Robust Sound Event Detection in Binaural Computational Auditory Scene Analysis

vorgelegt von
Dipl.-Ing.
Ivo Trowitzsch
ORCID: 0000-0001-7596-6377

an der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

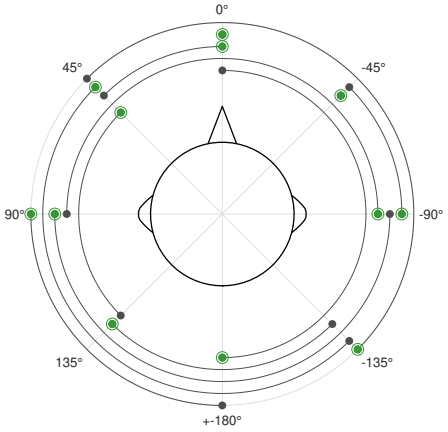
Vorsitzender: Prof. Dr. Reinhold Orglmeister
Gutachter: Prof. Dr. Klaus Obermayer
Gutachterin: Prof. Dr. Dorothea Kolossa
Gutachter: Prof. Dr. Thomas Sikora

Tag der wissenschaftlichen Aussprache: 19. November 2019

Berlin 2020

ROBUST SOUND EVENT DETECTION IN BINAURAL COMPUTATIONAL AUDITORY SCENE ANALYSIS

IVO TROWITZSCH



Building and Analyzing Models For Realistic Acoustic Environments: Dissertation

June 2019

Ivo Trowitzsch: *Robust Sound Event Detection in Binaural Computational Auditory Scene Analysis,* Building and Analyzing Models For Realistic Acoustic Environments: Dissertation, supervised by Prof. Dr. Klaus Obermayer, Neural Information Processing Group, Technische Universität Berlin. © Ivo Trowitzsch June 2019

ABSTRACT

Automatic sound event detection and computational auditory scene analysis gain importance through the increasing prevalence of technical systems operating autonomously or in the background, since such operation requires awareness of the system's environment.

In realistic scenes, reliable sound event detection, despite the big improvements of the related automatic speech recognition, still poses a difficult problem: general sounds often are less definable than speech and exhibit less regularities and rules; commonly, many sounds occur simultaneously and in all kinds of acoustic environments.

Binaural robotic systems are particularly interesting due to their resemblance of human means, but they are also more limited through the restriction to two microphones, specifically regarding spatial acoustic scene analysis. Spatial hearing figures prominently in humans, but for automatic sound event detection so far has gone mostly unregarded.

One of the core objectives running through the entire thesis is the development of fundamental systematic methodology with respect to (a) the building of robust sound event detection models, and (b) the elaborate analysis regarding their application in many different situations — both is underrepresented in available research publications.

In the hereinafter presented studies, sound event detection models are built in different training schemes and evaluated in detail with respect to their performance in various acoustic scene conditions. Analyses are conducted on scenes with one to four co-occurring sound events, with sound-to-sound energy ratios of $-20\,\mathrm{dB}$ to $+20\,\mathrm{dB}$, with different spatial source distributions, and in diverse acoustic environments from anechoic to church aula. It is shown (i) to which extent models that have been trained under specific acoustic conditions specialize to these, and (ii) that even with simple algorithms like logistic regression, through acoustically multifarious training almost optimal performances as achieved by the specialized models can be obtained consistently. The influence of temporal information integration is investigated, and it is shown that algorithms able to model context over longer durations benefit particularly in demanding scenes and get more precise in their detection.

Moreover, a method for joining sound event detection and source localization is presented by which coherent auditory objects can be created. The proposed system associates the attributes "sound type"

and “source location” successfully; for measuring success of such joint systems – almost uncharted territory –, performance measures are suggested. It is shown that in an active binaural system spatial sound event detection performance can be increased considerably through suited head orientation.

Finally, all developed models get tested in a simulated “online”-robotic system and their potential for forming integral components in computational auditory scene analysis is demonstrated.

ZUSAMMENFASSUNG

Automatische Geräuscherkennung und auditorische Szenenanalyse gewinnt mit der Verbreitung von technischen Systemen, die selbstständig oder im Hintergrund agieren, an Bedeutung, da selbstständiges Wirken ein Bewusstsein der Umgebung voraussetzt.

In realistischen Szenen stellt eine zuverlässige Geräuscherkennung trotz der Erfolge in der verwandten Spracherkennung allerdings nach wie vor ein schwieriges Problem dar: Geräusche sind oft weniger prominent abgrenzbar als Sprache und folgen weniger Regeln, und sie treten häufig vielfach überlappend auf und in verschiedensten akustischen Umgebungen.

Binaurale robotische Systeme sind auf Grund ihrer Ähnlichkeit mit dem Menschen besonders interessant, aber durch die Begrenzung auf zwei Mikrophone auch eingeschränkter, insbesondere in Hinsicht auf die räumliche akustische Szenenanalyse. Räumliches Hören spielt für den Menschen eine wesentliche Rolle, wurde bis jetzt aber in der automatischen Geräuscherkennung praktisch nicht beachtet.

Ein die gesamte Dissertation durchziehendes Kernanliegen ist die Erarbeitung von grundlegender, systematischer Methodik sowohl in Bezug auf die Erstellung von robusten Geräuscherkennungsmodellen, als auch in Bezug auf deren ausführliche Analyse hinsichtlich der Anwendung in verschiedenen Situationen — beides ist in verfügbaren Forschungsarbeiten unterrepräsentiert.

In den im folgenden präsentierten Studien werden Geräuscherkennungsmodelle in verschiedenen Trainingsschemata entwickelt und im Detail bezüglich ihrer Erkennungsleistung in verschiedensten akustischen Szenenkonfigurationen evaluiert. Analysen finden über Szenen mit ein bis vier gleichzeitig aktiven Geräuschen, mit Geräusch-zu-Geräusch-Energieverhältnissen von $-20\,\mathrm{dB}$ bis $+20\,\mathrm{dB}$, mit verschiedenen räumlichen Quellenverteilungen, und in verschiedenen akustischen Umgebungen von reflexionsfrei bis Kirchensaal statt. Es wird gezeigt, (i) wie stark Modelle, die unter bestimmten akustischen Bedingungen trainiert werden, sich auf diese spezialisieren, und (ii) dass selbst mit einfachen Algorithmen wie der logistischen Regression durch akustisch möglichst mannigfaltiges Training fast durchgehend optimale Erkennungsleistungen wie von den spezialisierten Modellen erreichbar sind. Der Einfluss von temporaler Informationsintegration wird untersucht, und gezeigt, dass Algorithmen, die einen Kontext über längere Zeiträume modellieren können, davon speziell in herausfordernden Szenen stark profitieren und präziser in ihrer Erkennung werden.

Schließlich wird eine Methode zur Verbindung von der Geräuscherkennung mit einer Quellenlokalisierung vorgestellt, mit der auditorische Objekte mit kohärenten Attributen erzeugt werden können. Das präsentierte System verknüpft die Attribute „Geräuschtyp“ und „Quellenort“ erfolgreich; zur Bemessung des Erfolgs eines solchen kombinierten Systems – fast komplettes Neuland – werden Leistungsmaße vorgeschlagen. Es wird gezeigt, dass in einem aktiven binauralen System die räumliche Erkennung durch passende Orientierung des Kopfes erheblich gesteigert werden kann.

Final werden alle entwickelten Modelle in einem simulierten „online“-Robotiksystem getestet und gezeigt, dass sie wie vorhergesagt funktionieren und integrale Bestandteile einer automatischen auditorischen Szenenanalyse darstellen können.

© *Ahoi Polloi* – Thank you for entertaining me.

ACKNOWLEDGMENTS

Many thanks go to Joris, who allowed me much time and waited patiently for me to finish "my book", which I'm afraid will not fulfill his expectations of an interesting story for (at least) many years. Many thanks go to Tina, who is the one that made possible for me to invest the effort needed into this work, and had to bear with my stress more often than we both liked it. It has not always been fun, and I promise to not do it again.

Many thanks go to Moritz, Youssef, and Johannes, who I could work together with at least temporally, which was great. Many thanks go to the Two!Ears members, particularly Christopher, Ning, Dorothea, Guy, for the close collaboration in this project. Many thanks go to Maziar, who was my beloved office companion for many years. Many thanks go to Moritz, again, for a lot of emotional support during my last years. In general, it was a privilege to be able to share work and family situation with you. Many thanks go to Caro for a great final countdown and many good dinners together. Thank you Franzi, Veronica, Florian, Caglar, Josef, Christoph, Rong, Vaios, and the others of the NI group for a lot of good times in the office together.

Many thanks go to Klaus, who put me onto the Two!Ears project and with it gave me the possibility to do research on sound-related machine learning, which is as close as it gets to my interests. I enjoyed fruitful discussions with him, he gave me encouragement about the quality of my work, and generally a lot of trust.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AFE Auditory Front-end
AMLTTP Auditory Machine Learning Training and Testing Pipeline
AMS amplitude modulation spectrogram
ASA auditory scene analysis

BAC balanced accuracy
BAPR best-assignment-possible-rate
BRIR binaural room impulse response

CASA computational auditory scene analysis
CNN convolutional neural network
CRNN convolutional recurrent neural network

DCASE Detection and Classification of Acoustic Scenes and Events
DNN deep neural network

FC fully connected
FN false negative
FP false positive

GLM generalized linear model
GMM Gaussian mixture model

HMM hidden markov model
HRIR head-related impulse response

ILD interaural level-difference
ITD interaural time-difference

KEMAR Knowles Electronic Manikin for Acoustic Research

LASSO Least Absolute Shrinkage and Selection Operator
LDNN long short-term memory (LSTM) + deep neural network (DNN) architecture
LSTM long short-term memory
LTI linear time-invariant

MC multi-conditional
MFCC Mel-frequency cepstral coefficients
MLD maximum lateral distance

MLP multilayer perceptron

NEP number of excess positives
NIGENS *N*eural *I*nformation processing group *GEN*eral *S*ounds
NMF non-negative matrix factorization

pCV partial cross-validation

ReLU rectified linear unit
RNN recurrent neural network

SED sound event detection
SELD sound event localization and detection
SNR signal-to-noise ratio
SSL sound source localization
SVM support vector machine

TCN temporal convolutional network
TF time-frequency
TN true negative
TP true positive

Part I

FIRST THINGS FIRST

1

INTRODUCTION

Sound, in common phrasing, is referring to two (tightly related) phenomena: the sound waves, oscillations in (usually air) pressure originating from a physical vibration source; and the auditory reception and perception of these waves in our brains, creating a sensation.

Hearing is the sense of sound perception, and thus a major part of life. It increases situational awareness, facilitates interaction, and can create sensations with strong emotional impact. While vision supposedly is more powerful with respect to comprehension of the world we live in, sound may be coupled tighter to emotional reactions. Music works very well without added visuals, but making movies work without music is a difficult task. The emotional aspect holds not only for music (Kryter 2013): hearing a crying baby without seeing it likely is emotionally more painful compared to seeing a crying baby, but not hearing it. Hearing birds sing feels peaceful. Hearing very deep sounds creates tension, because something big and potentially dangerous is anticipated[1]. Alarms, screeches, and screams are frightening and make one want to flee. Sound and hearing is an intuitive sense, with a shortcut to emotions — and because of that, I like sound, and find it a fascinating research object.

This chapter introduces the background, scope, related work, terminology, and research questions covered in this thesis.

1.1 AUDITORY SCENE ANALYSIS

In everyday life, we are surrounded by many different sources of sounds; isolated or simultaneously emitted, from clearly identifiable directions or diffuse, quiet or loud, continuous or discrete, structured or noise-like. Natural sounds, human sounds, sounds from human-made environments, machine sounds, music — there is a large range

1 which is why sound designers in movies make excessive use of a lot of deep sounds to make scenes exciting that otherwise wouldn't be

of different sounds and sources, and a wide range of transformations to these sounds and conditions in which they occur.

Usually, we identify these acoustic events instantaneously and subconsciously, often many events blur into an acoustic scene. If we want to (or sometimes even if we don't want to), we can focus and attend to a particular sound. The acoustic pressure waves reaching our ears are the arithmetic sum of the pressure waves generated from the individual sound sources, and energies from these individual sound sources often overlap in time and frequency. We only have two sensors – our ears –, but often efficiently process sounds from more sources, so the reliable human identification of sound events is remarkable. However, research on environmental sounds, compared to speech and music, has so far been underrepresented (Gygi and Shafiro 2007).

When our listening experience is rather occupied with the *sources* of what we hear, this process is called *everyday listening* (Gaver 1993a, 1993b), compared to when we focus on the actual sound and its attributes like pitch or timbre, which is then called musical listening. In this definition, everyday listening is the perception of sound-producing events, which in the mentioned work are divided into a hierarchy based on the materials and interactions of sources.

Relatedly, A. S. Bregman (1994) coined the term *auditory scene analysis (ASA)*. In a definition given by him in A. Bregman (2008),

> auditory scene analysis is the name for both a problem and a perceptual process. The problem is how to form mental representations of individual sounds from the summed waveform that reaches the ear of the listener. It is also the name of the brain process that accomplishes this result.

This process produces perceptual interpretations which he calls *auditory streams*: separate coherent patterns – "sounds" – attributed to individual acoustic sources present in the acoustic scene. Auditory scene analysis can thus be thought of as *de-mixing* of the mixture that physically occurs when acoustic sources generate signals simultaneously. Bregman described how the human auditory system uses "bottom-up" rules about acoustic regularities, sequential and simultaneous organization to build its mental representations of distinct sounds. "Top-down" rules are less clearly defined and investigated, and involve conscious processes like attention, or past experience with particular sound classes.

1.2 COMPUTATIONAL AUDITORY SCENE ANALYSIS

Computational auditory scene analysis (CASA) (Brown and Cooke 1994; Ellis 1996; Rosenthal and Okuno 1998; Wang and Brown 2006) is the

computer-based realization of ASA, in a human-like *binaural* scope. At the core is segregation into auditory streams[2] with individual sounds, as in ASA. Mostly (e. g., in all of above works except Ellis (1996)), CASA is speech-focused[3] (as arguably speech is the sound type most important for humans), and then CASA is defined through components like speech detection, segregation and separation of speech streams, localization of speakers (and potentially subsequent recognition of speech contents, but this is not classically part of the CASA process itself). Famous as the speech-centric description of the task of CASA is the "cocktail party problem" (Cherry 1957).

Lyon (2010, 2017) cover *machine hearing*, strongly related to CASA, but not coupled as tightly to the original concepts of stream formation of ASA, and without focus on speech or music. Finding "meaning" and "understanding" of sounds therein is not necessarily preceded by stream segregation (nor is stream segregation the goal); and this segregation anyway is difficult to completely be separated from actual identification of the sounds, Lyon (Lyon 2017, Ch. 23.3.2) writes:

> In all of these [CASA] approaches, some kind of attention mechanism is needed to decide which part of the mixture to pay attention to. This attention mechanism must be at least partly in, or controlled by [..] the application that defines the kind of meaning that the system is trained to extract.

Similarly, in Bregman's work on ASA, top-down "schema-based" processes aid in segregation, where auditory features belonging to learned patterns are grouped together. In this line of argumentation, *sound identification* can be considered an integral component of (a broader defined) CASA, because identifying sounds goes hand in hand with forming mental representations of individual sounds.

Sound identification, and, more general, CASA, are of similar importance for machines as for humans (Gygi and Shafiro 2007) — recognizing sounds and analyzing acoustic scenes helps understanding the current environment (Lyon 2010). Robots can achieve better situational awareness, and thus will be able to execute more targeted actions. Sounds can help identification of objects that are in sight, but not clearly recognized; and sounds can also be heard when sources are visually obstructed.

In medical technology, particularly hearing aids (maybe also cochlear implants) are still in need of significant improvements (Gygi and Ann Hall 2016), and hence certainly it is fair to assume that the more "understanding" of what is heard, the better the filtering relevant from

2 consisting of auditory objects
3 sometimes, additionally, music-focused

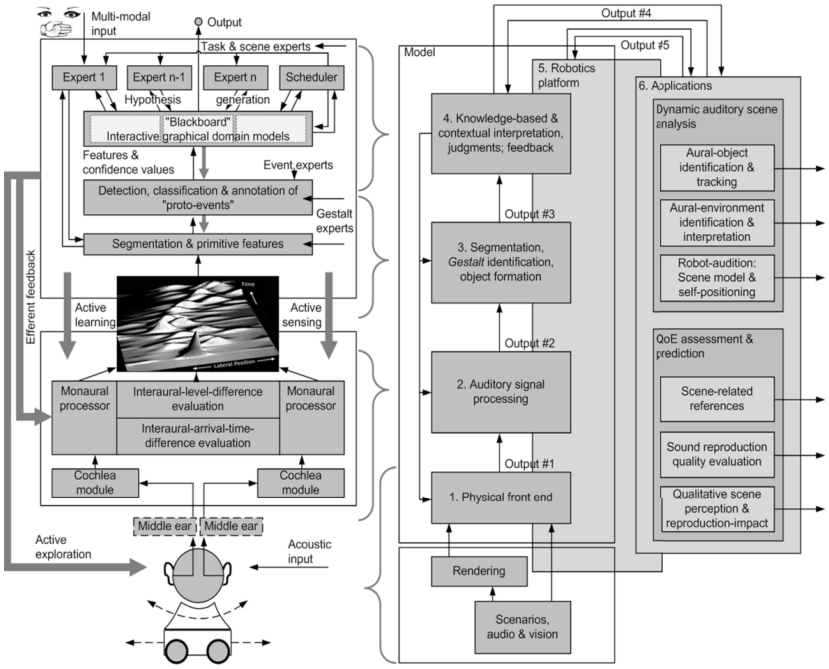


Figure 1.1: Two!Ears system overview. Left panel: Flow graph that illustrates the degree of detail in current discussions within the consortium. Two kinds of feedback are differentiated in the figure: low-level feedback akin to reflexive circuits and feedback from higher-level processing triggered by hypotheses in the blackboard system. Right panel: Diagram of the main functional blocks of Two!Ears, also referring to the technical work packages. © Two!Ears

irrelevant. In all kinds of bio-sensory technology, for example for detecting anomalies in breathing in a baby phone, but also like in animal sensing, detecting sounds can be important.

And of course just in any general kind of assistance technology – at home (Alexa etc.), in cars (check for snoring), in safety-critical situations (cocking a gun, smashing a window), and so on –, understanding observed acoustic events can be helpful.

1.3 THE TWO!EARS PROJECT

A considerable part (both time-wise and content-wise) of this thesis was realized during the course and in the frame of the Two!Ears EU-project (Raake et al. 2014). Two!Ears' goal was to "develop an intelligent, active computational model and platform of auditory perception and

experience"[4], with a strong emphasis on keeping the system comparable to human auditory processing such that it could help further understanding of human auditory perception. The model's core was a system of individual experts in a blackboard system responsible for different tasks like stream formation, sound event identification and localization, head rotation commandment, or acoustic quality assessment. A key factor of the project was considered the combination of bottom-up (signal-driven) and top-down (hypothesis-driven) processing, that is, the inclusion of feedback from higher-level experts to lower-level experts, to the robot or to the auditory processing. The system was developed to run in a simulated environment as well as on a robot; with an open and extendable architecture published as public-domain. Fig. 1.1 presents a diagram of the proposed Two!Ears system and functional blocks.

1.4 ENVIRONMENTAL SOUND IDENTIFICATION

This thesis is occupied with general (without restriction of types) *sound event detection (SED)* in binaural systems such as the Two!Ears system.

Sound event detection (or: acoustic event detection) refers to the detection of specific sound events (not the detection of general sound activity) — in its application, it basically is synonymous with what one would call *sound identification*. From a terminological point of view, a detector "searches" for occurrences of sound events of a given specific type, while identification searches for the correct type given an occurrence of a sound. However, in technical implementation (when using many sound event detectors for different sound types), both commonly boil down to the same thing, an attribution of sound event type to an occurrence of a sound, and this is always only possible within the range of known types of the system.

In a distinction, SED usually includes detecting on- and offsets of sound events and/or is performed on continuous streams, whereas sound identification is also called sound classification, sound recognition, audio classification, and audio tagging, when detecting sound events' temporal "borders" is irrelevant. Sound event detection, in contrast to other forms of sound identification, thus has to be the modus operandi of any online system required to identify sounds "live" and (at least almost) instantaneously.

Research until recently has mostly focused on classification of full-length sound events, and is still more active in this domain. However, although SED is more difficult than audio classification (McLoughlin et

Research Gap

4 www.twoears.eu

al. 2017; Huy Phan et al. 2017), the two share the largest part of related methodology, and results from one often carry over to the other. Very commonly – and in this thesis –, SED actually is implemented through very-short-segment audio classification (on partial sound events) with sliding windows (segments). Therefore, in the following, relevant research is presented and related with this work across both modes of sound identification.

For long, research on sound identification has been underrepresented compared to automatic speech recognition and music analysis (Lyon 2010). However, the analysis of environmental sounds is different from the analysis of these two special cases (Alıías et al. 2016) in that general sounds exhibit a wide range of variability from very fluctuating to very stationary (Yamakawa et al. 2010), and thus has become a field of its own. This research has been accelerated a lot by the CLEAR (Stiefelhagen et al. 2007) and Detection and Classification of Acoustic Scenes and Events (DCASE) challenges (A. Mesaros et al. 2018; Annamaria Mesaros et al. 2019; Plumbley et al. 2018; Dan Stowell et al. 2015) throughout the last decade.

1.4.1 *Common classifiers*

Progress in the field over the last years largely has been made due to improved classifiers. Commonly used classifiers for quite a while were particularly support vector machines (SVMs), Gaussian mixture models (GMMs), and hidden markov models (HMMs) (Sharan and Moir 2016; Stiefelhagen et al. 2007; Dan Stowell et al. 2015), but deep learning methods (LeCun et al. 2015) in various forms are predominant in sound event detection by now (Hertel et al. 2016; Li et al. 2017; Huy Phan et al. 2016; Purwins et al. 2019). A. Mesaros et al. (2018) and Annamaria Mesaros et al. (2019) analyzed the DCASE 2016 and DCASE 2017 SED challenges results, and found that in DCASE 2013, there were no DNN entries, in DCASE 2016, there were many but not all more successful than others, while in DCASE 2017 and 2018 there were almost only DNN systems and they have been always the top-performing ones.

1.5 INFLUENCE OF ACOUSTIC CONDITIONS ON BINAURAL SOUND EVENT DETECTION

Research on sound event detection has not gone back a very long way, even less so research on general sound event detection in *complex* acoustic scenes. Until recently, most work was done on *monophonic* SED, that is, only sounds occurring without superposition of other sound

events were identified (Sharan and Moir 2016; Dan Stowell et al. 2015). Consequentially, models usually have turned out very sensitive to perturbations and differing conditions, as e. g. in Dufaux et al. (2000).

1.5.1 *Noisy data*

Several groups demonstrated improvements based on engineering of potentially noise-robust features or models for audio classification in mismatched conditions with added noise down to 0 dB: for example with one-class SVM and wavelet features (Rabaoui et al. 2008), kNN classifiers on spectrogram image features (Dennis et al. 2012), convolutional neural networks (CNNs) on mel-frequency spectrograms (Haomin Zhang et al. 2015), or spiking-neurons-learning (Wu et al. 2018).

However, in their evaluations, they all have limited disturbances to diffuse background noise (of different types, like babble or jet cockpit noise), which make singular sound events stand out stronger compared to other disturbing sound events. Even though these background noises are not stationary (as e. g. white noise), the problem is that they are almost always *more* stationary than the target sound events. That such noise would be the only or main source of acoustic perturbation, is an unrealistic assumption in a lot of situations[5]. *Research Gap*

Typically, the described systems exhibit modest performance decreases for SNRs down to 10 dB, and serious degradation for values down to 0 dB.

McLoughlin et al. (2017) recognized above described shortcomings, and partially alleviated them by modifying the evaluation scheme to continuous sound event detection and slight *polyphony*. They find that these changes decrease system performances by as much as 50 % for 0 dB noise level, compared to the original tests without added polyphonic events and on the whole sound events.

1.5.2 *Polyphonic data*

The introduction of polyphony extends the problem depth significantly (Stiefelhagen et al. 2007): there may be an arbitrary number of sounds co-occurring, at different ratios of energy, overlapping sounds may be of very different and hence unpredictable structures, and one cannot assume that a particular sound dominates the auditory scene (Barker et al. 2005; López-Pacheco et al. 2016). Non-stationary, highly variable general disturbances emitted from distinct sources are different from

5 Gygi and Shafiro (2007) hypothesizes that the same is a problem in human auditory scene research: the common signal/masker scheme is often unrealistic in the noise being too stationary.

steady, diffuse background noise and might influence detection in more unpredictable ways.

