprecherspezifische Information aus Audiosignalen zu gewinnen, die von den herkömmlichen Systemen bisher nicht berücksichtigt bzw. verworfen wurde. Als Grundlage dienen zahlreiche Arbeiten aus verwandten Forschungsgebieten. Daraus sind geeignete Ansätze auszuwählen und auf die vorliegende Problemstellung zu übertragen. Erfolgreiche Verbesserungen, die daraus resultieren, kommen im Rahmen des Projekts *Talkalyzer* in einer Smartphone-Applikation zur Anwendung. Die Einbindung die Applikation, die parallel zu dieser Arbeit entwickelt wird (vgl. A.1.2), ist jedoch als langfristiges Ziel definiert, da im Rahmen dieser Arbeit die dafür notwendige effiziente Anwendbarkeit der entwickelten Algorithmen zweitrangig ist.
Eine wichtige Anforderung an die Resultate der vorliegenden Arbeit ist die Vergleichbarkeit, bezogen auf vorangegangene Forschung in diesem und verwandten Bereichen. Zusätzlich ist grosser Wert darauf zu legen, dass die Resultate aussagekräftig und präzise formuliert sind und dabei keine Informationen vorenthalten werden, welche die Ergebnisse entscheidend beeinflussen. Die dieser Arbeit zugrundeliegende Aufgabenstellung ist im Anhang unter A.1.1 zu finden.

1.3. Aufbau der Arbeit

Der weitere Aufbau dieser Arbeit gliedert sich in folgende Kapitel:
Kapitel 2 beinhaltet die Einführung in die Thematik der automatischen Sprechererkennung und verschafft gleichzeitig einen Überblick über relevante Ansätze und Arbeiten bezüglich der gegebenen Problemstellung.
Kapitel 3 befasst sich zunächst mit der Auswahl eines im vorhergehenden Kapitel beschriebenen Ansatzes und beschreibt anschliessend das Konzept, das diesen Ansatz auf die gegebene Problemstellung anwendet.
Kapitel 4 geht auf die Implementierung der durchgeführten Experimente sowie die verwendeten Soft- und Hardwarekomponenten ein.
Kapitel 5 beschreibt die Umsetzung der Experimente anhand der entscheidenden Parameter und definiert die Messgrössen sowie den verwendeten Datensatz. Zudem werden die erzielten Resultate der Experimente dargestellt und eingehend besprochen.
Kapitel 6 fasst die vorliegende Arbeit zusammen und erörtert im Ausblick Ansätze der weiterführenden Forschung.

2. Grundlagen

2.1. Sprachproduktion und Sprachverarbeitung

Um die heute gängigen Systeme der automatischen Sprechererkennung zu verstehen, muss zunächst verstanden werden, wie gesprochene Sprache erzeugt und vom Sprechenden beeinflusst wird. Gesprochene Sprache wird mit Hilfe der Stimmbänder erzeugt. Direkt nach dem Erzeugen des Audiosignals ist bereits sprecherspezifische Information enthalten, beispielsweise ausgedrückt durch die vorhandene Grundfrequenz. Ein erzeugtes Signal wird im Vokaltrakt weiter geformt, beeinflusst durch Zunge, Zähne, Form des Rachens und weitere Faktoren. Dieser Vorgang verleiht dem gesprochenen Audiosignal zusätzliche sprecherabhängige Information, entscheidet also darüber, wie einzelne Phoneme eines Sprechers klingen. Für detaillierte und weiterführende Informationen dazu, wird auf die Arbeit von Stadelmann [20], Seite 31 ff., verwiesen.
Eine grundlegende Anforderung ist es nun, diese charakteristischen Eigenschaften aus dem Signal zu gewinnen, um sie für die Erkennung einzusetzen.
Die Sprechererkennung kann in die drei Kategorien Verifizierung, Identifizierung und Clustering unterteilt werden. Verifizierung beinhaltet das Erkennen eines Sprechers anhand eines zuvor trainierten Sprechermodells. Ein Sprechermodell kann als Zusammenfassung der charakteristischen Merkmale eines Sprechers verstanden werden und wird in Abschnitt 2.2.3 genauer betrachtet. Es wird berechnet, mit welcher Wahrscheinlichkeit ein gegebenes Audiosignal vom trainierten Sprechermodell stammt und anhand eines festgelegten Grenzwertes bestimmt, ob es sich um den entsprechenden Sprecher handelt oder nicht. Bei der Identifikation von Sprechern geht es darum, für ein gegebenes Audiosignal zu entscheiden, welchem Sprechermodell es zuzuordnen ist. Dazu sind vorab mehrere Sprechermodelle zu trainieren, wodurch die Komplexität der Aufgabe im Vergleich zur Verifikation zunimmt. Beim Clustering wird für gegebene Audiodaten mit unterschiedlichen Sprechern festgestellt, welche Bereiche aus den gegebenen Daten zu welchem Sprecher gehören. Dazu werden die Audiodaten in Segmente unterteilt, die jeweils möglichst nur Daten eines einzigen Sprechers enthalten. Anschliessend werden aus diesen Segmenten, ohne Kenntnis über die effektive Anzahl unterschiedlicher Sprecher, Modelle erzeugt. Mittels dem eigentlichen Clustering werden nun die vorhandenen Modelle so kombiniert bzw. geclustert, dass sich möglichst alle Modelle, die den gleichen Sprecher beschreiben, im selben Cluster befinden.
Die heutigen Systeme sind im Bereich Sprecherverifikation und -identifikation bereits sehr fortgeschritten. Durch die hohe Komplexität des Clusterings stossen die bestehenden Verfahren jedoch, u.a. abhängig von der Anzahl unterschiedlicher Sprecher, an ihre Grenzen (vgl. Stadelmann und Freisleben [21]).

2.2. Automatische Sprechererkennung

Im Bereich der automatischen Sprechererkennung gibt es eine grosse Vielfalt an unterschiedlichen Systemen mit unterschiedlichen Stärken und Schwächen. Viele der heute gängigen Sprechererkennungssysteme besitzen aber dennoch eine im Grundsatz übereinstimmende Vorgehensweise. Der typische Ablauf dieser Systeme lässt sich anhand von vier Modulen erläutern, dem *Preprocessing*, der *Feature Extraction*, dem *Modeling* und der *Recognition*. Abbildung 2.1 zeigt grob den Ablauf und das Zusammenspiel dieser Module, unabhängig vom eigentlichen Einsatzgebiet, wie z.B. Identifikation, wobei (a) die Wellenform eines Audiosignals zeigt, in (b) 100 *Mel Frequency Cepstral Coefficients* (vgl. 2.3.1) abgebildet sind, (c) ein *Gaussian Mixture Model* (vgl. 2.3.2) repräsentiert und (d) den Vergleich zweier *Gaussian Mixture Models* symbolisiert.

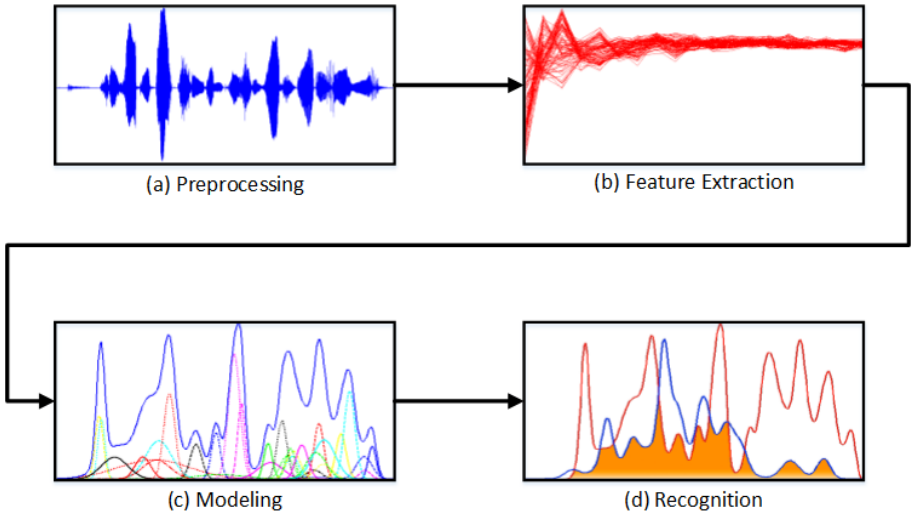

Abbildung 2.1.: Ablauf Sprechererkennung

Im Folgenden werden die einzelnen Module genauer erläutert, um dadurch das grundsätzliche Vorgehen der automatischen Sprechererkennung als Ganzes zu verdeutlichen.

2.2.1. Preprocessing

Der erste Schritt im Ablauf der automatischen Sprechererkennung ist das *Preprocessing*. Es umfasst alle notwendigen Aufgaben, um Audiodaten für die *Feature Extraction* vor- bzw. aufzubereiten. Dies beinhaltet zunächst das Entgegennehmen von Audiodaten, sei es beispielsweise aus Audiodateien oder direkt über ein Mikrophon. Im Anschluss werden die Daten in Frames eingeteilt, die als stationäre Ausgangspunkte (vgl. 2.3.1) für die weitere Verarbeitung eingesetzt werden. Ein Frame enthält dabei Audiodaten, gemessen über eine kurzen Zeitraum, typischerweise zwischen 20 bis 30 ms. In vielen Systemen werden die Frames während dem Erstellen normalisiert. Eine weitere Aufgabe, die während dem *Preprocessing* erledigt wird, ist das sogenannte *Pre-emphasizing*. Dies kann als Bearbeitung der Frames mittels eines Hochpass-Filters verstanden werden (vgl. Stadelmann [20], Seite 51). Dieser Vorgang verstärkt hohe Frequenzen im Signal und hebt sie gegenüber Störgeräuschen und irrelevanten Daten im tiefen Frequenzbereich hervor bzw. schwächt den Einfluss von Störgeräuschen und irrelevanten Daten ab.

2.2.2. Feature Extraction

Features bezeichnen in der Sprechererkennung Merkmale, die Sprecher voneinander unterscheiden und somit möglichst eindeutig identifizieren. Während der *Feature Extraction* wird daher pro Frame ein Featurevektor extrahiert. Dabei wird versucht, die entscheidenden Charakteristiken der Audiosignale innerhalb eines Frames zu verstärken und gleichzeitig irrelevante Informationen zu entfernen. Um Anpassung bzw. Kompression der Daten zu erreichen, gibt es sehr unterschiedliche Ansätze, wie beispielsweise die Berechnung von *Mel Frequency Cepstral Coefficients* (vgl. 2.3.1). Viele dieser Ansätze wurden bereits ausführlich analysiert und eingesetzt.
Nach dem Extrahieren von Features ist es zudem wichtig, zu erkennen, wann ein Sprecherwechsel stattfindet, um zusammengehörige Daten zu gruppieren. Im einfachsten Fall werden unterschiedliche Audiodateien mit jeweils nur einem Sprecher verwendet, so dass der Sprecherwechsel dadurch eindeutig festgestellt werden kann. Falls aber die verwendeten Daten beispielsweise direkt über ein Mikrophon aufgezeichnet werden, das von verschiedenen Sprechern benutzt wird, ist eine komplexe Segmentierung notwendig. Die segmentierten Features, im Folgenden als Featuresets bezeichnet, werden zur Erstellung von Sprechermodellen verwendet.

2.2.3. Modeling

Während dem *Modeling* werden anhand von Featuresets Sprechermodelle erzeugt und die Daten ein weiteres Mal komprimiert. Dies hat zur Folge, dass Vergleiche mit Modellen viel effizienter durchgeführt werden können als Vergleiche von Sets hochdimensionaler Vektoren, die auf Audioframes basieren. Die Modelle bilden somit die Grundlage für das Vergleichen von Sprechern. Ein Sprechermodell hat die Anforderung, die verfügbaren Daten möglichst präzise abzubilden und gleichzeitig einen Sprecher so generell wie möglich zu beschreiben. Passt sich ein Modell zu genau an die Trainingsdaten an, so werden zwar die entsprechenden Trainingsdaten jederzeit korrekt zugeordnet, neue Daten vom selben Sprecher oftmals jedoch nicht. Für die Umsetzung werden häufig probabilistische Modelle eingesetzt, z.B. *Gaussian Mixture Models*, welche die Aussprache von Phonemen eines Sprechers als Wahrscheinlichkeitsverteilung über die Featurevektoren abbilden (vgl. 2.3.2).

2.2.4. Recognition

Die *Recognition* stellt den letzten Schritt des Sprechererkennungsprozesses dar und hat zum Ziel, eindeutig zu bestimmen, welchem Modell eine Aussage zuzuordnen ist bzw. welche Modelle vom selben Sprecher stammen. Um diese Aufgabe zu lösen, werden häufig Distanzmessungen durchgeführt, welche die Wahrscheinlichkeit der Zugehörigkeit zwischen Aussage und Modell oder von Modellen zueinander angeben. Es wurden in der Vergangenheit verschiedene Distanzmasse vorgeschlagen, wie beispielsweise die *Generalized Likelihood Ratio* (GLR), die *Cross Likelihood Ratio* (CLR) oder das Distanzmass von Beigi et al. aus dem Jahr 1998 (vgl. Stadelmann[20], Seite 59 ff., und Beigi et al.[2]). Bei der Wahl des Distanzmasses sind Genauigkeit und Rechenaufwand oft die entscheidenden Kriterien, wobei grundsätzlich nur eines von beiden oder ein Kompromiss erreicht werden kann.

2.3. Baseline-Ansatz

In der vorliegenden Arbeit wird ein Baseline-Ansatz als Referenzsystem verwendet, der bereits vielfach angewandte und bewährte Verfahren aus der Sprechererkennung kombiniert und somit als solide Grundlage dient. Die eingesetzten Verfahren beziehen sich dabei auf *Feature Extraction*, *Modeling* und *Recognition* und wurden bereits 1995 von Reynolds und Rose[17] erfolgreich umgesetzt. Das *Preprocessing* wird in diesem Kapitel nicht weiter behandelt, da es keine Besonderheiten im Vergleich zu 2.2.1 aufweist.
In Abschnitt 2.3.1 werden die Features beschrieben, die im Baseline-Ansatz zum Einsatz kommen, und in Abschnitt 2.3.2 werden die während dem *Modeling* erstellten Sprechermodelle betrachtet. Unter 2.3.4 wird der Baseline-Ansatz bezüglich Erkennungsleistung analysiert. Detaillierte Angaben bezüglich verwendeter Parameter sind in Kapitel 5 beschrieben.

2.3.1. Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients (MFCC) wurden 1980 von Davis und Mermelstein[5] eingeführt und gehören heute zu den am häufigsten eingesetzten Features sowohl in der Sprecher- als auch in der Spracherkennung[20]. Um MFCC's zu extrahieren, ist im Vorfeld eine vereinfachende Annahme zu treffen. Für sich über die Zeit verändernde Audiosignale wird dabei angenommen, dass die innerhalb eines Frames enthaltenen Daten stationär (d.h. gleichbleibend) sind. Basierend auf dieser Annahme und motiviert durch das menschliche Gehör, werden die Frames, im Eintausch gegen den zeitlichen Verlauf, in das Frequenzspektrum transformiert. Dadurch werden die Frames anhand ihrer enthaltenen Frequenzen charakterisiert und so entscheidende sprecherspezifische Information offengelegt.
Da zwischen nahe beieinanderliegenden Frequenzen nur schwer unterschieden werden kann, werden in einem weiteren Schritt die Frequenzen anhand von Filtern aufsummiert bzw. zusammengefasst, woraus die sogenannten *Filterbank Energies* (FBE) resultieren. Dies geschieht mit Hilfe der *Mel Filterbank* (vgl. Abbildung 2.2), einem Set von Filtern, das Frequenzen anhand der *Mel Skala* (vgl. Abbildung 2.3)

zusammenfasst. Je näher die Frequenzen sich bei 0 Hz befinden, desto kleiner ist der Bereich, der durch die Filter zusammengefasst wird und umgekehrt. Weiterführende Informationen zur *Mel Filterbank* sind in der Arbeit von Stadelmann [20], Seite 52 f., beschrieben.

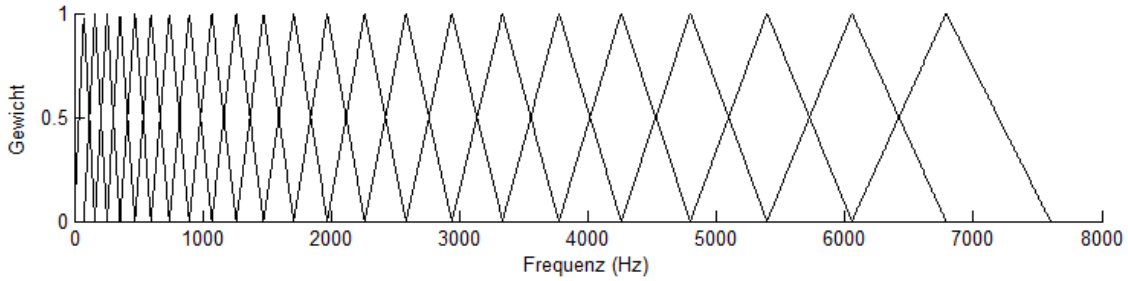

Abbildung 2.2.: Mel Filterbank mit 24 Filtern zwischen 0 und 7600 Hz

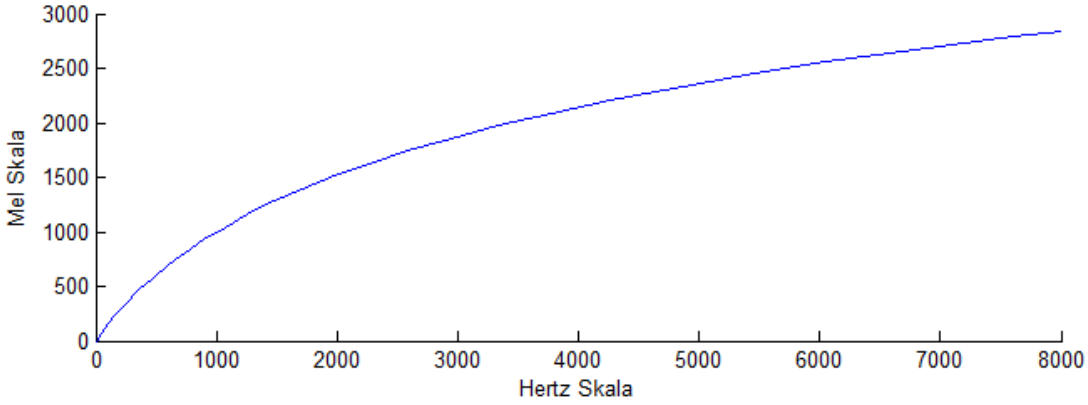

Abbildung 2.3.: Mel Skala der wahrgenommenen Tonhöhe gegenüber der Frequenz

Die FBE's werden anschliessend durch Logarithmieren der Wahrnehmung des menschlichen Gehörs angepasst. Da sich die verwendeten Filter stark überschneiden, gilt dies auch für die in den FBE's enthaltene Information. Um diese überschneidende Information zu trennen, werden im letzten Schritt die logarithmierten FBE's mittels der *Discrete Cosinus Transformation* (DCT) zu MFCC's transformiert und anschliessend die höheren der daraus resultierenden Koeffizienten verworfen, was als Glättung des Spektrums verstanden werden kann. Dies geschieht unter der Annahme, dass die höheren Koeffizienten schnelle Änderungen in den FBE's repräsentieren, welche die Erkennungsleistung negativ beeinflussen [25]. Ebenfalls ist in den MFCC's kein *Pitch* enthalten, also die Information über die Tonhöhe gegeben aus der Grundfrequenz einer Stimme.

2.3.2. Gaussian Mixture Model

Das *Gaussian Mixture Model* (GMM) wurde 1995 von Reynolds und Rose [17] in der Sprechererkennung eingeführt. Die Grundidee dieses Modells ist, dass der Klangraum der Stimme eines Sprechers durch ein Set von akustischen Klassen dargestellt werden kann [17]. Eine solche akustische Klasse wird dabei jeweils durch eine Gauss-Verteilung bzw. *Gaussian Mixture* (GM) modelliert, die nach Möglichkeit eine einzelne sprecherspezifische Phonemklasse abbildet. Die Anzahl akustischer Klassen, die für das erstellen eines GMM benötigt wird, hängt von den verfügbaren Trainingsdaten ab. Jede Verteilung beseht aus einem Durchschnittsvektor $\vec{\mu}$, einer Kovarianzmatrix Σ bzw. deren Diagonalen, einem Kovarianzvektor $\vec{\sigma^2}$, und einer Gewichtung w, so dass die Summe aller Gewichte 1 ergibt. Ein GMM wird normalerweise iterativ mittels des *expectation maximization*-Algorithmus (EM) trainiert, welcher mit jeder Iteration die Wahrscheinlichkeit der Übereinstimmung der Trainingsdaten mit dem Modell erhöht. Dieser Algorithmus kann als die probabilistische Variante des *k-Means*-Algorithmus interpretiert werden [20]. Ein Vorteil

von GMM's ist die relativ niedrige Anzahl benötigter Trainingsdaten, um einen Sprecher abzubilden. Abbildung 2.4 veranschaulicht die einzelnen Gauss-Verteilungen eines Sprechers, zusammengefasst als GMM.

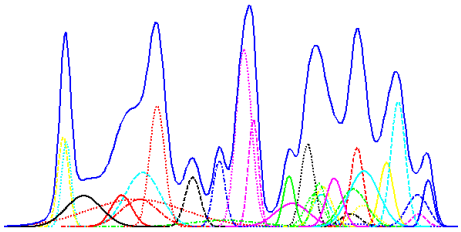

Abbildung 2.4.: Gaussian Mixture Model

2.3.3. Distanzberechnung

Um die Distanz zwischen zwei Modellen zu berechnen, wird das Distanzmass von Beigi et al.[2] eingesetzt. Dabei werden die einzelnen Distanzen zwischen GM's zweier Modelle zu einer Distanz zwischen GMM's erweitert. Drei typische Masse zur Berechnung der Distanz zwischen GM's sind die Euklidische Distanz, die Mahalanobis-Distanz und die Kullback-Leibler-Divergenz, wobei die Wahl der Distanz von der entsprechenden Problemstellung abhängt. Anhand zweier Modelle A und B mit GM's An bzw. Bn, wobei n für die n-te Verteilung steht, wird die Vorgehensweise im Folgenden veranschaulicht. Zunächst werden alle Distanzen zwischen den einzelnen Verteilungen in beide Richtungen, also z.B. von $A1$ zu $B3$ und von $B3$ zu $A1$ berechnet. Anschliessend werden die Distanzen pro Modell aufsummiert und mit der jeweiligen Anzahl vorhandener Verteilungen multipliziert. Zuletzt werden die daraus resultierenden Werte zusammengezählt und durch die Anzahl vorhandener Verteilungen der Modelle A und B dividiert. Der grösste Nachteil dieses Masses ist, dass es für kommerzielle Anwendungen aufgrund von Patentrechten nicht frei verfügbar ist[20]. Für weiterführende Informationen wird auf die Arbeiten von Beigi et al.[2] verwiesen.

2.3.4. Analyse des Baseline-Ansatzes

Das im Baseline-Ansatz verwendete Vorgehen hat sich über die Zeit als zuverlässig und robust erwiesen. Reynolds und Rose erzielten in ihren Experimenten im Bereich Identifikation von Sprechern eine Erkennungsrate 94.50% bei 16 unterschiedlichen Sprechern und jeweils 5 s Testdaten[17]. Obwohl diese Resultate in einem hohen Bereich liegen, gibt es durchaus Potential zur Steigerung der Erkennungsleistung. Die Schwächen des Ansatzes lassen sich gut anhand von Clustering-Experimenten verdeutlichen. Wie in der Arbeit von Stadelmann[21] anhand eines solchen Experiments gezeigt wurde, werden 20 Sprecher fehlerlos geclustert, jedoch liegt die Missklassifizierungsrate bei 40 Sprechern bereits bei 12.50%. Für 630 Sprecher ist die Missklassifizierungsrate mit 99.84% in einem Bereich, der unmissverständlich aufzeigt, dass die Aufgabe für die angewandte Vorgehensweise zu komplex ist.

