Hochschule Luzern T&A
Technik & Architektur

Master of Science in Engineering
Field of specialization: Industrial Technologies

Master Thesis, FS20

Acoustic Scene and Room Classification for Real-Time Applications

Creation of Binaural Multi-Label Audio Dataset including Scenes and Soundscapes
Training of Multi-Output Deep CNN in Python/Tensorflow with Keras
Implementation Concept of Optimized CNN on Dedicated Hardware

Author Silvio Emmenegger
Hochschule Luzern T&A

Advisor Prof. Dr. Jürgen Wassner
Hochschule Luzern T&A

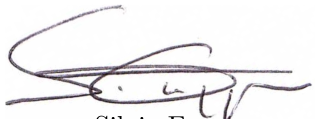
Industrial partner Prof. Dr. Thomas Graf
Hochschule Luzern T&A

Expert Dr. David Perels
Sonova AG

Document classification: Confidential
Horw, July 10, 2020

Probity Statement

I hereby declare that I have prepared the present work independently and have used no other than the specified aids. All used text excerpts, citations or contents of other authors were expressly marked as such.
Horw, July 10, 2020



Silvio Emmenegger

Abstract

Processing of acoustic signals is often accompanied by adaptive filtering and parameter adjustments in order to achieve optimal audio quality for specific tasks. In terms of hearing aids, the intention is an optimal speech intelligibility and environmental audio perception. Since acoustic scenes and soundscapes are constantly changing during operation, adjustments in parameters for hearing devices have to be executed in real-time. We introduce a system which is able to continuously recognize acoustic environments using **AI!** (**AI!**) in the form of a Deep Convolutional Neural Network (CNN) with focus on real-time implementation. Inspired by VGGNet-16, the CNN architecture was modified to a multi-label multi-output model which is able to predict combinations of scene and soundscape labels simultaneously while sharing the same feature extraction. For training we acquired a custom dataset consisting of 23.8h of high-quality binaural audio data including five classes per label which are clearly distinguishable by humans. Using a manual Grid Search method, we were able to optimize three models with respect to different complexity metrics for choosing a trade-off between accuracy and throughput. CNNs were then post-quantized to 8-bit which achieved an overall accuracy of 99.07% in the best case. After reducing the number of **MAC!** (**MAC!**) operations by a factor 154x and parameters by 18x, the classifier was still able to detect scenes and soundscapes with an acceptable accuracy of 94.82% which allows real-time inference at the edge on discrete low-cost hardware with a clock speed of 10 MHz and one inference per second.

Contents

PART I
Introduction

1 Initial Situation

1.1 State of the Art

Abbreviations of Adaptive Moment Estimation (ADAM) or Convolutional Neural Networks (CNNs) are introduced here. Sections can be referenced by Sec. B.0.1, while tables and figures can be referenced the same way (see Tab. 1 and Fig. 1 or Fig. 2).

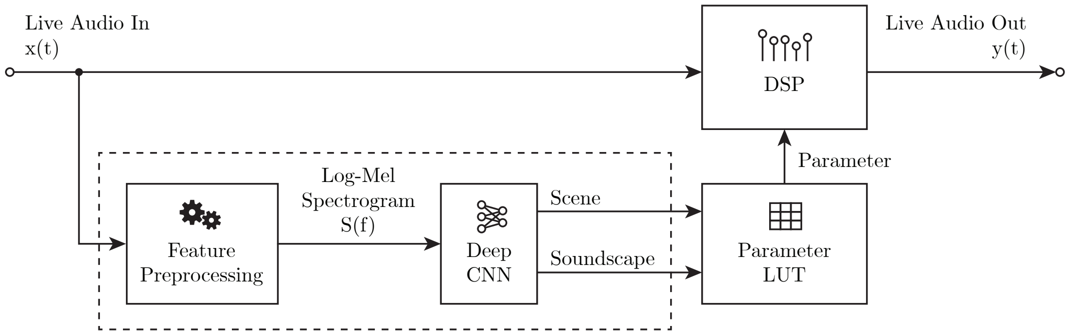


Figure 1: Caption 1 below figure [1].

