TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
Die Ressourcenuniversität. Seit 1765.

Fakultät für Mathematik und Informatik
Institut für Informatik
Lehrstuhl für Künstliche Intelligenz und Datenbanken

Masterarbeit

Emotionserkennung anhand der menschlichen Stimme mit rekurrenten neuronalen Netzen

Emotion recognition using the human voice with recurrent neural networks

Samuel Dressel

Angewandte Informatik
Vertiefung: Künstliche Intelligenz

Matrikel: 59963

7. Oktober 2020

Betreuer/1. Korrektor:
Prof. Dr. Heinrich Jasper

2. Korrektor:
M. Sc. Volker Göhler

Eidesstattliche Erklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, habe ich in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Diese Versicherung bezieht sich auch auf die bildlichen Darstellungen.

Samuel Dressel

Freiberg, den 7. Oktober 2020

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und begleitet haben. Dabei sei zunächst mein 2. Korrektor Volker Göhler genannt, der viel Zeit in eine intensive Betreuung investiert hat und mir mit konstruktiven Anregungen zur Seite stand. Desweiteren möchte ich meinem Betreuer, Prof. H. Jasper, für die unkomplizierte Hilfe bei der Organisation meiner Arbeit danken.
Außerdem danke ich meiner Familie und meiner Verlobten Anna, die mich während des Studiums sehr unterstützt und mir den Rücken freigehalten haben. Abschließend gilt mein Dank Gott, dessen Führung und Segen ich in den letzten Monaten und Jahren besonders spüren durfte.

Samuel Dressel

Freiberg, den 7. Oktober 2020

Inhaltsverzeichnis

1. Einleitung und Motivation

In mehreren wissenschaftlichen Studien wurde nachgewiesen, dass jeder Mensch am Tag durchschnittlich 60.000 verschiedene Gedanken hat [1], [2]. Für jeden dieser Gedanken besteht die Möglichkeit, dass selbiger eine Emotion hervorgeruft. Damit sind Emotionen eines der wichtigsten alltäglichen Bestandteile des menschlichen Erlebens. Dadurch, dass Emotionen nicht nur als Gedankenkonstrukt im menschlichen Gehirn existieren, sondern sich durch Mimik und Gestik auch äußerlich zeigen, bilden sie außerdem eine wichtige Komponente der Kommunikation und menschlichen Interaktion untereinander. Je nach gezeigter Emotion wird dem jeweiligen Gegenüber die Möglichkeit gegeben, entsprechend und angemessen zu reagieren. Diese Reaktion ist ausschlaggebend dafür, ob menschliches Zusammenleben gelingt.
Im Zuge der Digitalisierung aller Lebensbereiche ist es nicht nur für menschliche Wesen wichtig, den emotionalen Zustand eines Kommunikationspartners zu erkennen. Vielmehr bieten sich mehr und mehr Schnittpunkte für digitale Anwendungen, welche Emotionen in einen Entscheidungsprozess mit einbeziehen. Beispiele hierfür sind unter anderem die Marktforschung, ein individuelles Marketing oder der Einsatz in der Forensik (vgl. [3], [4]).
Zur Erkennung menschlicher Emotionen gibt es ein breites Spektrum an unterschiedlichen Ansätzen. Viele davon klassifizieren Emotionen anhand von Merkmalen mit künstlichen neuronalen Netzen. Diese Merkmale können die Mimik [4], die Herzfrequenz [5] oder auch die menschliche Stimme sein [6].

Den Ansatz, Emotionen anhand der menschlichen Stimme zu erkennen, verfolgt auch diese Arbeit. Dabei wird zur Klassifikation ein rekurrentes neuronales Netz verwendet, welches anhand bestimmter, aus der Stimme extrahierter Merkmale eine Entscheidung zugunsten oder gegen eine Emotion trifft. Die Emotionsmenge beinhaltet dabei die Emotionen Freude, Wut, Trauer, Angst, Langeweile und Ekel sowie die Emotionsneutralität.

Im nachfolgenden Kapitel 2 wird zunächst ein kleiner Überblick über Forschungsarbeiten gegeben, welche sich mit dem gleichen Problemfeld der Emotionserkennung anhand der Stimme beschäftigen. In Kapitel 3 wird danach zunächst auf den Grundbegriff der Emotion eingegangen und erörtert, inwieweit sich Emotionen anhand der menschlichen Stimme bestimmen lassen. Danach werden in Kapitel 4 verschiedene mathematische Grundlagen betrachtet, welche für die in Kapitel 5 vorgestellten Merkmale eines Audio-Signals benötigt werden. Weiterhin werden dann in Kapitel 6 fundamentale Begriffe und Eigenschaften des verwendeten künstlichen neuronalen Netzes erklärt.
Danach folgt in Kapitel 7 ein Überblick über den Ansatz und die Durchführung der im Rahmen dieser Arbeit durchgeführten Untersuchungen. Die generierten Ergebnisse werden dann in Kapitel 8 vorgestellt und ausgewertet. Letztlich findet sich in Kapitel 9 eine Zusammenfassung dieser Arbeit und ein Ausblick über mögliche weitergehende Forschungsansätze.

Alle im Rahmen dieser Arbeit generierten Daten und implementierten Python-Skripte finden sich in dem öffentlich zugänglichen GitHub-Repository unter folgendem Link:

`https://github.com/Samykolon/Master`

2. Ähnliche Arbeiten

Wie eingangs erwähnt, existieren eine Vielzahl an Forschungsarbeiten, die sich mit dem gleichen Problemfeld beschäftigt haben. Dabei gibt es neben wiederkehrenden Ansätzen immer wieder neue Ideen, um der Emotionserkennung anhand der menschlichen Stimme gerecht zu werden. In Tabelle 1 wird ein grober Überblick über vier solcher Forschungsarbeiten gegeben und auf deren Ansatz und die jeweils erreichte Validierungs-Genauigkeit eingegangen. Diese steht dabei als Maß für die Effizienz der jeweiligen Methode zur Emotionserkennung.

Forschungsarbeit	**Anzahl Features**	**Klassifizierungsmethode**	**Emotionsklassen**	**Validierungs-Genauigkeit**
EmoNets: Multimodal deep learning approaches for emotion recognition in video - 2015 [7]	29	Deep Belief Network (DBN)	7	34,2 Prozent
Emotion Recognition From Speech With Recurrent Neural Networks - 2017 [8]	34	Recurrent Neural Network (RNN)	4	54,0 Prozent
Emotion Recognition from Speech - 2019 [6]	13	Convolutional Neural Network (SVM)	7	70,0 Prozent
Machine Learning Approach for Emotion Recognition in Speech - 2014 [9]	300	Support Vector Machine (SVM)	7	86,0 Prozent

Tab. 1: Überblick ähnliche Forschungsarbeiten und deren Ergebnisse.

Wie die tabellarische Darstellung zeigt, gibt es im Fall der Emotionserkennung anhand der menschlichen Stimme in allen Aspekten der Methodik wesentliche Unterschiede.
In ihrem Paper *EmoNets: Multimodal deep learning approaches for emotion recognition in video* stellen S.E. Kahou et al. eine Methode vor, um mithilfe eines *Deep Belief Networks (DBN)* menschliche Emotionen anhand der Stimme und des Gesichtsausdrucks zu erkennen [7]. Auch wenn diese Arbeit im Jahr 2014 veröffentlich wurde, so ist das verwendete neuronale Netz zu diesem Zeitpunkt schon nicht mehr State of the Art. Eine Validierungs-Genauigkeit von 34,2 Prozent spiegelt diese Tatsache teilweise wieder.

Im Gegensatz dazu wurde durch V. Chernykh et al. die Eignung eines rekurrenten neuronalen Netzes [8] als Klassifizierungsmethode zur Emotionserkennung untersucht. Hierbei konnte eine Genauigkeit von 54 Prozent erreicht werden, jedoch nur bei einer Einteilung in vier verschiedene Emotionsklassen.

Die dritte Art von neuronalen Netz, welche man für dieses Problemfeld findet, sind *Convolutional Neural Networks (CNN)* [6]. Ein solches Faltungsnetzwerk wird z.B. von K. Venkataramanan und H. R. Rajamohan verwendet, um anhand von 13 Features eine Entscheidung zwischen sieben Emotionen mit einer Genauigkeit von 70 Prozent zu treffen.

Weiterhin gibt es auch noch den Ansatz, *Support Vector Machines (SVM)* für diese Problemstellung zu nutzen. Dabei erreichten M. und H. Gjoreski in ihrer Forschung eine Validierungs-Genauigkeit von 86 Prozent bei der Verwendung von 300 Features [9].

Neben den vorgestellten Methoden gibt es noch zahlreiche andere Forschungsarbeiten, welche u.a. *Random Forests* oder *Hidden Markov Models* für die Emotionserkennung verwenden. Die effizienteste und meist genutzte Methodik zum Zeitpunkt der Veröffentlichung dieser Arbeit beinhaltet jedoch die Verwendung von rekurrenten neuronalen Netzen bzw. Faltungsnetzen.

3. Psychophysiologische Grundlagen

Grundvoraussetzung für diese Arbeit ist die Tatsache, dass sich Emotionen aus der menschlichen Stimme und deren Klang ableiten lassen. Dieses Kapitel beschäftigt sich mit den Grundlagen der Auswirkungen von Emotionen auf die menschliche Stimme. Dazu wird in Kapitel 3.1 zunächst auf den Begriff der Emotion eingegangen. Danach wird in Kapitel 3.2 erörtert, inwieweit sich Emotionen anhand des Klangs der menschlichen Stimme ableiten lassen. Zuletzt wird in Kapitel 3.3 auf mögliche Ansätze zur Klassifizierung von Emotionen eingegangen und charakterisiert, welche Probleme sich dabei ergeben.

3.1. Der Begriff der Emotion

Das Wort Emotion stammt von dem lateinischen Wort *emovere* ab und bedeutet soviel wie „heraus bewegen, in Bewegung setzen, in einen erregten Zustand versetzen.“ [10]. Eine Emotion beschreibt somit eine Art Gemütsbewegung im Sinne eines Affektes [11]. Dabei ist diese Affektivität (Gefühlscharakter) gleichzeitig auch das auffälligste Merkmal einer Emotion.
Ein zweites Merkmal von Emotionen ist ihre Objektgerichtetheit (Intentionalität). Emotionen sind immer auf „etwas“ ausgerichtet. Dabei ist es unwesentlich, ob das Bezugsobjekt tatsächlich vorliegt oder nur als ein Gedankenkonstrukt existiert.
Desweiteren ist die Unwillkürlichkeit von Emotionen das dritte Merkmal. Emotionen sind automatisch ausgelöste Reaktionen auf bestimmte Situationen und Einschätzungen, denen man sich nicht entziehen kann. Auch wenn durch die Möglichkeit der Emotionsregulation bestimmten Emotionen strategisch begegnet werden kann, so kann die Auslösung einer Emotion letztendlich nicht vollständig kontrolliert werden [12].
Das letzte Merkmal von Emotionen ist ihre zeitlich begrenzte Dauer, die mehr oder weniger eng an das emotionale Bezugsobjekt gekoppelt ist.

Aus diesen Merkmalen lässt sich folgende Definition ableiten [10, S. 166]:

> Emotionen sind objektgerichtete, unwillkürlich ausgelöste affektive Reaktionen, die mit zeitlich befristeten Veränderungen des Erlebens und Verhaltens einhergehen.

Emotionen sind unbedingt von globalen Stimmungslagen abzugrenzen, die als diffuse positive und negative Gefühlszustände kein eindeutiges Bezugsobjekt haben und zeitlich länger andauern. Dasselbe gilt für emotionale Dispositionen (Temperamente), die als zeitlich stabile Persönlichkeitseigenschaften einen sehr allgemeinen Objektbezug besitzen. Die Abgrenzung dieser drei Begriffe wird durch die Abbildung 1 auf Seite 9 veranschaulicht.

3.2. Emotionalität der menschlichen Stimme

Grundlage für den Erfolg der in dieser Arbeit durchgeführten Untersuchungen ist die Tatsache, dass sich anhand der Stimme Emotionen ableiten lassen.
Um den Zusammenhang zwischen Emotion und Stimme zu verstehen, ist zunächst ein Blick auf die Erzeugung der Stimme notwendig. Die Erzeugung der menschlichen Stimme setzt sich aus zwei separaten mechanischen Prozessen zusammen. Dies ist unabhängig davon, ob im Sinne einer Sprache kommunziert wird oder nur Laute abgegeben werden [13]. Im ersten Prozess, der sogenannten *Phonation*, wird beim Ausatmen durch das Pressen von Luft an die sich im Kehlkopf befindlichen Stimmlippen ein einfacher Klang erzeugt.

Abb. 1: Abgrenzung der Gefühlsbegriffe Emotion, Stimmung und Temperament

Diese Stimmlippen werden durch Muskeln in ihrer Form und dem Abstand zueinander verändert. Gleichzeitig ist das schnelle Öffnen und Schließen der Stimmlippen die Grundlage für die Entwicklung von Sprache. Nachdem der grundlegende Klang der Stimme erzeugt wurde, wird dieser im Prozess der *Artikulation* angepasst. Wesentlich für diesen Teil der Stimmerzeugung sind die Lippen, die Zunge und das Gaumensegel im Rachenraum. Gleichzeitig dient der Nasenraum, der Rachen und die Brusthöhle als Resonanzkörper für die Verstärkung des artikulierten Klangs. Auch diese Organe und Bestandteile des oberen menschlichen Körpers werden durch Muskeln und Nerven gesteuert.
In Abbildung 2 auf Seite 10 findet sich eine Übersicht über die beteiligten Organe zur Stimmerzeugung, ihre Interaktion untereinander und ihren Einfluss auf bestimmte Parameter der Stimme. Dabei wird deutlich, dass der *Neokortex* nicht nur auf dem direkten Weg über das zentrale Nervensystem die einzelnen an der Stimmerzeugung beteiligten Organe und deren Muskeln ansteuert. Vielmehr gibt es auch eine Interaktion mit dem *Limbischen System* und dem autonomen Nervensystem (kurz ANS), welche unter anderem das emotionale Verhalten des Menschen steuert. Das autonome Nervensystem als Teil des zentralen Nervensystems koordiniert die zugehörigen Vorgänge „autonom“ , d.h. ohne dass diese Vorgänge vollständig vom Menschen beeinflusst werden können. Zu diesen Vorgängen gehören unter anderem die Atmung oder das Schlagen des Herzes [15].
Das autonome Nervensystem wird in das sympathische (Sympathikus) und das parasympathische Nervensystem (Parasympathikus) geteilt. Beide Teile agieren in Verbindung miteinander, um die größten Teile des unterbewussten menschlichen Verhaltens zu steuern, sowohl in Normal- als auch in Stresssituationen. Dabei bewirkt der Sympathikus eine Leistungssteigerung, während der Parasympathikus eine Leistungsminderung und Beruhigung des Organismus bewirkt. Beide Teile des ANS steuern also auch direkt oder indirekt die an der Stimmerzeugung beteiligten Muskeln und damit auch den Klang der Stimme.

Diese Funktionalität des menschlichen Organismus ermöglicht es, Emotionen anhand der Stimme zu erkennen. Da das ANS nur bedingt bewusst gesteuert werden kann, ist es deshalb auch nur eingeschränkt möglich, Emotionen vollständig zu regulieren. Sicherlich kann man die bewussten körperlichen Reaktionen auf eine bestimmte Emotion minimieren, doch im Unterbewusstsein bleibt diese doch bestehen. Dadurch stellt der Klang der Stimme eine zuverlässigere Quelle für Informationen über den emotionalen Zustand eines Menschen da, als der Inhalt der durch die Stimme erzeugten Sprache.

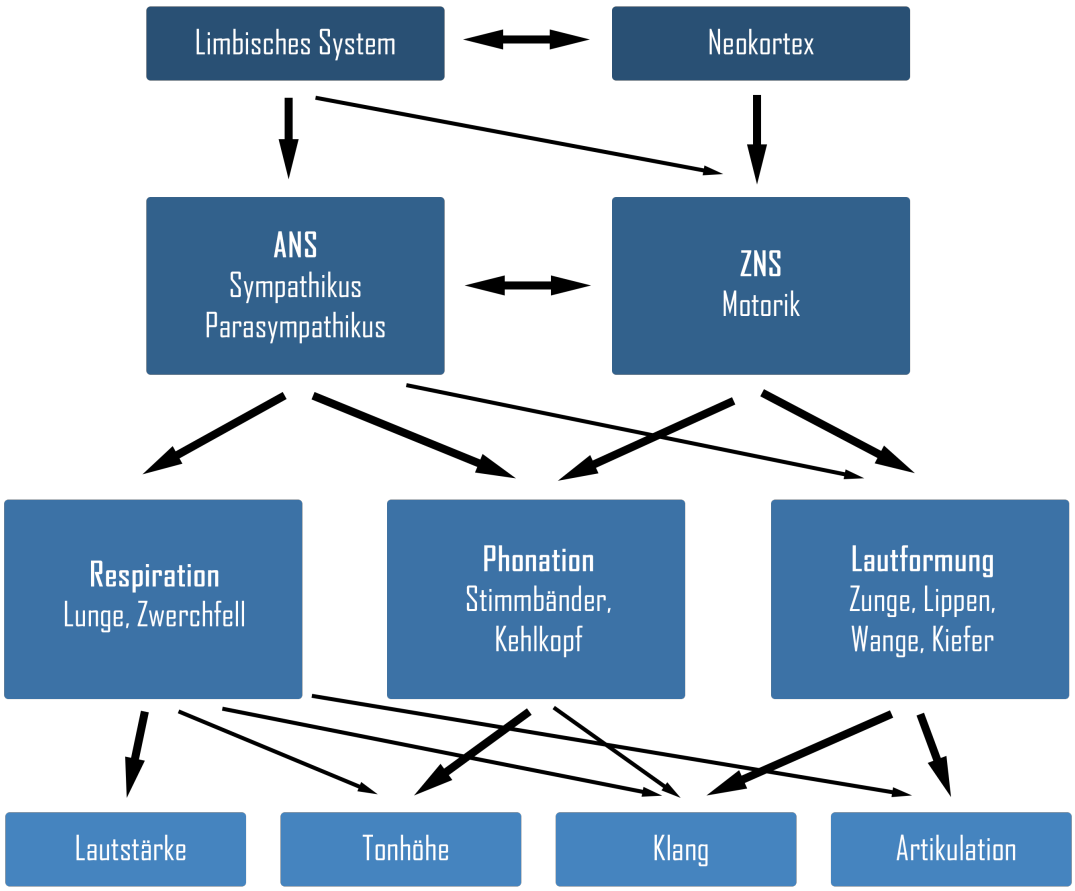


Abb. 2: Modell der Steuerung der Stimmerzeugung nach [14]. Die Pfeildicke repräsentiert dabei die hypothetische Stärke des Einflusses. Die Abkürzung ANS steht für das **a**utonome **N**ervensystem, die Abkürzung ZNS für das **z**entrale **N**ervensystem

Forscher der University of California kamen dabei zu dem Ergebnis, dass sich durch den Klang der menschlichen Stimme 24 verschiedene Emotionen ableiten lassen [16]. Trotz kleiner Unterschieden ist diese Tatsache kulturell unabhängig, auch wenn in jeder Kultur Emotionen unterschiedlich stark gezeigt werden. Diese Korrelation zwischen unterschiedlichen Sprachen und Kulturen wurde u. a. ausführlich durch die im Januar 2001 veröffentlichte Arbeit von K. Scherer et al. nachgewiesen [17].

3.3. Modellierung von Emotionen

Für die Modellierung und Klassifizierung von menschlichen Emotionen gibt es unterschiedliche Ansätze, welche Emotionen in unterschiedlich dimenensionierte Modelle einordnen.

Ein triviales Modell dabei ist die Einordnung in sechs verschiedene Basisemotionen: Freude, Trauer, Wut, Angst, Überraschung und Ekel. Dieses Modell wurde 1971 durch Paul Ekman erörtert [18]. Dabei stellte Ekman heraus, dass diese Emotionen sich auch bei blindgeborenen Menschen durch die jeweils typischen Gesichtsausdrücke äußerten. Nach Carrol E. Izard existieren zehn kulturunabhängige Formen von Emotionen.

