KI (CNN) gesteuerte Gefahrengeräuscherkennung für Gehörlose / -geschädigte Menschen

Ein AUDAS Projekt

von

Nick Jonas Kuhoff und Christian Böndgen

Inhaltsverzeichnis

1. Frage und Hypothese

- Ziel :
  - Anlernen eines Neuronalen Netzwerks für die Erkennung von Gefahren- und Signaltönen.

- Forschungsfragen:
  - Wie kann ein Neuronales Netzwerk auditive Indikatoren für Gefahren erkennen?
  - Welche Geräuschindikatoren für welche Gefahren gibt es?
  - Welche Techniken sind für den Aufbau des Neuronalen Netzwerks am effektivsten?
  - Welche Audiomerkmale werden als Inputdaten benötigt?
  - Gibt es weitere Kategorien, außer der von dem Geräuschquelle ausgehenden Gefahr, die eine wichtige Rolle spielen könnten, um Geräusche zu identifizieren und bewerten zu können?

- Hypothese :
  - Es ist möglich, für ein Neuronales Netzwerk mithilfe von Audiodaten, Gefahren zu erkennen und die Salienz und Wichtigkeit von Geräuschen zu bestimmen.

2. Stand der Forschung

1. Vereinzelte Systeme auch schon mit Smartphone Anwendung
   - Alarm Sound Classification System in Smartphones for the Deaf and Hard-of-Hearing Using Deep Neural Networks [Shi+20]
     - DNN mit 25k Samples, 1000 epochen
     - 5 Geräuscharten: horn, bicycle, bell, ambulance, fire alarm und noise
2. Bis jetzt noch keine Anwendung im Alltagsbereich
3. Derzeitige Techniken der Audiomerkmalsextraktion
4. Aktuelle Modelle

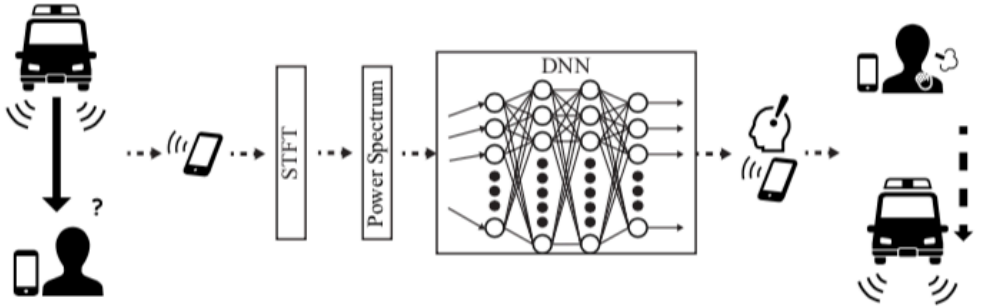


Figure 1. Alarm sound classification and transmission systems. [Shi+20]

2.3.1 Derzeitige Techniken der Audiomerkmalsextraktion

- Allgemein
  - Rate der Nulldurchläufe
  - Diverse Verhältnisse (z. B. Verhältnis von stillen zu lauten Segmenten)
- Zeitlich
  - Dynamik und Leistung über die Zeit
  - Onsets => BPM

2.3.2 Derzeitige Techniken der Audiomerkmalsextraktion

- Spektral
  - Spektrogramme
  - Chromagramme
  - Waveletts
  - Mel-Frequent Cepstrum
  - Spektraler Mittelpunkt, Spektrale Varianz, Spektrale Schwanungen

2.4 Aktuelle Modelle

- Yaganoglu und Köse [YaKö18]:
  - Klassifizierungsnetzwerk (8 Klassen)
  - Inputs:
    - Audio Fingerprints
    - Diverse Spektrale und Zeitliche Eigenschaften
- Veena und Aravindhar [SuAr22]:
  - Klassifizierungsnetzwerk (10 Klassen)
  - Inputs:
    - MFCC
    - Mel-Spektrogramm
    - Chroma-STFT
    - Spectraler Kontrast
    - Tonnetz

3.1 Methode

Vorarbeit und Arbeit mit Audio:

1. Grundkonzept überlegen
2. Audiodaten sammeln und aufnehmen
3. Metadaten hinzufügen
4. Audiodaten erweitern
5. Audiodaten bewerten
6. Audiodaten bewerten – Outputs der KI
7. Spektrogramme berechnen