The DCASE challenges went to tackle this gap, and introduced the first polyphonic SED challenge in 2013 (Dan Stowell et al. 2015), attracting modest participation, followed by 2016 (A. Mesaros et al. 2018) and 2017 (Annamaria Mesaros et al. 2019) with a lot more contributions. 2013 and 2016 events hosted synthetic polyphonic tasks (on synthesized scenes mixed from isolated sound events), 2016 and 2017 added real-life-audio polyphonic tasks.

In DCASE 2013, results still reflected strong difficulties in recognizing sound events from noisy scenes with the employed methods (mostly HMMs, some still with Mel-frequency cepstral coefficients (MFCC) features), but performances improved starting from 2016, where more than half of the submissions employed DNNs; MFCCs had been replaced by mel-scaled time-frequency representations. Success on the synthetic task in 2016 was evenly distributed between DNNs and traditional methods (non-negative matrix factorization (NMF), random forests, kNN); the winning system employed NMF (Komatsu et al. 2016), very closely followed by two deep learning systems (Choi et al. 2016; Hayashi et al. 2016).

The works of Benetos et al. (2016), Gemmeke et al. (2013), and Heittola et al. (2011) represent popular earlier attempts to deal with polyphony by building exemplar-based (also called dictionary-based) NMF-systems; the underlying idea was to find bases (exemplars) in time-frequency domain that are closer to monophonic representations of individual sources.

Representative for more recent approaches to polyphonic SED, Cakir et al. (2015b), Hayashi et al. (2017), and Parascandolo et al. (2016) presented well-performing deep learning architectures operating on mel-frequency spectrograms, using different methods of data augmentation to increase training data size and variability. Neural networks by design are able to predict multi-labels (several different positives at a time), and are thus inherently more directly relatable to polyphonic situations than many other binary models like SVMs.

However, to spoil the improved performances of presented systems: the levels of polyphony for the tasks actually have been modest, namely between averagely 1.1 and 1.8 simultaneously active sources for the 2013 event synthetic scenes, between 1.2 and 1.5 for the 2016 event synthetic scenes, and 2.53 for the real-life-audio — both measured excluding times of no sound event activity, i. e., with a lot of additional "gaps" more easy to classify and hence increasing average performance.

Research Gap

A. Mesaros et al. (2018) and Annamaria Mesaros et al. (2019) concluded that data-driven approaches were replacing manual design, but

that systems needed to improve handling data imbalance across sound types: contributions across the different events commonly failed to detect several small (with respect to number of occurrences) sound types, which hints on (a) problems in the definitions of optimization loss and (b) problems in the definitions of task metrics. Also, they stated in their review of DCASE 2016 and 2017 that databases with strongly labeled sound events (including on- and offset times) are still insufficient in size for robust learning of sound event detection. *Research Gap*

Unfortunately, work done on polyphonic SED like the ones introduced here usually presented "well-performing" systems, but no analysis with respect to the different conditions. *Research Gap*

1.5.3 *Binaural data*

Research on sound event detection has rarely been done on *binaural*[6] settings. Exemplary for a few exceptions (basically all working with the same dataset), Adavanne and Virtanen (2017), A. Mesaros et al. (2010), and Parascandolo et al. (2016) identified sounds from real life recordings with a person wearing microphones in his ears. Adavanne, Politis, and Virtanen (2018) and Xu, Kong, Huang, et al. (2017) have shown that models for acoustic event detection benefit from multi-channel acquisition with two or more microphones. However, no investigations regarding dependency on the added binaural acoustic modality of sources locations relative to the head have been published. *Research Gap*

1.5.4 *Systematic analyses*

Even though commonly, sound event detection research is not performed on "clean" data only any more, *analyses* with respect to the different conditions are rare. Sometimes, as in the examples above, this comes as a "small extra", most commonly then as an investigation of the effect of diffuse background noise (usually with SNRs only down to about 0 dB, although humans certainly are able to detect sounds with lower SNRs), but no systematic, fine-grained studies on different acoustic modalities in wide ranges have been published. There are only few works in which the dependency of sound event detection performance on number of sources and energy ratios between sources are analyzed (Adavanne, Politis, Nikunen, et al. 2018; Lafay et al. 2017); but the effects of different source positions – irrelevant for monaural (single-channel) data, more relevant for multi-channel data, and partic- *Research Gap*

6 Strictly used, "binaural" refers not to any two-channel data like stereo data, but two-channel data related to acquisition through two *ears*.

ularly interesting for binaural human-like robotic systems –, as well as the impact of different room acoustics are basically unexplored.

The above described lack of analyses is two-fold: it is a lack on studies about how the different conditions influence the *maximally* achievable model performances, and it is a lack on studies about how *deviations* of conditions from training conditions degrade model performances.

Research Question 1

What influence can be expected on binaural sound identification of various realistic different acoustic modalities? How big is the impact on performance upon deviations from training conditions?

Contribution 1

Thorough analyses on the influence of (i) the number of co-occurring sources up to 4, (ii) the energy ratios between co-occurring sources down to $-20\,\mathrm{dB}$, (iii) the locations of sources, and (iv) the room acoustics on sound event detection are provided. The aspect of correct and robust performance measurement and its importance is illuminated and contrasted to common practice. A sound database tailored to the task of polyphonic SED is made available.

1.6 ROBUST SOUND EVENT DETECTION

The main problem of robust sound event detection designed to be applied in the "real world" is: all of above described acoustic conditions usually are unknown a priori, that is, it is difficult to tune them to specific conditions to achieve maximum performance. Instead, it is necessary to build models that exhibit low degradation upon deviations from specific conditions.

Audio classification research has traditionally been close to signal processing domains, with stronger focus on engineering models than on building models from data. In this line, approaches to making models robust to disturbances like noise long time have been handcrafted methods to eliminate the impeding influences as far as possible (Bardeli et al. 2010; Cooke et al. 2001; Zhuang et al. 2010). Nowadays, with increasing computing power and increasing access to a wealth of data, this approach in general (for problems difficult to manually solve by handmade rules) is more and more replaced by data-driven model building through techniques of machine learning. This applies particularly to sense-related applications like visual object recognition, speech recognition, or also sound identification.

1.6.1 *Multi-conditional training*

Data-driven robustness through *multi-conditional training* across conditions has been applied successfully for a while in speech-related audio analysis systems. Saeidi et al. (2010) and H. Yu et al. (2016) have assessed multi-conditional training for speaker identification, and showed that error rates are clearly reduced under known and unknown noisy test conditions, while performance for undisturbed conditions almost remains optimal. Yin et al. (2015) trained a DNN speech recognition system individually under various different noisy (and clean) conditions, and cross-tested it on other conditions. They demonstrated that training on all noisy and clean data together resulted in optimal performance across different (a priori unknown) conditions. Rajnoha (2009) showed that multi-conditionally trained speech recognition models outperform models with a noise reduction system, but trained on clean data, in real environments. Hsiao et al. (2015) and Ko et al. (2017) showed that speech recognition can be improved strongly by multi-conditional training across real or even simulated reverberant room acoustics. Multi-conditional training is also found in the context of multi-speaker localization models, where diffuse Gaussian noise was superimposed on top of target speech sounds at different signal-to-noise ratios in order to reduce front-back confusions and increase generalization performance (May, Ma, et al. 2015).

For sound event detection, multi-conditional training has so far not been employed as regularly and even less analyzed. Using signals from multiple channels as multiple conditions has been demonstrated to yield increased performance in Giannoulis et al. (2014) and H. Phan et al. (2015). Dennis et al. (2010), Huy Phan et al. (2016), and Q. Yu et al. (2019) employed multi-conditional training over SNRs and noise types, and showed that this improved their performances over models trained on clean data in mismatched conditions significantly, but only diffuse background noise was used (no polyphony), and only down to 0 dB.

Research Gap

Research Gap

Martin-Morato et al. (2018) recently investigated the robustness of features from a very-large-scale-trained DNN ("SoundNet", Aytar et al. (2016)), with respect to background noise and reverberation. Their results, achieved without multi-conditional training, indicate severe drops in performance when testing in reverberant conditions, and very strong influence of background noise, even already for SNRs above 0 dB.

Since the prevalence of deep learning is accompanied with the need for lots of data, *data augmentation* is being used in increasing frequency; in the DCASE 2018 task 2 (DCASE Community 2019), there were almost

no contributions without any form of it. Data augmentation can be seen as a form of multi-conditional training, although usually it is less clearly tied to different acoustic conditions, and its goal commonly is not robustness across varying conditions, but to help prevent overfitting on training set sound events. Often, variants of time stretching or frequency shifting are employed (e. g. Piczak (2015a) and Salamon and Bello (2017)). So-called "block-mix", "mixup" or "between-class" augmentation (Takahashi et al. 2016; Tokozume et al. 2017; Wei et al. 2018) also involve superposition of different training samples.

The DCASE 2018 challenge task 3 (D. Stowell et al. 2018) on bird audio detection specifically involved the problem of generalizing across different (also unknown) acoustic conditions and recording equipment. About 35 000 weakly labeled audio clips of 10 s length, by design multi-conditional, were made available as training data. The winning system (a CNN on mel-frequency spectrograms, Lasseck (2018)) showed quite good performance, making extensive use of further data augmentation like superposition (increased polyphony and noise), time and frequency shifting, and others.

In general, it can be stated that research on sound identification so far has often focused on optimizing the newest/best training algorithm and tuning features, with the goal of achieving high performance on specific data under specific conditions[7], and less on developing generally applicable methodology on how to robustly train and test models.

Research Question 2

Can multi-conditional binaural sound event detection models for polyphonic scenes be built that will generalize with high performance across different number of co-occuring sources (1-4), across different SNRs (20 dB to −20 dB), across different room acoustics (anechoic to church), and across different locations of sources? How will performance compare to maximally achievable performances gained through training and testing under the same conditions?

Contribution 2

Robustness can be achieved in a data-driven way with auditory-inspired features even with simple linear classifiers far from the learning power of DNNs. Multi-conditional training produces robust sound event detection models able to generalize across the proposed range of acoustic conditions with performances matching up to train-test-matching single-conditional models. Thorough analyses comparing these performances in iso- and cross-test setups are provided.

7 as promoted by challenges like DCASE

1.6.2 *Temporal context*

As stated above, of the research done on sound identification, the larger part is on audio classification, i. e., on classification of whole sound events. The overlap is large, but sound event detection with its "online" character has the added difficulty of usually having available less complete information (Huy Phan et al. 2017). However, it is unclear to what extent sound event systems actually need access to context over time in order to efficiently detect. How long is long enough? It seems likely that acoustic scene complexity would influence the necessary temporal context. This question has been tackled for acoustic scenes (Huy Phan et al. 2018), but not sound events.

Research Gap

Certainly DNN model architectures – that are able to integrate information over time – are state of the art nowadays: in DCASE 2017, almost only DNNs models participated, and the winning systems were all deep learning-based (Adavanne and Virtanen 2017; Cakir and Virtanen 2017; Jeong et al. 2017; Lim and Park 2017; Xu, Kong, Wang, et al. 2017). However, a systematic comparison with respect to the size of temporal context, acoustic conditions, and time-integration model capability has not yet been conducted.

Even when using recurrent neural networks (RNNs), which are able to adjust the accessed temporal context data-driven in training, usually fixed length audio sequences are employed (e. g. Jeong and Lim (2018)), which then often implies truncation of streams. The trade-off is potential loss in performance due to not using all available information (full time context), versus ease and speed of training and data preparation, as sequences of varying lengths complicate batch creation.

Research Question 3

To what extent is temporal context necessary, and to what is it profitable for efficient sound event detection? How does robust performance in difficult acoustic scenes depend on it?

Contribution 3

An analysis on how sound event detection performance increases with the size of temporal context is presented, and it is shown that it increases particularly for difficult auditory conditions and that it is very class-specific. The extent to which context can be accessed by simple linear models through statistically summarized temporal features is compared with the extent to which models able to learn temporal integration themselves can profit from. Two different modes of defining training targets are analyzed, one more "physically" motivated and supposedly more "exact", and one more perceptually motivated with a "smoothed" interpretation.

1.7 SPATIAL SOUND EVENT DETECTION AND ACTIVE COMPUTATIONAL AUDITORY SCENE ANALYSIS

As mentioned in Section 1.2, CASA includes also *localization* of sound sources; and of course information on sound type and sound source locations shall be attributed coherently – that is, *jointly* – to auditory objects.

Even by SED standards, research on joining *polyphonic* sound event detection and sound source localization is very scarce; available have been a few investigations on parallel SED and sound source localization (SSL) on monophonic events, like Lopatka et al. (2016). Only very recently, studies on truly joint systems started to appear, and this year, there will for the first time be a specific sound event localization and detection (SELD) challenge held at DCASE 2019. SELD is a considerably more difficult task than pure sound event detection, and additionally, data for training SELD models is basically not available and manufacturable only at significantly higher effort than data for SED model training.

Research Gap

There seem to be three approaches that have been started to get explored: (i) *sound-type masked SSL*, detecting sound events and then localizing them isolated through their type-specific time-frequency mask (Chakraborty and Nadeu 2014; Ma et al. 2018), (ii) *spatially masked SED*, localizing active sources and then identifying them isolated through their location-specific time-frequency mask (May et al. 2012) or through beam-forming (Grobler et al. 2017), (iii) *joint SELD*, localizing and identifying sound events simultaneously (Adavanne, Politis, Nikunen, et al. 2018; W. He et al. 2018; Hirvonen 2015).

Because it fitted well into the TWO!EARS system and into the concepts of CASA, the work described in this thesis follows the approach of detecting sound events on spatial streams. Compared to sound-type masked SSL, this has the advantage of allowing localized identification of multiple sources of the same type or with similar frequency ranges, compared to joint SELD, this approach is feasible also with less powerful model classes which are faster to train and easier to understand.

The related SELD system published in May et al. (2012) was confined on speaker localization and identification. Opposed to the online mode of sound event detection, it performed SELD on whole audio excerpts including complete speech segments. Unlike the data-driven multi-conditional approach followed in this thesis, training was conducted on clean data, and source-masking only in testing. Despite employing a missing data approach (as presented in Cooke et al. (2001)), their results consequentially exhibited strong negative dependence on SNR even above 0 dB.

Research Gap

The only related publications covering *binaural* SELD are Ma et al. (2018) and May et al. (2012). In general, spatial segregation and localization often rely on the availability of multiple microphones for maximum performance (Gannot et al. 2017; Nadiri and Rafaely 2014; Sumitani et al. 2019). However, on the one hand, human-inspired systems are scientifically interesting because of the possibility to compare with human performance and behavior, and in the best case even draw inferences about human mechanisms. On the other hand (with a bit of human egocentrism), it can be claimed that over the last few billions of years, using two ears has turned out to be the best compromise between performance and resource-usage, otherwise we would have more ears — at least for reaching human performance, which machines are not even close at yet for CASA, two channels are enough. So it seems a valuable goal to restrict the scope to two channels, from two ears.

Research Gap

Research Question 4

How can binaural multi-source sound event detection and localization be joint? Is detection on spatially segregated streams an effective approach? How can efficiency of joint sound event localization and detection be measured? Does robustness through multi-conditional training carry over to localized detection?

Contribution 4

Sound event detection and sound source localization can be joined efficiently through spatial masking in a modular system; robust performance can be achieved through multi-conditional training. Measures for quantification of localized sound event detection success are developed and presented. Analyses with respect to the different acoustic conditions are given; particularly the influence of spatial source distribution, which in binaural robotic systems is adjustable, is elaborated on. The effects of precision in localization and source count estimation are investigated.

1.7.1 *Active CASA*

One explicit goal of the Two!Ears project (Section 1.3) was the development of a platform enabling *active* computational auditory perception, with intertwined bottom-up and top-down processing. One dimensionality of "activity" is reflective feedback from high-level models to low-level robotic sensomotoric functions in order to achieve a specific auditory effect[8]. One example for such reflective feedback or "attention" is found in Ma et al. (2018), where high-level knowledge about present sound types was used to "focus" localization on auditory cues relevant

8 Confer Blauert et al. 2014 for a literature review on auditory feedback.

to this sound. With more reflexive feedback models, Bustamante et al. (2016) and Ma et al. (2017) have developed sound source localization models able to reduce uncertainty about locations by commanding head rotation and movement towards active sources.

From above introduced investigation and analysis on the influence of spatial source distribution around the head on localized detection performance, rules can be concluded about how to optimally orient the head — in the domain of sound event detection, let alone joint sound event localization and detection, this has not been done before. Implementing these rules as reflective feedback in the TWO!EARS simulation framework with the models developed in this thesis would be the final showcase of the proposed techniques' effectiveness.

Research Gap

Research Question 5

Are the developed SED and SELD models suitable components for a robotic binaural system performing computational auditory scene analysis? Can SELD performance be improved if this system is active and if the models are allowed commandment of head movement?

Contribution 5

Binaural computational auditory scene analysis with the developed components employed in the TWO!EARS robotic simulation platform is performed and analyzed. Higher detection and localization performances are demonstrated when models are allowed to reflectively command head rotation according to the system's observations.

1.8 STRUCTURE OF THIS THESIS

This thesis is structured as follows. Continuing this part, first, in Chapters 2 and 3, the newly compiled sound database NIGENS, data representations, methods of auditory scene generation, model building and testing are presented.

Part ii then starts with an investigation in Chapter 4 on performance of "single-conditional" SED models not particularly trained for acoustic robustness, when deviating from training conditions in noise level and in location of sources. To overcome the therein demonstrated sensitivity, Chapter 5 introduces "multi-conditional" SED modeling and shows that it enhances robustness severely. Chapter 6 continues with the multi-conditional approach and analyzes it in regard to changes of room acoustics, i. e. particularly reverberation. Chapter 7 completes this part with a study on the effect of temporal context on robustness, at the level of model type (involving state-of-the-art DNN models), and with respect to length of temporal context accessible for models.

Part iii widens the scope from pure SED towards integral CASA, and proposes and analyzes an approach to multi-conditional joint SELD in Chapter 8. In the subsequent Chapter 9, in addition to the previous systematic (but static) evaluation, the obtained models are tested and evaluated in the dynamic Two!Ears development system including simulation of head rotation (and showing its advantage).

Part iv with Chapters 10 and 11 finally discusses various aspects of this work and puts them into relevant scientific context, and concludes this thesis with a summary and the mentioning of the contributions of this work.

The appendix in Part v contains a description of the developed Auditory Machine Learning Training and Testing Pipeline (AMLTTP), which was crucial for the investigations in this thesis and is accessible as open-source.

2

BINAURAL SOUND AND REPRESENTATIONS

Crucial for training well-generalizing models and testing their generalization well[1] is having available enough suitable data — this holds for any machine learning, and also for sound event detection (SED). "Enough" and "suitable" for the purposes followed in this thesis were achievable only through simulation of acoustic scenes (in contrast to acquiring audio from real scenes). Simulation allows to create arbitrary numbers of scenes in well-defined configurations.

This chapter introduces the data used throughout this work and associated methods on three levels: Section 2.1 presents the collected original sound data, Section 2.2 explains the simulation of binaural acoustic scenes through generation from the original data, and Section 2.3 covers the utilized representations computed from the binaural scenes by auditory processing.

2.1 SOUND DATA – THE NIGENS DATABASE

This section is based on Ivo Trowitzsch, Jalil Taghia, et al. 2019b. *The NIGENS General Sound Events Database.* Technical report. Technische Universität Berlin. eprint: arXiv:1902.08314.

Compared to speech recognition, which is more mature (and still one of the most active domains of applied machine learning research), general sound event detection has only recently picked up pace, particularly since the introduction of the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge series in 2013 (Dan Stowell et al. 2015). This is also reflected in the availability of general sound event databases, which are still scarce (Fonseca et al. 2017; "IEEE DCASE 2016 Challenge" 2016; Annamaria Mesaros et al. 2017; Piczak 2015b; Salamon et al. 2014) and have their limitations (see Section 2.1.1).

Since complex acoustic scenes should be simulated, a database of isolated high quality sound events was needed, big enough for the

1 pun intended

development of robust SED models. No suitable database was available unfortunately – available databases were either very small, or didn't contain isolated sound events, or didn't provide event on- and offset annotations, or all of these. Hence, it was decided to build a new database containing sound events of 14 different types[2], strongly labeled with perceptual on- and offset times: the *N*eural *I*nformation processing group *GEN*eral *S*ounds (NIGENS) database, tailored to the task of synthesizing complex acoustic scenes.

While real recordings of complete scenes would always be the gold standard, particularly for training on *spatial* acoustic scenes, a data set like NIGENS and simulated scenes are indispensable: labeling of (many) recorded real spatial audio with ground truth about sound types and source locations would be of prohibitive effort, and scene synthesis the only realistically viable option to obtain well-defined acoustic scenes of specific complexity.

To enable training of models which are able to cope with disturbances of unknown type, a large collection of "general" sounds of all kinds and sorts except the 14 target classes was included in addition to sounds of the detector target classes, which is another unique feature. Section 2.1.2 describes the database's contents more detailed.

NIGENS is accessible at Trowitzsch, Taghia, et al. (2019a) and described by Trowitzsch, Taghia, et al. (2019b), free to use non-commercially under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license.

2.1.1 *Other data sets*

The following is a list of the other mentioned datasets, that contain (more or less) isolated sound events:

- The DCASE 2016 ("IEEE DCASE 2016 Challenge" 2016) task 2 (synthetic audio sound event detection) data set consists of 20 short mono sound files for each of 11 sound classes (from office environments, like `clearthroat`, `drawer`, or `keyboard`), each file containing one sound event instance. Sound files are annotated with event on- and offset times, however silences between actual physical sounds (like with a phone ringing) are not marked and hence "included" in the event. This data set is very small.
- The DCASE 2017 (Annamaria Mesaros et al. 2017) rare sound events task data set contains isolated sound events for three classes: 148 crying babies (mean duration 2.25 s), 139 glasses

2 chosen to be able to realize an emergency scenario, which was one demonstration application in the Two!Ears project

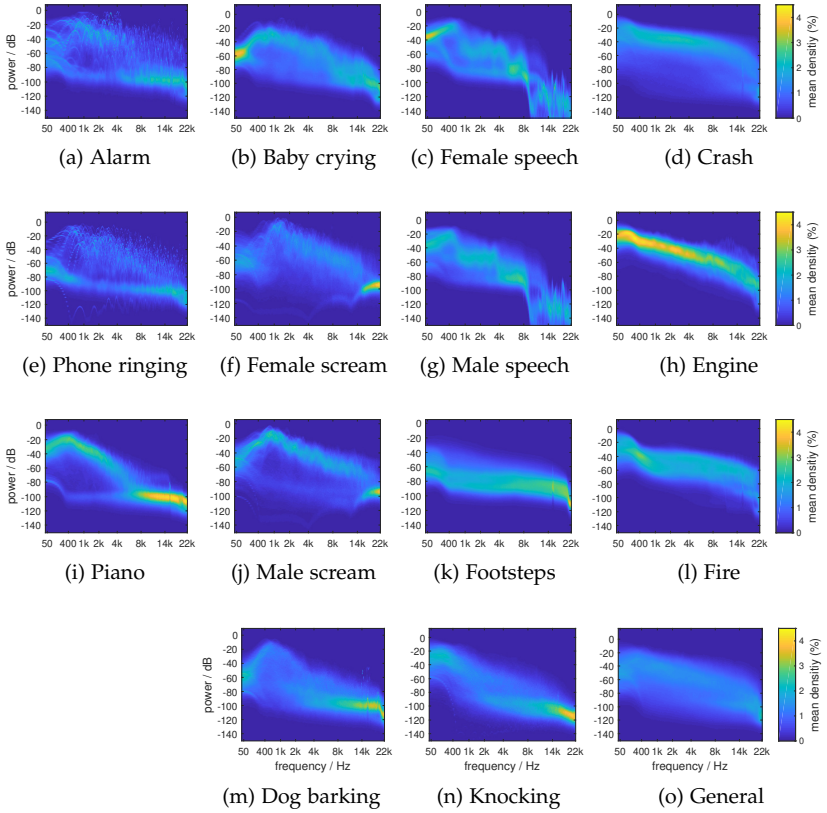
breaking (mean duration 1.16 s), and 187 gun shots (mean duration 1.32 s). As with the DCASE 2016 data, silences are not excluded from active event markings in the annotations. While this data set contains many samples per class, there are only three classes, which limits possible scene generation and also generalization of obtained results considerably.

