CODEC-DETEKTION IN AUDIODATEN MIT CNNS

[redacted]

Course – Deep Learning for Audio Event Detection
Technische Universität Berlin
Fakultät I - Geistes- und Bildungswissenschaften
Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation
Sek. EN-8 - Einsteinufer 17c - 10587 Berlin

KURZFASSUNG

Im folgenden Paper wird ein Deep Learning Ansatz zur Erkennung verlustbehafteter Codecs nach Konvertierung in das verlustfreie Dateiformat PCM vorgestellt. Nach Design und Optimierung eines Convolutional Neural Networks (CNN) können sehr hohe Treffergenauigkeiten (categorical accuracy) bei der Zuordnung zu den entsprechenden Codecs erreicht werden. Die Erkennung zeigt sich robust gegenüber variierenden Audioinhalten, und das vorgestellte CNN erzielt bessere Ergebnisse als ein vergleichbares CNN aus der Literatur.

Index Terms— Codec Detection, CNN, Audio Event Detection

1. EINLEITUNG

Immer mehr Streamingdienste und Online-Musik-Services bieten ihren Nutzer:innen unkomprimierte Audiodaten zu höheren Preisen an. Für die Qualitätssicherung der Dienste, als auch für die Endkonsument:innen ist die Sicherstellung der verlustfreien Qualität von hoher Bedeutung. Durch eine Konvertierung verlustbehafteter Audiodaten in verlustfreie Dateiformate kann eine hohe Qualität der Audiodaten suggeriert werden, welche in Wirklichkeit nicht vorliegt. Ziel der vorliegenden Untersuchung ist es, mit einem CNN verschiedene Codecs in verschiedenen Qualitätsstufen in scheinbar verlustfreien wav-Dateien zu erkennen und diese voneinander zu unterscheiden. [1] schlägt ein CNN vor, mit dem die Einflüsse verschiedener Codecs in Audiodateien mit einer Treffergenauigkeit von 98.6 % erkannt werden, jedoch nur binär zwischen verlustfrei und verlustbehaftet unterschieden wird. In [2] wird ein Ansatz ohne Verwendung eines neuronalen Netzes zur Erkennung verschiedener Codecs etabliert und erreicht dabei eine kategoriale Genauigkeit von 96 %.
Die Kompression von Audiodaten basiert auf psychoakustischen Verdeckungseffekten und einer angepassten Quantisierung des Signals, durch welche das Quantisierungsrauschen möglichst gut unhörbar gemacht wird. Durch die Wahl geeigneter Filter werden für den Menschen nicht hörbare Signalanteile aus den Daten entfernt und so die Menge an Daten reduziert. Durch die beschriebene verlustbehaftete Kompression kommt es jedoch immer zu einer Veränderung des Signals und letztlich einer Minderung der Qualität. Eine Konvertierung in ein verlustfreies Format ist dennoch möglich, jedoch kann das Signal nicht mehr vollständig rekonstruiert werden. [3] Im folgenden wird ein Modell etabliert, was Artefakte dieser verlustbehafteten Codecs innerhalb von wav-Dateien identifiziert. In Kapitel 2 werden die Generierung des Datensatzs, die Vorverarbeitung der Daten, das Netz und dessen Optimierung vorgestellt. In Kapitel 3 werden die Ergebnisse der Untersuchung mit dem optimierten Netz vorgestellt. Die Robustheit des Modells wird mit einem zweiten, generierten Datensatz überprüft. Das Modell aus [1] wird ebenfalls zum Vergleich implementiert und angewandt und die Ergebnisse dargestellt. Die Ergebnisse werden in Kapitel 4 diskutiert und in Kapitel 5 eine Zusammenfassung und ein Ausblick der Arbeit geboten.

