Fakultät I Geisteswissenschaften
Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation

Masterarbeit

Musikemotionserkennung durch Deep Learning auf Grundlage von audio- und textbasierten Informationen

vorgelegt von:

Philipp Scholze [redacted]

1. Gutachter: Prof. Dr. Stefan Weinzierl
2. Gutachter: Roman Gebhardt

Technische Universität Berlin, Fakultät I – Institut für Sprache und Kommunikation,
Fachgebiet Audiokommunikation
Berlin, 30. September 2021

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Fakultät I der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein. Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden.

Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Ort, Datum	Unterschrift

Zusammenfassung

Aufgrund des rasanten Wachstums digitaler Musikbibliotheken kann sich die Organisation und Bereitstellung dieser Daten problematisch gestalten. Die Handhabung solcher enormen Datenmengen bedarf neuer Methoden, wovon die automatische Musikemotionserkennung (engl. Music Emotion Recognition, kurz MER) einen interessanten Ansatz darstellt. In der Vergangenheit konnten vielversprechende Ergebnisse für diese Aufgabe durch die Betrachtung von audio- und textbasierten Informationen erzielt werden. Vor allem haben sich in jüngster Zeit Deep-Learning-Modelle als nützlich für die automatische MER erwiesen. Jedoch werden für die Entwicklung von aussagekräftigen Deep-Learning-Modellen in der Regel umfangreiche Datensätze benötigt, welche im Bereich der MER selten sind. Mit dem Konzept des Transfer Learnings kann die Entwicklung von aussagekräftigen Deep-Learning-Modellen auch bei dürftiger Datenlage begünstigt werden. Da bisher genauere Untersuchungen von Transfer-Learning-Strategien in der automatischen MER fehlen, wurden in dieser Arbeit drei unterschiedliche Modelle mit Transfer Learning betrachtet: ein auf Audiodaten gestütztes Modell („AudioNet“), ein textbasiertes Modell („LyricsNet“) und ein Modell, welches sowohl audio- als auch textbasierte Daten verarbeitet („FusionNet“). Diese Modelle wurden mit entsprechenden Modellen ohne Transfer Learning bezüglich ihrer Performanz und ihres Implementierungsaufwandes verglichen.

Es konnte gezeigt werden, dass das LyricsNet-Modell eine höhere Performanz und damit einen positiven Transfer aufweist als das vergleichbare Modell ohne Transfer Learning. Für die AudioNet- und FusionNet-Modelle wurde hingegen ein konträres Verhalten beobachtet (negativer Transfer), wodurch diese Modelle eine schlechtere Performanz aufwiesen. Das Auftreten des negativen Transfers wurde diskutiert und potentielle Lösungsansätze vorgestellt. Weiterhin konnte demonstriert werden, dass der Implementierungsaufwand durch den Einsatz von Transfer-Learning-Modellen erheblich reduziert werden kann.

Abstract

Due to the rapid growth of digital music libraries, the organization and accessibility of the emerging data can be problematic. Handling such enormous amounts of data requires new methods, of which automatic music emotion recognition (MER) is an interesting approach. In the past, promising results for this task could be achieved by considering audio- and text-based information. Most notably, deep learning models have recently been shown to be useful for automatic MER. However, the development of meaningful deep learning models usually requires large datasets, which are rare in the field of MER. With the concept of transfer learning, the development of meaningful deep learning models can be favored even with sparse data. Since more detailed studies of transfer learning strategies in automatic MER have been lacking so far, three different models with transfer learning were considered in this work: a model based on audio data ("AudioNet"), a text-based model ("LyricsNet"), and a model that processes both audio and text-based data ("FusionNet"). These models were compared with corresponding models without transfer learning for their performance and implementation effort. It could be shown that the LyricsNet model has a higher performance and thus a positive transfer than the comparable model without transfer learning. For the AudioNet and FusionNet models, on the other hand, a contrary behavior was observed (negative transfer), resulting in poorer performance for these models. The occurrence of negative transfer was discussed and potential solutions were presented. Furthermore, it was demonstrated that the implementation effort can be significantly reduced by using transfer learning models.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	**Bedeutung**
Acc	Accuracy (Genauigkeit)
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DenseNet	Dense Convolutional Network
DL	Deep Learning
DNN	Deep Neural Network
LR	Learning Rate
LSTM	Long Short-Term Memory
MER	Music Emotion Recognition
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
ML	Machine Learning
MSD	Million Song Dataset
NLP	Natural Language Processing
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RW	Random Weights (Modelle ohne Transfer Learning)
TL	Transfer Learning

Kapitel 1

Einleitung

1.1 Motivation und Zielsetzung

Musik ist ein integraler Bestandteil unserer Gesellschaft und allgegenwärtig. Sie begleitet uns in den meisten alltäglichen Situationen: Wir hören Musik morgens am Frühstückstisch, beim Autofahren, im Supermarkt während wir einkaufen, zum konzentrierten Lernen und beim Sport machen (Y.-H. Yang & Chen, 2012). Musik ist aber vor allem auch durch den Aufstieg von Streaming-Anbietern wie Spotify, Apple Music oder Deezer auf mobilen Endgeräten heutzutage nahezu unbegrenzt und jederzeit verfügbar (Schedl et al., 2014, S. 129). Das enorme und stetige Wachstum digitaler Musikbibliotheken wurde in den letzten Jahren nicht nur durch kompakte Audioformate wie dem MP3-Format, sondern auch durch das Internet maßgeblich begünstigt und beschleunigt. Die Handhabung von solch großen Datenmengen stellt jedoch neben der Musikindustrie ebenfalls ihre Anwender*innen vor unbekannte Probleme und Herausforderungen. So sind umfangreiche Musikbibliotheken erst dann für Anwender*innen nützlich, wenn sie sich leicht organisieren lassen und einfach Musik bereitstellen können. Aus diesem Grund erscheinen beispielsweise Suchanfragen, welche bereits Wissen über Interpret*in, Albumnamen oder Genre voraussetzen, dieser Tage weniger zielführend zu sein (Wieczorkowska et al., 2006). Es gilt also, neue Ansätze zu finden, welche die Organisation und Verfügbarkeit von digitaler Musik vereinfachen und/oder erweitern können.

Da Musik von Menschen vor allem zur Regulation der eigenen Gefühlslage bzw. Stimmung[1] gehört wird (Schäfer, Sedlmeier, Städtler & Huron, 2013), findet sich in der automatischen Musikemotionserkennung (engl. Music Emotion Recognition, kurz MER) ein vielversprechender Lösungsansatz für die oben beschriebene Problematik. Die Hoffnungen liegen hierbei darin, dass Emotionen

[1] Die Begriffe Emotion und Stimmung werden in der einschlägigen Literatur und auch in dieser Arbeit synonym verwendet.

als weitere Indikatoren für das Entdecken, Organisieren und Hören von Musik herangezogen werden können. Dazu sollen Algorithmen entwickelt werden, mit deren Hilfe die von einem Musikstück transportierten Emotionen automatisch vorhergesagt werden können. Allerdings stellt die MER eine recht komplexe und interdisziplinäre, wenngleich auch eine spannende Aufgabe dar. Sie erfordert einerseits Kenntnisse der Signalverarbeitung und des maschinellen Lernens. Andererseits spielen neben der auditiven Wahrnehmung auch Psychologie und Musiktheorie eine ebenso bedeutsame Rolle (Kim et al., 2010). Weiterhin weist Musik einen vielschichtigen Charakter auf: So können sowohl das reine Audiosignal als auch Partituren, Texte (Lyrics), Bilder (Albumcover) oder Gestik (Performer/Interpret:in) mögliche Ausdrucksformen von Musik sein (Schedl et al., 2014, S. 130). Im Zusammenhang mit der automatischen MER waren insbesondere das reine Audiosignal, aber auch Lyrics[2] bereits Gegenstand intensiver Forschungsbemühungen im Verlauf der letzten 20 Jahre (u. a. (Li & Ogihara, 2003; Eerola et al., 2009; Van Zaanen & Kanters, 2010; Malheiro, Oliveira et al., 2016)). Aber auch die Kombination von Audiosignal und Songtexten wurde bereits betrachtet (u. a. (D. Yang & Lee, 2004; Laurier et al., 2008; X. Hu et al., 2009)).

Mit dem Einzug von künstlichen neuronalen Netzwerken in den 2010er-Jahren konnten sogenannte Deep-Learning-Modelle (kurz DL-Modelle) vor allem im Bereich der Computer Vision[3] beträchtliche Erfolge verbuchen, welche letztendlich auch in der automatischen MER untersucht wurden (u. a. (Parisi et al., 2019; T. Liu et al., 2018; Delbouys et al., 2018)). Damit ein DL-Modell jedoch einen guten Prädiktor für eine gewisse Aufgabe darstellen kann, sind im Allgemeinen sehr umfangreiche Datensätze für die Entwicklung eines Modells notwendig, wobei solche Datensätze mitunter mehrere Millionen Einträge umfassen können (z. B. der *ImageNet*-Datensatz (Deng et al., 2009)). Innerhalb der MER sind Datensätze dieser Größe nicht realistisch, da in diesem Bereich die Erstellung aussagekräftiger Datensätze in der Regel mit einem erheblichen zeitlichen und finanziellen Aufwand verbunden ist, wodurch sich ihr Umfang meistens auf wenige hundert bis tausend Einträge beschränkt (z. B. (Çano & Morisio, 2017)).

Mit dem Konzept des Transfer Learnings wird versucht, die Probleme, welche bei der Entwicklung von DL-Modellen mit kleineren Datensätzen auftreten können, zu überwinden. Transfer Learning beruht auf der Idee, dass Wissen

[2] Analog zu Emotion und Stimmung werden in dieser Arbeit die Begriffe Lyrics und Songtext synonym verwendet.

[3] Die Computer Vision beschäftigt sich mit der Extraktion von Informationen aus visuellen Daten.

zwischen verschiedenen Anwendungsbereich übertragen werden kann (Zhuang et al., 2021). Im Kontext von DL-Modellen bedeutet dies, dass ein Modell zunächst mit einen umfangreichen Datensatz auf eine spezifische Aufgabe trainiert und anschließend mit einem kleineren Datensatz auf eine andere, aber ähnliche Aufgabe abgestimmt wird (u. a. durch sogenanntes *fine-tuning*). Von diesem Ansatz wird sich erhofft, dass die Performanz eines DL-Modells bezüglich einer neuen Aufgabe verbessert werden kann, indem auf Wissen aus einem anderen Bereich zurückgegriffen wird. Obwohl Transfer-Learning-Strategien mit DL-Modellen bereits in verschiedenen Bereichen wie der Genre-Klassifikation von Musikstücken erfolgreich angewandt wurde (Palanisamy, Singhania & Yao, 2020), fehlt eine solche Betrachtung innerhalb der automatischen MER fast vollständig. Zielsetzung der vorliegenden Arbeit ist es daher, an diesem Punkt anzuknüpfen und einen Ansatz für die fehlende Perspektive vorzuschlagen. Die daraus resultierende Forschungsfrage ist im folgenden Abschnitt 1.2 formuliert.

1.2 Formulierung der Forschungsfrage

Mit Ausnahme von Agrawal, Shanker und Alluri (2021) sowie G. Liu und Tan (2020), welche in einem ersten Versuch einen Transfer-Learning-Ansatz für die automatische MER mit textbasierten Informationen in Form von Songtexten untersuchten, sind bislang keine weiteren Arbeiten bekannt, welche Transfer-Learning-Strategien mit audio- und textbasierten Daten betrachtet haben. Auch die Kombination aus Audiosignal und Lyrics wurde im Zusammenhang mit Transfer Learning noch nicht ausgiebig erforscht. In dieser Arbeit sollen deshalb Deep-Learning-Modelle für die automatische Musikemotionserkennung mit Transfer Learning näher beleuchtet werden, wodurch ein Ausgangspunkt für weitere Forschung gelegt werden kann. Die Datenlage umfasst dabei Audiosignale und Lyrics von westlichen, kontemporären Musikstücken, welche einzeln und kombiniert als Grundlage für die Entwicklung der DL-Modelle verwendet werden sollen. Die entsprechende Fragestellung, welche sich aus diesen Betrachtungen ergibt und im Folgenden näher untersucht und beantwortet werden soll, lautet daher:

Kann mithilfe von Transfer Learning die Performanz von Deep-Learning-Modellen für die automatische Musikemotionserkennung im Vergleich zu entsprechenden DL-Modellen ohne Transfer Learning verbessert werden?

Darüber hinaus sollen bei der Beantwortung der Forschungsfrage auch die folgenden Aspekte näher erläutert werden:

- Art der verwendeten Daten (sowohl audio- und textbasiert, als auch die Kombination aus beiden),
- Auswahl der DL-Modelle,
- Implementierungs- und Rechenaufwand.

1.3 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich in folgende Abschnitte. In Kapitel 2 wird zunächst ein Überblick über die theoretischen Grundlagen gegeben. So soll in Abschnitt 2.1 die Beziehung zwischen Musik und Emotionen näher erläutern, während in Abschnitt 2.2 verschiedene Emotionsmodelle vorgestellt werden. Weiter wird das Forschungsfeld der Musikemotionserkennung beleuchtet (Abschnitt 2.3) und aktuelle Forschungsergebnisse mit Deep-Learning-Ansätzen vorgestellt (Abschnitt 2.4). Abschließend wird in Abschnitt 2.5 das Konzept des Transfer Learning erklärt. Kapitel 3 beschreibt die verwendeten Methoden. In den Abschnitten 3.1, 3.2 und 3.3 werden die in dieser Arbeit untersuchten DL-Modelle im Detail beschrieben und vorgestellt. Die Abschnitte 3.4 und 3.5 geben einen Überblick über den verwendeten Datensatz und die Vorverarbeitung der Daten. Nachdem in Abschnitt 3.6 die Trainingsparameter für die Modelle aufgelistet werden, schließt Abschnitt 3.7 mit einer Definition der Domänen für das Transfer Learning ab. In Kapitel 4 werden die Versuchsergebnisse vorgestellt und diskutiert. Die Diskussion geht dabei auf die in der Forschungsfrage genannten Aspekte ein. Zum Schluss dieser Arbeit wird in Kapitel 5 ein Fazit gezogen und weitere Ausblicke für die Zukunft gegeben.

Kapitel 2

Theorieteil

Wie eingangs bereits erwähnt, soll in diesem Kapitel eine Einführung in die automatische Musikemotionserkennung gegeben werden. Nachdem zunächst die Beziehung zwischen Musik und Emotionen in Abschnitt 2.1 und die in der Literatur verwendeten Emotionsmodelle in Abschnitt 2.2 näher beschrieben werden, folgt eine Übersicht über die automatische MER in Abschnitt 2.3. Deep-Learning-Ansätze in der MER werden in Abschnitt 2.4 behandelt. Abschließend findet sich eine Definition von Transfer Learning in Abschnitt 2.5.

2.1 Musik und Emotionen

Obwohl seit der von William James im Jahr 1884 gestellten Frage „Was ist eine Emotion?“[4] eine einheitliche und allgemein akzeptierte Definition von Emotionen bislang fehlt, konnten verschiedene Untersuchungen zeigen, dass Musik und Emotionen eng miteinander verknüpft sind und in einer komplexen Wechselbeziehung zueinander stehen. So lassen sich Ausdruck, Wahrnehmung und Induktion von Emotionen durch Musik unterscheiden (Panda, 2019, S. 21). Musikschaffende und Ausführende können dabei gezielt Stimmungen durch ihre Musik ausdrücken und so zu einer Zuhörerschaft transportieren. Auf Seite der Zuhörerschaft werden diese Emotionen wahrgenommen und auf gelernte Weise (unterbewusst) identifiziert. Oft wird hier jedem Musikstück eine vorherrschende Stimmung (zum Beispiel Traurigkeit, Fröhlichkeit etc.) von der Zuhörerschaft zugeordnet (Malheiro, Panda, Gomes & Paiva, 2016), wobei sich diese vorherrschende Stimmung durchaus von der beabsichtigten Emotion unterscheiden kann. So kann beobachtet werden, dass beispielsweise traurige Musikstücke von der Zuhörerschaft als angenehm wahrgenommen werden, welches auch als „Paradoxon negativer Emotionen“ bezeichnet wird (Pannese, Rappaz & Grandjean, 2016, S. 7). Weiterhin kann Musik auch auf individueller Ebene verschie-

[4] William James, II.—What is an Emotion?, Mind, Volume os-IX, Issue 34, 1 April 1884, Seiten 188–205.

Emotion	Musikalische Merkmale
Fröhlichkeit	U.a. schnelles Tempo, Dur-Tonraum, einfache und konsonante Harmonik, mittlere bis hohe Lautstärke, helle Klangfarben.
Traurigkeit	U.a. langsames Tempo, Moll-Tonraum, dissonante Harmonik, niedrige Lautstärke, düstere Klangfarben.
Wut	U.a. schnelles Tempo, Moll-Tonraum, Atonalität, dissonante Harmonik, hohe Lautstärke, scharfe Klangfarben.
Angst	U.a. schnelles Tempo, große Tempovariabilität, Moll-Tonraum, dissonante Harmonik, niedrige Lautstärke, rasche Lautstärkewechsel, gedämpfte Klangfarben.
Zärtlichkeit	U.a. langsames Tempo, Dur-Tonraum, konsonante Harmonik, mittlere bis geringe Lautstärke, sanfte Klangfarben.

Tabelle 2.1: Fünf der am häufigsten untersuchten Emotionen und ihre zugeschriebenen musikalischen Merkmale nach Juslin und Lindström (2010).

dene Stimmungen bei einer zuhörenden Person auslösen (Induktion) (Sloboda & Juslin, 2001, S. 32). Diese Unterscheidungen machen deutlich, dass Kontext für die Beschreibung von Emotionen elementar ist. Aus diesem Grund soll an dieser Stelle darauf hingewiesen werden, dass im weiteren Verlauf dieser Arbeit (siehe Kapitel 3 und 4) immer von durch eine Zuhörerschaft wahrgenommenen Emotionen gesprochen wird.

In Anbetracht der engen Verbindung von Musik und Emotionen wurde in der Vergangenheit versucht, bestimmten musikalischen Elementen bzw. Merkmalen entsprechende Stimmungen zuzuweisen. Infolge einer fehlenden, einheitlichen Definition von Emotionen sind diese Zuordnung teilweise widersprüchlich zueinander. Im weiteren Verlauf soll eine Auswahl an musikalischen Elementen und ihrer Verbindung zu Emotionen kurz vorgestellt werden. Ein besonders wichtiges dieser Elemente bilden die Melodien, welchen eine spezielle Rolle in der Vermittlung von Emotionen zugesagt wird. So werden Melodien mit großem tonalen Umfang oft mit Emotionen wie Freude (Balkwill & Thompson, 1999), aber auch Angst (Krumhansl, 1997) verbunden. Melodien mit einer kleineren tonalen Spannweite hingegen können mit Traurigkeit oder Melancholie assoziiert werden (Gundlach, 1935). Darüber hinaus findet sich mit der verwendeten Harmonik in einem Musikstück ein weiteres Merkmal, welchem Emotionen zugeordnet werden können. Eine der simpelsten Beschreibungen von Harmonik und Emotionen ist die Verknüpfung von konsonanten Klängen mit positiven und dissonanten

Klängen mit negativen Emotionen (Juslin & Lindström, 2010, S. 335). Neben Melodie und Harmonik stellt auch der Rhythmus eines Musikstückes ein weiteres wichtiges Merkmal bezüglich transportierter Emotionen dar. Faktoren wie Tempo oder Art des Rhythmus können unterschiedliche Emotionen auslösen. So wird ein schnelles Tempo als fröhlich oder angenehm, aber auch als wütend oder ängstlich empfunden, während ein langsames Tempo mit heiter oder traurig assoziiert wird. Weiterhin werden komplexe Rhythmen als wütend, gleichmäßige Rhythmen als fröhlich, ernst oder auch friedlich wahrgenommen (Panda, 2019, S. 52-53). Die Dynamik eines Musikstückes steht ebenfalls im Zusammenhang mit Emotionen. So ruft laute Musik intensive Emotionen wie Freude oder Angst hervor, wohingegen leise Musik eher Emotionen wie Trauer oder Gelassenheit bewirken kann (Gundlach, 1935). Weitere Merkmale, welche mit Emotionen in Verbindung gebracht werden können, sind Timbre/Klangfarbe, Spieltechniken (z.B. Legato oder Staccato), musikalische Texturen und Formen (Panda, Malheiro & Paiva, 2020). Eine Zuordnung von verschiedenen musikalischen Merkmalen zu fünf der nach (Juslin & Lindström, 2010) am häufigsten untersuchten Emotionen findet sich in Tabelle 2.1.

Insbesondere westliche, kontemporäre Musikstücke werden häufig durch Lyrics, welche einen gravierenden Einfluss auf die Wahrnehmung von Emotionen haben können, komplementiert (Malheiro, Panda et al., 2016). Songtexte stellen dabei meist einen Verbund aus mehreren Sätzen dar und weisen, analog zur musikalischen Struktur des Stückes, verschiedene Abschnitte wie Strophen, Refrains etc. auf. Im Vergleich zu anderen, textbasierten Medien wie Rezensionen sind Lyrics meist kurz, oft abstrakt und transportieren Emotionen eher implizit als explizit (Y. Hu, Chen & Yang, 2009). Unabhängig von der akustischen oder künstlerischen Umsetzung können jedoch auch einzelne Worte durch ihre reine Bedeutung eine inhärente Emotionen transportieren, wie dies zum Beispiel bei den Worten „fröhlich“ oder „tot“ der Fall ist (Van Zaanen & Kanters, 2010). Dass der emotionale Gehalt von textlichen Informationen wie Lyrics einen Einfluss auf die Emotionswahrnehmung von Musikstücken hat, wurde von (Ali & Peynircioğlu, 2006) in mehreren Experimenten gezeigt. So konnten die negativen Emotionen von traurigen oder wütenden Musikstücken durch ihre Songtexte verstärkt werden. Es ist daher nicht verwunderlich, dass in den vergangenen zwei Jahrzehnten sowohl textliche Informationen in Form von Lyrics (u. a. (Y. Hu et al., 2009; Van Zaanen & Kanters, 2010; Malheiro, Panda et al., 2016), auditive Informationen in Form von Melodie, Harmonik, Rhythmik etc. (u. a. (Li & Ogihara, 2003; Wieczorkowska et al., 2006; Eerola et al., 2009; Malik et al., 2017)) und deren Kombination (u. a. (Laurier et al., 2008; Schuller et al., 2010; Delbouys et al., 2018)) im Zusammenhang mit Emotionen untersucht wurden.

All diesen Arbeiten ist gemein, dass eine bestimmte Form der Konzeptualisierung von Emotionen vorgenommen wurde. Eine solche Konzeptualisierung ist notwendig, um Emotionen, trotz fehlender einheitlicher Definition, mess- und vergleichbar zu machen. Der folgende Abschnitt 2.2 soll deshalb einen Überblick über die gängigsten Emotionsmodelle geben.

2.2 Emotionsmodelle und Taxonomie

Seit Anfang des 20. Jahrhunderts haben sich etliche Emotionsmodelle entwickelt, welche sich näherungsweise in zwei Ansätze aufteilen lassen: Kategorische und dimensionale Modelle (Y.-H. Yang & Chen, 2012). Während kategorische Modelle Emotionen über einzelne Worte oder durch Gruppen von Worten (sogenannten Clustern) abbilden, verfolgen dimensionale Modelle den Ansatz, Emotionen in einem kontinuierlichen, mehrdimensionalen Raum darzustellen.