- The UrbanSound and UrbanSound8k datasets (Salamon et al. 2014) provide a large database with 1302 different sound files (containing 27 h of audio) distributed across ten classes of urban environments, like car horn, dog bark, or jackhammer. Sounds originated from `Freesound.org` and were enhanced by manual annotations of sound event starting and ending times. Unfortunately, sound events are not necessarily isolated, but instead marked with saliency annotations whether the respective event is perceived to be in the foreground or background. Using the UrbanSound8k dataset, which is a subset of UrbanSound with slices of 4 s length, and constraining to foreground instances, could be a way to at least obtain events that are perceived dominant.

- The ESC-50 dataset (Piczak 2015b) comprises 2000 5 s-clips of 50 different classes across natural, human and domestic sounds, again, drawn from Freesound.org. While it has been attempted to extract sounds restricted to foreground events with limited background noise, events are not truly isolated. Also, events are not annotated with event on- and offset times.

- The Freesound Datasets (Fonseca et al. 2017) consist of audio samples from `Freesound.org`, organized in a hierarchy based on the AudioSet Ontology, with verified event labels. It is an ongoing project (albeit a very large one already) more than a completed dataset, aiming to increase the number of audio files with (through crowd-sourcing) verified labels. However, these annotations are weak labels, as they only provide information about the existence of a sound event throughout the file, but no information about when it occurs. As with UrbanSound, events are not necessarily isolated, but are labeled to be predominant or not. As of presented in Fonseca et al. (2017), 20 206 audio clips (92.5 h) are already labeled and verified with predominant events.

All in all, there is no other database available with strongly labeled, isolated sound events, of reasonable size. As a side note, all of these data sets were published after the creation of NIGENS.



(a) Alarm (b) Baby crying (c) Female speech (d) Crash

(e) Phone ringing (f) Female scream (g) Male speech (h) Engine

(i) Piano (j) Male scream (k) Footsteps (l) Fire

(m) Dog barking (n) Knocking (o) General

Figure 2.1: Class-average persistence spectra. Temporal "density" of sounds over frequency and power is displayed, showing power distribution over frequencies, but also sound structure. Density scales equal for all plots.

2.1.2 *NIGENS contents*

NIGENS consists of 1017 audio files of various lengths (between 1 s and 5 min), in total comprising 4 h:45 min:12 s of sound material. Mostly, sounds are provided with 32-bit precision and 44 100 Hz sampling rate. Files contain isolated sound events, that is, without superposition of ambient or other foreground sources. The contained sound types are described in the subsections below.

ALARM Diverse alarm sounds from old-fashioned fire bells to electronic beeps. Mostly high-pitched, discrete, sequential, very structured events; some continuous wailing. As observable in the confusion matrix (Fig. 2.2), alarm is the NIGENS class with most overlap to the other

classes – alarm sound segments exhibit high erroneous detection rates by other models, e. g., 37 % are detected by the scream detector and 35 % by the phone detector. There seem to be particular ambiguities with "crying babies", "ringing phone", "piano", and "scream". 49 files, 19.4 s average length.

BABY CRYING Crying babies. Mostly sequences of cries, also single sobs and squeals. Looking at Fig. 2.2, crying babies seem to be easily mistaken for alarms[3], and, logically, for (adult) screams. Recommendation: don't listen. Will break your heart. 40 files, 27.1 s average length.

CRASH Crashing structures, destructive impacts; noise-like, but sudden, bursting, singular sounds. Lots of energy across wide range of frequencies. 50 files, 9.8 s average length.

DOG BARKING Dogs barking, mostly several times in a row. Peak of energy around 1kHz, short, discrete events. 45 files, 11.6 s average length.

ENGINE Long continuous sounds of running engines of different kinds, idling or changing speed. Engine sound segments have a very high misclassification rate by the fire model (and vice versa). 39 files, 53.6 s average length.

FEMALE SCREAM Short single screams of females, high-pitched, peak of energy around 1.8kHz. 45 files, 3.7 s average length.

FEMALE SPEECH Females calmly speaking short English sentences. Female and male speech are the sound classes best detectable and least confusable, as is observable in Fig. 2.2. A small amount of female speech segments gets mistakenly detected as alarm sounds; on the other hand, the femaleSpeech model takes some piano segments for female speech, which is maybe more flattering. 100 files, 2.9 s average length.

FIRE Long continuous sounds of burning fires. Noise-like broadband sounds, but with higher energy in low frequencies. Fire sound segments have a very high misclassification rate by the engine model (and vice versa, see Fig. 2.2), and additionally often wrongly get detected as crashes (most probably because the distinct difference of fire being continuous long sound versus the short sudden nature of impact sounds gets lost in the segmented prediction). 51 files, 53.4 s average length.

3 Which kind of makes sense — one could say: mission accomplished.

FOOTSTEPS Diverse sounds of (individual) people walking, on all kinds of surfaces from wood to snow. Sequences of very short events. From Fig. 2.2, it is apparent that footsteps sound segments are hardly mistaken as other sounds. 42 files, 26.8 s average length.

KNOCKING Knocking on something, mostly doors. Sequences of very short events. Most energy in low bands. 40 files, 2.6 s average length.

MALE SCREAM Short single screams of males. Peak of energy around 1.2kHz. 31 files, 6.4 s average length.

MALE SPEECH Males calmly speaking short English sentences. Male and female speech are the sound classes best detectable and least confusable, as is observable in Fig. 2.2. 100 files, 2.6 s average length.

PHONE RINGING Mostly classic phones, sequences of long ringings – notable overlap with alarm. 40 files, 18.4 s average length.

PIANO Playing piano. Both individual notes as well as monophonic sequences as well as polyphonic pieces. Significant potential for confusion with alarms, referencing the results in Fig. 2.2. 42 files, 20.8 s average length.

GENERAL Anything outside the other classes. Discrete and continuous, single or sequential events, peaked or broadband. 303 files, 18.2 s average length.

Often, sound event detectors are trained to discriminate between a target class and all other target classes, sometimes added by broadband-like ambient noise. If testing is done the same way, this certainly produces highest performances. However, this approach lacks a real-world circumstance: there will always be a lot of sound events occurring that were not part of any target training class, many of them discrete and not noise-like. A. Mesaros et al. (2018) also identify this as a key difference between the DCASE 2016 SED synthetic and real audio tasks. To explicitly take this into account, and help better define target detector models against sounds different from other target classes, the `general` class was collected. This class contains sounds intended both as "disturbance" sound events (superposing) and as counterexamples to the target sound classes.

The `general` class is a pool of sound events *other* than the 14 distinguished target sound classes, containing as heterogeneous sounds as possible. For example, it includes nature sounds such as wind, rain,

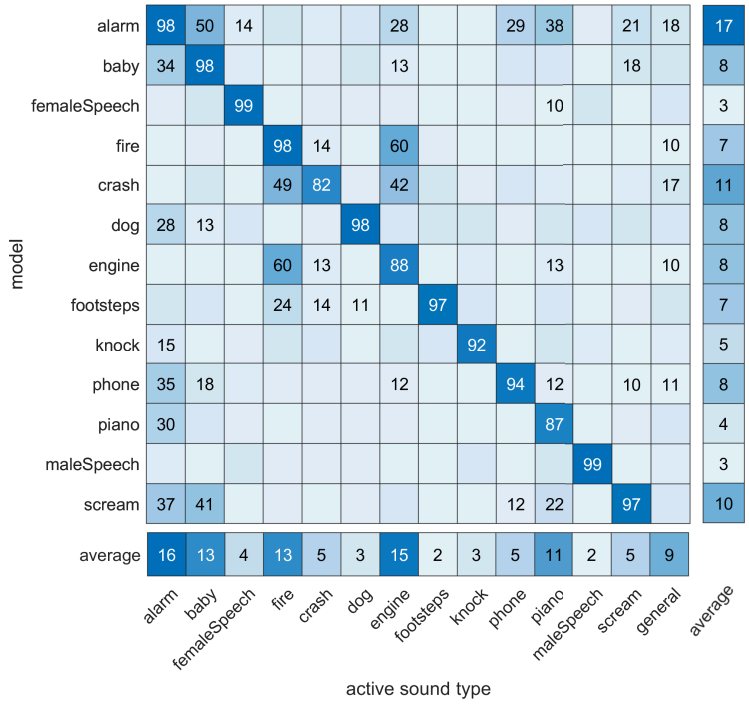


Figure 2.2: NIGENS confusion matrix, values are classification rates in percent in scenes with one source. Rows correspond to trained SED models (fullstream models described in Chapter 8), columns to types of active sound events. For better readability, only values of at least 10 % are displayed with label. The averages are without sensitivities (the values for matching sound and model type), that is, they are misclassification averages.

or animals, sounds from human-made environments such as honks, doors, or guns, as well as human sounds like coughs.

Fig. 2.1 shows *persistence spectra* for all classes, averaged over all sounds of each class. Persistence spectra display temporal "density" (rate of occurrence), over frequency and power. They are computed based on short-time Fast-Fourier Transform (FFT) spectrograms, but in contrast to them can be reasonably averaged. Compared to pure power spectra, they are able to also depict structure of sounds. Note, for instance, the similar structure of alarm and phone, versus the similar structures of engine and fire. The `general` (Fig. 2.10) class shows, as expected, a broadband, unstructured spectrum, since there are so many different types of sounds included.

Fig. 2.2, in advance of elaborating on the models producing these results[4], shows a classification rate confusion matrix of the NIGENS types and corresponding trained models. The values are percentages of (500 ms-) segments of binaural auditory one-source-scenes classified by the models as their corresponding type. On the diagonal are the sensitivities (positive classification rates, or detection rates), all other values are misclassifications: for example, the "femaleSpeech" model classified (correctly) 99 % of the actual femaleSpeech segments as femaleSpeech, but also (wrongly) 10 % of the "piano" segments. These classification rates, although of course specific to the models which produced them, can serve as an indicator of the overlap of the sound classes.

2.1.3 *Event on- and offset times*

In order to effectively train models that detect sound events of particular classes, sounds have been annotated by time stamps indicating perceptual on- and offsets of occurring sound events. Wave files are thus accompanied by an annotation (.txt) file that includes on- and offset times of that file's sound events. The `general` sounds do not come with on- and offset time annotations, since these files do not constitute any coherent sound class (to the contrary, by design) and are not intended to be positive examples for classifiers.

A specialized sound event labeling tool has been designed to enable efficient perceptual labeling by presenting aurally extracts of the sound files and letting the user label via a simple automated user interface.

In contrast to other SED data sets, only active sound were labeled as actual sound events, that is, times of silence are not part of sound events with this labeling. Positive labeling across "gaps" in sound events is more of a semantic-logical labeling (referring to a series of individual phone rings as "phone ringing", for example), but it can be assumed that this complicates training since the direct correlation of physical features and label gets lost.

2.1.4 *Attribution*

The largest part of the sounds was acquired from and kindly granted redistribution for research under above license by "StockMusic.com" (2014). Speech sounds were compiled from the GRID (Cooke et al. 2006) and TIMIT (Garofolo et al. 1993) copora. Several sounds were downloaded and included from `freesound.org` (Font et al. 2013) under attribution licenses, a list can be found in Trowitzsch, Taghia, et al. (2019a).

4 confer Chapter 8 about the respective models

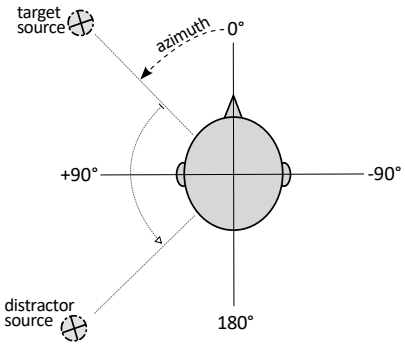


Figure 2.3: Coordinate system top view. Head and two sources are shown. Azimuth is always given with respect to the nose of the head (in counter-clockwise direction). Target and distractor sources in this example are located at 45°/135°.

2.2 GENERATION OF BINAURAL AUDITORY SCENES

Since the goal of this work was not classifying audio files (audio tagging), but performing sound event detection in auditory scenes, such scenes were needed to train and test the SED models.

Auditory refers to hearing, meaning processes in-ear and in-brain, and is thus related to perception of *acoustic* (sonic) signals. Acoustic and auditory scenes are thus directly connected: the acoustic scene consists of the sources producing sound at their locations, the environment and the listener, and the auditory scene – as the term is used here – is the product with respect to our hearing of it, i. e., the sonic mixture as it gets perceived through the ear and subsequent processing. In the most "basic" or low-level form, when using the term auditory scene, here the ear-signals (the waveform of the sonic pressure changes in the ears) generated by an acoustic scene are meant. In a more cognitive form, the term auditory scene refers to the interpretation of the acoustic scene produced by the auditory processing from the ear-signals. When writing "scene" without further specification, context makes clear whether acoustic or auditory scene is meant. The terms (acoustic) "scene" and "condition" sometimes are used interchangeably, referring to a particular acoustic "condition".

This section covers the definition of acoustic scenes and the generation of the (low-level) auditory scenes (that is, the ear-signals).

2.2.1 *Definition of scenes*

The definition of an acoustic/auditory scene (or "condition") throughout this work involves:

- The number of sources in the scene, varied between one and four.
- The type of sources. Mostly, point-sources are employed in this work, but in Chapter 5, there are also non-head-related sources (as from in-ear headphones, for example).
- The location of point-sources, *relative* to the listener's head. A restricted usage of the term "location" is employed, actually referring to the *azimuth* with respect to the nose of the head (counter-clockwise) regardless of the distance, and level with the head (disregarding elevation). See Fig. 2.3.
- The mean energy ratio between the sources, for ease of the term called signal-to-noise ratio (SNR), see Section 2.2.2.
- The sound types emitted by the sources. Two modes are differentiated: either sources emit any of the NIGENS sounds, or they emit only sounds of the "general" class.
- The room acoustics. Either free-field scenes (anechoic) are simulated, or rooms between office size to large concert halls, see Section 2.2.2.

In each scene, one source is declared the *target* source, the others *distractor* sources. The target source is the "signal" regarding the SNRs between sources.

Scene-instance denotes the combination of an original sound file with a particular scene, i. e. the application of this sound file as sound emitted by the target source defined in this scene. (Of course, scenes with more than one source (scenes with distractor sources) produce scene-instances created from more than one sound file.)

For all scenes, each of the NIGENS files (from the respective training or test split, see Section 3.3.1) was used once as target source sound (constituting a scene-instance). Sounds on the target source shorter than 30 s were looped, to create sufficient temporal context and reduce difference in amount of material between sound classes with long sounds, like fire, and sound classes with short sounds, like knocking. Distractor sources got "filled" with random sequences of their designated sounds until full overlap in length with the target source sound.

Figure 2.4: The Two!Ears robot. A Knowles Electronic Manikin for Acoustic Research (KEMAR) dummy head, rotatable on the torso, which is mounted on the moving platform. © Two!Ears

2.2.1.1 *Restrictions on test scenes*

A few arrangements have been made to facilitate a *systematic* evaluation of models' performances with respect to acoustic conditions:

1. The SNRs was set such that it was the SNR of the target to *each individual* distractor source.
2. Distractor sources never emitted sounds from the model target class.
3. Distractor sources never simultaneously emitted a sound from the same class as currently emitted by the target source.[5]

This serves the purpose of being able to characterize a scene with *one* SNR that describes the SNR of target sounds, these SNRs between scenes to be comparable, and hence the performances between scenes being evaluable with respect to SNR.

2.2.2 *Binaural scene synthesis*

To create the two-channel ear-signals, the "binaural simulator" of the Two!Ears system (Winter, Wierstorf, and Trowitzsch 2016, Ch. 2.2) was used. The binaural simulator produces ear-signals for point-sources by convolving (mono) audio signals with either an anechoic head-related

5 Note that target sources do not only emit sounds from the target class.

impulse response (HRIR) or a binaural room impulse response (BRIR)[6]. HRIRs and BRIRs are recorded impulse responses that characterize the acoustic "transfer" of an impulse sound signal from a source up to the microphone in the inner ear (of a dummy head, or human), in an open space (HRIR) or room (BRIR). Since the media transmitting sound (mostly air) are well-modeled as linear time-invariant (LTI)-systems, with such an impulse response, it is possible to simulate the transfer of *any* sound signal from that source to the inner ear, replicating the exact dynamics of the system on that particular transmission path. Confer books on signal processing related to audio for further information on this topic, e. g. Downey (2016).

For open-space scenes, an HRIR measured with a KEMAR head (see Fig. 2.4) was used, as described in Wierstorf et al. (2011). For reverberant scenes in rooms, several different BRIRs were used, referenced and described at the respective passages in the text (Chapter 6).

Non-head-related signals were employed by adding the mono sources to the ear-signals without any further processing.

Binaural simulation was conducted for each source separately, to allow control over the energy ratio of the sources *at the level of ear-signals*. To this end, the resulting ear-signals for each source were mixed at the scene-defined ratios referred to as SNR, even though there is no classic noise involved. For mean SNR calculation, ear-signal amplitudes were first squared and averaged over both binaural channels, and then averaged over the duration of sound activity, to not influence the SNR by periods of silence. That is, the SNRs determine the ratio of the average energy of the target source, while active, to the average energy of the distractor source, while active. This ratio is expressed in decibel (dB).

2.3 AUDITORY REPRESENTATIONS

The ear-signals generated through binaural simulation are representations of *physical* signals, namely of the sounds' waveforms in the inner ear. To extract meaning from them, it is reasonable to extract *auditory* representations (Brown and Cooke 1994), with specifically filtering into frequency bands being important, which of course also is a major step of human auditory processing. Confer e. g. Lyon (2017) and Wang and Brown (2006) for books covering human and machine auditory representations, including auditory physiology; Alías et al. (2016) provides a review on features without focusing on human auditory representations.

6 The naming convention admittedly is confusing. BRIRs are also HRIRs, because both use a head. But BRIRs are recorded in a room.

Audio signals were frequency-filtered using gammatone filters. For all representations, gammatone center frequencies were set to range from 80 Hz to 8 kHz linearly spaced on the (logarithmic) ERB scale (Glasberg and Moore 1990). Auditory-inspired features extracted from frequency filters on logarithmic scales are the standard by now and have been shown to perform better than linearly-scaled features (DCASE Community 2019; Huzaifah 2017; Annamaria Mesaros et al. 2019).

The simulated binaural signals (Section 2.2) for each scene-instance were processed by the Auditory Front-end (AFE) (May, Decorsière, et al. 2015) of the Two!Ears system (Two!Ears Team 2018) to obtain the auditory representations described in the following. Where not stated otherwise, default parameters were used as set by the AFE.

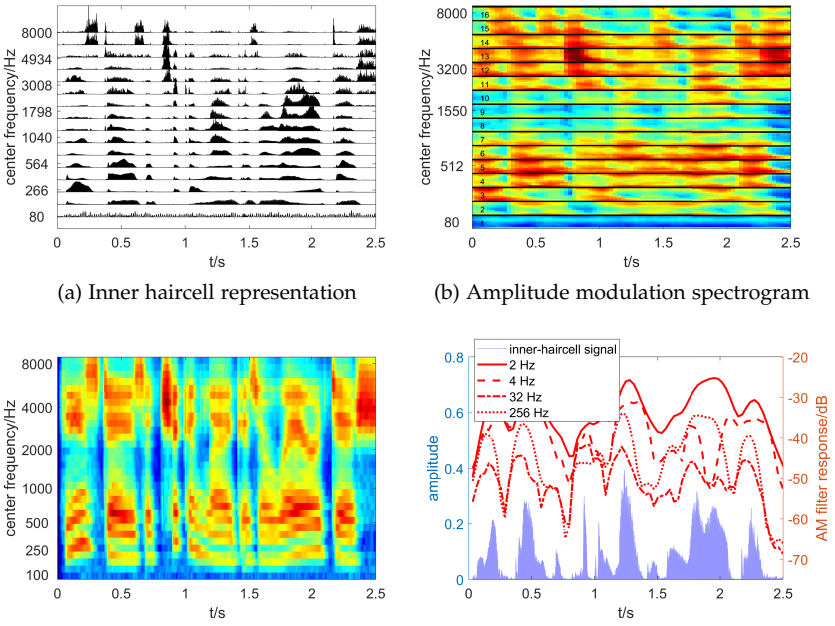
Section 10.1.4 discusses the selection of representations.

2.3.1 *Ratemaps*

Ratemaps are auditory spectrograms that resemble auditory nerve firing rates over time and frequency. Ratemaps are computed by first applying an inner hair cell transform (Lyon 2017, Ch. 18) to the gammatone-filtered signals, and then smoothing with a leaky integrator (typically with a time constant of 8 ms). The smoothed inner hair cell signal gets squared and averaged into overlapping frames of 20 ms length (10 ms shift). Ratemaps are used frequently in computational auditory analysis systems, often under the name of "Cochleagram", and are for example described in Brown and Cooke (1994), Patterson and Holdsworth (1996), or Wang and Brown (2006). Fig. 2.5c shows an example of a ratemap.

2.3.2 *Amplitude modulation spectrograms*

Amplitude modulation spectrograms represent envelope fluctuations, which have been shown to play an important role in the human auditory system (Luo et al. 2006; Shannon et al. 1995; Smith et al. 2002) and perform well in speech recognition systems (Mitra et al. 2014; Moritz et al. 2015). Logarithmically-spaced second-order bandpass modulation filters are used to extract these envelope characteristics from each channel of the gammatone-filtered inner hair cell signals in an auditory-inspired way (Ewert and Dau 2000; Tobias May and Torsten Dau 2014). The obtained amplitude modulation responses are, similar to the ratemaps, averaged into overlapping frames of 20 ms length with 10 ms shift. Fig. 2.5b shows an example of an amplitude modulation spectrogram together with inner haircell representation.



(a) Inner haircell representation

(b) Amplitude modulation spectrogram

(c) Ratemap

(d) AM filter responses

Figure 2.5: Inner haircell, amplitude modulation (16-channel) and ratemap (32-channel) representations of a speech sound. (d) shows, as an example, amplitude filter responses of frequency band 7 (center frequency 920 Hz) over the respective inner haircell representation.

2.3.3 *Spectral features*

What is call "Spectral features" in the following, in principle is no stand-alone auditory representation[7]. This term summarizes 14 different features condensing the spectral content of the ratemap for each time frame, i. e., these features are different statistics applied across frequency channels: Centroid, Spread, Brightness, High-frequency content, Spectral crest measure, Decrease, Entropy, Flatness, Irregularity, Kurtosis, Skewness, Roll-off, Flux, Variation. (Geiger et al. 2013; Jensen and Andersen 2003; Lerch 2012; Marchi et al. 2016; Misra et al. 2004)

7 But computed by the AFE, compared to the time-wise statistics explained in Section 3.1.1.1, which is why they are listed in this chapter.

2.3.4 *Interaural time and level differences*

With energies over frequencies and amplitude modulation carrying a great deal of information about the content of sound, interaural time-differences (ITDs) and interaural level-differences (ILDs) exhibit information particularly about the (head-relative) *location* – specifically the azimuth – of sound sources (May et al. 2013; Wightman and Kistler 1992). ITDs and ILDs are computed per time-frequency bin as in the ratemap (both are very frequency-dependent); the ITDs basically estimate the phase difference between left and right ear-signals, while ILDs compare the energy differences between left and right ear-signals.

2.3.5 *Other representations*

Among other representations that commonly are used in sound event detection, Mel-frequency cepstral coefficients (MFCCs) for a long time have been the most popular and successful (Serizel et al. 2018). However, they are sensitive to noisy conditions (Sharan and Moir 2016), and spectral energy features have gained a lot of room particular since the rising success of deep learning architectures, for instance Cakir et al. (2015b), Huzaifah (2017), and Takahashi et al. (2016) are only a few examples of models recently having produced better performance with deep neural network (DNNs) on logarithmic filter-bank energies compared to MFCCs. Throughout the Two!Ears project, MFCCs remained unused due to their lack of biological motivation.

Spectro-temporal Gabor filterbanks are used to capture not only temporal and spectral, but *joint* spectro-temporal modulations (Schädler et al. 2012; Schröder et al. 2015).

Spectrograms, usually created using the Fast Fourier transform, are very similar to ratemaps (Section 2.3.1), but less biologically founded. If phase information of spectrograms is not disregarded, it can be seen as the technical counterpart to ITDs (Section 2.3.4).

3

MODEL BUILDING AND EVALUATION

After the introduction of the data in the last chapter, here the general sound event detection (SED) model building and evaluation are covered.