2. METHODEN

2.1. Datensätze

Für Design und Training des CNNs wurde ein Ausschnitt des Datensatzes aus [4] verwendet. Der Datensatz besteht aus verlustfreien .wav-Dateien mit klassischen bzw. orchestralen Audioinhalten. Der verwendete Ausschnitt beinhaltet insgesamt 35 Stücke mit einer gesamten Abspieldauer von drei Stunden und 24 Minuten und einer Gesamtgröße von etwa 3 GB.
Zur Überprüfung der Robustheit des CNNs gegenüber unterschiedlichen Audioinhalten wurde ein zweiter Datensatz verwendet, der durch Digitalisierung von Schallplatten er-

stellt wurde. Die Musik der Schallplatten sind vorwiegend dem Genre Pop zuzuordnen. Es wurden insgesamt ebenfalls 35 Stücke digitalisiert mit einer Gesamtgröße von 2.8 GB.

Die Musikdaten wurden in einsekündige Abschnitte(Chunks) zerteilt und anschließend als Spektrogramme dem CNN übergeben.

In Abb. 1 sind beispielhaft die Spektrogramme eines Chunks mit linearer Frequenzskalierung dargestellt. Ein Unterscheidungsmerkmal, bereits mit bloßem Auge erkennbar, ist das Abschneiden der Frequenzanteile oberhalb einer gewissen Grenzfrequenz, welche vom Codec und der verwendeten Bitrate bzw. Qualität abhängig ist.

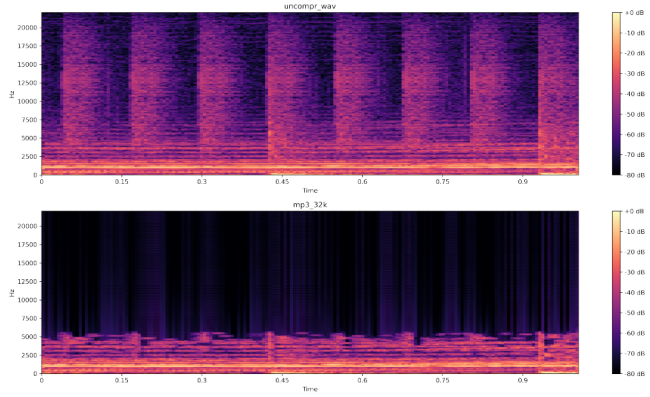

Abb. 1. Linear skalierte Spektrogramme eines verlustfreien Chunks (oben) und eines Chunks, das aus einer mp3 mit 32 kBit decodiert wurde

Der genaue Ablauf der Datenvorverarbeitung wird im nächsten Unterabschnitt dargestellt.

2.2. Datenvorverarbeitung

Für die Konvertierung mit verlustbehafteten Codecs und die Dekodierung in das verlustfreie PCM Format wird das Projekt FFMPEG genutzt, welches aus verschiedenen frei zugänglichen Computerprogrammen und Programmbibliotheken besteht. Innerhalb dieser Arbeit werden die oft genutzten Dateiformate MP3 (LAME MP3 Encoder), AAC (Advanced Audio Coding) und OGG (libvorbis) verwendet. Qualitätsstufen bzw. Bitraten und Anzahl der Samples in den Train- und Testsets sind in Tabelle 1 dargestellt.

Die Vorverarbeitung der Daten findet bei beiden verwendeten Quellen an Audiodaten nach den folgenden Schritten statt:

1. Die verlustfreien Dateien werden mit FFMPEG in die entsprechenden Dateiformate encodiert.

2. Die Dateien werden nach Kompression zurück in das PCM-Format konvertiert.

Codec	**Bitrate/Qualität**	**Elemente (train/test)**
PCM	16-bit Little-Endian Signed Integer	1400/350
Mp3	32 kBit	1400/350
Mp3	160 kBit	1400/350
Mp3	192 kBit	1400/350
Mp3	320 kBit	1400/350
AAC	128 kBit	1400/350
Vorbis	Quality 4 $\sim$ 128 kBit	1400/350

Tab. 1. Verwendete Codecs und zugehörige Bitraten /Qualitäten sowie Anzahl an Samples in Test- und Traningsdatensatz

3. Die Dateien werden in einsekündige Chunks zerschnitten, um die Datenmenge innerhalb einer Berechnungsepoche zu reduzieren.

4. Chunks mit einem Effektivwert kleiner 50% des Effektivwertes der ungeschnitten Datei werden aussortiert und nicht berücksichtigt.