Ein vergleichsweise einfaches kategorisches Modell bieten die Basisemotionen, welche von Ekman (1992) vorgeschlagen wurden. Dabei wird von einer beschränkten Anzahl von diskreten Emotionen ausgegangen, mit denen der Mensch evolutionsbedingt ausgestattet sei. Bezüglich ihrer psychologischen, physiologischen und verhaltensbezogenen Ausprägungen werden diese Basisemotionen als unabhängig voneinander betrachtet, wobei jede Emotion durch eine spezifische Aktivierungen im zentralen Nervensystem ausgelöst werde (Posner, Russell & Peterson, 2005). So identifizieren Ekman und Friesen (2003, S. 22) Fröhlichkeit (engl. happiness), Traurigkeit (sadness), Angst (fear), Ekel (disgust), Wut (anger) und Überraschung (surprise) als die sechs Basisemotionen, die durch Gesichtsmimik ausgedrückt werden können, wobei weitere Emotionen wie Scham (shame) und Begeisterung (excitement) als Mischung der sechs Basisemotionen verstanden werden können. Aufgrund der recht allgemeinen Beschaffenheit der Basisemotionen wurden weitere, bereichsspezifische Modelle zur Beschreibung von Emotionen in der Musik erarbeitet. Der von Hevner (1936) entwickelte Adjektivkreis (adjective circle) in Abbildung 2.1 stellt dabei ein bekanntes Beispiel für ein bereichsspezifisches, kategorisches Emotionsmodell dar. In dieser wegweisenden Arbeit untersuchte Hevner den affektiven Charakter von verschiedenen Musikstücken, wodurch letztendlich 66 englische Adjektive in acht Gruppen angeordnet wurden, wenngleich sich die Anzahl der Adjektive pro Gruppe unterscheiden kann. Das von X. Hu und Downie (2007) beschriebene Emotionsmodell zeigt dagegen einen neueren, datenbasierten Ansatz. Hier wurden Emotionscluster (siehe Abbildung 2.2) anhand der Auswertung von online verfügbaren Metadaten erstellt. Dieses Modell wurde inzwischen auch für

den MIREX[5] *Audio Mood Classification Task* adaptiert (Panda, 2019, S. 28).

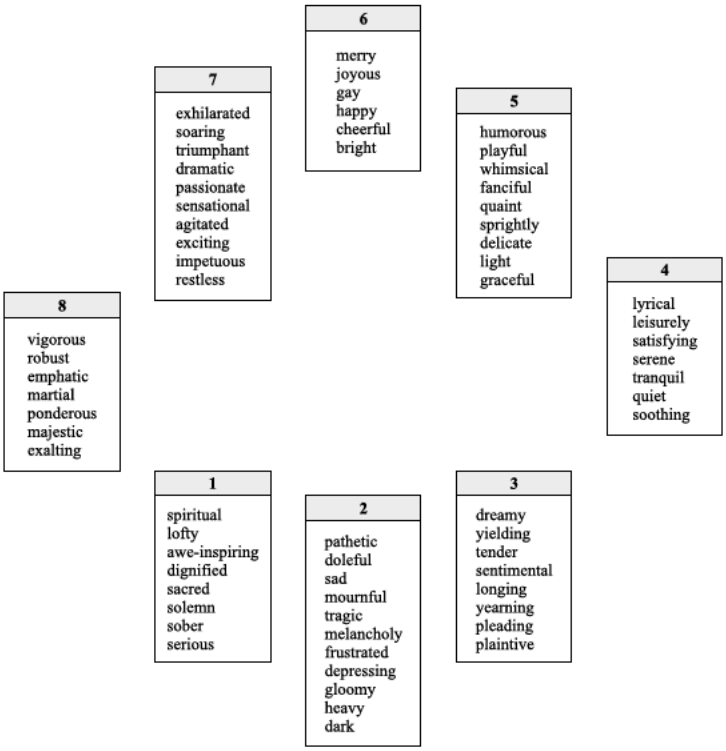


Abbildung 2.1: Adjektivkreis nach Hevner (1936).

Unter den dimensionalen Emotionsmodellen wird häufig das sogenannte „Circumplexmodell“ von Russell (1980) verwendet, welches Emotionen in einem zweidimensionalen Raum abbildet. Die beiden Achsen werden hierbei durch Valenz und Arousal beschrieben (VA-Raum). Unter dem Begriff Arousal wird die Intensität einer Emotion, welche durch einen Stimulus hervorgerufen wird, verstanden. Die Valenz hingegen beschreibt, wie angenehm ein Stimulus empfunden wird (Warriner, Kuperman & Brysbaert, 2013). Eine grafische Veranschaulichung des Circumplexmodells findet sich in Abbildung 2.3. Quadrant 1 wird mit „erfreut“ (engl. joyful), der zweite Quadrant mit „zufrieden“ (content) beschrieben. Für Quadrant 3 findet sich die Beschreibung „bedrückend“ (depressing), wohingegen Quadrant 4 die Emotion „wütend“ (angry) darstellt (Parisi et al., 2019). Das Circumplexmodell nach Russell (1980) hat sich inzwischen laut Panda (2019, S. 28) zum dimensionalen Standardmodell in der MER-Forschung entwickelt.

Zweidimensionale Modelle sind jedoch nicht frei von Kritik. So werden unmittelbar benachbarte Emotionen wie Angst und Wut im zweidimensionalen VA-Raum nicht ausreichend voneinander differenziert (Eerola & Vuoskoski, 2011),

[5] Music Information Retrieval Evaluation eXchange, siehe auch `https://www.music-ir.org/mirex/wiki/MIREX_HOME` , zuletzt abgerufen am 29.09.2021.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Rowdy Rousing Confident Boisterous Passionate	Amiable / Good natured Sweet Fun Rollicking Cheerful	Literate Wistful Bittersweet Autumnal Brooding Poignant	Witty Humorous Whimsical Wry Campy Quirky Silly	Volatile Fiery Visceral Aggressive Tense/anxious Intense

Abbildung 2.2: Cluster-Modell nach X. Hu und Downie (2007).

weshalb von manchen Forschenden eine dritte Achse eingeführt und somit ein zwei- in ein dreidimensionales Modell überführt wurde. Ein populärer Ansatz findet sich in dem von Schimmack und Grob (2000) vorgeschlagenem Modell, welches in Abbildung 2.4 dargestellt ist. Hier wurde die Dimension Arousal in zwei weitere Dimensionen aufgeteilt. Den dreidimensionalen Raum beschreiben die Achsen Anspannung (tense arousal), Energie (energetic arousal) und Valenz.

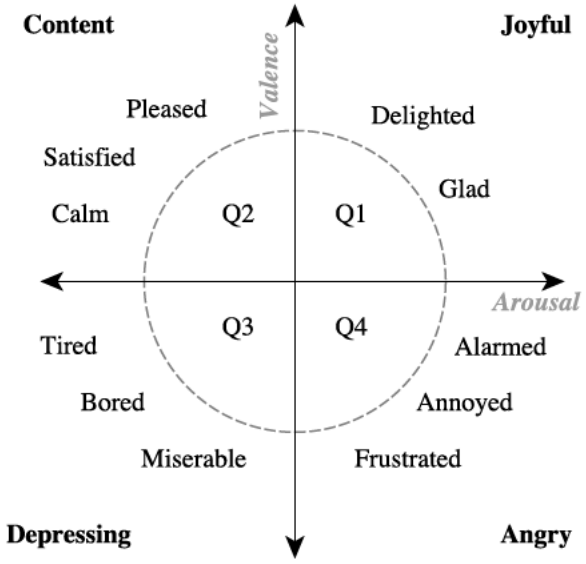


Abbildung 2.3: Circumplexmodell nach Russell (1980). Darstellung in Anlehnung an Parisi et al. (2019).

Die Wahl des Emotionsmodells ist jedoch nicht trivial und immer vom eigentlichen Forschungszweck abhängig. Bereichsspezifische, kategorische Modelle seien nach Panda (2019) besser für induzierte Emotionen geeignet, während sich wahrgenommene Emotionen vorteilhafter durch Basisemotionen ausdrücken ließen. Dimensionale Modelle können dagegen zwar das Problem der Mehrdeutigkeit von Emotionen beheben. Allerdings können sie auch einen weitaus größeren, rechnerischen Aufwand mit sich bringen und werden eventuell nicht so gut von Proband*innen in Hörtests verstanden wie einfache kategorische Emotionsmo-

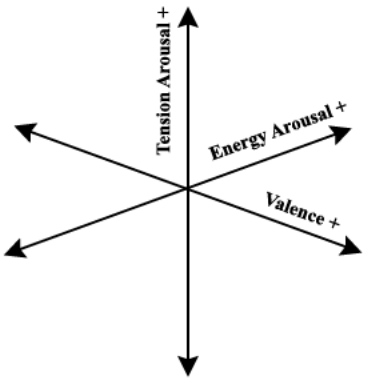


Abbildung 2.4: Dreidimensionales Emotionsmodell nach Schimmack und Grob (2000). Darstellung in Anlehnung an Eerola und Vuoskoski (2011).

delle. Neben den verwendeten Emotionsmodellen muss auch unterschieden werden, ob es sich um eine statische oder dynamische Emotionserkennung handelt. Statische Methoden weisen dabei einem Musikstück als Ganzes eine Emotion zu. Bei dynamischen Ansätzen werden die Musikstücke hingegen in Segmente aufgeteilt und jedes dieser Segmente wird mit einer Emotion versehen, wodurch eine zeitliche Änderung der Stimmung beschrieben werden kann (Du, Li & Gao, 2020).

In der vorliegenden Arbeit wird auf Grundlage des Datensatzes (beschrieben in Abschnitt 3.4) eine diskretisierte Version des Circumplexmodells nach Russell (1980) verwendet. Dabei dienen die vier Quadraten des Circumplexmodells als Klassen eines kategorischen Modells, nämlich „traurig“ (sad), „fröhlich“ (delighted), „entspannt“ (relaxed) und „wütend“ (angry), wie in Abbildung 2.5 dargestellt.

2.3 Musikemotionserkennung

Aufgrund der komplexen Wechselwirkung von Musik und Emotionen hat sich mit der Musikemotionserkennung (engl. Music Emotion Recognition, kurz MER) ein eigenes Forschungsfeld entwickelt, welches sich zur Aufgabe gemacht hat, die von einem Musikstück transportierten Stimmungen automatisch zu detektieren. Die MER kann dabei als ein Teilgebiet des Music Information Retrievals (kurz MIR) verortet werden, welches sich generell mit der Extraktion von aussagekräftigen Merkmalen von Musikstücken, entweder direkt aus dem Audiosignal oder aus den Metadaten (wie z. B. Genre) bezogen, beschäftigt (Schedl et al., 2014, S. 128). Mit Blick auf zwei der derzeit größten, online verfügbaren Musikbibliotheken wird die Notwendigkeit solcher Techniken ersichtlich. So bieten Spotify und Apple Music ihren Nutzer*innen (Stand August 2021) laut eige-

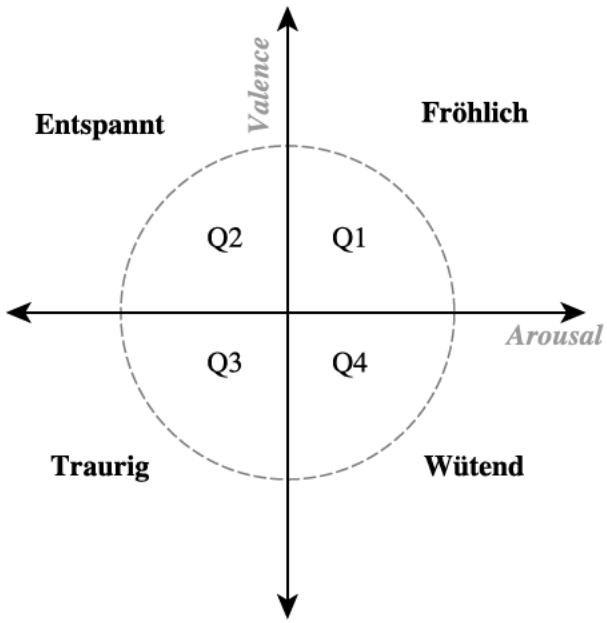


Abbildung 2.5: Das im untersuchten Datensatz verwendete Emotionsmodell.

nen Angaben über 70 Millionen[6] bzw. 75 Millionen[7] Musikstücke an. Von neuen Wegen, solche enormen Datenmengen zu organisieren und verfügbar zu machen, können sowohl die Musikindustrie als auch Nutzer*innen und professionelle Anwender*innen profitieren (Casey et al., 2008). Neben emotionsbasierten Musikempfehlungsalgorithmen (Andjelkovic, Parra & O'Donovan, 2019) oder der Erstellung von stimmungsorientierten Playlisten (Van Zaanen & Kanters, 2010) bietet die automatische MER vielseitige Anwendungsmöglichkeiten, welche ebenfalls in der Werbung, im Fernsehen oder der Gaming-Industrie zum Einsatz kommen können (Panda, 2019).

Für die Umsetzung der automatischen MER haben sich in den vergangenen zwei Jahrzehnten vor allem Techniken aus dem Bereich des maschinellen Lernens (engl. Machine Learning, kurz ML) etabliert. Allgemeines Ziel des maschinellen Lernens ist es, prädiktive Systeme zu entwickeln, welche ansonsten von Menschen durchgeführte Aufgaben durch Vorhersagen automatisieren können (Samek, Montavon, Lapuschkin, Anders & Müller, 2021). ML-Algorithmen sind dabei traditionell von der Güte der verfügbaren Daten abhängig, wodurch sich mit dem sogenannten Feature Engineering eine eigene Ausrichtung innerhalb des ML gebildet hat. Hierbei steht die Entwicklung von Merkmalsdeskriptoren im Fokus (Pouyanfar et al., 2018). So kann in Bezug auf Audiosignale die von Peeters, Giordano, Susini, Misdariis und McAdams (2011) entworfene *Timbre*

[6] `https://newsroom.spotify.com/company-info/`, zuletzt abgerufen am 29.09.2021.
[7] `https://www.apple.com/de/apple-music/`, zuletzt abgerufen am 29.09.2021.

Toolbox als Beispiel herangezogen werden. Hier wurde versucht, die Klangfarbe eines Musikstückes durch verschiedene temporale und spektrale Deskriptoren zu repräsentieren. Solche Vorgehensweisen werden u. a. auch als „klassische“ ML-Ansätze bezeichnet (Delbouys et al., 2018). Allerdings bedarf die Entwicklung aussagekräftiger Deskriptoren bereichsspezifisches Fachwissen, wodurch klassische Ansätze in der Regel zeit- und kostenintensiv sind (N. He & Ferguson, 2020). Aufgrund dessen erfreuen sich in jüngster Zeit sogenannte Deep-Learning-Algorithmen großer Beliebtheit, da sie die Extraktion von Merkmalen und Deskriptoren weitestgehend automatisiert ausführen, womit sich Entwicklungsaufwand und erforderliches Fachwissen minimieren lassen (Pouyanfar et al., 2018). Insbesondere kommen im Deep Learning mehrschichtige, neuronale Netzwerke (engl. Deep Neural Networks, kurz DNN) zum Einsatz, welche als eine Folge von Layern abstrahiert werden können

$$f(x) = f_L \circ \cdots \circ f_1(x),$$

wobei jeder Layer zuerst eine lineare Transformation, gefolgt von einer nichtlinearen Transformation auf die eingespeisten Daten ausübt. Derartige Modelle sind aufgrund ihrer hohen Anzahl an Layern meist gute Prädiktoren und haben sich vor allem im Bereich der Computer Vision als nützlich erwiesen (Samek et al., 2021). Aber auch in anderen Bereichen wie der Computerlinguistik (engl. Natural Language Processing, kurz NLP) haben DL-Methoden beträchtliche Ergebnisse erzielen können (Pouyanfar et al., 2018).

Ebenso konnten DL-Strategien innerhalb der automatischen MER erfolgreich eingesetzt werden, weswegen sich im weiteren Verlauf dieser Arbeit auf Deep Learning beschränkt werden soll. Eine ausführliche Übersicht über aktuelle merkmalsbasierte MER-Ansätze sind in (Panda et al., 2020) und (X. Yang, Dong & Li, 2018) zu finden. Der folgenden Abschnitt 2.4 hingegen soll einen Überblick über MER-Ansätze mit Deep Learning bieten.

2.4 Deep-Learning-Ansätze in der Musikemotionserkennung

Innerhalb der automatischen MER-Forschung wurden überwiegend zwei populäre DL-Netzwerkarchitekturen eingesetzt und untersucht: Convolutional Neural Networks (CNNs) und Recurrent Neural Networks (RNNs). Zusammen mit den recht neuen Transformer-Netzwerken sollen diese drei Architekturen zunächst kurz vorgestellt werden, bevor weiter unten auf aktuelle Forschungsergebnisse

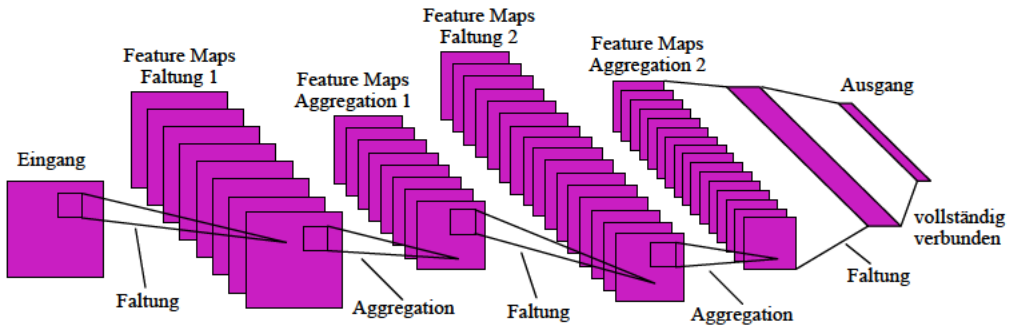


Abbildung 2.6: Beispiel für ein generisches CNN nach LeCun et al. (2010).

näher eingegangen wird.

CNNs sind trainierbare Netzwerke und basieren dabei auf einer internen, hierarchischen Struktur. Die Nützlichkeit einer derartigen Struktur wird durch die Beschaffenheit von digitalen Bildern deutlich. Ein digitales Bild besteht aus Pixeln, welche sich zu Kanten zusammensetzen, welche wiederum Muster und letztendlich Objekte ergeben. Um solche Strukturen erkennen zu können, bestehen CNN-Architekturen unter anderem aus mehreren Filter- und Aggregation[8]-Layern, welche die Eingangsdaten filtern und bündeln. Eine beispielhafte CNN-Architektur findet sich in Abbildung 2.6. Die Eingangsdaten eines CNNs können als ein- oder mehrdimensionale Matrizen vorliegen, wobei in der Computer Vision meistens mit zwei- bzw. dreidimensionalen Daten (d. h. Bilder mit RGB-Farbkanälen) gearbeitet wird (LeCun et al., 2010). CNNs konnten insbesondere bei der Verarbeitung von Bildern in den letzten Jahren beachtliche Ergebnisse erzielen (Shahroudnejad, 2021), welches sich auch auf Problemstellungen im Audiobereich übertragen hat. Mittels der Berechnung von Spektrogrammen[9] können so auch Audiosignale von ein- in zwei- bzw. dreidimensionale Daten umgewandelt werden. Im Audiobereich ist eine derartige visuelle Darstellung von Audiosignalen bei DL-Architekturen, welche auf CNNs aufbauen, durchaus üblich und findet sich so auch in der automatischen MER wieder (u. a. (X. Liu et al., 2017; Malik et al., 2017; Du et al., 2020)).

RNNs hingegen setzen einen sequenziellen Aufbau der gegebenen Daten voraus, wie es beispielsweise bei Texten der Fall ist[10]. So kann in Texten zwischen Wörtern, die Bezug aufeinander nehmen, ein weiteres nicht inhaltsbezogenes Wort stehen, was zu zeitlichen Abhängigkeiten zwischen den Satzbausteinen führt. Aus diesem Grund arbeiten RNNs mit internen versteckten Zuständen

[8] In der englischsprachigen Literatur auch als Pooling-Layer bekannt.
[9] Durch ein Spektrogramm wird der zeitliche Verlauf des Frequenzspektrums eines Audiosignals mithilfe von Farbkodierungen bildlich dargestellt.
[10] Audiosignale bilden im eindimensionalen Fall auch eine sequenzielle Datenstruktur.

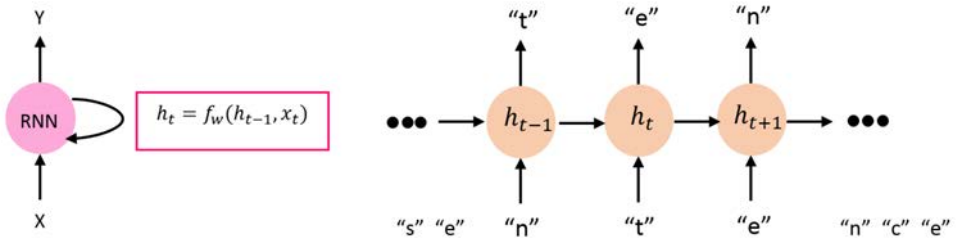


Abbildung 2.7: Beispiel für ein generisches RNN nach Shahroudnejad (2021).

(engl. hidden states h_t), welche mithilfe von Rekursion das dynamische zeitliche Verhalten von sequenziellen Daten modellieren und so langfristige zeitliche Abhängigkeiten einzelner Datenpunkte erkennen sollen. Häufig werden hierbei sogenannte *Long Short-Term Memory*-Architekturen (kurz LSTM) implementiert (J. Wang et al., 2016), welche sowohl in der audio- (G. Liu & Tan, 2020) als auch textbasierten MER (Delbouys et al., 2018; Abdillah et al., 2020) verwendet wurden. Ein Beispiel für ein einfaches, generisches RNN findet sich in Abbildung 2.7.

Mit den sogenannten Transformer-Netzwerken haben sich weitere Varianten in den letzten Jahren etabliert, welche einige Defizite von RNNs wie den LSTMs beheben konnten. Ursprünglich im NLP angesiedelt und für Sprachübersetzungen entwickelt, bringen Transformer interessante Neuerungen mit sich. Im Gegensatz zu RNNs verzichtet das erstmals von Vaswani et al. (2017) beschriebene Transformer-Modell auf Rekursion und führt stattdessen einen Aufmerksamkeitsmechanismus ein, welcher globale Abhängigkeiten zwischen Eingangs- und Ausgangsdaten erfassen kann. Diese Architektur ermöglicht unter anderem die Parallelisierung von Trainingsprozessen, wodurch die Entwicklungszeit derartiger Netzwerke enorm verkürzt werden kann. Weiterhin konnten mit der Transformer-Architektur vielversprechende Transfer-Learning-Modelle wie die von Google entwickelte *BERT*-Architektur erarbeitet werden (Devlin, Chang, Lee & Toutanova, 2018), welche vereinzelt in der textbasierten MER schon eingesetzt wurden (G. Liu & Tan, 2020). Ebenfalls gibt es bereits erste Ansätze, die Transformer-Architektur auf Audiosignale anzuwenden (Gong, Chung & Glass, 2021), welche allerdings noch nicht im Zusammenhang mit der automatischen MER stehen.

Im Folgenden sollen Beispiele aus der Literatur, aufgeteilt nach audio- und textbasierten MER-Ansätzen, vorgestellt werden. Abschließend werden weitere Arbeiten behandelt, welche audio- und textbasierte Informationen für die automatische MER kombinieren.