3.1.1 Das Grundkonzept

- **Gefahrengeräusche:** Martinshorn, Sirene, Hupe (Auto, Motorrad, LWK, Bahn), Feueralarm, Alarm, Fahrradklingel, Schreie (Hilferufe)

- **Literatur:** Verschiedene Paper über Geräuscherkennung für Gehörlose Menschen und Klassifizierung von deep neural networks

- **Python libraries:** librosa, tensorflow (xKeras), numpy, matplotlib, audiomentation, kapra

3.1.2 Audiodaten sammeln und aufnehmen

- Samples aus **Audiokits**: ESC50, Urbansoundkit, Youtube
- **Eigene Aufnahme** mit Field-Recordern: Zoom H6n/H2n und Tascam
- Abtastung in 44.1kHz und 16bit

- **Audiobeispiel** aus eigener Aufnahme:
  Feuerwehr-Horn ->

3.1.3 Metadaten hinzufügen

- CSV-Datei mit Werten für jedes einzelne Sample:

Name (String),
Samplerate (Int in hz),
Qualität(float 0-9),
Gefahr (Float 0-9),
Wichtigkeit (Float 0-9)

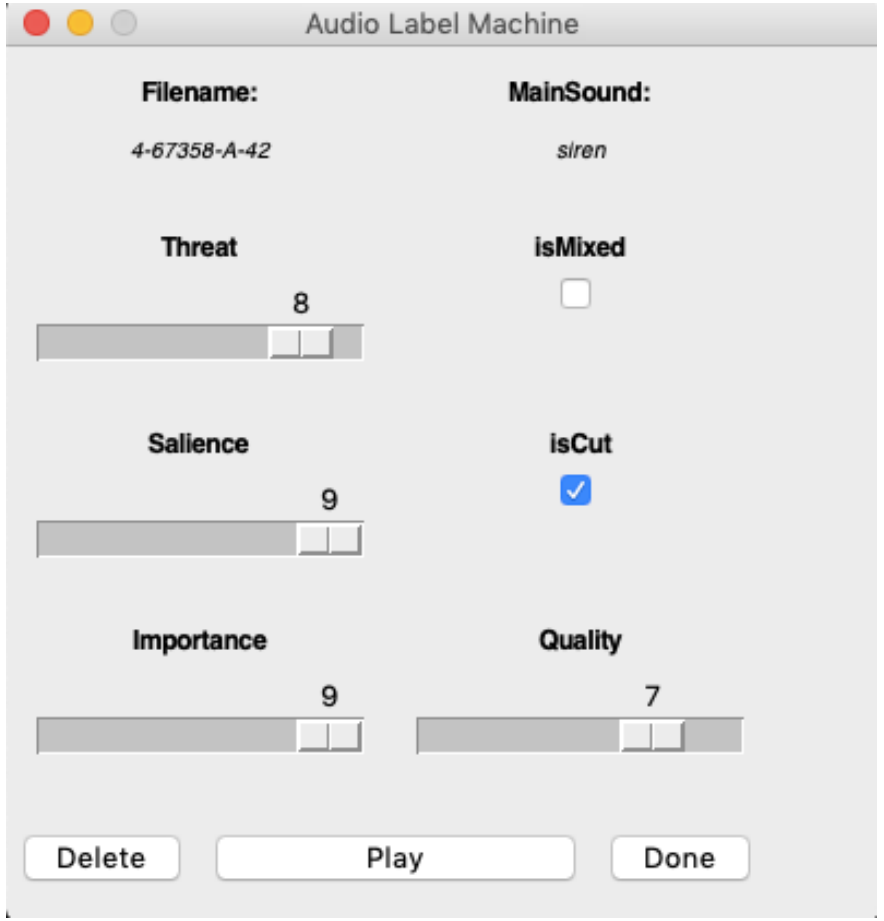
Länge (Float in s),
Ausschnitt (Bool 1/0),
Gemischt (Bool 1/0),
Salienz (Float 0-9),

3.1.4 Audiodaten erweitern

- **Audio Data Augmentation** mit: Audiomentations (open source)
- **Waveform-transform-chain** mit:
  PitchShift, AirAbsorption, AddGaussianNoise, HighPassFilter, ClippingDistortion, LowPassFilter
- Jeweils 15 durchläufe pro sample
- Raw audio augmentation