5. Audiodaten werden mit einer Kurzzeit-Fourier-Transformation (STFT) in den Frequenzbereich transformiert. (Fensterlänge = 1024 Samples, Hop-Legth = 512 Samples, Hann-Fenster)

6. Zur weiteren Datenreduktion und da bei Betrachtung der Spektrogramme besonders im hohen Frequenzbereich Artefakte identifiziert werden konnten, werden die erhaltenen Spektrogramme bei 4 kHz abgeschnitten und tiefere Frequenzen nicht weiter berücksichtigt.

7. Zur Annäherung an das Hörverhalten des Menschen werden die Amplituden der Spektrogramme in Pegel überführt.

8. Die Spektrogramme werden z-standardisiert.

9. Abschließend werden randomisiert in gleicher Anzahl pro Codec Chunks gewählt, um einen ausgeglichenen Train- bzw. Testdatensatz zu erhalten.

2.3. Entwicklung und Optimierung des CNNs

Ausgangspunkt der Untersuchungen bildet das Modell aus [1], welches ein klassisches Convolutional Neural Network bestehend aus vier *convolutional layers* und drei *fully connected layers* ist. Aufbauend auf diesem Modell werden mit *Hyperparameter Tuning* die *pooling* Größe, Filterkerngröße, die Anzahl der Filter und *Drop-out* Raten in den in Tabelle 2 aufgeführten Wertebereichen optimiert. Zum *Tuning* wird der *Hyperband* Algorithmus verwendet. Es werden der Adam-Optimizer und die Categorical-Cross-Entropy-Funktion verwendet.

Parameter	min	max	Schrittweite
kernel size (f-Achse)	2	16	2
kernel size (t-Achse)	2	8	2
pooling size (f-Achse)	1	2	1
pooling size (t-Achse)	1	2	1
filter 1	16	64	8
filter 2	16	64	8
filter 3	16	64	8
filter 4	16	64	8
drop-out 1	0.1	0.5	0.1
drop-out 2	0.1	0.5	0.1

Tab. 2. *Hyperband Search* Parameter des Netzes, Wertebereiche und Schrittweiten

3. ERGEBNISSE

3.1. Hypertuning

Mit Hilfe der *Hyperband Search* wird aus über 30 verschiedenen Modelle ein optimierter Parametersatz ermittelt. Dieser erreicht eine kategoriale Treffergenauigkeit von 98% auf dem Validierungsdatensatz. Das für das Training verwendete und mit der *Hyperband Search* ermittelte Netzwerk ist in Abbildung 2 im Anhang zu finden.

3.2. Training- und Testergebnisse

In Tabelle 3 sind die Metriken des Testdatensatzes dargestellt. Mit einer durchschnittlichen Treffergenauigkeit von über 97% zeigt sich eine gute Erkennung durch das optimierte CNN. Beim Training des Netzwerks werden bereits nach der ersten Epoche Treffergenauigkeiten von 98% auf dem Validierungsdatensatz erreicht, was für einen sehr schnellen Trainingsvorgang des CNNs spricht. In den folgenden Epochen werden nur noch kleine Verbesserungen erreicht, und das Training bricht nach Epoche 21 durch einen Early Stopping Callback ab. Die wenigen auftretenden Verwechslungen finden sich hauptsächlich zwischen den mp3-Codecs vergleichbarer Qualität der Bitraten 192 kBit und 160 kBit, was sich auch in den entsprechenden Werten von Recall und F1-Score widerspiegelt.

3.3. Ergebnisse mit alternativem Datensatz

Zur Überprüfung der Robustheit der Erkennung wurde eine Prädiktion auf einem zweiten Datensatz ausgeführt. Da für das Training ausschließlich Teile des ursprünglichen Datensatzes verwendet wurden, konnte für die Prädiktion der gesamte zusätzliche Datensatz verwendet werden, welcher aus insgesamt 12250 Spektrogrammen besteht. Die Ergebnisse der Prädiktion sind in Tabelle 4 aufgeführt. Auch hier wird eine hohe durchschnittliche Treffergenauigkeit von etwa 97% erreicht. Der Großteil der Verwechslungen findet hier zwischen den mp3-Codecs mit Bitraten von 192 kBit und 320 kBit statt, sowie zwischen dem vorbis-Codec und dem mp3-Codec der Bitrate 160 kBit.