In einem audiobasierten Ansatz konnten T. Liu et al. (2018) zeigen, dass der Einsatz von CNNs die durchschnittliche Genauigkeit (0,724) der Emotionserkennung im Vergleich zu klassischen Machine-Learning-Methoden wie SVMs[11] (0,385) deutlich steigern kann. Im von Russell (1980) adaptierten Circumplexmodell teilen T. Liu et al. (2018) die zweidimensionale Valenz-Arousal-Ebene (Abszisse: Valenz, Ordinate: Arousal) wie folgt auf: -45 bis 45 Grad „zufrieden"/„fröhlich" (pleased/happy), 45 bis 135 Grad „erregt"/„beunruhigt" (aroused/alarmed), 135 bis 225 Grad „unglücklich"/„traurig" (miserable/sad), 225 bis 315 Grad „beruhigt"/„müde" (soothing/tired). Es wurden über 700 Ausschnitte von Musikstücken aus dem *Free Music Archive*[12] mit je 45 Sekunden Länge analysiert, wobei die Ausschnitte noch einmal in Segmente von fünf Sekunden Länge unterteilt wurden, um den Datensatz zu vergrößern. Für jedes Segment wurde ein Spektrogramm berechnet und das CNN mithilfe dieser visuellen Darstellung trainiert. X. Liu et al. (2017) implementierten ebenfalls CNNs mit Spektrogrammen, allerdings mit einem kategorischen Emotionsmodell bestehend aus 18 verschiedenen Emotionen. Als Grundlage dienten der CAL500[13]-Datensatz mit über 500 Musikstücken und der CAL500exp[14]-Datensatz mit über 3000 Musiksegmenten. Für die MER konnte ein durchschnittliches F1-Maß von 0,709 erreicht werden. Auch die Kombination von mehreren Netzwerkarchitekturen zur automatischen MER wurde erforscht. So verbanden Malik et al. (2017) ein CNN mit einem RNN, wobei das RNN als bidirektionales LSTM-Netzwerk implementiert wurde. Als Emotionsmodell wurde hier ebenfalls das Circumplexmodell von Russell (1980) übernommen. Während für den Trainingsdatensatz über 400 je 30 Sekunden lange Ausschnitte von Musikstücken verwendet wurden, fungierten knapp 60 Musikstücke in ihrer kompletten Länge als Testdatensatz. Unter der Verwendung von grundlegenden Audiodeskriptoren wie unter anderem den Mel Frequency Cepstral Coefficients (MFFCs), Spectral Centroid und Spectral Rolloff konnte der mittlere quadratische Fehler (Root-Mean-Square Error, RMSE) auf 0,267 für Valenz und auf 0,202 für Arousal verringert werden. Du et al. (2020) konnten die Resultate von Malik et al. (2017) noch einmal deutlich verbessern und erreichten ebenfalls mit einer Kombination aus CNN und bidirektionalem LSTM einen RMSE-Wert für die Valenz von 0,06 bzw. 0,07 für Arousal. Allerdings wurde hier zum einen ein deutlich größerer Datensatz verwendet, welcher 1000 westliche Popsongs umfasst. Zum anderen haben Du et al. (2020) sowohl Mel-Spektrogramme als auch Cochleogramme der Musikstücke berechnet. Ein Cochleogramm überführt dabei das Audiosignal in einen multidimensionalen Vektor durch den Einsatz von Gammaton-Filtern und soll

[11] Support Vector Machines.
[12] `https://freemusicarchive.org/` , zuletzt abgerufen am 29.09.2021.
[13] `https://github.com/yzhaobk/CAL500`, zuletzt abgerufen am 29.09.2021.
[14] `http://slam.iis.sinica.edu.tw/demo/CAL500exp/`, zuletzt abgerufen am 29.09.2021.

so die Informationen, die vom Ohr zum Gehirn gesendet werden, repräsentieren.

Parisi et al. (2019) setzen in ihrem textbasierten MER-Ansatz auf verschiedene DNNs. Die Emotionserkennung beschränkte sich dabei auf ein kategorisches Emotionsmodell, welches aus den fünf Basisemotionen Traurigkeit (sadness), Freude (joy), Angst (fear), Wut (anger) und Ekel (disgust) besteht. Das beste F1-Maß (67,1%) wurde von einer Netzwerkarchitektur erzielt, die sich aus dem von Facebook entwickelten Word Embedding fastText[15] und einem LSTM-Netzwerk mit Aufmerksamkeitsmechanismus zusammensetzt. Obwohl hier ein gutes Ergebnis erzielt werden konnte, muss trotzdem erwähnt werden, dass die fünf Basisemotionen innerhalb des Datensatzes sehr ungleich verteilt sind: so sind lediglich 2,1% der Lyrics als „Ekel“ gekennzeichnet, während 43,9% mit der Emotion „traurig“ versehen wurden. Weiterhin ist die Größe des verwendeten Datensatzes nicht bekannt.

Sowohl Abdillah et al. (2020) als auch Agrawal et al. (2021) untersuchten den *MoodyLyrics*-Datensatz[16], welcher aus über 2500 Musikstücken verteilt auf vier Emotionsklassen (sad, happy, relaxed, angry) besteht, die den vier Quadranten des Circumplexmodells nach Russell (1980) entsprechen. Abdillah et al. (2020) setzten dabei auf ein bidirektionales LSTM-Netzwerk mit einem 100-dimensionalen GloVe-Word-Embedding und konnten ein F1-Maß von 91% erzielen. Agrawal et al. (2021) verfolgten hingegen einen Transfer-Learning-Ansatz mit dem zuvor trainierten Transformer-Netzwerk „XLNet“ und konnten damit eine Genauigkeit von 94,78% erreichen. Die Arbeit von Agrawal et al. (2021) stellt damit eine der ersten Untersuchungen von Transfer-Learning-Strategien im Bereich der textbasierten MER dar.

Weiterhin wurden auch MER-Ansätze mit einer Kombination von audio- und textbasierten Informationen betrachtet. Derartige Modelle werden mitunter auch als „Fusion-Modelle“ bezeichnet (z. B. bei (Delbouys et al., 2018; G. Liu & Tan, 2020)). Einen ersten Ansatz lieferten Jeon et al. (2017). Die verwendete Netzwerkarchitektur setzte sich aus einem Convolutional Recurrent Neural Network (CRNN) für die Audiodaten und einem CNN für die Lyrics zusammen, wobei Audiosignale als Mel-Spektrogramme und Songtexte als Wortvektoren repräsentiert wurden. Mit einer Genauigkeit von 80,46% konnte von diesem Modell ein achtbares Ergebnis erreicht werden. Es ist jedoch anzumerken, dass der hier analysierte Datensatz von über 7000 K-Pop-Musikstücken nur ein binäres Emotionsmodell (positiv oder negativ) abdeckt. Bhattacharya und Ka-

[15] `https://fasttext.cc/`, zuletzt abgerufen am 29.09.2021.

[16] `http://softeng.polito.it/erion/MoodyLyrics4Q.zip`, zuletzt abgerufen am 29.09.2021.

dambari (2018) entschieden sich für ein kategorisches Emotionsmodell mit fünf Emotionsgruppen. Als Architektur wurde ebenfalls ein CNN für beide Modalitäten verwendet, wobei Mel-Spektrogramme der Audiosignale berechnet und ein 100-dimensionales Word Embedding[17] angewandt wurden. Für zwei verschieden große Datensätze (der MIREX-Datensatz mit über 900 Einträgen und das *Million Song Dataset*[18] (MSD) mit über 48000 Einträgen) konnte ein F1-Maß für die Emotionserkennung von 66,28% (MIREX) bzw. 69,73% (MSD) erzielt werden. Delbouys et al. (2018) beziehen ebenfalls einen Teil ihres über 18000 Musikstücke umfassenden Datensatzes aus dem MSD. Neben verschiedenen DL-Methoden für Audiosignale und Lyrics werden hier allerdings auch klassische Feature-Engineering-Ansätze untersucht und miteinander verglichen. Auch hier konnte gezeigt werden, dass kombinierte Modelle eine bessere Emotionserkennung liefern als rein text- oder audiobasierte. Obwohl für die Arousaldetektion durch Deep Learning bessere Resultate erzielt werden konnten ($R^2 = 0,235$), waren Feature-Engineering-Ansätze für die Valenzerkennung gleichermaßen performativ wie DL-Methoden ($R^2 = 0,219$). Weiterhin konnte herausgefunden werden, dass sich insbesondere die Valenzerkennung verbessert, wenn die Ausgänge des Lyrics- und des Audionetzwerkes früher zusammengeführt werden. Weiterhin verfolgten G. Liu und Tan (2020) einen MER-Ansatz mit Transfer Learning, welches näher in Abschnitt 2.5 beschrieben wird. Hierbei wurde für die Lyrics mit BERT ein zuvor auf chinesischen Text trainiertes Transformer-Modell herangezogen, während die Audiodaten in Form von Spektrogrammen mit einem LSTM-Modell analysiert wurden. Die über 700 Musikstücke des *Music Mood Classification Dataset* wurden einer von vier Emotionsklassen zugeteilt, wobei eine Genauigkeit von 79,62% erreicht werden konnte.

Eine Übersicht über die zuvor besprochenen Arbeiten findet sich abschließend in den Tabellen 2.2 und 2.3. In dieser Übersicht werden Auskünfte über die verwendeten Emotionsmodelle, Deep-Learning-Ansätze, Datensätze und Resultate in gebündelter Form gegeben.

[17] GloVe-Repräsentation, siehe `https://nlp.stanford.edu/projects/glove/`, zuletzt abgerufen am 29.09.2021.

[18] `http://millionsongdataset.com/`, zuletzt abgerufen am 29.09.2021. Auch als MSD abgekürzt.

Literatur	Eingangs-daten	EMOTIONS-MODELLE Art	# Kl.	# Dim.	AUDIOBASIERTE ANSÄTZE Modell	Vorver-arbeitung	TEXTBASIERTE ANSÄTZE Modell	Vorver-arbeitung
(T. Liu et al., 2018)	Audio	Kat.	4	-	CNN	Spektrogramme	-	-
(X. Liu et al., 2017)	Audio	Kat.	18	-	CNN	Spektrogramme	-	-
(Malik et al., 2017)	Audio	Kont.	-	2	CRNN	U.a. MFCC, Log mel-Energie, Spectral Flux	-	-
(Du et al., 2020)	Audio	Kont.	-	2	CRNN	mel-Spektrogr. Cochleogramme	-	-
(Parisi et al., 2019)	Lyrics	Kat.	5	-	-	-	LSTM +Attention	fastText Emb. (300-dim)
(Abdillah et al., 2020)	Lyrics	Kat.	4	-	-	-	bi-LSTM	GloVe (100-dim)
(Agrawal et al., 2021)	Lyrics	Kat.	4	-	-	-	Transformer	XLNet
(Delbouys et al., 2018)	Komb.	Kont.	-	2	CNN	mel-Spektrogr.	LSTM	word2vec (100-dim)
(Bhattacharya & Kadambari, 2018)	Komb.	Kat.	5	-	CNN	mel-Spektrogr.	CNN	GloVe (100-dim)
(Jeon et al., 2017)	Komb.	Kat.	2	-	CRNN	mel-Spektrogr.	CNN	(27496x400) Wortvektoren
(G. Liu & Tan, 2020)	Komb.	Kat.	4	-	LSTM	U.a. mel-Spektrogr., Spec. Centroid	Transformer	BERT

Tabelle 2.2: Übersicht über untersuchte Deep-Learning-Ansätze in der MER (Teil 1).

	DATENSATZ			AUSWERTUNG		
Literatur	**Name / Art**	**Größe**	**Sprache**	**Problem-stellung**	**Modell-bewertung**	**Resultate**
(T. Liu et al., 2018)	Free Music Archive	744	-	Klassifikation	Genauigkeit	72,4 %
(X. Liu et al., 2017)	CAL500exp, CAL500	3223 502	-	Klass.	F1-Maß (Micro)	70,90 %
(Malik et al., 2017)	Free Music Archive	489	-	Regression	RMSE	0,267 (Valenz) 0,202 (Arousal)
(Du et al., 2020)	Mainstream Music	1000	-	Regr.	RMSE	0,07 (Arousal) 0,06 (Valenz)
(Parisi et al., 2019)	Musixmatch (Genres unbekannt)	N/A	N/A	Klass.	F1-Maß	67,10 %
(Abdillah et al., 2020)	MoodyLyrics	2189	EN	Klass.	F1-Maß	91 %
(Agrawal et al., 2021)	MoodyLyrics MER Data	2595 180	EN	Klass.	F1-Maß	94,77% (MoodyLyrics) 88,60% (MER)
(Delbouys et al., 2018)	MSD	18644	EN	Regr.	R^2	0,219 (Valenz) 0,235 (Arousal)
(Bhattacharya & Kadambari, 2018)	MIREX, MSD	903 48476	EN	Klass.	F1-Maß	66,28% (MIREX), 69,73% (MSD)
(Jeon et al., 2017)	K-Pop Songs	7484	KO/EN	Klass.	Genauigkeit	80,46 %
(G. Liu & Tan, 2020)	Music Mood Classification Data Set	1200	EN	Klass.	Genauigkeit	79,62 %

Tabelle 2.3: Übersicht über untersuchte Deep-Learning-Ansätze in der MER (Teil 2).

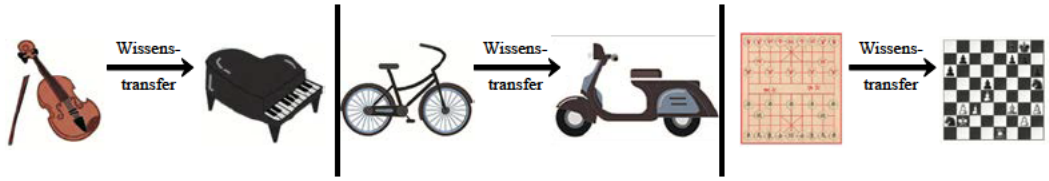


Abbildung 2.8: Intuitive Beispiel für Transfer Learning nach Zhuang et al. (2021).

2.5 Transfer Learning

Inspiriert durch die Transferleistung des Menschen hat sich das Konzept des Transfer Learnings (kurz TL) innerhalb des Machine Learnings etabliert. So ist der Mensch in der Lage, bereits vorhandenes Wissen bei der Erlernung neuer Fähigkeiten anzuwenden. So mag es für eine Person, welche Violine spielt, leichter sein, ein neues Instrument wie Klavier zu erlernen, als für eine Person, die kein Instrument spielt. In diesem Beispiel lassen sich sowohl Violine als auch Klavier der Domäne der Musikinstrumente zuordnen, deren Erlernung Gemeinsamkeiten wie das Lesen von Noten gemeinsam haben kann (Zhuang et al., 2021). In Abbildung 2.8 finden sich weitere intuitive Beispiele.

Die Generalisierung von Erfahrungen spielt bei der Fähigkeit des Menschen, Wissen zu transferieren, eine entscheidende Rolle und wurde auch auf den Bereich des Machine Learning übertragen (Zhuang et al., 2021). Die Idee des Transfer Learnings ist hier insbesondere durch die Tatsache motiviert, dass Datensätze zu spezifischen Bereichen meistens in ihrem Umfang limitiert oder nicht öffentlich verfügbar sind. Auch ist die Erstellung von umfangreichen und soliden Datensätzen in der Regel zeit- und kostenintensiv, wodurch davon in der Praxis häufig abgesehen wird (Weiss, Khoshgoftaar & Wang, 2016). Die Zahl solcher Datensätze nimmt inzwischen jedoch kontinuierlich zu, wodurch Transfer Learning eine attraktive Möglichkeit darstellt, die Datenlage in spezifischen Bereichen zu ergänzen. Grundlegend ist hierbei die Annahme, dass zwischen den verfügbaren Daten, welche einer Quelldomäne (engl. source domain, mit tiefgestelltem S gekennzeichnet) entspringen, und den zu untersuchenden Daten in einer Zieldomäne (engl. target domain, mit tiefgestelltem T gekennzeichnet) gewisse Gemeinsamkeiten bestehen. Neben Gemeinsamkeiten ist es auch vorteilhaft, wenn Quell- und Zieldomäne sich einer übergeordneten Domäne zuweisen lassen. So sind in dem eingangs beschriebenen Beispiel Violine (Quelldomäne) und Klavier (Zieldomäne) beides Musikinstrumente (übergeordnete Domäne), welche sich aber in ihrer Bauform und Spielweise unterscheiden (Zhuang et al., 2021).

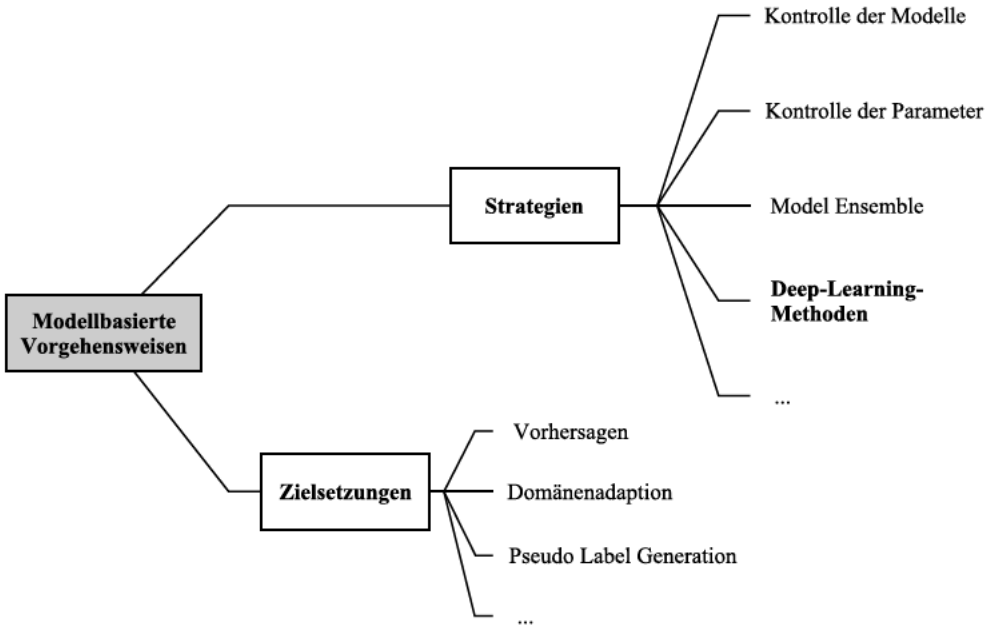


Abbildung 2.9: Einordnung von Deep-Learning-Methoden für modellbasierte Vorgehensweisen mit Transfer Learning nach Zhuang et al. (2021).

Um den Rahmen dieser Arbeit nicht zu sprengen, soll sich im Folgenden auf Transfer-Learning-Strategien beschränkt werden, die im Zusammenhang mit Deep Learning stehen. Solche Strategien, teilweise auch als *Deep Transfer Learning* (Tan et al., 2018) bezeichnet, lassen sich unter sogenannten modellbasierten Vorgehensweisen kategorisieren, wie in Abbildung 2.9 verdeutlicht wird. Einen Überblick über weitere Transfer-Learning-Ansätze bieten u. a. Zhuang et al. (2021); Tan et al. (2018); Weiss et al. (2016); Torrey und Shavlik (2010).

Damit eine Definition von Transfer Learning möglich wird, müssen zunächst die Begriffe der Domäne (engl. domain) und Aufgabe (engl. task) näher beschrieben werden. Eine Domäne $\mathcal{D}$ setzt sich aus einem Merkmalsraum $\mathcal{X}$ und einer Randverteilung $P(X)$ mit $X = \{x_1, \ldots, x_n \in \mathcal{X}\}$ zusammen. Dabei ist x_i der i-te Merkmalsvektor eines Datenpunktes x in einer Beispielmenge X, während n die Anzahl der Merkmalsvektoren in X angibt. Den Raum aller möglichen Merkmalsvektoren beschreibt $\mathcal{X}$. Eine Domäne kann demnach als $\mathcal{D} = \{\mathcal{X}, P(X)\}$ formuliert werden. Für eine gegebene Domäne $\mathcal{D}$ kann weiterhin eine Aufgabe $\mathcal{T}$ bestimmt werden. Eine Aufgabe $\mathcal{T}$ besteht aus einem Labelraum $\mathcal{Y}$ und einer Vorhersagefunktion $f(\cdot)$, womit sich $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ ergibt. Die Vorhersagefunktion $f(\cdot)$ wird hierbei durch Paare von Merkmalsvektoren und Labels x_i, y_i mit $x_i \in X$ und $y_i \in \mathcal{Y}$ erlernt. Die Datenmenge D_S einer Quelldomäne $\mathcal{D}_S$ ergibt sich zu $D_S = \{(x_{S1}, y_{S1}), \ldots, (x_{Sn}, y_{Sn})\}$, wobei $x_{Si} \in \mathcal{X}_S$ den i-ten Datenpunkt von D_S und $y_{Si} \in \mathcal{Y}_S$ das entsprechende Label für x_{Si} beschreiben. Analog lässt sich die Datenmenge D_T einer Zieldomäne $\mathcal{D}_T$ formulieren. Die Quell- und Zie-

laufgaben mit ihren entsprechenden Vorhersagefunktionen werden als $\mathcal{T}_S$ und $f_S(\cdot)$ bzw. $\mathcal{T}_T$ und $f_T(\cdot)$ bezeichnet. Bei gegebener Quell- und Zieldomäne $\mathcal{D}_S$ bzw. $\mathcal{D}_T$ mit Quellaufgaben $\mathcal{T}_S$ bzw. $\mathcal{T}_T$ wird unter Transfer Learning der Prozess verstanden, die Zielvorhersagefunktion $f_T(\cdot)$ mit den Informationen von $\mathcal{D}_S$ und $\mathcal{T}_S$ zu optimieren, wobei $D_S \neq D_T$ und/oder $T_S \neq T_T$ gilt (Weiss et al., 2016). Von Deep Transfer Learning ist die Rede, wenn $f_T(\cdot)$ eine nichtlineare Funktion in Form eines DNNs widerspiegelt (Tan et al., 2018).

Des Weiteren kann zwischen homogenem und heterogenem Transfer Learning unterschieden werden. Für homogenes Transfer Learning gilt $\mathcal{X}_S = \mathcal{X}_T$ und $\mathcal{Y}_S = \mathcal{Y}_T$, wohingegen für heterogenes Transfer Learning $\mathcal{X}_S \neq \mathcal{X}_T$ und/oder $\mathcal{Y}_S \neq \mathcal{Y}_T$ vorausgesetzt wird. Beim heterogenen Transfer Learning werden also Quell- und Zieldomäne in unterschiedlichen Merkmalsräumen repräsentiert (Zhuang et al., 2021). Im Fall dieser Arbeit handelt es sich demnach um einen heterogenen Transfer-Learning-Ansatz, welches näher in den Abschnitten 3.1, 3.2 und 3.4 beschrieben wird.

Wie gut eine Transfer-Learning-Strategie funktioniert, lässt sich nach Torrey und Shavlik (2010) mit drei weit verbreiteten Maßen überprüfen. Zum einen kann zuerst getestet werden, wie ein Transfer-Learning-Modell M_{TL} mit transferiertem Wissen aus der Quelldomäne in der Zieldomäne agiert, ohne auf die Zielaufgabe trainiert worden zu sein. Zum Vergleich wird ein entsprechendes Modell M ohne Transfer Learning herangezogen. Zum anderen kann als weiteres Maß die Zeit genommen werden, die ein Modell M_{TL} benötigt, um die Zielaufgabe zu erlernen, verglichen mit der Zeit eines Models M, welches die Zielaufgabe ohne transferiertes Wissen erlernt. Zuletzt kann die finale Leistungsfähigkeit eines trainierten Model M_{TL} mit der eines trainierten Model M verglichen werden. Diese drei Kriterien sollen auch im weiteren Verlauf dieser Arbeit für die Bewertung der zu untersuchenden Modelle herangezogen werden.

Generell konnte das Konzept des Transfer Learnings bereits in vielen Bereichen wie der Bild- und Textklassifizierung oder der Gesichts- und Gestenerkennung Einzug halten. Auch in der Sprachemotionserkennung oder der Sentimentanalyse von Texten wurden Transfer-Learning-Strategien untersucht (Zhuang et al., 2021; Weiss et al., 2016). Jedoch sind im Bereich der Musikemotionserkennung nur wenige Untersuchungen mit Transfer Learning durchgeführt wurden. Wie bereits im Abschnitt 2.4 beschrieben, wurden modellbasierte Transfer-Learning-Strategien bislang einzig für Lyrics betrachtet (Agrawal et al., 2021; G. Liu & Tan, 2020). Insbesondere sind keine Untersuchungen zu TL-Ansätze für die audiobasierte, automatische MER bekannt. Dementsprechend soll im

Folgenden ein erster Versuch unternommen werden, Transfer Learning im Zusammenhang mit der automatischen MER näher zu erforschen, wobei sowohl text- als auch audiobasierte Modellen untersucht werden sollen. Daneben soll auch die Kombination aus audio- und textbasierten Informationen im Zusammenhang mit Transfer Learning betrachtet werden. Die entsprechenden Modelle werden im nächsten Kapitel 3 im Detail vorgestellt.