3.1.5 Audiodaten bewerten

- Audio Label Machine
- Haupt-**Bewertungskriterien**: Threat, Salience und Importance
- Weitere: isMixed, isCut und Quality
- Delete-Button nützlich für unbrauchbare Samples(z.B. zu viel Noise zu niedrige Qualität oder andere Sound als beschrieben bzw. vermischt)



3.1.6 Audiodaten bewerten - Outputs der KI

- **Threat** (0-9):

  0: Keine Gefahr
  1-2: So gut wie keine Gefahrenquelle
  3-4: Könnte gefährlich sein
  5: Geringe Gefahr
  6-7: Gefährlich
  8: Sehr gefährlich
  9: Lebensgefährlich

- **Salience** (0-9):

  0: Unbemerkbar/Kaum wahrnehmbar
  1-2: Genaues hinhören erforderlich
  3-4: Wenn man darauf achtet, hört mans
  5: Gut unterscheidbar vom Umfeld
  6-7: Bemerkbare Salienz
  8: Auffälliges Geräusch
  9: Direkte Aufmerksamkeit

- **Importance** (0-9):

  0: Total unwichtig
  1-2: Unwichtig
  3-4: Könnte wichtig sein
  5: Je nach Kontext wichtig
  6-7: Meistens wichtig
  8: Wichtig, relevant zu wissen
  9: Sehr wichtig, definitiv Bescheidgeben!

3.1.7 Spektrogramme berechnen

- Spektrogramm Art und Größe:
  - Log-Stft (später Mel mit 64 bins)
  - Mono-Channel
  - sampleRate = 16kHz
  - audioFileLength = 2s / Spektrogramm (später insgesamt 3s)
  - spec_hopSize = 256 samples
  - spec_blockSize = 2048 bins (später 1024 bins)
- Zuerst:
  - Librosa per funktion: stft
  - Vorberechnet und als Numpy Arrays abgespeichert
- Später:
  - Per Kapre als Schicht integriert im Neuronalen Netzwerk

3.2 Methode

Hauptaufgaben:

1. Audiodaten gruppieren
2. Modell des Neuronales Netzwerks erstellen
3. Das Neuronal Netzwerk trainieren
4. Auswerten und vergleichen (zurück zu schritt 3.2.2 bis glücklich)

3.2.1 Audiodaten gruppieren

- Audiodaten durchmischen
- Audiodaten in Gruppen aufteilen, um kleinere Pakete an das Netzwerk weiterzugeben

- Übrige Audiodaten als Testdaten beiseitelegen

3.2.2 Modell des Neuronales Netzwerks erstellen

Unser Modell:

- Anfangs komplett selbst erstellt
- Meistens 2-3 Faltungsschichten und 2-4 Verdichtungsschichten

- Später mithilfe des VGG16 Netzwerks deutlich verbessert
- Nachbau und Anpassung des VGG16 Netzwerks bisher erfolgslos

3.2.2 Finaler Modellaufbau

1. Kapre Schicht, die aus einem 3s Audiostream ein Mel-Spektrogramm berechnet
2. Angepasstes VGG16-Netzwerk mit 30 output Klassen
3. Skalierung der 30 Klassen auf 3 Klassen mit einer Bewertung von 0-9 und Softmax-Aktivierungsfunktion