2.4. Verwandte Ansätze

Ausgehend von der Erkenntnis, dass der Baseline-Ansatz im Falle von Clustering keine befriedigenden Resultate liefert, sobald die Anzahl der Sprecher eine kritische Grenze überschreitet (vgl. 2.3.4), werden im Folgenden Ansätze vorgestellt, um diesem Problem Abhilfe zu schaffen. Als wichtige Basis für die Auswahl von potentiellen Vorgehensweisen werden hier weitere Experimente von Stadelmann[21] heran gezogen. Diese lieferten im Vergleich zum Baseline-Ansatz deutlich bessere Ergebnisse. Die Resultate der Forschung von Stadelmann zeigen, dass im mittellangen zeitlichen Verlauf von Audiosignalen (d.h. zwischen 80 bis 120 ms, was rund zwei Phonemen bzw. einem kurzen Wort entspricht) Information enthalten ist, die entscheidend dazu beitragen kann, die Erkennungsleistung zu steigern. Daraus ist zu

schliessen, dass die Annahme der statistischen Unabhängigkeit zwischen Frames eine zu starke Vereinfachung der realen Bedingungen ist. Als weiterführende Forschung wird deshalb der Frage nachgegangen, wie mittellange zeitliche Features in die Sprechermodelle integriert werden können.
In der Forschungsgruppe wurden verschiedene Ansätze und Arbeiten diskutiert und es wurde eine Vorselektierung vorgenommen. Im Folgenden werden die fünf vielversprechendsten Ansätze vorgestellt und in Bezug auf die zuvor genannte Fragestellung betrachtet. Der erste Ansatz befasst sich mit *Delta-Features*, beschrieben in Abschnitt 2.4.1. *Delta-Features* stellen eine zeitliche Ableitung von zuvor berechneten Features, wie beispielsweise MFCC's, dar. Abschnitt 2.4.2 behandelt die Arbeit von Joder et al.[11], die sich mit der Erkennung von Instrumenten in Musikstücken mittels Integration zeitlicher Features befasst. In der Arbeit von Ke et al.[12], beschrieben in Abschnitt 2.4.3, wird die Identifikation von Musikstücken anhand kurzer Audioausschnitte untersucht. Abschnitt 2.4.4 befasst sich mit der Arbeit von Typke et al.[23], die sich mit dem Vergleich melodischer Ähnlichkeit zwischen Musikstücken beschäftigt. Der fünfte und letzte Ansatz behandelt künstliche neuronale Netze, beschrieben in Abschnitt 2.4.5.

2.4.1. Ansatz 1: Delta-Features

Delta-Features wurden im Jahr 1986 von Furui[8] eingeführt und gehören zur Kategorie der *Meta-Features*, also Features, die auf sogenannten *Primary-Features* basieren. In die Kategorie der *Primary-Features* gehören wiederum diejenigen Features, die ohne die Berechnung vorhergehender Features auskommen. Ein Beispiel dafür sind MFCC's. *Delta-Features* werden zusammen mit *Primary-Features* eingesetzt, welche die zeitliche Information des Audiosignals während der Extraktion verlieren. Das Ziel ist es, diese Information bis zu einem bestimmten Grad wiederherzustellen. *Delta-Features* können dabei als eine erste zeitliche Ableitung von *Primary-Features* verstanden werden. Zusätzlich zu dieser ersten Ableitung werden auch *Delta-Delta-Features* (auch *Double-Delta-Features* genannt), also eine zweite zeitliche Ableitung von *Primary-Features*, verwendet. *Primary-* und *Meta-Features* werden nach der Berechnung direkt miteinander verkettet. Wenn also die *Primary-Features* sowie die *Delta-* und *Delta-Delta-Features* M-dimensionale Vektoren sind, resultieren daraus Featurevektoren, deren Dimension dem Dreifachen von M entspricht. Weiterführende Informationen sind in der Arbeit von Ye[29] beschrieben.
Diese Art von Features wurde im Bereich der automatischen Sprechererkennung bereits in verschiedenen Arbeiten eingesetzt, wie beispielsweise in der Arbeit von Yu[30]. Ein Vorteil bei der Verwendung von *Delta-Features* ist die relativ niedrige Komplexität in der Anwendung und Berechnung. Zu berücksichtigen ist jedoch, dass Modelle wie GMM's bei solch hochdimensionalen Vektoren schnell an ihre Grenzen stossen[22].

2.4.2. Ansatz 2: Joder et al.

In der Arbeit *Temporal integration for audio classification with application to musical instrument classification* aus dem Jahr 2009[11] haben sich Joder, Essid und Richard mit automatischer Erkennung bzw. Klassifizierung von Instrumenten in Solo-Musikstücken (d.h. keine gleichzeitigen Einsätze von Instrumenten) beschäftigt. Das Ziel der Arbeit ist, aufzuzeigen, wie mittellange zeitliche Eigenschaften des Audiosignals zur Steigerung der Erkennungsleistung eingesetzt werden können. Um die zeitlichen Aspekte zu integrieren werden zwei Varianten, die *frühe Integration* und die *späte Integration*, vorgeschlagen.
Die *frühe Integration* erweitert die eigentliche *Feature Extraction*, indem aus den *Primary-Features*, extrahiert anhand kurzer Zeitfenster, *Meta-Features* berechnet werden. Diese beinhalten Charakteristiken des Signals auf einer höheren Zeitskala, was durch Zusammenfassen mehrerer Frames erreicht wird. Ein Vorteil dieser Herangehensweise ist, dass die Anzahl Featurevektoren stark reduziert wird, was ebenfalls zu einer Reduktion der Komplexität der Klassifizierung führt und zeitliche Eigenschaften des Audiosignals während des *Modelings* berücksichtigt werden. Joder et al. schlagen vier Möglichkeiten vor, die zeitliche Information der Features zu extrahieren.

1. **Simple Statistics**: Über mehrere Frames hinweg werden die Durchschnittswerte bzw. Kovarianzmatrizen berechnet und anschliessend als Featurevektoren eingesetzt.

2. **Autoregressive (AR) Models**: AR-Modelle nutzen vorangegangene Featurevektoren, um die Vorhersage eines auf diesen Vektoren basierenden Featurevektors zu berechnen. Der Vorteil von AR-Modellen gegenüber *Simple Statistics* ist, dass sie die zeitliche Abhängigkeit der vorangegangenen Features berücksichtigen.

3. **Spectral Features**: Zeitliche Information wird aus den spektralen Charakteristiken eines Signals extrahiert. Dafür werden drei unterschiedliche Methoden eingesetzt, die verschiedene spektrale Eigenschaften der Features berücksichtigen.

4. **Feature Stacking**: Kurzzeitige Featurevektoren werden zusammengefügt und bilden so neue Featurevektoren, die den zeitlichen Verlauf des Signals implizit in sich tragen. Diese Vorgehensweise erhöht die Dimension der Featurevektoren enorm und somit auch die Komplexität der Klassifizierung.

Die *späte Integration* befasst sich nicht direkt mit dem Extrahieren von Features, sondern setzt bei der Klassifizierung an. Dies geschieht entweder durch Zusammenfassen von vorangegangenen Entscheidungen während der Klassifizierung oder durch Verwendung eines Klassifizierungsmodells, das Entscheidungssequenzen verarbeiten kann. Was das im Detail bedeutet, wird anhand der drei folgenden, von Joder et al. vorgeschlagenen, Vorgehensweisen ausgeführt.

1. **Fusion of Decisions**: Für eine Sequenz von Features werden pro Feature die Zugehörigkeitswahrscheinlichkeiten zu gegebenen Klassen berechnet und diese Wahrscheinlichkeiten pro Klasse aufsummiert. Anschliessend wird die Featuresequenz der Klasse mit der höchsten Wahrscheinlichkeit zugeordnet.

2. **Hidden Markov Model (HMM) Classifier**: Ein HMM besitzt die Eigenschaft, statistische Abhängigkeiten von Features festzuhalten. Dies erlaubt die Berechnung der Zugehörigkeitswahrscheinlichkeit einer gegebenen Featuresequenz zu einem Modell. Daher wird für jede Klasse jeweils ein HMM trainiert und die Zugehörigkeit der Sequenzen berechnet. Danach wird mittels der Wahrscheinlichkeiten die Klassifizierung durchgeführt. Für eine detaillierte Beschreibung von HMM's wird auf die Arbeit von Rabiner [15] verwiesen.

3. **Alignment Kernels**: Im Zusammenhang mit *Support Vector Machines* (SVM), statistischen Klassifikatoren die das Berechnen optimaler Hyperebenen erlauben, werden Kernels eingesetzt, um Gruppen von Objekten (d.h. in diesem Fall Features) zu trennen bzw. zu klassifizieren, die nicht linear trennbar sind. Ein *Alignment Kernel* ist eine spezielle Variante von *Sequenz Kernels*, Kernels also, die Sequenzen von Features vergleichen können anstatt nur jeweils einzelne Features. Für weiterführende Informationen zu SVM's wird auf die Arbeit von Joder et al. verwiesen [11].

Die von Joder et al. durchgeführten Experimente zeigten, dass eine Steigerung der Erkennungsleistung nur dann eintritt, wenn die *frühe Integration* in Kombination mit der *späten Integration* eingesetzt wird. Im Vergleich zum gewählten Referenzsystem konnte die Erkennungsleistung im besten Fall von 81.6% auf 84.5% erhöht werden. Diese Ergebnisse wurden erzielt unter Verwendung von *Simple Statistics* während der *frühen Integration* sowie eine SVM mit einem *Alignment Kernel* während der *späten Integration*.

2.4.3. Ansatz 3: Ke et al.

Die Arbeit *Computer vision for music identification* von Ke, Hoiem und Rahul aus dem Jahr 2005 [12] befasst sich mit *Audio-Fingerprinting*, oder genauer, mit der automatischen Identifikation von Musikstücken anhand weniger Sekunden qualitativ schlechter Audiodaten. Das Ziel der Arbeit ist es, aufzuzeigen, wie die gegebene Problemstellung mittels Techniken der *Computer Vision* erfolgreich gelöst werden kann. Dabei werden Audiosignale als Spektrogramme abgebildet (vgl. Abblidung 2.5) und aus diesen mittels einem trainierten Set von Filtern und zugehörigen Grenzwerten Deskriptoren (vergleichbar mit Features) extrahiert. Diese Deskriptoren werden wiederum zu Signaturen (vergleichbar mit Modellen) zusammengefügt, die das komplette Audiosignal beschreiben. Anhand dieser Signaturen, werden im Anschluss Abgleiche von Audioausschnitten mit einer Musikdatenbank durchgeführt. Die Idee hinter dieser

Vorgehensweise ist, dass bei der Abbildung von Audiosignalen als Spektrogramme Charakteristiken extrahiert werden können, die, unabhängig von der Qualität des Signals, eindeutig und robust sind. Um dies zu erreichen, benötigt es grundsätzlich vier Komponenten, die im Folgenden erläutert werden.

1. **Spektrogramme**: Die verwendeten Spektrogramme bilden die in Audiosignalen enthaltene Energie als zweidimensionales Bild ab, wobei die zwei Dimensionen für Zeit und Frequenz stehen. Sie dienen hier als Grundlage für die Gewinnung der Features zur Identifikation von Musikstücken.

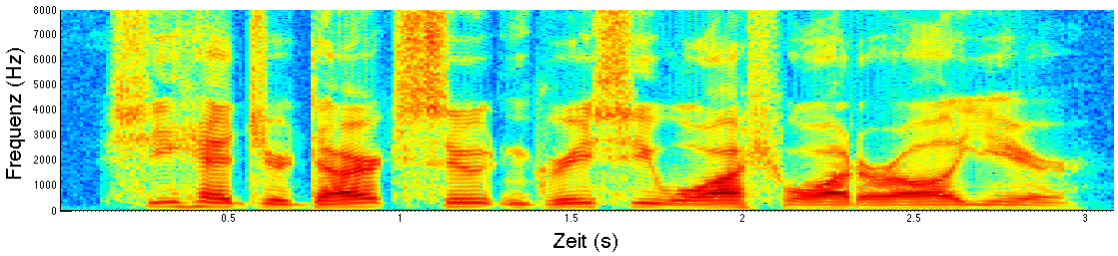

Abbildung 2.5.: Spektrogramm anhand von rund 3 s Audiodaten des Sprechers MJSW0

2. **Classifier**: Das direkte Vergleichen von Spektrogrammen, um Übereinstimmungen zu finden, führt zu Ungenauigkeiten und zeitintensiven Berechnungen. Aus diesem Grund werden aus den Spektrogrammen mittels einem kleinen Set von Filtern die entscheidenden Merkmale extrahiert. Bevor diese Merkmale jedoch extrahiert werden können, ist eine geeignete Auswahl von Filtern, wie sie in der Bildverarbeitung üblich sind, zu bestimmen. Für diese Aufgabe wird auf ein Verfahren aus der Gesichtserkennung, einem Teilbereich der *Computer Vision*, zurückgegriffen. Dieses Verfahren wurde vorgestellt von Viola und Jones im Jahr 2001 [26] bzw. 2004 [27]. Die Filter werden aus einem zuvor definierten Set von potentiellen Filtern, im Weiteren als Filterkandidatenset bezeichnet, selektiert. Anhand einer erweiterten Variante des *AdaBoost*-Algorithmus, dem *Pairwise Boost*, wird eine festgelegte Anzahl von Filtern und zugehörigen Grenzwerten selektiert. Der *AdaBoost*-Algorithmus wurde im Jahr 1995 von Freud und Schapire [7] eingeführt. Mittels Boosting-Algorithmen wie *AdaBoost* und *Pairwise Boost* wird ein *strong Classifier* trainiert, der sich aus *weak Classifiers* zusammensetzt. Ein *weak Classifier* besteht aus einem Filter und einem Grenzwert. Anhand des Filters und des Grenzwerts wird entschieden, welcher Kategorie ein Objekt zugehört. *Weak Classifiers* können jedoch ein Objekt meist nur wenig besser als zufällig klassifizieren. Werden diese *weak Classifiers* jedoch kombiniert, so entsteht dadurch der *strong Classifier*, der in der Lage ist, Objekte zuverlässig zu klassifizieren.
   Die Wahl der *weak Classifiers* geschieht iterativ. In jeder Iteration wird aus gegebenen Daten und zugehörigen Kategorien derjenige *weak Classifier* ausgewählt, welcher die unterschiedlichen Kategorien am besten trennt. Vertiefende Informationen zu AdaBoost sind in der Arbeit von Schapire [19] zu finden. Für weiterführende Informationen zu *Pairwise Boost* wird auf 3.2 sowie die Arbeit von Ke et al. [12] verwiesen.

3. **Signatur**: Die für die Erstellung der Signaturen verwendeten Deskriptoren sind M-Bit-Vektoren und werden anhand der erlernten *weak Classifiers* definiert, wobei M der Anzahl der *weak Classifiers* entspricht. Die Bitwerte in den Deskriptoren sind dabei abhängig vom entsprechenden Filterwert und ob dieser über- oder unterhalb des zugehörigen Grenzwerts liegt.

4. **Entscheidungsmodell**: Für die Vergleiche von Audiodaten wird die Hamming-Distanz zwischen Signaturen bzw. Deskriptoren berechnet. Für eine gegebene Signatur wird in einer Datenbank nach derjenigen Signatur gesucht, die mit grösster Wahrscheinlichkeit zur erstgenannten passt und anschliessend überprüft, ob diese Signaturen vom selben Original stammen. Da die zur Erkennung verwendeten Audiosignale jedoch oft Störgeräusche enthalten oder Abschnitte des Signals komplett von solchen überlagert sind, werden zudem solche Abschnitte selektiert und von der Identifikation ausgeschlossen.

Aus den durchgeführten Experimenten ist eine deutliche Verbesserung der Resultate mittels dem vorgeschlagenen Ansatz gegenüber den zum Vergleich herangezogenen Ansätzen erkennbar. Detaillierte Informationen zu den Ergebnissen sind in der Arbeit von Ke et al. [12] zu finden.

2.4.4. Ansatz 4: Typke et al.

Typke, Giannopoulos, Veltkamp, Wiering und van Oostrum befassten sich in ihrer Arbeit *Using transportation distances for measuring melodic similarity* aus den Jahr 2003 [23] mit Methoden, automatisch die Ähnlichkeiten von Melodien zu berechnen. Die Arbeit beschreibt, wie diese Berechnung anhand von *Incipits* (d.h. in diesem Zusammenhang dem Anfang von Notentexten, der Melodien eindeutig identifiziert) unter Verwendung der *Earth Mover's Distance* (EMD) bzw. *Proportional Transportation Distance* (PTD) durchgeführt werden kann. Die Durchführung der Experimente von Typke et al. benötigt grundsätzlich vier Schritte, die im Folgenden genauer beschrieben werden.

1. **Erstellung der Signatur**: Um die Distanz zwischen Melodien zu berechnen, werden die *Incipits* zu Signaturen transformiert. Eine Signatur bildet ein *Incipit* im zweidimensionalen Raum ab, wobei die beiden Dimensionen Zeit und Tonhöhe repräsentieren. Die im *Incipit* enthaltenen Noten werden in diesem Raum als Punkte dargestellt, deren Koordinaten den Beginn des Tons und die Tonhöhe wiedergeben. Die Dauer eines Tons wird mittels den Punkten zugeordneten Gewichten abgebildet.

2. **Anpassungen in der Zeit**: In der Zeitdimension werden zwei Anpassungen vorgenommen, um die Erkennungsleistung von Melodien zu steigern. Zum einen wird die Zeitachse mit dem Faktor 3 multipliziert, um sie der Tonhöhe anzugleichen. Dies führt dazu, dass bei der Ähnlichkeitsberechnung Verschiebungen in der Tonhöhe gleich gewichtet werden wie Verschiebungen in der Zeit. Zum anderen werden Signaturen unterschiedlicher Dauer angeglichen. Das wird erreicht, indem die kürzere Signatur auf die Zeitdauer der längeren gestreckt wird.

3. **Anpassungen in der Tonhöhe**: Die Tonhöhe von Melodien wird für den Vergleich angepasst, um unterschiedliche Tonarten von sonst gleichen Melodien auszugleichen. Das geschieht, indem die Tonhöhen von jeweils zwei zu vergleichenden Melodien so verschoben werden, dass die durchschnittlichen Werte der Tonhöhen übereinstimmen.

4. **Berechnung der Ähnlichkeit**: Nach dem Anpassen der Signaturen in Zeit und Tonhöhe wird mittels der EMD bzw. PTD die Ähnlichkeit zwischen diesen berechnet. Das folgende Beispiel dient der Veranschaulichung der EMD. Angenommen auf einer Fläche befinden sich mehrere Erdhaufen und verschiedene Vertiefungen. Werden nun diese Erdhaufen in die Vertiefungen verschoben, so beschreibt die EMD die minimalen Kosten, dies durchzuführen. Die Kosten berechnen sich dabei aus der Menge der verschobenen Erde multipliziert mit der dafür zurückgelegten Distanz. Für weiterführende Informationen betreffend EMD wird auf die Arbeit von Cohen [4] verwiesen. Auf der EMD basierend wurde von Giannopoulos und Veltkamp im Jahr 2002 [9] die PTD eingeführt. Der grundlegende Unterschied gegenüber der EMD ist, dass die PTD ein Überschuss an Erde bzw. Gewicht in der Distanz berücksichtigt und somit eine präzisere Aussage über die Ähnlichkeit zweier Signaturen macht.

Die Ergebnisse der Arbeit von Typke et al. zeigen deutliche Verbesserungen gegenüber zweier Experimente, die zum Vergleich verwendet wurden. Im ersten Experiment wurden anonyme Werke aus den Testdaten anhand der *Incipits* einem Komponisten zugeordnet. Dabei konnten 3.90% aller anonymen Werke einem Komponisten zugeordnet werden, wobei das Referenzsystem lediglich 2.08% erzielte. Obwohl nur ein sehr kleiner Prozentsatz der Daten zugeordnet werden konnte, zeigen die Ergebnisse dennoch eine deutliche Steigerung. Ebenfalls deutlich fielen die Ergebnisse beim zweiten Experiment aus. Hier wurden für eine gegebene Melodie alle Versionen dieser Melodie in einer Datenbank gesucht. Dabei erkannte das System von Typke et al. 73.33% aller Versionen der gegebenen Melodie, wohingegen das Referenzsystem im besten Fall eine Ausbeute von 46.15% erreichte.

2.4.5. Ansatz 5: Künstliche neuronale Netze

Neuronale Netze übertreffen herkömmliche Methoden in vielen Bereichen, wie beispielsweise der automatischen Objekterkennung. Dies wurde durch Fortschritte in jüngster Zeit im Bereich grosser mehrschichtiger Netzarchitekturen erreicht (vgl. Bengio [3]), obwohl die herkömmlichen Systeme bisher genauer und intensiver studiert wurden als neuronale Netze. Dies macht neuronale Netze zu sehr interessanten Kandidaten in der automatischen Sprechererkennung. Um zu verstehen, wie neuronale Netze

funktionieren, wird im Folgenden ein Überblick über die Grundlagen gegeben, wobei die Arbeit von Rey und Wender von 2010 [16] als Referenz dient. Anschliessend wird auf die Sprechererkennung Bezug genommen.
Die Basis von neuronalen Netzen bilden die sogenannten Neuronen, im Folgenden auch als Units bezeichnet. Neuronen übernehmen die Aufnahme, Modifikation und Übergabe von Informationen. Dabei wird zwischen drei Arten unterschieden, Input-, Output- und Hidden-Units. Die Input-Units sind dafür zuständig, von aussen (d.h. ausserhalb des Systems) Signale bzw. Daten entgegenzunehmen. Die Output-Units wiederum übergeben Signale nach aussen. Zwischen diesen beiden Arten Units befinden sich je nach System 0 bis beliebig viele Hidden-Units, welche die Daten von den Input-Units erhalten, modifizieren und an die Output-Units weiterleiten. Unabhängig von ihrer Funktion, generieren alle Units Input sowie Output. Die Units sind durch gewichtete Kanten miteinander verbunden, wobei die Gewichte aussagen, wie stark der Einfluss von verbundenen Units aufeinander ist. Der Lernprozess eines neuronalen Netzes wird durch die Veränderung von Gewichten ausgedrückt. Die Gewichte definieren also das Wissen, das in einem solchen Netz gespeichert ist.
Um neuronale Netze zu trainieren werden unterschiedliche Lernregeln verwendet. Zwei interessante Varianten von Lernregeln, die *Backpropagation* und das *Competitive Learning*, werden im Folgenden genauer betrachtet.

1. **Backpropagation**: Bei dieser Vorgehensweise werden als erstes den Input-Units Daten übergeben und anhand der bei den Output-Units gemessenen Werte zusammen mit den erwarteten Werten die Fehler bestimmt. Falls die Fehler einen zuvor bestimmten Grenzwert nicht überschreiten, ist die Trainingsphase beendet. Andernfalls werden die Gewichte von den fehlerhaften Output-Units rückwärts über die Hidden-Units bis zu den zugehörigen Input-Units angepasst.
2. **Competitive Learning**: Dieses Verfahren benötigt im Gegensatz zur *Backpropagation* keine Erwartungswerte für den Vergleich der Ergebnisse. Der erste Schritt ist, bei allen Output-Units den betreffenden Input aus dem Netz zu messen. Danach wird diejenige Output-Unit mit dem höchsten Input ermittelt und einzig diejenigen Gewichte zwischen dieser Unit und der mit dieser Unit in Verbindung stehenden Units (direkt sowie indirekt) angepasst.

Im Bereich der Sprach- und Sprechererkennung sind bereits viele Arbeiten im Zusammenhang mit neuronalen Netzen veröffentlicht worden, wie beispielsweise die Arbeiten von Guruprasad et al. [10], Balaska et al. [1] und auch Koutník et al. [13]. Sie zeigen, wie Problemstellungen der Sprach- und Sprechererkennung erfolgreich mit solchen Netzen gelöst werden können und bieten somit sehr vielversprechende Ansätze für eine Anwendung auf diese Arbeit. Der grosse Vorteil von neuronalen Netzen besteht in dem Versprechen, das auch belegbar eingehalten wird, beliebige (d.h. auch zeitliche) nicht-lineare Abhängigkeiten in den Daten erlernen zu können. Die aktuellen Erfolge (vgl. Koutník et al. [13]) beruhen jedoch auf dem Einsatz enormer Hardware-Ressourcen zum Trainieren tiefer Netze sowie noch nicht sehr weit verbreitetem Spezialwissen bezüglich praktischem Umgang mit solchen Architekturen.

3. Vorgehen

3.1. Beurteilung der vorgeschlagenen Ansätze

Die unter 2.4 beschriebenen Ansätze werden in diesem Abschnitt betreffend Anwendung auf die vorliegende Problemstellung beurteilt. Diese Beurteilung wird anhand der folgenden vier Kriterien vorgenommen:

1. **Erfolg**: Die Beurteilung des Erfolgs eines Ansatzes bezieht sich auf das Forschungsgebiet, in dem dieser vorwiegend eingesetzt wird bzw. auf das sich die konkrete Arbeit bezieht.
2. **Aussagekraft**: Anhand der Aussagekraft wird der Erfolg eines Ansatzes relativiert, um verfälschende Effekte, beispielsweise ausgelöst durch unterschiedliche Testdaten beim Vergleichen von Experimenten, auszugleichen.
3. **Potential**: Durch dieses Kriterium wird beurteilt, wie hoch die Erfolgsaussichten des jeweiligen Ansatzes in Anwendung auf die vorliegende Problemstellung sind.
4. **Machbarkeit**: Ein weiteres wichtiges Kriterium ist die Machbarkeit, das einerseits berücksichtigt, ob der Ansatz geeignet umgesetzt werden kann, und andererseits den dafür benötigten Aufwand abschätzt.