Kapitel 3

Methoden

Im vorliegenden Kapitel wird ein Überblick über die verwendeten Methoden dieser Arbeit gegeben. So sollen in den folgenden Abschnitten zunächst die zu untersuchenden Modelle vorgestellt werden. Um eine Aussage darüber treffen zu können, inwiefern Transfer-Learning-Methoden in der automatischen MER angewandt werden können, werden insgesamt sechs verschiedene Modelle untersucht, welche in Tabelle 3.1 aufgelistet sind. Es werden demnach drei Ansätze mit TL-Modellen betrachtet, welche jeweils audio- und textbasierte Informationen, sowie deren Kombination als Eingangsdaten erhalten. Die entsprechenden Netzwerkarchitekturen „AudioNet“, „LyricsNet“ und „FusionNet“ werden in den Abschnitten 3.1, 3.2 und 3.3 näher beschrieben. Zusätzlich werden drei Vergleichsmodelle ohne TL ausgewertet. Diese Vergleichsmodelle sind dabei von ihrer Struktur mit den TL-Modellen identisch und unterscheiden sich von diesen lediglich in der Gestaltung der internen Gewichtungen. Bei den Vergleichsmodellen werden die Gewichtungen des jeweiligen Netzwerkes anfangs zufällig initialisiert, während die TL-Modelle auf zuvor trainierte Gewichtungen zurückgreifen können. Aus diesem Grund werden die Vergleichsmodelle ohne Transfer Learning im weiteren Verlauf mit dem Namenszusatz „RW“ (für engl. „random weights“) gekennzeichnet. TL-Modelle werden analog durch die Ergänzung „TL“ im Namen kenntlich gemacht. Weiterhin werden der untersuchte Datensatz in Abschnitt 3.4 und die Vorverarbeitung der Daten in Abschnitt 3.5 näher beleuchtet. Abschließend werden die verwendeten Trainingsparameter in Abschnitt 3.6 beschrieben und die zugrundeliegenden TL-Domänen in Abschnitt 3.7 definiert.

Fälle	**Modelle**		
	Audio	**Lyrics**	**Fusion**
Mit Transfer Learning	AudioNetTL	LyricsNetTL	FusionNetTL
Ohne Transfer Learning („random weights“)	AudioNetRW	LyricsNetRW	FusionNetRW

Tabelle 3.1: Untersuchte Modelle.

3.1 AudioNet

Wie bereits in Abschnitt 2.4 beschrieben, soll auch in dieser Arbeit eine CNN-Architektur für audiobasierte Informationen verwendet werden. Neben „AlexNet“ (Krizhevsky, Sutskever & Hinton, 2012), „VGGNet“ (Simonyan & Zisserman, 2014) und „ResNet“ (K. He, Zhang, Ren & Sun, 2016) stellt das von Huang et al. (2017) beschriebene „DenseNet“ eine der meist genutzten CNN-Architekturen dar (Shahroudnejad, 2021) und soll aus diesem Grund als Ausgangsmodell für die Verarbeitung der Audiosignale verwendet werden. Ein weiterer Grund für die Verwendung des DenseNet-Modells stellt die Fähigkeit dieser Architektur für Transfer Learning im Audiobereich dar. So konnten Palanisamy et al. (2020) zeigen, dass u. a. die DenseNet-Architektur mithilfe von Transfer Learning in der Genre-Klassifikation von Musikstücken beachtliche Ergebnisse erzielen kann.

Bei der von Huang et al. (2017) entwickelten DenseNet-Architektur handelt es sich dabei um ein vielschichtiges CNN, welches eine direkte Verbindung von einem beliebigen Layer zu jedem nachfolgenden Layer implementiert. Eine derart gestaltete Verkettung von Layern wird auch als *Dense Block* bezeichnet (siehe Abbildung 3.1). Unter der Annahme, das Netzwerk bestehe aus L Layern und bekommt als Eingangsdaten ein Bild $\mathbf{x}_0$ übergeben, führt jeder Layer eine nichtlineare Transformation $H_\ell(\cdot)$ aus, wobei ℓ für den jeweiligen Layer steht. Die Transformation $H_\ell(\cdot)$ kann als eine zusammengesetzte Funktion verstanden werden, welche sich aus Operationen wie Batch Normalization (kurz BN), Aktivierungsfunktionen wie den Rectified Linear Units (kurz ReLU), Pooling-Layern oder Faltungen (engl. convolutions, kurz conv) ergibt. Entsprechend wird der Ausgang des ℓ-ten Layers als $\mathbf{x}_\ell$ bezeichnet. Der Ausgang $\mathbf{x}_\ell$ wird vor allem bei CNNs in der Literatur auch *Feature Map* genannt. Das DenseNet-Modell zeichnet sich durch die eingangs beschriebene direkte Verbindung von Layern aus, welche für einen Layer ℓ durch die Konkatenation von Ausgänge aller vorherigen Layer erreicht wird:

$$\mathbf{x}_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]).$$

Innerhalb der DenseNet-Architektur wird die Transformation $H_\ell(\cdot)$ aus drei aufeinanderfolgenden Operationen zusammengesetzt: Batch Normalization, ReLU und einer (3×3)-Faltung. Pooling-Layer, welche die Größe der Feature Maps reduzieren, werden beim DenseNet innerhalb eines Übergangslayer (engl. transition layer) implementiert. Diese Übergangslayer setzen sich aus einer Batch Normalization Operation, einem (1×1)-Faltungslayer und einem (2×2) Average-Pooling-Layer zusammen. Weiter definieren Huang et al. (2017) eine Wachs-

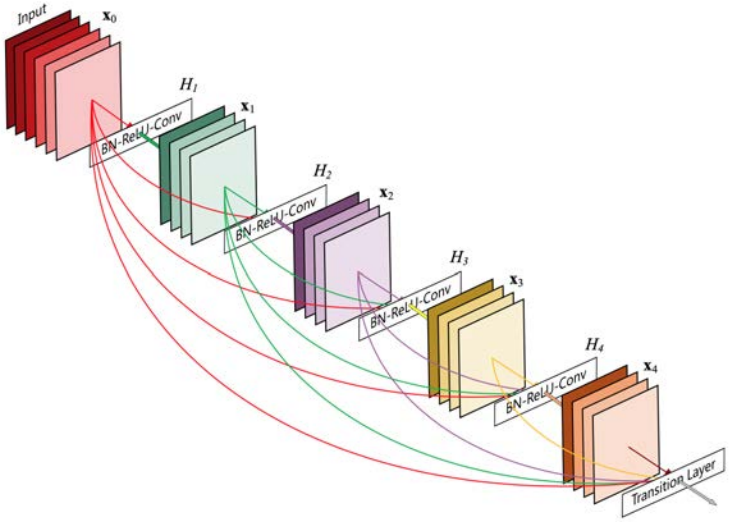


Abbildung 3.1: Dense Block mit fünf Layern und einer Wachstumsrate von $k = 4$ nach Huang et al. (2017).

tumsrate (growth rate) k, welche die Anzahl der von $H_\ell(\cdot)$ erzeugten Feature Maps beschreibt (siehe auch Abbildung 3.1). Um die Anzahl der Feature Maps an den Eingängen eines Layers zu reduzieren und dadurch die rechnerische Effizienz zu steigern, wurde ein Engpass-Layer (bottleneck) mit einer (1×1)-Faltung vor jeder (3×3)-Faltung implementiert. Außerdem konnte die Kompaktheit von DenseNet durch Einführung eines Kompressionsfaktors θ innerhalb der Transition Layer weiter gesteigert werden. Die Anzahl der Feature Maps m, welche einen Dense Block beinhaltet, werden im darauffolgenden Transition Layer um den Kompressionsfaktor θ mit $0 < \theta \leq 1$ reduziert. In dieser Arbeit wurde ein auf den ILSVRC2012-Datensatz[19] trainiertes DenseNet-Modell verwendet. Der ILSVRC2012-Datensatz bietet dabei 1,2 Millionen Bilder mit RGB-Farbkanälen aus 1000 Klassen für das Training an, welche ein Format von (224×224) Pixeln aufweisen.

Huang et al. (2017) entwickelten verschiedene DenseNet-Variationen, welche alle auf den ILSVRC2012-Datensatz trainiert wurden. In Anlehnung an Palanisamy et al. (2020) wird im weiteren Verlauf dieser Arbeit die DenseNet201-Architektur verwendet. Neben einer Wachstumsrate von $k = 32$ weist die DenseNet201-Architektur einen Kompressionsfaktor von $\theta = 0,5$ und Bottleneck-Layer auf.

Darüber hinaus wird eine AudioNet-Architektur mit DenseNet201 als zentralem Element definiert. Die DenseNet201-Parameter werden über das Python-

[19] ILSVRC: ImageNet Large Scale Visual Recognition Challenge 2012. Link: `https://www.image-net.org/challenges/LSVRC/2012/index.php`, zuletzt abgerufen am 29.09.2021.

Modul *PyTorch*[20] bezogen. Allerdings wird der Klassifikationslayer abgewandelt und von 1000 Klassen auf vier Emotionsklassen reduziert. Ebenso werden alle Layer mit Ausnahme des Klassifikationslayers eingefroren, was bedeutet, dass beim Trainieren des AudioNet-Modells die Parameter dieser Layer nicht verändert werden können. Aufgrund des relativ kleinen Datensatzes, welcher in Abschnitt 3.4 näher beschrieben wird, und der großen Anzahl an Parametern des DenseNet201 (20 Millionen Parameter laut Huang et al. (2017)) wird das Einfrieren der Layer empfohlen, um eine eventuelle Überanpassung des finalen Modells zu vermeiden (Yosinski, Clune, Bengio & Lipson, 2014). Eine Darstellung der AudioNet-Architektur findet sich in Abbildung 3.2.

[20] `https://pytorch.org/vision/0.8/_modules/torchvision/models/densenet.html` , zuletzt abgerufen am 29.09.2021.

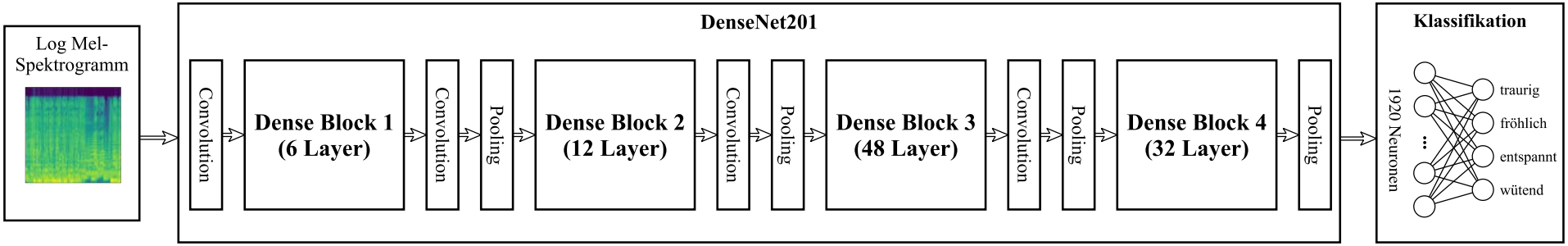


Abbildung 3.2: AudioNet-Architektur in Anlehnung an Huang et al. (2017).

3.2 LyricsNet

Inspiriert von Agrawal et al. (2021) soll für die Verarbeitung von textbasierten Informationen in Form von Lyrics das Transformer-Netzwerk *XLNet*, welches erstmals von (Z. Yang et al., 2019) vorgestellt wurde, verwendet werden. Genau wie beim BERT-Modell (Devlin et al., 2018) wurde beim XLNet die grundlegende Idee verfolgt, ein Modell zu entwickeln, welches zunächst auf einen großen Textkorpus ohne Labels trainiert wurde und anschließend für spezialisierte Aufgaben abgestimmt werden kann. Für den Trainingsprozess wurde bei der XLNet-Architektur ein bidirektionaler Ansatz und eine autoregressive Sprachmodellierung verwendet. Generell wird angestrebt, mithilfe von autoregressiven Sprachmodellen die Wahrscheinlichkeitsverteilung innerhalb eines Textkorpus zu schätzen. Die Wahrscheinlichkeit $p(\mathbf{x})$ einer Textsequenz $\mathbf{x} = (x_1, \ldots, x_T)$ mit Länge T wird so durch Faktorisierung entweder als Vorwärtsprodukt

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{<t})$$

oder als Rückwärtsprodukt

$$p(\mathbf{x}) = \prod_{t=T}^{1} p(x_t|\mathbf{x}_{>t})$$

angegeben. Beide Produkte beschreiben einen unidirektionalen Kontext, d. h. nur die Token[21] vor oder nach dem betrachteten Token x_t werden für die Berechnung der jeweiligen Wahrscheinlichkeit, dass x_t in der Sequenz $\mathbf{x}$ enthalten ist, berücksichtigt. Allerdings kann für die Erfassung des Inhalts eines Textkorpusses ein derartiger unidirektionaler Kontext unzureichend sein. Aus diesem Grund wird ein bidirektionaler Kontext im XLNet implementiert, welcher durch die Einführung von Permutationen erreicht wird. Für eine Sequenz x der Länge T gibt es $T!$ Permutationen, die für eine autoregressive Faktorisierung genutzt werden können. $\mathcal{Z}_T$ beschreibt die Menge aller möglichen Permutationen einer Sequenz mit den Indizes $[1, 2, ..., T]$. Weiterhin kennzeichnen z_t und $\mathbf{z}_{<t}$ das t-te Element und die ersten $t-1$ Elemente einer Permutation $\mathbf{z} \in \mathcal{Z}_T$. Mit der Maximum-Likelihood-Methode ergibt sich als Zielsetzung des XLNet-Sprachmodells folgende Gleichung (3.1):

$$\max_{\theta} = \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[\sum_{t=1}^{T} \log p_\theta(x_{z_t}|\mathbf{x}_{\mathbf{z}_{<t}}) \right]. \tag{3.1}$$

[21] Ein Token beschreibt eine Folge von Buchstaben oder Ziffern, die durch ein Leerzeichen oder Interpunktion getrennt werden.

Es wird also für eine Sequenz $\mathbf{x}$ jeweils eine Permutation $\mathbf{z}$ betrachtet und die entsprechende Likelihood $p_\theta(\mathbf{x})$ berechnet. Aufgrund der Tatsache, dass der Parameter θ während des Trainingsprozesses für alle Permutationen verwendet wird, ergibt sich als Erwartungswert $\mathbb{E}$, dass x_t jedes $x_i \neq x_t$ innerhalb der Sequenz einmal gesehen hat, wodurch ein bidirektionaler Kontext sichergestellt wird.

Um die Zielsetzung aus Gleichung (3.1) umzusetzen, wurde für das XLNet eine Architektur mit Aufmerksamkeitsmasken (engl. Attention Masks) und zweigleisigem Aufmerksamkeitsmechanismus (engl. two-stream self attention) implementiert. Da eine ausführliche Erklärung dieser beiden Bestandteile den Rahmen der vorliegenden Arbeit sprengen würde, soll für eine detaillierte Darstellung an dieser Stelle auf die Arbeit von Z. Yang et al. (2019) verwiesen werden. Jedoch lassen sich verkürzt zumindest die sogenannten Attention Masks als eine Methode beschreiben, welche die Reihenfolge der Token innerhalb einer Textsequenz berücksichtigen. Das über die *Huggingface*[22] Python-Bibliothek verfügbare XLNet-Modell wurde auf einen Textkorpus von über 30 Milliarden englischsprachigen Token trainiert, welche unter anderem aus englischsprachigen Wikipedia-Artikeln bezogen wurden. Das Modell weist damit einen großen Wissensschatz sowie ein tiefgehendes Verständnis von kontextuellen Informationen auf, wodurch es sehr attraktiv für TL-Anwendungen ist. Aus diesem Grund wird eine LyricsNet-Architektur mit XLNet als zentralem Bestandteil definiert. Es wird das „xlnet-base-cased"-Modell[23] mit zwölf Layern, einem Word Embedding mit 768-dimensionalen Wortvektoren und 110 Millionen Parametern verwendet. Ähnlich zum AudioNet werden alle Layer und deren Parameter bis auf den Klassifikationslayer des XLNet-Modells eingefroren. Ebenso wird die Anzahl der Klassen im Klassifikationslayer auf vier gesetzt. Die LyricsNet-Architektur ist in Abbildung 3.3 dargestellt.

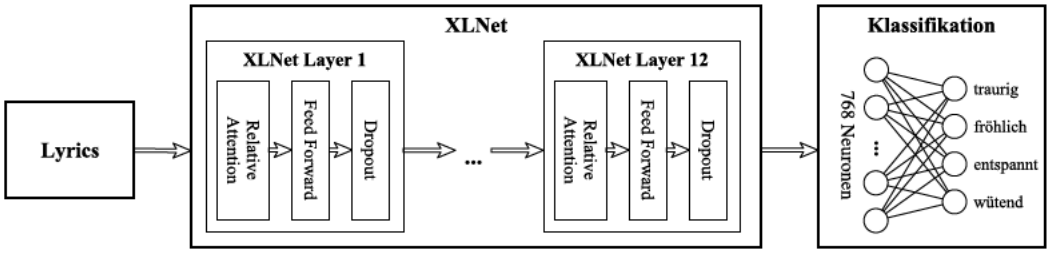


Abbildung 3.3: LyricsNet-Architektur in Anlehnung an Z. Yang et al. (2019).

[22] `https://huggingface.co/transformers/`, zuletzt abgerufen am 29.09.2021.
[23] `https://huggingface.co/transformers/pretrained_models.html`, zuletzt abgerufen am 29.09.2021.

3.3 FusionNet

Um die Kombination aus audio- und textbasierten Informationen für die automatische MER zu untersuchen, soll in diesem Abschnitt eine FusionNet-Architektur definiert werden, welche die Architekturen aus Abschnitt 3.1 und 3.2 miteinander verbindet. Die Fusion von AudioNet und LyricsNet ist dabei an die *MidFusion*-Strategie von Delbouys et al. (2018) angelehnt. Dabei werden die Ausgänge von AudioNet und LyricsNet zunächst durch Konkatenation miteinander kombiniert und anschließend in ein darauffolgendes Klassifikationsnetzwerk geführt. Das Klassifikationsnetzwerk besteht aus zwei Layern mit vollständig miteinander verbundenen Neuronen (engl. fully connected). Dem ersten Klassifikationslayer (FC1) mit 128 Eingangs- und Ausgangsneuronen folgt eine ReLU-Aktivierungsfunktion und ein Dropout-Layer mit $p = 0,1$. Das zweite Klassifikationslayer (FC2) weist ebenfalls 128 Eingangsneuronen sowie als Ausgang die vier Emotionsklassen auf. Damit der konkatenierte Ausgang von AudioNet und LyricsNet und der Eingang der Klassifikationsnetzwerkes übereinstimmen, wurden die Anzahl der Ausgänge von beiden Architekturen von vier auf 64 erhöht. Die Wahl der Anzahl der Ausgänge von AudioNet und LyricsNet ist an dieser Stelle an die Größe des mittleren Layers des von Delbouys et al. (2018) beschriebenen Audio-Netzwerkes angelehnt und könnte für weitere zukünftige Untersuchungen variiert werden. Im Fall der vorliegenden Arbeit soll die Anzahl der Ausgänge zunächst jedoch nicht weiter verändert werden. Die Struktur der beschriebenen FusionNet-Architektur findet sich in Abbildung 3.4.

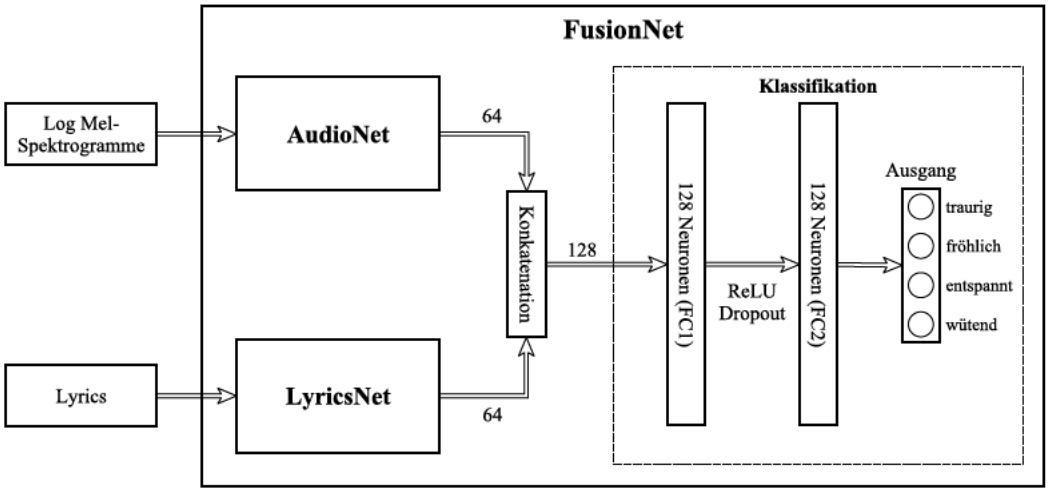


Abbildung 3.4: FusionNet-Architektur.

3.4 Datensatz

Der in dieser Arbeit verwendete Datensatz wurde freundlicherweise vom Technologieunternehmen Cyanite[24] zur Verfügung gestellt. Der Datensatz umfasst dabei insgesamt 5328 westliche kontemporäre Musikstücke mit ausschließlich englischsprachigen Songtexten, welche gleichmäßig auf vier Emotionsklassen verteilt sind. Entsprechend finden sich also 1332 Musikstücke pro Klasse. Die Emotionsklassen entsprechen dabei den vier Quadranten („traurig“, „fröhlich“, „entspannt“, „wütend“) des Circumplexmodells nach Russell (1980), welches in Abschnitt 2.2 und in Abbildung 2.5 vorgestellt wurde. Die Audiosignale der Musikstücke weisen eine Länge von 30 Sekunden auf und liegen mit einer Abtastrate von 44,1 kHz im verlustbehafteten MP3-Format vor. Die Lyrics sind zunächst nur in reiner Textform vorhanden. Einzelne Songtexte sollen im weiteren Verlauf auch als Textsequenzen oder weiter verkürzt als Sequenzen bezeichnet werden.

3.5 Vorverarbeitung der Daten

Aufgrund der in Abschnitt 3.1 und 3.2 beschriebenen Netzwerkarchitekturen für das AudioNet bzw. LyricsNet müssen die vorhandenen Daten (Audiosignale und Lyrics) entsprechend vorverarbeitet werden, damit sie als Eingangsdaten für das jeweilige Netzwerk genutzt werden können. Die genauen Prozesse der Vorverarbeitung sollen in den folgenden Abschnitten 3.5.1 und 3.5.2 erläutert werden.

3.5.1 Audiodaten

Das gewählte DenseNet-Modell erwartet, wie alle ImageNet-Modelle, als Eingangsdaten dreikanalige RGB-Bilder mit dem Format $(3, H, B)$, wobei Höhe H und Breite B der Bilder jeweils mindestens 224 betragen müssen. Jeder einzelne Datenpunkt (Pixel) eines Bildes muss zunächst im Wertebereich $[0, 1]$ liegen und anschließend pro Kanal mit den vorgegebenen Mittelwerten μ und Standardabweichungen σ von

$$\begin{aligned} \mu &= [0.485, 0.456, 0.406], \\ \sigma &= [0.229, 0.224, 0.225] \end{aligned}$$

erneut normalisiert werden[25]. Da die Berechnung eines Spektrogramms normalerweise nur ein einkanaliges Bild mit Grauwerten produziert, wurde sich deswegen am von Palanisamy et al. (2020) beschriebenen Ansatz zur Erzeugung von

[24] `https://cyanite.ai/`, zuletzt abgerufen am 29.09.2021.

[25] Die Mittelwerte und Standardabweichungen finden sich unter `https://pytorch.org/vision/stable/models.html`, zuletzt abgerufen am 29.09.2021.