3.3.1 Das VGG16 Netzwerk

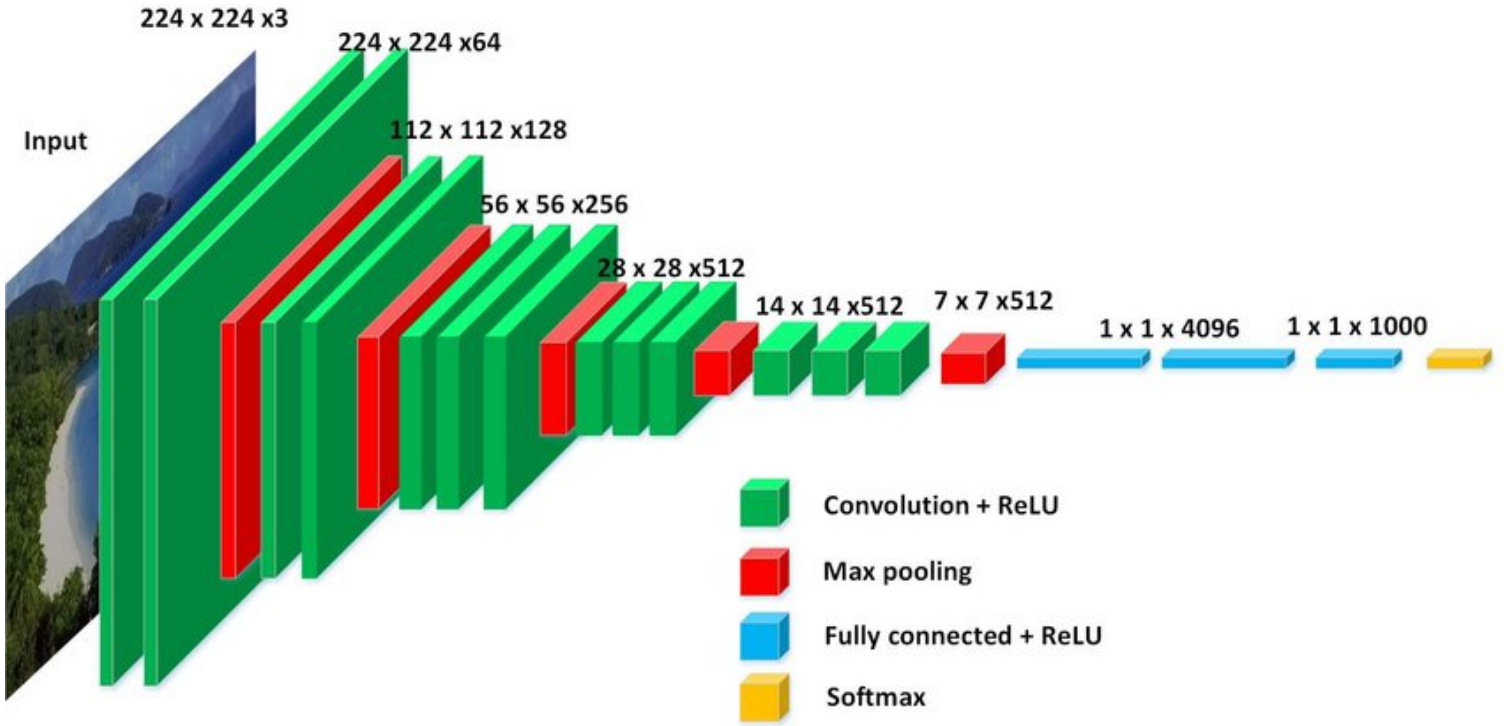
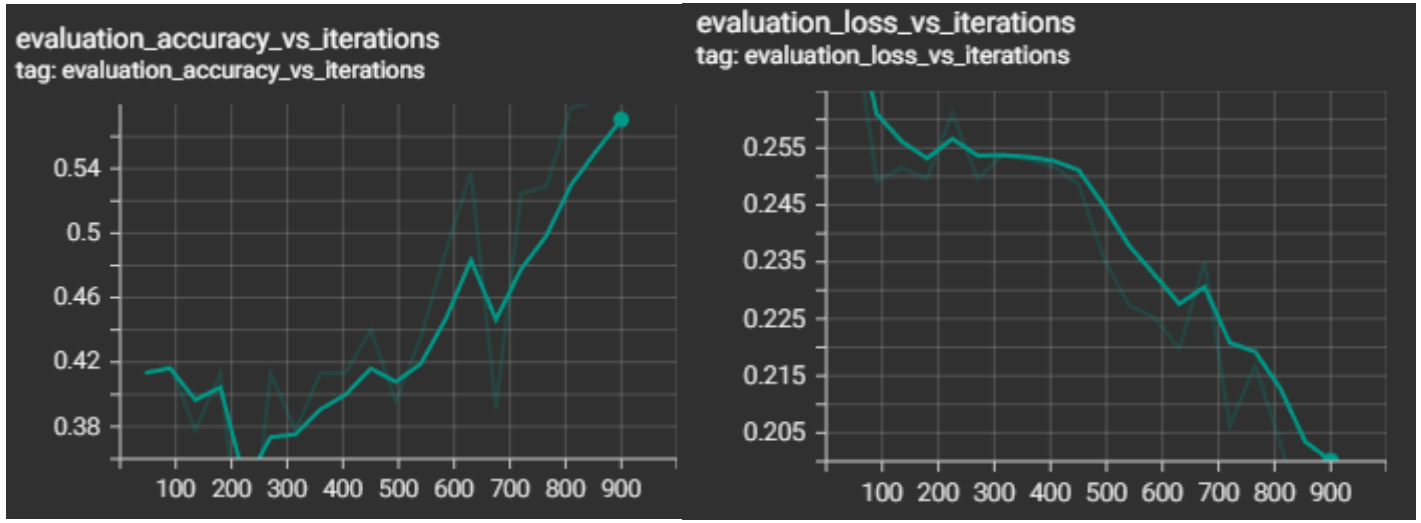
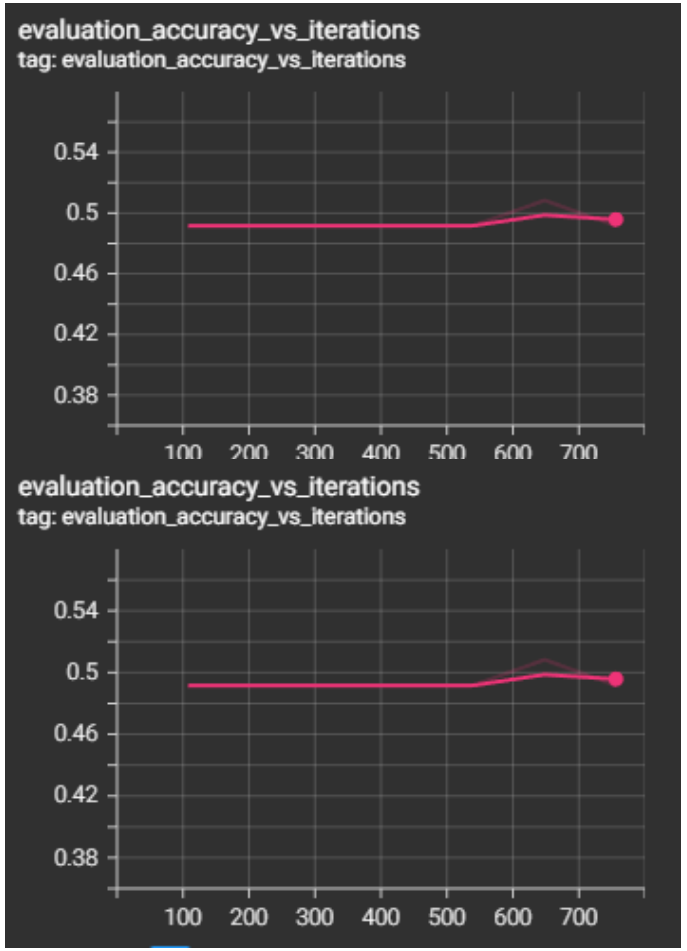