Dabei ergänzt er die oben genannte Liste um die Emotionen Schuldgefühl, Schamgefühl, Interesse und Widerwillen [19].
Ein weiterer bekannter Vertreter für die Modellierung von Emotionen ist James A. Russell, welcher Emotionen in ein zweidimensionalen Modell mit den orthogonalen Skalengrößen Valenz und Arousal einteilt [20]. Dieses Modell gilt bis heute als eines der Kernmodelle für Emotion in der Psychophysiologie.
Nicht zuletzt wurde bereits 1874 von Wilhelm Wundt eine dreidimensionales Modellierung der verschiedenen Emotionen vorgenommen [21]. Dabei wurden alle Emotionen auf die drei Skalen Arousal, Valenz und Angespanntheit eingeteilt.
In vielen Arbeiten, welche sich mit der maschinellen Erkennung von Emotionen beschäftigen, wird ein bestimmtes Muster einer bestimmten Emotion zugeordnet. Dies trifft auch auf die jeweiligen Datensätze zu, die für das Training bei der Emotionserkennung durch neuronale Netze verwendet werden. Es liegt in diesen Fällen dadurch eine eindimensionale Modellierung vor. Bezüglich der Zuordnung zu einem Skalenniveau gehört diese Modellierung der vorliegenden Emotionen zur Nominalskala. Diese nominale Einteilung von Emotionen hat aber auch Nachteile. Forschungen von K. R. Scherer und G. Ceschi aus dem Jahr 1997 zeigen, dass im realen Umfeld nur selten ein bestimmtes Verhaltensmuster zu 100 Prozent einer bestimmten Emotion zugeordnet werden kann [22]. In ihrem Paper *The Ordinal Nature of Emotions: An Emerging Approach* gehen Georgios N. Yannakakis und Carlos Busso auf diese Erkenntnis ein [23]. Sie verdeutlichen, dass „Emotionen von Natur aus ordinal sind“, also das Skalenniveau bezüglich des Vorhandenseins einer Emotion ordinal ist. Es wird also eine Rangfolge vorgenommen, welche besagt dass eine Emotion stärker oder weniger stark ist als eine andere. Im Bereich des maschinellen Lernens ist dieser Ansatz im Grunde genommen schon seit Beginn unbewusst verwendet worden. Bei dem Prognoseprozess nach erfolgreichem Trainierens eines Modells, welcher einen bestimmten Input einer Kategorie zuordnen soll, werden nur Wahrscheinlichkeiten für das Vorliegen einer bestimmten Kategorie zurückgegeben. Aus diesen Wahrscheinlichkeiten lässt sich eine Rangfolge hinsichtlich der Stärke der Emotionen ableiten.

In dieser Arbeit wurde sich aufgrund der vorliegenden Datensätze für eine eindimensionale Modellierung von Emotionen entschieden. Verwendet wurden sieben Kategorien, bei denen sechs Kategorien aus den Emotionen Freude, Trauer, Wut, Langeweile, Ekel und Angst bestehen. Die letzte Kategorie bildet die Emotionsneutralität.

4. Mathematische Grundlagen

Dieses Kapitel soll die mathematischen Grundlagen beleuchten, welche für die im nächsten Kapitel erläuterten Audio-Features von Bedeutung sind.

Audiosignale stellen im Regelfall eine komplexe Mischung von verschiedenen Soundkomponenten dar. Aus diesem Grund ist es sehr schwierig, bestimmte Merkmale eines Audiosignals aus dessen Wellenform zu extrahieren.
Ein Ansatz, um diesem Problem gerecht zu werden, ist die Aufteilung des Signales in mehrere sinusförmige Blöcke. Die Eigenschaft der Sinusförmigkeit bietet sich aufgrund des engen Zusammenhangs der Sinusfunktion mit der physikalischen Größe der Frequenz an (vgl. [24]). Aus dieser Umwandlung geht eine Darstellung des Frequenzspektrums eines Signals hervor. Diesen Prozess, der ein zeitabhängiges Signal in ein frequenzabhängiges Signal umwandelt, nennt man *Fourier-Transformation*. Das Wissen über die verschiedenen Frequenzen, die in einem Audiosignal vorkommen, ist fundamental für die Generierung von aussagekräftigen Features.

Im folgenden Abschnitt 4.1 werden die mathematischen Grundlagen der Fourier-Transformation dargestellt. In Abschnitt 4.2 wird die *Diskrete Fourier-Transformation* näher erläutert, welche mit der *schnellen Fourier-Transformation* als Berechnungsalgorithmus die Grundlage für die meisten in dieser Arbeit verwendeten Audio-Features bildet. Danach wird in Abschnitt 4.3 auf die *Kurzzeit-Fourier-Transformation* eingangen, bei welcher das Eingangssignal in mehrere Fenster eingeteilt wird und für diese das Frequenzspektrum berechnet wird. Letztlich wird in Abschnitt 4.4 die *diskrete Kosinus-Transformation* (DCT) definiert, welche für die Extraktion der MFCC-Features von Bedeutung ist.

4.1. Fourier-Transformation

Ein Signal in Form einer T-periodischen Zeitfunktion $f_p(t)$ lässt sich als *Fourierreihe* mit den *Fourierkoeffizienten* a_k und b_k darstellen [25]:

$$f_p(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \big[a_k \cdot cos(k \cdot t) + b_k \cdot sin(k \cdot t)\big] \tag{1}$$

Der Begriff *T-periodisch* ist für eine Funktion $f(t)$ dann erfüllt, wenn für alle Zeitpunkte t des Definitionsbereichs gilt:

$$f(t + T) = f(t) \tag{2}$$

Diese Funktion $f_p(t)$ muss dabei die sogenannten *Dirichletbedingungen* erfüllen.

Eine T-periodische Funktion $f(t)$ genügt den Dirichletbedinungen, wenn

1. $f(t)$ beschränkt ist,
2. $f(t)$ im Intervall $\big[0, T\big]$ höchstens endlich viele Unstetigkeitsstellen hat,
3. die Ableitung $f'(t)$ im Intervall $\big[0, T\big]$ bis auf höchstens endlich viele Stellen stetig ist.

Real auftretende Schwingungen sind jedoch meist nicht periodisch.

Eine nicht periodische Funktion $f(t)$ kann als periodische Funktion $f_p(t)$ mit unendlich langer Periodendauer $T_0 \to \infty$ betrachtet werden. Aufgrund dieser Tatsache wird der Abstand zwei benachbarter Frequenzen unendlich klein und damit auch der Abstand zweier diskreter Kreisfrequenzen, wobei die Kreisfrequnz als physikalische Größe für die Schnelligkeit der ablaufenden Schwingung dient. Als Folge dessen geht die diskrete Kreisfrequenz in eine kontinuierliche Kreisfrequenz ω über. Nun wird keine Reihe über die Kreisfrequenzen mehr gebildet, sondern vielmehr ein Integral (vgl. [26]). Dieses Integral wird als *Fourierintegral* bezeichnet:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \tag{3}$$

Die zugehörige Funktion der Kreisfrequenz ω heißt *Spektralfunktion*:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt \tag{4}$$

Letztendlich nennt man Zuordnung von einer Zeitfunktion $f(t)$ zu einer Spektralfunktion $F(\omega)$ *Fourier-Transformation.*

4.2. Diskrete Fourier-Transformation

Bei der Durchführung der Fourier-Transformation wird mit unendlichen Summen und unbegrenzten Integralen gearbeitet; die oben definierten Gleichungen gelten nur für zeitkontinuierliche Signale. Mit Computern können aber nur zeitdiskrete Signale verarbeitet werden. Dies macht die Fourier-Transformation in ihrer klassischen Form für eine rechnergestützte Kalkulation unbrauchbar. Deswegen bedient man sich der *Diskreten Fourier-Transformation* (kurz *DFT*), um den aus durch Abtastung gewonnenen zeitdiskreten Signalen gerecht zu werden. Die Berechnung der dabei entstehenden Fourierkoeffizienten $X(k)$ eines Signals x mit einer endlichen Länge N ist dabei wie folgt definiert (vgl. [24]):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-2\pi \cdot i \cdot k \cdot n}{N}} \tag{5}$$

Eine Diskrete Fourier-Transformation mit N Samples kann auch kurz mit der Gleichung $X = Wx$ abgebildet werden, wobei x das Eingangssignal darstellt, W für die $N \times N$ DFT-Matrix steht und X das Ergebnis der DFT ist. Die Transformationmatrix W ergibt sich dabei wie folgt:

$$W = \left(\frac{\omega^j k}{\sqrt{N}} \right)_{j,k=0,\ldots,N-1} \tag{6}$$

In ihrer klassischen Form besitzt die Diskrete Fourier-Transformation eine Komplexität von $\mathcal{O}(N^2)$. Bei einem Signal mit 10^3 Samples würden daher 10^6 Operationen für eine erfolgreiche Berechnung benötigt werden. Um dieser Komplexität entgegenzuwirken, wurde der Algorithmus der *schnellen Fourier-Transformation* (engl. *Fast Fourier Transformation*, kurz *FFT*) entwickelt. Hierbei wird die DFT-Matrix in ein Produkt von $\mathcal{O}(log\, N)$ dünnbesetzten Matrizen faktorisiert, welche die Komplexität der diskreten Fourier-Transformation auf $\mathcal{O}(N\, log\, N)$ reduziert.

4.3. Kurzzeit-Fourier-Transformation

Bei der oben erläuterten Diskreten Fourier-Transformation wird für ein Signal stets der Durchschnitt der Frequenzinformationen für die gesamte Signallänge zurückgegeben [24].
Für eine Anwendung, bei der Signale mit mehr als einer oder zwei Sekunden Länge verwendet werden, ist dies ungeeignet. Dies hat unter anderem den Grund, dass durch die Berechnung von durchschnittlichen Frequenzinformationen wichtige Daten verloren gehen. Für eine beliebige Signallänge jeweils immer nur die gleiche Anzahl an Informationen erhalten. Lokale Aspekte des Signals würden somit völlig außer Acht gelassen. Für alle in dieser Arbeit verwendeten Features mit der Fourier-Transformation als Grundlage wird deshalb die sogenannte *Kurzzeit-Fourier-Transformation* (engl. *Short-Time Fourier Transformation*, kurz *STFT*) verwendet. Die grundlegende Idee der STFT ist es, nur einen bestimmten Teilbereich des Signals zu betrachten. Dieser Teilbereich hat eine bestimmte Fenstergröße $w(n)$. Im Verlauf der Betrachtung des Gesamtsignals wird dieser Teilbereich um einen Wert m verschoben, für welchen $m < w(n)$ gilt. Mathematisch lässt sich die STFT wie folgt ausdrücken:

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{7}$$

4.4. Diskrete Kosinus-Transformation

Die diskrete Kosinus-Transformation (engl. *Discrete Cosine Transformation*, kurz *DCT*) transformiert ähnlich der diskreten Fourier-Transformation ein zeitdiskretes Signal vom Zeitbereich in den Frequenzbereich um. Bei der diskreten Fourier-Transformation bzw. der Fourier-Transformation im Allgemeinen wird ein komplexwertiges Spektrum erzeugt. Dies erfolgt unabhängig davon, ob das ursprüngliche Signal reelle Werte aufweist oder nicht [27]. Die DCT erzeugt im Gegensatz dazu nur reellwertige Signale und Spektralkoeffizienten. Daher findet diese meist als abschließende Transformation zur Generierung von komprimierten Spektralinformationen Einsatz, da sich die resultierenden Koeffizienten besser speichern und weiterverarbeiten lassen. Die in dieser Arbeit für die Berechnung der Audio-Features verwendete Form ist die *DCT-II*, welche wie folgt definiert ist:

$$X_k = \sum_{n=0}^{N-1} x_n cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \quad k = 0, ..., N-1 \tag{8}$$

5. Audio-Features zur Emotionserkennung

In Kapitel 3 wurde der Begriff der Emotion erklärt und gezeigt, dass sich Emotionen anhand der Stimme ableiten lassen. Um eine rechnerbasierte Aussage über eine vorliegende Emotion zu treffen, müssen bestimmte Merkmale aus dem jeweiligen Audio-Input extrahiert werden, für welche die im vorigen Kapitel 4 vorgestellten Grundlagen von wesentlicher Bedeutung sind. Dieses Kapitel soll sich mit diesen Merkmalen, im Folgenden *Audio-Features* genannt, beschäftigen. Das erste Unterkapitel 5.1 dieses Abschnitts ist dem Überblick und der Auswahl der verschiedenen Audio-Features gewidmet. Im Unterkapitel 5.2 bis 5.4 werden dann die im ersten Abschnitt vorgestellten Feature-Sets im Hinblick auf ihre Berechnung und Beschaffenheit näher erläutert.

5.1. Übersicht über die verwendeten Audio-Features

Im Bereich des Audio-Processings wurde in den letzten Jahren eine Vielzahl von Algorithmen für unterschiedliche Audio-Features entwickelt. Die Herausforderung besteht darin, die aussagekräftigsten Features für das jeweilige Problem auszuwählen. Für das Problemfeld der Emotionserkennung anhand der menschlichen Stimme bieten sich eine große Bandbreite von unterschiedlichen Features an. Dabei ist die Art, die Menge und die Parametrisierung dieser Features in der Literatur nicht klar festgelegt. Zudem wird auch oft keine Auskunft darüber gegeben, warum eine Entscheidung auf die jeweiligen Audio-Features und deren Parametrisierung getroffen wurde. Auch wenn es in Hinsicht auf diese Faktoren eine gemeinsame Schnittmenge gibt, so erfordert diese Vielzahl an unterschiedlichen Ansätzen eine genaue Untersuchung, welche Features sich besser eignen und welche nicht.

In dieser Arbeit wurden drei verschiedene Feature-Sets hinsichtlich ihrer Eignung für die Emotionserkennung näher untersucht. Dies umfasst die Betrachtung unterschiedlicher Parameter für die Berechnung einzelner Features. Hinsichtlich der Wahl der jeweiligen Audio-Features bzw. Feature-Sets wurde sich an oft verwendeten Features in der Literatur orientiert.
Das erste Feature-Set bilden die *Mel-Frequenz-Cepstrum-Koeffizienten* (engl. *Mel Frequency Cepstral Coefficients*, kurz *MFCC*). Diese Koeffizienten beschreiben eine kompakte Darstellung des Frequenzspektrums und geben Auskunft über die wahrgenommene Höhe eines Audiosignals. Die MFCC-Features finden sich in der modernen Forschung zur Emotionserkennung aus einem Audiosignal sehr häufig (u. a. [6], [28] und [29]). Genauere Erklärungen zu diesem Feature-Set sind im nachfolgenden Kapitel 5.2 zu finden.
Das zweite Feature-Set besteht aus einer Mischung acht verschiedener Audio-Features aus dem Zeit- und Frequenzbereich eines Audiosignals. Auf die genaue Auswahl der drei zeitabhängigen und fünf frequenzabhängigen Features, welche dieser Gruppe zugeordnet wurden, wird im zugehörigen Unterkapitel 5.3 näher eingegangen. Auch diese Features wurden in zahlreichen vorangegangenen Arbeiten im Bereich der Emotionserkennung ausgewählt (vgl. [6], [8] und [30]).
Das dritte und letzte Feature-Set, welches in Abschnitt 5.4 erläutert wird, besteht aus 13 *Chroma-Features*, welche Aufschluss über die vorhandene Energie jedes Tons der westlichen Tonskala gibt. Referenzen für die Verwendung dieser Chroma-Features findet sich unter anderem in [8], [6] oder [31].

5.2. MFCC-Features

Die aus der Fouriertransformation entstehenden sog. Cepstra sind eine der meist verwendeten Features im Bereich des Speech-Processings. Die am meisten verbreiteten Audiomerkmale dieses Typs sind die Mel-Frequenz-Cepstrum-Koeffizienten [32]. Der Vorteil dieses Feature-Sets ist, dass es die Charakteristiken des menschlichen Gehörs mit in Betracht zieht [33]. Die vom Menschen wahrgenommenen Frequenzen skalieren logarithmisch und nicht linear. Diese Eigenschaft wird durch die *Mel-Skala* abgebildet. Diese Skala wurde 1937 entwickelt und stellt die wahrgenommene Skala von Tonhöhen dar, welche bei testweise ausgesuchten Zuhörern als gleich im Abstand untereinander empfunden wurde [34].

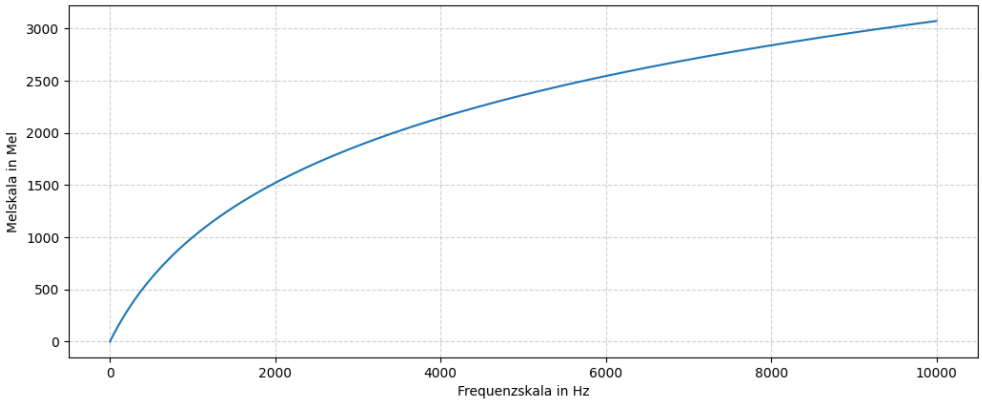


Abb. 3: Mel-Skala in Mel in Abhängigkeit von der Frequenzskala in Hz

Der Graph in Abbildung 3 zeigt den Zusammenhang zwischen Mel-Skala und Frequenzskala.

Um ein Audiosignal auf die Mel-Skala abzubilden und die dazugehörigen Koeffizienten zu extrahieren, sind eine Reihe von Schritten notwendig. Zur Veranschaulichung soll Abbildung 4 dienen, welche einen groben Überblick über diesen Prozess gibt.

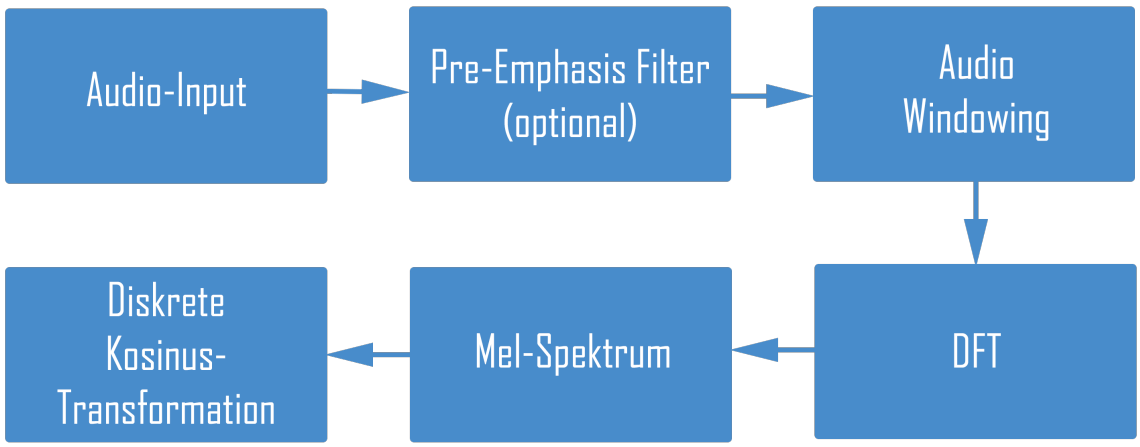


Abb. 4: Übersicht über den Prozess der MFCC-Extraktion. Dieses Schema ist nicht allgemeingültig, da es wie im Fall des Pre-Emphasis Filters optionale Schritte gibt.

Der erste Schritt stellt das Anwenden eines *Pre-Emphasis Filters* da. Dieser Schritt ist optional, da hier lediglich versucht wird, die in einem Audiosignal enthaltene Noise zu reduzieren. Dabei senkt dieser Filter die tiefen Frequenzen ab, während er die hohen Frequenzen anhebt. Dies führt unter Umständen aber auch gleichzeitig zum Verlust von Informationen, da die abgeschwächten Frequenzen auch wichtige Informationen enthalten können. Im zweiten Schritt wird das Signal in mehrere, sich überlappende Fenster geteilt (*Audio Windowing*). Dieses Vorgehen erzeugt zum einen eine Vielfaches von Features, zum anderen werden durch die vermiedene Erzeugung von Durchschnittswerten über das gesamte Audiosignal genauere Informationen erzeugt. Nach dem Splitten des Signals in mehrere Fenster folgt nun die Anwendung der diskreten Fourier-Transformation, welche Informationen über das Frequenzspektrum der einzelnen Fenster erzeugt:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-2\pi \cdot i \cdot k \cdot n}{N}} \quad 0 \leq k \leq N-1 \tag{9}$$

Dabei ist N die Anzahl der Punkte, die für die Berechnung der diskreten Fourier-Transformation verwendet wurde und stellt somit die Auflösung derselben dar. Als nächstes wird das transformierte Signal auf die Mel-Skala abgebildet. Diese Abbildung ergibt sich in der Theorie durch folgende Formel:

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \tag{10}$$

In der Praxis werden sogenannte *Mel-Filterbänke* für diese Abbildung verwendet. Eine Filterbank ist in diesem Fall dabei nichts anderes als eine Auswahl an Dreiecksfiltern, welche das Frequenzband durch Zusammenfassen mehrerer Frequenzen reduzieren. Je nach Größe der Filterbank erhält man nach diesem Schritt unterschiedlich viele Koeffizienten, welche jeweils den Informationsgehalt in einem bestimmten Frequenzbereich widerspiegeln. Da der Frequenzgang der menschlichen Stimme fließend ist, korrelieren die Koeffizienten benachbarter Frequenzbänder. Um diese Informationen über die grundlegende Form des Frequenzspektrums und der Anregungsfrequenz zu erhalten, wird abschließend die diskrete Kosinus-Transformation angewendet. Letztendlich erhält man dadurch die Mel-Frequenz-Cepstrum-Koeffizienten. Mathematisch wird dies mit untenstehender Formel ausgedrückt; wobei $c(n)$ für die Cepstrum-Koeffizienten und C für die Anzahl der MFCCs steht:

$$c(n) = \sum_{m=0}^{M-1} log_{10}(s(m)) cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad n = 0, ..., C-1 \tag{11}$$

Um zusätzlich zu den MFCCs weitere Informationen zu gewinnen, wird in einigen Anwendungsfällen auch die erste und die zweite Ableitung dieser Koeffizienten mit hinzugezogen.