Figure 2: Caption 2 below figure.

1.2 Our Approach

2 Project Scope

2.1 Time Horizon

2.2 Previous Work

2.3 Documentation Structure

PART II
Concept

3 Problem Identification

Col 1	**Col 2**	**Col 3**
left-aligned text	centered text	right-aligned text

Table 1: Caption 1 below table [2].

As you can see, there are two captions. Maybe you want to add a reference in the caption below the table but not in the list of tables.

3.1 Task Definition

Information [*].
Text-variants:

- **bold font**
- *italic font*
- `technical expressions`
- MATHEMATICAL EXPRESSIONS SUCH AS T_{inf}

3.2 Real-Time Requirements

4 Existing Methods

5 Chosen Approach

[*]Footnote text.

PART III
Realization

PART IV
Results

9 Results

10 Conclusion

10.1 Outlook

List of Abbreviations

ADAM	Adaptive Moment Estimation
CNN	Convolutional Neural Network
HSLU	Hochschule Luzern

List of Figures

List of Tables

Bibliography

[1] J.A.M. Vermaseren. Axodraw. *Computer Physics Communications*, 83(1):45 – 58, 1994. ISSN 0010-4655. doi: 10.1016/0010-4655(94)90034-5.

[2] Thorsten Ohl. Drawing Feynman diagrams with LaTeX and Metafont. *Computer Physics Communications*, 90:340–354, 1995. doi: 10.1016/0010-4655(95)90137-S.

[3] D. Binosi and L. Theussl. JaxoDraw: A Graphical user interface for drawing Feynman diagrams. *Computer Physics Communications*, 161:76–86, 2004. doi: 10.1016/j.cpc.2004.05.001.

[4] D. Binosi, J. Collins, C. Kaufhold, and L. Theussl. JaxoDraw: A Graphical user interface for drawing Feynman diagrams. Version 2.0 release notes. *Computer Physics Communications*, 180:1709–1715, 2009. doi: 10.1016/j.cpc.2009.02.020.

[5] Yifan Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10(1):37–71, 2005.

[6] Eades Peter and Sugiyama Kozo. How to draw a directed graph. *Journal of Information Processing*, 13(4):424–437, 1991.

[7] Jannis Pohlmann. *Configurable graph drawing algorithms for the TikZ graphics description language*. PhD thesis, Institute of Theoretical Computer Science, Universität zu Lübeck, Lübeck, Germany, 2011. URL `http://www.tcs.uni-luebeck.de/downloads/papers/2011/2011-configurable-graph-drawing-algorithms-jannis-pohlmann.pdf`.

[8] Till Tantau. The TikZ and PGF packages, 2015. URL `http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf`.

[9] R. P. Feynman. Space-time approach to quantum electrodynamics. *Phys. Rev.*, 76:769–789, Sep 1949. doi: 10.1103/PhysRev.76.769. URL `http://link.aps.org/doi/10.1103/PhysRev.76.769`.

[10] Joshua Ellis. TikZ-Feynman: Feynman diagrams with TikZ. 2016. URL `http://arxiv.org/abs/1601.05437`.

PART V
Appendix

A Attachments

Following documents are attached at the end of this document:

- 000: Task Formulation
- 001: Research Plan
- 002: Project Plan

For employees of Hochschule Luzern (HSLU), all source code and calculation documents are available inside a GitLab repository on the Enterprise Lab servers:

- User access by request: silvio.emmenegger@hslu.ch
- Current work: https://gitlab.enterpriselab.ch/acoustics-ai/asrc-for-real-time-applications
- Related work: https://gitlab.enterpriselab.ch/acoustics-ai

B Code snippets

B.0.1 Main script

```
import argparse
import yaml


def main():

  # parse arguments
  ap = argparse.ArgumentParser()
  ap.add_argument('--cfg_file', type=str)
  ap.add_argument('--do_train', type=int, default=1)
  ap.add_argument('--do_quant', type=int, default=0)
  ap.add_argument('--folds', nargs='+', default='all')
  args = ap.parse_args()
  if args.cfg_file is None:
    raise ValueError('Please specify a config file over run argument --cfg_file=..')

  # load optimization parameters
  config_file = open(args.cfg_file, 'r')
  config = yaml.load(config_file, Loader=yaml.FullLoader)
  config_file.close()

if __name__ == "__main__":
  main()
```

Listing 1: Python script.