Section 3.1 describes how the model input (features and labels) is constructed. In Section 3.2, model training aspects unspecific to particular algorithms are discussed, Section 3.3 elaborates on performance measurement of the models. The herein used model algorithms, logistic regression and deep neural networks, are introduced in Sections 3.4 and 3.5.

3.1 MODEL INPUT

To get from auditory representations to predictions about sound event activity by SED models, the auditory representations and scene annotations have to be worked up into actual model input. The following sections cover the production of features and labels for the models.

3.1.1 *Feature construction*

For segment-based (or block-based, the terms segment and block are used interchangeably) models, all representations obtained from the Auditory Front-end (AFE) (see Section 2.3) for SED (in general ratemaps, amplitude modulation spectrograms, and spectral features) were split into overlapping blocks. Only for the deep neural network (DNN) models used in Chapter 7, the full-length representations were used without any cutting. Depending on whether mean-channel or two-channel features were constructed (cf. Chapters 4 and 5), representations from left and right channel were then averaged, or they got concatenated.

Energy-based representations were compressed with a root function; ratemaps were additionally scaled for each block individually such that the median of all non-zero values was projected onto 0.5.

3.1.1.1 *Time-invariant features*

For model architectures that have no inherent mechanism for dealing with data varying over time – like logistic regression, support vector machines (SVMs), or Gaussian mixture models (GMMs) –, it can be very difficult to learn relationships from parts of time series, since data structures translate along the time dimension and usually will not be aligned with some standard point in time. There are model types that have inbuilt capabilities to model relationships over time, like hidden markov models (HMMs), or recurrent neural networks (RNNs), and model types that are able to build new feature representations including filters over time, like all sorts of DNNs, but particularly convolutional neural networks (CNNs). See Sections 3.4, 3.5.2 and 3.5.3 for descriptions of logistic regression, temporal convolutional network (TCN) and (long short-term memory (LSTM)) recurrent neural networks, and Chapter 7 for a comparison study on these different model types applied to the SED problem.

However, logistic regression models certainly are not able to learn relationships of data translating along time, the input features for the model need to be made *time-invariant*. To achieve this, a function can be applied onto the features mapping time series onto values without notion of time; common choices are statistical moments like the mean, variance, or also skewness and kurtosis. Here, L-statistics (L-mean, L-scale, L-skewness, L-kurtosis, Hosking (1990)) were chosen, which can be more robust than conventional statistics, particularly for higher moments and little data (David and Nagaraja 2003, Ch. 9).

To increase information extracted from the data, specifically information about the development of data over time, the first two discrete time derivatives ("deltas") of the auditory representations were computed prior to application of L-statistics.

With RM the ratemap, AM the amplitude modulation spectrogram, and SF the spectral features representations, the concatenated representations and their deltas are:

$$CR = \begin{pmatrix} RM \\ AM \\ SF \end{pmatrix}, \dot{CR} = CR_T - CR_{T-1}, \ddot{CR} = \dot{CR}_T - \dot{CR}_{T-1}, \tag{3.1}$$

and the time-independent feature vector F is defined as

$$F = \begin{pmatrix} f_{L-mean}(R) \\ f_{L-scale}(R) \\ f_{L-skewness}(R) \\ f_{L-kurtosis}(R) \end{pmatrix} \text{ with } R = \begin{pmatrix} CR \\ \dot{CR} \\ \ddot{CR} \end{pmatrix}. \tag{3.2}$$

The dimensionality d of feature vector F depends on the dimensions of the original auditory representations, the number of moments and deltas applied, but not on the length of the corresponding segment.

3.1.1.2 *Frame-based features*

In above-mentioned study with TCN and LSTM (Chapter 7), the DNN models are able (and supposed) to learn relationships including temporal development of data, thus the section above on constructing time-invariant features does not apply there. Instead, the models get frame-based features as input, that is, the "raw" auditory representations. The resulting feature matrix is of dimensionality dxT, where d depends on the dimensions of the original auditory representations, and T is the number of frames in the segment or the full scene-instance (in case of features for LSTM).

3.1.1.3 *Feature standardization*

As is common in many machine learning applications, feature centering and feature scaling was performed: before training models, each individual feature variable gets subtracted its mean (such that the mean becomes 0) and then divided by its variance (such that the variance becomes 1). Very importantly, the mean and variance are only calculated from the training set features (in cross-validation, from the training folds), and have to be saved to later apply the same procedures with the values from the training set onto test features. The goal of this procedure is to avoid a bias of models to prefer variables with larger variance. With L_1-regularized logistic regression, this furthermore makes betas from the model (see Section 3.4) comparable because they are on the same scale, and thereby enables feature analysis.

3.1.2 *Sample labels*

Supervised machine learning works by providing target values to the algorithms that provide ground truth about the produced output given the input features; the algorithm will optimize the model such that the defined loss between produced output and these target values is minimized. In classification, these target values are also called (class) labels. Labels (usually denoted y) and features (usually denoted x) together constitute a *sample* (x, y). For logistic regression, a feature vector/matrix is attached with *one* label. For LSTM and TCN, the feature matrix is attached with a label *vector* of length T.

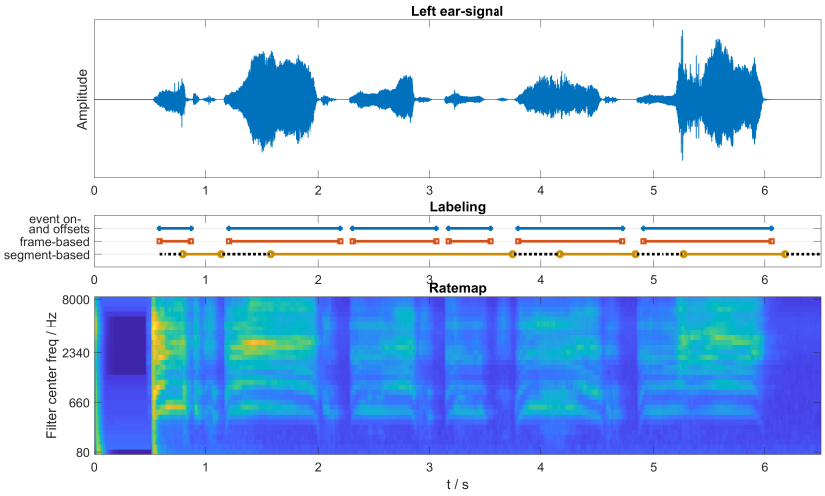


Figure 3.1: Example of representations and labelings of an auditory scene with a crying baby. Upper panel shows the left ear-signal waveform, lower panel the corresponding ratemap. The middle panel depicts the event on- and offset ground truth annotations, frame-based labeling, and 500 ms-segment-based labeling (black dots indicate "ambiguous" segments).

In the case of SED, labels are needed that express whether a sound event of particular type is present in the corresponding features (respective segment of auditory scene-instance), or not. The ground truth about the activity of sound events is available through the event on- and offset annotations of the original sound files (cf. Section 2.1.3); scene synthesis (Section 2.2) entailed generation of event on- and offset annotations for scene-instance scope. With knowledge about the actual event on- and offsets, the creation of labels about sound event presence is tantamount to reasonably defining *presence*, and in this work, two different definitions are used, explained in the following subsections.

3.1.2.1 *Segment-based labeling*

When using features including information of whole segments, it seems reasonable to also include information about the whole segment in the label. However, defining sound event presence in this case is not obvious. The simple approach of simply checking whether the sound event was active any time in the segment does not seem sensible, because this includes situations in which a sound event was active only for a few milliseconds at the border of the segment. While it is true then, that the sound event was active, (a) this definition is not a perceptual

one, because humans would (depending on the actual length) not detect the sound event in this situation, and (b), this definition would be counterproductive for model training, because it produces many samples with low correlation of features and label.

A sound event therefore is defined present, if

- either it overlapped at least 75 % of a segment,
- or, for events shorter than a segment, at least 75 % of the sound event was included in the segment.

Segments with sound event overlap of less than these 75 %, but more than 0 %, did neither get a "present" nor "not present" label, because they were considered ambiguous. To avoid confusion of the model algorithm and unclear performance measurement, these samples were excluded from training and testing.

Fig. 3.1 shows an example of segment-based labeling, next to the event annotation ground truth and frame-based labeling.

3.1.2.2 *Frame-based labeling*

As the segment-based labeling targets output with a certain lag and with a smoothing over time, additionally a frame-based, *instantaneous* labeling was produced, reflecting whether a sound event was active at the time of a frame, or not. Since frames are only of length 20 ms, the label of a frame was defined "present" if the sound event was active in it at any time, for any duration.

Usage of frame-based labels is obvious for LSTM and TCN, which use frame-based features also, and produce output for every time step (frame). For a logistic regression model, which produces one prediction for a whole segment of features, the label of the *last* frame was attached.

Fig. 3.1 shows an example of frame-based labeling, next to the event annotation ground truth and segment-based labeling.

3.2 MODEL TRAINING

This section covers general aspects, methods, and decisions regarding the building of the SED models.

3.2.1 *Binomial classification*

All sound detection models were trained as *binary* classifiers, detecting the *presence* (positive class) or *absence* (negative class) of the particular

sound type. This implies training in an one-vs-all scheme — discriminate one class (the target sound type) against all others.

Classic multinomial models (like k-nearest neighbors) are not an option to use in scenes with co-occurring sounds (polyphonic scenes), since they categorize instances into exactly one class at a time.

Using combined single-label models increases flexibility in applications and adding new models for other sound types. Also, it gives choice to more algorithms, since any classification algorithm can classify single-label binomial problems, but not all can produce multi-label output. Dealing with multi-label models can also be more difficult in training with respect to sub-sampling or weighting samples to adjust for unbalanced class distributions (which is necessary, see Section 3.2.3). On the upside, using multi-label models may, depending on the algorithm, decrease training time considerably, since only one model has to be trained.

Cakir et al. (2015a) investigated the difference of training combined single-label SED models versus training multi-label models (DNNs). They found a minimal reduction in performance of the combined single-label models (without statement of statistical significance of this reduction); and concluded that the correlation structure between labels does not add relevant information to the model training.

With logistic regression models (Section 3.4), true multi-label modeling is not possible. While GLMNET allows producing multi-label Gaussian regression models, these actually are combined single-label models with individual model parameters per class[1]. This is in contrast to DNN models (Section 3.5), which can use a large combined network with only smaller individual sub-networks in the last hidden layers and output layers providing an effective multi-label approach.

3.2.2 *Single- and multi-conditional training*

For building sound detection models, two schemes of training were employed in regard to what acoustic scenes were used. (Confer Section 2.2 for definition and explanation of acoustic scenes.)

With the term *single-conditional (sc) training*, training on data taken from one acoustic scene (condition) only is referred, i. e. a defined number of sources, at specific azimuths, with specific signal-to-noise ratios (SNRs), etc. All scene-instances of the chosen scene were then used for training. See Chapters 4 and 6 for the analysis of performance of single-conditional models.

1 One possibly interesting use-case is the application of *grouped* lasso, i. e. the coefficients for the multiple labels (multiple models) are penalized together across models.

The term *multi-conditional (mc) training* is used when training models on data taken across more than one scene (typically many, e. g. 80). The idea of multi-conditional training is to incorporate model *invariance* (or: *robustness*) against deviation of auditory testing/application conditions from training conditions. Building models able to be *invariant towards changes in the data unrelated* to the categorization in question and at the same time *sensitive towards variance in the data related* to the classification targeted really is at the core of machine learning application[2]. In principle, there exist two approaches to achieving such robustness:

1. Explicit engineering of invariance. The obvious necessity is prior knowledge about the irrelevant variance *and* knowledge about how to transform data to be invariant. An example for this approach would be de-mixing of sound mixtures to (hopefully) retain individual sounds such that a system trained on isolated sounds can be reasonably applied.

2. *Learning* invariance in a data-driven way. That is: the data presented to the training algorithm has to include the irrelevant variance, such that the algorithm can find out about it by itself — *if* it can. The obvious necessity here is an algorithm able to extract the irrelevance in question, often, this will imply more powerful/complex models. In connection with the example before, the system would be trained from the start on sound mixtures, and identifying target sounds in overlapped mixtures would be left to the model.

The latter is the multi-conditional approach followed and proposed here; the example given for explicit invariance engineering exemplary illustrates the motivation: without many constraints (on the number of channels, prior knowledge about occurring sounds, ...), de-mixing of sound mixtures ranges from difficult to hardly possible. That is, the author of this thesis doesn't see the necessary knowledge and techniques about how to transform binaural sound data to be invariant for example in SNR, location of sources, reverberation, or number of co-occurring sources. See Chapters 5 to 8 for the studies on performance and robustness of multi-conditional models.

3.2.3 *Sample subsets and cost*

A typical problem in machine learning, also in SED, is *data imbalance*, which refers to imbalance among the classes (or sub-classes) in the amount of data used for training. For one thing, classes often naturally

2 LeCun et al. (2015) call this "selectivity–invariance dilemma".

are of different size: sounds from the class "fire", for example, are on average much longer than sounds from the class "knocking". For another thing, data imbalance is especially likely to occur in multi-class settings that are solved through binomial classification (cf. Section 3.2.1), since the one-vs-all approach necessarily yields more negative than positive samples. The problem about data imbalance is, when treated naively (or not treated), bias in the model training towards optimizing the correctness of output of samples belonging to the dominant class(es), and neglecting the others. If, for instance, positive samples amount to very small percentages (they do here – about 2 % to 10 %), training algorithms can achieve high overall accuracy by producing models that always predict negative, however, nothing would have been "learned" and decisions be un-informed.

The following sample subsets were defined: "positive" samples, i. e., samples with label $+1$, and "negative" samples, i. e., samples with label -1. These were further sub-divided into samples from mixtures with one, two, three, or four sources active, respectively. For samples segregated from the ear-signals mixture, further distinction between segregated negative samples from mixtures with a positive active, and segregated negative samples from mixtures without positive active was needed, confer Chapter 8. Note that all these samples types are individual to each sound class, or rather, the designation of actual samples into subsets is individual to each sound class.

When full training data anyway could not be utilized because of algorithmic or computational restrictions – as was the case with GLMNET, which was not able to cope with more than 2 GB input –, first off *sub-sampling* in a way to already alleviate the data imbalance was performed. That is, higher proportions from the above mentioned subsets with lower percentage were sub-sampled, and vice versa; equal amounts of samples were drawn from each scene-instance.

After sub-sampling, samples got *weighted* according to their sample type and scene-instance length in order to achieve equal cost in model optimization for each of the types and scene-instances. The weight w_i of sample (x_i, y_i) was set to

$$w_i = 1/(N_{L(i),nas(i)} \cdot N_{si(i)}) \tag{3.3}$$

where N_S denotes the number of samples in subset S, indicated by $L(i)$, the label y_i of the sample, $nas(i)$, the number of sources in the mixture of the sample, and $si(i)$, the scene-instance the sample originated from.

3.3 EVALUATING MODELS

Goal of successful machine learning is building models able to *generalize* on the data. Generalization is the ability to (successfully) predict on data *unseen* in the model building process, and thus the opposite to memorization of the training data. It implies finding and extracting rules/patterns/structure in the training data that still hold for new data. If an algorithm is not able to learn training data structure, this is called *underfitting*. If an algorithm is able to learn the training data structure, but with the built model is not able to efficiently predict unseen (test or validation) data, it basically memorized training data instead of generalizing, which is called *overfitting*.

A necessity in building well-generalizing models is to actually *measure* the generalization performance, such that in the training process, these models can be found and selected. "Model selection" is part of many machine learning algorithms, particularly always when algorithms include hyper-parameter optimization, as is the case in this work with L_1-regularized logistic regression (λ parameter, see Section 3.4) and DNNs (number of layers, number of neurons, and many more, see Section 3.5).

Of course, even without hyper-parameter optimization and the need for measuring generalization performance as part of the training process, it will usually be necessary or at least desirable to evaluate the generalization capabilities of built models to report and analyze them.

3.3.1 *Training sets, test sets, and cross-validation*

To measure generalization of final models, it is necessary to keep parts of the available data for measurement and not use it for training; this part of the data is then called test set. It is important to not incorporate this data in any way during training and model selection, otherwise the performance estimate drawn from it is not a valid estimate of generalization. For estimating generalization in model selection during training, another part of the data not included in the test set and not to be used during model parameter optimization, needs to be retained. This part is called validation set.

Cross-validation

However, very commonly (and also in this work), there is a lack of data and cutting away large parts from training leads to underfitting. As there is no way to avoid holding out a test set, one has to balance between necessary training data to generate good models, and desirable

predictive power of the test data. To not additionally loose the validation data for training, often *cross-validation* is employed, and so it is done here as well.

For performing cross-validation, training data is split into k folds. Then $k-1$ models are trained on all fold combinations that leave one fold out, and each of those models is tested on the left-out fold (then the validation fold). The mean of the k measured performances then is an estimate of the average generalization performance of models built with the applied algorithm on an independent test set from the same data distribution. In hyper-parameter optimization, the cross-validated performances can then be used to select a model; and the final model sequentially be trained on the full training set. The "cost" of cross-validation is mainly increased computational effort. See Hastie et al. (2009, Ch. 7) for further information on model assessment.

Splitting the data

It is important not only that the training algorithm does not "touch" test or validation data, but equally important that test and validation data are independent from training data. This implies splitting of data at the right level in the sample generation process, and in this case, it was necessary to make sure that samples from the training set and samples from the test set, or samples from cross-validation folds used for training and for testing, never contained parts from the *same sound file*. Since this included sounds both emitted from target and distractor sources, mixtures in scenes with more than one source also had to be made from only sounds in the same training/validation/test set. To make this computationally manageable, from the start all sound files got divided into 8 stratified[3] folds and mixtures only generated from sounds within each fold. This way recombination of the folds into training, cross-validation, and test sets was possible; usually two folds were used for final testing, and 6-fold cross-validation was performed on the other six folds used for training the models.

In the employed approach to SED, there are two factors influencing the amount of training data: (1) the number of original sound files, that is, the number of unique examples for sound events of each class, and (2) the number of scenes applied onto them, that is, the number of scene-instances generated. However, it should be clear that the number of original sound event examples is the decisive factor regarding the potential for capturing the structure of the different sound event classes, and that the number of scenes applied can only help exploiting this potential through delivering different versions (particularly: different

3 replicating the files' distribution of sound classes

mixtures) of the sound event examples[4]. The number of sound files per class being the crucial variables relativized the size of the produced training data, and was the rationale for the choice of training versus test set sizes (75 % versus 25 % of the sound files) and utilizing cross-validation for model selection.

3.3.2 *Performance measures*

Measuring the performance of models presumes choice of a suitable measure. The most obvious measure is *accuracy*, percentage of correctly classified samples. However, as with not treating data imbalance in training (see Section 3.2.3), using accuracy on imbalanced testing data has performance dominated by the larger sample class (here, the negative class). It would thus not be very informative here.

3.3.2.1 *Balanced Accuracy*

One solution to this problem is equivalent to sample weighting in training: using weighted accuracy, and here, balanced accuracy (BAC). BAC is defined as the arithmetic mean of sensitivity (positive class accuracy, also called detection rate) and specificity (negative class accuracy). With TP, TN, FP, and FN the number of true positives, true negatives, false positives, and false negatives, balanced accuracy calculates as:

$$BAC = \frac{1}{2} \cdot \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) = \frac{1}{2} \cdot (SENS + SPEC) . \quad (3.4)$$

3.3.2.2 *BAC2*

For performance measurement in model selection during training, a modified version of balanced accuracy was constructed that penalizes differences between sensitivity and specificity:

$$BAC2 = 1 - \sqrt{((1 - SENS)^2 + (1 - SPEC)^2)/2)}. \quad (3.5)$$

This performance measure is upper bounded by the standard balanced accuracy. If sensitivity and specificity are exactly equal, the value of $BAC2$ is equal to the balanced accuracy. If classification is perfect, i. e. if both sensitivity and specificity are equal to 1, then $BAC2 = 1$. If sensitivity and specificity are both 0.5, then $BAC2 = 0.5$. Any difference between sensitivity and specificity is penalized. The motivation for this

4 Of course, here, this is rather a positive side-effect than the goal of applying many scenes, because the goal is to produce models robust to changes in acoustic conditions through the multi-conditional training (cf. Section 3.2.2).

is that in the absence of information about the true distribution of samples (and cost of errors) in later application of the trained models, a classifier would be preferred that shows a specificity of 0.8 and sensitivity of 0.8 ($BAC2 = 0.8$) over one that exhibits a specificity of 0.6 and a sensitivity of 1.0 ($BAC2 = 0.72$), for example. Classifiers that assign all data to a single category, such as the larger class, yield only a performance value of $BAC2 = 1 - 1/\sqrt{2} = 0.29$.

3.3.2.3 *F-score*

In SED, F-score (also called F_1 measure) is the most commonly used performance measure, calculated as follows:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} = \frac{2 \cdot PREC \cdot SENS}{PREC + SENS} \tag{3.6}$$

$$\text{with } PREC = \frac{TP}{TP + FP}. \tag{3.7}$$

Since the F-score does not take into account the true negatives, it depends on the data distribution. This gets transparent by rearranging the formula through a few steps such that it shows its relation to sensitivity and specificity:

$$F_1 = \frac{2 \cdot SENS}{1 + SENS + r_{NP} \cdot FPR} \tag{3.8}$$

$$\text{where } r_{NP} := \frac{N}{P} = \frac{TN + FP}{TP + FN} \tag{3.9}$$

$$\text{and } FPR = 1 - SPEC\,. \tag{3.10}$$

The entailment of r_{NP} shows the F-score's dependence on the ratio of negative to positive samples (because neither sensitivity nor specificity or false positive rate depend on the amounts of negatives or positives). F_1 increases with decreasing r_{NP}.

Because this dependency makes F-scores hard to compare and less informative with respect to how "informed" a classifier's decision is, the F-score was opted against, and instead chosen the BAC. See Section 10.2 for the extended rationale about this decision and a discussion on this topic.

3.3.3 *Class-average performances*

Mostly in this work, *class-average* performances are presented, compared, and analyzed, because the goal was developing methodology for robust SED model building in general, that is, not specific to a

particular sound class. There basically exist two widespread inter-class performance averaging methods: *micro-* and *macro*-averaging.

The first builds averages by summing true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) across all classes and then calculating the performance measure of choice. This reflects class distributions, such that large classes (the fire sound class, for instance, since fire sound events mostly are long) have stronger influence on the TPs and FNs, and small classes (the female scream sound class, for instance) have stronger influence on the TNs and FPs[5]. This can be desired, but consequentially makes performance numbers not representatives of all sound event detectors equally, but instead representatives of the conglomerate of all detectors together on the very actual data set.

Thus here, always the macro-average is used, which is built by first calculating class-wise performance measures like the BAC, and then averaging over these. This way, each class-specific performance obtains the same weight in the class-average performance, which is desirable in order to present performance numbers representing average capability of the sound event detector models.

3.4 MODEL BUILDING WITH LINEAR LOGISTIC REGRESSION

The majority of model building in this work was conducted using L_1-regularized linear logistic regression models (e. g. Hastie et al. (2009, Ch. 3.4)). Logistic regression is a simple yet very widespread method, with the intention of modeling the probabilities of data points belonging to particular classes. As elaborated on in Section 3.2.1, models are restricted to the binary two-class cases. Logistic regression is capable only of linear separation between classes; nonlinear dependencies can only be modeled if feature construction includes projecting these relationships into linear space. L_1-regularized Logistic regression's strengths, however, are high computation speed[6] both in training and testing, interpretability of the models and of the models' outputs (probabilities), and great efficiency in dealing with very high dimensional feature spaces, even for situations where the number of available feature vectors is lower than their dimensionality.

L_1 regularization applied to linear regression models is also called the Least Absolute Shrinkage and Selection Operator (LASSO) and was introduced in Tibshirani (1996). The L_1 penalty leads to sparse models by translating regression coefficients towards zero and truncating at

5 This is true if all models predict on all samples, because then classes with higher share of positives also have lower share of negatives, and vice versa.

6 magnitudes higher than that of most (if not all) nonlinear methods, particularly DNNs

zero, making Lasso a classification method with an embedded feature selection procedure. L_1-regularization provides high efficiency in preventing overfitting, which is theoretically underlined in Ng (2004). The equally popular (although maybe more in the context of linear Gaussian regression) L_2-regularization shrinks coefficients proportionally and pushes correlated predictors towards each other, but does not push onto and truncate at zero, hence does not select a subset of features.