5.3. Zeit- und frequenzabhängige Features

In diesem Abschnitt wird das zweite Feature-Set, welches drei zeitabhängige Features fünf frequenzabhängige Features beinhaltet. Diese Zusammenstellung ergibt sich aus der Python-Bibliothek *pyAudioAnalysis* von Theodoros Giannakopoulos [35]. Wie in dem einleitenden Abschnitt dieses Kapitels erwähnt, nutzen eine Reihe von Forschungsarbeiten im Bereich der Emotionserkennung anhand der menschlichen Stimme diese Bibliothek und die darin enthaltenen Features.

Somit bietet es sich an, diese acht Features als ein Feature-Set zu definieren, um letztendlich eine Aussage über die Effizienz derselben treffen zu können. Vor der Berechnung aller acht Features wird das Audiosignal in mehrere sich überlappende Fenster geteilt. Der Grund dafür ist derselbe wie bei Anwendung dieses Schrittes in der Berechnung der MFCC-Features, da durch diese Vorgehensweise eine höhere Anzahl an Informationen gewonnen wird. Ist also im Folgenden von „Frames“ die Rede, so sind damit jene sich überlappende Fenster gemeint.

Nulldurchgangsrate. Die Nulldurchgangsrate (engl. *zero-crossing rate*) gibt Aufschluss darüber, wie oft ein Audiosignal in einem bestimmten Signalbereich die x-Achse schneidet [36]. Im Bereich des Audio-Processings kann dieses Feature bei der Entscheidung helfen, ob in einem Audio-Signal gesprochen wird oder nicht [37]. Mathematisch ergibt sich die Nulldurchgangsrate ZCR eines Signals s der Länge T folgendermaßen:

$$ZCR = \frac{1}{2T} \sum_{t=1}^{T-1} (s_t \cdot s_{t-1}) \tag{12}$$

Energie. Um Aufschluss über die durchschnittliche Lautstärke in einem Frame des Audiosignals zu erhalten, wird die Energie (oft auch *Energy of Speech* genannt), berechnet. Diese ergibt sich durch Addieren des Quadrats aller Signalwerte in dem jeweiligen Frame:

$$E = \sum_{n=-\infty}^{\infty} x(n)^2 \tag{13}$$

Energie-Entropie. Der Begriff der Entropie steht neben der Verwendung in der Thermodynamik umgangssprachlich für das „Maß an Unordnung“. Dies trifft in gewisser Hinsicht auch für den Gebrauch im Audio-Processing zu. Hier steht die Energie-Entropie für die Stärke von abrupten Änderungen des Energie-Levels bzw. der Lautstärke [38]. Um diese Änderung zu berechnen, wird ein Frame des Audiosignals in mehrere Subframes geteilt, für welche jeweils die Energie E_j wie in Formel (13) berechnet wird. Diese einzelnen Energien werden anschließend durch die Energie E des ursprünglichen Frames geteilt. Die aus E_j und E entstandenen Quotienten e_j des dazugehörigen Subframes j führen letztendlich durch folgende Gleichung zur Energy-Entropie E_E:

$$E_E = -\sum_{j=1}^{J} e_j \cdot log_2(e_j), \quad e_j = \frac{E}{E_j} \tag{14}$$

Nachdem nun die verwendeten zeitabhängigen Features betrachtet wurden, wird nachfolgend jedes der fünf frequenzabhängigen Features dieses Feature-Sets näher erläutert [38]. Für die Berechnung dieser Features wird in jedem Fall von der diskreten Fourier-Transformation Gebrauch gemacht. Die mathematischen Grundlagen dafür finden sich in Kapitel 4.2.

Spektraler Schwerpunkt. Der spektrale Schwerpunkt (engl. *Spectral Centroid*) gibt den „Mittelpunkt“ des Frequenzspektrums an [39]. Je höher der Schwerpunkt bzw. Centroid gelegen ist, desto höher sind die Frequenzen im betrachteten Frequenzbereich.

Berechnet wird der spektrale Schwerpunkt SC eines Frames i als gewichtetes arithmetisches Mittel der einzelnen Amplituden:

$$S_{C_i} = \frac{\sum_{k=0}^{K-1} k \cdot |X_i(k)|^2}{\sum_{k=0}^{K-1} |X_i(k)|^2} \tag{15}$$

Hierbei steht $X_i(k)$ für den k-ten Fourierkoeffizienten des Frames i.

Spektrale Streuung. Die spektrale Streuung (engl. *Spectral Spread*) stellt eine Kennzahl für die Frequenzstreuung um den spektralen Schwerpunkt herum dar. Um diese Streuung σ zu berechnen, wird lediglich die Standardabweichung des Frequenzspektrums gebildet:

$$\sigma = \sqrt{\frac{\sum_{k=0}^{K-1} (k - S_C)^2 \cdot |X_i(k)|^2}{\sum_{k=0}^{K-1} |X_i(k)|^2}} \tag{16}$$

Spektrale Entropie. Die spektrale Entropie ist in ihrer Aussage und ihrer Berechnung der Energie-Entropie sehr ähnlich. Jedoch findet die Kalkulation in diesem Fall in der Frequenzdomäne statt. Das Frequenzspektrum eines Frames wird dazu in L Unterbereiche geteilt. Weiterhin ergibt sich die Energie E_f eines Unterbereichs f durch die Division mit der gesamten Energie im Frequenzspektrum:

$$n_f = \frac{E_f}{\sum_{l=0}^{L-1} E_l}. \tag{17}$$

Die spektrale Entropie geht letztlich durch folgende Gleichung hervor:

$$S_E = -\sum_{f=0}^{L-1} n_f \cdot log_2(n_f) \tag{18}$$

Spektraler Fluss. Der spektrale Fluss S_F (engl. *spectral flux*) beschreibt die Geschwindigkeit der Änderung des Frequenzspektrums von Frame zu Frame [40]. Dafür wird die euklidische Distanz zwischen den normalisierten Fourierkoeffizienten $X_f(k)$ und $X_{f-1}(k)$ beider Frames f und $f-1$ berechnet:

$$S_{F_f} = \sum_{k=0}^{K-1} |||X_f(k)| - |X_{f-1}(k)|||, \quad f \neq 0 \tag{19}$$

Spektraler Rolloff-Punkt. Das letzte Feature des betrachteteten Feature-Sets ist der spektrale Rolloff-Punkt, engl. *spectral rolloff point*. Dieser Punkt stellt die obere Grenze für das Frequenzspektrum dar, in welchem sich P Prozent der in einem Audiosignal vorhandenen Frequenzen befinden. Dabei wird P normalerweise im Bereich von 85% bis 95% gewählt [41]. Auch dieses Feature wird wie die Nulldurchgangsrate dazu verwendet, um Erkenntnisse über das Vorhandensein von menschlicher Stimme in einem Audiosignal zu gewinnen. Mathematisch ist der spektrale Rolloff-Punkt SRP wie folgt definiert:

$$S_{RP} = f(N) = \left(\frac{f_s}{K}\right) N \tag{20}$$

Dabei ist N der größte Frequenzbereich, für den folgende Gleichung gültig ist:

$$\sum_{k=0}^{N} |X(k)|^2 \leq \frac{P}{100} \cdot \sum_{k=0}^{K-1} |X(k)|^2 \tag{21}$$

Wie auch in den vorangegangenen Berechnungen im Bereich der Fouriertransformation steht $X(k)$ für den k-ten Fourierkoeffizienten.

5.4. Chroma-Features

Zwei Töne, deren Frequenzverhältnis einer beliebigen Zweierpotenz entspricht, werden hinsichtlich der Klangfarbe als sehr ähnlich empfunden. Diese Tatsache führt zum Grundbegriff der Oktave, die als das Intervall zwischen einer Musiknote und einer anderen mit der halben oder doppelten Grundfrequenz definiert ist. Nach dieser Definition ist eine Tonhöhenklasse eine Menge aller Tonhöhen bzw. Noten, die eine ganze Zahl von Oktaven voneinander entfernt sind [24]. Jede Tonhöhenklasse wird dabei durch jeweils einen von zwölf Chroma-Werten aus der Menge {C, C♯, D, D♯, E , F, F♯, G, G♯, A, A♯, B} beschrieben. Dabei entsprechen verschiedene Tonbezeichnungen wie C♯ und D♭ dem selben Chroma-Wert. Ein Chroma-Vektor ist demnach ein Vektor mit zwölf Elementen, welcher anzeigt, wieviel Energie jede Tonhöhenklasse in einem bestimmten Signalbereich besitzt [42].

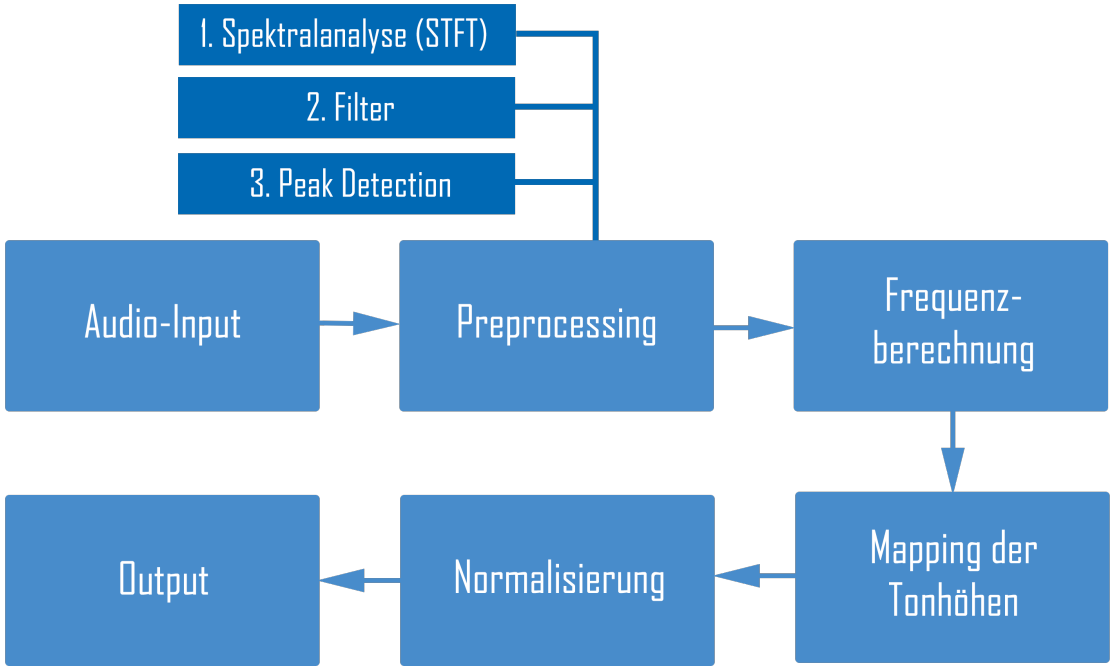


Abb. 5: Flow-Chart für die vollständige Extraktion der Chroma-Features aus einem Audiosignal nach [42]

Die Berechnung des Chroma-Vektors bzw. der Chroma-Features wird in Abbildung 5 schematisch dargestellt. Dabei wird im ersten Schritt die diskrete Fourier-Transformation/DFT (siehe Abschnitt 4.2) auf das Audiosignal angewendet.
Dadurch erhält man das Spektrogram des Signals, welche Informationen über das Frequenzspektrum beinhaltet.

Für diese Arbeit wurde die Spezialform der Kurzzeit-Fourier-Transformation/STFT angewendet, um mehrere, sich überlappende Ausschnitte des Frequenzspektrums zu erhalten. Für jeden Frame müssen nun die vorliegenden Frequenzen herausgefiltert werden. Dabei werden nur die lokalen Maxima innerhalb eines Frames berücksichtigt (*Peak Detection*). Nun folgt die Zuordnung der Frequenzen zu den verschiedenen Tönen und Chroma-Werten. Gemäß der wohltemperierten Skala, welche die Grundlage für die moderne westliche Musik bildet, gibt es im Frequenzbereich zwischen $100\,Hz$ und $5000\,Hz$ 128 verschiedene Töne. Dabei bildet der Ton $A4$ den Referenzpunkt mit einer Frequenz von $440\,Hz$. Für einen aus der STFT generierten Fourierkoeffizienten k gilt dabei folgende Zuordnung zu einem dieser Töne $p \in [0; 127]$:

$$P(p) = \big\{k : F(p - 0.5) \leq F(k) < F(p + 0.5)\big\} \tag{22}$$

Aufgrund der logarithmischen Frequenzabstände zwischen zwei Tönen mit gleichem Chroma-Wert bietet es sich an, das Spektrogram in ein Spektrogram $\mathcal{Y}$ mit logarithmischer x-Achse zu überführen:

$$\mathcal{Y}(n, p) = \sum_{k \in P(p)} \big|\mathcal{X}(n, k)\big|^2 \tag{23}$$

Dabei steht $\mathcal{X}(n, k)$ für den $k - ten$ Fourierkoeffizienten im Frame n. Diese Überführung in Spektrogram mit logarithmischer Frequenzeinteilung erleichtert nun die Zuordnung zu den einzelnen Chroma-Werten (*Pitch Class Mapping*):

$$\mathcal{C}(n, c) = \sum_{\{p \in [0;127] : p\, modulo\, 12 = c\}} \mathcal{Y}(n, p) \tag{24}$$

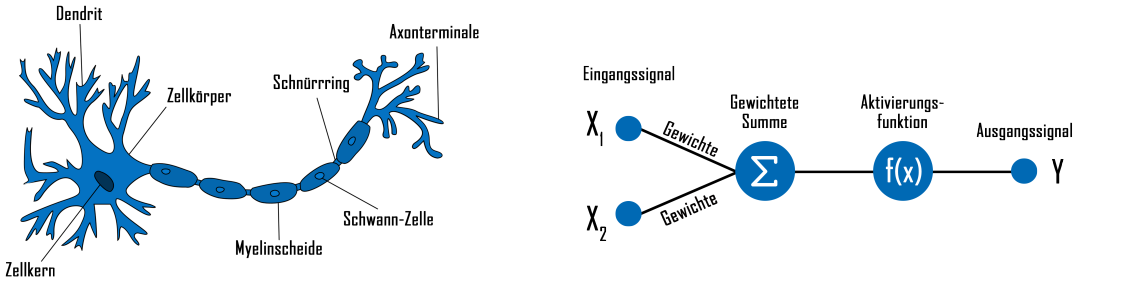
Im letzten Schritt (*Normalization*) werden die zwölf Chroma-Werte jedes Frames normalisiert, indem diese durch den jeweiligen Maximalwert geteilt werden. Dies eliminiert die Abhängigkeit von der globalen Lautstärke des Signals. Am Ende erhält man somit einen Chroma-Vektor für jeden Frame.
Für diese Arbeit wurde für das Chroma-Feature-Set zusätzlich zu den zwölf Chroma-Werten die Standardabweichung für den gesamten Chroma-Vektor als Feature benutzt. Somit erhält man die oben erwähnten 13 Chroma-Features.

6. Verwendung von neuronalen Netzen zur Emotionserkennung

In den vorangegangenen Kapiteln wurde gezeigt und erklärt, warum sich Emotionen aus der menschlichen Stimme ableiten lassen und welche Audio-Features sich dafür eignen. Die Herausforderung besteht nun darin, der Menge an Werten der einzelnen Audio-Features eine Emotion zuzuordnen, also die komplexe Verknüpfung der gewonnen Daten zu klassifizieren.
Forschungen im Bereich der Emotionserkennung zeigt, dass es eine Vielzahl an Klassifizierungsmethoden gibt. Neben altbekannten Ansätzen wie der Verwendung von *Support Vector Machines* [43] oder des *Hidden Markov Models* [44] steht hier in der aktuellen Forschung vor allem der Gebrauch von Methoden im Bereich des maschinellen Lernens im Vordergrund, zu welchem auch künstliche neuronale Netze gehören.
Ein großer Vorteil von künstlichen neuronalen Netzes ist die Fähigkeit, anhand einer vorgegebenen Datenmenge ein Klassifikations-Modell selbstständig anzulernen. Dieser Lernprozess und das Ermitteln von Abhängigkeiten einzelner Features wäre durch den Menschen selbst nur sehr schwer umzusetzen. Trotz des Nachteils, dass für die Erstellung bzw. das Trainieren eines solchen Netzes eine große Datenmenge benötigt wird, ist diese Methode im Hinblick auf die einfache Modellerstellung sehr effizient.
Diese Gegebenheit ist in erster Linie der Grund, weswegen künstliche neuronale Netze für diese Arbeit als Methode zur Klassifizierung ausgewählt wurden. Deswegen soll im nachfolgenden Unterkapitel 6.1 zunächst auf neuronale Netze im Allgemeinen eingegangen werden. Danach wird in Abschnitt 6.2 ein Blick auf die rekurrenten neuronalen Netze geworfen, welche sich durch ihre Netzstruktur gut für die Problemstellung dieser Arbeit eignen.

6.1. Künstliche neuronale Netze im Allgemeinen

Künstliche neuronale Netze gehören zu den naturanalogen Verfahren und werden dem Bereich des maschinellen Lernens zugeordnet [45]. Im Gegensatz zu explizit programmierten Algorithmen sind neuronale Netze in der Lage, durch Anpassung von inneren Parametern im Gehirn zu lernen. Die Grundlage dafür bilden die biologischen neuronalen Netze, welche aus der Verknüpfung von mehreren Neuronen bestehen. Ein solches Neuron und seine Übertragung in ein mathematisches Modell ist in Abbildung 6 grafisch dargestellt.



(a) Grafische Darstellung eines menschlichen Neurons, Bild entnommen aus [45]

(b) Modell eines menschlichen Neurons (Perzeptron)

Abb. 6: Modellierung eines Neurons

Bei der Modellierung wird die gesamte Nervenzelle durch mathematische Funktionen dargestellt. Die Eingangssignale $X_1, X_2, ..., X_n$ entsprechen dabei den Eingangspotentialen an den Dendriten.

Im Zellkörper (Soma) werden die Potentiale mit einer bestimmten Gewichtung W_i aufsummiert und als Eingabeparameter v an die Aktivierungsfunktion $f(x)$ übergeben. Das Ausgangssignal Y entspricht dem Aktivierungspotential an den Axonterminalen im biologischen Neuron.

$$v = \sum_{i=1}^{n} X_i \cdot W_i\,, \quad Y = f(v) \tag{25}$$

Diese mathematische Abbildung definiert letztendlich das sogenannte Perzeptron bzw. künstliche Neuron, welches in seinen Grundzügen im Jahr 1958 von David Rosenblatt definiert wurde [46]. Seitdem wurde das Modell des künstlichen Neurons zwar modifiziert, aber in seinen Grundzügen bildet es das Fundament aller modernen künstlichen neuronalen Netze.
Ein neuronales Netz besteht im Regelfall aus mehreren solcher Neuronen, welche miteinander vernetzt sind [47]. Jedes Neuron hat mehrere Eingabewerte und produziert einen Ausgabewert. Um diesen Ausgabewert zu berechnen, wird zunächst die Summe der Eingaben gebildet. Dabei wird die Gewichtung der einzelnen Eingabewerte berücksichtigt. Diese gewichtete Summe wird nun an eine Aktivierungsfunktion geleitet, welche den Ausgabewert erzeugt. Dieser Ausgabewert kann dann wiederum als Eingabewert für nachfolgende Neuronen dienen.
Abbildung 7 zeigt den Aufbau eines neuronalen Netzes, welches aus dem Verbund von mehreren Neuronen entsteht.