Da der Erfolg und die Aussagekraft voneinander abhängig sind, wird eine Gewichtung der Kriterien vorgenommen. Erfolg und Aussagekraft werden dabei einfach gewichtet und Potential sowie Machbarkeit zweifach, was dazu führt, dass die beiden erstgenannten ihren Einfluss teilen. Die Bewertungsskala unterscheidet fünf verschiedene Werte, dargestellt durch $--$, $-$, 0, $+$ und $++$. Tabelle 3.1 zeigt zusammengefasst die Bewertungen der Ansätze. Übereinstimmend mit der Reihenfolge aus 2.4, steht Ansatz 1 für *Delta-Features*, Ansatz 2 für die Arbeit von Joder et al., Ansatz 3 für die Arbeit von Ke et al., Ansatz 4 für die Arbeit von Typke et al. und Ansatz 5 für künstliche neuronale Netze.

	Ansatz 1	Ansatz 2	Ansatz 3	Ansatz 4	Ansatz 5
Erfolg	$+$	$++$	$++$	$++$	$++$
Aussagekraft	0	$++$	$++$	$+$	$++$
Potential	$+$	$+$	$++$	$+$	$++$
Machbarkeit	$++$	$-$	$+$	0	$--$
Gesamtbeurteilung	**+**	**+**	**++**	**+**	**+**

Tabelle 3.1.: Beurteilung der vorgeschlagenen Ansätze

Aus den Beurteilungen geht hervor, dass, obwohl für alle vorgeschlagenen Ansätze Potential bezüglich der vorliegenden Problemstellung besteht, die Machbarkeit stark variiert. Dies ist einerseits durch die Komplexität des jeweiligen Ansatzes und andererseits durch benötigtes Grundlagenwissen, was beispielsweise die Wahl entsprechender Parameter angeht, begründet. Erfolg und Aussagekraft sind überwiegend positiv bewertet. Einzig bei den *Delta-Features* wird die Aussagekraft neutral eingestuft, hauptsächlich aus dem Grund, dass der Erfolg stark von weiteren Komponenten des jeweiligen Systems abhängt, wie z.B. dem *Modeling*, und daher nicht unabhängig betrachtet werden kann.
Für eine Anwendung auf die vorliegende Problemstellung wird die Arbeit von Ke et al. ausgewählt. Sie erzielte in ihrem Forschungsgebiet deutliche Verbesserungen, die anhand herkömmlicher Vorgehensweisen verifiziert wurden. Ihr wird hohes Potential zugeschrieben, da sie zeigt, wie zeitliche Aspekte erfolgreich extrahiert werden können. Die Komplexität des Ansatzes ist dabei vergleichbar mit derjenigen des Baseline-Ansatzes, wobei die benötigten Grundlagen in weiten Teilen die gleichen sind.

3.2. Detaillierte Beschreibung der Arbeit von Ke et al.

In diesem Abschnitt wird die Umsetzung der Arbeit von Ke et al. [12] im Detail betrachtet. Dies wird anhand der unter 2.4.3 beschriebenen und für das System benötigten Komponenten gegliedert.

3.2.1. Spektrogramme

Für die Erstellung der Spektrogramme werden jeweils 82 Frames mit 11.6 ms Länge verwendet. Mit Hilfe der *Short-Term Fourier Transformation* (STFT) werden die Frames transformiert und zu Spektrogrammen zusammengefügt. Die Spektrogramme repräsentieren die in 33 logarithmisch verteilten Frequenzbändern enthaltene Energie, gemessen über Zeitfenster von 0.372 s, die jeweils um 11.6 ms inkrementiert werden.

3.2.2. Classifier

Für die Wahl der Filter wird ein Filterkandidatenset definiert, das aus Variationen von fünf Basisfiltern bzw. Basisklassen (vgl. Abbildung 3.1) besteht. Diese Basisfilter wurden von Viola und Jones zur automatischen Erkennung von Objekten [26] entworfen.

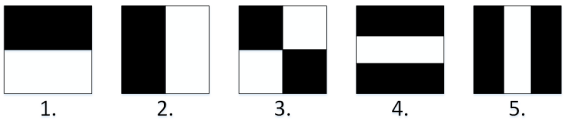

Abbildung 3.1.: Basisfilter aus dem Bereich der automatischen Objekterkennung

Der Wert eines Filters berechnet sich durch die Differenz zwischen den weissen und schwarzen Flächen, welche die aufsummierten Pixel innerhalb der entsprechenden Fläche repräsentieren. Um für ein gegebenes Bild den Wert des Filters an einer bestimmten Position zu berechnen, werden daher zunächst die Summen der Pixel innerhalb der Rechtecke benötigt. Diese Summen lassen sich effizient mittels integraler Bilder berechnen. Für eine detaillierte Beschreibung zur Berechnung der Filterwerte sowie zu integralen Bildern wird auf die Arbeit von Viola und Jones verwiesen [26]. Die Filterwerte beschreiben dabei, in übereinstimmender Reihenfolge der Filter aus Abbildung 3.1, eine der folgenden Informationen:

1. Differenzen der Energie in benachbarten Frequenzbändern in einem bestimmten Zeitraum
2. Differenzen der Energie über die Zeit innerhalb bestimmter Frequenzbänder
3. Verschiebungen der dominanten Frequenz über die Zeit
4. Energiespitzen über Frequenzbänder hinweg in einem bestimmten Zeitraum
5. Energiespitzen über die Zeit innerhalb bestimmter Frequenzbänder

Im verwendeten System kann jeder Basisfilter in exponentiellen Schritten in Frequenzbandposition von 1 bis 33, in Frequenzbandbreite von 1 bis 33 und in der Zeit (d.h. Frames) von 1 bis 82 variieren. Dies führt zu einem Set von rund $25'000$ Filterkandidaten. Aus diesem Set werden mittels *Pairwise Boost* die 32 aussagekräftigsten *weak Classifierts* berechnet. *Pairwise Boost* unterscheidet sich von *AdaBoost* hauptsächlich darin, dass bei der Berechnung der Filter anstelle einzelner Audiosignale Audiosignalpaare verarbeitet werden. Hierfür wurde die Gewichtungsstrategie des Algorithmus, mit der die Reihenfolge der *weak Classifiers* festgelegt wird, angepasst. Eine genaue Beschreibung des Algorithmus findet sich in der Arbeit von Ke et al. [12].

3.2.3. Signatur

Anhand der 32 erlernten *Classifiers* werden aus allen Musikdateien der verwendeten Datenbank die Deskriptoren bzw. Signaturen extrahiert. Pro Spektrogramm wird also ein Deskriptor mit 32 Binärwerten berechnet. Für 10 s Audiodaten führt dies zu einer Signatur mit rund 860 Deskriptoren. Die Signaturen werden anschliessend in Hashtabellen hinterlegt und in der Datenabk gespeichert, was sehr effizientes Zugreifen auf diese Daten ermöglicht.

3.2.4. Entscheidungsmodell

Für Vergleiche einer Query-Signatur mit der Datenbank werden zunächst die einzelnen Deskriptoren verglichen. Dabei werden alle in der Datenbank vorhandenen Deskriptoren abgefragt, die eine Hamming-Distanz von maximal 2 gegenüber einem der Deskriptoren der Query-Signatur aufweisen. Um anschliessend die am besten passende Signatur zu bestimmen, wird für jede Signatur, die mindestens einen der zuvor selektierten Deskriptoren beinhaltet, die Wahrscheinlichkeit der Übereinstimmung berechnet. Diese Berechnung erfolgt mittels dem *Random Sample Consensus*-Algorithmus (RANSAC), eingeführt von Fischler und Bolles im Jahr 1981 [6], und der EM. Gleichzeitig werden die Deskriptoren, die von Störgeräuschen überlagert sind, bestimmt und von der Wahrscheinlichkeitsberechnung ausgeschlossen. Dazu wird jeweils die Differenz zwischen Datenbank- und Query-Deskriptor gebildet und unter Zuhilfenahme von unabhängigen Bernoulli-Zufallsvariablen deren Verteilung modelliert. Anhand dieser Verteilung wird schliesslich mit der EM und einem entsprechenden Grenzwert entschieden, ob der Deskriptor von Störgeräuschen überlagert ist oder nicht. Für eine vertiefende Ausführung wird auf die Arbeit von Ke et al. [12] verwiesen.

3.3. Konzept

Im Folgenden wird das Konzept beschrieben, das sich aus der Arbeit von Ke et al. [12] ableitet. Es basiert grundsätzlich auf dem Basline-Ansatz, der entsprechend angepasst und erweitert ist. Der Ablauf des Konzepts ist aufgeteilt in eine Trainings- und eine Testphase.
In Abschnitt 3.3.1 werden die aus der Arbeit von Ke et al. übernommenen Verfahren anhand der grundlegenden Unterschiede beschrieben. Anschliessend wird in den Abschnitten 3.3.2 und 3.3.3 die Trainings- bzw. Testphase genauer erklärt. Da die Schritte *Preprocessing* und *Recognition* mit dem Baseline-Ansatz identisch sind, werden sie hier nicht weiter ausgeführt. Abbildung 3.2 zeigt den Ablauf des vorgeschlagenen Konzepts.

3.3.1. Konzeptionelle Unterschiede

In der vorliegenden Arbeit werden für die Durchführung der Experimente ausschliesslich Daten verwendet, die keinerlei Störgeräusche enthalten. Aus diesem Grund wird das von Ke et al. vorgeschlagene Entscheidungsmodell im Konzept nicht weiter berücksichtigt. Die übernommenen Verfahren lassen sich daher vollständig der *Feature Extraction* sowie der Filterselektion, einem zusätzlichen Schritt während der Trainingsphase, zuordnen.
Zur Erstellung von Spektrogrammen werden, anstelle mittels STFT transformierten Frames, FBE's eingesetzt. Dabei wird davon ausgegangen, dass in den FBE's ebenfalls die gesamte notwendige Information enthalten ist. Der daraus resultierende Vorteil besteht darin, dass die *Feature Extraction* des Baseline-Ansatzes weitgehend übernommen werden kann. Weil die von Ke et al. eingesetzten *weak Classifiers* binäre Werte generieren, die für das Erstellen von Modellen ungeeignet sind, werden während der Filterselektion nur die zugehörigen Filter bestimmt. Dadurch werden die entscheidenden Merkmale unabhängig von Grenzwerten aus den Spektrogrammen extrahiert und die Deskriptoren somit durch die konkreten Filterwerte ersetzt.

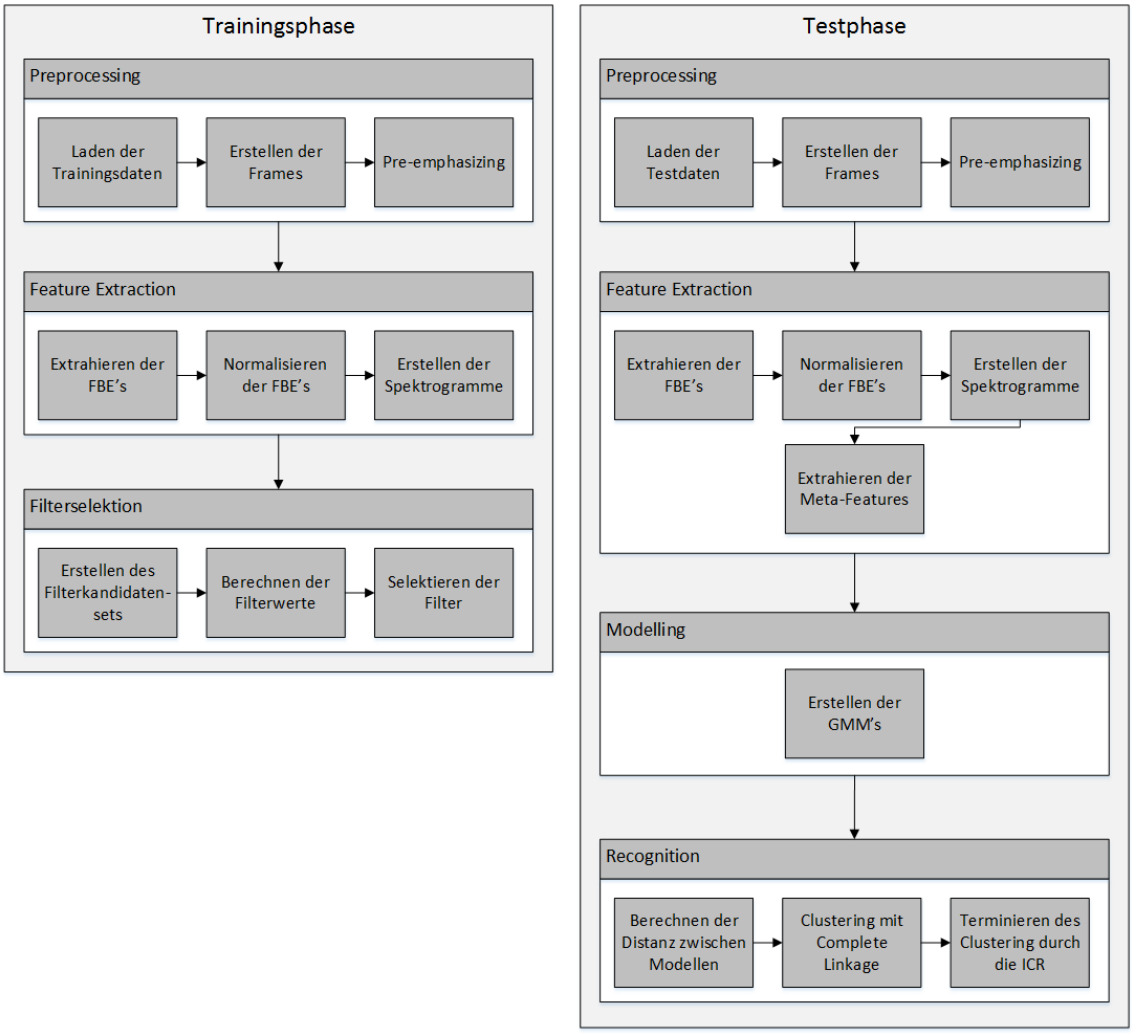

Abbildung 3.2.: Ablauf des erarbeiteten Konzepts

3.3.2. Trainingsphase

Die Beschreibung der Trainingsphase gliedert sich in die einzelnen Schritte, die notwendig sind, um eine geeignete Auswahl von Filtern zu treffen. Dabei wird bei der *Feature Extraction* bzw. beim Extrahieren der FBE's begonnen (vgl. Abbildung 3.2).

1. **Extrahieren der FBE's**: Gleich dem Extrahieren von MFCC's (vgl. 2.3.1) werden die FBE's berechnet, mit dem Unterschied, dass der Vorgang direkt vor der DCT beendet wird.

2. **Normalisieren der FBE's**: Ziel dieser Normalisierung ist es, verfälschende Gewichtungen von Phonemen unterschiedlicher Länge auszugleichen und wurde übernommen aus der Arbeit von Stadelmann [21]. Dazu werden die FBE's zu Clustern zusammengefasst und pro Cluster durch den jeweiligen Zentroid ersetzt. Anschliessend werden Sequenzen gleicher FBE's auf eine einzelne FBE reduziert.

3. **Erstellen der Spektrogramme**: Anhand der normalisierten FBE's werden die Spektrogramme pro Sprecher erstellt. Diese dienen als Grundlage für die Selektion der Filter.

4. **Erstellen des Filterkandidatensets**: Das Filterkandidatenset enthält Variationen der unter 3.2 beschriebenen Basisfilter (vgl. Abbildung 3.1). Jeder Basisfilter kann, ähnlich zur Arbeit von Ke et al. [12], in Frequenzbandposition, in Frequenzbandbreite und in der Zeit (d.h. Frames) variieren.

5. **Berechnen der Filterwerte**: Mit Hilfe von integralen Bildern werden pro Spektrogramm die Werte aller vorhandenen Filter berechnet. Die berechneten Werte werden zudem mit jeweils einem Label gekennzeichnet, das den zugehörigen Sprecher identifiziert.
6. **Selektieren der Filter**: Für das Selektieren der Filter wird eine Variante von *AdaBoost* eingesetzt, die in der Lage ist, Multilabels (d.h. drei oder mehr unterschiedliche Labels) zu verarbeiten. Anhand der berechneten Filterwerte werden so die aussagekräftigsten Filter gewählt. Für diesen Schritt wurde *AdaBoost* gewählt, weil *Pairwise Boost* auf Daten ausgelegt ist, die Störgeräusche enthalten, was bei den Daten, die in dieser Arbeit verwendet werden, nicht der Fall ist.

3.3.3. Testphase

Die Beschreibung der Testphase bezieht sich auf das Extrahieren der *Meta-Features* sowie das Erstellen der GMM's (vgl. Abbildung 3.2), da sich das Vorgehen bis und mit dem Erstellen der Spektrogramme im Vergleich zur Tainingsphase einzig durch unterschiedliche Audiodaten differenziert.

1. **Extrahieren der *Meta-Features***: Anhand der selektierten Filter werden für die zuvor erstellten Spektrogramme alle Filterwerte berechnet. Gleich wie in der Trainingsphase wird dies mit Hilfe von integralen Bildern durchgeführt. Die daraus resultierenden Werte repräsentieren dabei die *Meta-Features*.
2. **Erstellen der GMM's**: Bei der Erstellung der Modelle werden zwei verschiedene Arten von Features eingesetzt. Zum einen werden GMM's trainiert, die ausschliesslich auf *Meta-Features* basieren. Aufgrund der Annahme, dass *Meta-Features* in Kombination mit MFCC's die Erkennungsleistung weiter steigern, werden GMM's ebenfalls anhand von kombinierten Features berechnet.

4. Implementierung

4.1. Software

Im Folgenden werden die einzelnen Softwarekomponenten bezüglich Implementierung der Experimente beschrieben. Weiter wird die Implementierung selbst anhand der wichtigsten Funktionen aufgezeigt. Abschnitt 4.1.1 gibt einen Überblick über die Herkunft des Programmcodes und beschreibt die verwendete Entwicklungsumgebung sowie die Programmiersprache. In den Abschnitten 4.1.2 und 4.1.3 wird auf den Ablauf der Implementierung betreffend Filterselektion bzw. Clustering eingegangen.

4.1.1. Softwarekomponenten

Das in Kapitel 3 beschriebene Konzept sowie der Baseline-Ansatz wurde mit MATLAB umgesetzt. MATLAB ist einerseits eine von MathWorks vertriebene Entwicklungsumgebung und gleichzeitig eine Programmiersprache der vierten Generation, die darauf ausgelegt ist, numerische Berechnungen mithilfe von Matrizen durchzuführen. In dieser Arbeit wurde MATLAB Version 8.3.0.532 (R2014a) verwendet. Der Programmcode bezüglich Implementierung der Experimente kann in drei Gruppen unterteilt werden. Gruppe 1 beschreibt den von MathWorks bereitgestellten Code, der bereits in die Entwicklungsumgebung integriert ist. Zur Gruppe 2 gehört der im Rahmen dieser Arbeit umgesetzte Programmcode, im Weiteren als *ba14_lib* bezeichnet. Gruppe 3 beinhaltet die Bibliotheken von Drittentwicklern, wobei in der vorliegenden Arbeit lediglich eine solche Bibliothek eingesetzt wird.

1. **Gruppe 1**: Nebst den Basisfunktionen, die hauptsächlich für Arrayoperationen sowie Ein- und Ausgabe von Daten verwendet werden, beinhaltet Gruppe 1 sogenannte *Toolboxen* (d.h. Erweiterungen der Entwicklungsumgebung). Zwei dieser *Toolboxen* werden in dieser Arbeit eingesetzt. Zum einen ist dies die *Statistics Toolbox*, Version 9.0 (R2014a), zum anderen die *Parallel Computing Toolbox*, Version 6.4 (R2014a). Aus der *Statistics Toolbox* werden die Algorithmen *k-Means* und *AdaBoost* sowie die Funktionen zur Berechnung der GMM's und der *Complete Linkage* (vgl. 5.1.3) eingesetzt. Die *Parallel Computing Toolbox* wird verwendet, um Berechnungen auf die vorhandenen Prozessorkerne zu verteilen und die Experimente dadurch um ein Vielfaches zu beschleunigen.
2. **Guppe 2**: Die *ba14_lib* enthält 28 Funktionen mit insgesamt knapp 2'000 Zeilen Code. Zudem ist jeweils ein Skript zum Starten des Clusterings sowie der Filterselektion vorhanden. Pro Skript existiert eine Konfigurationsdatei, hauptsächlich zur Festlegung der zugrundeliegenden Daten und der Art der Experimente im Falle des Clusterings. Weiter ist eine gemeinsame Konfigurationsdatei vorhanden, anhand derer die Parameter (vgl. Kapitel 5) definiert werden.
3. **Gruppe 3**: Um die Extraktion der FBE's und MFCC's durchzuführen, wird auf die Bibliothek *HTK MFCC MATLAB* von Wojcicki zurückgegriffen. Diese Bibliothek steht unter der *Berkeley Software Distribution* (BSD) Lizenz, die zusammen mit detaillierten Angaben zur Bibliothek im Anhang unter A.2.2 zu finden ist.

4.1.2. Ablauf Filterselektion

Anhand des in Abblidung 4.1 dargestellten Ablaufs werden in diesem Abschnitt die wichtigsten Funktionen bezogen auf die Selektion der Filter beschrieben. Die Dateiendung *.m* definiert dabei eine MATLAB-Datei und wird bei der Beschreibung weggelassen.

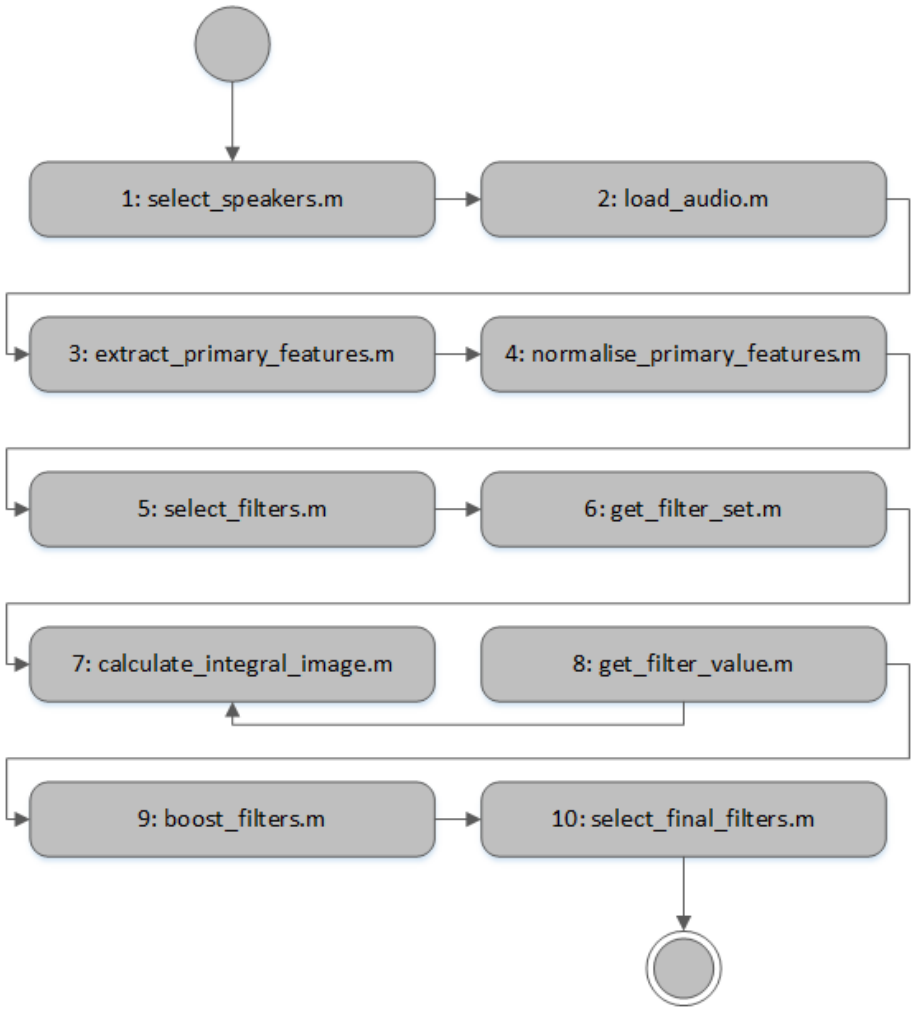

Abbildung 4.1.: Ablauf Filterselektion

1. **select_speakers**: Lädt die Verzeichnisse aller in der Konfigurationsdatei definierten Sprecher aus einer Textdatei.
2. **load_audio**: Lädt die **.wav*-Dateien, die sich in den zuvor geladenen Verzeichnissen befinden und erstellt die normalisierten Frames.
3. **extract_primary**: Extrahiert die FBE's sowie die MFCC's aus den Frames.
4. **normalise_primary_features**: Reduziert Sequenzen gleicher Phoneme der FBE's auf ein einzelnes Phonem mittels *k-Means*. Anschliessend werden die entsprechenden MFCC's ebenfalls reduziert. Die MFCC's werden während der Filterselektion nicht weiter benötigt.
5. **select_filters**: Erstellt die Spektrogramme anhand der FBE's und unterteilt die Sprecher in Gruppen (vgl. 5.1.4).
6. **get_filter_set**: Erstellt das Filterkandidatenset anhand der Spektrogramme.
7. **calculate_integral_image**: Berechnet pro Spektrogramm das integrale Bild.
8. **get_filter_value**: Berechnet pro integralem Bild die Filterwerte aller Filterkandidaten.
9. **boost_filters**: Selektiert pro Sprechergruppe im Minimum die 30 aussagekräftigsten Filter anhand der zuvor berechneten Filterwerte mittels *AdaBoost*.