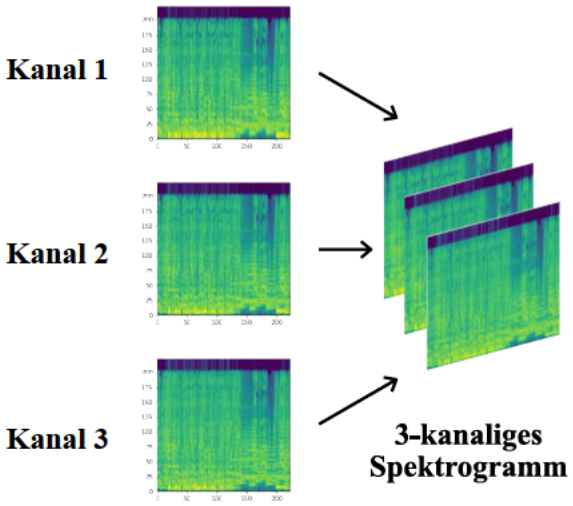


Abbildung 3.5: Beispiel für ein dreikanaliges Spektrogramm.

mehrkanaligen Spektrogrammen orientiert. Um die Spektrogramme in dreikanalige Spektrogramme zu überführen, wurde für jeden Kanal ein eigenes Spektrogramm mit verschiedenen Fenster- und Hop-Längen L_w bzw. L_h berechnet mit $L_w \in \{25, 50, 100\}$ ms und $L_h \in \{10, 25, 50\}$ ms. Verschiedene Fenster- und Hop-Längen pro Kanal sollen gewährleisten, dass pro Kanal unterschiedliche spektrale und temporale Informationen vorhanden sind. Da in Palanisamy et al. (2020) logarithmierte Mel-Spektrogramme mit 128 Mel-Filtern für jeden Kanal die besten Ergebnisse lieferten, wurde diese als Repräsentation der audiobasierten Informationen auch in dieser Arbeit übernommen. Die logarithmierten Mel-Spektrogramme wurden mit den Python-Bibliotheken *librosa*[26] und *torchaudio*[27] für die komplette Länge der Audiosignale von 30 Sekunden berechnet und anschließend mit der Reshape-Funktion der *torchvision*-Bibliothek[28] in ein DenseNet-konformes Format von $(3 \times 224 \times 224)$ gebracht. Eine beispielhafte Darstellung eines solchen dreikanaligen Spektrogramms findet sich in Abbildung 3.5.

3.5.2 Lyrics

Da die Lyrics im gegebenen Datensatz in reiner Textform vorlagen, mussten mehrere Vorverarbeitungsschritte vorgenommen werden. Ein Beispiel für die angewandten Schritte findet sich in Tabelle 3.2. Einige Songtexte enthielten Strukturangaben wie „Chorus“ oder „Verse“, oft in eckige Klammern gesetzt, welche aus den jeweiligen Lyrics entfernt werden mussten, da sie für den Inhalt oder Kontext nicht relevant sind. Weiter wurde eine Rechtschreibprüfung

[26] `https://librosa.org/doc/latest/index.html`, zuletzt abgerufen am 29.09.2021.
[27] `https://pytorch.org/audio/stable/index.html`, zuletzt abgerufen am 29.09.2021.
[28] `https://pytorch.org/vision/stable/index.html`, zuletzt abgerufen am 29.09.2021.

aller Wörter vorgenommen und anschließend Satzzeichen und Stoppwörter[29] entfernt. Zusätzlich wurden alle Groß- in Kleinbuchstaben umgewandelt. Diese Schritte waren notwendig, damit die einzelnen Token der Lyrics auf ihre grammatikalischen Lemmas reduziert werden können. Ein Lemma beschreibt dabei die normalisierte Form eines Wortes. So werden beispielsweise die englischen Wörter „computes“, „computing“ und „computed“ auf die Grundform des Verbs „compute“ normalisiert (Plisson, Lavrac, Mladenic et al., 2004). Da ein Algorithmus ohne die Lemmatisierung die drei Beispielwörter zunächst als eigenständige Wörter identifizieren würde, werden durch die Lemmatisierung Wörter mit gleicher Bedeutung sofort ersichtlich. Die Lemmatisierung wurde mit dem WordNet-Lemmatizer[30] aus der Python-Bibliothek *nltk*[31] durchgeführt. Nach der Lemmatisierung der Lyrics wurden die Textsequenzen für das ausgewählte XLNet-Modell entsprechend vorbereitet. XLNet bietet dabei einen eigenen Tokenizer an, welcher jedes Wort in eine einzigartige ID umwandelt. Die verwendeten IDs stammen aus dem 32 000 Wörter umfassenden Wortschatz[32] von XLNet. Im Gegensatz zum BERT-Modell (Devlin et al., 2018) gibt es beim XLNet-Modell keine maximale Länge der Eingangssequenzen (Z. Yang et al., 2019), d. h. die Songtexte können aus beliebig vielen Wörtern bestehen. Aufgrund des erhöhten Rechenaufwands und um eine Überlastung des GPU-Arbeitsspeichers zu vermeiden, wurde die maximale Länge der Textsequenzen jedoch auf 160 Token pro Songtext begrenzt. Dieser Wert resultiert aus der durchschnittlichen Anzahl an Token pro Songtext und ergibt sich im Mittel zu 134,8 Token/Songtext (ohne Stoppwörter) über den gesamten Datensatz. In Tabelle 3.3 finden sich die durchschnittlichen Tokenanzahlen sowie die minimale und maximale Anzahl an Token pro Songtext für jede der vier Emotionsklassen. Sollte eine Textsequenz kürzer als 160 Token lang sein, wird sie mit Nullen bis auf eine Länge von 160 Token aufgefüllt (engl. zero padding), andernfalls wird die Sequenz nach 160 Token abgeschnitten (engl. truncating). Aufgrund des Zero Paddings sind zusätzliche Attention Masks[33] für jede Textsequenz notwendig, welche sich aber von den internen Attention Masks, beschrieben in Abschnitt 3.2, unterscheiden. Diese „externen“ Attention Masks geben dem XLNet-Modell durch Maskierung zu erkennen, an welcher Stelle innerhalb der Sequenz auch tatsächlich ein Token vorhanden ist. An diesen Stellen werden die Attention Masks auf den Wert 1

[29] Stoppwörter beschreiben Wörter, welche sehr häufig in Textdokumenten vorkommen, aber nur wenig zum Inhalt der Dokumente beitragen. Ein klassisches Beispiel sind Artikel wie „der“, „die“ und „das“ auf Deutsch bzw. „the“ auf Englisch.

[30] `https://wordnet.princeton.edu/`, zuletzt abgerufen am 29.09.2021. WordNet ist eine umfangreiche, englischsprachige Wörterdatenbank der Princeton University.

[31] `http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer#module-nltk.stem.wordnet`, zuletzt abgerufen am 29.09.2021.

[32] `https://huggingface.co/transformers/model_doc/xlnet.html`, zuletzt abgerufen am 29.09.2021.

[33] `https://huggingface.co/transformers/glossary.html`, zuletzt abgerufen am 29.09.2021.

gesetzt, während sie an Stellen, an denen Zero Padding vorgenommen wurde, auf den Wert 0 gesetzt werden. Ein Beispiel für die oben beschriebenen Verarbeitungsschritte der Songtexte findet sich in Tabelle 3.2.

Verarbeitungsschritte	**Lyrics**
Original	[Verse 1] Midnigt, you could see me dangling Glow like a cherri falling, now it's a downpour You could see me racing the rain to the ground
Entfernung von Satzzeichen, Strukturangaben, Großbuchstaben, Rechtschreibfehlern	midnight you could see me dangling glow like a cherry falling now it is a downpour you could see me racing the rain to the ground
Entfernung von Stoppwörtern, Lemmatisierung	midnight could see dangle glow like cherry fall downpour could see race rain ground
Tokenisierung mit XLNet-Tokenizer	[8453, 121, 197, 17, ... 2655, 881, 1331, 0, 0]
Attention Mask	[1.0, 1.0, 1.0, 1.0, ... 1.0, 1.0, 1.0, 0.0, 0.0]

Tabelle 3.2: Beispiel für die Verarbeitungsschritte der Lyrics. Songtext entnommen aus *Julien Baker - Stay Down*.

	Anzahl Token		
Emotionsklasse	**Mininum**	**Maximum**	**Mittelwert**
Traurig	10	486	115,6
Fröhlich	9	590	149,3
Entspannt	11	509	123,8
Wütend	9	819	150,6

Tabelle 3.3: Anzahl der Token pro Textsequenz und Emotionsklasse.

3.6 Trainingsparameter

Im Folgenden sollen die Parameter beschrieben werden, die für das Training und Testen der in Tabelle 3.1 gelisteten Modelle verwendet wurden. Für den Trainingsprozess von Deep-Learning-Modellen mit Transfer Learning wird im Allgemeinen eine geringe Anzahl an Epochen empfohlen. Devlin et al. (2018) schlagen für das von ihnen entwickelte BERT-Modell zwei bis vier Epochen vor um eine Überanpassung des Modells zu vermeiden. Damit ein Eindruck gewonnen werden kann, wie sich die Anzahl der Epochen auf die Performanz der Modelle auswirkt, werden alle Modelle für $i \in \{2, 4, 6, 8\}$ Epochen trainiert. Wenn

im weiteren Verlauf der Arbeit von Modellen gesprochen wird, welche mit einer bestimmten Anzahl an Epochen trainiert wurden, soll dies durch die Anzahl der Epochen im Namen des jeweiligen Modells gekennzeichnet werden (z. B. AudioNetTL4). Die Batch Size wurde auf vier gesetzt, um einer potentiellen Auslastung des Arbeitsspeichers der verwendeten GPUs vorzubeugen. Weiter wurde als Verlustfunktion die Kreuzentropie (engl. cross-entropy loss) implementiert. Für den Optimierungsprozess wurde der AdamW-Optimierer (Loshchilov & Hutter, 2019) mit *Weight Decay Fix* verwendet. Z. Yang et al. (2019) schlagen für TL-Anwendungen eine eher kleine Learning Rate LR zwischen $1 \cdot 10^{-5}$ und $3 \cdot 10^{-5}$ vor, weswegen die LR für alle Versuche in dieser Arbeit auf den Wert $LR = 2 \cdot 10^{-5}$ gesetzt wird. Weiterhin empfehlen Z. Yang et al. (2019) einen linearen Abfall der LR, weshalb auf ein linearen LR-Scheduler ohne sogenanntes *Warm-Up*[34] zurückgegriffen wird. Während 80% der verfügbaren Daten für das Training genutzt wurden, waren die übrigen 20% zum Testen der Modelle vorgesehen. Zusätzlich wurde eine fünffache Kreuzvalidierung durchgeführt, um sicherzustellen, dass jeder Datenpunkt im Datensatz mindestens einmal zum Testen der Modelle genutzt wird. Die Modelle wurden auf GPUs von Nvidia (Tesla T4, P100), welche von Google Colab[35] zur Verfügung gestellten wurden, trainiert und getestet. In Tabelle 3.4 findet sich eine Auflistung der verwendeten Trainingsparameter.

Parameter	**Wert**
Epochen	2, 4, 6, 8
Batch Size	4
Learning Rate	$2 \cdot 10^{-5}$
LR-Scheduler	Linear
Optimizer	AdamW
Verlustfunktion	Kreuzentropie
GPUs	Nvidia Tesla [T4, P100]

Tabelle 3.4: Verwendete Trainingsparameter.

[34] Wenn der Warm-Up-Parameter auf Null gesetzt wird, startet das Training mit der gesetzen $LR = 2 \cdot 10^{-5}$, welche anschließend linear reduziert wird. Siehe auch: `https://huggingface.co/transformers/main_classes/optimizer_schedules.html`, zuletzt abgerufen am 29.09.2021.

[35] `https://colab.research.google.com/`, zuletzt abgerufen am 29.09.2021.

3.7 Definition der Domänen

Abschließend müssen für die oben beschriebenen Modelle Quell- und Zieldomänen erschlossen werden. Dafür werden insgesamt für audio- und für textbasierte Informationen zwei Quell- und zwei Zieldomänen definiert. Für audiobasierte Informationen wird die Quelldomäne als $\mathcal{D}_{S,A}$ bezeichnet und beschreibt die im ImageNet-Datensatz vorhandenen Bilder, mit denen das DenseNet201-Modell trainiert wurde. Die Zieldomäne wird analog als $\mathcal{D}_{T,A}$ bezeichnet und umfasst die visuelle Darstellung der Audiosignale in Form der berechneten logarithmierten Mel-Spektrogramme, beschrieben in Abschnitt 3.5.1. Weiterhin charakterisiert die Quelldomäne $\mathcal{D}_{S,L}$ für textbasierte Informationen den Datensatz, mit denen das XLNet-Modell trainiert wurde. Dieser Datensatz, wie in Abschnitt 3.2 erläutert, setzt sich aus mehreren großen Textkorpora wie u. a. englischsprachigen Wikipedia-Artikeln zusammen. Als Zieldomäne $\mathcal{D}_{T,L}$ werden die zu den Musikstücken gehörigen Lyrics definiert.

Kapitel 4

Auswertung

Im vorliegenden Kapitel soll die Forschungsfrage, welche in Abschnitt 1.2 formuliert wurde, anhand der zu untersuchenden Modelle erörtert werden. Dazu werden zunächst die Ergebnisse der Untersuchungen in Abschnitt 4.1 präsentiert und anschließend in Abschnitt 4.2 unter Zuhilfenahme von verschiedenen Bewertungskriterien ausgewertet und diskutiert.

4.1 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der untersuchten Modelle aus Tabelle 3.1 vorgestellt. Da es sich generell um ein Klassifizierungsproblem mit gleichverteilten Klassen handelt, wurde als Maß für die Performanz der Modelle die Genauigkeit (*Accuracy*, kurz *Acc*, siehe Gleichung (4.1)) verwendet. Die Genauigkeit *Acc* gibt dabei das Verhältnis von allen korrekten Klassifizierungen (Summe aus richtig positiven r_p und richtig negativen r_n Fällen) und der Anzahl aller Klassifizierungen an (Summe aus richtig positiven r_p, falsch positiven f_p, richtig negativen r_n und falsch negativen f_n Fällen):

$$Acc = \frac{\text{Anzahl korrekter Klassifizierungen}}{\text{Anzahl aller Klassifizierungen}} = \frac{r_p + r_n}{r_p + r_n + f_p + f_n} \tag{4.1}$$

Weiterhin wurden in Anlehnung an Torrey und Shavlik (2010) drei Bewertungskriterien herangezogen:

- **Kriterium 1:** Alle Modelle werden ohne Training in der Zieldomäne $\mathcal{D}_T \in \{\mathcal{D}_{T,A}, \mathcal{D}_{T,L}\}$ ausgewertet. In diesem Fall wird als Maß für die Performanz die initiale Genauigkeit Acc_{init} definiert.

- **Kriterium 2:** Alle Modelle werden für $i \in \{2, 4, 6, 8\}$ Epochen trainiert und in der Zieldomäne $\mathcal{D}_T \in [\mathcal{D}_{T,A}, \mathcal{D}_{T,L}]$ ausgewertet. Für diese Fälle wird als Maß für die Performanz die über alle Testdurchläufe $k = 5$ gemittelte Genauigkeit $Acc_i = \frac{1}{k}\sum_{k=1}^{5} Acc_{i,k}$ definiert.

- **Kriterium 3:** Für alle Modelle, die für $i \in \{2, 4, 6, 8\}$ Epochen trainiert wurden, wird die Trainingsdauer t in Minuten gemessen. Für die Vergleichbarkeit wird relatives Verhältnis Q_t definiert.

Leider konnte für das dritte Bewertungskriterium kein Einfluss auf die verwendete GPU innerhalb von Google Colab[36] genommen werden, wodurch über alle Versuche verteilt mehrere GPUs zum Einsatz gekommen sind. Es wurde allerdings sichergestellt, dass Modelle immer in Paaren wie z. B. [AudioNetTL2, AudioNetRW2] mit derselben GPU trainiert und ausgewertet wurden. Um die Modelle bezüglich ihrer Trainingsdauer auch außerhalb von diesen Paarungen miteinander vergleichen zu können, wird ein relatives Verhältnis der Trainingsdauern Q_t definiert:

$$Q_t = \frac{t_m}{t_{m_{TL}}}.$$

Dabei beschreibt t_m die Trainingsdauer des jeweilig betrachteten Modells m. $t_{m_{TL}}$ dient als Referenzwert und beschreibt die Trainingsdauer des Modells mit Transfer Learning (TL-Modell). So ergibt sich für die TL-Modelle innerhalb der Paare immer $Q_t = 1$. Für die Modelle ohne Transfer Learning (RW-Modelle) kann Q_t drei verschiedene Fälle annehmen:

- $Q_t < 1$: Das RW-Modell weist eine kürzere Trainingsdauer als das entsprechende TL-Modell auf.
- $Q_t = 1$: RW- und TL-Modell weisen die gleiche Trainingsdauer auf.
- $Q_t > 1$: Das RW-Modell weist eine längere Trainingsdauer als das entsprechende TL-Modell auf.

Die Ergebnisse aller Modelle für das erste Bewertungskriterium sind in Tabelle 4.1 auf S. 41 zu finden. Weiterhin zeigt Abbildung 4.1 auf S. 42 die Wahrheitsmatrizen der sechs Modelle. Es wird ersichtlich, dass sowohl die untersuchten TL- als auch RW-Modelle mit leichten Schwankungen ein rein zufälliges Vorhersageverhalten aufweisen. Die größte Genauigkeit liefert das FusionNetTL-Modell mit 26,258%, wobei dies nur marginal besser als eine rein zufällige Prädiktion ist.

Tabelle 4.2 auf S. 41 listet die Resultate für das zweite Bewertungskriterium auf, wobei sich in Abbildung 4.2 auf S. 44 ebenfalls die entsprechenden Wahrheitstabellen der Modelle, welche auf sechs Epochen trainiert wurden,

[36] Google Colab bietet nur begrenzten Zugriff auf GPUs und TPUs, wodurch einzelne Einheiten wie eine bestimmte GPU nicht gezielt angesteuert werden können. Durch diese Nutzungslimitierungen soll laut Google Colab eine dynamischen Verteilung der zur Verfügung stehenden Ressourcen gewährleistet werden. Siehe auch `https://colab.research.google.com/`, zuletzt abgerufen am 29.09.2021.

AudioNet		**LyricsNet**		**FusionNet**	
Modell	Acc_{init} [%]	Modell	Acc_{init} [%]	Modell	Acc_{init} [%]
TL	24,062	TL	23,987	TL	**26,258**
RW	25,000	RW	24,268	RW	25,000

Tabelle 4.1: Genauigkeiten Acc_{init} der Modelle für das erste Bewertungskriterium.

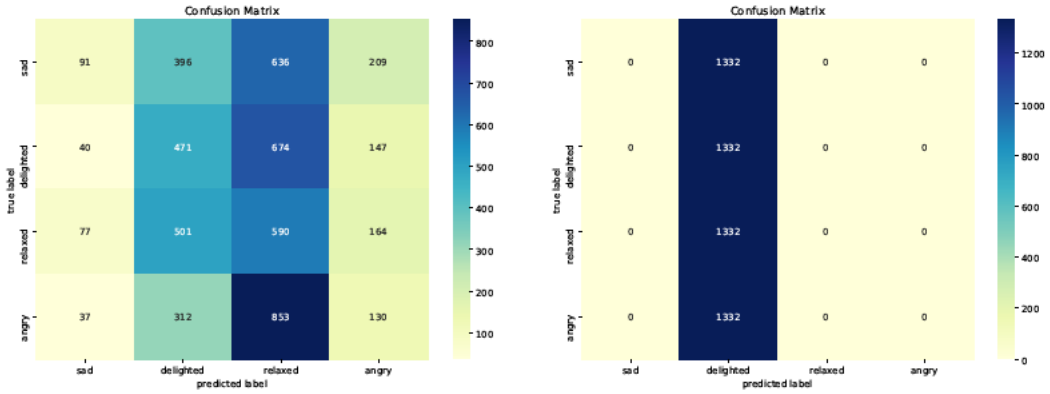
finden. Für die restlichen Epochen sind die jeweiligen Wahrheitstabellen im Anhang A hinterlegt. Wie in Abbildung 4.3 auf S. 45 zu erkennen, nimmt die durchschnittliche Treffergenauigkeit für alle Modelle, mit Ausnahme des LyricsNetRW-Modells, bis $i = 6$ Epochen zu. Das LyricsNetRW-Modell zeigt über alle untersuchten Epochen ein rein zufälliges Prädiktionsverhalten. Für die Modelle AudioNetTL, AudioNetRW und FusionNetRW nimmt die erzielte durchschnittliche Genauigkeit wieder leicht ab. Die Modelle LyricsNetTL (36,505%) und FusionNetTL (57,245%) erreichen bei einem Training von $i = 8$ Epochen sogar noch leicht höhere Werte, während die Modelle AudioNetTL (50,639%), AudioNetRW (62,634%) und FusionNetRW (62,669%) ihr Maximum für einen Trainingsprozess von $i = 6$ Epochen erzielen. Entsprechend wird die höchste Genauigkeit aller Modelle mit $Acc_6 = 62,669\%$ vom FusionNetRW6-Modell erreicht.

Darüber hinaus sind in Tabelle 4.3 auf S. 43 die Abdeckungen (Recall) der einzelnen Emotionsklassen für jedes Modell aus Tabelle 3.1 gegeben. Dies soll einen genaueren Einblick darüber geben, wie gut die jeweiligen Modelle die unterschiedlichen Emotionsklassen vorhersagen können. Die Recall-Werte Rec in Gleichung (4.2) berechnen sich aus dem Quotienten der richtig positiven Fälle (r_p) und der Summe aus richtig positiven (r_p) und falsch negativen (f_n) Fällen:

$$Rec = \frac{r_p}{r_p + f_n} \tag{4.2}$$

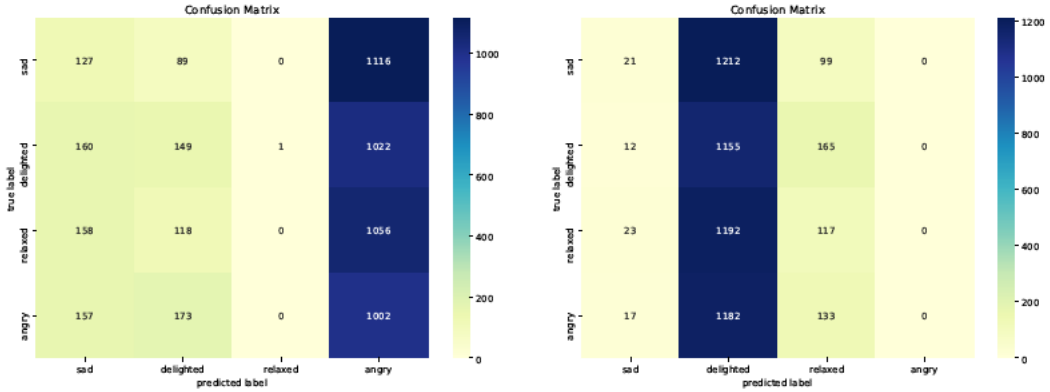
Modelle	**2 Epochen** Acc_2 [%]	**4 Epochen** Acc_4 [%]	**6 Epochen** Acc_6 [%]	**8 Epochen** Acc_8 [%]
AudioNetTL	42,53 ± 1,37	47,335 ± 1,1	**50,639 ± 1,27**	50,525 ± 1,33
AudioNetRW	58,896 ± 2,57	60,492 ± 2,84	**62,632 ± 1,73**	61,675 ± 3,06
LyricsNetTL	29,523 ± 1,31	33,878 ± 1,73	35,473 ± 1,44	**36,505 ± 1,2**
LyricsNetRW	25,3 ± 1,09	**25,432 ± 0,76**	25,131 ± 1,27	24,531 ± 0,85
FusionNetTL	49,775 ± 1,54	51,614 ± 2,15	55,556 ± 2,8	**57,245 ± 1,49**
FusionNetRW	57,546 ± 1,56	59,102 ± 3,53	**62,669 ± 0,53**	62,369 ± 2,3

Tabelle 4.2: Genauigkeiten Acc und Standardabweichung aller Modelle für das zweite Bewertungskriterium.