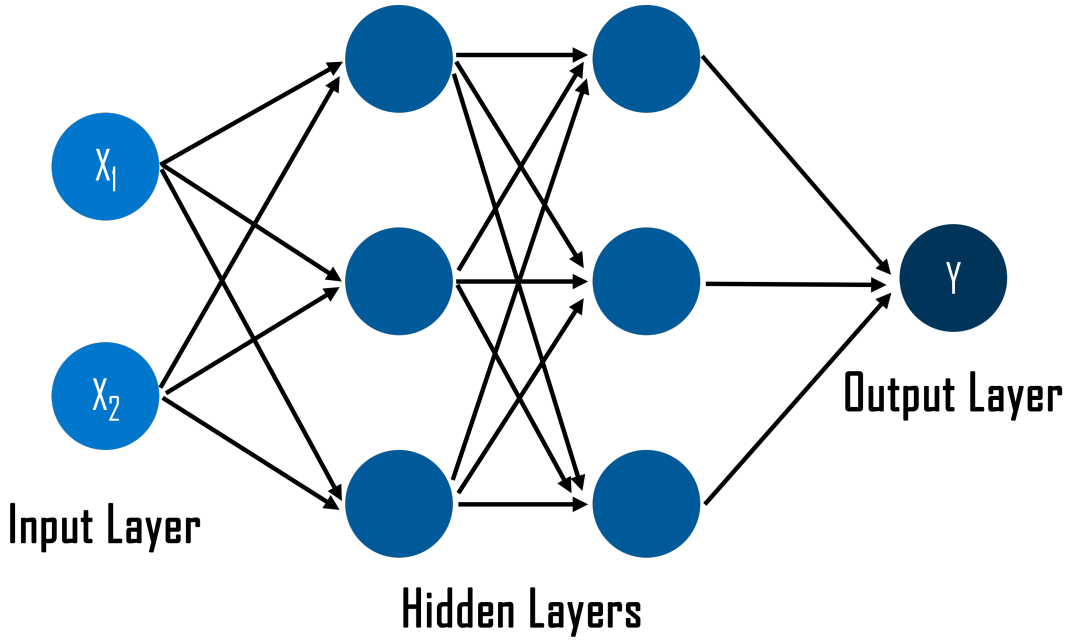


Abb. 7: Einfache Darstellung des Aufbaus eines neuronalen Netzes. Das Eingangssignal X_i durchläuft mehrere Hidden Layers, bevor der Ausgabewert Y erzeugt wird.

Das Eingangssignal X_i durchläuft dabei mehrere *Hidden Layer*, bevor im letzten Neuron der Ausgabewert Y erzeugt wird. Der Lernvorgang (Trainingsphase) eines künstlichen neuronalen Netzes entsteht dadurch, dass die Gewichtungen der Verbindungen zwischen den einzelnen Neuronen verändert werden.

Diese Gewichtungen werden im Realfall meist durch den Prozess der *Backpropagation* angepasst. Dabei wird der Ausgabewert des Netzes mit dem gewünschten Ausgabewert verglichen. Die Differenz beider Werte wird als Fehler des Netzes erachtet. Dieser Fehler wird nun wieder über die Ausgabe- zur Eingabeschicht zurück propagiert. Dabei werden die Gewichtungen der Neuronenverbindungen abhängig von ihrem Einfluss auf den Fehler geändert.

6.2. Rekurrente neuronale Netze

Wie im vorigen Abschnitt erwähnt, werden die einzelnen Neuronen eines neuronalen Netzes miteinander verbunden. Der Aufbau und die Struktur des Netzes und der zugehörigen Neuronen sind je nach Art des gewählten Netzes unterschiedlich. In dieser Arbeit wurde ein sogenanntes *rekurrentes neuronales Netz* verwendet. Das rekurrente neuronale Netz zeichnet sich im Gegensatz zu klassischen Netzen dadurch aus, dass es Verbindungen von Neuronen einer Schicht zu Neuronen derselben oder einer vorangegangenen Schicht gibt [48]. Dies wird in der Fachsprache auch Rückkopplung genannt. Je nach Rückkopplung lassen sich vier grundsätzliche Typen rekurrenter neuronaler Netze unterscheiden (vgl. [49]):

- Direct-Feedback-Netze (direkte Rückkopplungen)
- Indirect-Feedback-Netze (indirekte Rückkopplungen)
- Lateral-Feedback-Netze (seitliche Rückkopplungen)
- Complete-Feedback-Netze (vollständige Rückkopplungen)

Bei Direct-Feedback-Netzen ist der Ausgang eines Neurons ein weiterer Eingang des gleichen Neurons. Netze mit indirekten Rückkopplungen verbinden den Ausgang eines Neurons mit einem Eingang eines Neurons einer vorherigen Schicht. Eine seitliche Rückkopplung stellt eine Verbindung eines Ausgangs eines Neurons mit dem Eingang eines Neurons der gleichen Schicht dar. In Complete-Feedback-Netzen sind alle Neuronenausgänge vollständig mit allen anderen Neuronen verbunden. Es entsteht eine Totalvermaschung der Neuronen.
Prinzipiell lässt sich für rekurrente neuronale Netze also sagen, dass diese eine Art Kurzzeitgedächtnis besitzen. Ein Neuron in einem solchen Netz benutzt die Ausgaben vorhergehender Zyklen, um für nachfolgende Zyklen den Ausgabewert anzupassen. Je weiter ein Zyklus zurückliegt, desto weniger hat dieser Zyklus und die dort entstandene Ausgabe Einfluss auf die Generierung eines neuen Ausgabewertes. Sollen jedoch Ausgabeinformationen eines lang zurückliegenden Zyklus mit in Betracht gezogen werden, so kommen klassische rekurrente Netze an ihre Grenzen. Um dieses Problem zu beheben, wurde das sogenannte *Long Short-Term Memory (LSTM)* entwickelt. Dieses 1997 von Sepp Hochreiter und Jürgen Schmidhuber vorgestellte Netzmodul ist in der Lage, langfristige Abhängigkeiten zu erlernen [50].

Der Aufbau und die Funktionsweise einer LSTM-Zelle innerhalb eines neuronalen Netzes wird anhand von Abbildung 8 erklärt.

Eine LSTM-Zelle besteht aus einem Zellzustand C_t, welcher den Speicherzustand repräsentiert, und aus drei Regulatoren, den sogenannten *Gates*. Dieses Gates beeinflussen den Zellzustand einer LSTM-Zelle. Das erste Gate, welches den Namen *Forget Gate* trägt, trifft Entscheidungen über das Beibehalten oder Verwerfen von Informationen im Zellzustand.

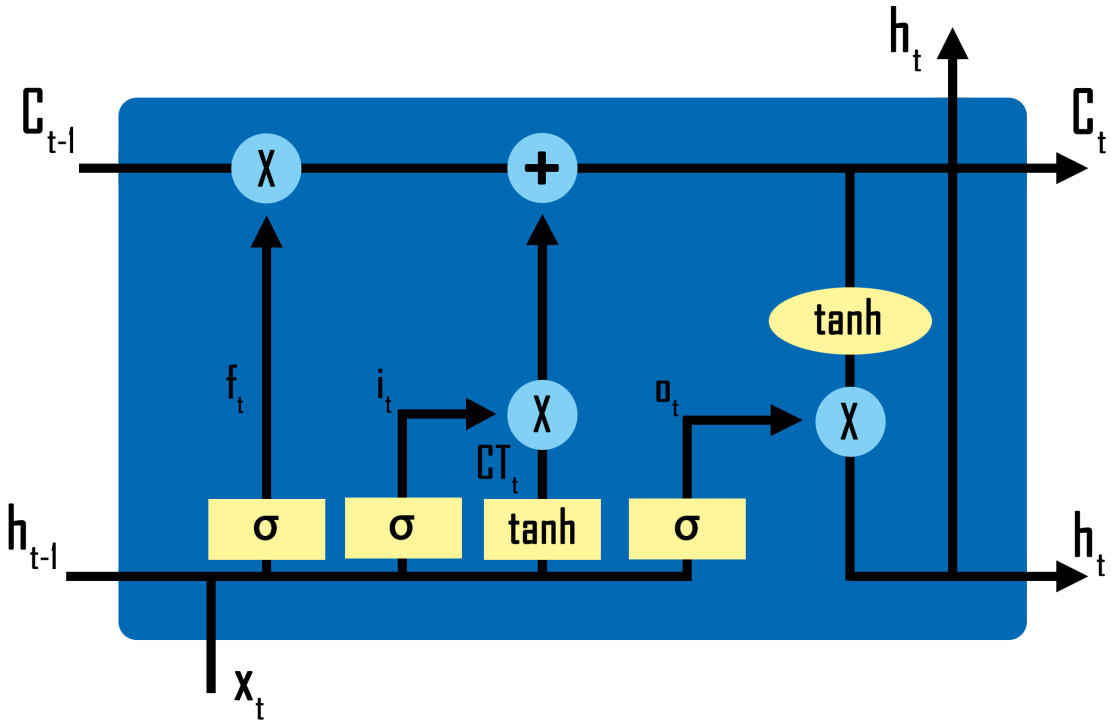


Abb. 8: Schematischer Aufbau einer LSTM-Zelle. Die Variable C_t steht dabei für den Zustand der Zelle zu einem Zeitpunkt t, x_t für den Eingabewert der Zelle und h_t für den Ausgabewert der Zelle.. Die gelb hinterlegten Felder stellen die Aktivierungsfunktionen *tanh* bzw. σ dar.

Diese Entscheidung wird von einer Sigmoidfunktion anhand der Eingabewerte x_t und h_{t-1} getroffen. Mathematisch wird dies durch folgende Formel ausgedrückt:

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \tag{26}$$

Die Variablen $W \in \mathbb{R}^{h \times d}$ und $U \in \mathbb{R}^{h \times h}$ stehen dabei für Gewichtungsmatrizen mit d als Menge der Eingabewerte und h als Variable für die Anzahl der sogenannten *hidden units*.
Im zweiten Schritt wird dann entschieden, welche neuen Informationen im Zellzustand gespeichert werden. Dabei wird zunächst mithilfe des *Input Gates* mithilfe einer Sigmoidfunktion eine Aussage über zu aktualisierende Werte getroffen; danach werden durch eine Hyperbelfunktion (*tanh*) neue Informationen im Zellzustand gespeichert:

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \tag{27}$$

$$CT_t = tanh(W_C x_t + U_C h_{t-1}) \tag{28}$$

Diese zwei Gates erzeugen somit durch Verwerfen von alten Informationen und Hinzufügen von neuen Informationen den aktualisierten Zellzustand C_t:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot CT_t \tag{29}$$

Zum Schluss wird mithilfe des *Output Gates* entschieden, welche Informationen durch die LSTM-Zelle ausgegeben werden sollen.

Dazu wird wieder durch eine Sigmoidfunktion eine Entscheidung über auszugebende Werte getroffen und anschließend durch eine Hyperbelfunktion im Intervall $[-1, 1]$ angeordnet:

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \tag{30}$$

$$h_t = o_t \cdot tanh(C_t) \tag{31}$$

Das in diesem Abschnitt vorgestellte rekurrente neuronale Netz findet aufgrund der Eigenschaft des Zurückblickens häufig bei der Bearbeitung von sequentiellen Daten Anwendung. Da Audio-Signale und die daraus extrahierten Features in ihrer Natur auch sequentiell sind und sich im Zeitverlauf gegenseitig beeinflussen, eignet sich ein solches, aus LSTM-Zellen bestehendes rekurrentes Netz hervorragend als Netzstruktur für die Klassifizierung von Emotionen in dieser Arbeit.

7. Ansatz und Durchführung

In den vorangegangenen drei Kapiteln wurden die Grundlagen dieser Arbeit erläutert. Dabei wurde gezeigt, wie Emotion und menschliche Stimme zusammenhängen und miteinander korrelieren. Weiterhin wurde ein Überblick gegeben, anhand welcher Audio-Features diese Korrelation deutlich wird und wie sie zu einer Aussage über eine vorliegende Emotion verwendet werden können. Letztlich wurden die Grundlagen des verwendeten rekurrenten neuronalen Netzes näher erläutert.
In diesem Kapitel wird nun auf die Durchführung und den Ansatz der Untersuchungen eingegangen, welche zur Konstruktion eines effizienten Prozesses zur Emotionserkennung genutzt wurden. Dabei wird im ersten Unterkapitel 7.1 eine Übersicht über den gesamten Prozess vom Audiosignal hin zur Emotionsklasse gegeben. Danach wird im nächsten Unterkapitel 7.2 auf die Validierung der generierten Modelle eingegangen. Abschließend folgt dann in 7.3 eine Übersicht über die experimentelle Durchführung und deren Ablauf, welche den praktischen Kern dieser Arbeit bildet.

7.1. Übersicht über den Prozess der Emotionserkennung

Der gesamte Prozess, der benötigt wird, um anhand eines digitalen Audiosignals auf die vorliegenden Emotionen Rückschlüsse zu ziehen, ist in Abbildung 9 grafisch dargestellt.

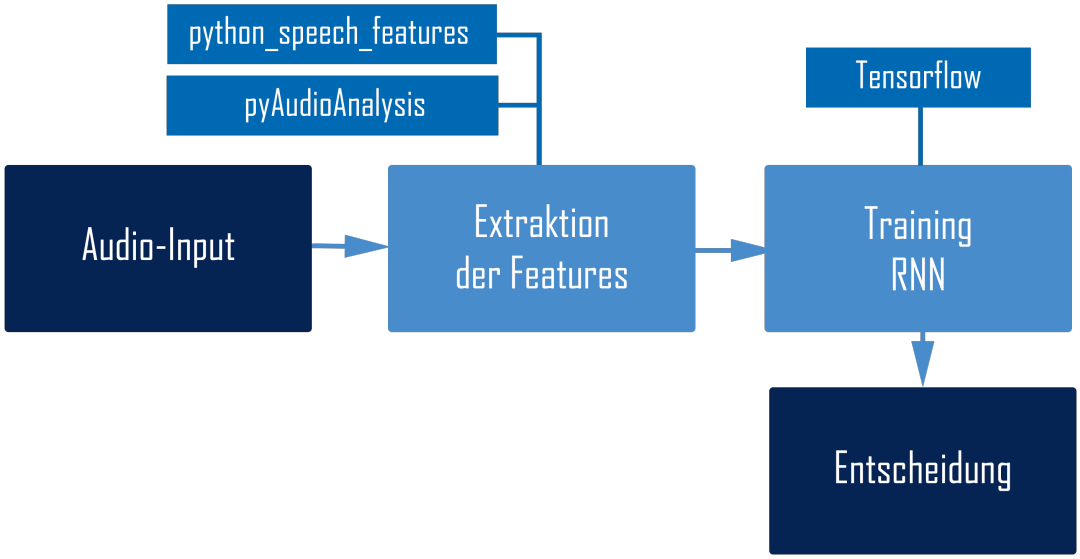


Abb. 9: Prozess der Emotionserkennung vom Audiosignal bis zur Emotionsklasse mithilfe eines rekurrenten neuronalen Netzes. Die Abbildung zeigt den Ablauf der Trainingsphase des Modells; bei einer anschließenden Nutzung des trainierten Modells zur Validierung desselben entfällt der Trainingsschritt.

Den Anfang dieses Prozesses bildet ein digitales Audiosignal. Hierbei muss dieses Audiosignal durch einige Anpassungen in eine einheitliche Form gebracht werden, um ebenso einheitliche Features zu generieren, welche für das rekurrente neuronale Netz benötigt werden. Im nächsten Schritt werden die einzelnen Features berechnet. Im Fall dieser Arbeit werden drei verschiedene Feature-Sets genutzt, welche einzeln oder in Kombination verwendet werden. Danach kann das rekurrente neuronale Netz trainiert und im letzen Schritt eine Entscheidung über die vorliegende Emotion getroffen werden. Diese drei Schritte, welche vor dem Ergebnis dieses Prozess stehen, werden in den nachfolgenden Abschnitten genauer betrachtet.

7.1.1. Auswahl und Vorbereitung des Audio-Inputs

Um ein künstliches neuronales Netz zu trainieren, bedarf es einer gewissen Menge an Trainingsdaten. Die Auswahl und die Vorbereitung dieser Daten, aus denen dann für das neuronale Netz Features extrahiert werden, steht am Anfang des Gesamtprozesses.
Eine Möglichkeit, um an diese, in einem Datensatz eingebundenen Daten zu gelangen, sind vorangegangene Forschungsarbeiten im Bereich der Emotionserkennung, welche die für diese Projekte generierten Datensätze öffentlich zur Verfügung stellen. Der Vorteil dieser vorgefertigten Datensätze ist dabei, dass die einzelnen Daten bereits mit Labels versehen sind. Insgesamt zwei dieser Datensätze wurden als Datengrundlage für diese Arbeit ausgewählt.
Dies ist zum einen der deutschsprachige Datensatz *EmoDB*, welcher von Forschern der Universität Berlin zusammengestellt wurde [51]. Dieser Datensatz beinhaltet 535 Samples, welche von zehn verschiedenen Schauspielern beider Geschlechter in einer Studioumgebung aufgenommen wurden. Hierbei werden die einzelnen Daten den Emotionsklassen Trauer, Wut, Freude, Ekel, Langeweile, Angst und Emotionsneutralität zugeordnet. Zum anderen wurde für diese Arbeit der englischsprachige *Ryerson-Datensatz* gewählt [52]. Es liegen in diesem Datensatz insgesamt 7356 Samples vor, von denen sich aber nur die beinhalteten 1440 Sprachaufnahmen als Samples für diese Arbeit verwenden lassen. Diese Sprachaufnahmen wurden unter perfekten Bedingungen von 20 männlichen und weiblichen Schauspielern eingesprochen. Zusätzlich zu den Emotionsklassen des deutschen Datensatzes beinhaltet der Ryerson-Datensatz die Emotionsklasse Überraschung, welche aber der Einheitlichkeit halber vernachlässigt wurde - die zugehörigen Daten dieser Emotionsklasse wurden demnach nicht mit für das Training verwendet. Neben diesen zwei schon vorhandenen Datensätze wurde zusätzlich ein eigener Datensatz erstellt, der lediglich mich selbst als Sprecher beinhaltet. Insgesamt entstanden dabei 420 Samples in deutscher Sprache, welche mittels eines Studiomikrofons aufgenommen wurden. Die Emotionsklassen, in welche sich die einzelnen Samples einteilen lassen, sind dabei dieselben wie in dem EmoDB-Datensatz. Eine Übersicht über diese drei vorgestellten Datensätze ist in der nachfolgenden Tabelle 2 ersichtlich.

	Sprache	Anzahl der Samples	Anzahl der Sprecher
EmoDB	deutsch	535	10 (5x weiblich, 5x männlich)
Ryerson	englisch	1035	20 (10x weiblich, 10x männlich)
Eigene Aufnahmen	deutsch	420	1x männlich

Tab. 2: Übersicht über die verwendeten Datensätze zum Trainieren des rekurrenten neuronalen Netzes

Die zwei deutschen Datensätze stellen in Summe 975 Samples bereit. Um die deutsche und die englische Sprache im Hinblick auf die Güte der Emotionserkennung fair vergleichen zu können, wurde die Anzahl der Samples in der Ryerson-Datenbank von 1440 auf 1035 Samples reduziert.

7.1.2. Preprocessing

Um die oben vorgestellten Datensätze in einem neuronalen Netz verwenden zu können, müssen alle Daten die gleiche Länge besitzen. Dies trifft auch auf die Samplerate und die Bittiefe der einzelnen Samples zu. Dazu wurde zunächst mit einem Python-Skript und der Python-Bibliothek `soundfile` die Samplerate auf $48\,kHz$ und die Bittiefe auf $768\,kbps$ festgelegt.
In einer Forschungsarbeit von Michael Neumann und Ngoc Thang Vu konnte gezeigt werden, dass eine Samplelänge von sechs Sekunden einen optimale Wert für die Emotionserkennung darstellt [53]. Deswegen wurde diese Länge auch für die in dieser Arbeit verwendeten Samples übernommen. Die Samples der zwei externen Datensätze besitzen hierbei lediglich eine Länge von durchschnittlich zwei bis drei Sekunden. Daher wurden die Samples solange an sich selbst angefügt, bis eine Audiolänge von sechs Sekunden erreicht war. Bei den Samples des eigenen Datensatzes wurde schon bei der Erstellung eine Aufnahmelänge von sechs Sekunden gewählt.
Weiterhin wurde, ebenfalls mit der Python-Bibliothek `soundfile`, die durchschnittliche Lautstärke der einzelnen Datensätze normalisiert. Dieser Vorgang schließt aus, dass grundsätzliche Lautstärkeunterschiede zwischen den Datensätzen das spätere Training beeinflussen.
Wie schon erwähnt, wurden alle Datensätze in Studioqualität aufgenommen. Während durch die nicht vorhandenen Hintergrundgeräusche die Emotionserkennung vereinfacht wird, kann durch diese Eigenschaft der Audio-Samples die Realität nur bedingt abgebildet werden, da dort Störgeräusche allgegenwärtig sind. Deswegen wurden die bestehenden Datensätze erweitert, indem jedes Sample mit einem zufälliges Umgebungsgeräusch wie das Schreien eines Kindes oder das Bellen eines Hundes unterlegt wurde. Diese Hintergrundgeräusche entstammen einem von Marc Moreaux zusammengestellten Datensatz mit ingesamt 2000 verschiedenen Samples [54]. Somit liegen insgesamt sechs verschiedene Datensätze vor, da jeder der drei Basis-Datensätze einmal mit und einmal ohne Hintergrundgeräusch vorhanden ist.