10. **select_final_filters**: Wählt die 30 aussagekräftigsten Filter aus den pro Sprechergruppe berechneten Filtern aus.

4.1.3. Ablauf Clustering

Anhand des in Abblidung 4.2 dargestellten Ablaufs werden in diesem Abschnitt die wichtigsten Funktionen des Clusterings beschrieben. Die Funktionen 1 bis 4 sind dieselben wie bei der Filterselektion (vgl. 4.1.2) und werden hier daher nicht ausgeführt.

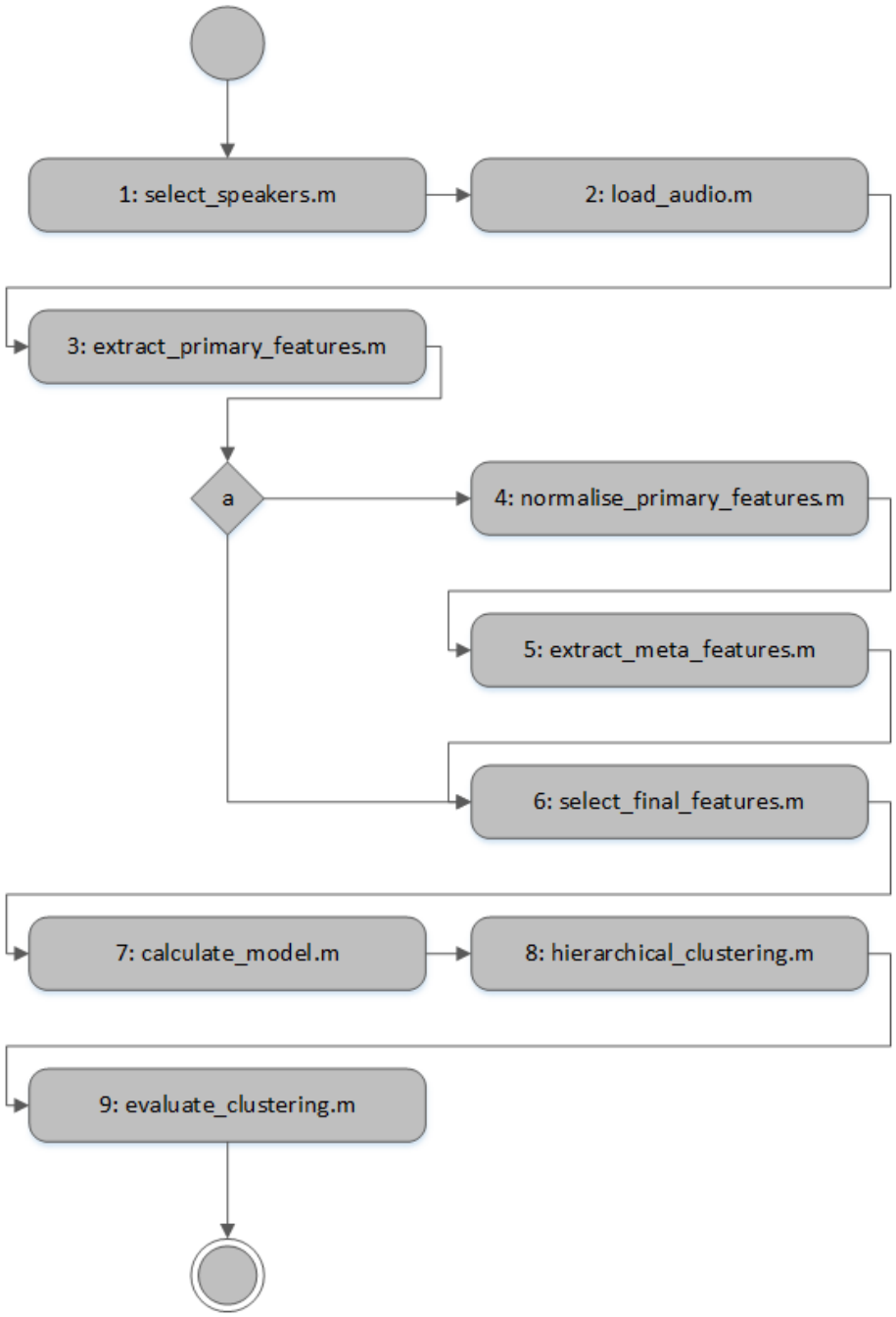

Abbildung 4.2.: Ablauf Clustering

a. **Art der Experimente**: Entscheidet abhängig von der Art des aktuellen Experiments, ob die Berechnungen der Funktionen 4 und 5 notwendig sind. Falls es sich um ein Experiment bezüglich des Basline-Ansatzes handelt, wird direkt bei Funktion 6 fortgefahren.

5. **extract_meta_features**: Erstellt zunächst die Spektrogramme anhand der FBE's und berechnet daraus die integralen Bilder. Extrahiert anschliessend die *Meta-Features* mittels den während der Filterselektion berechneten Filtern (vgl. 4.1.2).

6. **select_final_features**: Selektiert die Features für die weiteren Schritte, abhängig von der Kategorie der Experimente (vgl. 5).

7. **calculate_model**: Berechnet die GMM's anhand der zuvor selektierten Features.

8. **hierarchical_clustering**: Berechnet zunächst die Distanz zwischen den GMM's mit dem Distanzmass von Beigi et al.[2] und führt anschliessend das eigentliche Clustering mittels *Complete Linkage* durch.

9. **evaluate_clustering**: Berechnet die ICR und legt fest, nach welchem Schritt das Clustering die bestmöglichen Resultate liefert. Berechnet zudem Präzision, Ausbeute und *Diarization Error Rate* (vgl. 5.1.2).

4.2. Hardware und Betriebssystem

Für die Durchführung der Experimente wird ein Mac Pro aus der Serie März 2009 verwendet. Dieser verfügt über zwei 2.26 GHz *Quad-Core Intel Xeon* Prozessoren und 14 GB 1066 MHz *DDR3 ECC* Hauptspeicher. Als Betriebssystem wird Mac OS X, Version 10.9.2 (Mavericks) eingesetzt.

5. Experimente

5.1. Aufbau der Experimente

Die durchgeführten Experimente lassen sich in drei verschiedene Kategorien unterteilen. Experimente der Kategorie 1 beziehen sich auf den Baseline-Ansatz. Für diese Versuche wurden ausschliesslich MFCC's verwendet. Kategorie 2 beinhaltet diejenigen Experimente, die sich allein auf den neu erarbeiteten Ansatz stützen. Die Experimente der Kategorie 3 vereinen schliesslich die MFCC's aus Kategorie 1 mit den *Meta-Features* aus Kategorie 2.
Abschnitt 5.1.1 beschreibt den Datensatz, der den durchgeführten Versuchen zugrunde liegt. Die Bewertungskriterien, anhand derer die Experimente beurteilt werden, sind beschrieben unter 5.1.2. Weiter sind in den Abschnitten 5.1.3, 5.1.4 und 5.1.5 die detaillierten Parameter der Experimente der Kategorien 1 bis 3 beschrieben.

5.1.1. Datensatz

Für alle durchgeführten Experimente wurde der TIMIT-Datensatz verwendet. Dieser umfasst 630 Sprecher mit jeweils 10 in Studioqualität aufgenommenen Aussagen, gesprochen in einem von acht vorherrschenden Dialekten aus dem amerikanischen Englisch. Für die Experimente wurden zwei verschiedene Sets von Sprechern eingesetzt. Sprecherset 1 beinhaltet 20 unterschiedliche Sprecher, wovon 8 männlich und 12 weiblich sind. Sprecherset 2 beinhaltet 40 unterschiedliche Sprecher, wovon 25 männlich und 15 weiblich sind. Pro Sprecher wurden 8 der 10 Aussagen zu einer einzigen zusammenhängenden Aussage kombiniert. Das gleiche wurde für die 2 verbleibenden Aussagen vorgenommen. Daraus resultierten pro Sprecher 2 kombinierte Aussagen von rund 30 s sowie rund 7 s Dauer. Die vollständige Auflistung der verwendeten Audiodaten aus Sprecherset 1 und 2 sind im Anhang unter A.2.5 sowie A.2.6 ersichtlich.

5.1.2. Bewertungskriterien

Um eine vergleichbare und aussagekräftige Beurteilung der Experimente zu ermöglichen, wurden drei in der Sprechererkennung häufig eingesetzte Kriterien gewählt. Diese sind Ausbeute (*rec*), Präzision (*prec*) und *Diarization Error Rate* (DER).

1. **Präzision**: Die Präzision definiert das Verhältnis zwischen passend zugeordneten Segmenten und allen geclusterten Segmenten.

$$prec = \frac{\#\ passende\ Segmente}{\#\ geclusterte\ Segmente} * 100$$

   Segmente werden pro Cluster als passend bezeichnet, falls sie von demjenigen Sprecher stammen, der den überwiegenden Anteil an Segmenten im Cluster ausmacht.

2. **Ausbeute**: Die Ausbeute definiert das Verhältnis zwischen korrekt zugeordneten Segmenten und den gesamt verfügbaren Segmenten.

$$rec = \frac{\#\ korrekte\ Segmente}{\#\ verfügbare\ Segmente} * 100$$

   Segmente werden als korrekt bezeichnet, wenn sie einerseits die Bedingung der passenden Segmente erfüllen. Andererseits müssen sie sich in demjenigen Cluster befinden, welcher die meisten Segmente des entsprechenden Sprechers enthält.

3. **DER**: Die DER beschreibt Segmente, die nicht als das erkannt wurden, was sie repräsentieren. Dazu gehören grundsätzlich drei unterschiedliche Aspekte. Zum einen beinhaltet das diejenigen Segmente, die von einem Sprecher stammen aber dem falschen Sprecher zugeordnet wurden. Weiter gehören Segmente dazu, die nicht als Sprache erkannt wurden, obwohl sie von einem Sprecher stammen. Als dritter Aspekt beinhaltet die DER Segmente, die nicht von einem Sprecher stammen und dennoch einem Sprecher zugeordnet wurden. Da jedoch im verwendeten Datensatz nur Daten vorhanden sind, die von Sprechern stammen, sind die beiden letztgenannten Aspekte für diese Arbeit irrelevant. In diesem speziellen Fall macht die DER daher eine Aussage darüber, wie viele der vorhandenen Segmente den falschen Sprechern zugeordnet sind.

Ein Segment wird in dieser Arbeit durch die in einer Aussage enthaltenen Frames definiert. Dies hat zum Vorteil, dass unterschiedlich lange Aussagen entsprechend gewichtet werden können. Eine falsch zugeordnete kurze Aussage hat dadurch einen viel kleineren Einfluss auf die Erkennungsleistung als eine längere.

5.1.3. Umsetzung der Experimente aus Kategorie 1

Anhand der in der Sprechererkennung (vgl. 2.2) üblichen Abläufe wird in diesem Abschnitt beschrieben, mit welchen Parametern die Experimente der Kategorie 1 umgesetzt wurden. Die verwendeten Parameter wurden soweit als möglich aus der Arbeit von Stadelmann [21] übernommen.

1. **Preprocessing**: Während dem *Preprocessing* werden die Audiodaten mit einer Rate von 16 kHz geladen und auf den Bereich zwischen -1 und 1 normalisiert. Anschliessend werden Frames von 20 ms Länge und 50% Überschneidung erstellt und mittels *Pre-emphasizing* und einem α von 0.97 tiefe Frequenzen gefiltert bzw. hohe Frequenzen verstärkt.

2. **Feature Extraction**: Aus den Frames werden nach dem *Preprocessing* mittels Hamming-Fenstern und der 512-Punkte DFT sowie 24 Filtern der *Mel Filterbank* FBE's extrahiert. Die verwendeten Filter gehen dabei von 0 Hz bis $7'600$ Hz. Anhand der DCT werden diese weiter zu 19-dimensionalen MFCC's transformiert (Koeffizient 0 sowie die höchsten 4 werden verworfen) (vgl. 2.3.1).

3. **Modeling**: Die verwendeten GMM's bestehen jeweils aus 32 Gauss-Verteilungen und einer diagonalen Kovarianzmatrix. Sie werden mit zufällig aus den gegebenen Features gewählten Werten initialisiert und mit maximal $1'000$ Iterationen des EM-Algorithmus auf die gegebenen Daten angepasst. Um singuläre Kovarianzen zu vermeiden, wird ein Minimum von 0.00001 für die Werte der Kovarianzmatrizen verwendet.

4. **Recognition**: Für die Distanzberechnung zwischen zwei Modellen wird die Vorgehensweise von Beigi et al. [2] (vgl. 2.3.3) eingesetzt. Weiter wird während dem Clustering *Complete Linkage* verwendet, um die Distanz zwischen zwei Clustern zu bestimmen. Bei *Complete Linkage* berechnet sich die Distanz zwischen zwei Clustern anhand der zwei Elemente (d.h. eines pro Cluster) mit zueinander maximalem Abstand. Um das Clustering nach demjenigen Schritt zu beenden, der eine möglichst hohe Erfolgsrate liefert (d.h. im Bezug auf die im Experiment bestmögliche Erfolgsrate), wird die ICR zusammen mit einem optimalen Grenzwert verwendet. Für die Berechnung des optimalen Grenzwerts wird als erstes bestimmt, bei welchem Schritt des Clusterings die DER minimal ist. Anschliessend wird die ICR dieses Clustering-Schritts als optimaler Grenzwert eingesetzt.

5.1.4. Umsetzung der Experimente aus Kategorie 2

Dieser Abschnitt definiert die Parameter für die Experimente der Kategorie 2 bezüglich der unter 3.3 beschriebenen Vorgehensweise. Die Vorgehensweise während dem *Preprocessing*, dem *Modeling* und der *Recognition* sind mit 5.1.3 identisch und werden hier deshalb nicht aufgeführt. Als Trainingsdaten für die Filterselektion wurde Sprecherset 2 verwendet, das anschliessend auch während der Testphase eingesetzt wurde.

1. **Filterselektion**: Der erste Schritt zur Selektion der Filter ist das Erstellen von Spektrogrammen. Dafür werden, gleich dem Baseline-Ansatz, FBE's extrahiert. Das Normalisieren der FBE's (vgl. 3.3) wird mit *k-Means* und einer Anzahl von $\lfloor \frac{2T}{3} \rfloor$ Clustern umgesetzt (T entspricht der Anzahl FBE's). Die aus den normalisierten FBE's erstellten Spektrogramme repräsentierten die in 24 logarithmisch verteilten Frequenzbändern enthaltene Energie, gemessen über Zeitfenster von 120 ms, die jeweils um 10 ms inkrementiert werden. Die Bereiche, in denen ein Basisfilter (vgl. Abbildung 3.1) variieren kann, sind Frequenzbandposition von 1 bis 24, Frequenzbandbreite von 1 bis 24 und Zeit (d.h. Frames) von 1 (20 ms) bis 11 (120 ms). Der Wahl der zeitlichen Länge eines Spektrogramms liegt die Annahme zugrunde, dass zeitliche Information im längerfristigen Kontext (d.h. länger als 120 ms) einen zu hohen Anteil an sprachabhängigen Eigenschaften aufweist. Die Schritte der Frequenzbandposition und Positionierung in der Zeit sind linear. Die Frequenzbandbreite und die Ausdehnung über die Zeit wird in exponentiellen Schritten von 1.5 berechnet, so dass

$$x_n = \begin{cases} \lfloor x_{n-1}^{1.5} \rfloor + 1 & \text{wenn } \lfloor x_{n-1}^{1.5} \rfloor = x_{n-1} \\ \lfloor x_{n-1}^{1.5} \rfloor & \text{sonst} \end{cases}$$

gilt, wobei x_n der Frequenzbandbreite bzw. der Ausdehnung über die Zeit entspricht (vgl. Tabelle 5.1). Die Startgrösse eines Basisfilters berechnet sich jeweils anhand der enthaltenen Rechtecke (vgl. 3.1), wobei ein Rechteck zu diesem Zeitpunkt jeweils die Grösse $1x1$ hat.

	Schritt 0	Schritt 1	Schritt 2	Schritt 3	Schritt 4
Basisfilter 1					
Frequenzbandbreite	1	2	3	5	11
Ausdehnung Frames	2	4	8	–	–
Basisfilter 2					
Frequenzbandbreite	2	4	8	22	–
Ausdehnung Frames	1	2	3	5	11
Basisfilter 3					
Frequenzbandbreite	2	4	8	22	–
Ausdehnung Frames	2	4	8	–	–
Basisfilter 4					
Frequenzbandbreite	1	2	3	5	11
Ausdehnung Frames	3	6	–	–	–
Basisfilter 5					
Frequenzbandbreite	3	6	15	–	–
Ausdehnung Frames	1	2	3	5	11

Tabelle 5.1.: Exponentielle Schritte innerhalb eines Spektrogramms

Dies führt zu einem Filterkandidatenset mit genau $9'589$ Filtern. Anhand dieser Filter wird anschliessend pro Spektrogramm ein Vektor mit $9'589$ Filterwerten berechnet und zusammen mit den zugehörigen Sprecherlabels *AdaBoost* zur Selektion der aussagekräftigsten Filter übergeben. Aufgrund des enormen Bedarfs an Hauptspeicher während der Selektion, werden die Daten in Gruppen mit jeweils 4 unterschiedlichen Sprechern eingeteilt. Für das Sprecherset 2, das während der Filterselektion verwendet wird, werden also 5 unabhängige Filtersets berechnet. Aus diesen Filtersets werden danach die häufigsten 30 Filter ausgewählt, die schliesslich bei der *Feature Extraction* zum Einsatz kommen. 30 Filter entsprechen dabei nahezu der Anzahl der von Ke et al. verwendeten Filter (d.h. 32). Zudem ist bei Verwendung einer höheren Anzahl von Filtern zu erwarten, dass der Einsatz von GMM's die Erkennungsleistung senkt und somit keine geeignete Wahl darstellt.

2. **Feature Extraction**: Gleich wie bei der Filterselektion werden aus den normalisierten FBE's Spektrogramme erstellt. Anhand der 30 selektierten Filter werden aus den Spektrogrammen anschliessend die *Meta-Features* extrahiert.

5.1.5. Umsetzung der Experimente aus Kategorie 3

Die Parameter der Experimente aus Kategorie 3 sind weitgehend identisch mit denjenigen der Kategorien 1 und 2, weshalb in diesem Abschnitt nur auf die Unterschiede eingegangen wird. Das Kombinieren der *Primary-* und *Meta-Features* wird der *Feature Extraction* zugeordnet. Damit die Features kombiniert werden können, werden zwei Anpassungen vorgenommen.

1. **Phonemsequenzen**: Sequenzen gleicher Phoneme werden aus den FBE's gekürzt (vgl. 3.3) und gleichzeitig die entsprechenden MFCC's entfernt. Die Anpassung der MFCC's ist notwendig, um Verschiebungen in der Zugehörigkeit der Features zu verhindern.
2. **Normalisierung**: Die Features werden auf den Bereich zwischen 0 und 1 normalisiert, wodurch ein direktes Verketten der Features ermöglicht wird. Durch die Verkettung aller *Primary-* und *Meta-Features* ergeben sich 49-dimensionale Featurevektoren, wobei die *Meta-Features* klar in der Überzahl sind. Aus diesem Grund wird in dieser Kategorie eine zusätzliche Variante zur Verkettungen der Features verwendet. In der ersten Variante werden alle vorhandenen Features verwendet, wie zuvor beschrieben. Um den Einfluss des Übergewichts der *Meta-Features* auf die Erkennungsleistung zu eruieren, werden in der zweiten Variante, nur die 19 aussagekräftigsten Filter eingesetzt, was zu 38-dimensionalen Featurevektoren führt.

5.2. Resultate

Im Folgenden werden zum einen die während der *Feature Extraction* berechneten Filter beschrieben und analysiert und zum anderen die Ergebnisse aus den Experimenten der Kategorie 1 bis 3. Die Ergebnisse der Experimente basieren auf Durchschnittswerten, die anhand von 10 Wiederholungen pro Experiment ermittelt wurden. Dem liegt zugrunde, dass die verwendeten Modelle mittels Zufallswerten initialisiert wurden, wodurch die Ergebnisse variieren.
Abschnitt 5.2.1 befasst sich mit den berechneten Filtern. In Abschnitt 5.2.2 werden die erzielten Resultate beschrieben, die anschliessend in Abschnitt 5.2.3 interpretiert werden.

5.2.1. Beschreibung der Filter

Neben den erzielten Ergebnissen betreffend den unter 5.1.2 beschriebenen Bewertungskriterien sind die durch *AdaBoost* selektierten Filter ein wichtiger Bestandteil der Resultate. Um zu verstehen, welche Merkmale aus den Spektrogrammen extrahiert wurden, wird in diesem Abschnitt auf die wichtigsten Eigenschaften dieser Filter eingegangen. Abbildung 5.1 zeigt eine Auswahl von Filtern, die unterschiedliche Charakteristiken der selektierten Filter repräsentieren. Eine komplette Abbildung der Filter findet sich im Anhang unter A.2.4.

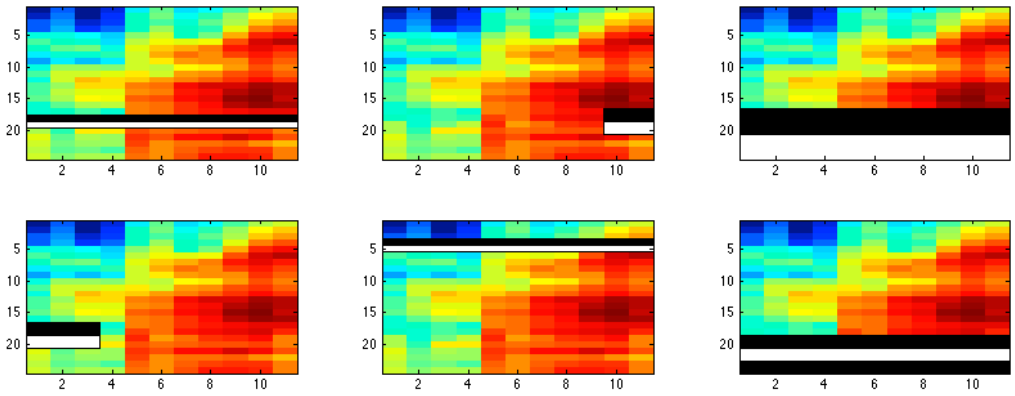

Abbildung 5.1.: Auswahl unterschiedlicher Filter aus dem selektierten Filterset

Mittels *AdaBoost* wurden insgesamt 187 unterschiedliche Filter selektiert, wobei keiner dieser Filter auf den Basisfiltern 2 und 3 (vgl. 3.1) basiert. Besonders auffallend ist zudem, dass 110 dieser 187 Filter Basisfilter 1 als Grundlage haben. Von den 30 aus diesen 187 ausgewählten Filtern basieren 29 ebenfalls auf Basisfilter 1 und nur ein einziger auf Basisfilter 4. Das bedeutet, dass sich die entscheidenden Informationen hauptsächlich durch Unterschiede der in benachbarten Frequenzbändern enthaltenen Energie ausdrückt.
Im Folgenden werden die Filter anhand der Ausdehnung in Frequenz und Zeit genauer betrachtet. 20 der ausgewählten Filter erstrecken sich über die gesamte pro Spektrogramm zur Verfügung stehende Zeit (d.h. über 11 Frames). Von diesen 20 extrahieren 4 Filter Informationen aus den Frequenzbändern 1 bis 4 (d.h. aus den hohen Frequenzen). Die übrigen 16 befinden sich in den Frequenzbändern zwischen 15 und 24 (d.h. in den tiefen Frequenzen). Dabei wird die Energiedifferenz überwiegend in jeweils 2 benachbarten Frequenzbändern gemessen, gefolgt von Messungen in jeweils 4 benachbarten Frequenzbändern. 10 der 30 Filter erstrecken sich über 1 bis 5 Frames und befinden sich alle in den Frequenzbändern zwischen 15 und 24. Auch hier zeigen sich die selben Tendenzen, was die Anzahl der gleichzeitig berücksichtigten Frequenzbänder angeht. Ebenfalls von Bedeutung ist, dass 9 der 30 selektierten Filter, dargestellt in Abbildung 5.2, die Energiedifferenz in denselben 4 Frequenzbändern messen und sich dabei nur in Position und Ausdehnung auf der Zeitachse unterscheiden. Die vollständigen Angaben betreffend Positionierung und Ausdehnung der Filter befindet sich im Anhang unter A.3.