Codecs	Precision	Recall	F1-Score
Mp3 32 kBit	1.00	1.00	1.00
Mp3 160 kBit	0.97	0.88	0.92
Mp3 192 kBit	0.89	0.97	0.93
Mp3 320 kBit	0.99	1.00	0.99
AAC 128 kBit	0.99	0.99	0.99
Vorbis	0.96	0.97	0.97
PCM	1.00	0.98	0.99
Treffergenauigkeit			**97%**

Tab. 3. Metriken der Modellevaluation mit dem Testdatensatz.

Codecs	Precision	Recall	F1-Score
Mp3 32 kBit	1.00	1.00	1.00
Mp3 160 kBit	0.97	0.84	0.90
Mp3 192 kBit	0.85	0.97	0.90
Mp3 320 kBit	0.97	1.00	0.99
AAC 128 kBit	1.00	1.00	1.00
Vorbis	1.00	0.96	0.98
PCM	1.00	1.00	1.00
Treffergenauigkeit			**97%**

Tab. 4. Metriken der Prädiktion mit dem zweiten Datensatz.

3.4. Vergleich mit alternativem Modell

Zum Vergleich der Performance des vorgeschlagenen CNNs, wurde das CNN aus [1] implementiert und ebenfalls mit dem ursprünglichen Datensatz trainiert und eine Prädiktion auf dem Vinyldatensatz ausgeführt. Die Ergebnisse sind in Tabelle 5 dargestellt. Mit einer durchschnittlichen Treffergenauigkeit von etwa 73% der Prädiktion ist die Performance des nicht optimierten CNNs deutlich schlechter. Auch der Trainingsprozess ist wesentlich langsamer: Die Treffergenauigkeit auf dem Validierungsdatensatz ist nach Epoche 1 bei etwa 63% und nach 30 Epochen bei 97%, wobei auch nach 30 Epochen noch keine Konvergenz erreicht wird. Trotz guter Trainingsperformance zeigt das nicht optimierte CNN also wesentlich schlechtere Treffergenauigkeiten auf dem zweiten Datensatz. Dabei finden, wie beim optimierten CNN, die meisten Verwechslungen bei den mp3-Codecs der Bitraten 160 kBit, 192 kBit und 320 kBit und dem Vorbis-Codec statt, allerdings in viel stärkerem Ausmaß.

Codecs	Precision	Recall	F1-Score
Mp3 32 kBit	1.00	1.00	1.00
Mp3 160 kBit	0.84	0.16	0.27
Mp3 192 kBit	0.11	0.09	0.10
Mp3 320 kBit	0.48	0.99	0.65
AAC 128 kBit	0.99	0.97	0.98
Vorbis	0.99	0.90	0.94
PCM	1.00	0.99	1.00
Treffergenauigkeit		**73%**	

Tab. 5. Metriken der Prädiktion mit dem nicht optimierten CNN

4. DISKUSSION

Das durch den *Hyperband Search* optimierte CNN liefert sowohl auf dem Testdatensatz mit Audioinhalten des Datensatzes aus [4], als auch auf dem alternativen Datensatz aus digitalisierten Schallplatten sehr gute Erkennungsraten. Treffergenauigkeiten von über 97% bestätigen das Abschneiden der Frequenzen unterhalb von 4 kHz als geeignete Maßnahme zur Datenreduktion. Zusätzlich zum wesentlich schnelleren Training zeigte sich das optimierte CNN, im Gegensatz zum verglichenen, nicht optimierten Modell aus [1], als robust gegenüber den unterschiedlichen Audioinhalten des zweiten Datensatzes.