7.1.3. Berechnung der Audio-Features

Aus den in den verschiedenen Datensätzen enthaltenen Daten müssen nun in diesem Schritt die jeweiligen Audio-Features extrahiert werden. Dazu wurden die in den Kapiteln 5.2 bis 5.4 vorgestellten Feature-Sets verwendet. Diese Feature-Sets dienen dann einzeln oder in Kombination mit den anderen Sets als Input für das neuronale Netz.

Das Feature-Set, welches die MFCC-Features beinhaltet, wurde als Erstes näher betrachtet. Die Berechnung erfolgte dabei unter Zuhilfenahme der Python-Bibliothek `pythonspeechfeatures` [55]. Das Ergebnis der Berechnung hängt von mehreren Parametern ab, welche dann letztlich auch die Genauigkeit des neuronalen Netzes beeinflussen:

- Fenstergröße - Bei der Berechnung der MFCC-Features wird das Audiosignal automatisch in mehrere sich überlappende Fenster geteilt. Dies hat zur Folge, dass am Ende für ein Audiosignal eine Vielzahl an MFCC-Features vorliegt und somit der Inhalt desselben besser abgebildet werden kann. Die Fenstergröße gibt die Länge eines solchen Fensters in Sekunden an.

- Schrittgröße - Die Schrittgröße gibt den Abstand zwischen zwei Fenstern an. Um den Überlappungseffekt zu erzielen, sollte dieser merklich kleiner sein als die Fenstergröße.

Im Normalfall beträgt die Schrittgröße ein Achtel der Fenstergröße.

- Pre-Emphasis Filter - Vor die Extraktion der MFCC-Features kann ein Pre-Emphasis Filter gesetzt werden, welcher potentielle Störgeräusche reduziert.
- Anzahl der Filter - Dies ist der Wert für die Anzahl der Filter in der Filterbank, welche das Frequenzband logarithmisch in mehrere Frequenzbänder teilen.
- Anzahl der MFCC-Features - Dieser Parameter gibt die Anzahl der aus dem Berechnungsprozess zurückgegebenen Cepstra an, welche dann letztendlich die MFCC-Features bilden. Dabei darf der Wert des Parameters nicht größer sein als die Anzahl der Filter in der Filterbank.

Die MFCC-Features und insbesondere die unterschiedlichen Parameter während der Berechnung machen einen nicht unwesentlichen Teil der Arbeit aus. Ausschlaggebend für die Güte der Parameterwahl ist die Validierungs-Genauigkeit des rekurrenten neuronalen Netzes. Auf den genauen Ablauf der Experimente wird in Kapitel 7.3 eingegangen. Neben den MFCC-Features wurden das in Kapitel 5.3 erläuterte Feature-Set der frequenz- und zeitabhängigen Features verwendet. Dieses Feature-Set wird genauso wie das in Abschnitt 5.4 betrachtete Chroma-Feature-Set mithilfe der Python-Bibliothek `pyAudioAnalysis` berechnet [35]. Dabei sind für beide Berechnungen nur die Parameter der Fenstergröße und der Schrittgröße von Bedeutung. Diese ergeben sich aus den Werten dieser Größen bei der Berechnung der MFCC-Features. Kombiniert man verschiedene Feature-Sets und nutzt dabei unterschiedliche Werte für die Fenster- und Schrittgröße, so ergeben sich für jedes Feature-Set eine unterschiedliche Anzahl von Frames. Die Anzahl der Frames muss aber für das Trainieren des neuronalen Netzes in jedem Fall einheitlich sein.
Jedes dieser Feature-Sets und die darin enthaltenen Features für jedes einzelne Fenster des Audiosignals liegen dann am Ende als jeweils eine Matrix vor. Diese Matrix muss hinsichtlich ihrer Form bzw. Dimension gleich sein. Im Fall der Kombination der Feature-Sets aus den zwei verwendeten Bibliotheken erfordert dies eine Transponierung. Beide Bibliotheken geben nach der Berechnung die Matrix in einem Numpy-Array zurück, was diese Transponierung erleichtert. Zudem lassen sich aus diesem Numpy-Array vergleichsweise einfach Tensoren im Sinne des von Tensorflow verwendeten Datentyps erstellen.

7.1.4. Training des neuronalen Netzes

Nachdem die einzelnen Features berechnet wurden, können diese verwendet werden, um ein künstliches neuronales Netz zu trainieren. Wie das Kapitel 6.2 zeigt, eignen sich für sequentielle Daten besonders rekurrente neuronale Netze. Die aus dem Audiosignal extrahierten Features haben auch nach ihrer Berechnung noch einen sequentiellen Charakter, da diese nach wie vor eine Zeitabhängigkeit besitzen. Es wurde sich in dieser Arbeit deswegen für ein rekurrentes neuronales Netz entschieden, dessen Schichten (*layer*) aus mehreren LSTM-Zellen bestehen.
Um dieses Netz zu realisieren, wurde das von Google entwickelte Framework *Tensorflow* verwendet. Dieses Framework bietet eine Menge Tools, um eine neuronales Netz zu erstellen, zu trainieren und auszuwerten [56]. Zusätzlich bietet Tensorflow die Möglichkeit, das Netz auf einer GPU des Herstellers Nvidia zu trainieren. Dadurch kann die Zeit, welche das Training benötigt, um ein Vielfaches reduziert werden. Von dieser Möglichkeit wurde aufgrund des verwendeten Rechners, in welchem eine *Nvidia RTX 2060* integriert ist, Gebrauch gemacht.

Das Ergebnis der Trainingsphase bzw. die Genauigkeit des daraus entstehenden Modells ist massiv von der Struktur und der Parametrisierung des Netzes abhängig. Es ergeben sich folgende Faktoren, welche wiederum während der experimentellen Durchführung untersucht wurden:

- Netzstruktur - Die Netzstruktur hat den größten Einfluss auf das Ergebnis der Trainingsphase. Hierbei spielt die Menge und Art der Schichten, sowie ihre Verknüpfung und Anordnung untereinander eine tragende Rolle. Der Hauptbestandteil in jeder Ausprägung des Netzes sind die einzelnen LSTM-Zellen.
- Anzahl der LSTM-Zellen - Diese Größe beschreibt die Menge an LSTM-Zellen, welche in der dazugehörigen Schicht integriert sind.
- Optimierer - Der Optimierer hat in einem neuronalen Netz die Aufgabe, die Gewichte während des Trainings anzupassen. Je nach Wahl und Parametrisierung dieses Optimieres lernt ein neuronales Netz unterschiedlich gut und schnell; eine falsche Wahl und Einstellung des Optimierers führt unter Umständen zum sogenannten *Overfitting*. Overfitting beschreibt den Sachverhalt, dass ein Modell zu stark an einen Trainingsdatensatz angepasst wird und deswegen die Fähigkeit verliert, auch bei trainingsfremden Daten korrekte Aussagen zu treffen.

Die Wahl der Netzstruktur und der zugehörigen Parameter ist ein wesentlicher Teil dieser Arbeit. Angefangen von der Entscheidung für einen bestimmten Optimierer wurden nach und nach verschiedene Netzstrukturen untersucht und hinsichtlich der Validierungs-Genauigkeit des entstandenen Modells untersucht. Für eine bessere Auswertung wurde von dem in Tensorflow integrierten Tool *Tensorboard* gemacht. Tensorboard macht es unter anderem möglich, die Trainingsphase und die zugehörigen Werte grafisch festzuhalten. Zudem lässt sich eine *Wahrheitsmatrix* integrieren, welche neben einer Aussage über die gesamte Genauigkeit des Modells zusätzlich eine Aussage über die Genauigkeit der einzelnen Klassen (in diesem Fall Emotionen) treffen kann.

Der gesamte Prozess des Trainings und der zugehörigen Netzerstellung wurde wie auch die Berechnung der Audio-Features mittels eines Python-Skripts realisiert.

7.2. Validierung des Netzes

Während des Trainings wird das entstandene Modell automatisch durch die von den Trainingsdaten gesonderten Testdaten validiert. Dies ist ein erster und wichtiger Anhaltspunkt, um eine Aussage über die Effizienz des Modells treffen zu können. Um in diesem Hinblick eine zuverlässigere Aussage zu treffen, wurden alle trainierten Modelle zusätzlich mit zwei kleineren Datensätzen trainiert (siehe Tabelle 3).
Der erste verwendete Datensatz zur Validierung stammt aus dem *CAER-Datensatz* [57]. Dieser Datensatz enthält 13.000 Filmauschnitte von amerikanischen Serien, welche jeweils mit der vorliegenden Emotion gekennzeichnet sind. Zur Validierung der in dieser Arbeit entstandenen Modelle wurden 35 ausdrucksstarke Samples daraus verwendet. Desweiteren wurde zur Untersuchung der trainierten Modelle ein Mix von 28 verschiedenen Samples aus den Trainingsdatensätzen benutzt, welche gleichzeitig aber zum Trainieren oder Testen des jeweiligen Modells benutzt wurden.
Diese Samples werden deshalb nach der Feature-Berechnung bzw. vor der Trainingsphase gesondert gespeichert. Dieser Validierungs-Datensatz beinhaltet sowohl Samples mit und Samples ohne Umgebungsgeräusche.

	Sprache	Anzahl der Samples	Anzahl der Sprecher
CAER	englisch	35	-
Mix aus den für das Training verwendeten Datensätzen	deutsch/englisch	28 je Modell	31 (15x weiblich, 16x männlich)

Tab. 3: Übersicht über die verwendeten Datensätze zur Validierung der einzelnen Modelle. Im Falle des CAER-Datensatzes kann nur bedingt eine Aussage über die Anzahl der Sprecher gegeben werden, da die einzelnen Samples aus Filmausschnitten stammen.

7.3. Experimentelle Durchführung

In diesem Abschnitt wird ein Überblick über die im Rahmen dieser Arbeit durchgeführten Untersuchungen gegeben. Eine Aussage über die Güte der einzelnen Parameter wurde aufgrund der Validierungs-Genauigkeit des rekurrenten Netzes nach Durchlaufen von 50 Epochen getroffen. Alle Experimente, welche die Berechnung der Features und das Trainieren und Validieren der einzelnen Modelle beinhalten, wurden auf einem Rechner mit den folgenden Hard- und Softwarekomponenten durchgeführt:

- CPU: AMD Ryzen 7 2700
- GPU: Nvidia RTX 2060
- Arbeitsspeicher: 32 GB DDR4
- Betriebssystem: Linux Ubuntu 20.04 LTS
- Tensorflow: Tensorflow Version 2.2.0

In Abbildung 10 auf der nächsten Seite wird der Ablauf der getätigten Untersuchungen gezeigt.. Wie in der Grafik ersichtlich ist, wurde als Erstes die Parametrisierung der Feature-Berechnung genauer betrachtet. Für die Berechnung aller drei verwendeten Feature-Sets ist zunächst die Fenster- und Schrittgröße von großer Bedeutung. Diese müssen bei der Kombination von mehreren Feature-Sets in jedem Fall einheitlich sein. Daher wurden diese Parameter nur für die MFCC-Features getestet und die optimalen Werte dann analog bei der Berechnung der anderen zwei Feature-Sets verwendet. Um einen Testreihe durchzuführen und die einzelnen Werte hinsichtlich ihrer Güte für die Verwendung zur Emotionserkennung zu bewerten, wurde ein rekurrentes Netz mit fünf LSTM-Schichten gewählt (siehe Anhang A auf Seite 46, Abbildung 19). Insgesamt wurden zehn verschiedene Fenstergrößen im Intervall von $0.05\,s$ bis $1.6\,s$ evaluiert. Die Schrittgröße errechnet sich dann analog durch das Teilen der Fenstergröße durch acht.

Das Erschließen effizienter Werte für die Fenster- und Schrittgröße erlaubt es, verschiedene Netz- bzw. Modellstrukturen zu testen. Der grundlegende Ansatz ist dabei die Verwendung eines sequentiellen Netzes mit unterschiedlich vielen LSTM-Schichten. Jedes betrachtete Netz besitzt eine Eingabeschicht, auf welche die LSTM-Schichten folgen. Nach diesen Kernschichten folgt eine Dropout-Schicht und zwei Dense-Schichten. Während die Dropout-Schicht das sogenanne Overfitting reduziert, passen die Dense-Schichten die Ausgabe des Netzes so an, dass am Ende die Klassifizierung in die sieben verschiedenen Emotionsklassen vorliegt.

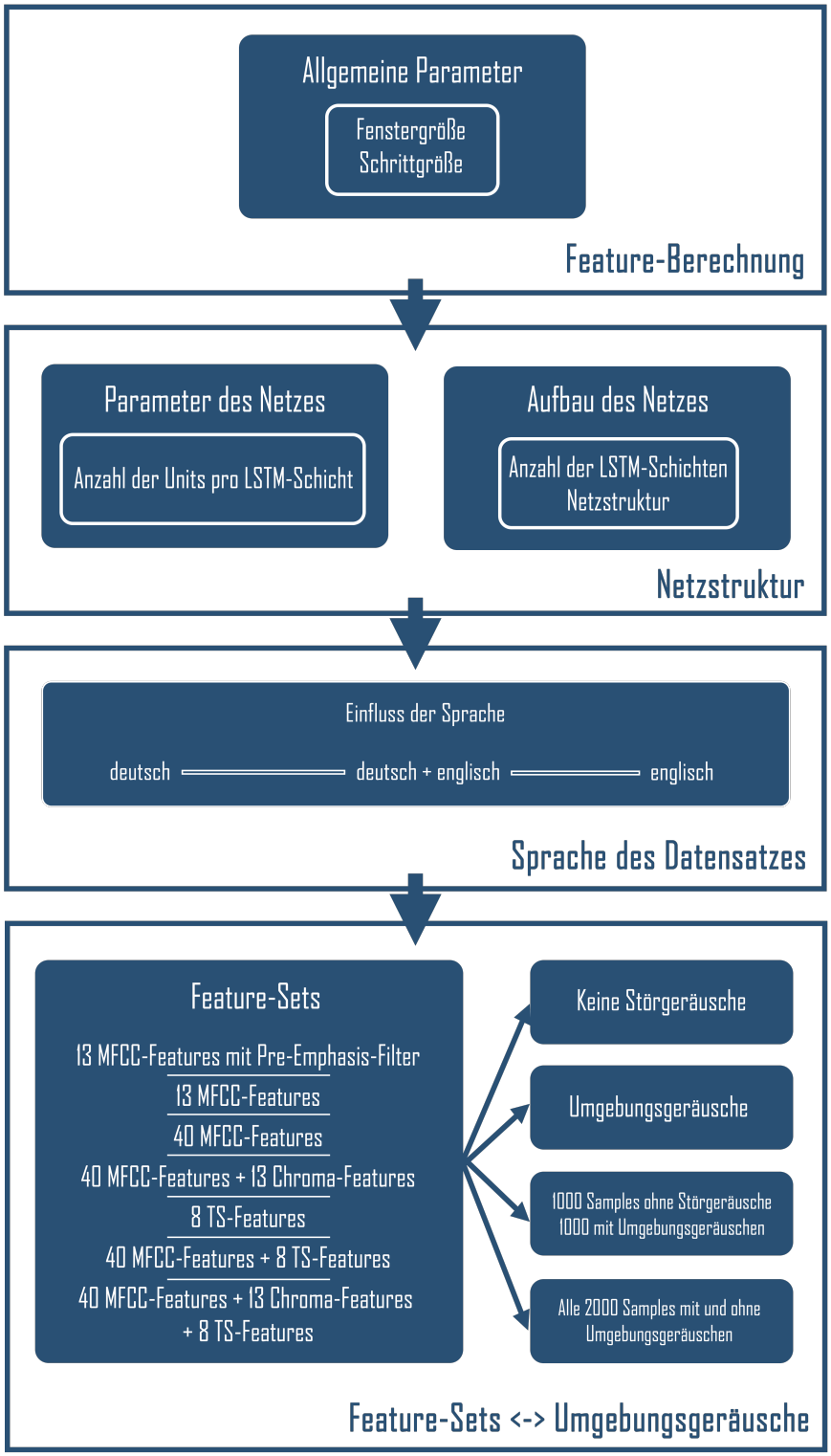


Abb. 10: Übersicht über den Ablauf der experimentellen Untersuchungen, welche im Rahmen dieser Arbeit durchgeführt wurden. Die Abkürzung `TS` steht dabei für das Feature-Set der zeit- und frequenzabhängigen Features.

Beginnend mit einem aus einer LSTM-Schicht bestehenden Netz wurden schrittweise das Verhalten der Validierungs-Genauigkeit bis zum Erreichen von neun LSTM-Schichten beobachtet.
Während der Durchführung dieser Testreihen entstand aufgrund der nachlassenden Validierungs-Genauigkeit bei mehr als fünf LSTM-Schichten die Idee, die Struktur des Netzes in Richtung eines *Residual Networks*, kurz *RESNET* zu verändern. Diese Form eines neuronalen Netzes wurde erstmals 2016 auf der *IEEE Conference on Computer Vision and Pattern Recognition* vorgestellt und ist ursprünglich als Faltungsnetzwerk zur Bilderkennung konzipiert wurden [58]. Die Idee hinter dieser Netzstruktur ist, dass die einzelnen Schichten in einem mehrschichtigen Netz nicht nur die Ausgabe der vorhergenden Schicht erhalten, sondern auch die Ausgabe weiter zurückliegender Schichten in den Lernvorgang mit einbeziehen. Durch diese Netzstruktur wird der Effekt des sogenannten *verschwindenden Gradienten* (engl. *Vanishing Gradient*) verringert. [59]. Dieser Effekt tritt dadurch auf, dass der für die Anpassung der Gewichte (Backpropagation) verantwortliche Gradient immer kleiner wird, je mehr Schichten durchlaufen werden. Bei der Verwendung eines residualen Netzwerks kann dieser Effekt vermieden werden, indem bei der Backpropagation mehrere Schichten übersprungen werden und der Gradient eine höhere Auswirkungen auf die vorderen Schichten des Netzwerks hat.

Der Ansatz eines residualen Netzes wurde für diese Arbeit auf ein rekurrentes Netz übertragen. Nach zahlreichen Entwürfen kristallisierte sich ein LSTM-RESNET heraus, welches neun LSTM-Schichten beinhaltet. Der Modellgraph zu diesem Netz ist in Anhang A, Abbildung 20 auf Seite 47 ersichtlich. Aufgrund der besseren Ergebnisse im Vergleich zum fünfschichtigen LSTM-Netz wurde dieses Netz für allen folgenden Untersuchungen verwendet.

Weiterhin wurden Untersuchungen zur Ermittlung einer optimalen Anzahl an LSTM-Zellen pro Schicht (*units*) durchgeführt. Dabei wurden die Ergebnisse zum einen für das entwickelte LSTM-RESNET generiert, zum anderen für das fünfschichtige LSTM-Netz.

Nach Abschluss der Untersuchung der Netzstruktur wurde im nächsten Schritt die Abhängigkeit des Trainingserfolges eines Netzes von der gewählten Sprache betrachtet. Die Genauigkeit der entstandenen Aussagen wird durch die annähernd gleiche Aufnahmequalität der deutschen und englischen Audio-Samples gerechtfertigt. Grund für diese Untersuchung ist dabei, Indizien für die Unterstützung der in Kapitel 3.2 getätigten Aussage zu generieren, dass sich Emotionen kultur- und sprachübergreifend aus der menschlichen Stimme ableiten lassen. Dazu wurden die deutschen und englischen Datensätze mit sieben verschiedenen Feature-Sets (siehe Abbildung 10) untersucht. Dieser Vorgang wurde für die Kombination aus deutschen und englischen Daten wiederholt, welche den ansonsten in dieser Arbeit verwendeteten Trainingsdatensatz bilden und insgesamt 1990 Samples bereitstellen.

Den größten Teil der durchgeführten Untersuchungen bildet die Erprobung der einzelnen Feature-Sets in Abhängigkeit von der Beschaffenheit der Audio-Samples. Es wurden hierfür aus den drei grundlegenden Feature-Sets, welche in Kapitel 5 definiert wurden, insgesamt sieben neue Feature-Sets generiert. Die Zusammensetzung dieser Sets ist in Abbildung 10 ersichtlich. Für alle Feature-Sets wurde daraufhin die Effizienz für vier verschiedene Datensätze ermittelt. Diese Datensätze setzen sich aus englischen und deutschen Audio-Samples in Kombination zusammen.