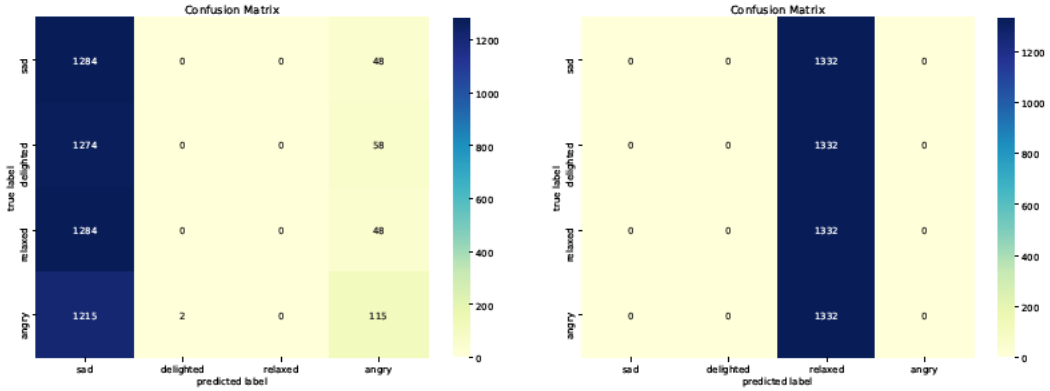
(a) AudioNetTL.

(b) AudioNetRW.



(c) LyricsNetTL.

(d) LyricsNetRW.



(e) FusionNetTL.

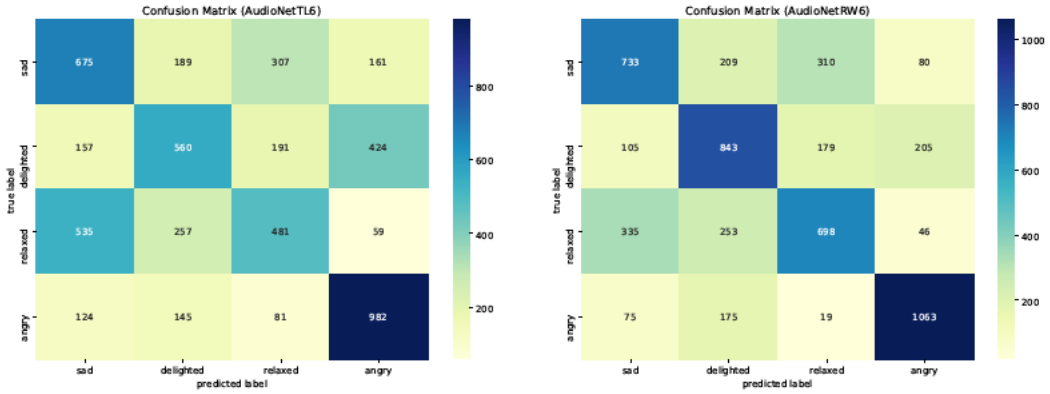
(f) FusionNetRW.

Abbildung 4.1: Wahrheitsmatritzen aller Modelle für das erste Bewertungskriterium.

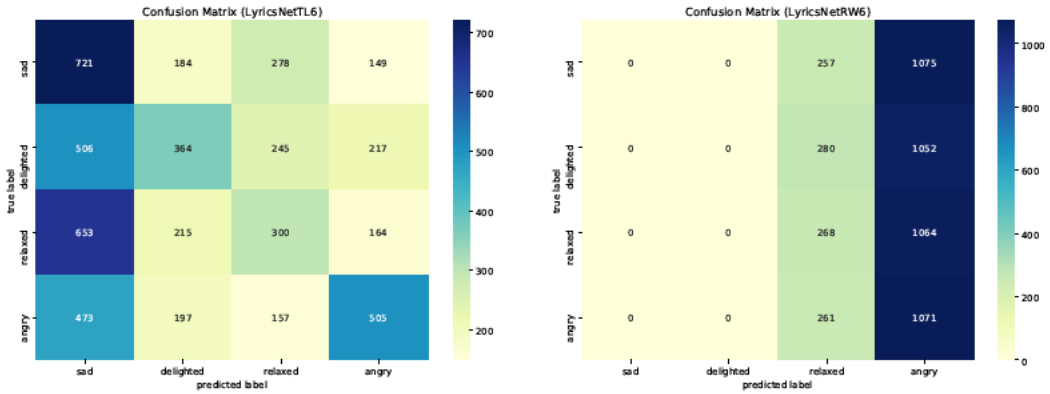
Die höchsten Recall-Werte werden für die Klassen „traurig“, „fröhlich“ und „wütend“ von Modellen erreicht, welche für sechs bzw. acht Epochen trainiert wurden. So erreicht das Modell AudioNetRW88 für die Klasse „traurig“ einen Recall-Wert von $Rec = 0,69$. Das Modell FusionNetRW6 erzielt für „fröhlich“ $Rec = 0,652$, während der Recall-Wert für „wütend“ vom FusionNetRW8-Modell 0,889 beträgt. Für „entspannt“ erreicht das Modell AudioNetRW2, welches für zwei Epochen trainiert wurde, einen Wert von $Rec = 0,568$.

Modelle		**Emotionsklassen**			
Name	Epochen	Traurig	Fröhlich	Entspannt	Wütend
AudioNetTL	2	0,341	0,314	0,444	0,602
	4	0,333	0,313	0,538	0,709
	6	0,507	0,42	0,361	0,737
	8	0,351	0,357	0,503	0,809
AudioNetRW	2	0,39	0,574	**0,568**	0,824
	4	0,638	0,525	0,415	0,842
	6	0,55	0,633	0,524	0,798
	8	**0,69**	0,531	0,413	0,833
LyricsNetTL	2	0,366	0,34	0,297	0,177
	4	0,546	0,29	0,15	0,369
	6	0,541	0,273	0,225	0,379
	8	0,314	0,387	0,397	0,362
LyricsNetRW	2	0,0	0,407	0,39	0,215
	4	0,405	0,0	0,201	0,411
	6	0,0	0,0	0,201	0,804
	8	0,598	0,0	0,199	0,185
FusionNetTL	2	0,369	0,323	0,529	0,771
	4	0,419	0,369	0,469	0,807
	6	0,462	0,468	0,47	0,822
	8	0,534	0,465	0,455	0,836
FusionNetRW	2	0,49	0,568	0,489	0,755
	4	0,516	0,583	0,507	0,758
	6	0,529	**0,652**	0,509	0,817
	8	0,565	0,573	0,468	**0,889**

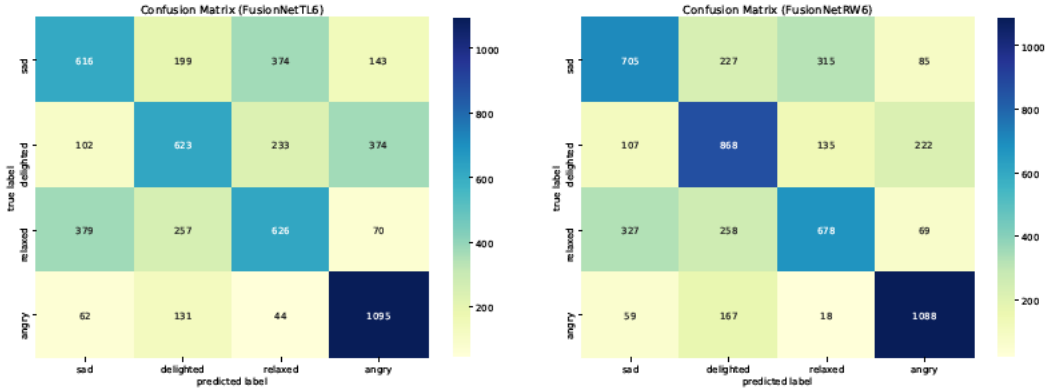
Tabelle 4.3: Recall-Werte pro Emotionsklasse.



(a) AudioNetTL6. (b) AudioNetRW6.



(c) LyricsNetTL6. (d) LyricsNetRW6.



(e) FusionNetTL6. (f) FusionNetRW6.

Abbildung 4.2: Wahrheitsmatritzen aller Modelle, die für 6 Epochen trainiert wurden.

Das dritte Bewertungskriterium wird abschließend in Tabelle 4.4 auf S. 45 durch das relative Verhältnis der Trainingsdauern Q_t für die untersuchten Epochen angegeben. Die absoluten Trainingsdauern t_i sind im Anhang B zusammen mit den verwendeten GPUs aufgeführt. Es zeigt sich, dass alle RW-Modelle ein relatives Verhältnis von $Q_t > 1$ aufweisen und somit mehr Zeit für den Trainingsprozess in Anspruch nehmen als die entsprechenden TL-Modelle. Die Implementierung der RW-Modelle, welche auf der LyricsNet-Architektur basieren, benötigt dabei ungefähr neunmal so viel Zeit für den Trainingsprozess wie die kon-

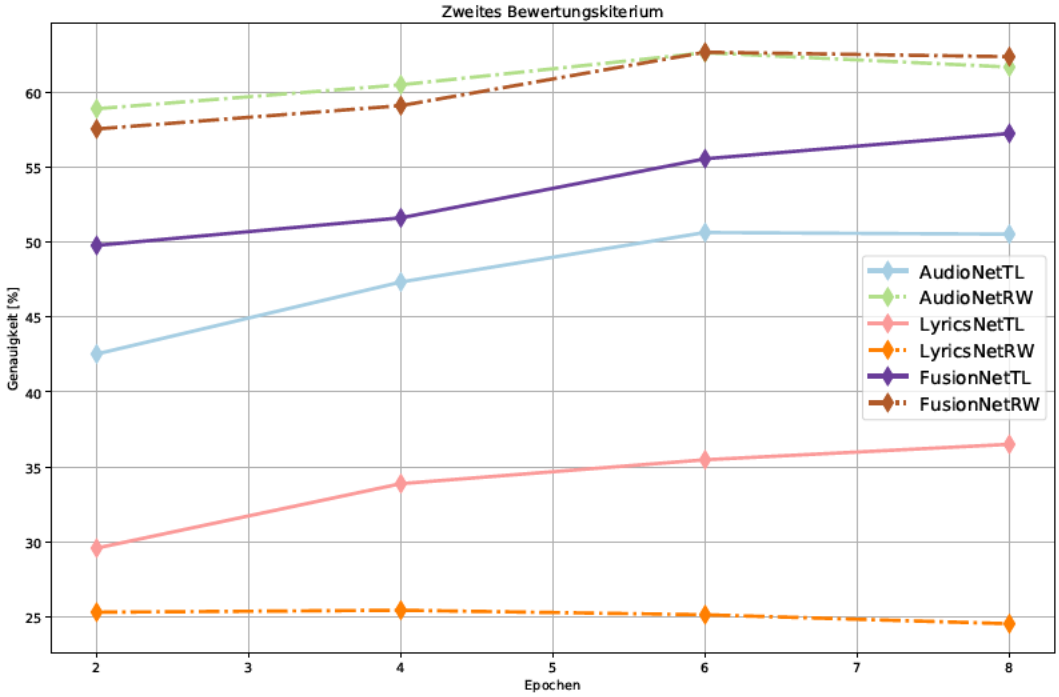


Abbildung 4.3: Verlauf der erzielten Genauigkeiten aller Modelle über die Anzahl der Epochen.

Modelle	**2 Epochen** $Q_{t,2}$	**4 Epochen** $Q_{t,4}$	**6 Epochen** $Q_{t,6}$	**8 Epochen** $Q_{t,8}$
AudioNet	3,4	3,5	3,6	3,6
LyricsNet	9,0	8,8	9,6	9,8
FusionNet	5,3	7,2	5,7	5,8

Tabelle 4.4: Relative Trainingsdauern.

kurrierenden TL-Modelle. Während die RW-Modelle mit AudioNet-Architektur jedoch nur mit einem ca. 3,5-mal längeren Training aufwarten können, bewegt sich die Trainingsdauer der FusionNetRW-Modelle in der Größenordnung von $Q_t = 5,3$ (zwei Epochen) bis $Q_t = 7,2$ (vier Epochen).

4.2 Diskussion

Im Folgenden soll die in Abschnitt 1.2 gestellte Forschungsfrage anhand der Ergebnisse aus dem vorangegangen Abschnitt diskutiert werden. Dabei sollen zunächst die drei Bewertungskriterien aus dem vorangegangenen Abschnitt 4.1 betrachtet werden. Weiterhin sollen anhand der ausgewerteten Kriterien die verwendeten Modelle, welche in den Abschnitten 3.1, 3.2, 3.3 vorgestellt wurden, erörtert werden. Abschließend sollen mit Blick auf den zugrundeliegenden Datensatz aus Abschnitt 3.4 die verwendeten Daten und ihre Rolle bei der Aus-

wertung der Modelle näher beleuchtet werden.

Wie sich beim Betrachten der Ergebnisse für das erste Bewertungskriterium in Tabelle 4.1 zeigt, fungieren sowohl RW- als auch TL-Modelle ohne vorheriges Training als schlechte Prädiktoren für die gestellte Aufgabe der automatischen Emotionserkennung. Dies ist zumindest für die RW-Modelle durchaus zu erwarten, da diese Modelle mit zufälligen Gewichtungen zwischen den verschiedenen Layern implementiert wurden. Ein besseres Vorhersageverhalten als eine zufällige Vorhersage von $Acc = 25\%$ wäre in diesem Fall eher ungewöhnlich gewesen. Ein derartiges Verhalten war allerdings nicht unbedingt für die TL-Modelle zu erwarten. Aufgrund des bereits vorhandenen Wissens aus einer anderen Quelldomäne $\mathcal{D}_{S,A}$ bzw. $\mathcal{D}_{S,L}$ war es im Vorhinein unklar, ob die Modelle für die Bewältigung einer neuen Aufgabe wie der automatischen Emotionserkennung dieses Wissen ohne weiteres Training würden nutzen können, welches letztendlich zu einer verbesserten Performanz hätte führen können. Der Test des ersten Bewertungskriteriums zeigt allerdings deutlich, dass sowohl RW- als TL-Modelle weitere Trainingsprozesse in der Zieldomäne benötigen, um in der automatischen Emotionserkennung bestehen zu können. Außerdem lassen sich diese Ergebnisse als erste Indikatoren dafür verstehen, dass die verwendeten TL-Modelle für die Aufgabe der automatischen Emotionserkennung nur bedingt geeignet sein könnten.

Die Eignung der TL-Modelle kann mit einer Betrachtung der Ergebnisse des zweiten Bewertungskriteriums näher erörtert werden. Im Vergleich zum ersten Bewertungskriterium fällt auf, dass von allen untersuchten Modelle, mit Ausnahme des LyricsNetRW-Modells, größere Genauigkeiten als $Acc = 25\%$ erreicht werden und somit die Modelle keine rein zufällige Vorhersage mehr treffen. Dies kann damit gleichgesetzt werden, dass sich die Modelle über die Trainingsprozesse auf die neue Aufgabe der automatischen Emotionserkennung anpassen und diese Aufgabe mit unterschiedlich großem Erfolg erlernen konnten. Generell wird ersichtlich, dass mit zunehmender Trainingsdauer auch die Performanz der Modelle steigt, wenn auch die Genauigkeit für die Modelle AudioNetTL8, AudioNetRW8 und FusionNetRW8 wieder leicht abnimmt. Dies könnte ein Hinweis dafür sein, dass diese Modelle bereits eine Überanpassung an die verfügbaren Daten aufweisen. Um eine genauere Aussage darüber treffen zu können, wann die Gefahr einer Überanpassung bei den untersuchten Modellen besteht, müssten jedoch längere Trainingsdauern untersucht werden. Aufgrund des angewandten fünffachen Kreuzvalidierungsverfahren werden in Tabelle 4.2 weiterhin die Standardabweichungen der Modelle angegeben. Es lässt sich für alle Modelle eine geringe Standardabweichung erkennen, was für ihre Robustheit

spricht (Laurier et al., 2008). Die geringste Abweichung mit $\pm 0,53\%$ findet sich beim FusionNetRW6-Modell, welches ebenfalls die insgesamt beste Performanz mit $Acc = 62,669\%$ aufweist.

Die LyricsNetRW-Modelle erweisen sich jedoch auch nach längeren Trainingsprozessen als schlechte Prädiktoren, was ein Hinweis darauf sein kann, dass entweder ein größerer Datensatz oder eine längere Trainingsdauer benötigt wird, um das XLNet-Modell von Grund auf auf eine neue Aufgabe zu trainieren. Mit Blick darauf, wie das XLNet-Modell aus Z. Yang et al. (2019) trainiert wurde, fällt vor allem der sehr viel größere Datensatz und die längere Trainingsdauer auf. Z. Yang et al. (2019) orientierten sich dabei am BERT-Modell von Devlin et al. (2018) und trainierten ihr XLNet-Modell für 40 Epochen. Diesbezüglich ist es also nicht verwunderlich, dass die LyricsNetRW-Modelle für alle untersuchten Trainingsdauern eine schlechte Performanz zeigen.

Weiterhin ist in Tabelle 4.2 zu erkennen, dass einzig die LyricsNetTL-Modelle eine höhere Genauigkeit als ihre gegenüberstehenden RW-Modelle aufweisen. Der größte Unterschied ergibt sich für $i = 8$ Epochen mit einer Differenz von $\Delta Acc = 11,974\%$. In diesem Fall kann von einem „positiven“ Transfer gesprochen werden. Allgemein wird unter einem positiven Transfer bei gegebener Quell- und Zieldomäne $\mathcal{D}_S$ und $\mathcal{D}_T$ mit entsprechenden Quell- und Zielaufgaben $\mathcal{T}_S$ und $\mathcal{T}_T$ verstanden, dass eine Zielvorhersagefunktion $f_{T2}(\cdot)$, welche mit einem Transfer-Learning-Prozess aus $\mathcal{D}_S$ und $\mathcal{D}_T$ bestimmt wurde, eine höhere Performanz zeigt als eine Zielvorhersagefunktion $f_{T1}(\cdot)$, welche nur aus $\mathcal{D}_T$ bestimmt wurde (Weiss et al., 2016). In diesem Fall kann der positive Transfer als Indikator gesehen werden, dass Quell- und Zieldomäne $\mathcal{D}_{S,L}$ und $\mathcal{D}_{T,L}$ miteinander verknüpft sind und Transfer Learning einen positiven Effekt auf die Entwicklung derartiger Modelle haben kann. Mit einer durchschnittlichen Genauigkeit von $Acc = 36,505\%$ stellt sich das LyricsNetTL8-Modell im Vergleich mit den AudioNet- und FusionNet-Modellen jedoch als eher schwacher Prädiktor für die vier Emotionsklassen dar. Mit Blick auf die Recall-Werte in Tabelle 4.3 schneiden auch die anderen LyricsNetTL-Modelle als schwache bis durchschnittliche Prädiktoren ab. So erlangen die Modelle LyricsNetTL4 und LyricsNetTL6 mit über 0,54 für die Klasse „traurig“ respektable Werte, erzielen aber schlechte Ergebnisse mit 0,15 (LyricsNetTL4) bzw. 0,225 (LyricsNetTL6) für die Klasse „entspannt“. Das ausgewogenste Vorhersageverhalten zeigt auch hier das LyricsNetTL8-Modell mit $Rec = 0,314$ für „traurig“, $Rec = 0,387$ für „fröhlich“, $Rec = 0,397$ für „entspannt“ und $Rec = 0,362$ für „wütend“. Bezüglich der erzielten Recall-Werte ist es ebenso interessant zu sehen, dass alle LyricsNetRW-Modelle für eine oder mehrere Klassen keine Vorhersage treffen

können. So erreicht LyricsNetRW6 zwar mit $Rec = 0,804$ ein gutes Prädiktionsverhalten für Musikstücke der Klasse „wütend“, ist aber nicht in der Lage, die Klassen „traurig“ und „fröhlich“ (beide mit $Rec = 0,0$) zu erkennen und vorherzusagen.

Als nächstes sollen die AudioNet-Modelle untersucht und diskutiert werden. Vorangestellt fällt auf, dass die größten Genauigkeiten von den Modellen erreicht werden, welche für sechs Epochen trainiert wurden. So erzielt das AudioNetTL6-Modell eine Genauigkeit von $Acc = 50,639\%$, während das AudioNetRW6-Modell sogar $Acc = 62,634\%$ erreicht. Im Kontrast zu den LyricsNet-Modellen wird ersichtlich, dass für alle AudioNet-Modelle ein negativer Transfer vorliegt. Unter negativem Transfer wird in Analogie zum positiven Transfer verstanden, dass eine Zielvorhersagefunktion $f_{T1}(\cdot)$ eine bessere Performanz als eine Funktion $f_{T2}(\cdot)$ aufweist (Weiss et al., 2016). So ergibt sich für das AudioNetTL6- und das AudioNetRW6-Modell eine Differenz von $\Delta Acc = 11,995\%$. In diesem Fall scheint also der Einsatz von TL-Modellen, wie dem DenseNet-Modell, einen negativen Effekt auf die Performanz der untersuchten Modelle zu haben. Wann genau ein negativer Transfer zwischen Quell- und Zieldomäne jedoch zu erwarten ist, ist noch nicht hinreichend erforscht. Obwohl Z. Wang, Dai, Póczos und Carbonell (2019) in ihrer Studie einen ersten Ansatz dazu vorstellen, wie negativer Transfer erkannt und vermieden werden kann, bedarf es laut Zhuang et al. (2021) noch weiterer Forschung, wann genau mit einem negativem Transfer zu rechnen ist.

Als Begründung soll im Falle dieser Arbeit die Diskrepanz zwischen der verwendeten Quell- und Zieldomäne $\mathcal{D}_{S,A}$ und $\mathcal{D}_{T,A}$ herangezogen werden. So scheinen $\mathcal{D}_{S,A}$ und $\mathcal{D}_{T,A}$ bezüglicher ihrer zugrundeliegenden Datenstrukturen weit genug auseinander zu liegen, sodass es zu einem negativen Transfer kommt. Die verwendete Quelldomäne $\mathcal{D}_{S,A}$ umfasst dabei digitale Bilder aus 1000 Klassen wie u. a. Zitronen, Ballons oder Lokomotiven[37], wohingegen die Zieldomäne $\mathcal{D}_{T,A}$ eine Repräsentation von Emotionen in Form von Spektrogramme voraussetzt. Da Modelle wie das DenseNet-Modell auf die Detektion von Mustern in digitalen Bildern wie etwa Kanten, Formen etc. trainiert wurden, ist es nach wie vor unklar, anhand welcher Muster Emotionen innerhalb eines Spektrogrammes erkannt werden können. Die Bestimmung solcher Muster (z. B. über Feature Maps), welche das DenseNet-Modell erlernt haben könnte, würde einen interessanten Ansatz für weitergehende Untersuchungen darstellen. Da dies im gegebenen Rahmen allerdings nicht ausführlich zu erörtern ist, soll dieser Ansatz als Vermerk für zukünftige Arbeiten verstanden und im weiteren Verlauf nicht

[37] siehe auch `https://www.image-net.org/challenges/LSVRC/2012/browse-synsets.php`, zuletzt abgerufen am 29.09.2021.

näher beleuchtet werden. Auch wenn die Arbeitsweise des DenseNet-Modells ferner als eine Art „Blackbox“ angenommen wird, kann trotzdem auch die Vorverarbeitung der Audiosignale kritisch hinterfragt werden. Obwohl die Berechnung von Spektrogrammen durchaus gängig ist (siehe Tabelle 2.2), könnten durch die Umformung der Spektrogramme in ein $(3 \times 224 \times 224)$-Format mithilfe der Reshape-Funktion Artefakte entstanden sein, welche die MER zusätzlich erschweren und die Diskrepanz zwischen Quell- und Zieldomäne weiter vergrößern.