Das codec- und bitratenabhängige Abschneiden hoher Frequenzanteile (vgl. Abb. 1) der verlustbehafteten Codecs, der sehr schnelle Trainingsvorgang, so wie die hohen Treffergenauigkeiten ohne Betrachtung von Frequenzen unterhalb von 4 kHz lassen die Vermutung zu, dass das vorgeschlagene CNN stark auf die Frequenzbereiche jenseits der jeweiligen Grenzfrequenzen der Codecs fokussiert sein könnte. Dies würde nahelegen, dass die Codecerkennung bei schmalbandigen Audioinhalten, bei denen also bereits die ursprüngliche, verlustfreie Datei wenig oder keine Frequenzen jenseits von z.B. 18 kHz enthält, stark eingeschränkt werden könnte. Da die meisten Musikformen darauf ausgelegt sind, den ganzen hörbaren Frequenzbereich zu bedienen und zumindest im professionellen Bereich entsprechend gemixt und gemastert wird, stellt dies zwar eine wichtige, aber keine kritische Limitation dar. Für andere Audioinhalte, wie z.B. Sprache (Podcasts, Hörbücher etc.), kann im Rahmen dieser Untersuchungen keine Aussage getroffen werden.

Weitere Limitationen der vorgestellten Erkenntnisse ergeben sich durch die verwendete Abtastfrequenz und den ausgewählten Codecs. Es wurden lediglich Audiodateien hoher Qualität mit einer Abtastfrequenz von 44.1 kHz verwendet, für andere Formate der Quelldateien können keine Aussagen gemacht werden. Die Anzahl der verwendeten Codecs war zudem sehr beschränkt. Es existiert eine Vielzahl verschiedener Codecs und entsprechender Encoder mit unterschiedlichsten Eigenschaften, von denen nur wenige ausgewählt werden konnten.

Es ist zudem denkbar, dass Audiodateien nicht nur einen „Codierzyklus“ durchlaufen, sondern zwischen verschiedenen Codecs transkodiert wird. Über Erkennungserfolge in diesem Fall kann durch die vorgestellten Ergebnisse ebenfalls keine Aussage gemacht werden.

5. ZUSAMMENFASSUNG UND AUSBLICK

Mit Audiodateien aus [4] und eigens digitalisierten Schallplatten wurden Datensätze aus einsekündigen Spektrogrammen mit verschiedenen Verlustartefakten erstellt. Mit dem ersten Datensatz wurde ein CNN trainiert und mittels *Hyperparameter Tuning* für die Erkennung und Zuordnung der verwendeten Codecs optimiert. Das vorgestellte CNN zeigt Treffergenauigkeiten von über 97% auf beiden Datensätzen und erzielt deutlich bessere Ergebnisse als das verglichene CNN aus [1].

Um die Robustheit gegenüber schmalbandigeren Audioinhalten zu überprüfen, sollten die Datensätze durch weitere Inhalte erweitert werden. Denkbar ist hier sowohl die künstliche Erstellung von bandbegrenzten .wav-Dateien, als auch die Verwendung von natürlicherweise schmalbandigeren Inhalten wie z.B. gesprochener Sprache, etwa aus Podcasts oder Hörbüchern. Zudem sollten weitere Codecs als Klassen mit in die Datensätze aufgenommen werden, um die Artefakte von möglichst vielen Codecs zu untersuchen und erkennbar zu machen. Auch Audiodateien, die mehrfache Codierzyklen durchlaufen haben und daher Artefakte verschiedener Codecs enthalten, könnten mit in die Datensätze aufgenommen werden, wodurch sich jedoch die Problemstellung auf ein Multiclass-Problem erweitern würde.

Zusätzlich zu Kompressionsartefakten sollte der Einfluss durch Artefakte anderer Ursprünge auf die Treffergenauigkeit untersucht werden. Breitbandiges Rauschen könnte gewisse Kompressionsartefakte maskieren und somit den Einfluss ursprünglicher Kompressionen unkenntlich machen. Weitere denkbare Artefakte wären Packet Loss z.B. durch verlustbehaftete, drahtlose Broadcastprotokolle oder z.B. durch Übersteuerung auftretende Verzerrungen. Möglicherweise könnte auch eine Codecerkennung unter schwierigeren Bedingungen, z.B. in einer Bar oder in einem Club, möglich sein, wodurch das Audiosignal zusätzlich durch Lautsprecher- und Raumimpulsantworten und auftretende Umgebungsgeräusche beeinflusst werden würde. Die Erstellung eines geeigneten Datensatzes wäre hierfür jedoch sehr aufwändig.