Der erste Datensatz besteht dabei aus Audio-Samples ohne Umgebungsgeräuschen; der zweite Datensatz beinhaltet diese Audio-Samples mit Umgebungsgeräuschen. Für den dritten Datensatz wurden 995 Audio-Samples ohne und 995 Audio-Samples mit Umgebungsgeräuschen ausgewählt. Der vierte Datensatz stellt letztlich die die Kombination desersten und zweiten Datensatz dar, wodurch dieser insgesamt 3980 Samples enthält. Da die Trainings- und Testdaten vor Beginn der Trainingsphase zufällig aus den jeweiligen Datensätzen zusammengestellt werden, erfolgte jeder Versuchsdurchgang jeweils fünfmal, um durch eine anschließende Ermittlung des Durchschnittswertes der Validierungs-Genauigkeit Messfehler zu vermeiden.

Im Anschluss an diese Testreihen wurde die in Abschnitt 7.2 erläuterte Validierung mit trainingsfremden Daten durchgeführt. Da der Ablauf der Validierung bereits im betreffenden obigen Kapitel erklärt wurde, wird an dieser Stelle auf eine wiederholte Betrachtung verzichtet.

8. Ergebnisse und Diskussion

Nachdem in Abschnitt 7.3 ein Überblick über die experimentellen Untersuchungen im Rahmen dieser Arbeit gegeben wurde, werden in diesem Kapitel die Ergebnisse vorgestellt. Dabei wird gleichzeitig auch erörtert, welche Erkenntnisse sich aus diesen Ergebnissen gewinnen lassen.

8.1. Parametrisierung der Feature-Berechnung

Die ersten durchgeführten Tests beziehen sich auf die Parametrisierung der Feature-Berechnung. Hierfür wurde zunächst die Fenster- und die Schrittgröße bei der Berechnung von 13 MFCC-Features betrachtet. Dafür wurde für die Ermittlung der als Maßstab dienenden Validierungs-Genauigkeit ein aus fünf LSTM-Schichten bestehendes rekurrentes Netz verwendet (siehe Anhang A, Abbildung 19).
In Abbildung 11 sind die Ergebnisse grafisch dargestellt, die sich für die Validierungs-Genauigkeit nach Durchlaufen von 50 Epochen ergeben:

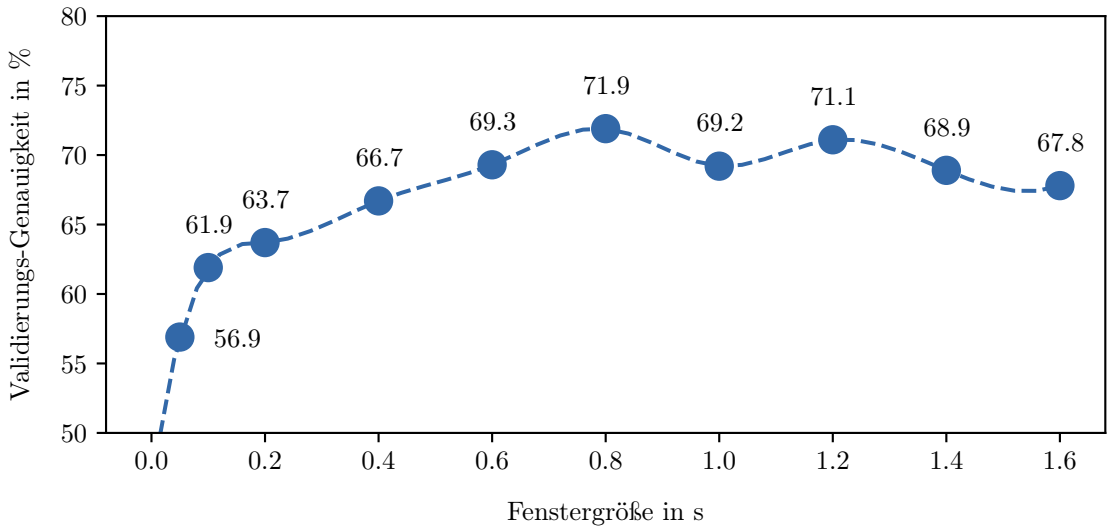


Abb. 11: Validierungs-Genauigkeit in Abhängigkeit von der gewählten Fenstergröße zur Feature-Berechnung. Die einzelnen Werte wurden mithilfe eines rekurrenten Netzes mit fünf LSTM-Schichten ermittelt. Die gestrichelte Linie stellt die kubische Interpolation der Datenpunkte dar.

In der Realität beträgt die Länge eines gesprochenen Wortes durchschnittlich 200 ms, die eines einzelnen Lautes 30 ms [8]. Dies ist der Grund, warum die Fenstergröße in ähnlichen Forschungen zur Emotionserkennung anhand der Stimme sich meist in diesem Intervall befindet. Dennoch zeigen die in dieser Arbeit durchgeführten Untersuchungen, dass sich die Validierungs-Genauigkeit durch eine wesentliche größere Fenstergröße deutlich verbessern lässt. So beträgt die Effizienzsteigerung bei einer Fenstergröße von 800 ms im Vergleich zu einer Fenstergröße von 200 ms immerhin acht Prozent. Die Untersuchungen zeigen auch, dass die Validierungs-Genauigkeit bei einer Fenstergröße größer als 800 ms wieder abfällt. Offensichtlich enthält also ein Frame dieser Länge die optimale Menge an Informationen. Aus diesem ermittelten Wert der Fenstergröße ergibt sich analog eine Schrittgröße von 100 ms. Beide ermittelten Optimalwerte für die Fenster- bzw. Schrittgröße wurden für alle anderen getätigten Untersuchungen verwendet.

8.2. Netzstruktur

Einen großen Teil der in dieser Arbeit geleisteten Forschung macht die Struktur des trainierten Netzes aus. Hierbei beeinflussen schon kleinste Änderungen der Netzstruktur das Ergebnis wesentlich. Betrachtet man die Netzstrukturen von Forschungsarbeiten, die in ihrer Problemstellung dieser Arbeit ähnlich sind, so sind die jeweiligen Ansätze zur Gestaltung des Netzes sehr unterschiedlich. Dies umfasst nicht nur die Anordnung und die Anzahl der einzelnen Schichten, sondern auch die Anzahl der LSTM-Zellen pro Schicht.
Um deshalb eine eigene Aussage über eine effiziente Netzstruktur tätigen zu können, wurde zunächst die Anzahl und Anordnung der einzelnen Schichten des Netzes untersucht. Dabei wurde die Validierungs-Genauigkeit von einem rekurrenten Netz mit einer LSTM-Schicht bis hin zu einem aus neun LSTM-Schichten bestehendem rekurrentem Netz durchgeführt. Zusätzlich wurde im Hinblick auf die Netzstruktur die Idee des residualen Netzes für ein rekurrentes Netz umgesetzt (siehe 7.3). Als Input des Netzes diente das MFCC-Feature-Set mit 40 Koeffizienten; als Datensatz wurden alle drei in Kapitel 7.1.1 definierten Datensätze in Kombination verwendet. Die Ergebnisse finden sich in der nachfolgenden Abbildung 12:

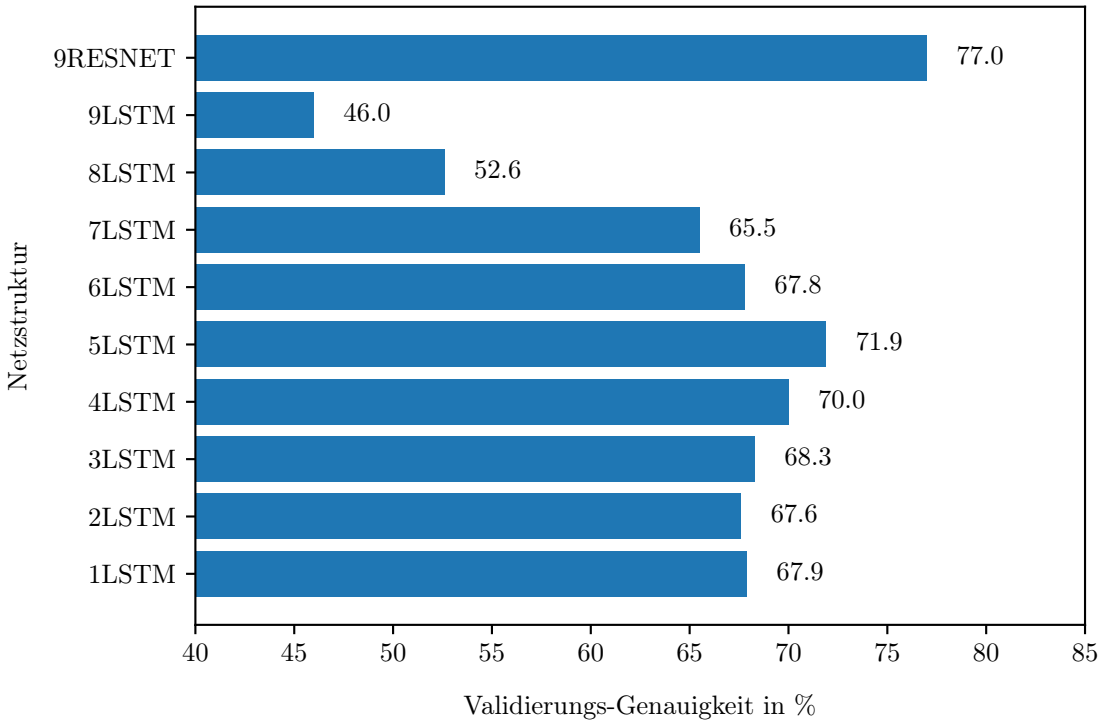


Abb. 12: Validierungs-Genauigkeit in Abhängigkeit von der gewählten Netzstruktur. Die dargestellten Werte ergeben sich aus den Durchschnittswerten von fünf Durchläufen mit jeweils 50 Epochen. Die Zahl vor dem Namen der Netzstruktur gibt jeweils die Anzahl der hintereinandergereihten LSTM-Schichten an.

Auffällig an den erzielten Ergebnissen ist in erster Linie die hohe durchschnittliche Validierungs-Genauigkeit des implementierten rekurrenten RESNETs. Desweiteren fällt auf, das bei der bloßen Aneinanderreihung der LSTM-Schichten bei einer Anzahl von fünf LSTM-Schichten die besten Ergebnisse erzielt werden. Vielmehr fällt die Validierungs-Genauigkeit bei einem Netz mit mehr als fünf LSTM-Schichten stark ab. Vermutlich ist dies auf den Effekt des verschwindenden zurückzuführen.

Dass die Verwendung eines residualen Netzes dem entgegenwirkt zeigt die Steigerung der Validierungs-Genauigkeit des RESNETs mit neun LSTM-Schichten gegenüber dem einfachen neunschichtigen LSTM-Netzes mit einem Unterschied von 31 Prozent.

Neben der Netzstruktur als solche wurde weiterhin die Anzahl der LSTM-Zellen pro Schicht in Abhängigkeit von der Validierungs-Genauigkeit untersucht. Als Untersuchungsgrößen dienten die Zweierpotenzen 32, 64, 128, 256, 512 und 1028. Dieser Test wurde zum einen mit einem einfachen, aus fünf LSTM-Schichten bestehenden Netz durchgeführt, zum anderen mit entwickelten residualen Netzwerk. Die Ergebnisse, die sich bei Verwendung von 40 MFCC-Features angewandt auf die deutschen und englischen Datensätze in Kombination ergeben, finden sich in Abbildung 13. Die genauen Werte sind in Tabellenform in Anhang B, Tabelle 5 auf Seite 48 ersichtlich.

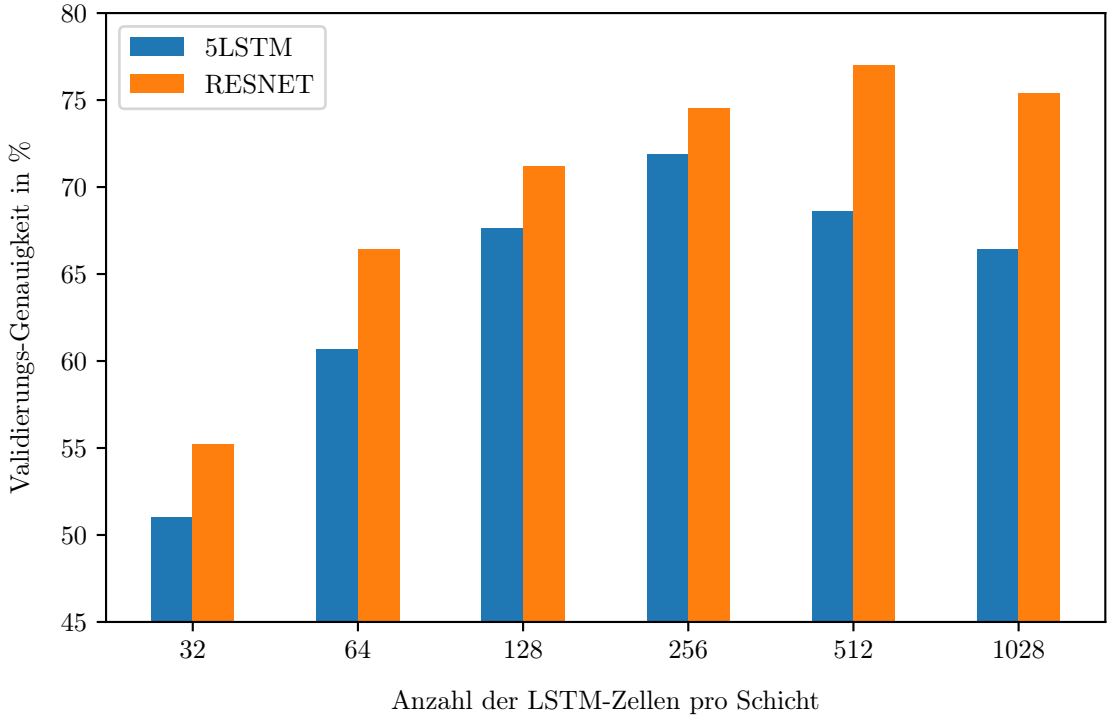


Abb. 13: Validierungs-Genauigkeit in Abhängigkeit von der Anzahl der Zellen pro LSTM-Schicht. Dabei wurden die entsprechenden Werte zum einen für ein rekurrentes Netz mit fünf LSTM-Schichten (5LSTM) ermittelt, zum anderen für das residuale rekurrente Netz (RESNET). Hierbei wurde der jeweilige Maximalwert nach dem Durchlaufen von 50 Epochen festgehalten.

Bei der Betrachtung der Ergebnisse wird deutlich, dass sich die zwei untersuchten Netzstrukturen durchaus unterscheiden. Für das fünfschichtige LSTM-Netz liegt die beste Validierungs-Genauigkeit bei 256 LSTM-Zellen pro Schicht, für das RESNET bei 512 Zellen. Auch lässt sich aus den Ergebnissen ableiten, dass die Validierungs-Genauigkeit ab einer gewissen Zahl an LSTM-Zellen wieder absteigt. Weitere, nicht in der Abbildung dargestellte Tests haben gezeigt, dass dies auch für das RESNET bei mehr als 512 Zellen der Fall ist.

8.3. Sprache des Datensatzes

Nachdem durch die oben ausgewerteten Ergebnisse eine geeignete Parametrisierung der Feature-Berechnung und eine effiziente Netzstruktur ermittelt werden konnte, wurde die Auswirkung der Sprache auf die Emotionserkennung untersucht. Hierbei stand vor allem eine Bestätigung oder Ablehnung der These im Vordergrund, dass sich Emotionen trotz kultureller Unterschieder von der Sprache ableiten lassen [17]. Die einheitliche Qualität der Audio-Samples aller Datensätze war die grundlegende Voraussetzung für diese Auswertung.

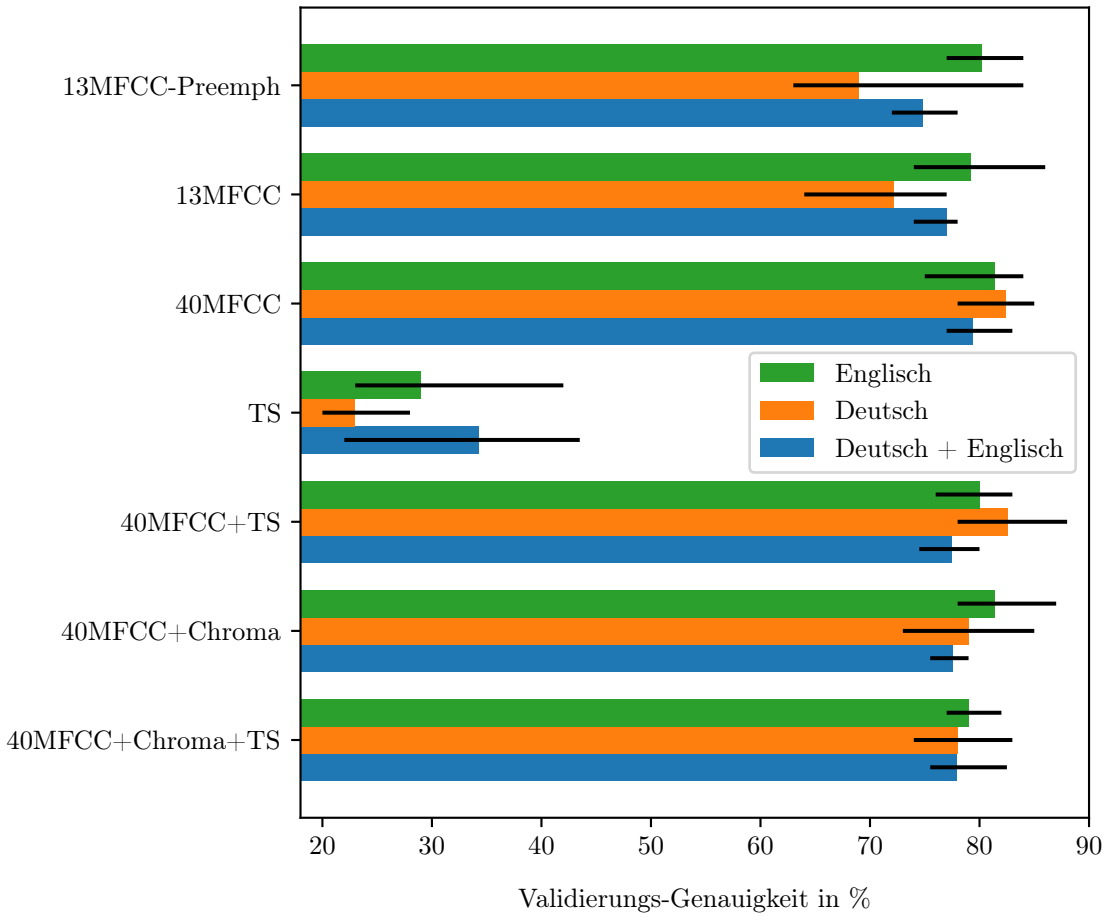


Abb. 14: Validierungs-Genauigkeit in Abhängigkeit von der Sprache des Datensatzes nach Durchlaufen von 50 Epochen. Dabei stellt die schwarze Linie die jeweilige Varianz dar. Die Abkürzung `TS` steht für das Feature-Set der zeit- und frequenzabhängigen Features. Genaue Zahlenwerte der Ergebnisse finden sich in Anhang B auf Seite 48, Tabelle 6.

Die obenstehenden Ergebnisse zeigen die Unterschiede zwischen deutsch und englisch. Dabei wurden sieben Feature-Sets jeweils mit den zwei deutschen Datensätzen, dem englischen Datensatz und den drei Datensätzen in Kombination untersucht. Auf die deutlichen Unterschiede der Feature-Sets im Allgemeinen wird im nächsten Kapitel 8.4 in einem anderen Experiment eingegangen.
Anhand der Ergebnisse ist zunächst ersichtlich, dass die Kombination aller Datensätze für alle Feature-Sets schlechter abschneidet als die anhand der Sprache gruppierten Datensätze.

Dies kann zum einen den Grund haben, dass es in der Qualität und in der emotionalen Ausdrucksstärke der Aufnahmen doch Unterschiede gibt. Zum anderen kann die Unterschiedlichkeit der Ausdrucksstärke von Emotionen in der Sprache allgemein die Ursache für diesen Sachverhalt sein. Dem widerspricht aber die Tatsache, dass je nach Feature-Set die englischen oder die deutschen Datensätze bessere Ergebnisse erzielen.
Insgesamt deuten die Ergebnisse darauf hin, dass sich in der englischen und deutschen Sprache Emotionen gleichermaßen gut ableiten lassen. Diese Erkenntnis führte wiederum dazu, dass für die Generierung der im nächsten Kapitel vorgestellten Ergebnisse für alle Untersuchungen die Kombination aus allen drei Datensätzen verwendet wurde.