Logistic regression in practice often will perform about similar to other popular linear methods, like the linear support vector machine (Hastie et al. 2009, Ch. 12) and linear discriminant analysis (Hastie et al. 2009, Ch. 4.3). Compared to the first, logistic regression has the advantage of providing probabilistic output, compared to the latter, it has the advantage of not assuming Gaussian distribution of data[7]. Also, most other models including the two mentioned ones are not as interpretable, whereas logistic regression allows added insight into the dependencies of classification on the individual variables.

The "GLMNET" package (Friedman et al. 2010; Qian et al. 2013) was utilized for training these models. Through cyclical coordinate descent, GLMNET – in the binomial mode – minimizes the following objective function (with a quadratic approximation to the binomial log-likelihood in there):

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} -\frac{1}{N}\sum_{i=1}^{N} w_i \cdot l(b, x_i, y_i) + \lambda\left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1\right], \quad \text{with} \quad l(b, x, y) = yb^T x - \log(1 + e^{b^T x}). \tag{3.11}$$

With the model trained, the probability of any x to belong to the "positive" class ($G = 1$) is calculated as

$$Pr(G = 1|X = x) = \frac{e^{b^T x}}{1 + e^{b^T x}}. \tag{3.12}$$

In these equations, (x, y) are the N data points and their respective labels, β are the model coefficients and β_0 is the intercept; x includes an extra leading 1 for multiplication with the intercept in the combined coefficients variable $b = \{\beta_0, \beta\}$. α controls the regularization, which can be anything between sole L_1 ($\alpha = 1$, LASSO) and pure L_2 ($\alpha = 0$, ridge); λ determines the strength of the overall penalty. Because strict L_1-regularization can lead to numerical instabilities, α was actually set to 0.99.

For adjusting the regularization parameter λ, k-fold stratified cross-validation was performed (see Section 3.3.1) on the training set. The

7 Of course, this is only an advantage if the data is not distributed Gaussian.

value with the best cross-validation performance was chosen and used to train the model on the full training set.

Importantly, GLMNET supports usage of *observation weights* (w_i), which allow for sample-specific cost inflicted on the objective function. This is particularly important in cases where class distributions are highly imbalanced (as is the case in this work), but prediction quality for the different classes shall be equally important. It thus enables optimizing for balanced accuracy (Section 3.3.2.1), and weighing cost according to additional intra-class rules, as described in Section 3.2.3.

3.5 MODEL BUILDING WITH DEEP NEURAL NETWORKS

Although "conventional" methods like the LASSO can still achieve good performance (as will be shown in the next chapters), state-of-the-art nowadays is *deep learning* (LeCun et al. 2015) with deep neural network (DNN) architectures. This is due mainly to two factors: (i) DNNs are powerful representation learners, i. e., they can learn complex features which often are difficult to hand-craft as efficiently, and (ii) they have inherent capabilities to access data *context*, i. e., exploit locality in features along dimensions such as time or space[8]. Two architectures have emerged as particularly successful with respect to the second point: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Two variants are considered in this work:

- Long short-term memory (LSTM) (Greff et al. 2017; Hochreiter and Schmidhuber 1997), a very popular and successful RNN model designed to learn temporal relationships over potentially long durations. LSTM has solved problems hindering effective usage of RNNs before, like the vanishing gradient problem.
- Temporal convolutional network (TCN) (Bai et al. 2018), a recent CNN architecture based on work in Kalchbrenner et al. (2016), Long et al. (2015), and Oord et al. (2016). TCN provides an efficient feed-forward alternative to LSTM with supposedly comparable sequence modeling capability.

LSTM are commonly assumed to be the natural choice for temporal sequence processing, because their access to temporal information in principle is unrestricted and can be learned from data. CNN, on the other hand, are more efficiently parallelizable for GPU utilization in training, and excel at spatial feature extraction.

8 (i) and (ii) are not necessarily separate points, actually they go together very well.

3.5.1 *Shared training methodology*

Models were coded and trained in Python with Keras (Chollet et al. 2015) with TensorFlow back-end, utilizing the cuDNN (Chetlur et al. 2014) library for fast GPU training.

Output layers always consisted of neurons (one per sound type) with logistic sigmoid activation function, in each frame producing probabilities about the presence of sound events. Weighted cross entropy was used as loss function for model optimization:

$$-w_i \cdot (y_{i,m} \log(p(x_i)) + (1 - y_{i,m}) \log(1 - p(x_i))) \tag{3.13}$$

with y and $p(x)$ being the target and predicted output for features x, w the sample weight, i referring to the sample index, and m the sound type/output neuron index.

Sample weights were set as described in Section 3.2.3, such that balanced accuracy was optimized and scene-instances of different length and scenes with different number of sources are equally important.

The minimization of the loss with respect to network model parameters (connection weights) was performed using the Adam optimizer (Kingma and Ba 2014). Cross-validation as described in Section 3.3.1 was performed for hyperparameter optimization, conducted as random search (Bergstra and Bengio 2012).

Model training was terminated once the balanced accuracy on the respective validation set did not improve for five consecutive epochs. The number of training epochs for the final model got set to the median of the validation sets best epoch numbers.

3.5.2 *Temporal convolutional networks*

The TCN differs from a conventional convolutional neural network (e. g. Espi et al. (2015) and Piczak (2015a) for sound event classification) in four ways:

1. It is based on a one-dimensional (time) fully-convolutional architecture Long et al. 2015, implicating an output sequence of the same length as the input sequence,
2. the convolutions are causal (future points in time cannot be accessed),
3. convolutions are dilated to create a receptive field that exponentially increases with the number of layers (Oord et al. 2016),
4. instead of stacked convolutional layers, a stack of residual blocks compose the network (Kalchbrenner et al. 2016).

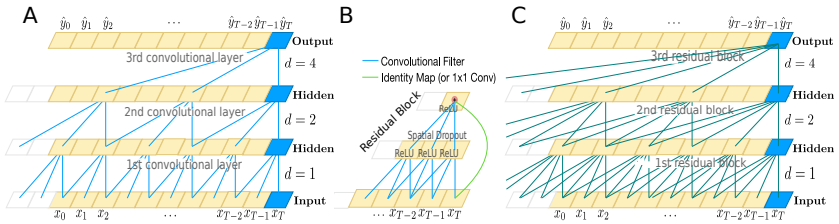


Figure 3.2: Temporal convolutional network architecture. **A**: Dilated causal convolution with dilation factor $d = 2^0, 2^1, 2^2$, and filter size 3. **B**: Residual block with convolution filter size 3 and dilation factor 1 mapping from input x_t to first hidden layer. Rectified linear unit (ReLU) activations for both convolutional layers and spatial dropout between these layers are indicated. **C**: Temporal convolutional network with $M = 3$ residual blocks, each with dilation factor 1, and convolution filter size $K = 3$ (i.e., $T_{3,3} = 17$). Modified from Bai et al. (2018) with permission.

Fig. 3.2 A depicts the ideas of points 1-3.

Each residual block consists of two consecutive convolutions with the same (internal) dilation factor, and after each follows a ReLU as activation function. An identity map is added to the output of the second convolution (see Fig. 3.2 B). This allows learning of modifications of the identity mapping rather than the entire transformation, which has been shown to ease deep learning (Bai et al. 2018; K. He et al. 2016). Spatial dropout – zeroing out whole feature maps (Tompson et al. 2015) – within each residual block is employed for regularization. Although part of the original model formulation, weight normalization (Salimans and Kingma 2016) in this work was omitted, since it did not accelerate optimization in the conducted experiments.

In Fig. 3.2 C, a temporal convolutional network with a stack of 3 residual blocks and filter size 3 is shown; each residual block has internal dilation factor 1 (the receptive field of each residual block hence is of length 5).

TRAINING The following hyperparameters had to be optimized: the convolution filter size K; the batch size; the initial learning rate of the Adam optimizer; the maximum gradient norm for clipping; the number of feature maps; and the (spatial) dropout rate. Depending on the required temporal context length T (in frames), the number of residual blocks M was chosen as smallest upper bound for $T \leq T_{K,M} = (K-1) \cdot 2^M + 1$, where $T_{K,M}$ is the resulting effective receptive field/history length.

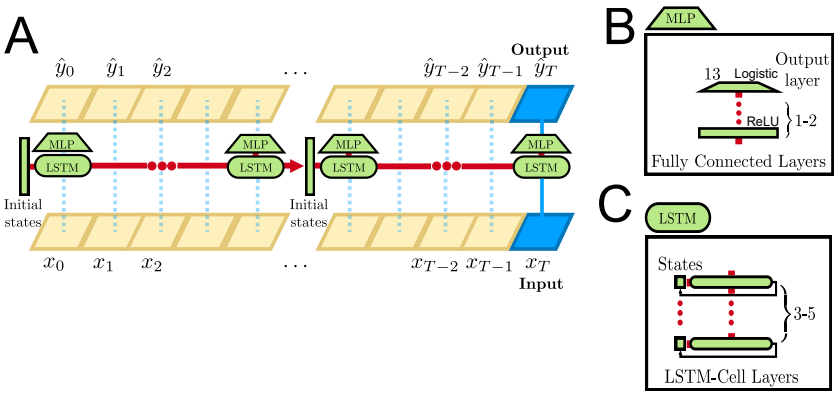


Figure 3.3: LSTM + DNN architecture (LDNN). **A**: LDNN consists of an LSTM with a multilayer perceptron (MLP) stacked on top. LDNN process input in a recurrent manner producing output each step in time. Input sequences may be spread across multiple batches which truncates the gradient backward pass in the backpropagation but the internal states of the LSTM-cells are propagated across batch boundaries. **B**: The MLP consists of 1-2 fully connected (FC) layers with ReLU activation function followed by an output layer with logistic sigmoid as activation function. **C**: The LSTM part consists of 3-5 LSTM-Cell layers where every layer processes its input by an individual internal state.

The network weights were initialized by independently drawing from an appropriate uniform distribution (Glorot and Bengio 2010), biases were initially set to zero.

When scene-instances were longer than chosen batch lengths, remainders of scene-instances were copied to a later batch (with overlap in the size of the receptive field minus one). Consequentially, the first $T_{K,M} - 1$ frames of the later batch got marked with an ignore mask in order to not double measure the respective loss.

3.5.3 *LDNN*

LDNNs (Sainath and Li 2016) consist of LSTM layers for processing of the temporal input, with FC layers stacked on top for transformation of the extracted intermediate representation to the output. LSTM are specifically designed for processing sequences containing long-term dependencies (Hochreiter and Schmidhuber 1997), by maintaining an internal state (modified sequentially by the input) from which the necessary information will be extracted. Fig. 3.3 depicts the architecture.

TRAINING Training batches get composed of many stacked scene-instance feature sequences, but scene-instances are of different lengths. To avoid (i) padding with zeros at the end and (ii) assembling batches from scene-instances with the same length, which would e. g. introduce dependencies on sound type in sequence sampling, scene-instances were sampled and stacked randomly, but were then split into parts to create batches with the same finite length. This may lead to distribution of scene-instances on multiple batches, resulting in truncated backpropagation at the batch boundaries. In the forward pass, LSTM cell internal states can still be propagated across batch boundaries. The length of a batch – the range of input frames in between batch boundaries – then is the effectively accessible temporal context of the LDNN, as the back-propagated feedback can only lead to adjusting weights to incorporate temporal dependencies within that range.

Dropout regularization (Srivastava et al. 2014) was applied to the activations of the FC layers (except the output layer) and to the activations of the hidden states of the LSTM layers. Furthermore, for regularization of the internal cell states of the LSTM with a comparable effect to Recurrent Dropout (Gal and Ghahramani 2016) (which was not available in the cuDNN implementation used), a variant of it was developed, setting the recurrent weights connected to the dropped neurons to zero. This resembles DropConnect (Wan et al. 2013), with the difference that not the weights are selected randomly to be dropped, but the neurons they are connected to (matching the original Dropout formulation). A similar approach to this one is used in Merity et al. (2017).

Weights connecting the input with the LSTM cell states and weights of the FC layers were initialized by independently drawing from an appropriate uniform distribution (Glorot and Bengio 2010), whereas the weights connecting the cell states with the different gates and to itself were initialized as random orthogonal matrices (Saxe et al. 2013). Biases of the forget gates were initially set to one and all others to zero (Jozefowicz et al. 2015). These initialization choices follow the Keras defaults.

The following hyperparameters had to be optimized: the number of LSTM layers; the number of FC layers; the number of total neurons; the fraction of neurons contained in the LSTM layers; the maximum gradient norm for clipping; the Dropout regularization strengths.

FC layers were set to use ReLU activation functions.

Part II

ROBUST SOUND EVENT DETECTION

Robustness is key in the real world. Hard training makes tough models.

4

SINGLE-CONDITIONAL MODELS

This chapter is based on Ivo Trowitzsch et al. 2017. "Robust Detection of Environmental Sounds in Binaural Auditory Scenes." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (6): 1344–1356.

The goal of this work was investigating the behavior of sound event detection (SED) models under varying auditory conditions, and developing methods for increasing the robustness of models with respect to these conditions.

To begin with, it was necessary to establish what performances can be achieved in optimal situations, and what performance losses are to be expected when deviating from optimal situations. This chapter thus provides a systematic assessment of performances of SED models

1. trained on single-source auditory scenes ("clean data", "monophonic models"), tested on the same scenes; as baseline,
2. trained on single-source scenes, but tested on two-source scenes; to evaluate the impact superimposed sounds have on models naively trained on "clean" data,
3. trained on various two-source scenes ("polyphonic models"), tested on the same scenes; to rate the performance achievable when models are trained under the same auditory condition as in testing,
4. trained on various two-source scenes, but tested on *other* two-source scenes; to quantify how deviations from training conditions affect performance.

To address this question, *single-conditional* models were built and analyzed, that is, models which are trained on data from exactly one scene (as described in Section 3.2.2).

In the work described in this chapter, the effects of different angular source configurations and signal-to-noise ratios (SNRs)[1] were studied.

1 Chapter 6 covers the effects rooms (i. e. among other factors, reverberation) impose on SED models.

Since binaural data was used, two different approaches to constructing features for the models were investigated: averaging the two channels' representations, or building features from each channel individually and concatenate them.

Section 4.1 describes any particularities or amendments to Chapters 2 and 3 with respect to data and methods concerning this chapter, Section 4.2 then elaborates on results of situations 1 to 3 from the list above, and Section 4.3 presents the analysis of generalization across conditions, situation 4.

4.1 DATA AND METHODS

Four aspects are to be described additionally to the general data and methods description of the chapters before: the actual auditory scenes used in this study, the two different feature sets, the model training specifics, and the two evaluation modes.

4.1.1 *Auditory scenes*

The original sound data for synthesizing auditory scenes was taken from the NIGENS database, described in Section 2.1. For methodology and terms regarding the scene rendering, confer Section 2.2.

Two sets of scenes were defined:

1. Five single-source scenes with a point source at azimuths {0°, 22.5°, 45°, 67.5°, 90°}. These rendered scene-instances in the following also are referred to as *clean* sounds or data, because there was no simultaneous disturbance.

2. Scenes containing two point sources emitting superimposed sounds. The target source emitted sounds from all classes, including "general", the distractor source emitted sounds only from the general class. Target and distractor sound sources were located at the following combinations of azimuths[2]:
   - 0°/{0°, 45°, 90°, 180°}
   - 22.5°/{−22.5°, −67.5°, 112.5°, −157.7°}
   - 45°/{0°, −45°, 135°, −135°}
   - 67.5°/{112.5°, −112.5°}
   - 90°/{180°, 0°, −90°}

2 listed as: target source azimuth / *set* of azimuths for distractor source

Each of these azimuth combinations got combined with four different SNRs (10 dB, 0 dB, −10 dB, −20 dB). Together, this amounted to $17 \times 4 = 68$ scenes in this set.

In the analyses below, sometimes the two sets are referred to together; the "clean" sounds (set 1) then got designated an SNR of inf dB.

Set 2 was compiled to provide high angular resolution, both with respect to the target source azimuths, as well as to the number of azimuth combinations with each target source. As the number of combinations needed to be limited because of the amount of tests implied by each (see Section 4.1.4), this came at the cost of coverage: the target source was located between 0° and 90° in all configurations. Fig. 5.1b shows a diagram of the azimuth scene configurations in the set.

4.1.2 *Feature sets*

Features for the models were constructed as described in Section 3.1.1, specifically Section 3.1.1.1. As base auditory representations, ratemaps, spectral features, and amplitude modulation spectrograms were used, described in Section 2.3, plus onset strengths. The onset strength representation is another derivation of the ratemap, measured as the frame-based increase in logarithmically-scaled energy of the ratemap (Klapuri 1999). The four representations were split into overlapping blocks of 500 ms length, with a shift of 167 ms.

Since SED models were built from binaural signals, and the described investigation among others was to analyze dependency on azimuths of sources, two different feature sets were constructed:

MEAN-CHANNEL FEATURES Auditory representations were first averaged over the left and right channel. Generating time-invariant features by applying L-moments over time as described in Section 3.1.1.1 finally amounted to 1082 dimensions per feature vector in this set.

TWO-CHANNEL FEATURES Instead of averaging the auditory representations over the two channels, features were constructed for each channel separately. Applying the same procedure as for the *mean-channel features* resulted consequentially in 2164 feature dimensions.

The two-channel features were assumed to contain more information about sound events that were emitted from sources at one side of the head, but to also be more specialized to the particular azimuth configuration used in training and hence more sensitive to deviations therefrom.

4.1.3 *Training*

Due to the high computational demand of the analyses conducted in this part (cf. Section 4.1.4), SED models were built only for four of the sound classes: alarm, crying baby, female speech, and fire. For these, segment-based labels were produced as described in Section 3.1.2.1. However, the sounds from all other classes of course still served as negative examples during the training and testing of these four classifiers.

For each of the four target sound types, binary one-vs-all classifiers were trained with GLMNET single-conditionally, as described in Sections 3.2 to 3.4. Models were trained on three different (overlapping) training-test-set splits. Each of the training sets consisted of 75 % of the sound files, which amounted to roughly 75 000 samples for each scene.

To enforce the effect of SNR on training and testing, thus helping systematic evaluation with respect to it, blocks with inactive distractor source were removed, since sounds can exhibit silences and the SNR naturally varies over time[3]. The distractor source was defined inactive in a block if its energy in the block was below -30 dB compared to the 99th-percentile of the distractor energy in the whole scene-instance.

4.1.4 *Evaluation*

For this study, performance was evaluated either

- on test data from the same scene (hence combination of SNR and azimuth configuration) as in training, called *iso-testing*, or
- on test data from a different scene than in training, which is called *cross-testing*.

The first tests the achievable performance on this scene (with the used sound data, features, algorithm, and methodology), the second tests the robustness of models with respect to deviations from training conditions.

Below, further in-between terms will be used:

- *iso-azimuth* to refer to tests where the same azimuth configuration was used in training and testing, but where SNR values between training and testing could differ, and
- *iso-SNR* to refer to tests where the same SNR was used in training and testing, but where azimuth configurations between training and testing could differ.

3 In scene rendering, the *time-average* SNR *excluding silences* is fixed, cf. Section 2.2.2.

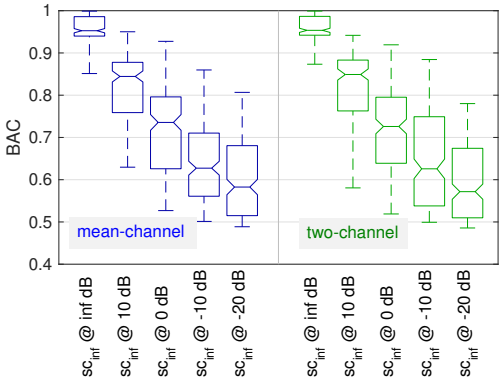


Figure 4.1: Iso- and cross-test performances of the sc_{inf} models, separately for mean- and two-channel features, each boxplot aggregating results from the three dataset splits, four target types, and all scenes with the respective SNRs (denoted "@<snr> dB"). © 2017 IEEE (Trowitzsch et al. 2017)

Evaluation was carried out for the models on the three test sets for each scene separately. With $(68 + 5) \times 3 = 219$ (scenes from sets 1 and 2, data splits) single-conditionally trained models, therefore $219 \times (68 + 5) \sim 16k$ (models, scenes from sets 1 and 2) iso and cross tests *per class*, it becomes apparent why only models of four target classes were trained and tested for this study.

4.2 GENERALIZATION OF ONE-SOURCE MODELS ON TWO-SOURCE SCENES

In the first analysis, it was investigated how well sound detection models trained on clean sounds (scenes set 1, cf. Section 4.1.1) generalize to noisy situations with an additional distractor source present. For this purpose, models trained on clean sounds (in the following with the token sc_{inf}, for "trained single-conditionally on a scene with inf dB SNR") were iso-tested, as well as cross-tested on scenes from set 2, where a distractor source emitted co-occurring sounds at SNRs of 10 dB, 0 dB, -10 dB and -20 dB.

Fig. 4.1 shows the test set model performances pooled over the four target classes, 17 angular configurations, and three data set splits, for each of the SNRs. The figure shows two groups of box-plots, corresponding to mean-channel features (left) and two-channel features (right). The first box-plot in each group corresponds to iso-testing (trained and tested on clean sounds, with the same azimuth configuration), whereas the next four plots correspond to cross-testing (trained on single-source

scenes, tested on scenes with two sources; same target source azimuth configuration) with increasing distractor energy level. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles. The notches represent the 95 % confidence interval for the median, while the whiskers indicate the minimum and maximum values.

While models trained on clean sounds perform well under iso-conditions with a median balanced accuracy (BAC) of about 0.96, their median cross-test performance is significantly worse, even when only a relatively quiet distractor sound is added to the scene (SNR of 10 dB), and decreases strongly further with decreasing SNR. There is no considerable difference between the median performance of the mean-channel and two-channel feature sets. This was expected, because in the case of training on one-source scenes, models did not have to learn to favor the stronger of the two channels for separating from a distractor source – if only one sound is emitted at a time, both ears receive a sufficiently strong signal, regardless of the azimuth the source is placed.

4.2.1 *Achievable performance with specialized two-source models*

This result gave rise to the question whether the bad generalization of monophonic models to polyphonic data was due to the intrinsic increased difficulty of the task, i. e. a better performance could not be expected, or whether models trained on clean sounds might be sub-optimal when applied in a setting where multiple sources are present simultaneously.

To answer this question, a second analysis was conducted, where not only the detection models trained on clean sounds were considered, but also detection models trained on scenes with a distractor source (set 2 in Section 4.1.1). SNR and angular configuration were set to the same values that were later used for testing (*iso*-testing). The corresponding performances are shown in Fig. 4.2, pooled again over model tests of all azimuth configurations, dataset splits, and target classes.

Again a drop in performance with increasing noise level is observed – however, it is much less steep compared to Fig. 4.1. It can be seen that for each SNR, the iso models perform considerably better than the models trained only on clean sounds, with differences in BAC up to 0.15. In addition to this, the performance decrease with SNR can be observed to be shaped differently from the performance decrease of the clean models, namely going from a small drop to higher ones, instead of the other way around.

For models trained on scenes set 2, the iso-test performances are notably higher for the two-channel than for the mean-channel feature

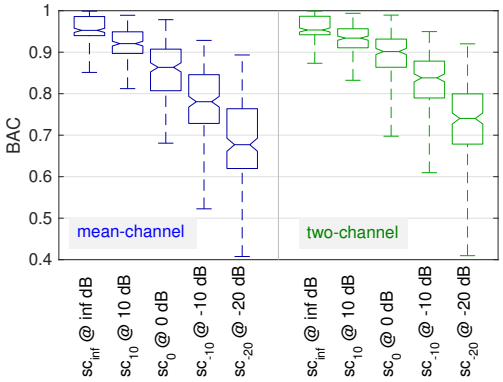
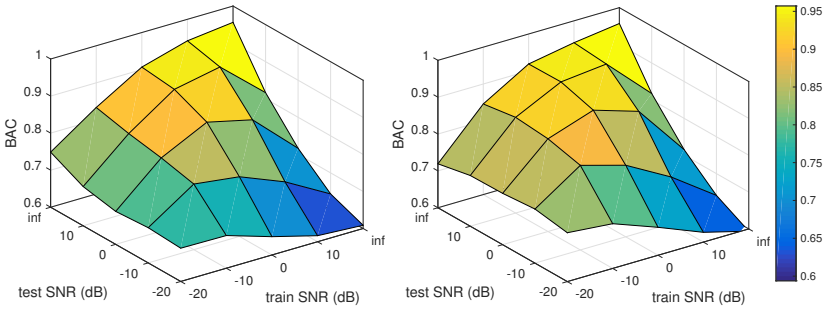


Figure 4.2: Iso-performances of models trained at various SNRs, separately for mean- and two-channel features, each boxplot aggregating results from the three dataset splits, four target types, and all scenes with the respective SNRs (denoted "@<snr> dB)". © 2017 IEEE (Trowitzsch et al. 2017)

set, and the difference grows with decreasing SNR. Contrary to the case of single-source training of models, the two-channel feature set enables models in training on two-source scenes to profit from angular separation between target and distractor sources by favoring the channel on the side of the target source.