6. REFERENZEN

[1] R. Hennequin, J. Royo-Letelier, and M. Moussallam, “Codec independent lossy audio compression detection,” in *2017 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), pp. 726–730, 2017.

[2] Bongjun Kim and Zafar Rafii, *EUSIPCO 2018: 2018 26th European Signal Processing Conference, Rome, Italy, September 3-7, 2018*. Piscataway, NJ: IEEE, 2018.

[3] K. Petermichl, "Dateiformate für Audio," in *Handbuch der Audiotechnik* (S. Weinzierl, ed.), pp. 687–718, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[4] R. M. Bittner, J. Wilkins, H. Yip, and J. P. Bello, *MedleyDB 2.0 : New Data and a System for Sustainable Data Collection*. New York, NY, USA: International Conference on Music Information Retrieval (ISMIR-16), 2016.

7. ANHANG

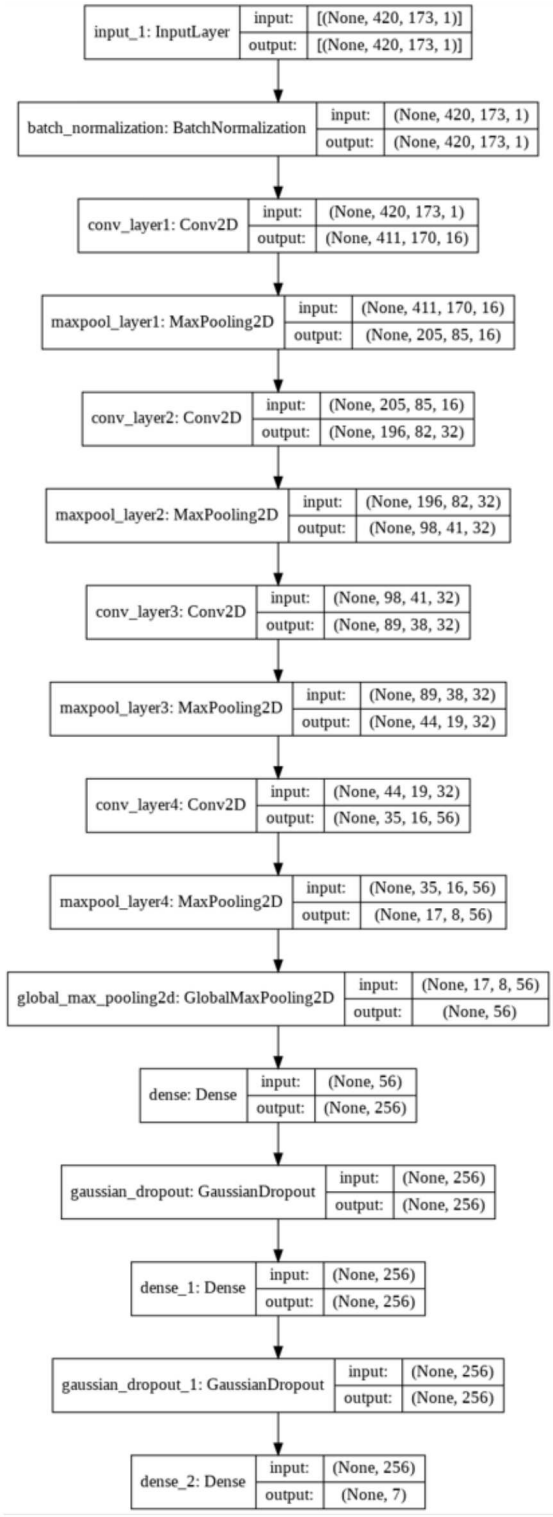

Abb. 2. Per Hyperband Tuning optimierte und für das Training verwendete Netzwerkarchitektur