8.4. Effizienz der Feature-Sets

Dieses Kapitel beschäftigt sich mit der Auswertung der Ergebnisse von sieben Feature-Sets. Diese Feature-Sets wurden in Abhängigkeit von vier Datensätzen untersucht, welche jeweils aus den zwei deutschen und dem englischen Datensatz bestehen. Der Unterschied zwischen den Datensätzen liegt jeweils in dem Vorhandensein von Umgebungsgeräuschen. Die Abbildung 15 auf Seite 41 gibt einen grafischen Überblick über die generierten Ergebnisse; die genauen Werte der Validierungs-Genauigkeiten finden sich in Tabelle 4 ebenfalls auf Seite 41. Alle diese Werte ergeben sich bei Verwendung des in dieser Arbeit entwickelten residualen Netzwerks nach einem Training von jeweils 50 Epochen. Dabei wurde jeweils aus fünf Durchgängen der Durchschnittswert gebildet.

Zunächst fällt bei Betrachtung der erzielten Werte auf, dass mit der maximalen Validierungs-Genauigkeit **81,2 Prozent** ein relativ hoher Wert erreicht wurde.

Weiterhin fällt auf, dass der Einsatz eines Pre-Emphasis-Filters keine Auswirkung auf den Trainingserfolg hat. Im Fall des Datensatzes ohne Umgebungsgeräusche verschlechtert sich das Ergebnis sogar um 2,2 Prozent. Vermutlich liegt dies daran, dass durch das Absenken der tiefen Frequenzen durch den Pre-Emphasis-Filter Audio-Informationen verloren gehen.

Das Feature-Set der zeit- und frequenzabhängigen Features (TS) erreichte bei allen verwendeten Datensätzen die schlechtesten Ergebnisse. Auch ist in diesem Fall die ebenfalls in der Grafik dargestellte Streuung im Vergleich zu den anderen Feature-Sets verhältnismäßig hoch. Beide Tatsachen lassen vermuten, dass sich dieses Feature-Set eher weniger für die Emotionserkennung eignet.

Eine der Ideen während der Entstehung dieser Arbeit war es, nicht nur die häufig verwendeten 13 MFCC-Features zu verwenden, sondern diese Anzahl zu erhöhen. Die Ergebnisse zeigen, dass sich dieser Ansatz bewährte. Das Feature-Set mit 40 MFCC-Features erzielte bei drei von vier verwendeten Datensätzen den Maximalwert. Dieses Feature-Set konnte jedoch durch Kombination mit den anderen zwei Feature-Sets keine besseren Ergebnisse erzielen.

Betrachtet man die Unterschiede zwischen den vier Datensätzen, so erreicht der Datensatz, welcher alle Samples einmal mit und einmal ohne Umgebungsgeräusche enthält, die besten Werte. Es liegt die Vermutung nahe, dass durch das Trainieren aller Samples mit und ohne Noise das neuronale Netz lernt, Stimme und Störgeräusch zu unterscheiden.

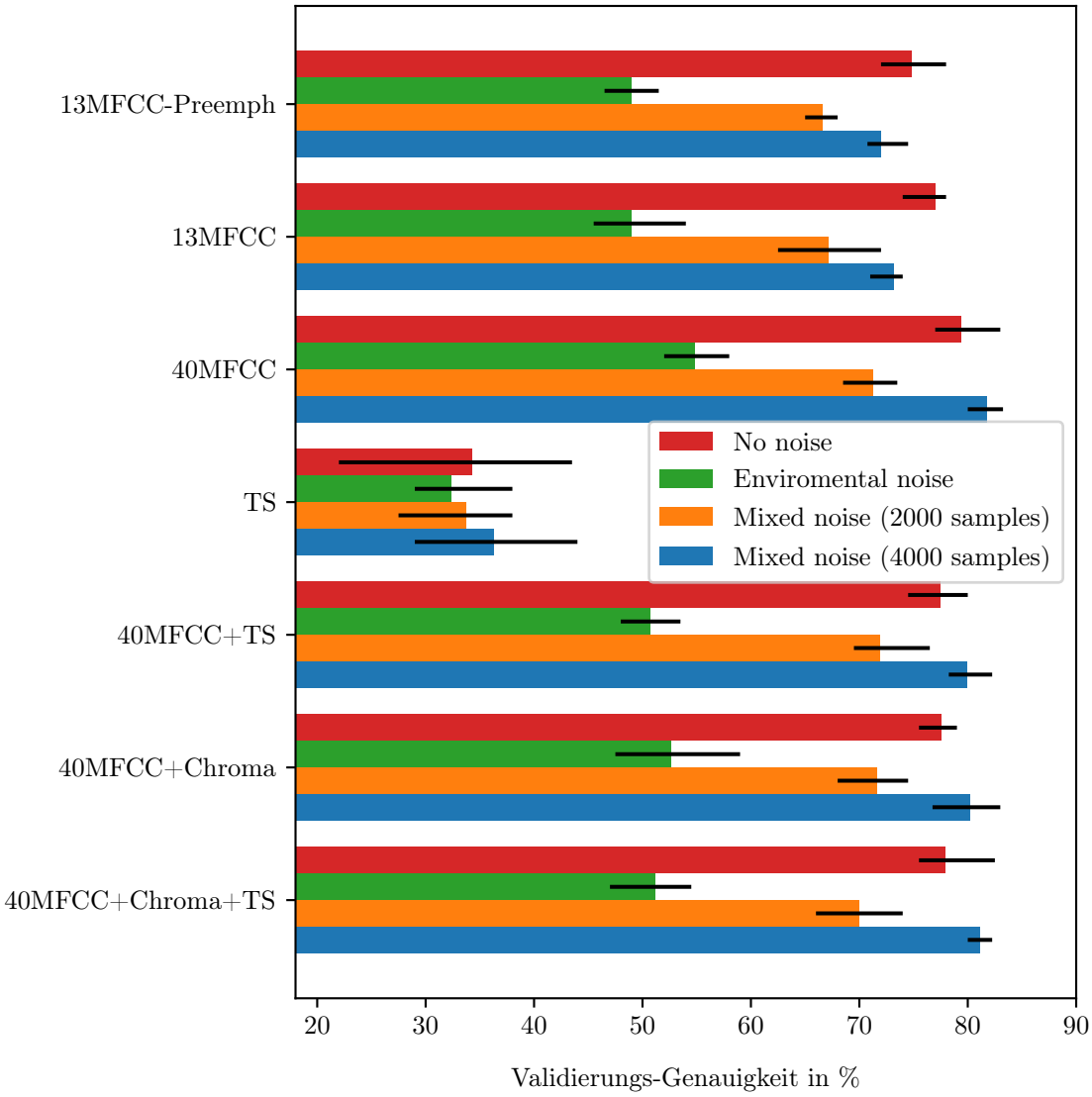


Abb. 15: Validierungs-Genauigkeit der sieben Feature-Sets in Abhängigkeit von dem Vorhandensein von Umgebungsgeräuschen. Die Ergebnisse stellen die Durchschnittswerte von fünf Durchläufen dar, welche nach jeweils 50 Epochen erzielt wurden. Die Abkürzung `TS` steht für das Feature-Set der zeit- und frequenzabhängigen Features.

	No noise	Enviromental noise	Mixed noise (2000 samples)	Mixed noise (4000 samples)
13MFCC-Preemph	74,8 %	49,0 %	66,6 %	72,0 %
13MFCC	77,0 %	49,0 %	67,2 %	73,2 %
40MFCC	**79,4 %**	**54,8 %**	71,3 %	**81,2 %**
TS	34,3 %	32,4 %	33,7 %	36,3 %
MFCC+TS	77,5 %	50,7 %	**71,9 %**	79,9 %
MFCC+Chroma	77,6 %	52,6 %	71,6 %	80,3 %
MFCC+Chroma+TS	77,9 %	51,2 %	70,0 %	81,1 %

Tab. 4: Genaue Angabe der erzielten Werte der Validierungs-Genauigkeit, welche in Abbildung 15 dargestellt sind. Die fettgedruckten Werte in der Tabelle stellen die Bestwerte für den jeweiligen Datensatz an.

An zweiter Stelle steht der Datensatz, welcher alle Samples ohne Hintergrundgeräusche beinhaltet. Wird das Netz nur mit Samples mit Hintergrundgeräuschen trainiert, fällt das Ergebnis deutlich schlechter aus. Der Grund für diese zwei Ergebnisaussagen ist vermutlich darin begründet, dass sich die Emotionen aus Samples ohne Störgeräusche besser extrahieren lassen.

8.5. Wahrheitsmatrix

Neben der Validierungs-Genauigkeit einer Trainingskonfiguration in ihrer Gesamtheit lohnt es sich, die Wahrheitsmatrix (Confusion Matrix) zu betrachten. Diese gibt Aufschluss darüber, welche Emotionen besonders gut und welche Emotionen eher schlecht erkannt werden. Die Auswertung jeder einzelnen Wahrheitsmatrix für jedes trainierte Modell würde den Rahmen dieses Kapitels sprengen. Deshalb ist in Abbildung 16 stellvertretend die Wahrheitsmatrix für die beste Trainingskonfiguration (40 MFCC-Features, Datensatz mit 4000 Samples) dargestellt:

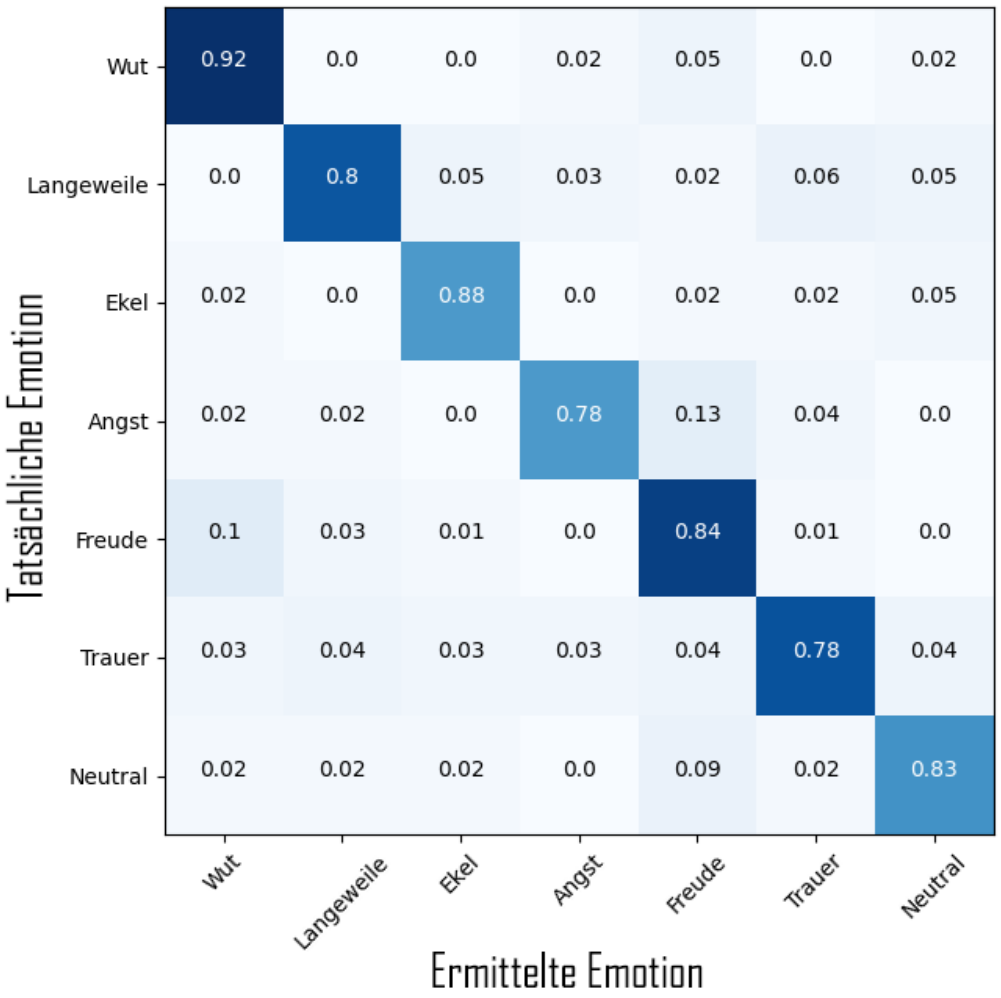


Abb. 16: Wahrheitsmatrix nach 50 Epochen bei Training des residualen Netzes. Als Feature-Set dienten 40 MFCC-Features; der Datensatz beinhaltet alle Datensätze einmal mit und einmal ohne Umgebungsgeräusche.

Die Ergebnisse zeigen, dass sich die Emotion Wut mit 92 Prozent am Besten erkennen lässt. Gleichzeitig wird diese Emotion mit einer Rate von 10 Prozent auch dann ermittelt, wenn eigentlich die Emotion Freude vorliegt. Dies ist in der Tatsache begründet, dass beide Emotionen im Vergleich zu anderen fünf Emotionen hohe Sprachenergien besitzen.
Weiterhin grenzt sich die Emotion Ekel mit 88 Prozent und die Emotion Freude mit 84 Prozent relativ stark von den anderen Emotionen ab. Wie die Emotion Wut auch lassen sich diese zwei Emotionen auch vom Menschen relativ eindeutig erkennen.
Insgesamt spiegelt diese Wahrheitsmatrix mit ihren Werten in gewisser Weise die Realität wieder - analog zu der menschlichen Auffassung von Emotionen werden auch hier Emotionen wie Freude und Wut besser unterschieden als Langeweile und Emotionsneutralität.

8.6. Validierung mit trainingsfremden Daten

Auch wenn die in Kapitel 8.4 erreichten Validierungs-Genauigkeiten schon eine Aussage über die Effizienz der Audio-Features und der Netzstruktur treffen können, wurde zusätzlich mit zwei Validierung-Datensätzen gearbeitet. Eine genaue Beschreibung dieser Datensätze findet sich in Kapitel 7.2. Der CAER-Datensatz ist ein Datensatz zur multimodalen Emotionserkennung. In den in diesem Datensatz enthaltenen Samples gibt meist nur das Gesicht Aufschluss über die empfundene Emotion. Weiterhin ist das Audiosignal aller Samples mit Störgeräuschen behaftet.

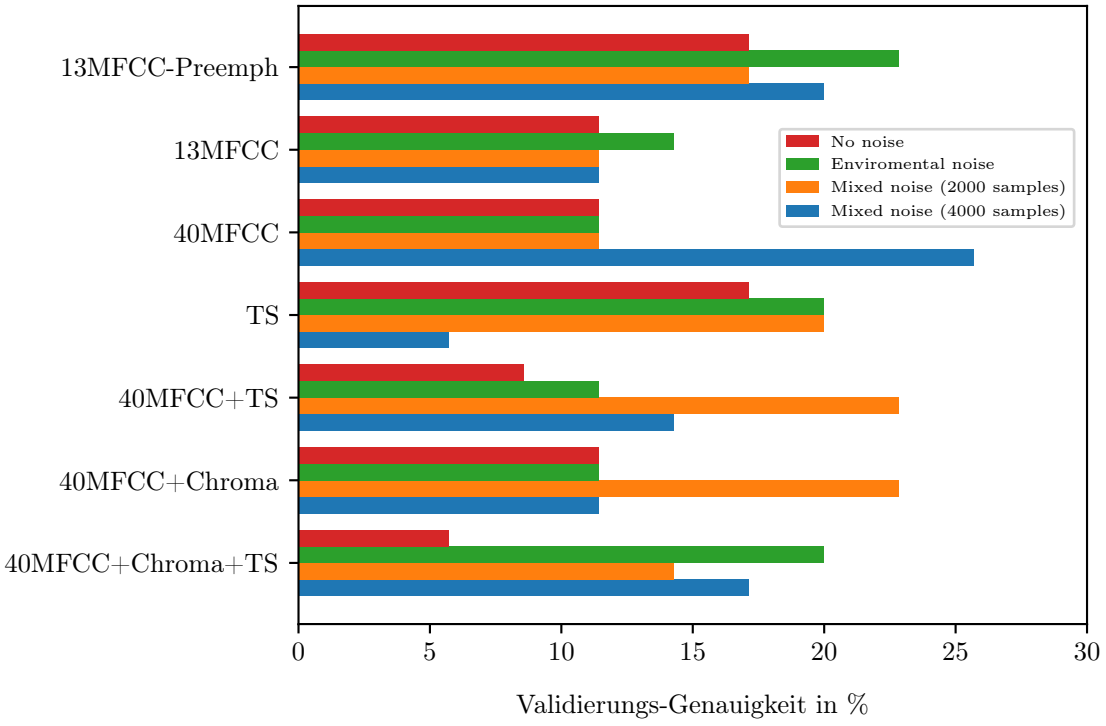


Abb. 17: Validierungs-Genauigkeit der sieben Feature-Sets in Abhängigkeit von dem Vorhandensein von Umgebungsgeräuschen. Die Werte ergeben sich aus der Validierung der jeweiligen Modellkonfiguration mit 35 Samples aus dem CAER-Datensatz. Die Abkürzung `TS` steht für das Feature-Set der zeit- und frequenzabhängigen Features. Die genauen Zahlenwerte finden sich in Anhang B auf Seite 49 in Tabelle 7.

Bei der Betrachtung der Ergebnisse in Abbildung 17 auf Seite 43 lässt sich eine sehr schlechte Genauigkeit feststellten. Der Maximalwert ist mit 25,71 Prozent nur wenig höher als die Ratewahrscheinlichkeit von 14,28 Prozent. Dies bestätigt die Notwendigkeit eines multimodalen Systems zur Emotionserkennung für diesen Datensatz. Auch kann die Erhöhung der Menge der Trainingsdaten zu einer Ergebnisverbesserung beitragen.

Weiterhin wurden die einzelnen Modellkonfigurationen mit Daten aus den Trainingsdatensätzen validiert, welche dem Netz in der Trainingsphase vorenthalten wurden. Eine Übersicht über die Menge und die Zusammensetzung dieser Daten findet sich in Kapitel 7.2 auf Seite 31.

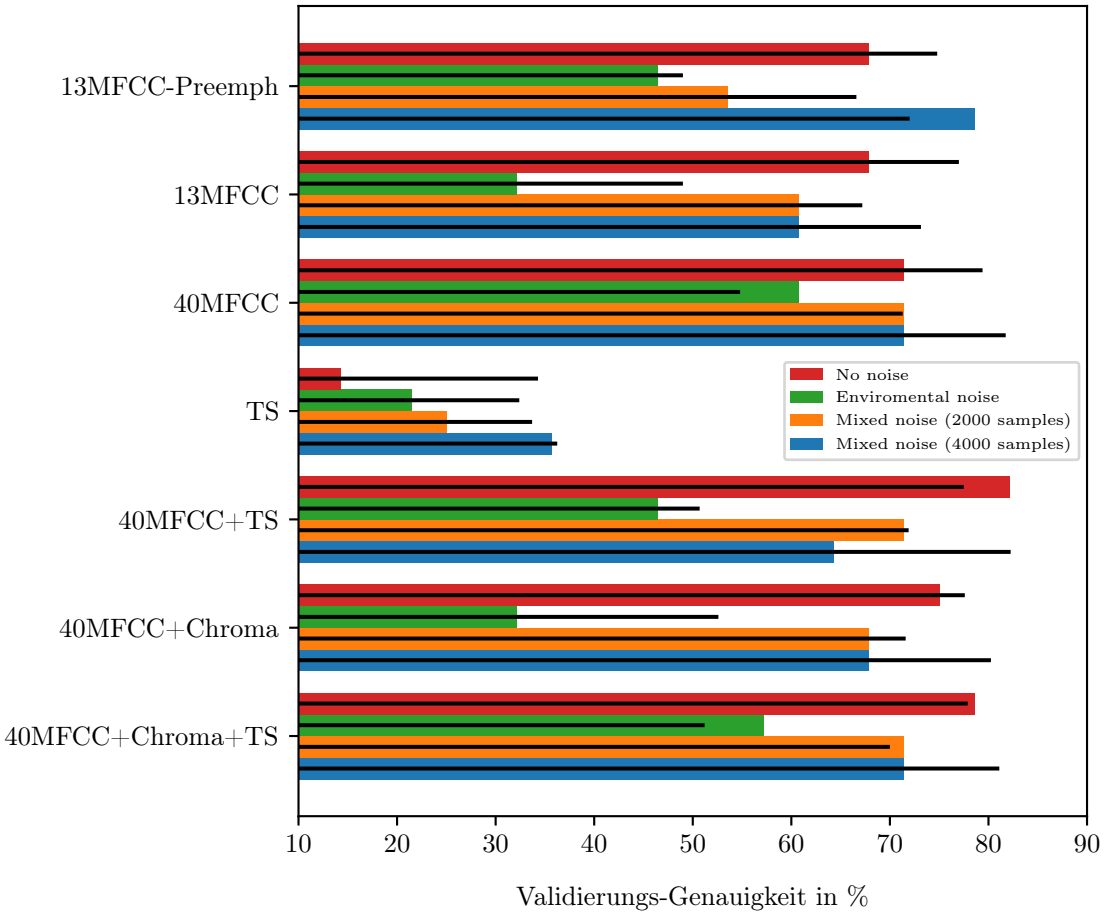


Abb. 18: Validierungs-Genauigkeit der sieben Feature-Sets in Abhängigkeit von dem Vorhandensein von Umgebungsgeräuschen. Die Werte ergeben sich aus der Validierung der jeweiligen Modellkonfiguration mit den für das Training vorenthaltenen Daten; die Validierungs-Genauigkeit der Testdaten für das Training ist als schwarze Linie dargestellt. Die Abkürzung `TS` steht für das Feature-Set der zeit- und frequenzabhängigen Features. Die genauen Zahlenwerte finden sich in Anhang B auf Seite 49 in Tabelle 8.