These results suggest that models trained on clean sounds do not generalize well to realistic acoustic environments occupied by additional distractor sound sources. Instead, models specialized to the particular average SNR condition at training time were shown to yield much better performance at test time. One likely explanation is that the models trained on clean data emphasize features that are well discriminating in the monophonic case, but no longer in the polyphonic scenarios. However, if distracting sounds are already superimposed during training, models obtain a higher robustness against such noise.

Note that the general sound class used for generating the distractor signals contains a very diverse set of environmental sounds, hardly showing any similarities other than being a sound at all, and that neither distractor nor target sounds used for testing have been involved in the training process at any point. Therefore, the models could not adapt to a particular type of distractor signal and filter it out, but rather had to learn to be robust against a wide spectrum of possible distractors by finding the features that uniquely discriminate the target class from all others.



(a) mean-channel feature set (b) two-channel feature set

Figure 4.3: Iso-azimuth-performances of models displayed over train and test SNR, averaged over all iso-azimuth scenes, data set splits, and sound classes. © 2017 IEEE (Trowitzsch et al. 2017)

4.3 GENERALIZATION ACROSS CONDITIONS

The previous analysis addressed the generalization of monophonic models to superimposed sounds from two-source scenes. In the following, the question of how robust models are against deviation from training conditions is approached more generally, and it is investigated in how performance of such cross-tests depends on the similarity between training and test conditions.

In the scenes with two simultaneously present sound sources two factors describe the conditions: (i) the SNR between the two sources, and (ii), the azimuths of the two sources. In order to understand how cross-testing the two factors affects generalization, each factor was modulated separately, and performed *iso-azimuth* and *iso-SNR* (cf. Section 4.1.4) tests. These two test paradigms have an overlap: the tests in which both azimuth and SNR are iso-tested, i. e. the tests that perfectly match the training conditions.

4.3.1 *Cross-SNR generalization*

The effects of average distractor energy level differences between testing and training on the generalization performance are depicted in Fig. 4.3, summarizing the *iso-azimuth* tests. The graphs show the BAC, averaged over all azimuth configurations, data set splits and target sound types, as a function of the SNR at test and train time. Results are plotted separately for mean-channel and two-channel feature sets.

The values on the diagonals, i. e. values of models trained at the same SNR as tested, are highest with respect to fixed testing SNRs. As the

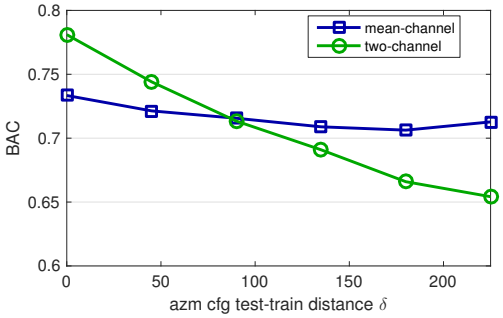


Figure 4.4: Performance of sc model cross-azm tests, averaged over all data set splits, classes, and iso-SNR configurations. The x-axis groups cross tests by "azimuth distance" of the cross configurations. © 2017 IEEE (Trowitzsch et al. 2017)

training SNR deviates away from the testing SNR, the performance compared to iso-models decreases monotonically, across all testing SNRs, and in *both directions* of deviation. There are a few combinations with only small effects, like testing at 10 dB of models trained at 0 dB or vice versa, but in general the performance decrease is quite large. Looking at it the other way around, i. e. at a fixed training SNR, performances do (mostly) increase for higher testing SNRs, but never reach the values of models that were trained at the same SNR.

As was observable in the study above, for training or testing SNRs below inf dB, in general the two-channel feature set models performed stronger.

4.3.2 *Cross-azimuths generalization*

In order to assess the effect of changing the angular configuration of target and distractor source, a simple (geometric) distance measure for two-source azimuth configurations was defined.

Let (φ_1, θ_1) and (φ_2, θ_2) be two azimuth configurations c_1, c_2. The distance $\delta(c_1, c_2) \in [0°, 360°]$ between these two azimuth configurations is defined as

$$\delta(c_1, c_2) := \alpha(\varphi_1, \varphi_2) + \alpha(\theta_1, \theta_2), \tag{4.1}$$

where $\alpha(\phi_1, \phi_2) \in [0°, 180°]$ is the smallest angular distance between two azimuths $\phi_1, \phi_2 \in (-180°, 180°]$, defined by

$$\alpha(\phi_1, \phi_2) := \min\left(|\phi_1 - \phi_2|,\, 360° - |\phi_1 - \phi_2|\right). \tag{4.2}$$

Fig. 4.4 shows the average BAC of the *iso-SNR* tests as a function of this distance δ between testing and training azimuth configurations, for

both the mean-channel and the two-channel feature set. The average includes all iso-SNR performances of the three data set splits and four target classes. Three observations are notable:

- For the mean-channel feature set, the performance decreases only slightly with the distance in angular source configuration. The increase at the end of the distance curve can be explained by the geometric nature of the distance measure, which does not account for symmetry effects in binaural auditory perception (there is symmetry between sounds coming from the front and the back of the head, i. e. models trained for a particular azimuth configuration in the frontal hemisphere may also work well for the mirrored configuration in the back hemisphere).
- For the two-channel feature set, the performance drops much more steeply as δ increases. For a δ of 180°, the drop in average BAC is about 0.11.
- For $\delta = 0°$ (*iso-azimuth*), the two-channel feature set models exhibit a higher average performance than the mean-channel models. The two curves cross at a distance of about 90°. This shows that sound event detection models with the two-channel feature set are better at making use of directional separation of sources, but consequently are at the same time more strongly affected by deviations of the true (testing) angular source configuration from the training situation.

4.4 SUMMARY

In this chapter,

- it was shown that binaural sound event detection models trained on one-source scenes degrade strongly in cases when a distractor source is present simultaneously emitting unspecific sounds, but by superimposing highly variable general sounds at training time, models can learn to focus on the target class in the presence of distracting sounds.
- it has been established that the performance of such models still depends on how similar the training conditions (azimuth configurations and SNRs) are to test conditions. In particular, deviations in SNR lead to distinct performance drops. Effects of deviations in the azimuth configuration were different for the two feature sets that were analyzed: while the mean-channel feature set handled deviations in this parameter more tolerantly

than the two-channel set, the latter showed significantly better performance if training and testing configurations were close enough.

- methodology for investigation of model robustness with respect to acoustic conditions has been introduced.

All in all, the results presented here make clear that models trained at particular auditory conditions are *specialized* to these conditions, and do *not* generalize well to different conditions.

5

MULTI-CONDITIONAL MODELS: DATA-DRIVEN ROBUSTNESS

This chapter is based on Ivo Trowitzsch et al. 2017. "Robust Detection of Environmental Sounds in Binaural Auditory Scenes." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (6): 1344–1356.

In the last chapter, the impact of deviations in acoustic conditions from the ones in sound event detection (SED) model training was analyzed. Section 4.3 showed that the application of models is very sensitive to deviations of the environment from training conditions. Practical applications are confronted with two problems:

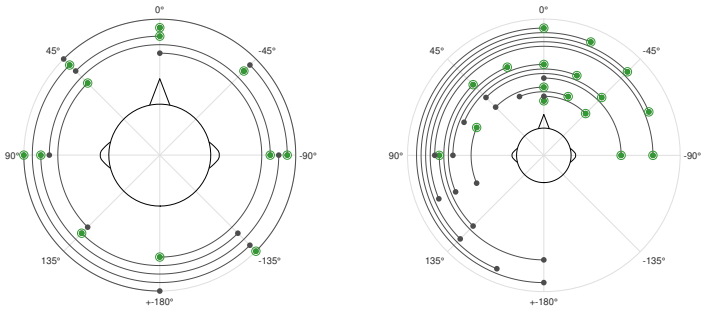
1. Training specialized models for every possible combination of conditions would be extremely demanding, if not prohibitive.

2. Inferring the current conditions is very difficult, in particular for the signal-to-noise ratio (SNR), which also varies strongly over time. Unfortunately, model performance is particularly sensitive to the deviation of the SNR from the training situation.

In this chapter, a method to increase the robustness of SED models by performing multi-conditional (MC) training is suggested and evaluated. In this scheme (introduced in Section 3.2.2), training data for model building gets composed from many different auditory scene configurations of target and superimposing sources. As in the chapter before, SNRs and angular source directions were varied — multi-conditional models varying room acoustics are investigated in the next chapter.

It will be shown that it is possible to a large extent to make models *learn robustness* against varying and deviating acoustic conditions, completely data-driven, and, looking at it the other way around, specifically without any engineering like noise-suppression, source separation, et cetera.

First, Section 5.1 describes any particularities or amendments to Chapters 2 and 3 with respect to data and methods concerning this chapter. Study results are presented then starting in Section 5.2, which elaborates



(a) multi-conditional training scene set

(b) single-conditional scene set

Figure 5.1: Depiction of sets of scenes. Black filled circles depict distractor sources, target sources (green) are highlight by an enclosing open circle. Each scene is indicated by one circle fragment. The head is at the center. In (b), some scenes are mirrored along the 0°-axis such that distractor sources always are positioned counter-clockwise, as was the interpretation in the analysis of azimuth-configuration dependence (Section 5.3).

on overall and SNR-specific performance of multi-conditional models in comparison to single-conditional ones; Section 5.3 analyzes dependence on actual azimuth configurations. In Section 5.4, the multi-conditional approach is applied to Detection and Classification of Acoustic Scenes and Events (DCASE)-2013 challenge data and performance compared to other systems.

5.1 DATA AND METHODS

Data and methods in this chapter follow up on the ones described in Section 4.1, that is, elucidations given therein are applicable here too. For this study, mainly auditory scene sets were added to the ones defined in the chapter before; feature sets, model training, and evaluation in large parts are equal to the single-conditional training.

5.1.1 *Auditory scenes*

The original sound data for synthesizing auditory scenes was taken from the NIGENS database, described in Section 2.1. For methodology and terms regarding the scene rendering, confer Section 2.2.

Additionally to the scene sets defined in Section 4.1.1, the following sets were used in this study:

3. Scenes containing two point sources emitting superimposed sounds. The target source emitted sounds from all classes, including "general", the distractor source emitted sounds only from the general class. Target and distractor sound sources were located at the following combinations of azimuths[1]:
   - $0°/\{0°, 45°\}$
   - $45°/\{135°, -135°\}$ & $-45°/\{-45°, -90°\}$
   - $90°/\{180°, -90°\}$ & $-90°/\{-45°, 90°\}$
   - $135°/\{-135°, 45°\}$ & $-135°/\{-90°, 45°\}$
   - $180°/\{180°, 0°\}$

   Each of these azimuth combinations got combined with four different SNRs (10 dB, 0 dB, −10 dB, −20 dB). A one-source scene was added for each of above target source azimuths to this set. Together, this amounted to $16 \times 4 + 8 = 72$ scenes in this set.

4. Four scenes containing a non-head-related target signal with sounds from all classes, and a non-head-related distractor signal with sounds from the general class, at SNRs of 10 dB, 0 dB, −10 dB and −20 dB. A scene with only one non-head-related source emitting sounds from all classes was added.

In set 3, the target and distractor sources are distributed uniformly around the circle. This set was used for training the multi-conditional point-source models. Since it provided higher angular resolution and to keep it comparable to single-conditional performance, set 2 defined in Section 4.1.1 was used for testing. Fig. 5.1 depicts the two sets.

5.1.2 *Training*

For multi-conditional model building, the same feature sets as in the study on single-conditional models was used, cf. Section 4.1.2, and segment-based labels as described in Section 3.1.2.1.

Two types of *mc* models were developed:

1. mc_{nhr}: Models trained by combining data from all 5 non-head-related signal configurations from scene set 4 in Section 5.1.1. Specialization to any directional head-related changes of the signals was impossible for these models since the signals didn't convey these by definition (cf. Section 2.2.2).

1 listed as: target source azimuth / *set* of azimuths for distractor source

2. mc_{ps}: Models trained by combining data from all 72 point source configurations from scene set 3 in Section 5.1.1. In this set, target and distractor point sources were uniformly distributed around the circle. In contrast to the mc_{nhr} models, in this case directional and head-related changes of the signal were included in the training, and investigation directed at whether this model type would also be able to generalize across azimuth configurations.

Training was performed as with the single-conditional models (Section 4.1.3), with the difference of using training data across above specified scenes multi-conditionally (Section 3.2.2). For each model training, 100 000 samples were subsampled as described in Section 3.2.3 across samples from all scenes (about 5 400 000).

5.1.3 *Evaluation*

To compare the generalization performance of multi-conditional models to single-conditional models, single-conditional performances were summarized in the following different paradigms:

- sc_{iso}: training and testing under the same conditions, both with respect to SNR and azimuth.
- sc_{isoAzm}: training and testing were performed at the same azimuth configuration. All combinations of SNRs in training versus testing were evaluated (that is, including both iso and cross). This displays a situation in which no information about the true testing SNR is available and thus a model trained at an arbitrary SNR is chosen.
- sc: training and testing at arbitrary iso- and cross-configurations, resembling a situation in which there is no information about the true SNR and azimuth configuration.

The multi-conditional models were tested on the same single-conditional data as the sc models, using sets 1 and 2 defined in Section 4.1.1. Note that the mc_{nhr} models were also tested on the point-source test scenes; training with non-head-related signals was only a mean to gain models robust to varying and deviating directional source configurations.

5.2 MULTI-CONDITIONAL MODEL PERFORMANCE

First, results of "grand-average" evaluation are presented, that is, without analyzing performance condition-specific, but rather pooling across all conditions. This is then followed by studying the dependence of different model types' performances depending on the testing SNR.

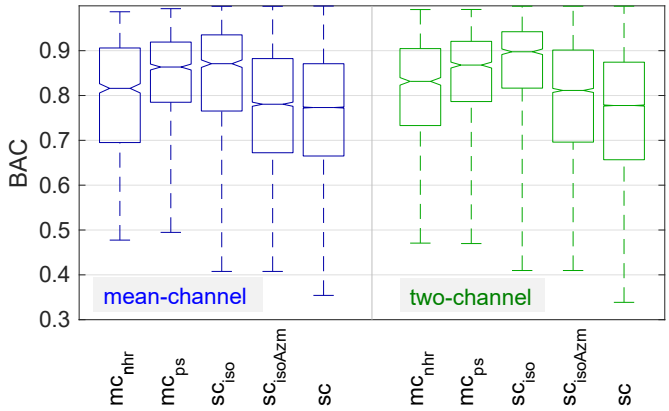
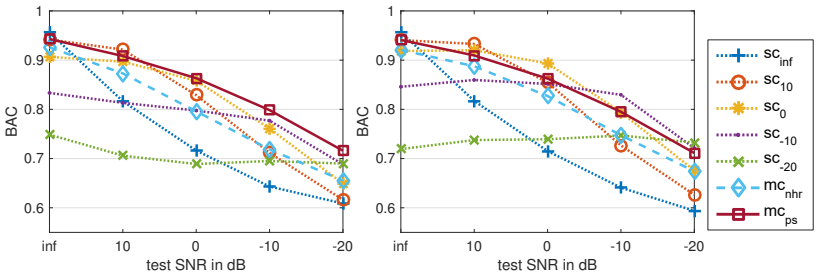


Figure 5.2: Performances of multi-conditional and single-conditional models under different testing paradigms (iso and cross), boxes pooling over SNRs $\{inf, 10, 0, -10, -20\}$, all corresponding iso- and/or cross-azimuth configurations, data set splits and sound classes. © 2017 IEEE (Trowitzsch et al. 2017)

5.2.1 *Grand-average multi-conditional performance*

In Fig. 5.2 depicts generalization performances of the mc_{nhr} and mc_{ps} models next to the performances of the single-conditional models. The figure shows box-plots of the balanced accuracies (BACs) for each model/test situation. Results were pooled over all azimuth configurations in the respective test paradigm, all SNRs, all data set splits, and all target sound classes, both for the mean-channel and two-channel feature set. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles. The notches represent the 95 % confidence interval for the median, while the whiskers indicate the minimum and maximum values.

The best median performance is obtained for the sc_{iso} models for both feature sets (the two-channel sc_{iso} models stronger than mean-channel). This is not surprising, as these models are specialized to the particular angular source configuration and SNR present in the test set. The sc_{isoAzm} models, which are only specialized to the correct angular source configuration but trained at an arbitrary SNR, perform on average much worse (about 0.1). For the "arbitrary" sc model, performance drops even further, by about 2% on the mean-channel feature set and by about 4% on the two-channel feature set (recall from Section 4.3 that the two-channel models suffer from stronger degradation in cross-azimuth situations).



(a) mean-channel feature set (b) two-channel feature set

Figure 5.3: Performance of different model groups at particular SNRs, averaged over all azimuth (iso-azimuth, for *sc* models) configurations, data set splits, and sound classes. © 2017 IEEE (Trowitzsch et al. 2017)

The multi-conditional non-head-related (mc_{nhr}) models, which were not trained on signals containing head-related directional information, (i) perform on average slightly better than the sc_{isoAzm} models, although the latter were specialized on the correct angular configuration, and (ii) clearly outperform the *sc* models trained at an arbitrary SNR and azimuth configuration. This means that in the absence of *reliable* information about the testing conditions, which would allow a suitable choice of trained model, the mc_{nhr} models offer a stronger performance – without the need to train many models and predetermine conditions.

However, the multi-conditional non-head-related models are clearly outperformed by the multi-conditional point source models (mc_{ps}), which were trained on a directionally on the circle uniformly distributed set of target and distractor sources under varied SNR. The median performance of the mc_{ps} models lies close to the performance of sc_{iso} models that were trained at the true angular source distribution and SNR. Note that this is the case even though most of the azimuth configurations from set 2 that had been used for testing are *not* in the training set for the mc_{ps} models, which underlines the strong generalization of these models. All models except the mc_{ps} models exhibited a higher median performance on the two-channel feature set than on the mean-channel feature set.

5.2.2 *Dependence of multi-conditional performance on SNR*

In Fig. 5.3, a more detailed plot is given of how the mc_{nhr} and mc_{ps} models compare to single-conditional models that were trained at specific SNRs and tested *iso-azimuth* at all SNR conditions. It shows the

average BAC as a function of the SNR at test time for the mean-channel feature set (a), and the two-channel feature set (b).

The following effects are observed:

- The performance of the mc_{ps} models always lie close to the *iso-SNR/iso-azimuth sc* performance, and even exceed it on the mean-channel feature set for low SNRs. When not being able to use the features of both channels to better separate spatially distributed sources, the mc_{ps} models seem to generalize better at low SNRs than the specialized sc_{iso} models. With the two-channel feature set, the single-conditional models perform better in iso-SNR/-iso-azimuth tests for all SNRs.

- The mc_{nhr} models do not quite reach the performance of the *iso-SNR/iso-azimuth sc* models, but outperform in many *cross-SNR/iso-azimuth* tests. This is without making use of any directional and head-related information during the training phase.

Inspecting the *sc* models again, the performance curves of the sc_{inf} models are exhibiting a convex shape, whereas the curves of other *sc* models are mostly concave. A possible explanation for this finding is that the models trained on clean sounds have not learned to accommodate the presence of simultaneous distractors at all, and are more strongly affected even by small amounts of noise than the other *sc* models. For these other *sc* models, performance leveling out at SNRs higher than the SNR at which they were trained can be observed (sc_{-10} shows this most obviously). One reason for this could be that these models specialize on features that do not get more discriminative with increasing SNR.

In summary, the results show that multi-conditional training by including different SNRs in the training data produces robust models that perform well over a wide range of SNRs. Although this also works if training is conducted with non-head-related signals, close-to-optimal performance can be achieved when the multi-conditional training is conducted on point sources varying not only in SNR but also including multiple angular source configurations.

5.3 DEPENDENCE OF PERFORMANCE ON AZIMUTH CONFIGURATION

After studying the dependence of models on SNR, the azimuth configuration's influence on sound event detection performance was investigated, with the implication of addressing whether a binaural robotic system could improve performance by turning its head.

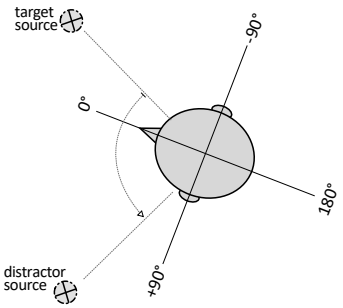
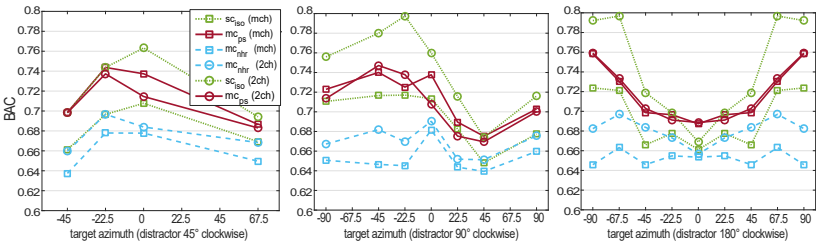


Figure 5.4: Compared to Fig. 2.3, the head has turned and the target-distractor azimuth configuration is now $-22.5°/67.5°$. © 2017 IEEE (Trowitzsch et al. 2017)



(a) 45° azm distance (b) 90° azm distance (c) 180° azm distance

Figure 5.5: Performance of different models at -20 dB for different azimuth distances between target and distractor, averaged over data set splits and classes; plotted over target azimuth with the distractor always put counter-clockwise. © 2017 IEEE (Trowitzsch et al. 2017)

For this purpose, model performance was analyzed separately for all azimuth configurations of set 2 except $0°/0°$. These were assigned to three groups based on the azimuth distance between target and distractor source (α), namely 45°, 90°, and 180°. Changing the azimuth configuration while keeping α fixed can be interpreted as a rotation of the head (see Fig. 5.4), if target and distractor are kept in the same topology.

Exploiting the mirror symmetry between right and left hemisphere of the binaural system, performances of models tested and trained at azimuth configuration $t°/d°$ could be re-interpreted as performances of models tested and trained at azimuth configuration $-t°/-d°$. This trick enabled interpreting model performances as if all azimuth configurations had been ordered with the distractor put counter-clockwise from the target source, see Fig. 5.1b.

For each azimuth distance and target azimuth, the BAC of multi-conditional models trained on non-head-related sounds (mc_{nhr}), multi-conditional models trained on point source sounds (mc_{ps}), and single-conditional iso-models (sc_{iso}) was evaluated. The results of these tests at an SNR of $-20\,\mathrm{dB}$[2] are shown in Fig. 5.5; separately for azimuth distances $\alpha = 45°$ (a), $\alpha = 90°$ (b), and $\alpha = 180°$ (c). The x-axis denotes the azimuth of the target source; the distractor is always assumed to be counter-clockwise from the target at a relative angle of α. Color and line style indicate the different models sc_{iso}, mc_{ps} and mc_{nhr}, each on the mean-channel and the two-channel feature set indicated through different markers.

Strong effects (up to around 0.13 BAC difference between best-performing and worst-performing head orientation) are found with the following qualities:

- For the sc_{iso} and mc_{nhr} models, the two-channel feature set models perform better than their mean-channel counterpart on all azimuth configurations, but the difference between the two varies greatly with head orientation and azimuth distance.
- A higher performance can be reached with larger azimuth distance[3], although a saturation seems to be reached at $\alpha = 90°$. This follows the intuition that sources lying closely together are harder to discriminate.
- The three azimuth distance groups show distinct performance profiles over the target azimuth.
    1. For $\alpha = 45°$, best performance can be reached at a target azimuth of 0°, for $\alpha = 90°$ performance peaks at $-22.5°$ target azimuth (distractor thus at 67.5° – the situation depicted in Fig. 5.4), and for $\alpha = 180°$, the highest performance is found at a configuration with the target at $\pm 67.5°$. All of these are configurations with the nose of the head being *close to the angle bisector* of target and distractor azimuths.
    2. Sub-optimal performance is particularly found at configurations that put both target and distractor on one side of the head, which is well-observable in the 90° azimuth distance plot, and at configurations where one source is in the front and the other is in the back, which is well-observable in the 180° azimuth distance plot.