Darüber hinaus zeigen auch bisherige Forschungsergebnisse, dass eine komplexe Repräsentation von Emotionen durch u. a. Spektrogramme durchaus schwierig von Algorithmen zu durchdringen ist. Sowohl die subjektive Natur von wahrgenommenen Emotionen als auch die fehlende, einheitliche Taxonomie von Emotionen sind Gründe, die die automatische Emotionserkennung erschweren können (Y.-H. Yang & Chen, 2012). Die bereits von Celma Herrada, Boyer, Serra et al. (2006) formulierte „semantische Lücke“ zwischen objektiven Merkmalen von Musikstücken und ihrer subjektiven Wahrnehmung zeigte sich insbesondere durch die geringen Fortschritte, welche zwischen 2007 und 2019 bei der MIREX *Audio Mood Classification Task* von modernsten Lösungsansätzen erzielt wurden. Über einen langen Zeitraum wurde hier vom besten Algorithmus für fünf Emotionsklassen eine Genauigkeit von 69,8% erreicht (Panda et al., 2020). Zwar konnten Deep-Learning-Ansätze, welche u. a. von T. Liu et al. (2018) implementiert wurden, diesen Wert leicht verbessern, jedoch scheint die semantische Lücke nach wie vor für die Emotionserkennung zu bestehen und auch nicht von Deep-Learning-Strategien vollständig überwunden werden zu können. So konnten z. B. T. Liu et al. (2018) mit einem audiobasierten Ansatz für vier Emotionsklassen unter Verwendung eines CNNs eine Genauigkeit von 72,4% erzielen. Die semantische Lücke wird insbesondere mit Blick auf andere Bereiche wie der Genre-Klassifikation von Musikstücken ersichtlich. Palanisamy et al. (2020) konnten so ebenfalls mit der DenseNet201-Architektur und Transfer Learning für drei verschiedene Datensätze Genauigkeiten von über 85% erreichen. Die Ergebnisse in Tabelle 4.2 spiegeln allerdings die bekannten Schwierigkeiten der automatischen Emotionserkennung auch für TL-Modelle deutlich wieder. Trotz der Diskrepanz zwischen Quell- und Zieldomäne erzielen die AudioNetTL-Modelle mit einer Genauigkeit von 50,639% (AudioNetTL6) dennoch ein durchaus beachtliches Ergebnis.

Ein genauerer Einblick in die Funktionsweise der AudioNet-Modelle wird durch die Recall-Werte aus Tabelle 4.3 gegeben. So zeigen sowohl die AudioNetTL- als auch die AudioNetRW-Modelle ein gutes Vorhersageverhalten für die Klas-

se „wütend“. Den besten Wert erreicht hierbei das AudioNetRW4-Modell mit $Rec = 0,842$. Im Vergleich erzielt das korrespondierende AudioNetTL4-Modell einen ebenfalls guten Recall-Wert von 0,709. Die erreichten Recall-Werte offenbaren, dass die Modelle mithilfe der gegebenen Spektrogramme die Emotion „wütend“ gut identifizieren können. Dies kann damit begründet werden, dass die Emotion „wütend“ im zweidimensionalen VA-Raum von Russell den 4. Quadranten mit hohem Arousalanteil und niedriger Valenz beschreibt (siehe auch Abbildung 2.5). Arousal wird in der Regel mit Merkmalen wie Rhythmus und der im Audiosignal enthaltenen Energie assoziiert (Delbouys et al., 2018), welches offensichtlich gut durch die AudioNet-Modell in den Spektrogrammen detektiert werden kann. Das Vorhersageverhalten aller AudioNet-Modelle ist hingegen für die übrigen drei Emotionsklassen deutlich schwächer ausgeprägt, auch wenn es hier zu Schwankungen der erlangten Recall-Werte in Abhängigkeit von den untersuchten Trainingsdauern kommt. Aus diesem Grund sollen für die restlichen Klassen ebenfalls die beiden AudioNet-Modelle betrachtet werden, welche für vier Epochen trainiert wurden. Interessant hierbei ist, dass beide Modelle für die Klassen „traurig“, „fröhlich“ und „entspannt“ ein konträres Vorhersageverhalten aufzeigen. So erweist sich das AudioNetTL4-Modell als besserer Prädiktor für die Klasse „entspannt“ mit $Rec = 0,538$, während die Klassen „traurig“ und „fröhlich“ von AudioNetRW4 mit Recall-Werten von $Rec = 0,638$ („traurig“) und $Rec = 0,525$ („fröhlich“) bestimmt werden können. Die Performanz vom AudioNetRW4-Modell steht dabei im Kontrast zu Befunden in der Literatur (Laurier et al., 2008; Delbouys et al., 2018), da die im Audiosignal befindlichen Informationen zu Rhythmus und Energie ebenfalls für „entspannte“ bzw. „ruhige“ Musikstücke gute Indikatoren darstellen sollen. Demgegenüber steht das AudioNetTL4-Modell im Einklang mit den vorangegangenen Feststellungen aus der bekannten Literatur.

Des Weiteren können auch bei den untersuchten FusionNet-Modellen interessante Resultate beobachtet werden. Zum einen zeigt sich, dass zumindest alle FusionNet-Modelle mit Transfer Learning höhere Genauigkeiten erreichen als die entsprechenden AudioNet- und LyricsNet-Modelle. Dies kann als Hinweis darauf verstanden werden, dass die Kombination von audio- und textbasierten Informationen für die automatische Emotionserkennung bessere Ergebnisse liefern kann. Die höchste Genauigkeit erzielt das FusionNetTL8-Modell mit $Acc = 57,245\%$, welches eine Verbesserung von $\Delta Acc = 6,72\%$ im Vergleich zum AudioNetTL8-Modell und sogar $\Delta Acc = 20,74\%$ verglichen mit dem LyricsNetTL8-Modell mit sich bringt. Werden jedoch FusionNetTL- und FusionNetRW-Modelle nebeneinander gestellt, fällt erneut auf, dass auch hier ein negativer Transfer zwischen Quell- und Zieldomäne vorliegt. So zeigt das beste FusionNetTL-

Modell eine um $\Delta Acc = 5,124\%$ schlechtere Performanz als das entsprechende FusionNetRW-Modell (FusionNetRW8 mit $Acc = 62,369\%$). Eine mögliche Begründung hierfür könnte sein, dass sich der bei den AudioNet-Modellen vorliegende negative Transfer ebenfalls auf die FusionNet-Modelle übertragen hat. Durch den positiven Transfer der LyricsNet-Modelle fällt der negative Transfer bei den FusionNet-Modellen jedoch nicht so prägnant aus wie bei den AudioNet-Modellen. Es zeigt sich also auch hier, dass eine semantische Lücke auch bei Modellen mit einem kombinierten Ansatz aus audio- und textbasierten Daten vorzuliegen scheint. Dies ist auch anhand von Beispielen aus der Literatur zu erkennen. So konnten u. a. G. Liu und Tan (2020) für vier Emotionsklassen mit einer Deep-Learning-Architektur ohne Transfer Learning, bestehend aus einem LSTM-Netzwerk für Audiosignale und einem Transformer-Netzwerk für Lyrics, eine Genauigkeit von 79,62% erreichen. Bemerkenswert ist auch das Verhalten der FusionNetRW-Modelle im Vergleich mit den AudioNetRW-Modellen. Obwohl mit dem FusionNetRW6-Modell die insgesamt höchste Genauigkeit aller Modelle mit $Acc = 62,669\%$ erzielt werden konnte, ist die Performanz der FusionNetRW-Modelle für sechs bzw. acht Epochen nur geringfügig besser als die der jeweiligen AudioNetRW-Modelle. Für eine Trainingsdauer von zwei und vier Epochen schneiden die FusionNetRW-Modelle sogar leicht schlechter ab als die entsprechenden AudioNetRW-Modelle. Ein Argument für das Vorhersageverhalten der FusionNetRW-Modelle könnte die durchweg schlechte Performanz der LyricsNetRW-Modelle sein. Die LyricsNetRW-Modelle weisen, wie bereits weiter oben beschrieben, für alle betrachteten Trainingsdauern ein rein zufälliges Vorhersageverhalten auf, welches die Gesamtperformanz der FusionNetRW-Modelle zusätzlich negativ beeinflussen könnte.

Mit Blick auf die Recall-Werte in Tabelle 4.3 lässt sich feststellen, dass es hier ein ähnliches Verhalten zwischen AudioNet- und FusionNet-Modellen gibt. So zeigt sich auch hier, dass alle FusionNetTL- und FusionNetRW-Modelle für die Emotionsklasse „wütend“ die höchsten Recall-Werte erreichen. Auch die Ergebnisse für die restlichen drei Klassen sind vergleichbar mit denen der AudioNet-Modelle. Insgesamt zeigt das FusionNetRW6-Modell mit $Rec = 0,652$ für die Klasse „traurig“ die beste Performanz, während FusionNetRW8 mit $Rec = 0,889$ für „wütend“ das beste Resultat liefert. Da die verwendete AudioNet-Architektur innerhalb der FusionNet-Struktur einen größeren Einfluss auf die Gesamtperformanz der FusionNet-Modelle zu haben scheint, sind die Ähnlichkeiten mit den reinen AudioNet-Modellen durchaus zu erwarten.

Abschließend sollen die Ergebnisse des dritten Bewertungskriteriums, aufgelistet in Tabelle 4.4, näher beleuchtet werden. Wie bereits in Abschnitt 4.1

beschrieben, zeigt sich, dass alle RW-Modelle eine größere Trainingsdauer aufweisen als ihre entsprechenden TL-Modelle. Aufgrund dieser teilweise enormen Zeitersparnis bieten TL-Modelle einen spannenden Vorteil im Vergleich zu Modellen ohne Transfer Learning. Die kürzeren Trainingsdauern waren allerdings durchaus zu erwarten, da durch das Einfrieren der Gewichtungen innerhalb der verwendeten Architekturen ein nicht unerheblicher Anteil der durchzuführenden Berechnungen wegfällt. Verkürzte Trainingsprozesse haben ebenfalls einen entscheidenden Einfluss auf die Bewertung der Praktikabilität von TL-Modellen in der automatischen Emotionserkennung. Verbindet man die deutlich kürzeren Trainingsdauern mit den erzielten Resultaten aus Tabelle 4.2, präsentieren sich TL-Modelle auch für die MER als eine durchaus ernsthafte Alternative zu vergleichbaren Modellen ohne Transfer Learning, da sie wesentlich schneller zu implementieren sind. Im Falle dieser Arbeit weisen vor allem die FusionNetTL-Modelle eine maximal 7,8% geringere Genauigkeit auf als die jeweiligen FusionNetRW-Modelle. Zusammen mit einer über fünf mal schnelleren Entwicklung derartiger FusionNet-Modelle könnte der Einsatz von TL-Modellen für MER-Anwendungen vergleichsweise interessant sein. Wenn aber ein besseres Vorhersageverhalten gefordert ist und zusätzlich die Trainingsdauer keine entscheidende Rolle bei der Modellentwicklung spielt, sollte auf die in dieser Arbeit untersuchten Modelle ohne Transfer Learning zurückgegriffen werden.

Ein weiterer Punkt, auf den eingegangen werden soll, ist die Wahl der TL-Modelle für die AudioNet- und LyricsNet-Architekturen. Anhand der Erkenntnisse, welche die drei Bewertungskriterien nach Torrey und Shavlik (2010) liefern, zeigen sowohl die verwendete DenseNet201- als auch die XLNet-Architektur diverse Vor- und Nachteile. Beide Architekturen sind hinsichtlich ihres Implementierungsaufwandes und ihrer Verfügbarkeit positiv zu bewerten. Sie sind über Python-Programmbibliotheken wie PyTorch leicht zugänglich, gut dokumentiert und wurden auf Grundlage von sehr umfangreichen Datensätzen trainiert, wodurch die Modelle für vielfältige Anwendungsbereiche eingesetzt werden können. Diese positiven Aspekte waren letztendlich auch ausschlaggebend für die finale Entscheidung, diese beiden Modelle in der vorliegenden Arbeit näher zu beleuchten. Wie sich aber vor allem in Hinblick auf das zweite Bewertungskriterium gezeigt hat, wirkt sich der Einsatz des gewählten DenseNet201-Modells mit Transfer Learning negativ auf die Aufgabe der automatischen Emotionserkennung aus. Obwohl die XLNet-Architektur mit Transfer Learning einen positiven Effekt aufweist, überwiegt der negative Transfer durch das DenseNet201-Modell in der Kombination aus beiden Architekturen. Aus diesem Grund würde es sich lohnen, weitere CNN-Modelle bezüglich ihrer Eignung für Transfer Learning innerhalb der automatischen MER zu untersuchen. Als Ausgangspunkt für

weitere Analysen würden sich zunächst andere populäre Modelle aus dem *torchvision*-Katalog[38] anbieten, wo sich eine breit gefächerte Auswahl an Modellen wie „AlexNet“ (Krizhevsky et al., 2012), „VGGNet“ (Simonyan & Zisserman, 2014) und „ResNet“ (K. He et al., 2016) wiederfindet. Allerdings ist hierbei zu beachten, dass diese Modelle ebenfalls auf einen ImageNet-Datensatz trainiert wurden, wodurch auch im Falle dieser Modelle die Gefahr von negativem Transfer besteht. Abhilfe könnte hier die von Pons und Serra (2019) beschriebene „MusiCNN“-Architektur schaffen. MusiCNN bietet dabei eine musikalisch motivierte CNN-Architektur, welche mithilfe von Spektrogrammen trainiert wurde. Pons und Serra (2019) stellen mehrere Modelle zur Verfügung, die jeweils auf einen von zwei Datensätzen trainiert wurden. Während für das Training einerseits auf den MSD-Datensatz mit 200 000 Musikstücken zurückgegriffen wurde, wurde andererseits auch der MagnaTagATune-Datensatz[39] mit einem Umfang von 19 000 Musikstücken verwendet. Der Einsatz eines MusiCNN-Modells für die automatische MER könnte aufgrund der stärkeren Verbindung von Quell- und Zieldomäne potentiell zu einem positiven Transfer führen. Allerdings fehlen im Gegensatz zu Modellen wie AlexNet oder VGGNet bislang exzessive Untersuchungen hinsichtlich der Eignung des MusiCNN-Modells für Transfer Learning. Dieser Umstand könnte zur Folge haben, dass die Verwendung eines MusiCNN-Modells mit einem anfänglich größeren Implementierungsaufwand verbunden ist. Weiterhin könnte ein ähnlicher Ansatz ebenfalls für textbasierte Modelle verfolgt werden. Da es jedoch bisher an TL-Modellen fehlt, welche auf einen großen Lyrics-Datensatz trainiert wurden, müssten zuerst derartige TL-Modelle entwickelt werden. Eine Annäherung von Quell- und Zieldomäne könnte in diesem Fall auch ein mögliches Resultat sein, wodurch der positive Transfer und damit die Performanz eines entsprechenden LyricsNet-Modells verbessert werden könnten.

Ferner soll ebenfalls der verwendete Datensatz diskutiert werden. Wie bereits weiter oben erwähnt, stellt die Subjektivität von Emotionen eine erhebliche Schwierigkeit für die automatische MER dar. Zusätzlich fehlt es an einer einheitlichen Taxonomie, wodurch es eine Vielzahl an konkurrierenden Emotionsmodellen gibt. All diese Modelle haben gemeinsam, dass sie nur ein vereinfachtes Abbild der Realität widerspiegeln können. Solche Vereinfachungen sind selbstverständlich notwendig, um Emotionen überhaupt mess- und greifbar zu machen. Allerdings werden dadurch auch komplexe Zusammenhänge unterschlagen, welches sich letztendlich auf die Performanz von MER-Modellen auswirken kann. In dieser Arbeit wurde ein Datensatz untersucht, in dem jedem Musik-

[38] `https://pytorch.org/vision/stable/models.html`, zuletzt abgerufen am 29.09.2021.
[39] kurz MTT, siehe auch `https://mirg.city.ac.uk/codeapps/the-magnatagatune-dataset`, zuletzt abgerufen am 29.09.2021.

stück genau eine von vier Emotionen zugeteilt wird, wobei die Beschränkung auf lediglich vier Emotionsklassen eine durchaus gängige Abstraktion der Realität ist. Durch die Verwendung einer größeren Anzahl an Emotionsklassen kann zwar die Realität detaillierter dargestellt werden, wodurch allerdings auch die Performanz von MER-Modellen in der Regel verringert wird (X. Yang et al., 2018, S. 17).

Darüber hinaus wird in diesem Fall auch außer Acht gelassen, dass aufgrund der Subjektivität von Emotionen einem Musikstück mehrere Emotionen zugeordnet werden können. So kann es sein, dass ein bestimmtes Musikstück von einer Person als traurig, von einer anderen Person aber als fröhlich empfunden wird. Eine solche Vereinfachung kann ebenfalls die automatische Emotionserkennung erschweren. Weiterhin werden sowohl dem Audiosignal als auch den Songtexten die gleiche Emotion zugewiesen. Dass es jedoch auch Musikstücke gibt, bei denen die vom Audiosignal und den Lyrics transportierten Emotionen deutlich auseinander gehen, kann am Beispiel des Songs „Pumped Up Kicks“ der US-amerikanischen Band „Foster the People“ beobachtet werden. Während die Musik als heiter oder fröhlich empfunden werden kann, zeichnet der Songtext ein düsteres Bild eines Amoklaufes[40]. Derartige Diskrepanzen zwischen den transportierten Emotionen von Audiosignal und Lyrics könnten sich ebenfalls negativ auf die automatische Emotionserkennung auswirken.

Abschließend soll erwähnt werden, dass sich die Stimmung eines Musikstückes im zeitlichen Verlauf sowohl innerhalb des Audiosignals als auch der Lyrics verändern kann. Eine solche dynamische Repräsentation von Emotionen stellt die automatische MER ebenfalls vor Probleme, da in diesem Fall kritisch hinterfragt werden kann, welche Emotion als primäre Emotion bei einem statischen Ansatz wie dem hier vorliegenden Datensatz gewählt wird. Die Erstellung eines entsprechend umfangreichen und aussagekräftigen Datensatzes, welcher die oben beschriebenen Limitierungen vermeidet, könnte die automatische MER gegebenenfalls ebenso vorantreiben. Im Zusammenspiel mit spezialisierten TL-Modellen wie dem MusiCNN-Modell von Pons und Serra (2019) wäre es vorstellbar, dass ein solcher Datensatz nur aus ein paar hundert bis tausend Datenpunkten bestehen müsste, um zufriedenstellende Ergebnisse im Bereich der automatischen MER zu erzielen.

[40] `https://www.chicagotribune.com/entertainment/ct-xpm-2011-10-03-ct-ent-1004-foster-lyrics-20111004-story.html`, zuletzt abgerufen am 29.09.2021.

Kapitel 5

Fazit

In der vorliegenden Arbeit wurde ein erster Versuch unternommen, Deep-Learning-Modelle mit Transfer Learning für die Aufgabe der automatischen Musikemotionserkennung einzusetzen und genauer zu erforschen. Hierbei wurde sowohl auf audio- und textbasierte Daten als auch auf die Kombination aus beiden Datentypen zurückgegriffen. Es konnte zunächst gezeigt werden, dass es bislang keine umfangreichen Betrachtungen von TL-Strategien im Bereich der automatischen MER gibt, weswegen mit dieser Arbeit eine Grundlage diesbezüglich geschaffen werden sollte.

Es zeigt sich allerdings, dass die in Abschnitt 1.2 gestellte Forschungsfrage dabei nicht eindeutig beantwortet werden kann. Während die untersuchten textbasierten DL-Modelle in Form von LyricsNet-Modellen von einem positiven Transfer profitieren, wird die Performanz von audiobasierten DL-Modellen (AudioNet-Modelle) und dadurch letztendlich auch von kombinierten Modellen (FusionNet-Modelle) durch einen negativen Transfer gemindert. Für die Minderung der Performanz konnte die Diskrepanz zwischen den verwendeten Quell- und Zieldomänen verantwortlich gemacht werden. Diese Diskrepanz wird durch die Wahl des DL-Modells „DenseNet201“, welches die Grundlage für die entwickelten AudioNet-Modelle darstellt, bedingt. Die Wahl des DenseNet201-Modells hat auch Auswirkung auf die Vorverarbeitung der Audiodaten, was als zusätzlicher Faktor für den beobachteten negativen Transfer in Betracht gezogen werden kann. Obwohl es bereits erste Untersuchungen gibt (Z. Wang et al., 2019), fehlt es immer noch an etablierten Mitteln, die Ursache und Wirkung von negativem Transfer im Vorfeld bestimmen zu können. Aus diesem Grund kann auch in dieser Arbeit die Frage, warum es zu einem negativen Transfer kommt, nicht endgültig geklärt werden. Es bedarf also noch weiteren Bemühungen, um die Anwendbarkeit von Transfer-Learning-Strategien in der MER genauer zu verstehen. Unabhängig von der Performanz der Modelle konnte für alle TL-Modelle ein wesentlich geringerer Implementierungsaufwand festgestellt

werden, wodurch Transfer Learning auch für die MER weiterhin einen interessanten Ansatz darstellen kann.

Die unbeantworteten Fragen können allerdings auch als Möglichkeiten für zukünftige Arbeiten verstanden werden. So können anhand der in dieser Arbeit beschriebenen Bewertungskriterien weitere Deep-Learning-Modelle, welche mutmaßlich eine geringere Diskrepanz aufweisen könnten, mit Transfer-Learning-Strategien für die MER untersucht werden. Eine solche potentielle Architektur wurde mit dem MusiCNN-Modell von Pons und Serra (2019) vorgestellt. Da eine derartige Architektur für textbasierte Ansätze bisher fehlt, könnte hingegen die Entwicklung eines generischen Lyrics-Modells, welches allgemein mit Songtexten trainiert wurde, ebenfalls spannend sein.

Darüber hinaus könnten neben Transfer Learning auch verwandte Konzepte wie *Semi-Supervised Learning* oder *Multi-Task Learning* für die automatische MER interessant sein und untersucht werden. Beide Konzepte hätten, wie Transfer Learning, das Potential, die Performanz von MER-Modellen, welche mit kleinen Datensätzen entwickelt wurden, zu verbessern, verfolgen jedoch unterschiedliche Ansätze. Semi-Supervised Learning kombiniert dabei während des Trainingsprozesses einen umfangreichen, ungelabelten Datensatz mit einem kleinen, gelabelten Datensatz. Ein entsprechend umfangreicher Datensatz ohne Informationen über die von einem Musikstück transportierten Emotionen könnte leicht aus verfügbaren Datensätze wie dem Million Song Dataset erstellt werden, während die jeweiligen Songtexten von Diensten wie `genius.com` bezogen werden könnten. Beim Multi-Task Learning wird hingegen ein Modell entwickelt, welches mehrere Aufgaben gleichzeitig erlernen soll. Hierbei wird die Hoffnung darauf gesetzt, dass sich die erlernten Aufgaben gegenseitig positiv beeinflussen. Im Falle der MER könnten so eventuell Emotionen zusammen mit dem Genre oder der Instrumentierung von Musikstücken simultan von einem Modell erlernt und vorhergesagt werden.

Literaturverzeichnis

Abdillah, J., Asror, I., Wibowo, Y. F. A. et al. (2020). Emotion Classification of Song Lyrics Using Bidirectional LSTM Method with Glove Word Representation Weighting. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4* (4), 723–729.

Agrawal, Y., Shanker, R. G. R. & Alluri, V. (2021). Transformer-based Approach Towards Music Emotion Recognition from Lyrics. *arXiv preprint arXiv:2101.02051*.

Ali, S. O. & Peynircioğlu, Z. F. (2006). Songs and Emotions: Are Lyrics and Melodies Equal Partners? *Psychology of Music*, *34* (4), 511–534.

Andjelkovic, I., Parra, D. & O'Donovan, J. (2019). Moodplay: Interactive Music Recommendation Based on Artists' Mood Similarity. *International Journal of Human-Computer Studies*, *121*, 142–159.

Balkwill, L.-L. & Thompson, W. F. (1999). A Cross-Cultural Investigation of the Perception of Emotion in Music: Psychophysical and Cultural Cues. *Music Perception*, *17* (1), 43–64.

Bhattacharya, A. & Kadambari, K. (2018). A Multimodal Approach Towards Emotion Recognition of Music Using Audio and Lyrical Content. *arXiv preprint arXiv:1811.05760*.

Çano, E. & Morisio, M. (2017). MoodyLyrics: A Sentiment Annotated Lyrics Dataset. In *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence* (S. 118–124).

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C. & Slaney, M. (2008). Content-based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, *96* (4), 668–696.

Celma Herrada, Ò., Boyer, H., Serra, X. et al. (2006). Bridging the Music Semantic Gap. In *Proceedings of the Workshop on Mastering the Gap, From Information Extraction to Semantic Representation, held in conjunction with the European Semantic Web Conference; 2006 jun 11-14; budva, montenegro.[aachen]: CEUR Workshop Proceedings; 2006.*

Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J. & Moussallam, M. (2018). Music Mood Detection Based on Audio and Lyrics with Deep

Neural Net. *Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 370—375.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). Imagenet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (S. 248–255).