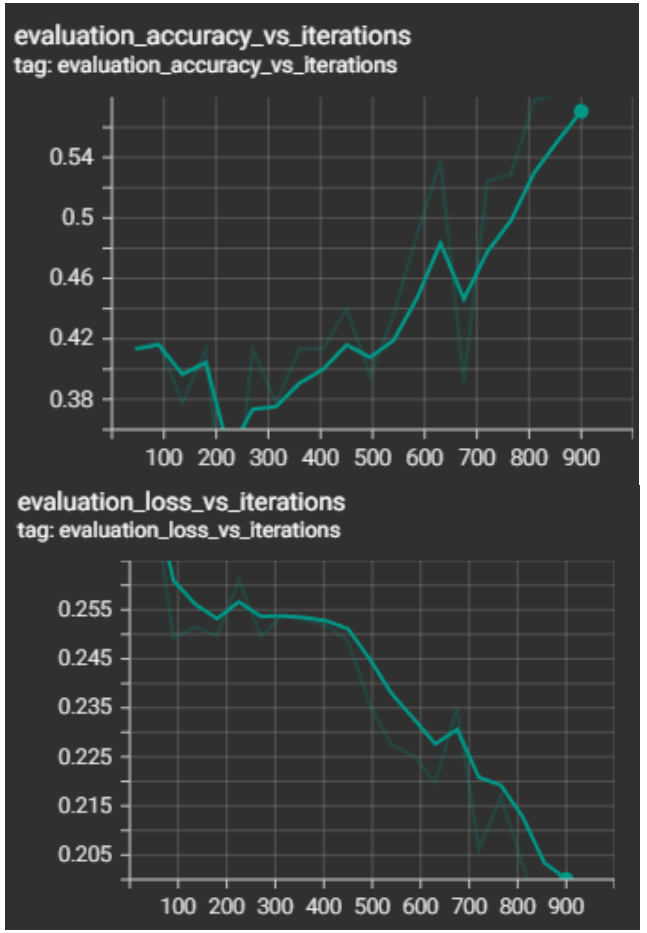
Abbildung 1: [Li+19]