B.0.2 Configuration Generator

```
Sub GenerateConfigs()
  Dim Row As Integer

  Row_Last = 35
  Col_Key = 1
  Col_Default = 2
  Key_Padding = 15

  Worksheets("RunSettings").Activate
  Prefix = Cells(Row_Last + 2, 3).Value
  Run_IDs = Split(Cells(Row_Last + 3, 3).Value, "-")
  Output_Dir = Cells(Row_Last + 4, 3).Value
  First_Run_ID = CInt(Run_IDs(0))
  Last_Run_ID = CInt(Run_IDs(1))

  For RUN_ID = First_Run_ID To Last_Run_ID
    Col_Run = RUN_ID + 3
    Str_Run_ID = CStr(RUN_ID)
    Str_Run_ID = WorksheetFunction.Rept("0", 2 - Len(Str_Run_ID)) & Str_Run_ID
    file = Output_Dir & Prefix & "_" & Str_Run_ID & ".yaml"
    Open file For Output As #1
    For Row = 3 To Row_Last
      K = Cells(Row, Col_Key).Value
      V = Cells(Row, Col_Default).Value
      V_Run = Cells(Row, Col_Run).Value
      If V_Run <> "" Then
        V = V_Run
      End If
      K = K & ":"
      K_Padded = K & WorksheetFunction.Rept(" ", Key_Padding - Len(K))
      If IsNumeric(V) Then
        V = Replace(V, ",", ".")
      End If
      If V <> "" Then
        out = K_Padded & V
      Else
        out = K
      End If
      Print #1, out

    Next Row
    Close #1
  Next RUN_ID

End Sub
```

Listing 2: VBA script.

MSE - Masterthesis

Aufgabenstellung für:

	Silvio Emmeneggr	(Masterstudierende/r)
	Industrial Technologies	(Fachgebiet)
von	Prof. Dr. Jürgen Wassner	(AdvisorIn)
	Dr. David Perels	(Experte/Expertin)

1. Arbeitstitel

Acoustic Scene and Room Classification for Real-Time Applications

2. Fremdmittelfinanziertes Forschungs-/Entwicklungsprojekt

-

3. Industrie-/Wirtschaftspartner

-

4. Fachliche Schwerpunkte

Deep Learning
Raumakustik
Acoustic Scene Classification

5. Inhalt

Bei der Verarbeitung von Akustiksignalen ist es oftmals notwendig den Signalverarbeitungs-algorithmus bzw. dessen Parameter an die aktuelle akustische Umgebung (Raumgeometrie und –eigenschaften, Geräuschkulisse und Störquellen) zu adaptieren um optimale Ergebnisse zu erzielen. Im Fall von Hörgeräten kann dies z.B. eine optimale Sprachverständlichkeit sein, wobei die Algorithmus- bzw. Parameteranpassungen dann in Echtzeit erfolgen müssen, da die akustische Umgebung ständig variiert.

In der vorliegenden Arbeit soll ein System entwickelt werden, welches die für eine Echtzeit-Adaptierung nötige fortlaufende Erkennung der akustischen Umgebung mit Hilfe von Deep Learning Methoden realisiert. Ausgehend von den Ergebnissen der beiden Vorgängerprojekte [1][2] sollen dafür folgende Punkte bearbeitet werden:

- Das System soll die Umgebung möglichst gleichzeitig bezgl. akustischer Szenerie sowie Raumtyp klassifizieren können. Optional soll das System zusätzlich ausgewählte Stichworte in gesprochener Sprache detektieren können.
- Die Klassen der zu unterscheidenden akustischen Szenen und Raumtypen sollen so gewählt werden, dass sie für eine Hörgeräte-Applikation repräsentativ sind.
- Für Training und Test des zu entwerfenden neuronalen Netzes soll ein Datensatz durch Messungen in realer Umgebung erstellt und durch geeignete Methoden augmentiert werden.
- Die Architektur des trainierten Netzwerkes soll durch Anwendung eines bestehenden evolutionären Suchalgorithmus [3][4] für eine Echtzeitimplementierung mittels des in [5] beschriebenen Verfahrens optimiert werden.
- Für die effiziente Implementierung des Klassifizierungsvorganges inkl. aller nötigen Vorverarbeitungsschritte nach der Mikrofon-A/D-Wandlung soll ein Systemkonzept entwickelt werden, welches realistische Anforderungen bezgl. Latenz, Kosten und Leistungsaufnahme erfüllen kann. Die vollständige Realisierung dieses Systems ist nicht Teil der Aufgabe.

6. Fachliteratur/Web-Links/Hilfsmittel

[1] S. Emmenegger. Acoustic Scene Classification with Neural Networks. MSE Vertiefungsarbeit 1. Hochschule Luzern – Technik &Architektur 2019.

[2] S. Emmenegger. Classification of Acoustic Room Properties from Speech Samples. MSE Vertiefungsarbeit 2. Hochschule Luzern – Technik &Architektur 2020

[3] F. Johner, J. Wassner. Efficient Evolutionary Architecture Search for CNN Optimization on GTSRB. 18th IEEE International Conference on Machine Learning and Applications. 2019.

[4] M. Kurmann. Optimierung Neuronaler Netze für die FPGA Implementierung. MSE Vertiefungsarbeit 1. Hochschule Luzern – Technik &Architektur 2020.

[5] M. Fischer. BinArray: A Scalable Hardware Architecture for Binary Approximated CNNs. Master Thesis. Hochschule Luzern – Technik &Architektur 2020.

7. Durchführung der Arbeit

Termine

Start der Arbeit:	Mo. 17.02.2020 (Semesterbeginn FS20)
Abgabe Prüfungsexemplar:	bis Fr. 19.06.2020 um 17.00 Uhr im Sekretariat BA&MA oder direkt an ExpertIn und AdvisorIn (Sekretartiat BA&MA muss darüber informiert werden)
Verteidigung:	bis spätestens Mi, 01.07.2020
Meldung Grade:	Do. 02.07.2020
Abgabe def. Masterthesis:	Fr. 10.07.2020 bis 17.00 Uhr auf Ilias
Diplomausstellung:	Fr. 03.07.2020

→ Weitere Termine gemäss Ablauf Master-Thesis

8. Dokumentation

Die definitive Masterthesis ist in **doppelter Ausführun**g (für AdvisorIn und Experte/Expertin) zu erstellen. Die Masterthesis enthält zudem zwingend