2 The effect was stronger for lower SNRs, since higher distractor energy increased the positive impact of spatial separation.

3 results for the 0°/0°-condition, i. e. $\alpha = 0°$, are not shown in this figure, but performance is lower than for the other azimuth distances.

- The performance differences between head orientations are stronger for the two-channel feature set (for sc_{iso} and mc_{nhr} models). This is to be expected; it is more of a surprise that even the mean-channel feature set exhibits such a clear effect.

- Although the mc_{nhr} models did not learn any directional information during training and the mc_{ps} models had to learn from uniformly distributed azimuth configurations, the performance profiles of the three different model types for varying head orientations are, qualitatively, very similar. This indicates that changes in head orientation have beneficial or detrimental effects on performance that are similar for all models, but that the effect size differs between models. Single-conditional models that are specialized to particular azimuth configurations can more effectively exploit the spatial separation of target and distractor at the beneficial head orientations than multi-conditional models.

It can thus be concluded that the azimuth configuration plays a substantial role not only for single-conditional models, but also for multi-conditional models, even though they have learned to generalize across target and distractor sources being distributed around the circle. In a binaural system able to rotate the head, this configuration can be influenced such that the nose directs between the two sources, to increase detection accuracy.

5.4 MULTI-CONDITIONAL DCASE-2013 MODELS

To also validate the multi-conditional approach on other publicly available data, models were built using the DCASE-2013 (Dan Stowell et al. 2015) event detection *Office Synthetic* (OS) task training data, and tested on both the OS and *Office Live* (OL) test data.

The DCASE training data consists of 320 audio files from 16 sound classes (from office environments, like speech or printer) containing individual sound events. Single-conditional and multi-conditional models were trained in similar fashion as before, with the difference that only *non-head-related* signals instead of point sources were used (because DCASE tests are not binaural). Also, since a "general" class is not employed in DCASE, the distractor source emits sounds from all but the target class. Models were optimized using BAC.

The DCASE testing data consists of 12 (OS) plus 11 (OL) wav-files containing sequences of overlapping (OS) or non-overlapping (OL) sound events. The OL test files are actual recordings while the OS files are "synthetic" mixtures. The frame-based evaluation was conducted on non-overlapping 10 ms blocks per file.

Table 5.1: Performance on DCASE-2013 event detection test sets.

MODEL	BAC_{os}	$F1_{os}$	BAC_{ol}	$F1_{ol}$
sc_{inf}	59.20	17.73	68.28	16.65
sc_{10}	63.53	21.21	66.85	17.93
sc_{0}	67.85	23.08	68.30	20.30
sc_{-10}	71.50	22.78	72.62	17.05
sc_{-20}	68.43	18.43	67.86	11.53
mc_{nhr}	**71.97**	**24.56**	**73.91**	18.51
$dcase_{baseline}$	–	12.76	–	10.72
$dcase_{winner}$	–	21.28	–	**61.52**

Table 5.1 presents the results of the models on the OS task (columns 2 and 3) and on the OL task (columns 4 and 5), BAC and F-score values are given. For comparison, the DCASE-2013 baseline system's results as well as the challenge-winning system's results are added ("GVV" system for the OS task, an NMF decomposition with HMM post-processing (Gemmeke et al. 2013), and "SCS_2" system for the OL task, a Gabor filter bank feature extraction followed by a 2-layer HMM (Schröder et al. 2013), overview in Dan Stowell et al. (2015)). All performance values were computed using the published metrics code of the challenge to ensure results are completely comparable, F-score values were thus micro-averaged (cf. Section 3.3.3). BAC value computation was added to this code (thus for the DCASE systems these numbers are not available), macro-averaging over the 12 respective 11 performances was applied here.

The following points are noteworthy:

- Most importantly and supporting the results on NIGENS presented above, the multi-conditional model performs better than the single-conditional models, demonstrating the increased robustness and invariance to different conditions.

- On the (supposedly more difficult) OS task, the mc_{nhr} model outperforms the winner of the DCASE-2013 challenge. Without any temporal modeling (through HMM or the like) and detecting purely block-based, furthermore without utilizing the DCASE development data sets which would improve model selection, this is only a lower bound on the possible performance of this training scheme. It serves though as validation of the reasonability of the approach as well as of the used algorithm and features.

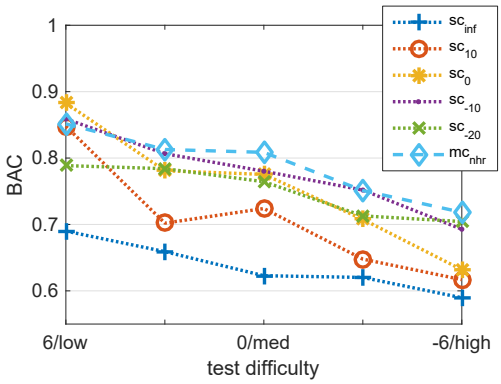


Figure 5.6: Average performance of mean-channel models on DCASE-2013 office synth test sets. Averages are over models' performances of all sound classes and individual test data files (sorted into the respective difficulty levels: the number denotes the SNR between events and background noise, and low/med/high denotes the events' density, i. e. degree of overlap). © 2017 IEEE (Trowitzsch et al. 2017)

- Comparing the F-scores of the OL task, the multi-conditional model is far away from the winning system, but still reasonably over the provided baseline system's performance. Again, the DCASE development set was not utilized, which may have been particularly useful for the OL task, because it included background noise not included in the training set. Also, models were optimized for BAC (with consistent value compared to the OS task), and thresholds for logistic regression models optimizing BAC will usually *not* optimize the F-score. It is thus fair to assume that the models easily could be tuned to a higher F-score.

Fig. 5.6 shows – similar to the plots in Fig. 5.3 – in more detail the behavior of the single-conditional and multi-conditional models in dependency of the *difficulty* of the test case. Unlike before, difficulty here not only refers to the SNR between target and distractor source, but is a combination of (a) SNR between events and background noise and (b) "density" of events, i. e. their degree and frequency of overlap. As in the extensive tests on NIGENS data, the multi-conditional outperforms almost all single-conditional models on almost all conditions.

5.5 SUMMARY

In this chapter,

- the use of multi-conditional training was proposed and introduced in order to obtain robust classifiers independent of testing condition, which in real-world applications can change quickly.
- it was found that multi-conditional training on point source superpositions from multiple azimuth configurations at multiple SNRs resulted in models with very stable performance competitive to specialized models trained under exact testing conditions, and that clearly outperformed single-conditional models trained at arbitrary SNRs and azimuth configurations. Two-channel features do not provide an advantage for these models.
- the approach was further validated by application of multi-conditional models onto the DCASE-2013 event detection challenge data.
- the effect of azimuth configuration and head orientation on sound event detection performance was investigated for different azimuth distances between target and distractor source. It was found that for both multi-conditional and single-conditional models, there is an optimal head orientation depending on the azimuth distance, at which the performance is maximized.

In conclusion, multi-conditional point-source models are a good choice for practical binaural applications: only a single model needs to be trained for each target sound class, inferring the conditions a priori is not required, the resulting models are robust with respect to conditions and even reach a close-to-optimal performance.

6

MULTI-CONDITIONAL MODELS FOR REVERBERANT SCENES

This chapter is based on results produced in the context of Jan Dikow. 2018. "Analyse des Einflusses von Räumlichkeit auf die Robustheit maschineller Geräuscherkennung in binauralen Hörszenen." Master thesis, Technische Universität Berlin.
The thesis was conceptualized and supervised by me, data (NIGENS database) and tools (AMLTTP) were also provided. The actual design of experiments, including the search for suitable BRIRs, the experiments' coding and conduction was done by Jan Dikow. Graphs, texts and analyses in this chapter are new and produced by me.

In Chapters 4 and 5, the sensitivity of *single-conditional* sound event detection (SED) models was analyzed regarding deviations of acoustic conditions from training conditions; and *multi-conditional* modeling proposed and investigated to overcome this behavior and gain robust performance. So far, analysis was conducted with respect to signal-to-noise ratio (SNR) and azimuth, under free-field (anechoic) conditions.

However, in realistic applications, there will often be room acoustics, that is, specifically reverberation, involved. While it is more possible to know room conditions of models in application in advance than it is for the SNR or azimuth configuration, one can think of more applications in which models should be robustly able to detect in varying or a priori unknown room acoustics, than the other way around.

Therefore this chapter presents a study on the effect deviations of room acoustics impose on SED models trained at other room conditions, and demonstrates that the scheme of multi-conditional training is applicable also to this facet of robust model building.

Section 6.1 introduces particularities or amendments to Chapters 2 and 3 and Sections 4.1 and 5.1 with respect to data and methods concerning this chapter. Analysis of single-conditionally trained models' performance and robustness is presented in Section 6.2, followed by Section 6.3 in which the insensitivity of multi-conditionally trained models is demonstrated.

6.1 DATA AND METHODS

Data and methods in this chapter follow up on the ones presented in Sections 4.1 and 5.1, so descriptions given therein are applicable here too. For this study, different auditory scene sets were defined; feature sets, model training, and evaluation in large parts are equal to the single- and multi-conditional training methodology introduced before.

6.1.1 *Auditory scenes*

The original sound data for synthesizing auditory scenes was taken from the NIGENS database, described in Section 2.1. For methodology and terms regarding the scene rendering, confer Sections 2.2 and 2.2.2.

Room acoustics

Head-related impulse responses (HRIRs) and binaural room impulse responses (BRIRs) were selected for the experiments in this study to exhibit a wide range of different acoustics:

- The anechoic HRIR with KEMAR manikin at 3 m distance (Wierstorf et al. 2011), which was used in Chapters 4 and 5.
- Two BRIRs recorded in the audio lab of the University of Rostock with a KEMAR manikin at about 2 m distance (Erbes et al. 2015). The lab is of shoe-box type, sized 5 m × 5.75 m × 3 m height, has plastered walls and optional broadband absorbers (in which case it is also almost anechoic).
- Six BRIRs recorded in a lecture room from the Aachen Impulse Response (AIR) database described in Jeub et al. (2009). The room is of size 10.8 m × 10.9 m × 3.15 m height, with parquet, glass windows, one concrete wall, wooden tables and chairs. A HEAD HMS2 dummy head with microphones positioned next to the pinna (that is: outside the ear) was placed distanced between about 2 m to 10 m from the source.
- Six BRIRs recorded in the Aula Carolina, an old church, taken from an updated version of above mentioned AIR database. The room has stone floor and walls, large glass windows, and very high ceiling, size 30 m × 19 m. Same head and microphone position as above, at six different distances from the source between 1 m to 15 m.
- Two BRIRs recorded in the Promenadikeskus concert hall in Pori, Finland (Merimaa et al. 2005). It is sized 33 m × 23 m × 15 m

Table 6.1: Room acoustics configurations overview

ROOM	ABBR.	distance (m)	T_{60} (s)	BR	D_{50} (%)
free-field	*ane*	3	0.07	1.4	100
audio lab URO (abs.)	*labAbs*	1.88	0.29	1.27	98
audio lab URO	*lab*	1.88	1.06	1.33	83
lecture room	*lr*2	2.25	0.86	0.96	93
	*lr*4	4	0.85	0.86	86
	*lr*6	5.56	0.89	0.92	73
	*lr*7	7.1	0.89	0.95	71
	*lr*9	8.68	0.88	0.92	65
	*lr*10	10.2	0.87	0.95	62
concert hall	*co*5	4.6	2.22	1.09	68
	*co*12	11.7	2.22	1.08	21
church	*ch*1	1	3.43	2.53	96
	*ch*2	2	3.47	2.21	91
	*ch*3	3	3.51	2.26	85
	*ch*5	5	3.50	2.21	72
	*ch*10	10	3.45	2.49	31
	*ch*15	15	3.47	2.13	16

height, and has a variety of diseffusers and reflectors to enhance acoustics for audience and musicians, rising floor, upholstered seats, and balconies on the sides. A Brüel and Kjær HATS dummy head with in-ear microphones was used at two distances to the source of about 5 m and 12 m.

Table 6.1 lists all room/position configurations together with reverberation times T_{60}, bass ratio BR, and Deutlichkeit D_{50} (all computed from the actual BRIRs, cf. Weinzierl (2008))).

Scenes rendered

For each of the room/position configurations, two scenes were rendered:

1. A single-source scene with the source emitting sounds from all classes. The head faces the source, i. e., the source's azimuth is (about) 0°.

2. A two-source scene with the target source emitting sounds from all classes, including "general", and the distractor source emitting sounds only from the general class. Both sources are placed at the same position with the head facing the sources, i. e., the sources' azimuths are (about) 0°. The SNR was set to 0 dB.

All experiments were restricted to using scenes either of type 1 or type 2, both for training and testing. That is, no cross-testing with regard to the number of sources was conducted, and no multi-conditional training across number of sources was performed.

6.1.2 *Training*

As in Chapters 4 and 5, the study was restricted to the four sound classes alarm, crying baby, female speech, and fire. For each of these four target sound types, binary one-vs-all classifiers were trained with GLMNET single-conditionally (models in the following abbreviated *sc*) or multi-conditionally (models abbreviated *mc*), as described in Sections 3.2 to 3.4. Models were trained on four overlapping training-test-set splits, each of the training sets consisting of 75 % of the sound files.

Single-conditional models were developed from the following 9 room/position configurations: *ane*, *labAbs*, *lab*, *co*5, *co*12, *lr*2, *lr*10, *ch*1, *ch*10. (Confer Table 6.1 regarding the abbreviations.)

Multi-conditional models were developed from the following configurations:

- mc_1: *ane* + *ch*10
- mc_2: *ane* + *ch*10 + *lr*10
- mc_3: all of above stated configurations used for *sc* model training.

Since the analysis in Chapter 5 revealed that for the superior multi-conditional (point-source) models, the two-channel feature set does not yield advantages, the study was confined to the mean-channel feature set, cf. Section 4.1.2.

6.1.3 *Evaluation*

For performance evaluation, the terms of iso- and cross-testing introduced in Section 4.1.4 were adhered to, relating not to SNRs and azimuth configurations, but to the particular room acoustics configurations used for training and testing.

To compare the generalization performance of multi-conditional models to single-conditional models, single-conditional performances were summarized in the following different paradigms:

- sc_{iso}: training and testing under the same conditions, both with respect to SNR and azimuth.
- $sc_{isoRoom}$: training and testing were performed in the same room. All combinations of positions in the room in training versus testing were evaluated (that is, including both iso and cross).
- sc: training and testing at arbitrary iso- and cross-configurations, resembling a situation in which there is no a-priori information about the true room acoustics.

The multi-conditional models were tested on the same single-conditional data as the sc models. Evaluation was carried out for the models on the four test sets for each scene separately. Since results for both scene types (one-source and two-source) were similar with respect to behavior regarding room acoustics[1], in the following, they are always pooled together and the different room/position configurations are referred.

6.2 PERFORMANCE OF SINGLE-CONDITIONAL MODELS

Two questions are to be answered in the evaluation of single-conditional models in this section:

1. How well can optimally trained SED models perform under different room acoustics?
2. How does generalization of models across room acoustics depend on the training acoustics?

Investigation is started with a special case of question two and the question: is it good enough to train sound event detection models in free-field conditions? Do these models generalize to other room acoustics? Fig. 6.1 therefore presents model performances of the sc_{ane} models, which were trained single-conditionally on the anechoic HRIR, both iso-tested on the same room, and cross-tested on the other 16 room acoustics. Boxes pool performances from the four dataset splits, four sound target classes, and two scenes; central marks indicate the medians, bottom and top edges of the boxes indicate the 25th and 75th percentiles. The notches represent the 95 % confidence interval for the median, while the whiskers indicate the minimum and maximum values.

1 of course the one-source scenes obtained higher detection performances

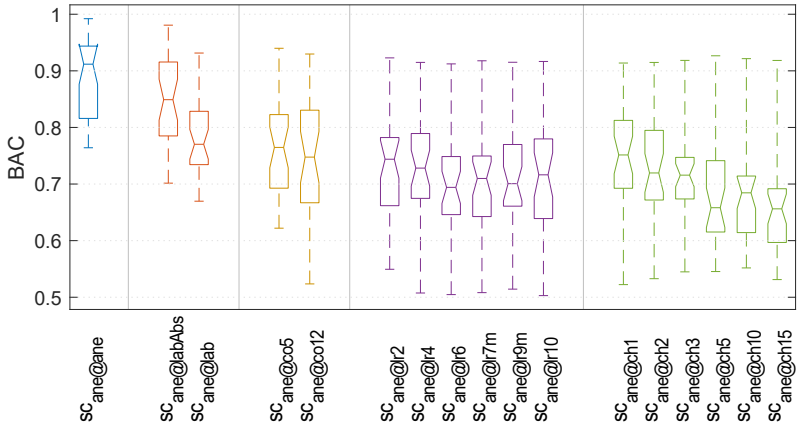


Figure 6.1: Iso- and cross-test performances of the sc_{ane} models, each boxplot aggregating results from the four dataset splits, four target types, and two scenes from the respective room/position (denoted "@<roomPos-abr>").

From these results, it is very clear that models trained under free-field conditions do not perform well on other room situations. A strong decrease of performances can be observed for all but the *labAbs* configuration (which however also exhibits moderately lower performance). The worst cross-performance is obtained in the Aula Carolina (church) room with 15 m distance from the source — also intuitively the room acoustics situation most different to an anechoic chamber.

6.2.1 *Achievable performance for different acoustics*

To look at whether the low performances of above tests are due to bad generalization of the sc_{ane} models, or due to intrinsic difficulty of sound event detection in the other rooms, sc_{iso} performances have to be compared, that is, the performances of the models trained under the same condition as tested.

Fig. 6.2 shows a box-plot of the 9 single-conditional *iso* performances, and it is apparent that room acoustics have no significant effect on the fundamentally achievable detection performance, at least not in the regions they were varied here. Medians, 25th and 75th percentiles are almost the same at all configurations. The low cross-test performance of the anechoic model hence is bad acoustic generalization (specialization to the anechoic case).

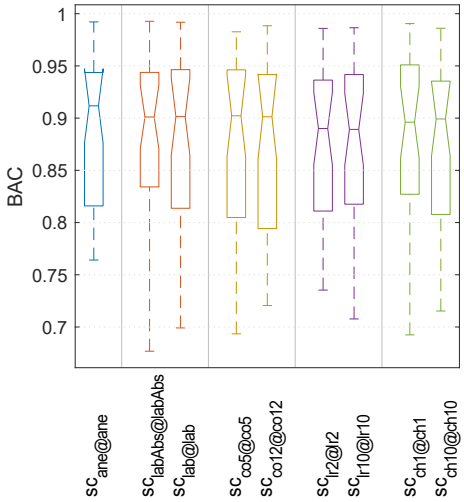


Figure 6.2: Iso-test performances of single-conditional SED models in different rooms. Each boxplot aggregates results from the respective models from the four dataset splits, four target types, and two scenes from the respective room/position (denoted "@<roomPos-abr>")

6.2.2 *Test performances across room acoustics conditions*

If in principle achievable performances are the same for different room acoustics, the question is how cross-test performances depend on the used room/position configuration in training. Additionally to the results presented in Fig. 6.1, Fig. 6.3a depicts the average performance of all test/training configuration combinations. Training room acoustics are put along the x-axis, test configurations along the y-axis. Configurations are ordered by cross-test performance with respect to the sc_{ane} iso-performance, with configurations from the same room next to each other. Combinations inside one room (*isoRoom*) are highlight by thin frames.

Several things are observable:

- As expected, iso-performances are highest.
- For testing under room acoustics other than anechoic, the two training conditions *ane* and *labAbs* (which is almost anechoic) are producing very bad-performing models.
- Iso-room test performances are high, that is, the differences in room acoustics introduced through varying distances and positions in rooms are having small (but observable) impact.

- Models trained at (supposedly) more "difficult" room acoustics seem to generalize better to "easier" conditions (upper right triangle of the cross-test matrix) than vice versa (lower left triangle of the cross-test matrix).
- Models trained in the Aula Carolina or in the lecture room generalize surprisingly well to each other, and better than to configurations from other rooms — although the two rooms are, acoustically, quite different.

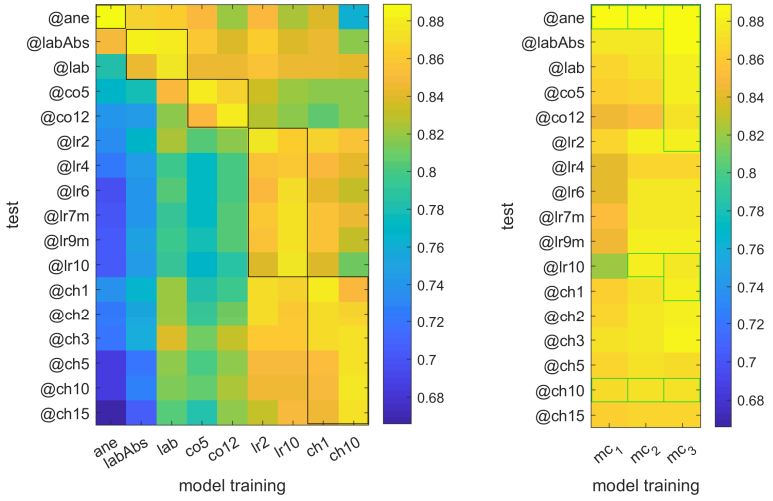
The latter may very well be indicative of a confounding effect: cross-performances in this study are not only affected by acoustic properties of the rooms, but also by the different recording setups, particularly the different heads, and most of all the usage of inner-ear versus outer-ear microphones. Both the lecture room and the Aula Carolina recordings have been done with outer-ear microphones. To only measure effects of room acoustics, all BRIR recordings would have to be done with the completely same setup; and such BRIRs were unfortunately not found.

6.2.3 *Cross-test performances depending on room acoustic parameters*

In Fig. 6.4, *sc* test performances are displayed in a different way: in each of the four panels, performances are plotted against test-training differences in acoustic parameters, defined as $d_{AP} = AP_{test} - AP_{train}$ (AP being the acoustic parameter of choice). $d_{AP} = 0$ thus corresponds to iso-tests (with respect to this parameter), any other values on the x-axes to cross-tests. Each of the dots represents the mean performances of models trained and tested at a particular combination of conditions over the dataset splits, target classes, and scenes.

The following trends can be pulled out of these results:

1. The best performance is always achieved if training models at the parameter values of the testing environment. For all four parameters, the top performances are lower for stronger deviation of the parameter from training condition.
2. Cross-test performance decreases are lower when training was conducted under more difficult conditions than testing:
   - Regarding the reverberation time T_{60}, performance seems to suffer less if training was conducted under room acoustics with higher T_{60} than in testing (negative d_{T60}), than vice versa.
   - The bass ratio BR shows the least clear effect, but tendentiously, higher values in training than in testing produce less degradation compared to the other way around.



(a) SC mean iso- and cross performances

(b) SC mean iso- and cross performances

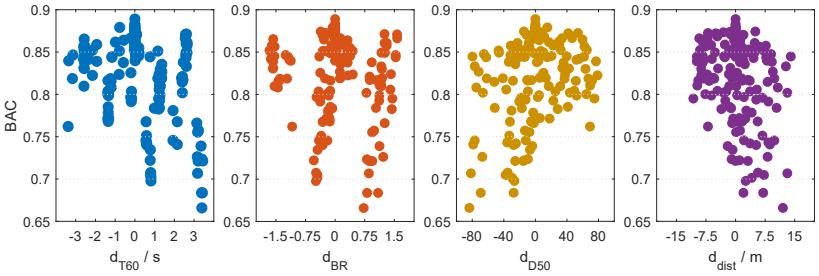
Figure 6.3: Performances of single-conditional and multi-conditional SED models in different rooms. (a) depicts mean iso- and cross-test single-conditional performances (isoRoom-tests framed black), (b) displays mean multi-conditional iso- and cross-test performances (configurations included in training framed green). Each average aggregates results from the respective models from the four dataset splits, four target types, and two scenes from the respective room/position (denoted "@<roomPos-abr>"). Color scales are equal for both plots.

- For the Deutlichkeit $D50$, cross-test performances are better for low values at training time.
- A higher distance between head and source in training results in higher cross-test performances than the other way around.

However, since the sample size with respect to different rooms was small and impulse response recording setups were not equal for all rooms, those are not more than assumptions indeed.