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Du, P., Li, X. & Gao, Y. (2020). Dynamic Music Emotion Recognition Based on CNN-BiLSTM. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (S. 1372–1376).

Eerola, T., Lartillot, O. & Toiviainen, P. (2009). Prediction of Multidimensional Emotional Ratings in Music from Audio Using Multivariate Regression Models. In *ISMIR* (S. 621–626).

Eerola, T. & Vuoskoski, J. K. (2011). A Comparison of the Discrete and Dimensional Models of Emotion in Music. *Psychology of Music*, *39* (1), 18–49.

Ekman, P. (1992). An Argument for Basic Emotions. *Cognition & emotion*, *6* (3-4), 169–200.

Ekman, P. & Friesen, W. V. (2003). *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*. ISHK.

Gong, Y., Chung, Y.-A. & Glass, J. (2021). AST: Audio Spectrogram Transformer. *arXiv preprint arXiv:2104.01778*.

Gundlach, R. H. (1935). Factors Determining the Characterization of Musical Phrases. *The American Journal of Psychology*, *47* (4), 624–643.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 770–778).

He, N. & Ferguson, S. (2020). Multi-view Neural Networks for Raw Audio-based Music Emotion Recognition. In *2020 IEEE International Symposium on Multimedia (ISM)* (S. 168–172).

Hevner, K. (1936). Experimental Studies of the Elements of Expression in Music. *The American Journal of Psychology*, *48* (2), 246–268.

Hu, X. & Downie, J. S. (2007). Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata. In *ISMIR* (S. 67–72).

Hu, X., Downie, J. S. & Ehmann, A. F. (2009). Lyric Text Mining in Music Mood Classification. *American Music*, *183* (5,049), 2–209.

Hu, Y., Chen, X. & Yang, D. (2009). Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method. In *ISMIR* (S. 123–128).

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 4700–4708).

Jeon, B., Kim, C., Kim, A., Kim, D., Park, J. & Ha, J. (2017). Music Emotion Recognition via End-to-End Multimodal Neural Networks. In *RecSys Posters.*

Juslin, P. N. & Lindström, E. (2010). Musical Expression of Emotions: Modelling Listeners' Judgements of Composed and Performed Features. *Music Analysis*, *29* (1-3), 334–364.

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., ... Turnbull, D. (2010). Music Emotion Recognition: A State of the Art Review. In *Proc. ISMIR* (Bd. 86, S. 937–952).

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Krumhansl, C. L. (1997). An Exploratory Study of Musical Emotions and Psychophysiology. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, *51* (4), 336.

Laurier, C., Grivolla, J. & Herrera, P. (2008). Multimodal Music Mood Classification Using Audio and Lyrics. In *2008 Seventh International Conference on Machine Learning and Applications* (S. 688–693).

LeCun, Y., Kavukcuoglu, K. & Farabet, C. (2010). Convolutional Networks and Applications in Vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (S. 253–256).

Li, T. & Ogihara, M. (2003). Detecting Emotion in Music. In *4th International Symposium on Music Information Retrieval – ISMIR 2003* (S. 239—240). Johns Hopkins University.

Liu, G. & Tan, Z. (2020). Research on Multi-modal Music Emotion Classification Based on Audio and Lyirc. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (Bd. 1, S. 2331–2335).

Liu, T., Han, L., Ma, L. & Guo, D. (2018). Audio-based Deep Music Emotion Recognition. In *AIP Conference Proceedings* (Bd. 1967, S. 040021).

Liu, X., Chen, Q., Wu, X., Liu, Y. & Liu, Y. (2017). CNN Based Music Emotion Classification. *arXiv preprint arXiv:1704.05665*.

Loshchilov, I. & Hutter, F. (2019). Decoupled Weight Decay Regularization. *Preprint published January*, *4*.

Malheiro, R., Oliveira, H. G., Gomes, P. & Paiva, R. P. (2016). Keyword-based Approach for Lyrics Emotion Variation Detection. In *KDIR* (S. 33–44).

Malheiro, R., Panda, R., Gomes, P. & Paiva, R. P. (2016). Emotionally-relevant Features for Classification and Regression of Music Lyrics. *IEEE Transactions on Affective Computing*, *9* (2), 240–254.

Malik, M., Adavanne, S., Drossos, K., Virtanen, T., Ticha, D. & Jarina, R. (2017). Stacked Convolutional and Recurrent Neural Networks for Music Emotion Recognition. *arXiv preprint arXiv:1706.02292*.

Palanisamy, K., Singhania, D. & Yao, A. (2020). Rethinking CNN Models for Audio Classification. *arXiv preprint arXiv:2007.11154*.

Panda, R. (2019). *Emotion-based Analysis and Classification of Audio Music* (Unveröffentlichte Dissertation). 00500:: Universidade de Coimbra.

Panda, R., Malheiro, R. M. & Paiva, R. P. (2020). Audio Features for Music Emotion Recognition: A Survey. *IEEE Transactions on Affective Computing*.

Pannese, A., Rappaz, M.-A. & Grandjean, D. (2016). Metaphor and Music Emotion: Ancient Views and Future Directions. *Consciousness and Cognition*, *44*, 61–71.

Parisi, L., Francia, S., Olivastri, S. & Tavella, M. S. (2019). Exploiting Synchronized Lyrics and Vocal Features for Music Emotion Detection. *arXiv preprint arXiv:1901.04831*.

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N. & McAdams, S. (2011). The Timbre Toolbox: Extracting Audio Descriptors from Musical Signals. *The Journal of the Acoustical Society of America*, *130* (5), 2902–2916.

Plisson, J., Lavrac, N., Mladenic, D. et al. (2004). A Rule Based Approach to Word Lemmatization. In *Proceedings of IS* (Bd. 3, S. 83–86).

Pons, J. & Serra, X. (2019). MusiCNN: Pre-trained Convolutional Neural Networks for Music Audio Tagging. *arXiv preprint arXiv:1909.06654*.

Posner, J., Russell, J. A. & Peterson, B. S. (2005). The Circumplex Model of Affect: An Integrative Approach to Affective Neuroscience, Cognitive Development, and Psychopathology. *Development and Psychopathology*, *17* (3), 715.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... Iyengar, S. S. (2018). A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Computing Surveys (CSUR)*, *51* (5), 1–36.

Russell, J. A. (1980). A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, *39* (6), 1161.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J. & Müller, K.-R. (2021). Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proceedings of the IEEE*, *109* (3), 247–278.

Schäfer, T., Sedlmeier, P., Städtler, C. & Huron, D. (2013). The Psychological Functions of Music Listening. *Frontiers in Psychology*, *4*, 511.

Schedl, M., Gómez Gutiérrez, E. & Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, *8* (2–3), 127–261.

Schimmack, U. & Grob, A. (2000). Dimensional Models of Core Affect: A Quantitative Comparison by Means of Structural Equation Modeling. *European Journal of Personality*, *14* (4), 325–345.

Schuller, B., Dorfner, J. & Rigoll, G. (2010). Determination of Nonprototypical Valence and Arousal in Popular Music: Features and Performances. *EURASIP Journal on Audio, Speech, and Music Processing*, *2010*, 1–19.

Shahroudnejad, A. (2021). A Survey on Understanding, Visualizations, and Explanation of Deep Neural Networks. *arXiv preprint arXiv:2102.01792*.

Simonyan, K. & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

Sloboda, J. A. & Juslin, P. N. (2001). Psychological Perspectives on Music and Emotion. *Music and Emotion: Theory and Research*, 71–104.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. & Liu, C. (2018). A Survey on Deep Transfer Learning. In *International Conference on Artificial Neural Networks* (S. 270–279).

Torrey, L. & Shavlik, J. (2010). Transfer Learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (S. 242–264). IGI global.

Van Zaanen, M. & Kanters, P. (2010). Automatic Mood Classification Using TF*IDF Based on Lyrics. In *ISMIR* (S. 75–80).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems* (S. 5998–6008).

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C. & Xu, W. (2016). CNN-RNN: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 2285–2294).

Wang, Z., Dai, Z., Póczos, B. & Carbonell, J. (2019). Characterizing and Avoiding Negative Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (S. 11293–11302).

Warriner, A. B., Kuperman, V. & Brysbaert, M. (2013). Norms of Valence, Arousal, and Dominance for 13,915 English Lemmas. *Behavior Research Methods*, *45* (4), 1191–1207.

Weiss, K., Khoshgoftaar, T. M. & Wang, D. (2016, Mai). A Survey of Transfer Learning. *Journal of Big Data*, *3* (1), 9. Zugriff auf `https://doi.org/10.1186/s40537-016-0043-6` doi: 10.1186/s40537-016-0043-6

Wieczorkowska, A., Synak, P. & Raś, Z. W. (2006). Multi-label Classification of Emotions in Music. In *Intelligent Information Processing and Web Mining* (S. 307–315). Springer, Berlin, Heidelberg.

Yang, D. & Lee, W.-S. (2004). Disambiguating Music Emotion Using Software Agents. In *ISMIR* (Bd. 4, S. 218–223).

Yang, X., Dong, Y. & Li, J. (2018, Juli). Review of Data Features-based Music Emotion Recognition Methods. *Multimedia Systems*, *24* (4), 365–389. Zugriff auf `https://doi.org/10.1007/s00530-017-0559-4` doi: 10.1007/s00530-017-0559-4

Yang, Y.-H. & Chen, H. H. (2012). Machine Recognition of Music Emotion: A Review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8* (3), 1–30.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In H. Wallach, H. Larochelle & A. Beygelzimer (Hrsg.), *Advances in Neural Information Processing Systems* (Bd. 32). Curran Associates, Inc. Zugriff auf `https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf`

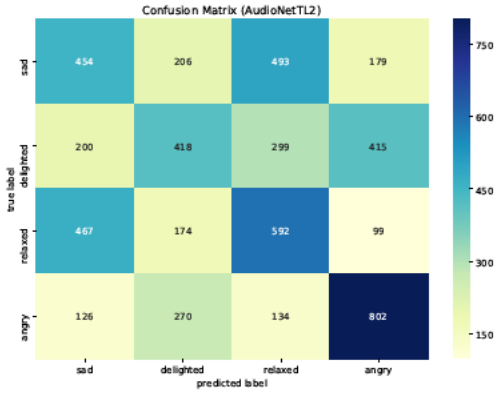
Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (S. 3320–3328). Cambridge, MA, USA: MIT Press.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109* (1), 43-76. doi: 10.1109/JPROC.2020.3004555

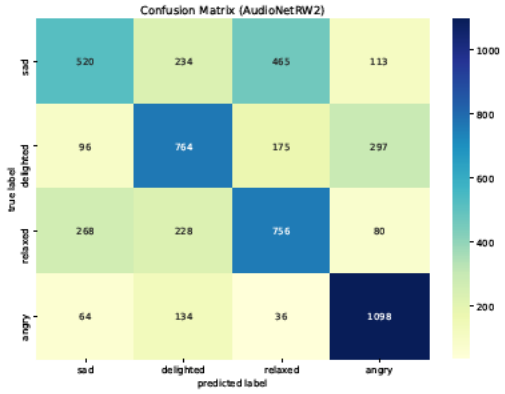
Anhang A

Wahrheitsmatrizen

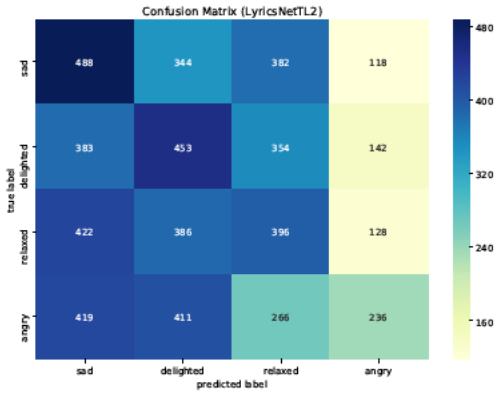
Die Wahrheitsmatrizen der Modelle, welche für $i = \{2, 4, 8\}$ Epochen trainiert wurden, finden sich in den Abbildungen A.1, A.2 und A.3.



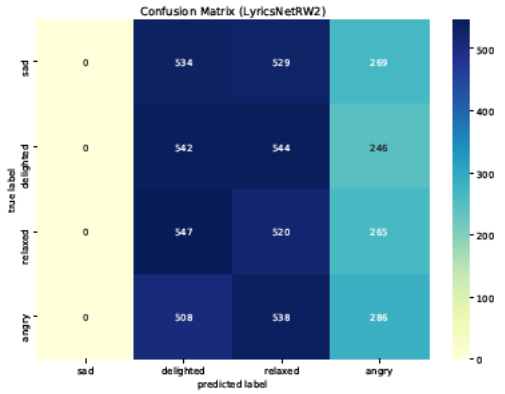
(a) AudioNetTL2.



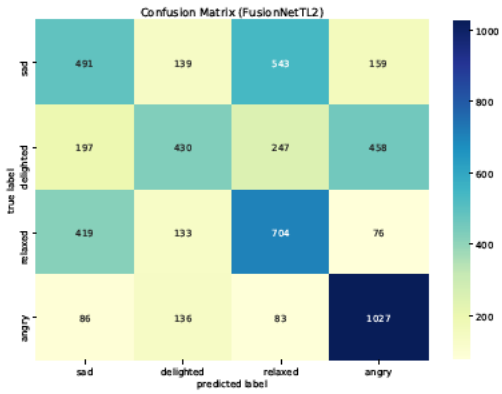
(b) AudioNetRW2.



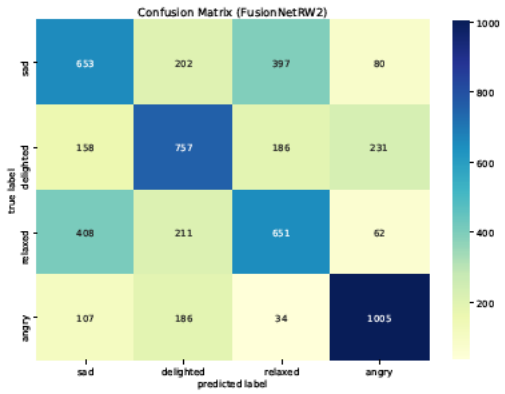
(c) LyricsNetTL2.



(d) LyricsNetRW2.

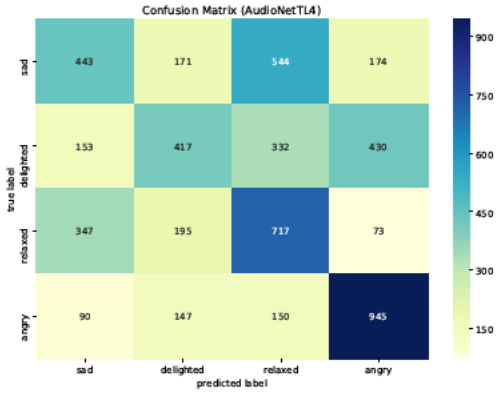


(e) FusionNetTL2.

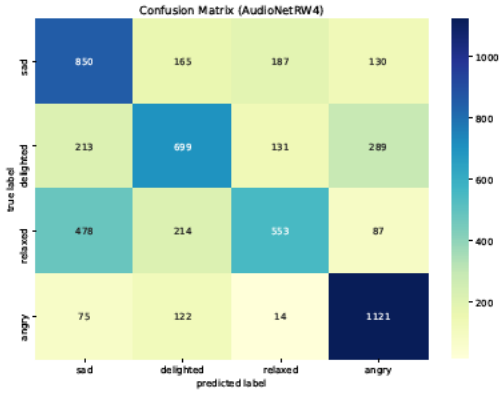


(f) FusionNetRW2.

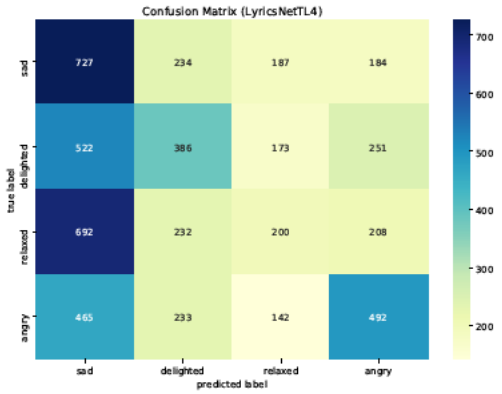
Abbildung A.1: Wahrheitsmatritzen aller Modelle, die für 2 Epochen trainiert wurden.



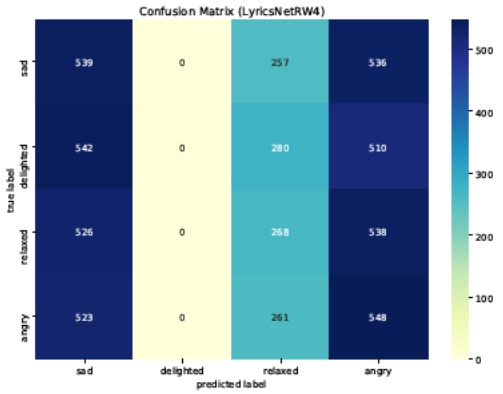
(a) AudioNetTL4.



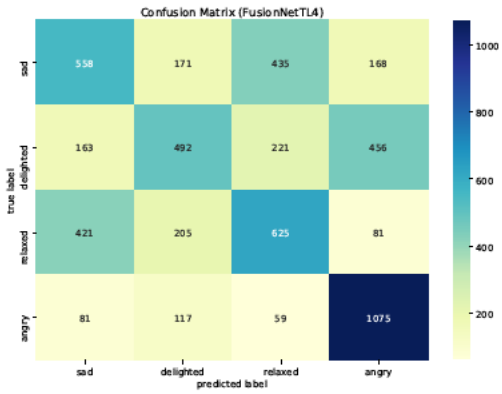
(b) AudioNetRW4.



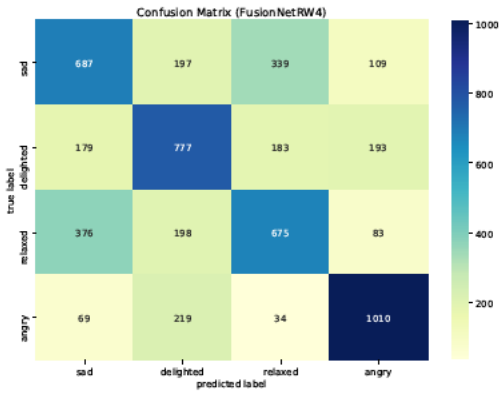
(c) LyricsNetTL4.



(d) LyricsNetRW4.

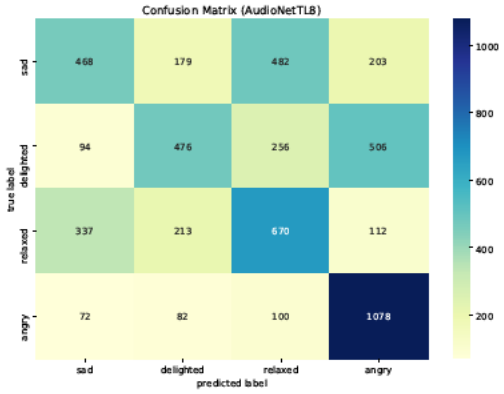


(e) FusionNetTL4.

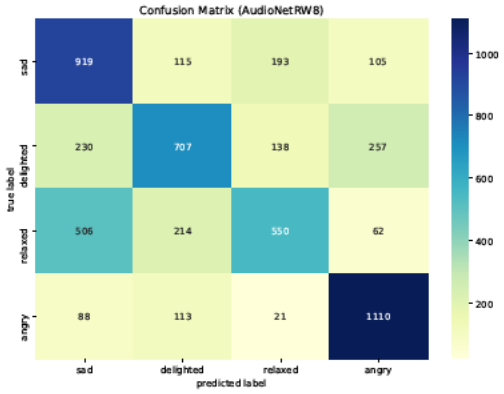


(f) FusionNetRW4.

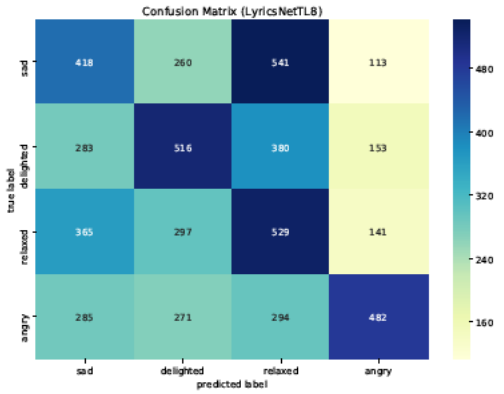
Abbildung A.2: Wahrheitsmatritzen aller Modelle, die für 4 Epochen trainiert wurden.



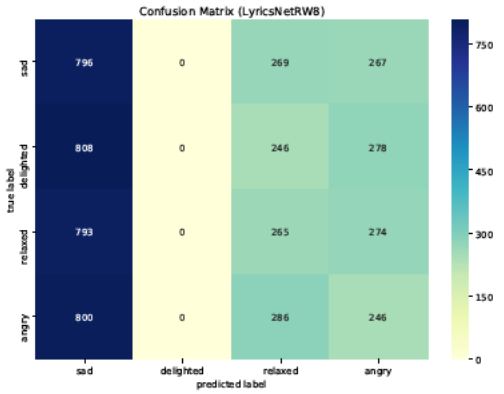
(a) AudioNetTL8.



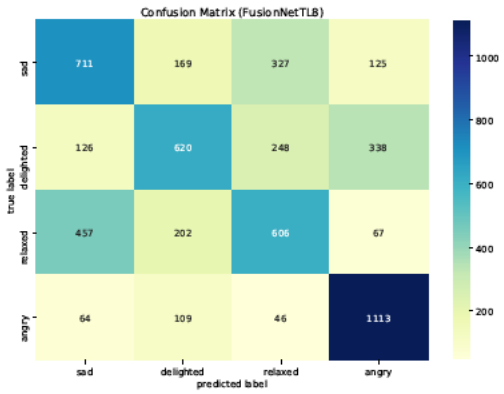
(b) AudioNetRW8.



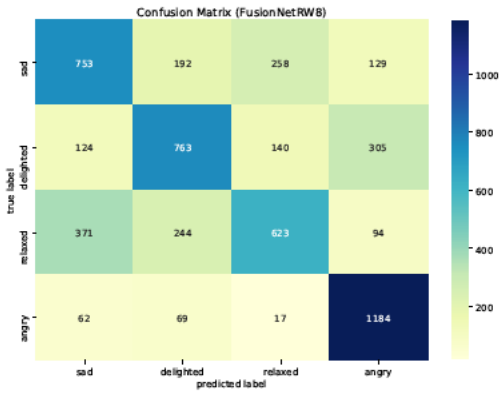
(c) LyricsNetTL8.



(d) LyricsNetRW8.



(e) FusionNetTL8.



(f) FusionNetRW8.

Abbildung A.3: Wahrheitsmatritzen aller Modelle, die für 8 Epochen trainiert wurden.

Anhang B

Absolute Trainingsdauern

Tabelle B.1 listet die absoluten Trainingsdauern in Minuten für die untersuchten Modelle auf. Weiterhin zeigt Tabelle B.2 die jeweils verwendete GPU.

Modelle		**2 Epochen** t_2 [min]	**4 Epochen** t_4 [min]	**6 Epochen** t_6 [min]	**8 Epochen** t_8 [min]
AudioNet	TL	10,10	13,05	18,68	37,42
	RW	34,50	45,05	67,49	133,20
LyricsNet	TL	11,77	13,07	31,85	41,74
	RW	105,34	114,95	306,03	408,35
FusionNet	TL	17,16	36,46	46,93	62,41
	RW	90,51	261,34	265,97	360,94

Tabelle B.1: Absolute Trainingsdauern (drittes Bewertungskriterium).

Modelle		**2 Epochen**	**4 Epochen**	**6 Epochen**	**8 Epochen**
AudioNet	TL	P100	T4	T4	P100
	RW	P100	T4	T4	P100
LyricsNet	TL	T4	P100	T4	T4
	RW	T4	P100	T4	T4
FusionNet	TL	P100	T4	P100	P100
	RW	P100	T4	P100	P100

Tabelle B.2: Verwendete GPUs der Nvidia Tesla-Reihe.