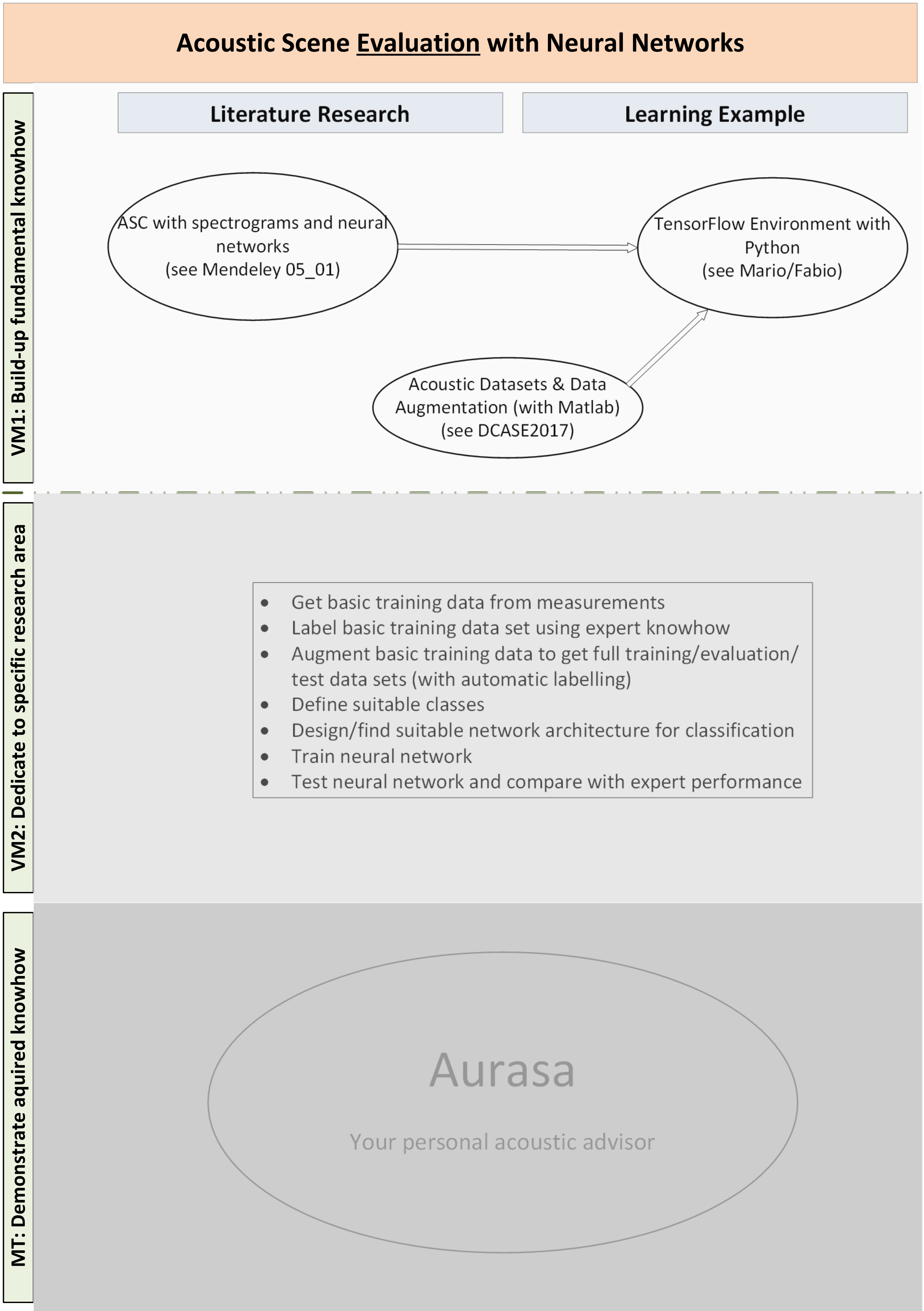
- Selbstständigkeitserklärung anhand der Vorgaben der Bibliothek (verfügbar auf MyCampus)
- Titelblatt anhand der Vorgaben der Bibliothek (verfügbar auf MyCampus)
- einen Abstrakt in deutscher und englischer Sprache
- Die Abgabe der vollständigen elektronische Daten (Berichte, Präsentationen, Messdaten, Programme, Auswertungen, etc.)

9. Zusätzliche Bemerkungen

Betreffend Geheimhaltung und Rechte am Geistigen Eigentum ist die Vereinbarung zwischen dem Studierendem, der HSLU und dem Industriepartner massgeblich.

Horw, *Datum*

AdvisorIn	Experte/Expertin	Studierende/-r
________________________	________________________	________________________


Acoustic Scene Evaluation with Neural Networks
VM1: Build-up fundamental knowhow
Literature Research
Learning Example
ASC with spectrograms and neural networks (see Mendeley 05_01)
TensorFlow Environment with Python (see Mario/Fabio)
Acoustic Datasets & Data Augmentation (with Matlab) (see DCASE2017)
VM2: Dedicate to specific research area
Get basic training data from measurements
Label basic training data set using expert knowhow
Augment basic training data to get full training/evaluation/test data sets (with automatic labelling)
Define suitable classes
Design/find suitable network architecture for classification
Train neural network
Test neural network and compare with expert performance
MT: Demonstrate aquired knowhow
Aurasa
Your personal acoustic advisor

Overview/Intro ASC

Barchiesi, D., Giannoulis, D. D., Stowell, D., & Plumbley, M. D. (2015). Acoustic Scene Classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, *32*(3), 16–34. https://arxiv.org/pdf/1411.3715.pdf