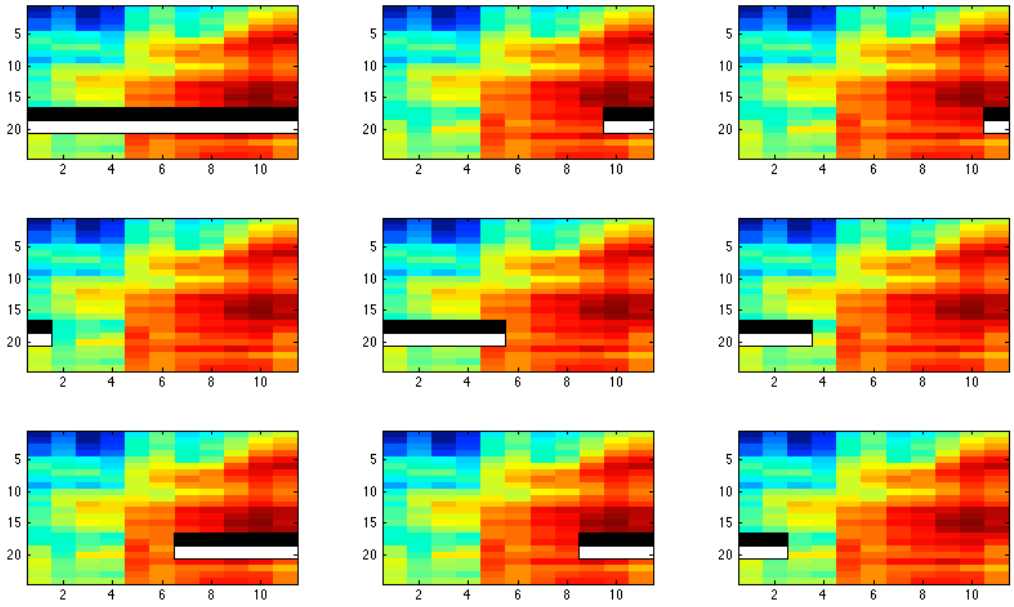

Abbildung 5.2.: Filter der Basisklasse 1 innerhalb derselben Frequenzbänder

5.2.2. Beschreibung der Messwerte

Die Messwerte werden einerseits pro Kategorie und andererseits zusammengefasst dargestellt. Dies erlaubt das unabhängige Betrachten der Ergebnisse sowie den direkten Vergleich. Die Resultate sind jeweils anhand der Sprechersets unterteilt, definiert durch die Anzahl unterschiedlicher Sprecher. Nebst den Kennzahlen Präzision, Ausbeute und DER wird auch der Geschlechteranteil falsch zugeordneter Segmente berücksichtigt, da dieser bei der Interpretation der Resultate eine wichtige Rolle spielt. Dabei steht die Abkürzung *gpw* für den Anteil falsch zugeordneter weiblicher und *gpm* für den Anteil falsch zugeordneter männlicher Sprecher. Die im Anhang dargestellten Tabellen A.1 und A.2 beinhalten neben den Anteilen falsch zugewiesener Segmente ebenfalls die zugehörigen Sprecher.

1. **Resultate Kategorie 1**: Tabelle 5.2 zeigt die erzielten Resultate bezüglich dem Baseline-Ansatz. Mit einer DER von 3.95% für 20 Sprecher stimmt das Ergebnis nicht mit demjenigen von Stadelmann [21] überein, wobei sich die DER von 12.76% für 40 Sprecher nahezu mit den von Stadelmann

erzielten 12.50% deckt. Auffallend ist, dass die falsch zugeordneten Segmente bei 40 Sprechern zum grössten Teil und bei 20 Sprechern ausschliesslich von weiblichen Sprechern stammen.

	prec [%]	rec [%]	DER [%]	gpw [%]	gpm [%]
20 Sprecher	97.68	96.05	3.95	100.00	0.00
40 Sprecher	89.10	87.24	12.76	89.36	10.64

Tabelle 5.2.: Resultate der Experimente aus Kategorie 1

2. **Resultate Kategorie 2**: Tabelle 5.3 zeigt die Resultate, die anhand des erarbeiteten Ansatzes erzielt wurden. Dabei fällt auf, dass die Ergebnisse für 40 Sprecher deutlich besser ausfallen als für 20 Sprecher. Zudem zeigt die Verteilung der falsch zugeordneten Segmente bei 20 Sprechern ein deutliches Übergewicht an weiblichen Sprechern, wohingegen die Verteilung bei 40 Sprechern relativ ausgeglichen ist. Dies lässt einen Zusammenhang mit den erzielten Resultaten erkennen.

	prec [%]	rec [%]	DER [%]	gpw [%]	gpm [%]
20 Sprecher	76.64	73.80	26.20	87.18	12.82
40 Sprecher	87.37	83.05	16.95	56.10	43.90

Tabelle 5.3.: Resultate der Experimente aus Kategorie 2

3. **Resultate Kategorie 3**: Tabelle 5.4 zeigt die Resultate, die anhand aller vorhandenen *Primary-* sowie *Meta-Features* erzielt wurden. Die Resultate der angepassten Variante der Verkettung sind in Tabelle 5.5 enthalten. Es ist zu erkennen, dass die angepasste Variante in allen Fällen deutlich bessere Ergebnisse erzielte. Gleichzeitig zeigt die Geschlechterverteilung der falsch zugeordneten Segmente ein sehr ähnliches Bild wie bei den Ergebnissen aus Kategorie 2.

	prec [%]	rec [%]	DER [%]	gpw [%]	gpm [%]
20 Sprecher	69.72	66.33	33.67	93.75	6.25
40 Sprecher	21.66	20.85	79.15	56.44	43.56

Tabelle 5.4.: Resultate der Experimente aus Kategorie 3 - Komplettes Featureset

	prec [%]	rec [%]	DER [%]	gpw [%]	gpm [%]
20 Sprecher	81.37	76.76	23.24	100.00	0.00
40 Sprecher	51.53	49.59	50.41	57.24	42.76

Tabelle 5.5.: Resultate der Experimente aus Kategorie 3 - Angepasstes Featureset

Abbildung 5.3 stellt die verketteten *Primary-* und *Meta-Features* des Sprechers FDAC1 basierend auf 8 zusammenhängenden Aussagen grafisch dar. Die ersten 19 Features repräsentieren dabei die MFCC's. Die darauf folgenden 30 Features sind *Meta-Features* von denen die ersten 29 auf der Basisklasse 1 basieren und das letzte Feature auf der Basisklasse 4. Auffallend ist, dass sich die Werte der Features 26 bis 35 und 39 bis 48 in einem sehr kleinen Bereich befinden. Zudem ist festzustellen, dass sich die Features 1 und 49 deutlich von den übrigen Features abheben.

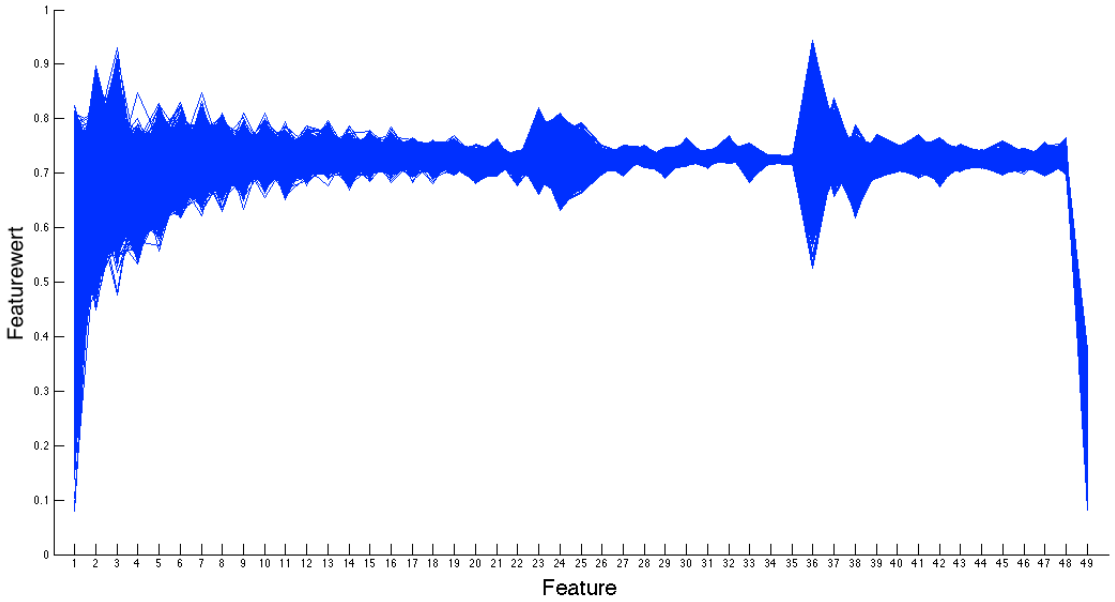

Abbildung 5.3.: Darstellung der verketteten Features des Sprechers FDAC1

4. **Vergleich der Resultate**: Aus Abbildung 5.4 sowie Tabelle 5.6 ist ersichtlich, dass der Baseline-Ansatz die besten Ergebnisse hervorbringt, sowohl gegenüber dem erarbeiteten Ansatz als auch gegenüber der Kombination aus beiden Ansätzen. Die Resultate der Experimente aus Kategorie 2 für 40 Sprecher sind mit einer Präzision von 87.37% und einer Ausbeute von 83.05% gegenüber dem Basline-Ansatz mit 89.10% Präzision und 87.24% Ausbeute dennoch in einem interessanten Bereich. Weit entfernt von den Resultaten der Kategorien 1 und 2 sind diejenigen der Kategorie 3 bezüglich 40 Sprecher. Bei 20 Sprechern hingegen, schneidet die angepasste Variante gegenüber Kategorie 2 besser ab.

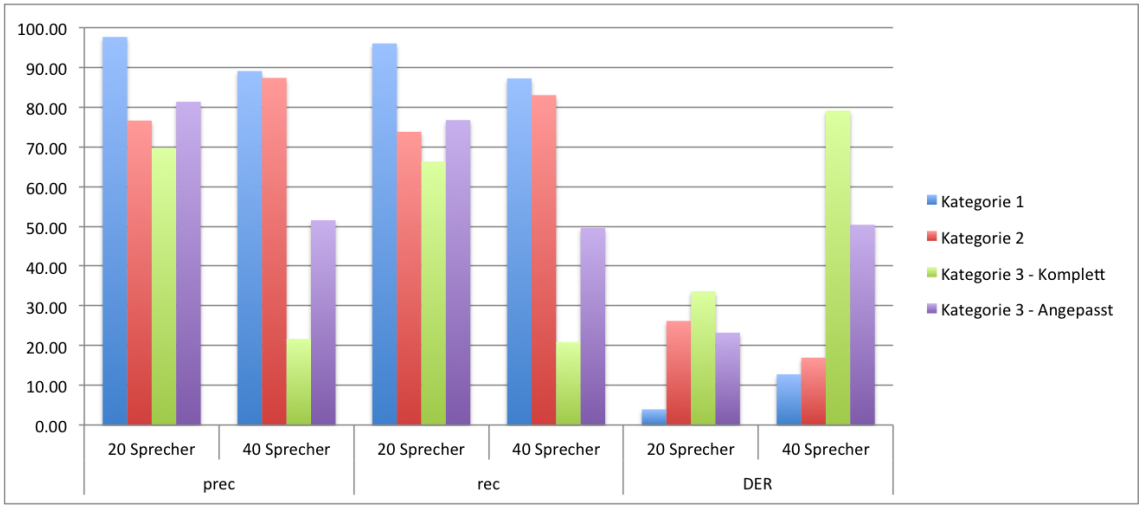

Abbildung 5.4.: Übersicht aller Resultate

	prec [%]		rec [%]		DER [%]	
	20 Sprecher	40 Sprecher	20 Sprecher	40 Sprecher	20 Sprecher	40 Sprecher
Kategorie 1	97.68	89.10	96.05	87.24	3.95	12.76
Kategorie 2	76.64	87.37	73.80	83.05	26.20	16.95
Kategorie 3 Komplett	69.72	21.66	66.33	20.85	33.67	79.19
Kategorie 3 Angepasst	81.37	51.53	76.76	49.59	23.24	50.41

Tabelle 5.6.: Übersicht aller Resultate

	gpw [%]		gpm [%]	
	20 Sprecher	40 Sprecher	20 Sprecher	40 Sprecher
Kategorie 1	100.00	89.36	0.00	10.64
Kategorie 2	87.18	56.10	12.82	43.90
Kategorie 3 Komplett	93.75	56.44	6.25	43.56
Kategorie 3 Angepasst	100.00	57.24	0.00	42.76

Tabelle 5.7.: Geschlechteranteil der falsch zugewiesenen Segmente

5.2.3. Interpretation der Resultate

Die Interpretation der Resultate wird schrittweise durchgeführt. Als erstes wird nach einer Ursache für die Unterschiede in der Umsetzung des Baseline-Ansatzes gegenüber derjenigen von Stadelmann gesucht und zudem eine Schwachstelle des Baseline-Ansatzes genauer betrachtet. Weiter werden die Ergebnisse der Experimente aus Kategorie 2 sowie die Vorgehensweise während der Filterselektion analysiert und auf Verbesserungspotential der angewandten Verfahren eingegangen. Abschliessend werden die Experimente der Kategorie 3 genauer betrachtet und ebenfalls nach Möglichkeiten zur Verbesserung gesucht.
Die Umsetzung des Baseline-Ansatzes unterscheidet sich hauptsächlich in zwei Bereichen von derjenigen von Stadelmann. Zum einen ist dies das verwendete Mass zur Berechnung der Distanz zwischen den Modellen und zum anderen die Art der Initialisierung von GMM's.

1. **Distanzmass**: Anstelle des Distanzmasses von Beigi et al. [2] wurde in der Arbeit von Stadelmann die CLR verwendet. Da diese Masse einen starken Einfluss auf die gesamte Erkennungsleistung haben, sind unterschiedliche Endergebnisse nicht zu vermeiden, was die Abweichungen teilweise erklärt.

2. **Initialisierung der Modelle**: Für die Initialisierung der GMM's wurden stets zufällige Werte gewählt, wodurch die Ergebnisse einerseits nicht exakt wiederholt werden konnten und andererseits ebenfalls gewisse Abweichungen gegenüber den Ergebnissen von Stadelmann mit sich bringen. Diese Abweichungen in Kombination mit denjenigen des Distanzmasses werden für die Unterschiede verantwortlich gemacht.

Abgesehen von unterschiedlichen Implementierungen lässt die Betrachtung der falsch zugeordneten Segmente auf Unzulänglichkeiten des Baseline-Ansatzes während der *Feature Extraction* schliessen.

3. **MFCC's**: Das Übergewicht von falsch zugeordneten Segmenten weiblicher Sprecher lässt erkennen, dass die in den MFCC's enthaltene Information nicht gleichermassen zur Unterscheidung männlicher und weiblicher Sprecher geeignet ist. Welcher Schritt der *Feature Extraction* dabei den Unterschied ausmacht, ist nur schwer zu lokalisieren. Eine mögliche Erklärung für die Unterschiede ist, dass aufgrund der Filter aus der *Mel Filterbank* diejenigen Frequenzen, welche die weibliche Stimme zu einem Grossteil ausmachen, zu stark zusammengefasst werden.

Die Experimente der Kategorie 2 legen nahe, dass das berechnete Set von 30 Filtern nicht ausreichend in der Lage ist, die entscheidenden Sprechermerkmale von den restlichen Daten zu trennen. Die zeitliche Abhängigkeit der Filter zeigen dennoch, dass die angewandten Verfahren im Grundsatz richtig sind. Die Resultate der Experimente werden einerseits auf ein Übertraining der Filter zurückgeführt und andererseits auf die Vorgehensweise während der Selektion der Filter.

4. **Zeitliche Abhängigkeit**: Zwei Drittel der selektierten Filter beziehen sich auf alle in den Spektrogrammen vorhandenen Frames. Dies bestätigt, dass der zeitliche Verlauf der Signale für die Erkennung von Sprechern eine wichtige Rolle spielt. Im Gegensatz dazu stehen drei Filter, die Informationen ausschliesslich aus einzelnen Frames beziehen und dadurch zeigen, dass entscheidende Informationen ebenfalls unabhängig von zeitlichen Aspekten vorhanden sind. Die klare Tendenz der Filter lässt aber dennoch erkennen, dass das Einbeziehen des zeitlichen Verlaufs der richtige Ansatz ist.

5. **Übertraining**: Die Ergebnisse der Experimente aus Kategorie 2 deuten auf ein Übertraining der Filter bezüglich Sprecherset 2 hin. Dies wird umso deutlicher unter Berücksichtigung der Geschlechterverteilung falsch zugeordneter Segmente zusammen mit den Anteilen männlicher und weiblicher Sprecher in den Sprechersets 1 und 2. Es ist erkennbar, dass die Filter hauptsächlich auf die Unterscheidung männlicher Sprecher ausgelegt sind, deren Stimmen tendenziell durch tiefere Frequenzen definiert werden als diejenigen weiblicher Sprecher. Das Fehlen der Filter in den mittleren Frequenzbandpositionen sowie die überwiegende Positionierung der Filter in den tieferen Frequenzbändern bestätigt diese Schlussfolgerung.

6. **Gruppierung der Sprecher**: Der Gruppierung der Sprecher, die aufgrund des begrenzten Hauptspeichers vorgenommenen wurde, wird ebenfalls ein negativer Effekt bezüglich Erkennungsleistung zugeschrieben. Durch den Kompromiss des Gruppierens gehen Zusammenhänge zwischen Sprechern unterschiedlicher Gruppen verloren. Es ist anzunehmen, dass Filter, die zur Unterscheidung aller vorhandenen Sprecher besser geeignet sind, dadurch tiefer gewichtet und somit nicht selektiert wurden. Weiter ist anzunehmen, dass das Übertraining der Filter durch das Gruppieren zusätzlich verstärkt wurde.

7. **Filterkandidatenset**: Die von Ke et al.[12] angepasst übernommene Vorgehensweise der Auswahl von Filterkandidaten mittels exponentiellen Schritten schliesst einen Grossteil von möglichen Filtern aus. Dabei lässt sich nur schwer feststellen, ob gleichzeitig nicht auch Filter mit hoher Aussagekraft ausgeschlossen werden. Ist dies jedoch der Fall, so kann allein dadurch nicht das volle Potential des erarbeiteten Ansatzes ausgeschöpft werden.

8. **Anzahl der Filter**: Die gewählte Anzahl Filter wurde stark von der Arbeit von Ke et al.[12] beeinflusst, in der damit überzeugende Resultate erzielt wurden. Im Bereich der automatischen Objekterkennung werden für eine verlässliche Klassifikation mindestens mehrere Dutzend, üblicherweise aber mehrere hundert bis mehrere tausend *weak Classifiers* eingesetzt[14]. Dies führt zum Schluss, dass die Aussagekraft von 30 Filtern höchstens genügend ist, um gegenüber dem Baseline-Ansatz verbesserte Ergebnisse zu liefern.

9. **Überschneidung der Filter**: Filter, die wichtige Informationen aus den Spektrogrammen extrahieren, sich diese Informationen aber teilweise oder ganz mit anderen Filtern teilen, verlieren dadurch an Aussagekraft. Durch diese Annahme werden die 9 Filter, dargestellt in Abbildung 5.2, als kontraproduktiv eingestuft. Dies aus dem Grund, dass bezüglich der relativ kleinen Anzahl von 30 Filtern (vgl. zuvor beschriebener Punkt betreffend Anzahl der Filter) die Aussagekraft einzelner besonders wichtig ist.

10. **Filterklassen**: Das deutliche Übergewicht der auf Basisfilter 1 basierenden Filter, die durch *AdaBoost* selektiert wurden, belegt, dass zur Unterscheidung von Sprechern wichtige Information in Energiedifferenzen von benachbarten Frequenzbändern enthalten ist. Dies deckt sich auch mit den Beobachtungen von Ke et al.[12] aus dem Bereich der Musikidentifikation. Dennoch kann nicht ausgeschlossen werden, dass anhand von Filtern, die auf den Basisfiltern 2 bis 5 basieren, entscheidende Information extrahiert werden kann. Die Verwendung von Filtern unterschiedlicher

Basisklassen wird zudem motiviert durch die Annahme, dass sich die Aussagekraft überschneidender Filter teilweise aufhebt, was insbesondere bei Filtern innerhalb derselben Basisklasse auftreten kann.

Die Resultate der Experimente aus Kategorie 3 zeigen deutlich, dass direktes Verketten der Features eine zu stark vereinfachte Vorgehensweise darstellt. Im Folgenden wird nach Erklärungen gesucht, weshalb die Kombination von *Primary-* und *Meta-Features* die Erkennungsleistung nicht positiv unterstützte und welche Aspekte zu berücksichtigen sind.

11. **Normalisierung**: Die Skala, auf der sich die Werte der *Meta-Features* befinden, ist rund zwanzig mal höher als diejenige der *Primary-Features*, was eine Angleichung der Skalen unerlässlich macht. *Meta-Features*, die bereits vor der Normalisierung nahe beieinanderliegende Werte aufwiesen, werden dadurch im Verhältnis zu den MFCC's in einen noch kleineren Bereich abgebildet. Dies führt dazu, dass viele der Filterwerte an Aussagekraft bzw. Individualität verlieren und somit einen negativen Einfluss auf die Modelle haben, was die ernüchternden Resultate aus Kategorie 3 zumindest teilweise erklärt. Teilweise deshalb, weil der Unterschied zwischen den beiden Vorgehensweisen aus Kategorie 3 nur begrenzt auf die unterschiedliche Anzahl der *Meta-Features* zurückgeführt wird (vgl. folgender Punkt betreffend Modell).
12. **Modell**: Wie bereits unter 2.4.1 erwähnt, ist die Verwendung von GMM's ist für die Verarbeitung hochdimensionaler Features, wie sie aus der Verkettung von *Primary-* und *Meta-Features* entstehen, keine geeignete Wahl [22]. Es wird davon ausgegangen, dass dies, zusätzlich zur Normalisierung, dazu beigetragen hat, die Erkennungsleistung zu schmälern, worauf der Vergleich der Ergebnisse aus Kategorie 3 ebenfalls hindeutet.

Zusammenfassend lässt sich sagen, dass die Ergebnisse hauptsächlich auf die Selektion der Filter zurückzuführen sind und die Vorgehensweise grosses Verbesserungspotential beinhaltet. Obwohl das Kombinieren der Features ebenfalls Potential zur Verbesserung aufweist, ist der Fokus zunächst auf Optimierungen der Filterselektion zu legen und erst anschliessend auf geeignetere Verfahren bezüglich Kombination von MFCC's und *Meta-Features*. Das Ziel der Verbesserung bestehender Verfahren im Bereich der automatischen Sprechererkennung wurde bezüglich der Aufgabenstellung nicht erreicht. Dennoch ist mit dieser Arbeit eine Grundlage geschaffen, anhand derer die erarbeiteten Verfahren weiter ausgebaut werden können, um dadurch die gewünschte Steigerung der Erkennungsleistung zu erreichen.