3.3.2 Finale Trainingsergebnisse

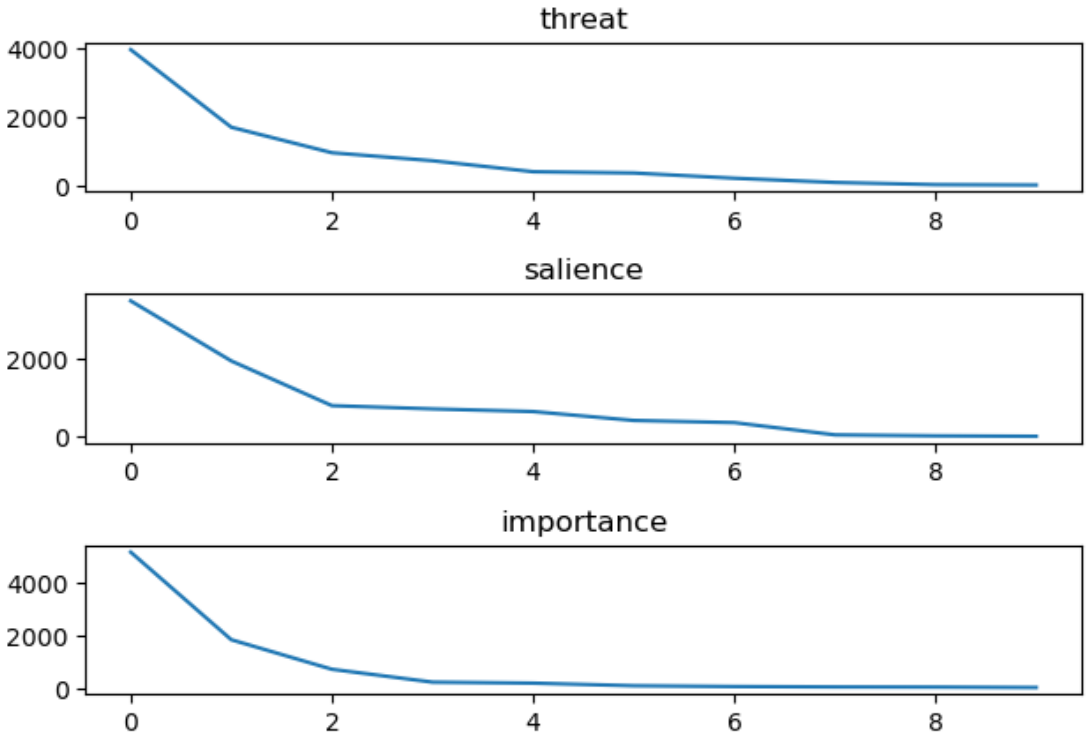


3.4.1 Vergleich mit anderen eigenen Modellen





3.4.2 Vergleich mit anderen eigenen Modellen



4. Probleme

- An *nützliche* Audiodaten kommen, bzw. großen Datensatz erstellen
- „Kurve abflachen“ sonst gibt KI nur average Wert aus
- Modell erstellen (zu viele Möglichkeiten)
- Python / Programmieren
- Organisation (Zeitmanagement, Zeitaufteilung, Aufgabenaufteilung, Strukturierung)
- Forschungsfragen sollten so früh wie möglich formuliert werden, um Ziele klar zu setzen

5. Diskussion

- Improvisierung für zukünftige Projekte
  - Erhöhter Datensatz an (wichtigen) Samples
  - „Cleanere“ Samples (bessere Qualität, ohne Soundlücken usw.)
  - Realitätsnähe
  - Anwendungsgebiete / Gründe
  - Andere Bauart (Reinforcement learning)
- Eine große Menge an Daten alleine macht keinen guten Datensatz
- Kleinere Tests wären vor einer größeren Umsetzung sinnvoll

6.1 Quellen

[Ab+16] Abadi, M. et al.: TensorFlow: A system for large-scale machine learning: arXiv, 2016, URL: https://arxiv.org/abs/1605.08695.

[CAH20] Cheuk, K. W.; Agres, K.; Herremans, D.: The Impact of Audio Input Representations on Neural Network based Music Transcription: 2020 International Joint Conference on Neural Networks (IJCNN): IEEE, 2020, S. 1–6, URL: https://ieeexplore-ieee-org.ezp.hs-duesseldorf.de/document/9207605.