Wang, D. (2017). Deep learning reinvents the hearing aid. *IEEE Spectrum*, *54*(3), 32–37. https://ieeexplore.ieee.org/document/7864754

Datasets

DCASE2017. (n.d.). Retrieved June 8, 2018, from http://www.cs.tut.fi/sgn/arg/dcase2017/index

ASC with NN

Kahl, S., Hussein, H., Fabian, E., Schloßhauer, J., Thangaraju, E., Kowerko, D., & Eibl, M. (2017). Acoustic Event Classification Using Convolutional Neural Networks. *Informatik 2017*, 2177–2188. https://doi.org/10.18420/in2017_217

Park, S., Mun, S., Lee, Y., & Ko, H. (2017). Acoustic scene classification based on convolutional neural network using double image features. In *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*. Retrieved from https://pdfs.semanticscholar.org/f0a7/1758980b22356e56d36ecbe243e8ea5ce8da.pdf

Lu, L., Yang, Y., Jiang, Y., Ai, H., & Tu, W. (2018). Shallow Convolutional Neural Networks for Acoustic Scene Classification. *Wuhan University Journal of Natural Sciences*, *23*(2), 178–184. https://doi.org/10.1007/s11859-018-1308-z

Han, Y., & Lee, K. (2016). Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation. Retrieved from https://arxiv.org/abs/1607.02383

Weiping, Z., Jiantao, Y., Xiaotao, X., Xiangtao, L., & Shaohu, P. (2017). Acoustic Scene Classification Using Deep Convolutional Neural Network and Multiple Spectrograms Fusion. Retrieved November 6, 2018, from https://www.cs.tut.fi/sgn/arg/dcase2017/documents/challenge_technical_reports/DCASE2017_Xing_158.pdf

Dang, A., Vu, T. H., & Wang, J.-C. (2018). Acoustic scene classification using convolutional neural networks and multi-scale multi-feature extraction. In *2018 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–4). IEEE. https://doi.org/10.1109/ICCE.2018.8326315

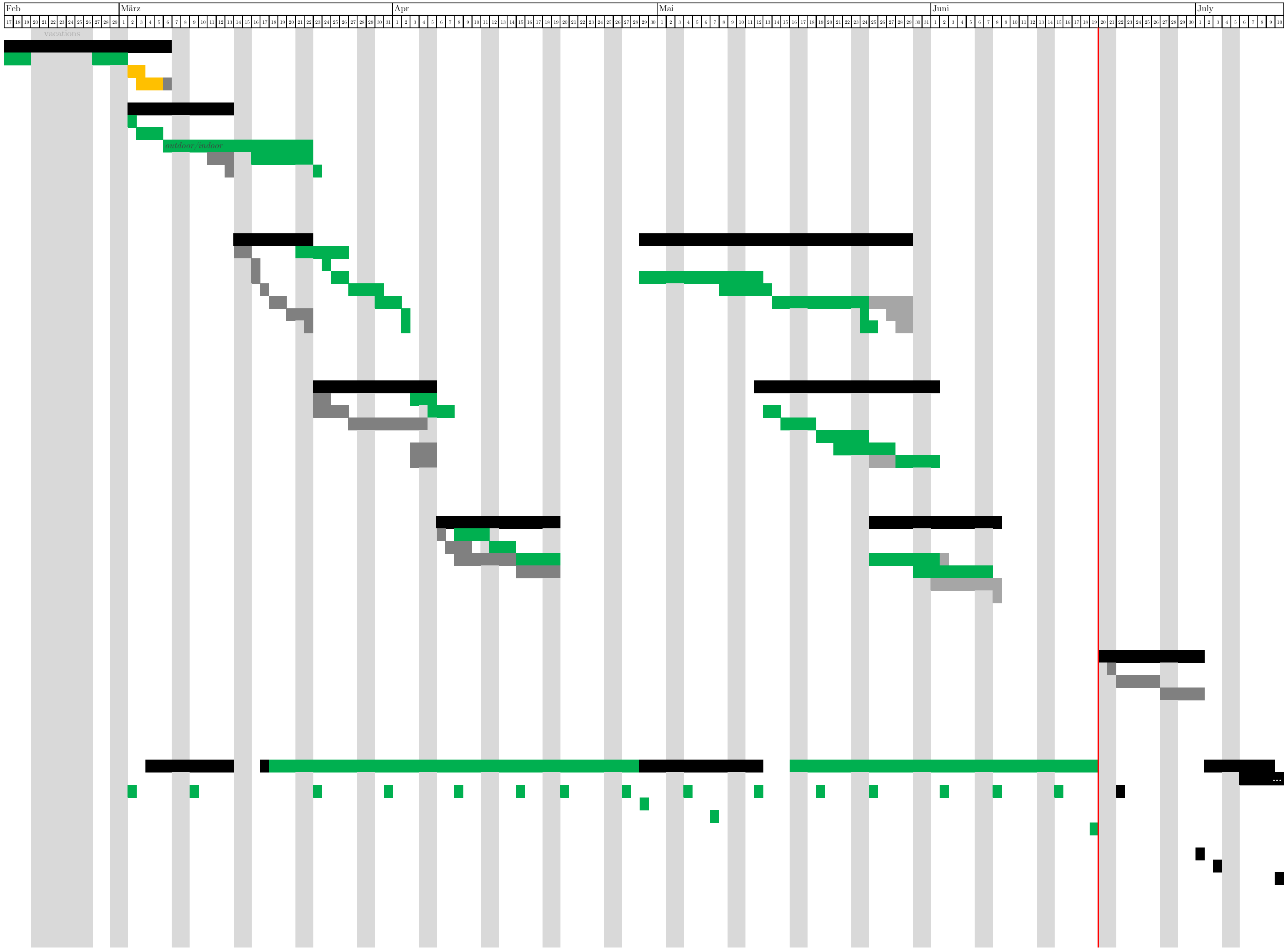
Battaglino, D. (n.d.). ACOUSTIC SCENE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS. Retrieved from http://www.eurecom.fr/fr/publication/4982/download/sec-publi-4982.pdf