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

Das Ziel dieser Arbeit war es, ein geeignetes Verfahren zu erarbeiten, das es ermöglicht, die Erkennungsleistung der heute gängigen Systeme im Bereich der automatischen Sprechererkennung zu steigern. Stadelmann zeigt in seiner Arbeit [21] anhand eines Clustering-Experiments deutlich, dass die heutigen Verfahren mit der Komplexität der Aufgabe überfordert sind, sobald die Anzahl Sprecher eine kritische Grenze zwischen 20 und 40 überschreitet. Stadelmanns Ergebnisse zeigen weiter, dass die zeitliche Abhängigkeit von Frames, die bei gängigen Vorgehensweisen während der *Feature Extraction* verloren geht, erfolgreich zur Steigerung der Erkennungsleistung eingesetzt werden kann. Um eine Möglichkeit zu finden, diese Abhängigkeit in den Sprechermodellen zu berücksichtigen, wurden fünf Ansätze analysiert, die sich auf ähnliche Problemstellungen beziehen. Aus diesen Ansätzen ging die Arbeit *Computer vision for music identification* von Ke et al. [12] als vielversprechendster Kandidat hervor.
Ke at al. befassten sich mit der Identifikation von Musikstücken anhand weniger Sekunden Audiodaten. Ihr Ansatz war, aus gegebenen Audiodaten Spektrogramme zu erstellen, um daraus die eindeutigen Merkmale der Musikstücke zu gewinnen. Die Abbildung der Audiosignale als Spektrogramme erlaubte dabei den Einsatz von bewährten Techniken aus dem Bereich des *Image Retrievals*. Für das Extrahieren der entscheidenden Merkmale aus den Spektrogrammen trainierten sie ein Set von Filtern, das auf den von Viola und Jones [26] definierten Basisfiltern zur automatischen Erkennung von Objekten beruht.
Für die vorliegende Arbeit wurde aus der Vorgehensweise von Ke et al. ein Konzept erarbeitet, das die entscheidenden Schritte auf die gegebene Problemstellung überträgt. Dieses Konzept wurde aufgeteilt in eine Trainings- und eine Testphase.
In der Trainingsphase wurde ein geeignetes Set von Filtern berechnet, um damit eindeutige Merkmale zur Erkennung von Sprechern aus Spektrogrammen zu extrahieren. Die Testphase wurde konzipiert, um das eigentliche Clustering durchzuführen, anhand dessen Ergebnissen der Erfolg des erarbeiteten Ansatzes definiert wurde. Beide Phasen nutzten dieselbe Vorgehensweise bei der Erstellung der Spektrogramme. Während dem *Preprocessing* wurden die Audiosignale geladen und in Frames eingeteilt. Mittels *Pre-emphasizing* wurden danach die hohen Frequenzen in den Frames verstärkt und mögliche Störgeräusche sowie sprecherunspezifische Information in den tiefen Frequenzen abgeschwächt. Anschliessend wurden während der *Feature Extraction* FBE's aus den Frames extrahiert und gleichzeitig Sequenzen gleicher Phoneme durch ein entsprechendes einzelnes Phonem ersetzt. Anhand der angepassten FBE's wurden danach die Spektrogramme erstellt. Während der Trainingsphase wurde neben der Erstellung der Spektrogramme ebenfalls ein Filterkandidatenset definiert. Dieses Set enthielt Variationen der von Viola und Jones [26] definierten Basisfilter. Die Basisfilter variierten innerhalb der Spektrogramme in Frequenzbandposition, Frequenzbandbreite und Zeit. Dies führte zu einem Kandidatenset von knapp $10'000$ Filtern. Für jedes Spektrogramm wurden anschliessend alle Filterwerte berechnet und mit einer Variante des *AdaBost*-Algorithmus die 30 aussagekräftigsten Filter selektiert. Anhand dieser 30 Filter wurden in der Testphase aus den Spektrogrammen die *Meta-Features* extrahiert, welche die zeitliche Abhängigkeit der Frames beinhalteten. Um aus diesen Features Modelle der Sprecher zu erzeugen, wurden GMM's eingesetzt. Die Distanz zwischen Modellen wurde mit dem Distanzmass von Beigi et al. [2] berechnet und während dem Clustering zusammen mit *Complete Linkage* eingesetzt. Mit Hilfe des ICR wurde entschieden, bei welchem Schritt das Clustering zu beenden ist.
Um die Ergebnisse dieser Arbeit zu vergleichen, wurde zudem derselbe Baseline-Ansatz umgesetzt, der auch in der Arbeit von Stadelmann [21] eingesetzt wurde. Dieser nutzt dieselbe Vorgehensweise beim *Preprocessing*, dem *Modeling* sowie der *Recognition* wie der erarbeitete Ansatz. Als Features wurden jedoch MFCC's eingesetzt.
Für die Durchführung der Experimente wurde der TIMIT-Datensatz genutzt. Um die Vergleichbarkeit sowie die Aussagekraft der Ergebnisse zu gewährleisten, wurden als Kriterien zur Beurteilung die drei

Masse Präzision, Ausbeute und *Diarization Error Rate* verwendet. Die Experimente wurden in drei Kategorien unterteilt. Kategorie 1 beinhaltet diejenigen Experimente, die ausschliesslich MFCC's für das *Modeling* verwendeten, wohingegen bei den Experimenten der Kategorie 2 ausschliesslich *Meta-Features* eingesetzt wurden. Zur Kategorie 3 gehören schliesslich diejenigen Experimente, welche die Features der Kategorien 1 und 2 vereinten. Dabei wurden die MFCC's mit den entsprechenden *Meta-Features* verkettet.
Die Resultate der Experimente zeigen, dass die vorgeschlagene Vorgehensweise weiterer Anpassung bedarf. Das beste Resultat, neben dem Baseline-Ansatz, wurde erzielt mit den Experimenten aus Kategorie 2 für 40 unterschiedliche Sprecher. Daraus resultierte eine Präzision von 87.37% gegenüber dem Baseline-Ansatz mit 89.10% und eine Ausbeute von 83.05% gegenüber 87.24%. Der Vergleich von Kategorie 1 zu Kategorie 2 bei 20 unterschiedlichen Sprecher ergab dabei deutlich schlechtere Resultate. Dieser Umstand ist auf ein Übertraining der Filter zurückzuführen, da die Filter anhand derselben 40 Sprecher trainiert wurden, die auch in der Testphase zum Einsatz kamen.
Anhand der Ergebnisse für 40 Sprecher wird dem vorgeschlagenen Ansatz dennoch viel Potential beigemessen, da die Werte zeigen, dass ein grosser Teil der charakteristischen Information extrahiert werden kann. Weitere Gründe, warum keine Verbesserung gegenüber dem Baseline-Ansatz erreicht wurde, sind hauptsächlich bei der Wahl der Filter zu suchen. Es ist zu erwarten, dass z.B. die Verwendung eines geeigneteren Kandidatenfiltersets oder auch die Anpassung der Trainingsdaten eine entscheidende Veränderung mit sich bringt. Nebst diesen und weiteren vielversprechenden Anpassungsmöglichkeiten betreffend Filterwahl sind auch die Experimente der Kategorie 3 zu berücksichtigen. Die in diesen Experimenten im besten Fall erzielte Präzision von 81.37% und Ausbeute von 76.76% ist hauptsächlich auf die Vorgehensweise bei der Kombination der Features zurückzuführen. Es ist anzunehmen, dass diesbezüglich ein geeigneteres Verfahren entscheidend zur Verbesserung der Erkennungsleistung beitragen kann.

6.2. Taktischer Ausblick

Aus der vorliegenden Arbeit gehen mehrere Ziele hervor, die kurzfristig umsetzbar sind. Ein wichtiger Faktor zur Umsetzung sind dabei die zur Verfügung stehenden Ressourcen, um entsprechende Experimente durchzuführen.
Die Interpretation der Resultate lässt (vgl. 5.2.3) erkennen, wie sich die Filterselektion anhand verschiedener Anpassungen, die im Folgenden aufgeführt sind, entscheidend verbessern lässt.

1. **Übertraining**: Um eine geeignete Generalisierung der Filter zu erreichen, ist bei der Berechnung der Filter ein möglichst ausgeglichenes Set von Sprechern bezüglich Geschlechteranteil zu verwenden. Gleichzeitig wird die Generalisierung der Filter umso stärker gefördert, je mehr Trainingsdaten während der Berechnung zur Verfügung stehen. Die Verwendung eines Sprechersets, das eine maximale und gleichzeitig ausgewogene Anzahl männlicher und weiblicher Sprecher aus dem TIMIT-Datensatz enthält, ist für weitere Experimente zu bevorzugen. Zu berücksichtigen ist aber, dass dadurch der Bedarf an Hardware-Ressourcen und Zeit für die Berechnungen stark gesteigert wird.

2. **Gruppierung der Sprecher**: Auf die Gruppierung der Sprecher ist nach Möglichkeit gänzlich zu verzichten. Dadurch kann sichergestellt werden, dass während der Filterselektion jegliche zur Verfügung stehende Information berücksichtigt wird und die gewählten Filter nicht nur auf wenige Sprecher abgestimmt sind. Ebenfalls kann so, unter Verwendung eines zuvor beschriebenen Sprechersets, ein Übertraining nahezu ausgeschlossen werden. Um dies umzusetzen, bedarf es jedoch deutlich mehr Hardware-Ressourcen, hauptsächlich in Form von Hauptspeicher, als für diese Arbeit zur Verfügung stand.

3. **Filterkandidatenset**: Je genauer ein Filter in der Lage ist, die Sprecher zu unterscheiden, desto besser wird die zu erwartende Erkennungsleistung. Um sicherzustellen, dass das Filterkandidatenset eben diese Filter enthält, sind die Basisfilter in Frequenzbandbreite und Ausdehnung über die Zeit ebenfalls in linearen Schritten zu variieren.

4. **Filterklassen**: Je weniger Filter zur Bestimmung der Merkmale eines Audiosignals eingesetzt werden, desto wichtiger ist die Aussagekraft eines einzelnen Filters. Überschneidungen der Filter und der dadurch angenommene Verlust an Aussagekraft ist daher zu vermeiden. Ein Ansatz, dies zu erreichen, ist, Filter aus unterschiedlichen Basisklassen zu verwenden und dabei einen minimalen Anteil pro Basisklasse zu erzwingen. Dabei ist aber zunächst zu eruieren, welche Basisfilter sich dazu überhaupt eignen. Aus diesem Grund sind Experimente sinnvoll, die Filtersets basierend auf jeweils einem Basisfilter einsetzen und dadurch die Erkennungsleistung pro Basisklasse klar aufzeigen.

6.3. Strategischer Ausblick

Im Hinblick auf mittel- bis langfristige Ziele, bietet diese Arbeit sowie die Sprechererkennung im Allgemeinen verschiedene Ansatzpunkte. Die Steigerung der Effizienz der eingesetzten Verfahren, um das Einbinden in eine Smartphone-Applikation zu ermöglichen, ist dabei nicht explizit aufgeführt, bleibt jedoch ein weiteres langfristiges Ziel.
Die ersten vier Punkte beziehen sich auf Verbesserungen und Ergänzungen bezüglich des Systems, das in dieser Arbeit umgesetzt wurde. Der fünfte und letzte Punkt befasst sich noch einmal mit neuronalen Netzen und distanziert sich damit klar von den vorhergehenden Punkten.

1. **Anzahl der Filter**: Ausgehend von der Annahme, dass 30 Filter für eine zuverlässige Sprechererkennung kaum ausreichend sind, ist die Wahl eines grösseren Filtersets durchaus naheliegend. Die optimale Grösse des entsprechenden Sets ist anhand von Experimenten zu ermitteln. Dies bedingt jedoch ebenfalls den Einsatz eines Verfahrens, das in der Lage ist, hochdimensionale Features zu verarbeiten, wie beispielsweise SVM's. In diesem Zusammenhang sind Alignment Kernels denkbar, anhand derer eine späte Integration von zeitlichen Aspekten ermöglicht wird (vgl. Joder et al.[11]). Die *Meta-Features* stellen dabei gleichzeitig eine Form der frühen Integration dar.

2. **Entscheidungsmodell**: Bezüglich des Einsatzes der erarbeiteten Verfahren in einer Smartphone-Applikation vernachlässigt die Verwendung von Daten, die keinerlei Störgeräusche enthalten eine wichtigen Aspekt, den es zu berücksichtigen gilt. Das von Ke et al.[12] vorgeschlagene Entscheidungsmodell, um zu bestimmen, ob Ausschnitte aus Audiodaten von Störgeräuschen überlagert sind, ist daher eine vielversprechende Komponente. Ein solches Entscheidungsmodell kann jedoch nicht direkt übernommen werden und verlangt zunächst die Ausarbeitung eines entsprechenden Konzepts.

3. **CLR und gemeinsame Varianz**: Neben der Umsetzung des Baseline-Ansatzes, welcher das Distanzmass von Beigi et al.[2] zur Berechnung der Distanz zwischen Modellen verwendet, wurde versuchsweise auch die CLR eingesetzt. Um die CLR zwischen zwei Modellen zu berechnen, spielt die Varianz beider Modelle eine wichtige Rolle. Aufgrund eines Fehlers, der während der Umsetzung übersehen wurde, kam bei der Berechnung jedoch nur die Varianz des einen Modells zum Einsatz. Die Ergebnisse, die daraus resultierten, waren verblüffend. 40 Sprecher beispielsweise wurden fehlerfrei geclustert und in einem weiteren Versuch mit 200 Sprechern lagen Präzision und Ausbeute bei 87.77% bzw. 89.67%, also in einem Bereich, der anhand des Baseline-Ansatzes für 40 Sprecher zu erwarten ist. Der Grund, warum diese Ergebnisse nicht bereits in Kapitel 5 präsentiert wurden, ist, dass die zur Erklärung dieser Resultate notwendige theoretische Grundlage im Rahmen dieser Arbeit nicht eruiert werden konnte. Es existieren in der Literatur ähnliche Herangehensweisen bezüglich gemeinsamer Varianzen, wie z.B. einer der Ansätze aus der Arbeit von Reynolds et al.[18], bei dem Modelle eingesetzt werden, die eine übereinstimmende Varianz besitzen.

4. **Pitch**: Pitch ist eines der wichtigsten Features, um die Stimmerzeugung der Stimmbänder zu charakterisieren. Pitch enthält viel sprecherspezifische Information, die bereits in verschiedenen Systemen erfolgreich zur Sprechererkennung eingesetzt wurde, wie z.B. die Arbeit von Zheng[31]. Als Ergänzung des erarbeiteten Ansatzes ist das Berücksichtigen von Pitch eine s Herangehensweise. Die Arbeit von Zhu et al. aus dem Jahr 2009[32] zeigt, wie im Pitch enthaltene Informationen in Verbindung mit MFCC's erfolgreich zur Steigerung der Erkennungsleistung eingesetzt werden

können. Eine solche Vorgehensweise ist durchaus denkbar in Verbindung mit den in dieser Arbeit vorgeschlagenen *Meta-Features*, wobei eine geeignete Variante zur Kombination der Features gefunden werden muss.

5. **Neuronale Netze**: Wie bereits aus der Beurteilung der fünf Ansätze (vgl. 3.1) ersichtlich wurde, wird neuronalen Netzen aufgrund ihrer Erfolge in vielzähligen Disziplinen grosses Potential zugesprochen. Aufgrund der hohen Komplexität und dem nur schwer zugänglichen Spezialwissen (vgl. 2.4.5), wurde jedoch auf den Einsatz neuronaler Netze in dieser Arbeit verzichtet. Die Resultate aus der jüngsten Forschung im Bereich *Deep Learning* (vgl. Koutník et al.[13]) lassen erwarten, dass mit solchen Netzen auch im Bereich der Sprechererkennung ein neuer Massstab gesetzt werden kann. Die erfolgreiche Umsetzung eines entsprechenden Systems verlangt jedoch neben ausreichenden Hardware-Ressourcen einen hohen Zeitaufwand.

Es bleibt abzuwarten, welcher Ansatz schlussendlich den entscheidenden Durchbruch in der Sprechererkennung bringt und welche Rolle die weitere Forschung am InIT dabei spielt. Das Bedürfnis eines ausreichend zuverlässigen Systems zur automatischen Sprechererkennung bleibt indessen bestehen und gibt dadurch weiterhin Anlass zur Erforschung neuer Verfahren.

A. Anhang

A.1. Projektmanagement

A.1.1. Offizielle Aufgabenstellung

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Talkalyzer: Neue Algorithmen für automatische Sprecher-Erkennung
BA14_stdm_1

BetreuerInnen:	Thilo Stadelmann, stdm Mark Cieliebak, ciel
Fachgebiete:	Datenanalyse (DA) Digitale Signalverarbeitung (DSV) Information Security (IS) Software (SOW)
Studiengang:	ET / IT
Zuordnung:	Institut für angewandte Informationstechnologie (InIT)
Gruppengrösse:	1

Kurzbeschreibung:
Die computergestützte Erkennung von Personen anhand ihrer Stimme hat viele spannende Anwendungen, z.B.:

- In biometrischen Sicherheitssystemen (etwa zur Zutrittskontrolle in Kraftwerken)
- In Verfahren zur automatischen semantischen Analyse von Audio- und Videorecordings (etwa, um in Spielfilmen nach Szenen mit bestimmten Schauspielern zu suchen)

Am InIT wurde daher ein Prototyp (in Matlab und Java) entwickelt, der die Besonderheiten der Stimme eines Menschen aus Beispielen extrahiert und danach in der Lage ist, für weiteren Sprachinput zu entscheiden, ob dieser von dem zuvor "trainierten" Sprecher stammt oder nicht. Der bestehende Code ist sehr kompakt und übersichtlich. Er implementiert die wesentlichen Stufen der Stimmerkennung und kann in dieser Bachelor-Arbeit (BA) als Beispiel und/oder Grundlage dienen.

Im Rahmen dieser BA sollen nun neue Verfahren implementiert und ausprobiert werden, um die Erkennungsrate von Sprechern zu steigern: Wie kann man beispielsweise die zeitliche Abfolge von Lauten besser für die Identifikation der Stimme nutzen? Hierzu existieren Ansätze in der Literatur und bei den Betreuern, aber auch eigene Ideen dürfen gerne eingebracht werden.

Diese BA richtet sich an Studierende, die sich gerne der Herausforderung einer (sehr praxisrelevanten) Forschungsfrage stellen möchten. Dabei werden keinerlei Vorkenntnisse in Sprachverarbeitung oder Datenanalyse vorausgesetzt. Vielmehr werden Sie durch ein erfahrenes Team und gute Materialien schnell up-to-date gebracht und starten dann Ihre eigenen Untersuchungen. Dabei bietet sich Ihnen die Chance auf eine wirklich neue Entwicklung!

Parallel zu dieser Arbeit findet eine weitere BA statt, in der eine Android-App für Sprechererkennung entwickelt wird. Beide Arbeiten werden gemeinsam betreut und sollen in Abstimmung realisiert werden. Insbesondere soll es möglich sein, die neuen/verbesserten Verfahren einfach in der App zu integrieren.

Voraussetzungen:
- Freue am Programmieren
- Affinität zu algorithmischen Fragestellungen
- Lust, sich vielleicht auch erstmals mit wissenschaftlichen Publikationen zu beschäftigen
- Hilfreich, aber nicht notwendig können sein: Erste Erfahrungen mit Matlab oder C++ (für Lesen von Beispielcode)

Weiterführende Informationen:
https://dublin.zhaw.ch/~stdm/?page_id=77

Donnerstag 5. Dezember 2013 11:09

A.1.2. Aufgabenstellung Bachelor-Arbeit Kündig

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Talkalyzer: Mobile-App zur automatischen Sprecher-Erkennung
BA14_ciel_1

BetreuerInnen:	Mark Cieliebak, ciel Thilo Stadelmann, stdm
Fachgebiete:	Software (SOW)
Studiengang:	IT
Zuordnung:	Institut für angewandte Informationstechnologie (InIT)
Gruppengrösse:	1

Kurzbeschreibung:
In dieser Bachelor-Arbeit (BA) soll eine Mobile-App entwickelt werden, die für eine Diskussion oder Besprechung live ermittelt, wer wie viel redet. Damit soll es z.B. einem Vorgesetzen möglich sein zu erkennen, ob er in einem Mitarbeitergespräch zu viel redet oder ob das Gespräch ausgeglichen ist. Analog könnte auch ein Ehepaar feststellen, wer in den Diskussionen das Sagen hat.

Hintergrund: Am InIT wurde in einem Forschungsprojekt bereits ein Prototyp entwickelt, der die Besonderheiten der Stimme eines Menschen aus Tonaufnahmen extrahiert. Das System ist anschliessend in der Lage, für neue Tonaufnahmen zu erkennen, wann der zuvor "trainierte" Sprecher redet. Dies funktioniert auch, wenn mehrere Personen miteinander sprechen. Der Code wurde in Matlab und Java implementiert und ist sehr kompakt und übersichtlich.

Aufgabe: Im Rahmen dieser BA entwickeln Sie eine Mobile-App, mit der Benutzer live erkennen können, wer wann und wie viel spricht. Dazu soll der vorhandene Code auf Android portiert und rundherum eine ansprechende App gebaut werden. Das Ziel ist eine marktreife Applikation, die Sie im App-Store publizieren können.

Hier einige wesentliche Teilaufgaben, die Sie im Laufe dieser BA bearbeiten werden:

- Entwicklung einer intuitiven Visualisierung der Sprecher-Anteile
- Entwurf einer Architektur, die es erlaubt neue Algorithmen zu Sprechererkennung einfach zu integrieren
- Portierung der existierenden Verfahren/Libraries auf die Mobile Plattform
- Performance-Optimierung, um Analysen in Quasi-Echtzeit (live) zu ermöglichen
- Evaluation, wie gut die Analysen in der Praxis funktionieren.

Parallel zu dieser Arbeit findet eine weitere BA statt, in der die Verfahren zur Sprechererkennung weiterentwickelt und optimiert werden. Beide Arbeiten werden gemeinsam betreut und sollen in Abstimmung realisiert werden. Insbesondere soll es möglich sein, die neuen/verbesserten Verfahren einfach in der App zu integrieren.

Voraussetzungen:
- Freude an intuitiven GUIs und gutem Design
- Entwicklung von Mobile Apps auf Android
- Hilfreich, aber nicht notwendig: Erfahrung mit Matlab

Donnerstag 5. Dezember 2013 11:09

A.2. Weiteres

A.2.1. Beschreibung der elektronischen Daten

Dieser Arbeit liegt eine CD mit der elektronischen Form dieses Berichts bei. Ebenfalls auf der CD enthalten sind die in MATLAB umgesetzten Experimente in Form einer *ZIP*-Datei. Der Inhalt der ZIP-Datei ist folgendermassen aufgebaut:

1. **0_ba14_stdm_1** (Dateiordner): Enthält alle Daten und Funktionen Bezüglich den Experimenten sowie die Datei *readme.txt* als Anleitung zum Starten der Experimente.

2. **0_settings** (Dateiordner): Enthält die Konfigurationsdateien bezüglich der Filterselektion sowie dem Clustering (vgl. 4.1.1).
3. **1_clustering** (Dateiordner): Enthält das Skript zum Starten des Clusterings (vgl. 4.1.1).
4. **2_filterselection** (Dateiordner): Enthält das Skript zum Starten der Filterselektion (vgl. 4.1.1).
5. **3_ba14_lib** (Dateiordner): Enthält die im Rahmen dieser Arbeit umgesetzten MATLAB-Funktionen (vgl. 4.1.1).
6. **4_ext_libs** (Dateiordner): Enthält die Bibliotheken von *HTK MFCC MATLAB* von Wojcicki (vgl. 4.1.1).
7. **5_evaluation_data** (Dateiordner): Enthält bestimmte Daten zur Auswertung der Experimente (die vollständigen Daten überschreiten das Volumen des Datenträgers).
8. **6_audiodata** (Dateiordner): Enthält den TIMIT-Datensatz als ZIP-Datei (muss vor den Experimenten im selben Ordner entpackt werden) sowie die Dateien mit den Sprechersets (vgl. 4.1.1).
9. **7_final_filters** (Dateiordner): Enthält die den Experimenten zugrundeliegenden 30 selektierten Filter.

A.2.2. Externe Software

1. **Name der Software**: HTK MFCC Matlab
2. **Author**: Kamil Wojcicki
3. **Erstellungsdatum**: 11.09.2011
4. **Downloadlink**: http://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab
5. **BSD Lizenz**:
   Copyright (c) 2011, Kamil Wojcicki
   All rights reserved.

   Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

   - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
   - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution
   - Neither the name of the University of Texas at Dallas nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS „AS IS“AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2.3. Messwerte der Experimente

	Kat. 1 - 20 Spr.		Kat. 1 - 40 Spr.		Kat. 2 - 20 Spr.		Kat. 2 - 40 Spr.	
Anteil	Häufigk.	Name	Häufigk.	Name	Häufigk.	Name	Häufigk.	Name
Weibliche Sprecher								
	10	FJWB0	8	FCMH0	4	FDAC1	4	FAKS0
	6	FPAS0	10	FDRD1	10	FDRD1	2	FCMH0
	14	FRAM1	2	FJAS0	2	FJAS0	2	FDAC1
			6	FJEM0	18	FJEM0	14	FDRD1
			8	FJWB0	16	FJWB0	12	FJAS0
			8	FKMS0	6	FPAS0	14	FJEM0
			20	FPAS0	12	FRAM1	2	FJRE0
			2	FPKT0			16	FJWB0
			20	FRAM1			6	FKMS0
							10	FPAS0
							4	FPKT0
							6	FRAM1
Männliche Sprecher								
			4	MDAB0	4	MREB0	8	MCCS0
			2	MPGL0	2	MRJO0	12	MDAB0
			2	MRGG0	4	MSJS1	16	MGWT0
			2	MSJS1			2	MMDB1
							14	MPGL0
							6	MREB0
							2	MRGG0
							6	MRJO0
							6	MSJS1
Auswertung								
gpw [%]	100.00		89.36		87.18		56.10	
gpm [%]	0.00		10.64		12.82		43.90	

Tabelle A.1.: Falsch zugewiesene Segmente der Kategorien 1 und 2

	Kat. 3 - 20 Spr. Komplett		Kat. 3 - 40 Spr. Komplett		Kat. 3 - 20 Spr. Angepasst		Kat. 3 - 40 Spr. Angepasst	
Anteil	Häufigk.	Name	Häufigk.	Name	Häufigk.	Name	Häufigk.	Name
Weibliche Sprecher								
	14	FAKS0	16	FAKS0	14	FAKS0	14	FAKS0
	2	FDAC1	14	FCMH0	2	FDAC1	4	FCMH0
	12	FDRD1	4	FDAC1	20	FJAS0	10	FDAC1
	18	FJAS0	16	FDRD1	10	FJEM0	8	FDRD1
	18	FJEM0	18	FJAS0	4	FJRE0	2	FELC0
	2	FJRE0	20	FJEM0	16	FJWB0	20	FJAS0
	16	FJWB0	8	FJRE0	18	FPAS0	20	FJEM0
	18	FPAS0	20	FJWB0	20	FRAM1	20	FJWB0
	20	FRAM1	16	FKMS0			16	FKMS0
			16	FPAS0			20	FPAS0
			14	FPKT0			12	FPKT0
			20	FRAM1			20	FRAM1
			2	FSLB1				
Männliche Sprecher								
	4	MREB0	4	MABW0			6	MABW0
	4	MSJS1	12	MBJK0			4	MBJK0
			2	MCCS0			6	MCCS0
			6	MCEM0			16	MCEM0
			20	MDAB0			20	MDAB0
			2	MDLD0			2	MGWT0
			2	MGWT0			4	MJAR0
			14	MJAR0			2	MMDB1
			4	MMDB1			16	MPGL0
			4	MMDM2			4	MRCZ0
			8	MPGL0			8	MRGG0
			6	MRCZ0			2	MRJO0
			6	MREB0			20	MSTK0
			10	MRGG0			2	MTAS1
			8	MSJS1			10	MTMR0
			16	MSTK0			2	MWEW0
			6	MTAS1				
			12	MTMR0				
Auswertung								
gpw [%]	93.75		56.44		100.00		57.24	
gpm [%]	6.25		43.56		0.00		42.76	

Tabelle A.2.: Falsch zugewiesene Segmente der Kategorie 3

	Position x (Zeitachse)	Position y (Frequenzachse)	Ausdehnung w (Zeitachse)	Ausdehnung h (Frequenzachse)
Filter 1	1	22	11	2
Filter 2	1	18	11	2
Filter 3	1	23	11	2
Filter 4	1	17	11	4
Filter 5	1	19	11	4
Filter 6	1	20	11	4
Filter 7	1	21	11	2
Filter 8	1	15	11	2
Filter 9	1	2	11	2
Filter 10	1	16	11	2
Filter 11	1	3	11	2
Filter 12	10	17	2	4
Filter 13	1	1	11	2
Filter 14	1	20	11	2
Filter 15	11	17	1	4
Filter 16	1	17	1	4
Filter 17	1	17	11	8
Filter 18	1	16	11	4
Filter 19	1	21	11	4
Filter 20	1	17	5	4
Filter 21	1	17	3	4
Filter 22	7	17	5	4
Filter 23	7	19	5	4
Filter 24	9	17	3	4
Filter 25	1	17	2	4
Filter 26	1	19	11	2
Filter 27	1	17	1	8
Filter 28	1	4	11	2
Filter 29	1	17	11	2
Filter 30	1	19	11	6

Tabelle A.3.: Positionierung und Ausdehnung der selektierten Filter

	prec [%]		rec [%]		DER [%]	
	20 Sprecher	40 Sprecher	20 Sprecher	40 Sprecher	20 Sprecher	40 Sprecher
Kategorie 1	97.68	89.10	96.05	87.24	3.95	12.76
Kategorie 2	76.64	87.37	73.80	83.05	26.20	16.95
Kategorie 3 Komplett	69.72	21.66	66.33	20.85	33.67	79.19
Kategorie 3 Angepasst	81.37	51.53	76.76	49.59	23.24	50.41

Tabelle A.4.: Übersicht der Resultate aller Experimentkategorien

A.2.4. Darstellung des kompletten Filtersets

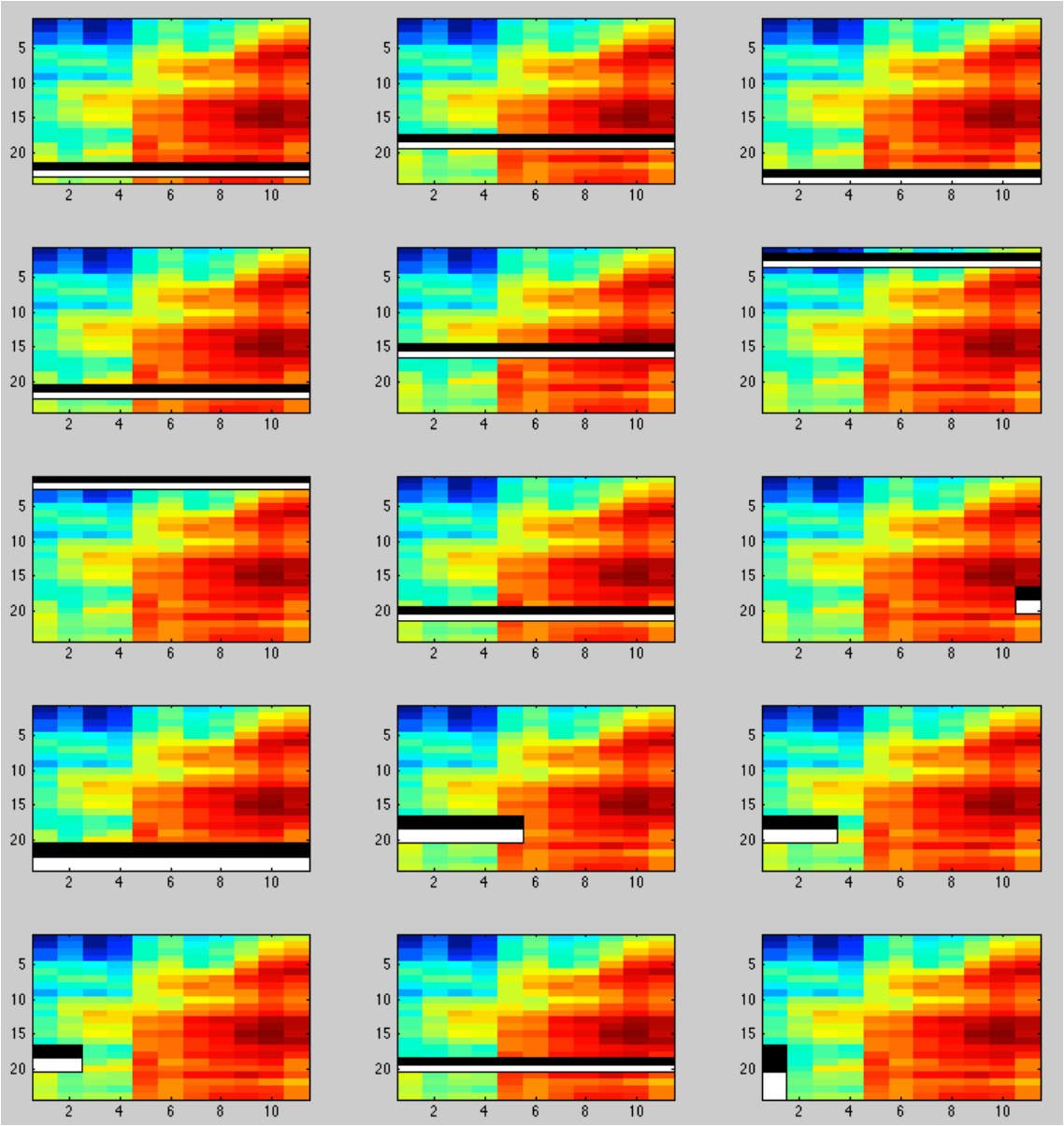

Abbildung A.1.: Erste Hälfte der 30 Filter

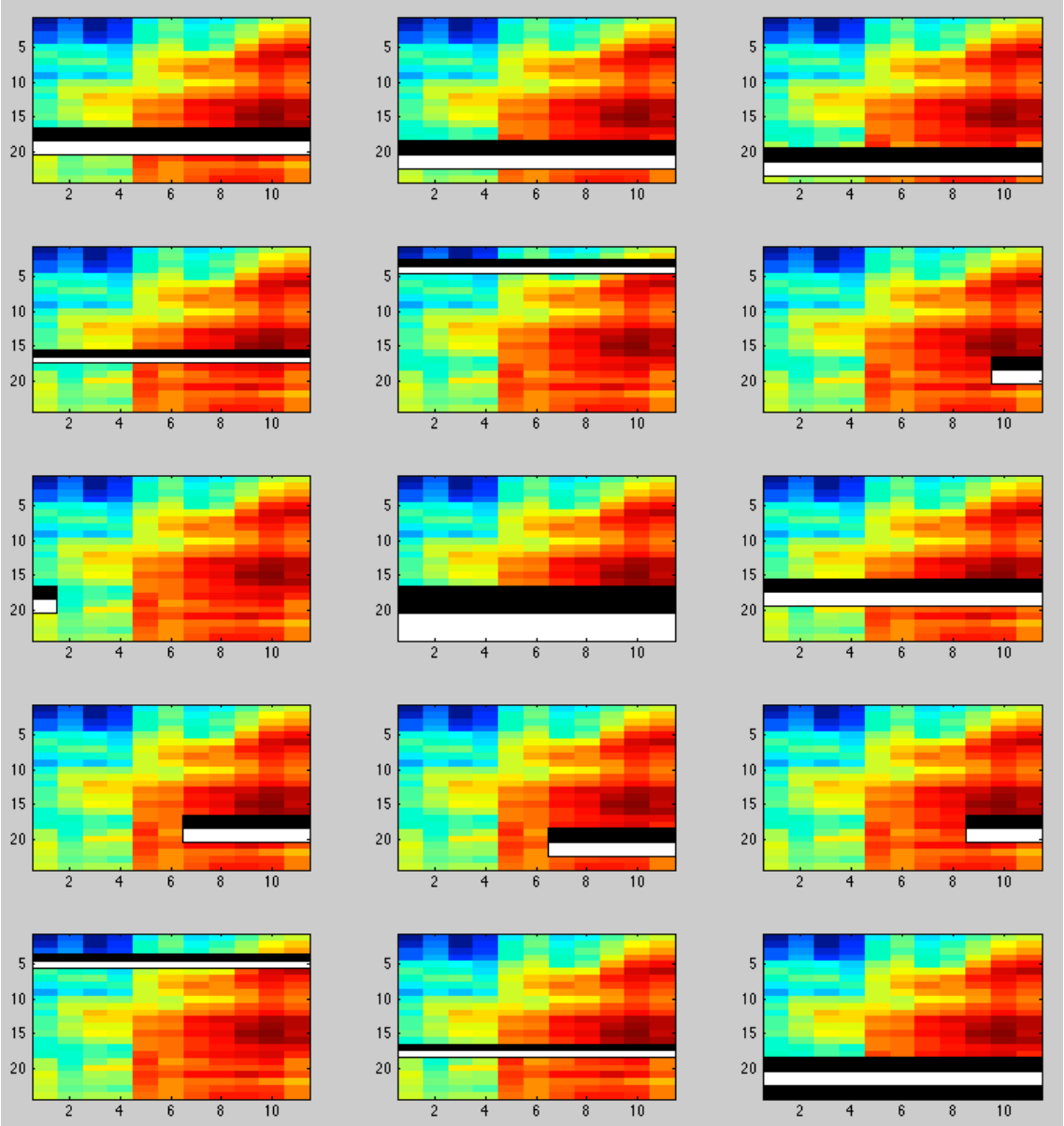

Abbildung A.2.: Zweite Hälfte der 30 Filter

A.2.5. Sprecherset 1: 20 Sprecher

TEST/DR1/FAKS0/SA1.WAV
TEST/DR1/FAKS0/SA2.WAV
TEST/DR1/FAKS0/SI1573.WAV
TEST/DR1/FAKS0/SI2203.WAV
TEST/DR1/FAKS0/SI943.WAV
TEST/DR1/FAKS0/SX133.WAV
TEST/DR1/FAKS0/SX223.WAV
TEST/DR1/FAKS0/SX313.WAV
TEST/DR1/FDAC1/SA1.WAV
TEST/DR1/FDAC1/SA2.WAV
TEST/DR1/FDAC1/SI1474.WAV
TEST/DR1/FDAC1/SI2104.WAV
TEST/DR1/FDAC1/SI844.WAV
TEST/DR1/FDAC1/SX124.WAV
TEST/DR1/FDAC1/SX214.WAV
TEST/DR1/FDAC1/SX304.WAV
TEST/DR1/FELC0/SA1.WAV
TEST/DR1/FELC0/SA2.WAV
TEST/DR1/FELC0/SI1386.WAV
TEST/DR1/FELC0/SI2016.WAV
TEST/DR1/FELC0/SI756.WAV
TEST/DR1/FELC0/SX126.WAV
TEST/DR1/FELC0/SX216.WAV
TEST/DR1/FELC0/SX306.WAV
TEST/DR1/FJEM0/SA1.WAV
TEST/DR1/FJEM0/SA2.WAV
TEST/DR1/FJEM0/SI1264.WAV
TEST/DR1/FJEM0/SI1894.WAV
TEST/DR1/FJEM0/SI634.WAV
TEST/DR1/FJEM0/SX184.WAV
TEST/DR1/FJEM0/SX274.WAV
TEST/DR1/FJEM0/SX364.WAV
TEST/DR1/MDAB0/SA1.WAV
TEST/DR1/MDAB0/SA2.WAV
TEST/DR1/MDAB0/SI1039.WAV
TEST/DR1/MDAB0/SI1669.WAV
TEST/DR1/MDAB0/SI2299.WAV
TEST/DR1/MDAB0/SX139.WAV
TEST/DR1/MDAB0/SX229.WAV
TEST/DR1/MDAB0/SX319.WAV
TEST/DR1/MJSW0/SA1.WAV
TEST/DR1/MJSW0/SA2.WAV
TEST/DR1/MJSW0/SI1010.WAV
TEST/DR1/MJSW0/SI1640.WAV
TEST/DR1/MJSW0/SI2270.WAV
TEST/DR1/MJSW0/SX110.WAV
TEST/DR1/MJSW0/SX20.WAV
TEST/DR1/MJSW0/SX200.WAV
TEST/DR1/MREB0/SA1.WAV
TEST/DR1/MREB0/SA2.WAV
TEST/DR1/MREB0/SI1375.WAV
TEST/DR1/MREB0/SI2005.WAV
TEST/DR1/MREB0/SI745.WAV
TEST/DR1/MREB0/SX115.WAV
TEST/DR1/MREB0/SX205.WAV
TEST/DR1/MREB0/SX25.WAV
TEST/DR1/MRJO0/SA1.WAV
TEST/DR1/MRJO0/SA2.WAV
TEST/DR1/MRJO0/SI1364.WAV
TEST/DR1/MRJO0/SI1624.WAV
TEST/DR1/MRJO0/SI734.WAV
TEST/DR1/MRJO0/SX104.WAV
TEST/DR1/MRJO0/SX14.WAV
TEST/DR1/MRJO0/SX194.WAV
TEST/DR1/MSJS1/SA1.WAV
TEST/DR1/MSJS1/SA2.WAV
TEST/DR1/MSJS1/SI1899.WAV
TEST/DR1/MSJS1/SI639.WAV
TEST/DR1/MSJS1/SI869.WAV
TEST/DR1/MSJS1/SX189.WAV
TEST/DR1/MSJS1/SX279.WAV
TEST/DR1/MSJS1/SX369.WAV
TEST/DR1/MSTK0/SA1.WAV
TEST/DR1/MSTK0/SA2.WAV
TEST/DR1/MSTK0/SI1024.WAV
TEST/DR1/MSTK0/SI2222.WAV
TEST/DR1/MSTK0/SI2284.WAV
TEST/DR1/MSTK0/SX124.WAV
TEST/DR1/MSTK0/SX214.WAV
TEST/DR1/MSTK0/SX304.WAV
TEST/DR1/MWBT0/SA1.WAV
TEST/DR1/MWBT0/SA2.WAV
TEST/DR1/MWBT0/SI1553.WAV
TEST/DR1/MWBT0/SI2183.WAV
TEST/DR1/MWBT0/SI923.WAV
TEST/DR1/MWBT0/SX113.WAV
TEST/DR1/MWBT0/SX203.WAV
TEST/DR1/MWBT0/SX23.WAV
TEST/DR2/FCMR0/SA1.WAV
TEST/DR2/FCMR0/SA2.WAV
TEST/DR2/FCMR0/SI1105.WAV
TEST/DR2/FCMR0/SI1735.WAV
TEST/DR2/FCMR0/SI475.WAV
TEST/DR2/FCMR0/SX115.WAV
TEST/DR2/FCMR0/SX205.WAV
TEST/DR2/FCMR0/SX25.WAV
TEST/DR2/FDRD1/SA1.WAV
TEST/DR2/FDRD1/SA2.WAV
TEST/DR2/FDRD1/SI1544.WAV
TEST/DR2/FDRD1/SI1566.WAV
TEST/DR2/FDRD1/SI2149.WAV
TEST/DR2/FDRD1/SX104.WAV
TEST/DR2/FDRD1/SX14.WAV
TEST/DR2/FDRD1/SX194.WAV
TEST/DR2/FJAS0/SA1.WAV
TEST/DR2/FJAS0/SA2.WAV
TEST/DR2/FJAS0/SI1400.WAV
TEST/DR2/FJAS0/SI2030.WAV

TEST/DR2/FJAS0/SI770.WAV
TEST/DR2/FJAS0/SX140.WAV
TEST/DR2/FJAS0/SX230.WAV
TEST/DR2/FJAS0/SX320.WAV
TEST/DR2/FJRE0/SA1.WAV
TEST/DR2/FJRE0/SA2.WAV
TEST/DR2/FJRE0/SI1116.WAV
TEST/DR2/FJRE0/SI1587.WAV
TEST/DR2/FJRE0/SI1746.WAV
TEST/DR2/FJRE0/SX126.WAV
TEST/DR2/FJRE0/SX216.WAV
TEST/DR2/FJRE0/SX306.WAV
TEST/DR2/FJWB0/SA1.WAV
TEST/DR2/FJWB0/SA2.WAV
TEST/DR2/FJWB0/SI1265.WAV
TEST/DR2/FJWB0/SI635.WAV
TEST/DR2/FJWB0/SI992.WAV
TEST/DR2/FJWB0/SX185.WAV
TEST/DR2/FJWB0/SX275.WAV
TEST/DR2/FJWB0/SX365.WAV
TEST/DR2/FPAS0/SA1.WAV
TEST/DR2/FPAS0/SA2.WAV
TEST/DR2/FPAS0/SI1272.WAV
TEST/DR2/FPAS0/SI2204.WAV
TEST/DR2/FPAS0/SI944.WAV
TEST/DR2/FPAS0/SX134.WAV
TEST/DR2/FPAS0/SX224.WAV
TEST/DR2/FPAS0/SX314.WAV
TEST/DR2/FRAM1/SA1.WAV
TEST/DR2/FRAM1/SA2.WAV
TEST/DR2/FRAM1/SI1360.WAV
TEST/DR2/FRAM1/SI522.WAV
TEST/DR2/FRAM1/SI730.WAV
TEST/DR2/FRAM1/SX10.WAV
TEST/DR2/FRAM1/SX100.WAV
TEST/DR2/FRAM1/SX190.WAV
TEST/DR2/FSLB1/SA1.WAV
TEST/DR2/FSLB1/SA2.WAV
TEST/DR2/FSLB1/SI1904.WAV
TEST/DR2/FSLB1/SI644.WAV
TEST/DR2/FSLB1/SI891.WAV
TEST/DR2/FSLB1/SX104.WAV
TEST/DR2/FSLB1/SX14.WAV
TEST/DR2/FSLB1/SX194.WAV
TEST/DR2/MABW0/SA1.WAV
TEST/DR2/MABW0/SA2.WAV
TEST/DR2/MABW0/SI1230.WAV
TEST/DR2/MABW0/SI1664.WAV
TEST/DR2/MABW0/SI2294.WAV
TEST/DR2/MABW0/SX134.WAV
TEST/DR2/MABW0/SX224.WAV
TEST/DR2/MABW0/SX314.WAV
TEST/DR1/FAKS0/SX403.WAV
TEST/DR1/FAKS0/SX43.WAV
TEST/DR1/FDAC1/SX34.WAV
TEST/DR1/FDAC1/SX394.WAV
TEST/DR1/FELC0/SX36.WAV
TEST/DR1/FELC0/SX396.WAV
TEST/DR1/FJEM0/SX4.WAV
TEST/DR1/FJEM0/SX94.WAV
TEST/DR1/MDAB0/SX409.WAV
TEST/DR1/MDAB0/SX49.WAV
TEST/DR1/MJSW0/SX290.WAV
TEST/DR1/MJSW0/SX380.WAV
TEST/DR1/MREB0/SX295.WAV
TEST/DR1/MREB0/SX385.WAV
TEST/DR1/MRJO0/SX284.WAV
TEST/DR1/MRJO0/SX374.WAV
TEST/DR1/MSJS1/SX9.WAV
TEST/DR1/MSJS1/SX99.WAV
TEST/DR1/MSTK0/SX34.WAV
TEST/DR1/MSTK0/SX394.WAV
TEST/DR1/MWBT0/SX293.WAV
TEST/DR1/MWBT0/SX383.WAV
TEST/DR2/FCMR0/SX295.WAV
TEST/DR2/FCMR0/SX385.WAV
TEST/DR2/FDRD1/SX284.WAV
TEST/DR2/FDRD1/SX374.WAV
TEST/DR2/FJAS0/SX410.WAV
TEST/DR2/FJAS0/SX50.WAV
TEST/DR2/FJRE0/SX36.WAV
TEST/DR2/FJRE0/SX396.WAV
TEST/DR2/FJWB0/SX5.WAV
TEST/DR2/FJWB0/SX95.WAV
TEST/DR2/FPAS0/SX404.WAV
TEST/DR2/FPAS0/SX44.WAV
TEST/DR2/FRAM1/SX280.WAV
TEST/DR2/FRAM1/SX370.WAV
TEST/DR2/FSLB1/SX284.WAV
TEST/DR2/FSLB1/SX374.WAV
TEST/DR2/MABW0/SX404.WAV
TEST/DR2/MABW0/SX44.WAV

A.2.6. Sprecherset 2: 40 Sprecher

TEST/DR1/FAKS0/SA1.WAV
TEST/DR1/FAKS0/SA2.WAV
TEST/DR1/FAKS0/SI1573.WAV
TEST/DR1/FAKS0/SI2203.WAV
TEST/DR1/FAKS0/SI943.WAV
TEST/DR1/FAKS0/SX133.WAV
TEST/DR1/FAKS0/SX223.WAV
TEST/DR1/FAKS0/SX313.WAV
TEST/DR1/FDAC1/SA1.WAV
TEST/DR1/FDAC1/SA2.WAV
TEST/DR1/FDAC1/SI1474.WAV
TEST/DR1/FDAC1/SI2104.WAV