[CJK17] Choi, K.; Joo, D.; Kim, J.: Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras: arXiv, 2017, URL: https://arxiv.org/abs/1706.05781.

[Di+22] Dim, C. A. et al.: Alert systems to hearing-impaired people: a systematic review. In Multimedia Tools and Applications, 2022, Jg. 81, H. 22, S. 32351–32370, URL: https://link.springer.com/article/10.1007/s11042-022-13045-1.

[Ge+17] Gemmeke, J. F. et al.: Audio Set: An ontology and human-labeled dataset for audio events: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): IEEE, 2017, S. 776–780, URL: https://ieeexplore.ieee.org/document/7952261.

[Hu+14] Hua, K.A. et al., Hrsg.: Proceedings of the 22nd ACM international conference on Multimedia. New York, NY, USA: ACM, 2014.

[KuCh20] Kumar, K.; Chaturvedi, K.: An Audio Classification Approach using Feature extraction neural network classification Approach: 2nd International Conference on Data, Engineering and Applications (IDEA): IEEE, 2020, S. 1–6, URL: https://www.semanticscholar.org/paper/An-Audio-Classification-Approach-using-Feature-Kumar-Chaturvedi/bbba6ed17708bbfff217351ca21cbaa013fbf08b.

[Ll20] Lloret Mauri, J., Hrsg.: ACHI 2020. Wilmington, DE, USA: IARIA, 2020.

[DAM19] Das, P. P.; Acharjee, A.; Marium-E-Jannat: Double Coated VGG16 Architecture: An Enhanced Approach for Genre Classification of Spectrographic Representation of Musical Pieces: 2019 22nd International Conference on Computer and Information Technology (ICCIT): IEEE, 2019, S. 1–5, URL: https://ieeexplore-ieee-org.ezp.hs-duesseldorf.de/document/9038339

6.2 Quellen

[Mc+23] McFee, B. et al.: librosa/librosa: 0.10.0: Zenodo, 2023.

[Mc+15] McFee, B. et al.: librosa: Audio and Music Signal Analysis in Python: Proceedings of the 14th Python in Science Conference: SciPy, 2015, S. 18–24, URL: https://www.semanticscholar.org/paper/librosa%3A-Audio-and-Music-Signal-Analysis-in-Python-McFee-Raffel/e5c114afc8c4d4e10ae068ba8e3387cc13e17a6e.

[SJB14] Salamon, J.; Jacoby, C.; Bello, J. P.: A Dataset and Taxonomy for Urban Sound Research. In (Hua, K. A. et al. Hrsg.): Proceedings of the 22nd ACM international conference on Multimedia. New York, NY, USA: ACM, 2014, S. 1041–1044, URL: https://dl.acm.org/doi/10.1145/2647868.2655045.

[SuAr22] Sundareswaran, V.; Aravindhar, J.: Sound Classification System Using Deep Neural Networks for Hearing Impaired People. In Wireless Personal Communications, 2022, H. 1, S. 385–399, URL: https://link.springer.com/article/10.1007/s11277-022-09750-7.

[Wa15] Warsaw University of Technology (Karol J. Piczak): ESC: Dataset for Environmental Sound Classification: Harvard Dataverse, 2015, URL: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YDEPUT.

[YaKö18] Yağanoğlu, M.; Köse, C.: Real-Time Detection of Important Sounds with a Wearable Vibration Based Device for Hearing-Impaired People. In Electronics, 2018, Jg. 7, H. 4, S. 50, URL: https://www.semanticscholar.org/paper/Real-Time-Detection-of-Important-Sounds-with-a-for-Yaganoglu-K%C3%B6se/ad7d23e8a45fd69e4215f50d6b74601e341f45da.

[Li+19] Liu, F. et al.: Intelligent and Secure Content-Based Image Retrieval for Mobile Users. In IEEE Access, 2019, Jg. 7, S. 119209–119222, URL: https://ieeexplore.ieee.org/document/8798734

[Shi+20] Shiraishi, Y. et al.: Alarm Sound Classification System in Smartphones for the Deaf and Hard-of-Hearing Using Deep Neural Networks: ACHI 2020 URL: https://www.thinkmind.org/articles/achi_2020_3_10_28007.pdf

Vielen Dank für eure Aufmerksamkeit!