Project Plan

Master Thesis MSE: Acoustic Scene and Room Classification for Real-Time Applications

silvio.emmenegger@hslu.ch

Last Update: June 19, 2020



Tasks

Prestudies

Read previous papers and works

Research similar image classification methods

Build CNN strategy (Single-Shot Detector)

WP1: Dataset Creation

Collect label informations & discuss

Order and setup recording equipment

Record dataset

Postprocess dataset

Review recorded dataset

Outcomes WP1:

- Recording equipment and software

- 24h of qualitative audiological recordings in indoor/outdoor locations resp. Rooms

WP2: CNN Training

Write import adapter for recorded dataset

Plan final learning architecture (2D labels)

Setup Keras learning scripts

Train NN and tune optimization parameters

Retrain & Apply Crossvalidation

Tune model (opt. build ensembles)

Review and collect results

Outcomes WP2:

- accurate NN model with ≈70% classification accuracy

- dedicated label prediction system

WP3: CNN Optimization

Introduction to EA library (Fabio)

Create adapter for pretrained model from WP2

Optimize architecture on MAC

Optimize architecture on accuracy

Review results and select best model

Quantize model to 8 bit resolution

Outcomes WP3:

- optimized CNN model with ≈90% reduced architecture

- EA adaper for acoustic problems

WP4: Implementation Concept

Introduction to BinArray (Mario)

Build basic concept for implementation on FPGA

Refine implementation concept (Mainly Preprocessing)

Review concept and make first coarse predictions

Design specific hardware preprocessing architecture

Review hardware architecture (with Mario/Jürgen)

Outcomes WP4:

- Overview about predicted performance and accuracy for implementations on different FPGA families and subtypes.

WP5: Demonstrator

Setup live recording

Build Python live demonstrator for optimized CNN

Refine demo GUI

Outcomes WP5:

- Interactive Python demonstrator with live plots

- (First implementation steps on FPGA of optimized model)

Documents & Deadlines

Documentation

Paper

Meeting

Midterm Presentation

Colloquium MSE

Deadline Documentation Official

Grade Fix

Final Presentation

Diploma Exhibition

Deadline Documentation Complete *(17:00 ILIAS)*