In diesem Fall sind die Ergebnisse deutlich besser als die Ergebnisse bei der Validierung des CAER-Datensatzes. Vergleicht man zudem die erzielten Werte mit der Validierungs-Genauigkeit der Trainingskonfiguration, so werden durchschnittlich die gleichen Werte erreicht. Dies zeigt, dass die trainierten Modelle durchaus robust sind. Wichtig ist hier anzumerken, dass die jeweilige Trainingskonfiguration mit trainingsähnlichen Daten validiert wurde.

9. Zusammenfassung

Die durchgeführten Betrachtungen und generierten Ergebnisse dieser Arbeit zeigen eindeutig, dass sich Emotionen anhand der menschlichen Stimme erkennen lassen. Mit dem Erreichen einer maximalen durchschnittlichen Validierungs-Genauigkeit von **81,2 Prozent** wurde außerdem gezeigt, dass sich rekurrente neuronale Netze durchaus eignen, um dieser Problemstellung gerecht zu werden. Damit liegt diese Genauigkeit zwar unter der in Kapitel 2 vorgestellten SVM-Klassifizierung von 86,0 Prozent [9], übertrifft aber gleichzeitig die der anderen vorgestellten Forschungsarbeiten, welche ein neuronales Netz zur Klassifizierung nutzen.

Es wurde weiterhin dargestellt, dass sich der Einsatz eines residualen Netzwerkes mit LSTM-Zellen bewährt und im Gegensatz zu herkömmlichen rekurrenten Netzen eine Effizienzsteigerung bewirkt. Ein rekurrentes Netz erreicht in der Arbeit von V. Chernykh et al. bei einer Einteilung in vier Emotionsklassen gerade einmal eine Validierungs-Genauigkeit von 54,0 Prozent [8].

Die in dieser Arbeit genutzten sieben Feature-Sets unterscheiden sich sehr deutlich. Am besten schnitt dabei das Feature-Set mit 40 MFCC-Features ab. Dies zeigt, dass der Standardwert von 13 MFCC-Features nicht unbedingt immer optimal ist. Eher ungeeignet sind dagegen die in dem Feature-Set der zeit- und frequenzabhängigen Features enthaltenen Merkmale. Eine Kombination mehrerer Features kann eine Verbesserung bringen, kann aber unter Umständen das Ergebnis auch verschlechtern.

Eine Validierung der generierten Modelle mit dem multimodalen Datenset CAER hat gezeigt, dass diese für einen Einsatz in einem sehr störbehaftetem realem Umfeld eher ungeeignet sind. Hier ist der Einsatz einer multimodalen Emotionserkennung angebracht, um auch Merkmale wie die Mimik und Gestik in die Emotionserkennung mit einfließen zu lassen. Die in dieser Arbeit erstellten Modelle zur Emotionserkennung anhand der Sprache eignen sich für eine Verwendung in einem solchen multimodalen System und können zu robusteren Aussagen über eine vorliegende Emotion führen. Dieser Aspekt eignet sich in jedem Fall als Grundlage für weiterführende Forschungsarbeiten.

Insgesamt zeigt diese Arbeit und die Beschäftigung mit anderen Forschungsergebnissen auch, dass es trotz des Erreichens einer hohen Aussagegenauigkeit der aktuelle Stand der Forschung nicht erlaubt, Systeme zur Emotionserkennung für folgenreiche Entscheidungen zu benutzen. Die im Rahmen dieser Arbeit gewonnenen Erkenntnisse bilden dennoch einen kleinen Baustein auf dem Weg hin zu einer robusten computergestützten Emotionserkennung.

A. Netzgraphen

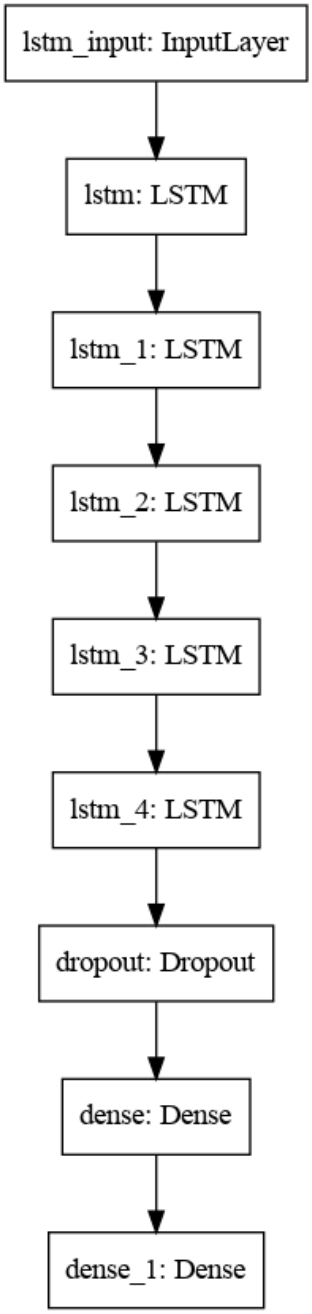


Abb. 19: Rekurrentes Netz mit 5 LSTM-Schichten

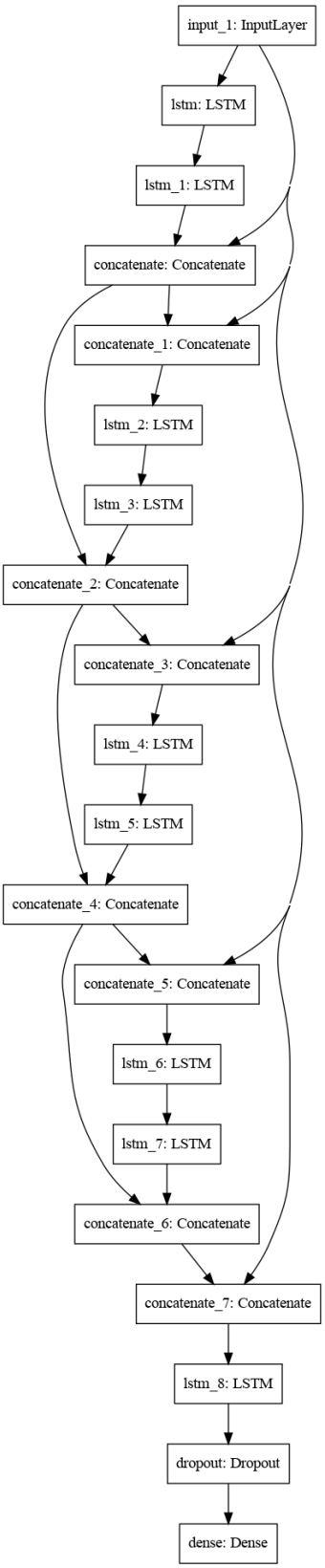


Abb. 20: Strukturmodell des LSTM-RESNETs. Dieses Modell ist das finale, im Rahmen dieser Arbeit entwickelte Modell, da es hinsichtlich der Validierungs-Genauigkeit die besten Ergebnisse erzielte.

B. Ergebnistabellen

Anzahl der LSTM-Zellen	32	64	128	256	512	1028
Validierungs-Genauigkeit in % - 5LSTM	51,0	60,7	67,6	71,9	68,6	66,4
Validierungs-Genauigkeit in % - RESNET	55,2	66,4	71,2	74,5	77,0	75,4

Tab. 5: Validierungs-Genauigkeit in Abhängigkeit von der Anzahl der Zellen pro LSTM-Schicht. Dabei wurden die entsprechenden Werte zum einen für ein rekurrentes Netz mit fünf LSTM-Schichten (5LSTM) ermittelt, zum anderen für das residuale rekurrente Netz (RESNET). Hierbei wurde der jeweilige Maximalwert nach dem Durchlaufen von 50 Epochen festgehalten.

	Validierungs-Genauigkeit in %		
Feature-Set	Englisch	Deutsch	Deutsch + Englisch
13MFCC-Preemph	80,2	69,0	74,8
13MFCC	79,2	72,2	77,0
40MFCC	81,4	82,4	79,4
TS	29,0	23,0	34,3
40MFCC+TS	80,0	82,6	77,5
40MFCC+Chroma	81,4	79,0	77,6
40MFCC+Chroma+TS	79,0	78,0	77,9

Tab. 6: Validierungs-Genauigkeit in Abhängigkeit von der Sprache des Datensatzes nach Durchlaufen von 50 Epochen. Die Abkürzung `TS` steht dabei für das Feature-Set der zeit- und frequenzabhängigen Features.

	Validierungs-Genauigkeit in %			
Feature-Set	No noise	Enviromental Noise	Mixed Noise (2000 samples)	Mixed Noise (4000 Samples)
13MFCC-Preemph	17,14	22,86	17,14	20,0
13MFCC	11,43	14,29	11,43	11,43
40MFCC	11,43	11,43	11,43	25,71
TS	17,14	20,0	20,0	5,71
40MFCC+TS	8,57	11,43	22,86	14,29
40MFCC+Chroma	11,43	11,43	22,86	11,43
40MFCC+Chroma+TS	5,71	20,0	14,29	17,14

Tab. 7: Validierungs-Genauigkeit der sieben Feature-Sets in Abhängigkeit von dem Vorhandensein von Umgebungsgeräuschen. Die Werte ergeben sich aus der Validierung der jeweiligen Modellkonfiguration mit 35 Samples aus dem CAER-Datensatz. Die Abkürzung TS steht für das Feature-Set der zeit- und frequenzabhängigen Features.

	Validierungs-Genauigkeit in %			
Feature-Set	No noise	Enviromental Noise	Mixed Noise (2000 samples)	Mixed Noise (4000 Samples)
13MFCC-Preemph	67,86	46,43	53,57	78,57
13MFCC	67,86	32,14	60,71	60,71
40MFCC	71,43	60,71	71,43	71,43
TS	14,29	21,43	25,0	35,71
40MFCC+TS	82,14	46,43	71,43	64,29
40MFCC+Chroma	75,0	32,14	67,86	67,86
40MFCC+Chroma+TS	78,57	57,14	71,43	71,43

Tab. 8: Validierungs-Genauigkeit der sieben Feature-Sets in Abhängigkeit von dem Vorhandensein von Umgebungsgeräuschen. Die Werte ergeben sich aus der Validierung der jeweiligen Modellkonfiguration mit den für das Training vorenthaltenen Daten; die Validierungs-Genauigkeit der Testdaten für das Training ist als schwarze Linie dargestellt. Die Abkürzung TS steht für das Feature-Set der zeit- und frequenzabhängigen Features.

C. Erklärung des Repositorys auf dem beigefügten Datenträger

Das Repository auf dem beigefügten Datenträger enthält vier Ordner, in welchem sich alle im Rahmen dieser Arbeit erstellten Python-Skripte befinden.

Im Ordner `graphs` finden sich die Python-Skripte zur Erstellung der Diagramme für die schriftliche Ausarbeitung. Enthalten sind dort außerdem die jeweils zugehörigen Wertetabellen im `.csv` Format. Der Ordner `lib` enthält Bibliotheksdateien für die Feature-Extraktion. Die generierten Python-Skripte für das Preprocessing der Audio-Samples sowie die Skripte für die Validierung der einzelnen Netzkonfigurationen sind im Ordner `scripts` gespeichert. Im letzten Ordner `training_scripts` finden sich die Skripte für das Training der rekurrenten neuronalen Netze.

Alle beigefügten Dateien und Ordner sind außerdem über folgenden Link auf GitHub einsehbar:

`https://github.com/Samykolon/Master`

D. Abbildungs- und Tabellenverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] Der Fokus unseres Lebens: Gedanken und Emotionen. https://gedankenwelt.de/der-fokus-unseres-lebens-gedanken-und-emotionen/. Zuletzt aufgerufen: 25.09.2020.

[2] Sabine Hockling. Wir sind auf Fehler fokussiert. https://www.zeit.de/karriere/beruf/2015-08/positives-denken-karriere-job. Zuletzt aufgerufen: 05.10.2020.

[3] How Emotion-Detection Technology Will Change Marketing. https://blog.hubspot.com/marketing/emotion-detection-technology-marketing. Zuletzt aufgerufen: 25.09.2020.

[4] Anil K. Jain and Stan Z. Li. *Handbook of Face Recognition*. Springer-Verlag Berlin Heidelberg, 2005.

[5] Mickaël Ménard, Paul Richard, Hamza Hamdi, Bruno Daucé, and Takehiko Yamaguchi. Emotion recognition based on heart rate and skin conductance. *PhyCS 2015 - 2nd International Conference on Physiological Computing Systems*, 2015.

[6] Kannan Venkataramanan and Haresh Rengaraj Rajamohan. Emotion recognition from speech. *ArXiv*, 2019.

[7] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Aaron Courville, Pascal Vincent, Roland Memisevic, Christopher Pal, and Y. Bengio. EmoNets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 2015.

[8] Vladimir Chernykh, Grigoriy Sterling, and Pavel Prihodko. Emotion Recognition From Speech With Recurrent Neural Networks. *ArXiv*, 2017.

[9] Martin Gjoreski and Hristijan Gjoreski. Machine Learning Approach for Emotion Recognition in Speech. *Informatica*, 2014.

[10] Klaus Rothermund and Andreas Eder. *Motivation und Emotion (Basiswissen Psychologie*. VS Verlag, 2001.

[11] Georgi Schischkoff. *Wörterbuch der Philosophie*. Kröner Stuttgart, 1991.

[12] Sven Barnow, Eva Reinelt, and Christina Sauer. *Emotionsregulation: Manual und Materialien für Trainer und Therapeuten*. Springer-Verlag Berlin Heidelberg, 2016.

[13] Arvid Kappas, Ursula Hess, and Klaus Scherer. *Voice and Emotion*, pages 200–234. 01 1991.

[14] Thomas M. Scherer. *Stimme, Emotion und Psyche - Untersuchungen zur emotionalen Qualität der menschlichen Stimme*. PhD thesis, Universität Marburg, 2000.

[15] Curt Elze. *Periphere Leitungsbahnen II Haut und Sinnesorgane Vegetatives Nervensystem*. Springer-Verlag Berlin Heidelberg, 1940.

[16] Alan Cowen, Hillary Elfenbein, Petri Laukka, and Dacher Keltner. Mapping 24 Emotions Conveyed by Brief Human Vocalization. *American Psychologist*, 12 2018.

[17] Klaus Scherer, Rainer Banse, and Harald Wallbott. Emotion Inferences from Vocal Expression Correlate Across Languages and Cultures. *Journal of Cross-cultural Psychology*, 32:76–92, 01 2001.

[18] Paul Ekman. Universal Facial Expressions of Emotion. *California Mental Health Research Digest*, 1971.

[19] Carroll E. Izard. *Human Emotions*. Plenum Press New York, 1977.

[20] Jonathan Posner, James A. Russell, and Bradley S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 2005.

[21] Wilhelm Wundt. *Grundzüge der physiologischen Psychologie*. Engelmann, 1874.

[22] Klaus Scherer and Grazia Ceschi. Lost Luggage: A Field Study of Emotion–Antecedent Appraisal. *Motivation and Emotion*, 09 1997.

[23] G. N. Yannakakis, R. Cowie, and C. Busso. The Ordinal Nature of Emotions. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017.

[24] Meinard Müller. *Fundamentals of Music Processing - Audio, Analysis, Algorithms, Applications*. Springer-Verlag Berlin Heidelberg, 2015.

[25] Hubert Weber and Helmut Ulrich. *Laplace-, Fourier- und z-Transformation - Grundlagen und Anwendungen für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg, 2011.

[26] Grundlagen der Fourier-Transformation. `https://www.eit.hs-karlsruhe.de/mesysto/teil-a-zeitkontinuierliche-signale-und-systeme/spektrum-eines-signals/grundlagen-der-fourier-transformation/definitionsgleichung-der-fourier-transformation.html`. Zuletzt aufgerufen: 05.08.2020.

[27] Wilhelm Burger and Mark Burge. *Die diskrete Kosinustransformation (DCT)*. Springer Professional, 2005.

[28] L. Kerkeni, Y. Serrestou, M. Mbarki, K. Raoof, M. A. Mahjoub, and C. Cleder. Automatic Speech Emotion Recognition Using Machine Learning. In *Social Media and Machine Learning*. InTechOpen, 2019.

[29] S. Lalitha, D. Geyasruti, R. Narayanan, and Shravani M. Emotion Detection Using MFCC and Cepstrum Features. *Procedia Computer Science*, 70, 2015.

[30] Gustavo Tomas. Speech Emotion Recognition Using Convolutional Neural Networks. Master's thesis, Technische Universität Berlin, 2019.

[31] Gebremedhin T. Abreha. An Environmental Audio–Based ContextRecognition System Using Smartphones. Master's thesis, Universität Stuttgart, 2011.

[32] Sid-Ahmed Selouani (auth.). *Speech Processing and Soft Computing*. Springer-Verlag New York, 2011.

[33] Santhy Viswam and Sajeer Karattil. A Novel Approach for Software Requirement Specification. *International Journal of Computer Applications*, 2013.

[34] S. Stevens, J. Volkmann, and E. B. Newman. A Scale for the Measurement of the Psychological Magnitude Pitch. 1937.

[35] pyAudioAnalysis - A Python library for audio feature extraction, classification, segmentation and applications. `https://github.com/tyiannak/pyAudioAnalysis`. Zuletzt aufgerufen: 09.09.2020.

[36] D. S. Shete and S. B. Patil. Zero crossing rate and Energy of the Speech Signal of Devanagari Script. *IOSR journal of VLSI and Signal Processing*, 2014.

[37] Bachu R.G, Kopparthi S., Adapa B., and Buket Barkana. *Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy.* 2010.

[38] M. F. Pradier. Emotion Recognition from Speech Signals and Perception of Music. Master's thesis, University of Twente, 2014.

[39] Park T.H. *Introduction to digital signal processing: Computer musically speaking.* World Scientific, 2010.

[40] Theodoros Giannakopoulos and Aggelos Pikrakia. Introduction to Audio Analysis. Academic Press Oxford, 2014.

[41] William Sethares, Robin Morris, and James Sethares. Beat tracking of musical performances using low-level audio features. *Speech and Audio Processing, IEEE Transactions*, 2005.

[42] Ayush Shah, Manasi Kattel, Araju Nepal, and D. Shrestha. Chroma feature extraction. 2019.

[43] Prajakta Dahake, Kailash Shaw, and P. Malathi. Speaker dependent speech emotion recognition using MFCC and Support Vector Machine. 09 2016.

[44] Yelin Kim and Emily Mower Provost. Emotion classification via utterance-level dynamics: A pattern-based approach to characterizing affective expressions. 2013.

[45] Nils Kornfeld. Optimierung eines neuronalen Netzes zur Objekterkennung unter Verwendung evolutionärer Algorithmen. Master's thesis, Freie Universität Berlin, 2017.

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

[47] Patrick Henry Winston. *Artificial Intelligence (3rd Ed.).* Addison-Wesley Longman Publishing Co., Inc., 1992.

[48] Rudolf Kruse, Christian Borgelt, Christian Braune, Frank Klawonn, Christian Moewes, and Matthias Steinbrecher. *Computational Intelligence Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze.* Springer Vieweg, 2015.

[49] Was ist ein rekurrentes neuronales Netz (RNN)? `https://www.bigdata-insider.de/was-ist-ein-rekurrentes-neuronales-netz-rnn-a-843274`. Zuletzt aufgerufen: 10.09.2020.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural computation*, 1997.

[51] Felix Burkhardt, Astrid Paeschke, M. Rolfes, Walter Sendlmeier, and Benjamin Weiss. A database of German emotional speech. 2005.

[52] Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). 2018.

[53] Michael Neumann and Ngoc Thang Vu. Attentive Convolutional Neural Network Based Speech Emotion Recognition: A Study on the Impact of Input Features, Signal Length, and Acted Speech. *Interspeech 2017*, 2017.

[54] Environmental Sound Classification. `https://www.kaggle.com/mmoreaux/environmental-sound-classification-50`. Zuletzt aufgerufen: 15.09.2020.

[55] python_speech_features. `https://github.com/jameslyons/python_speech_features`. Zuletzt aufgerufen: 14.09.2020.

[56] Tensorflow - An end-to-end open source machine learning platform. `https://www.tensorflow.org/`. Zuletzt aufgerufen: 05.10.2020.

[57] Context-Aware Emotion Recognition. `https://caer-dataset.github.io/`. Zuletzt aufgerufen: 15.09.2020.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. 2016.

[59] Y. Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 1994.