TEST/DR1/FDAC1/SI844.WAV
TEST/DR1/FDAC1/SX124.WAV
TEST/DR1/FDAC1/SX214.WAV
TEST/DR1/FDAC1/SX304.WAV
TEST/DR1/FELC0/SA1.WAV
TEST/DR1/FELC0/SA2.WAV
TEST/DR1/FELC0/SI1386.WAV
TEST/DR1/FELC0/SI2016.WAV
TEST/DR1/FELC0/SI756.WAV
TEST/DR1/FELC0/SX126.WAV
TEST/DR1/FELC0/SX216.WAV
TEST/DR1/FELC0/SX306.WAV
TEST/DR1/FJEM0/SA1.WAV
TEST/DR1/FJEM0/SA2.WAV
TEST/DR1/FJEM0/SI1264.WAV
TEST/DR1/FJEM0/SI1894.WAV
TEST/DR1/FJEM0/SI634.WAV
TEST/DR1/FJEM0/SX184.WAV
TEST/DR1/FJEM0/SX274.WAV
TEST/DR1/FJEM0/SX364.WAV
TEST/DR1/MDAB0/SA1.WAV
TEST/DR1/MDAB0/SA2.WAV
TEST/DR1/MDAB0/SI1039.WAV
TEST/DR1/MDAB0/SI1669.WAV
TEST/DR1/MDAB0/SI2299.WAV
TEST/DR1/MDAB0/SX139.WAV
TEST/DR1/MDAB0/SX229.WAV
TEST/DR1/MDAB0/SX319.WAV
TEST/DR1/MJSW0/SA1.WAV
TEST/DR1/MJSW0/SA2.WAV
TEST/DR1/MJSW0/SI1010.WAV
TEST/DR1/MJSW0/SI1640.WAV
TEST/DR1/MJSW0/SI2270.WAV
TEST/DR1/MJSW0/SX110.WAV
TEST/DR1/MJSW0/SX20.WAV
TEST/DR1/MJSW0/SX200.WAV
TEST/DR1/MREB0/SA1.WAV
TEST/DR1/MREB0/SA2.WAV
TEST/DR1/MREB0/SI1375.WAV
TEST/DR1/MREB0/SI2005.WAV
TEST/DR1/MREB0/SI745.WAV
TEST/DR1/MREB0/SX115.WAV
TEST/DR1/MREB0/SX205.WAV
TEST/DR1/MREB0/SX25.WAV
TEST/DR1/MRJO0/SA1.WAV
TEST/DR1/MRJO0/SA2.WAV
TEST/DR1/MRJO0/SI1364.WAV
TEST/DR1/MRJO0/SI1624.WAV
TEST/DR1/MRJO0/SI734.WAV
TEST/DR1/MRJO0/SX104.WAV
TEST/DR1/MRJO0/SX14.WAV
TEST/DR1/MRJO0/SX194.WAV
TEST/DR1/MSJS1/SA1.WAV
TEST/DR1/MSJS1/SA2.WAV
TEST/DR1/MSJS1/SI1899.WAV
TEST/DR1/MSJS1/SI639.WAV
TEST/DR1/MSJS1/SI869.WAV
TEST/DR1/MSJS1/SX189.WAV
TEST/DR1/MSJS1/SX279.WAV
TEST/DR1/MSJS1/SX369.WAV
TEST/DR1/MSTK0/SA1.WAV
TEST/DR1/MSTK0/SA2.WAV
TEST/DR1/MSTK0/SI1024.WAV
TEST/DR1/MSTK0/SI2222.WAV
TEST/DR1/MSTK0/SI2284.WAV
TEST/DR1/MSTK0/SX124.WAV
TEST/DR1/MSTK0/SX214.WAV
TEST/DR1/MSTK0/SX304.WAV
TEST/DR1/MWBT0/SA1.WAV
TEST/DR1/MWBT0/SA2.WAV
TEST/DR1/MWBT0/SI1553.WAV
TEST/DR1/MWBT0/SI2183.WAV
TEST/DR1/MWBT0/SI923.WAV
TEST/DR1/MWBT0/SX113.WAV
TEST/DR1/MWBT0/SX203.WAV
TEST/DR1/MWBT0/SX23.WAV
TEST/DR2/FCMR0/SA1.WAV
TEST/DR2/FCMR0/SA2.WAV
TEST/DR2/FCMR0/SI1105.WAV
TEST/DR2/FCMR0/SI1735.WAV
TEST/DR2/FCMR0/SI475.WAV
TEST/DR2/FCMR0/SX115.WAV
TEST/DR2/FCMR0/SX205.WAV
TEST/DR2/FCMR0/SX25.WAV
TEST/DR2/FDRD1/SA1.WAV
TEST/DR2/FDRD1/SA2.WAV
TEST/DR2/FDRD1/SI1544.WAV
TEST/DR2/FDRD1/SI1566.WAV
TEST/DR2/FDRD1/SI2149.WAV
TEST/DR2/FDRD1/SX104.WAV
TEST/DR2/FDRD1/SX14.WAV
TEST/DR2/FDRD1/SX194.WAV
TEST/DR2/FJAS0/SA1.WAV
TEST/DR2/FJAS0/SA2.WAV
TEST/DR2/FJAS0/SI1400.WAV
TEST/DR2/FJAS0/SI2030.WAV
TEST/DR2/FJAS0/SI770.WAV
TEST/DR2/FJAS0/SX140.WAV
TEST/DR2/FJAS0/SX230.WAV
TEST/DR2/FJAS0/SX320.WAV
TEST/DR2/FJRE0/SA1.WAV
TEST/DR2/FJRE0/SA2.WAV
TEST/DR2/FJRE0/SI1116.WAV
TEST/DR2/FJRE0/SI1587.WAV
TEST/DR2/FJRE0/SI1746.WAV
TEST/DR2/FJRE0/SX126.WAV
TEST/DR2/FJRE0/SX216.WAV
TEST/DR2/FJRE0/SX306.WAV
TEST/DR2/FJWB0/SA1.WAV
TEST/DR2/FJWB0/SA2.WAV
TEST/DR2/FJWB0/SI1265.WAV
TEST/DR2/FJWB0/SI635.WAV

TEST/DR2/FJWB0/SI992.WAV
TEST/DR2/FJWB0/SX185.WAV
TEST/DR2/FJWB0/SX275.WAV
TEST/DR2/FJWB0/SX365.WAV
TEST/DR2/FPAS0/SA1.WAV
TEST/DR2/FPAS0/SA2.WAV
TEST/DR2/FPAS0/SI1272.WAV
TEST/DR2/FPAS0/SI2204.WAV
TEST/DR2/FPAS0/SI944.WAV
TEST/DR2/FPAS0/SX134.WAV
TEST/DR2/FPAS0/SX224.WAV
TEST/DR2/FPAS0/SX314.WAV
TEST/DR2/FRAM1/SA1.WAV
TEST/DR2/FRAM1/SA2.WAV
TEST/DR2/FRAM1/SI1360.WAV
TEST/DR2/FRAM1/SI522.WAV
TEST/DR2/FRAM1/SI730.WAV
TEST/DR2/FRAM1/SX10.WAV
TEST/DR2/FRAM1/SX100.WAV
TEST/DR2/FRAM1/SX190.WAV
TEST/DR2/FSLB1/SA1.WAV
TEST/DR2/FSLB1/SA2.WAV
TEST/DR2/FSLB1/SI1904.WAV
TEST/DR2/FSLB1/SI644.WAV
TEST/DR2/FSLB1/SI891.WAV
TEST/DR2/FSLB1/SX104.WAV
TEST/DR2/FSLB1/SX14.WAV
TEST/DR2/FSLB1/SX194.WAV
TEST/DR2/MABW0/SA1.WAV
TEST/DR2/MABW0/SA2.WAV
TEST/DR2/MABW0/SI1230.WAV
TEST/DR2/MABW0/SI1664.WAV
TEST/DR2/MABW0/SI2294.WAV
TEST/DR2/MABW0/SX134.WAV
TEST/DR2/MABW0/SX224.WAV
TEST/DR2/MABW0/SX314.WAV
TEST/DR2/MBJK0/SA1.WAV
TEST/DR2/MBJK0/SA2.WAV
TEST/DR2/MBJK0/SI1175.WAV
TEST/DR2/MBJK0/SI2128.WAV
TEST/DR2/MBJK0/SI545.WAV
TEST/DR2/MBJK0/SX185.WAV
TEST/DR2/MBJK0/SX275.WAV
TEST/DR2/MBJK0/SX365.WAV
TEST/DR2/MCCS0/SA1.WAV
TEST/DR2/MCCS0/SA2.WAV
TEST/DR2/MCCS0/SI1469.WAV
TEST/DR2/MCCS0/SI2099.WAV
TEST/DR2/MCCS0/SI839.WAV
TEST/DR2/MCCS0/SX119.WAV
TEST/DR2/MCCS0/SX209.WAV
TEST/DR2/MCCS0/SX29.WAV
TEST/DR2/MCEM0/SA1.WAV
TEST/DR2/MCEM0/SA2.WAV
TEST/DR2/MCEM0/SI1398.WAV
TEST/DR2/MCEM0/SI2028.WAV
TEST/DR2/MCEM0/SI768.WAV
TEST/DR2/MCEM0/SX138.WAV
TEST/DR2/MCEM0/SX228.WAV
TEST/DR2/MCEM0/SX318.WAV
TEST/DR2/MDBB0/SA1.WAV
TEST/DR2/MDBB0/SA2.WAV
TEST/DR2/MDBB0/SI1195.WAV
TEST/DR2/MDBB0/SI1825.WAV
TEST/DR2/MDBB0/SI565.WAV
TEST/DR2/MDBB0/SX115.WAV
TEST/DR2/MDBB0/SX205.WAV
TEST/DR2/MDBB0/SX25.WAV
TEST/DR2/MDLD0/SA1.WAV
TEST/DR2/MDLD0/SA2.WAV
TEST/DR2/MDLD0/SI1543.WAV
TEST/DR2/MDLD0/SI2173.WAV
TEST/DR2/MDLD0/SI913.WAV
TEST/DR2/MDLD0/SX103.WAV
TEST/DR2/MDLD0/SX13.WAV
TEST/DR2/MDLD0/SX193.WAV
TEST/DR2/MGWT0/SA1.WAV
TEST/DR2/MGWT0/SA2.WAV
TEST/DR2/MGWT0/SI1539.WAV
TEST/DR2/MGWT0/SI2169.WAV
TEST/DR2/MGWT0/SI909.WAV
TEST/DR2/MGWT0/SX189.WAV
TEST/DR2/MGWT0/SX279.WAV
TEST/DR2/MGWT0/SX369.WAV
TEST/DR2/MJAR0/SA1.WAV
TEST/DR2/MJAR0/SA2.WAV
TEST/DR2/MJAR0/SI1988.WAV
TEST/DR2/MJAR0/SI2247.WAV
TEST/DR2/MJAR0/SI728.WAV
TEST/DR2/MJAR0/SX188.WAV
TEST/DR2/MJAR0/SX278.WAV
TEST/DR2/MJAR0/SX368.WAV
TEST/DR2/MMDB1/SA1.WAV
TEST/DR2/MMDB1/SA2.WAV
TEST/DR2/MMDB1/SI1625.WAV
TEST/DR2/MMDB1/SI2255.WAV
TEST/DR2/MMDB1/SI995.WAV
TEST/DR2/MMDB1/SX185.WAV
TEST/DR2/MMDB1/SX275.WAV
TEST/DR2/MMDB1/SX365.WAV
TEST/DR2/MMDM2/SA1.WAV
TEST/DR2/MMDM2/SA2.WAV
TEST/DR2/MMDM2/SI1452.WAV
TEST/DR2/MMDM2/SI1555.WAV
TEST/DR2/MMDM2/SI2082.WAV
TEST/DR2/MMDM2/SX102.WAV
TEST/DR2/MMDM2/SX12.WAV
TEST/DR2/MMDM2/SX192.WAV
TEST/DR2/MPDF0/SA1.WAV
TEST/DR2/MPDF0/SA2.WAV
TEST/DR2/MPDF0/SI1542.WAV
TEST/DR2/MPDF0/SI2172.WAV

TEST/DR2/MPDF0/SI912.WAV
TEST/DR2/MPDF0/SX12.WAV
TEST/DR2/MPGL0/SA1.WAV
TEST/DR2/MPGL0/SI1099.WAV
TEST/DR2/MPGL0/SI469.WAV
TEST/DR2/MPGL0/SX19.WAV
TEST/DR2/MRCZ0/SA1.WAV
TEST/DR2/MRCZ0/SI1541.WAV
TEST/DR2/MRCZ0/SI911.WAV
TEST/DR2/MRCZ0/SX11.WAV
TEST/DR2/MRGG0/SA1.WAV
TEST/DR2/MRGG0/SI1199.WAV
TEST/DR2/MRGG0/SI569.WAV
TEST/DR2/MRGG0/SX209.WAV
TEST/DR2/MTAS1/SA1.WAV
TEST/DR2/MTAS1/SI1473.WAV
TEST/DR2/MTAS1/SI838.WAV
TEST/DR2/MTAS1/SX208.WAV
TEST/DR2/MTMR0/SA1.WAV
TEST/DR2/MTMR0/SI1303.WAV
TEST/DR2/MTMR0/SI673.WAV
TEST/DR2/MTMR0/SX223.WAV
TEST/DR2/MWEW0/SA1.WAV
TEST/DR2/MWEW0/SI1361.WAV
TEST/DR2/MWEW0/SI731.WAV
TEST/DR2/MWEW0/SX11.WAV
TEST/DR2/MWVW0/SA1.WAV
TEST/DR2/MWVW0/SI1476.WAV
TEST/DR2/MWVW0/SI846.WAV
TEST/DR2/MWVW0/SX216.WAV
TEST/DR3/FCMH0/SA1.WAV
TEST/DR3/FCMH0/SI1454.WAV
TEST/DR3/FCMH0/SI824.WAV
TEST/DR3/FCMH0/SX14.WAV
TEST/DR3/FKMS0/SA1.WAV
TEST/DR3/FKMS0/SI1490.WAV
TEST/DR3/FKMS0/SI860.WAV
TEST/DR3/FKMS0/SX230.WAV
TEST/DR3/FPKT0/SA1.WAV
TEST/DR3/FPKT0/SI1538.WAV
TEST/DR3/FPKT0/SI908.WAV
TEST/DR3/FPKT0/SX278.WAV
TEST/DR1/FAKS0/SX403.WAV
TEST/DR1/FDAC1/SX34.WAV
TEST/DR1/FELC0/SX36.WAV
TEST/DR1/FJEM0/SX4.WAV
TEST/DR1/MDAB0/SX409.WAV
TEST/DR1/MJSW0/SX290.WAV
TEST/DR1/MREB0/SX295.WAV
TEST/DR1/MRJO0/SX284.WAV
TEST/DR1/MSJS1/SX9.WAV
TEST/DR1/MSTK0/SX34.WAV
TEST/DR1/MWBT0/SX293.WAV
TEST/DR2/FCMR0/SX295.WAV
TEST/DR2/FDRD1/SX284.WAV
TEST/DR2/FJAS0/SX410.WAV
TEST/DR2/MPDF0/SX102.WAV
TEST/DR2/MPDF0/SX192.WAV
TEST/DR2/MPGL0/SA2.WAV
TEST/DR2/MPGL0/SI1729.WAV
TEST/DR2/MPGL0/SX109.WAV
TEST/DR2/MPGL0/SX199.WAV
TEST/DR2/MRCZ0/SA2.WAV
TEST/DR2/MRCZ0/SI2171.WAV
TEST/DR2/MRCZ0/SX101.WAV
TEST/DR2/MRCZ0/SX191.WAV
TEST/DR2/MRGG0/SA2.WAV
TEST/DR2/MRGG0/SI1829.WAV
TEST/DR2/MRGG0/SX119.WAV
TEST/DR2/MRGG0/SX29.WAV
TEST/DR2/MTAS1/SA2.WAV
TEST/DR2/MTAS1/SI2098.WAV
TEST/DR2/MTAS1/SX118.WAV
TEST/DR2/MTAS1/SX28.WAV
TEST/DR2/MTMR0/SA2.WAV
TEST/DR2/MTMR0/SI1933.WAV
TEST/DR2/MTMR0/SX133.WAV
TEST/DR2/MTMR0/SX313.WAV
TEST/DR2/MWEW0/SA2.WAV
TEST/DR2/MWEW0/SI1991.WAV
TEST/DR2/MWEW0/SX101.WAV
TEST/DR2/MWEW0/SX191.WAV
TEST/DR2/MWVW0/SA2.WAV
TEST/DR2/MWVW0/SI2106.WAV
TEST/DR2/MWVW0/SX126.WAV
TEST/DR2/MWVW0/SX306.WAV
TEST/DR3/FCMH0/SA2.WAV
TEST/DR3/FCMH0/SI2084.WAV
TEST/DR3/FCMH0/SX104.WAV
TEST/DR3/FCMH0/SX194.WAV
TEST/DR3/FKMS0/SA2.WAV
TEST/DR3/FKMS0/SI2120.WAV
TEST/DR3/FKMS0/SX140.WAV
TEST/DR3/FKMS0/SX320.WAV
TEST/DR3/FPKT0/SA2.WAV
TEST/DR3/FPKT0/SI2168.WAV
TEST/DR3/FPKT0/SX188.WAV
TEST/DR3/FPKT0/SX368.WAV
TEST/DR1/FAKS0/SX43.WAV
TEST/DR1/FDAC1/SX394.WAV
TEST/DR1/FELC0/SX396.WAV
TEST/DR1/FJEM0/SX94.WAV
TEST/DR1/MDAB0/SX49.WAV
TEST/DR1/MJSW0/SX380.WAV
TEST/DR1/MREB0/SX385.WAV
TEST/DR1/MRJO0/SX374.WAV
TEST/DR1/MSJS1/SX99.WAV
TEST/DR1/MSTK0/SX394.WAV
TEST/DR1/MWBT0/SX383.WAV
TEST/DR2/FCMR0/SX385.WAV
TEST/DR2/FDRD1/SX374.WAV
TEST/DR2/FJAS0/SX50.WAV

TEST/DR2/FJRE0/SX36.WAV
TEST/DR2/FJWB0/SX5.WAV
TEST/DR2/FPAS0/SX404.WAV
TEST/DR2/FRAM1/SX280.WAV
TEST/DR2/FSLB1/SX284.WAV
TEST/DR2/MABW0/SX404.WAV
TEST/DR2/MBJK0/SX5.WAV
TEST/DR2/MCCS0/SX299.WAV
TEST/DR2/MCEM0/SX408.WAV
TEST/DR2/MDBB0/SX295.WAV
TEST/DR2/MDLD0/SX283.WAV
TEST/DR2/MGWT0/SX9.WAV
TEST/DR2/MJAR0/SX8.WAV
TEST/DR2/MMDB1/SX5.WAV
TEST/DR2/MMDM2/SX282.WAV
TEST/DR2/MPDF0/SX282.WAV
TEST/DR2/MPGL0/SX289.WAV
TEST/DR2/MRCZ0/SX281.WAV
TEST/DR2/MRGG0/SX299.WAV
TEST/DR2/MTAS1/SX298.WAV
TEST/DR2/MTMR0/SX403.WAV
TEST/DR2/MWEW0/SX281.WAV
TEST/DR2/MWVW0/SX36.WAV
TEST/DR3/FCMH0/SX284.WAV
TEST/DR3/FKMS0/SX410.WAV
TEST/DR3/FPKT0/SX8.WAV

TEST/DR2/FJRE0/SX396.WAV
TEST/DR2/FJWB0/SX95.WAV
TEST/DR2/FPAS0/SX44.WAV
TEST/DR2/FRAM1/SX370.WAV
TEST/DR2/FSLB1/SX374.WAV
TEST/DR2/MABW0/SX44.WAV
TEST/DR2/MBJK0/SX95.WAV
TEST/DR2/MCCS0/SX389.WAV
TEST/DR2/MCEM0/SX48.WAV
TEST/DR2/MDBB0/SX385.WAV
TEST/DR2/MDLD0/SX373.WAV
TEST/DR2/MGWT0/SX99.WAV
TEST/DR2/MJAR0/SX98.WAV
TEST/DR2/MMDB1/SX95.WAV
TEST/DR2/MMDM2/SX372.WAV
TEST/DR2/MPDF0/SX372.WAV
TEST/DR2/MPGL0/SX379.WAV
TEST/DR2/MRCZ0/SX371.WAV
TEST/DR2/MRGG0/SX389.WAV
TEST/DR2/MTAS1/SX388.WAV
TEST/DR2/MTMR0/SX43.WAV
TEST/DR2/MWEW0/SX371.WAV
TEST/DR2/MWVW0/SX396.WAV
TEST/DR3/FCMH0/SX374.WAV
TEST/DR3/FKMS0/SX50.WAV
TEST/DR3/FPKT0/SX98.WAV

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] N. Balaska, Z. Ahmida, and A. Goutas. Speaker recognition using artificial neural networks: Rbfnns vs. ebfnns. Bericht, Skikda: University 20 Août 55, LRES Lab., 2007.

[2] H. S. M. Beigi, S. H. Maes, and J. S. Sorensen. A distance measure between collections of distributions and its application to speaker recognition. Bericht, New York: IBM Research, T.J. Watson Center, Human Language Technologie Group, 1998.

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, Bd. 2, Nr. 1, S. 1-127, 2009.

[4] S. Cohen. Finding color and shape patterns in images. Doktorarbeit, Stanford: Stanford University, Department of Computer Science, 1999.

[5] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Bd. 28, Nr. 4, S. 357-366, Aug. 1980.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography. Bericht, Kalifornien: SRI International, Artificial Intelligence Center, 1981.

[7] Y. Freud and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, Bd. 55, Nr. SS971504, S. 119-139, 1997.

[8] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Bd. 34, Nr. 1, S. 52-59, 1986.

[9] P. Giannopoulos and R. C. Veltkamp. A pseudo-metric for weighted point sets. Bericht, Utrecht: University of Utrecht, Institute of Information and Computing Sciences, 2002.

[10] S. Guruprasad, N. Dhananjaya, and B. Yegnanarayana. Aann models for speaker recognition based on diffrerence cepstrals. Bericht, Chennai: Indian Institute of Technology, Speech and Vision Laboratory, 2003.

[11] C. Joder, S. Essid, and G. Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE Transactions on Audio, Speech, and Language Processing*, Bd. 17, Nr. 1, S. 174-186, Jan. 2009.

[12] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. Bericht, Pittsburgh: Carnegie Mellon University, Intel Research, 2005.

[13] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork rnn. Bericht, Manno-Lugano: IDSIA, USI, SUPSI, 2014.

[14] MathWorks. Fitted ensemble for classification or regression [online]. URL: http://www.mathworks.ch/ch/help/stats/fitensemble.html [20.05.2014].

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, Bd. 77, Nr. 2, S. 257-286, Feb. 1989.

[16] G. D. Rey and K. F. Wender. *Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Hans Huber, Okt. 2010. ISBN 978-3-456-84513-5.

[17] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Audio, Speech, and Language Processing*, Bd. 3, Nr. 1, S. 72-83, Jan. 1995.

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Learning deep architectures for ai. *Digital Signal Processing*, Bd. 10, S. 19-41, 2000.

[19] R. E. Schapire. Explaining adaboost. Bericht, Princeton: Princeton University, Dept. of Computer Science, 2013.

[20] T. Stadelmann. Voice modeling methods for automatic speaker recognition. Doktorarbeit, Marburg: Philipps-Universität, Apr. 2010.

[21] T. Stadelmann and B. Freisleben. Unfolding speaker clustering potential: A biomimetic approach. *Proceedings of the 17th ACM international conference on Multimedia*, S. 185-194, 2009.

[22] T. Stadelmann and B. Freisleben. Dimension-decoupled gaussian mixture model for short utterance speaker recognition. *20th International Conference on Pattern Recognition, ICPR 2010*, S. 1602-1605, 2010.

[23] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. Tech. rep. uu-cs-2003-024, Utrecht: University of Utrecht, Institute of Information and Computing Sciences, 2003.

[24] Author unbekannt. International association for forensic phonetics and acoustics [online]. URL: http://www.iafpa.net/ [29.05.2014], .

[25] Author unbekannt. Mel frequency cepstral coefficient (mfcc) tutorial [online]. URL: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs [22.05.2014], .

[26] P. Viola and M. J. Jones. Robust real-time object detection. Bericht, Camebridge: Mitsubishi Electric Research Labs, Camebridge / Compaq Cambridge Research Laboratory (CRL), 2001.

[27] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, Bd. 57, Nr. 2, S. 137-154, 2004.

[28] VoiceTrust. Voicetrust - voice biometrics solutions [online]. URL: http://www.voicetrust.com/solutions/ [29.05.2014].

[29] J. Ye. Speech recognition using time domain features from phase space reconstructions. Masterarbeit, Milwaukee, Wisconsin: Marquette University, Mai 2004.

[30] H. Yu. Phase-space representation of speech, revisiting the delta and double delta features. Bericht, Pittsburgh: Carnegie Mellon University, Interactive Systems Labs, 2003.

[31] N. Zheng. Speaker recognition using complementary information from vocal source and vocal tract. Doktorarbeit, The Chinese University of Hong Kong, 2005.

[32] J. Zhu, S. Sun, X. Liu, and B. Lei. Pitch in speaker recognition. *Ninth International Conference on Hybrid Intelligent Systems, 2009*, Bd. 1, S. 33-36, Aug 2009.