6.3 PERFORMANCE OF MULTI-CONDITIONAL MODELS

At least for situations in which the general room acoustics in model application are not clear a priori, single-conditional SED models come with a probability of reduced performance. As before for situations



(a) d_{TR60} ($\rho = -0.35$) (b) d_{BR} ($\rho = -0.09$) (c) d_{D50} ($\rho = 0.32$) (d) d_{dist} ($\rho = -0.31$)

Figure 6.4: Single-conditional SED performances depending on train-test-distances of room/position attributes. 0 distances correspond to iso-tests, others to cross-tests. Each dot represents the average results from the four dataset splits, four target types, and two scenes of a training-test-room/position combination. ρ states the corresponding Pearson correlation coefficient.

with unknown SNR and azimuth configuration (Chapter 5), multi-conditional training (Section 3.2.2) is hoped to be able to build models invariant to variations of room acoustics.

Fig. 6.3b displays the mean performances of the three multi-conditional models (as defined in Section 6.1.2) under all test room/position configurations, next to the single-conditional performances. It is eye-catching that all three models perform much stronger on unseen configurations than the single-conditional models; even the mc_1 model, trained only on two different room acoustic configurations[2], only has one stronger "slip-up" at the lecture room in distance 10 m. Both mc_1 and mc_2 demonstrate that the multi-conditional models generalize well also to completely other rooms (and heads). The mc_3 model, which includes training data from all rooms (but not all positions), excels across all configurations with optimal performance.

Fig. 6.5 summarizes these results in a box-plot aggregating performances for the multi-conditional models, the single-conditional models tested at the same room/position as trained (sc_{iso}), and tested in the same room as trained ($sc_{isoRoom}$), and tested on arbitrary configurations (sc). To keep the boxes comparable, here test results were only taken from the 9 configurations that were also used for training.

Tightly, sc_{iso} models exhibit the highest median performance, but the difference to mc_2 and mc_3 median performance is not significant, and mc_1 median performance is within reach. All three multi-conditional models show notably higher performance than single-conditional mod-

2 deliberately two configurations with very different acoustic properties

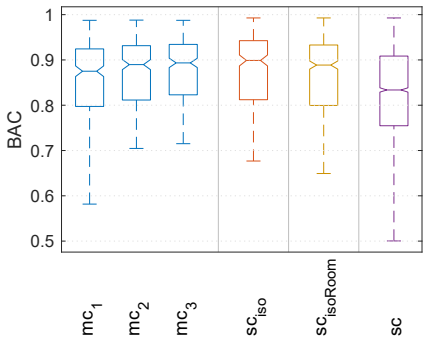


Figure 6.5: Performances of single-conditional and multi-conditional SED models tested under different room acoustic conditions, each boxplot aggregating results from the four dataset splits, four target types, and two scenes from all respective room/positions.

els trained at arbitrary configurations. However, if the room is known a priori, a single-conditional model trained on data from that room can perform similarly strong as the multi-conditional models.

6.4 SUMMARY

In this chapter,

- it was shown that single-conditional SED models specialize to the room and head acoustics they are trained on, and hence are sensitive to deviation at test time from these.
- results indicate that single-conditional models are less prone to cross-test performance degradation, if they were trained under "difficult" room acoustics.
- it was found that single-conditional models can perform adequately if they were trained on data from the room they are applied in later, but they do not need to be trained on data from the exact same position in the room.
- it was demonstrated that models trained multi-conditionally generalize very well across and to other room acoustics.

Therefore, it is concluded that multi-conditional models trained on data from several different room acoustics are the means of choice when acoustic conditions are not known a priori.

7

THE INFLUENCE OF TEMPORAL CONTEXT

This chapter is partly based on results produced by Heiner Spieß in the context of the seminar "NI-project". The conducted project was conceptualized and supervised by me in collaboration with my colleague Moritz Augustin, data for training and testing was created and provided by me. LSTM-coding and training was done by Heiner Spieß, TCN-coding and training by Moritz Augustin. Everything else was done by me, including specifically production of all graphs, texts and analyses in this chapter.

Sound is an inherently temporal information; successful sound event detection (SED) therefore relies on representations efficiently describing and classifiers efficiently modeling temporal data.

The first has been treated through construction of time-invariant features over blocks of 500 ms lengths in the studies presented so far, but the latter has remained untackled: the employed Least Absolute Shrinkage and Selection Operator (LASSO) classifier has no capabilities of modeling temporal relationships.

Deep neural network (DNNs), on the other hand, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are famous for their capabilities of modeling any kind of context, including temporal context — which is why they are the dominant architectures in sound event detection nowadays. Recent top-performing models and advances made have been primarily achieved by employing DNNs (Cakir and Virtanen 2017; Hayashi et al. 2017; Hertel et al. 2016; Jeong et al. 2017; Li et al. 2017; Huy Phan et al. 2016; Purwins et al. 2019; Xia et al. 2019).

Hence, this chapter describes experiments on multi-conditional SED in demanding polyphonic situations, comparing the already introduced LASSO models with two DNN architectures specifically designed for modeling of temporal sequences: (i) the very popular long short-term memory (LSTM) (Greff et al. 2017; Hochreiter and Schmidhuber 1997), which by design is constructed to learn temporally distant relationships through gating information over long durations, and (ii) temporal convolutional network (TCN) (Bai et al. 2018), a recent architecture providing an efficient feed-forward alternative to LSTM with supposedly comparable sequence modeling capability.

Motivated by DNNs' ability to naturally output predictions every time step, the problem of temporal modeling is furthermore augmented by a comparison of perceptually-motivated smooth segment-based and "instantaneous" frame-based labeling.

Analyses are conducted (i) with regard to behavior in different acoustic conditions, and particularly (ii) with regard to the influence of the size of temporal context accessible to the models.

First the used acoustic scenes and different models are introduced in Section 7.1. Results are presented in Section 7.2 with an evaluation of scene-average results about model types and sound classes, temporal context length and labeling methods; Section 7.3 analyzes scene-specific results about robustness across acoustic conditions. The two different DNN models and how they were employed in this study are discussed in Section 7.4.

7.1 METHODS AND DATA

The basic methods and data introduced in Chapters 2 and 3 are also used here. Anything specific to the experiments described in this chapter in this regard is elaborated on in the following sections. Additionally to the description of the auditory scenes rendered for training and testing, particularly the training of the DNN models is explained.

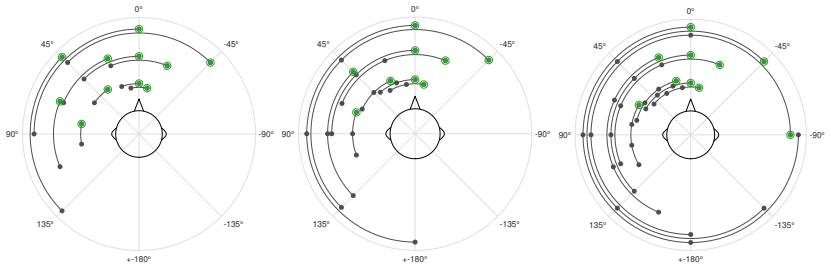
All scene generation, data processing, LASSO model training and testing was done using the Auditory Machine Learning Training and Testing Pipeline (AMLTTP) (Appendix A), which wraps all related steps described in the following sections. DNN training and testing was done using Python and KERAS (cf. Section 3.5), but fed with data produced by the AMLTTP.

7.1.1 *Auditory scenes*

The original sound data for synthesizing auditory scenes was taken from the NIGENS database, described in Section 2.1. For methodology and terms regarding the scene rendering, confer Section 2.2.

A set of binaural auditory scenes was rendered for training the detection models, and another set for testing. Compared to the studies in Chapters 4 to 6, the scope was extended beyond two-source scenes up to scenes with four simultaneously active sources. All sources, whether target source or distractor source, emitted sounds from all sound classes.

Eighty training scenes were defined for multi-conditional training (Section 3.2.2). Due to the increased number of free parameters of scenes



(a) Two sources (b) Three sources (c) Four sources

Figure 7.1: Test scene configurations (restricted to scenes with at least two sources and spread larger than 0°), sorted with respect to number of sources. Black filled circles depict distractor sources, target sources (green) are highlight by an enclosing open circle. Each scene is indicated by one circle fragment. The head is at the center. Neighboring scenes have the same number of sources and inter-source distances to make apparent the potential for investigation of the effect of head rotation.

with more than two sources, it seemed more efficient to randomly sample the parameter space compared to manual definition of scenes (as was done in Chapter 5). Randomly chosen were

- the number of sources (one to four)
- the azimuths of sources (uniformly between $\pm 180°$, discretized to 22.5°-steps[1])
- the signal-to-noise ratios (SNRs) between target and other sources (uniformly between $-20\,\mathrm{dB}$ and $+20\,\mathrm{dB}$).

For *testing, 168 scenes* were defined (manually) such that it would be possible to look at only one scene parameter changing and keeping the others constant — the higher number of scenes compared to the training set is due to this constraint. The following parameters were varied:

- the number of sources (one to four)
- the SNRs between target and distractor sources ($-20\,\mathrm{dB}$, $-10\,\mathrm{dB}$, $0\,\mathrm{dB}$, $+10\,\mathrm{dB}$, $+20\,\mathrm{dB}$)
- the azimuth difference between sources (0°, 20°, 45°, 90°)
- the "scene mode":

1 This discretization served computation efficiency; 22.5° is a compromise between smaller number of renderings and more dense spatial sampling.

1. *bisecting*: the nose (0° azimuth) points between target and distractor source(s)
2. *target@o*: the nose points towards the target source
3. *front-left*: sources are mainly between 0° and 90°; they are not bisected and targets are not at 0°, and they are not symmetric around the ear
4. *ear-centered*: sources are distributed in the left hemisphere symmetrically around the ear (90°)

Fig. 7.1 depicts the scenes.

7.1.2 *Model input*

As base auditory representations, ratemaps and amplitude modulation spectrograms were used, plus spectral features for the LASSO models, described in Section 2.3. Models were trained multi-conditionally (cf. Section 3.2.2 and Chapter 5) aggregating over all 80 training scenes (see Section 7.1.1). Training was performed on one training-test-set split, with the training set consisting of 75 % of the sound files.

Segment-based (referencing 0.5 s segments) and frame-based labels as described in Sections 3.1.2.1 and 3.1.2.2 were produced for each feature vector. For the segment-based labeling mode, temporal context size T was varied between 0.5 s to 20 s, for the frame-based labeling mode, the lower end was 50 ms.

LASSO MODEL FEATURES were constructed as described in Section 3.1.1.1. All three representations were split into overlapping blocks of length T, with a shift of 200 ms. The mean-channel feature set, as defined in Section 4.1.2, was constructed from these representations.

DNN MODEL FEATURES were constructed frame-based as described in Section 3.1.1.2. Ratemaps and amplitude modulation spectrograms (averaged over the two channels) were used as model input amounting to a 160-dimensional feature vector in each frame, i. e., every 10 ms, cut to length T. Spectral features were not used since in principle, DNNs should be able to build useful statistics over frequency themselves from the ratemaps input.

7.1.3 *Lasso Models*

LASSO models were trained with GLMNET for each of the thirteen target sound types defined by NIGENS as described in Sections 3.2

Table 7.1: Lasso model variants, both segment-based and frame-based labeling modes (SL and FL). The times specify block boundaries from segment ends; multi-block models can include overlapping or consecutive blocks. As an example, $[1-0.5, 0.5-0]$ describes that the respective model uses two consecutive 0.5 s blocks as input, thus the segment length with this model is 1 s. The variants achieving best mean performance of a respective temporal context are marked with *.

T (s)	Block boundaries from segment end (s) / labeling mode
0.05	$[0.05-0, 0.01-0]$* FL
0.2	$[0.2-0, 0.01-0]$* FL
0.5	$[0.5-0]$* SL
	$[0.5-0, 0.01-0]$* FL
1.0	$[1-0]$ SL $[1-0.5, 0.5-0]$ SL $[1-0, 0.5-0]$* SL
	$[1-0, 0.01-0]$ FL $[1-0.5, 0.5-0, 0.01-0]$* FL
2.0	$[2-0]$ SL $[2-0, 0.5-0]$* SL $[2.0-1.5, 1.5-1, 1-0.5, 0.5-0]$ SL
	$[2-1.25, 1.25-0.5, 0.5-0, 0.01-0]$* FL
10.0	$[10-0]$ SL $[10-0, 2-0, 0.5-0]$* SL
	$[10-0, 1-0, 0.5-0, 0.01-0]$* FL
20.0	$[20-0, 10-0, 2-0, 0.5-0]$* SL

to 3.4 and below. For each model training, 2×10^5 samples (out of about 12×10^6) across all scenes were sub-sampled[2] from the complete training set. The sub-sampling was done as described in Section 3.2.3; furthermore, the sub-sampling process enforced using equally many samples from each scene-instance, so that long sound files would not be overrepresented in the training set.

MULTI-BLOCK SEGMENT-BASED-LABEL MODELS The time-invariant features construction process described in Section 3.1.1.1 has shown in the studies before (Chapters 4 to 6) to produce descriptive features for blocks of 0.5 s length. However, since this method is based on the application of statistical moments over time, conjecturally this descriptiveness would not transfer to arbitrary segment lengths — through time-averaging, longer segments will get more "blurry". To assess this, for LASSO models for temporal contexts of longer than 0.5 s, additionally, *multi-block* models were trained, which were fed with concatenated features (built as usual) from consecutive and/or overlapping blocks out of the respective temporal context segment. All frame-based labeling LASSO models were multi-block models, as the features from

2 As the Lasso model has few free parameters (number of features + 1), this amount was enough — actually performance saturated even before.

the last frame (corresponding to the label) were always concatenated additionally. Table 7.1 lists all variants trained.

7.1.4 *Deep neural network models*

Both LSTM + DNN architecture (LDNN) and TCN are described in Section 3.5, along with methodology. Only the non-general details are given account of in the next sections.

DNNs are inherently able to learn multi-label problems. That is, compared to LASSO for which one binary model needed to be trained per sound type (13), with the DNNs architectures it was only one model with 13 output neurons[3].

Since nonetheless many models with different temporal context sizes had to be trained on the very extensive multi-conditional training data (about 242×10^6 frames – 672 h –, all together), and in lack of access to big GPU clusters, optimization of hyperparameters was reduced to partial cross-validation (pCV) (Girard 1998) on the three most representative (determined by LASSO cross-validation performance) splits.

7.1.4.1 *LDNN models*

Section 3.5.3 describes the model and further training methodology; Fig. 3.3 depicts the employed architecture. The following elaborates on concretely used hyperparameter values and further training details.

The number of LSTM layers was sampled from $\{3, 4, 5\}$ and the number of fully connected (FC) layers from $\{1, 2\}$ (excluding the output layer). The number of total neurons was sampled uniformly in $[500, 3000] \cap \mathbb{N}$, and then distributed between the LSTM and FC layers in a ratio sampled from $\{0.25, 0.5, 0.75\}$. All LSTM layers and all FC layers got the same number of neurons, respectively. The remaining neurons were allocated to the FC layers.

The FC layers' dropout rate was sampled uniformly in $[0.25, 0.75]$. In addition, binary hyperparameters determined whether recurrent variational dropout, dropout on the LSTM-cell hidden units, or both got employed; and if yes, whether at the FC layers' rate, or half of it.

The Adam optimizer was set to an initial learning rate of 0.001 and the remaining parameters were left at the defaults published in the original paper (Kingma and Ba 2014). Gradient clipping got set to 1.0 to prevent back-propagated gradients from becoming to large, which LSTM-Cells can still suffer from (Sutskever et al. 2014). 128 sequences were used per batch.

3 Cakir et al. (2015a) discussed multi-label vs single-label DNNs for SED

Due to limited computational resources, the optimal hyperparameter set was evaluated for the model trained with segment-based labels and a temporal context length of 10 s only. A small exploration led to the conclusion that this set would also be a good choice for models with different temporal context and trained with frame-based labels.

THE BEST-PERFORMING HYPERPARAMETER COMBINATION was found to be employing 5 LSTM layers with each 420 neurons, 2 FC layers (plus output layer) with each 370 neurons; a FC layer dropout rate of 0.7, and a rate of 0.35 each for the dropout on the LSTM-cell hidden units and for the recurrent variational dropout.

7.1.4.2 *TCN models*

Section 3.5.2 describes the architecture and further training methodology; Fig. 3.2 depicts the employed model. The following elaborates on concretely used hyperparameter values and further training details.

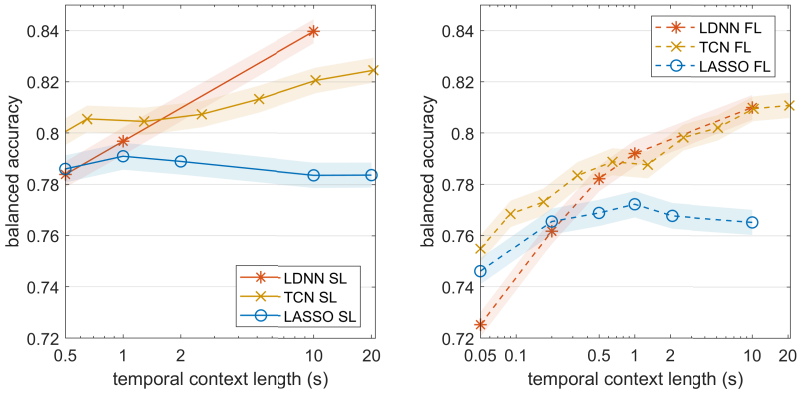
The values of several hyperparameters got fixed without complete search: the time convolution filter size K was set to 3, as a non-exhaustive exploration of kernel sizes between 3-5 showed no significant differences. The batch size was set to 128 and the initial learning rate of the Adam optimizer to 2×10^{-3} (in a small exploration with batch sizes 64, 128, 256 and learning rates 0.0005, 0.001, 0.002, this was identified as best combination). Gradient clipping was set to 1.5 (monitored gradient norms before clipping were usually below the value of 1.0). The length of each batch was set to 2500 frames (chosen for efficiency of the parallel convolution operation).

In systematic search, the number of feature maps was uniformly sampled within $[20, 160] \cap \mathbb{N}$. The (spatial) dropout rate was uniformly sampled from $[0, 0.25]$.

Due to limited computational resources, the optimal hyperparameter set was evaluated for the model trained with segment-based labels and a temporal context length of 20.49 s only. A small exploration led to the conclusion that this set would also be a good choice for models with different temporal context and trained with frame-based labels.

THE BEST-PERFORMING HYPERPARAMETER COMBINATION was found (after 43 random samples) to be employing 119 feature maps and a spatial dropout rate of 0.095.

For $T_{K,M} = 2049$ frames and $K = 3$, consequentially $M = 10$ stacked residual blocks were used. The number of epochs to train the final model (found from early stopped training) varied between minimally 8



(a) Segment-based labels (b) Frame-based labels

Figure 7.2: BAC depending on model type and temporal context. Performances are averaged over all test scenes, all test files, and all classes. Lines display arithmetic means, shaded areas the 95 % confidence intervals for the mean.

($T_{3,10} = 2049$ frames) and maximally 31 ($T_{3,5} = 65$ frames), and seemed to be anti-correlated with depth.

7.1.5 *Model testing*

While training was conducted multi-conditionally, tests were performed on individual scenes in order to conclude on relations between scene parameters and performances. All samples from the test set (168 scenes, each with 243 scene-instances) were used without sub-sampling, amounting to about 8.5×10^6 samples tested with each LASSO model and about 168×10^6 frames with each DNN model.

Balanced accuracy (BAC) (cf. Section 3.3.2.1) was employed as performance measure with macro-averaging (cf. Section 3.3.3) over classes and scene-instances.

7.2 INFLUENCE OF TEMPORAL CONTEXT AND LABELING METHOD

Firstly, model performances subject to different temporal contexts were evaluated disregarding scene dependencies. Fig. 7.2 shows the performance for the different model types and labeling methods, depending on temporal context size, averaged over classes and scene-instances.

Standing out immediately, there is a clear mean performance correlation with temporal context size; and it is obvious that the DNNs are

able to accumulate evidence about active sound events over long durations. The best-performing model overall is the LDNN with a temporal context of 10 s. Interestingly, the LDNNs seem to be less efficient for short temporal contexts; this shows both for the segment-based and for the frame-based modes.

Pooling LDNNs and TCNs together, the DNN models throughout all temporal context sizes perform stronger than the LASSO models. The difference particularly gets larger beyond 1 s, after which LASSO fails to make use of the additional information, while the DNNs are capable of extracting more useful information with longer temporal context accessible.

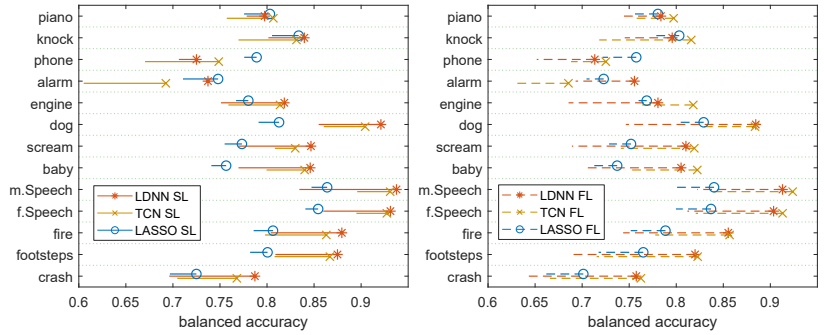
Notably, the LASSO models are *not* able to make use of temporal context beyond 1 s, which is their peak for both modes. This is despite the multi-block modeling; shown are the performances of the best-performing variants (cf. Table 7.1). For the other variants, performance was significantly degraded with longer contexts, which confirms the hypothesis that through the employed time-invariant feature construction (Section 3.1.1.1), information from longer segments will get less descriptive with regard to the sound event potentially occurring at the end of the segment.

Overall, however, the LASSO models come off surprisingly well in comparison to the much more powerful DNNs. For small temporal context lengths, the difference is as low as 0.01 to 0.02. This could imply two things: either the features constructed for the LASSO models are well-engineered and include a lot of relevant information for the SED problem, or the DNN architectures and/or training employed have not exploited their full potential — not unlikely, it is both.

FRAME-BASED LABELS pose the more difficult problem. Intuitively, this makes sense: the segment-based labels are smoother, since they express whether a sound event has been active for the larger part of the last half second. The frame-based labels, on the other hand, express whether a sound event is active right in this moment, which particularly for the beginning of an event can be hard to predict. Performances for equal temporal context are consistently lower by about 0.01 to 0.02; which fairly can be called a small difference, however. The dependency on sufficient context size is even more pronounced by the stronger decreasing performances for very short context sizes.

7.2.1 *Sound event type specifics*

Fig. 7.3 depicts performances for individual sound classes and model types. Performances are averaged over all test scenes and all test files.



(a) Segment-based labels (b) Frame-based labels

Figure 7.3: BAC for individual classes and model types. Performances are averaged over all test scenes and all test files. Lines depict the range of performances over temporal context length. Sound types are sorted from top to down by increasing difference between the best (over model types) minimum and maximum (over context size) performance.

Lines depict the range of performances over temporal context length. Sound types are sorted from top to down by increasing difference between the best (over model types) minimum (over context size) and best (over model types) maximum (over context size) performance, averaged over both labeling modes. That is, the classes at the top profited least from additional temporal context, and the classes at the bottom profited most.

The degree to which the models are able to increase performance by including more temporal context varies considerably between the different sound event types: from piano with a difference of 0.025 to crash with a difference of 0.085. This influence does not seem to be correlated with the sound event's general detectability; actually, no clear relationship is identifiable.

Not shown here, for the LASSO and TCN models it is very class-dependent which temporal context leads to the best performance. For example, with the phone class, the best LASSO and TCN frame-based models use a 1 s and 0.33 s temporal context, with the knock class, they use 0.5 s and 20.49 s temporal context, and with the fire class, they use 10 s and 10.25 s temporal context. The same effect does not hold for the LDNN models: for basically all sound classes, the longest temporal context makes up the best performing model — one of LSTM's most advertised features is that one does not need to know a priori what context size is appropriate.

For the phone class, and (less pronouncedly) for the alarm and piano classes, the LASSO models perform stronger than the DNNs. In general, this is not to be expected, and implies that there are features that the LASSO models can use that the DNNs failed to learn — specifically, these likely may be the spectral features (cf. Section 7.1.2), which were selected with high weights by the LASSO algorithm particularly for the mentioned sound classes.

On the two speech classes, the DNN models reach very high detection performance – close to 0.95 –, when considering that these values are averages over all scenes (including SNRs down to −20 dB and up to 4 simultaneously active sources).

7.3 INFLUENCE OF SCENE CONFIGURATION

The presented all-scenes average results showed that using longer temporal context considerably helps increasing detection performances. The following analyses are supposed to show in which situations it helps particularly.

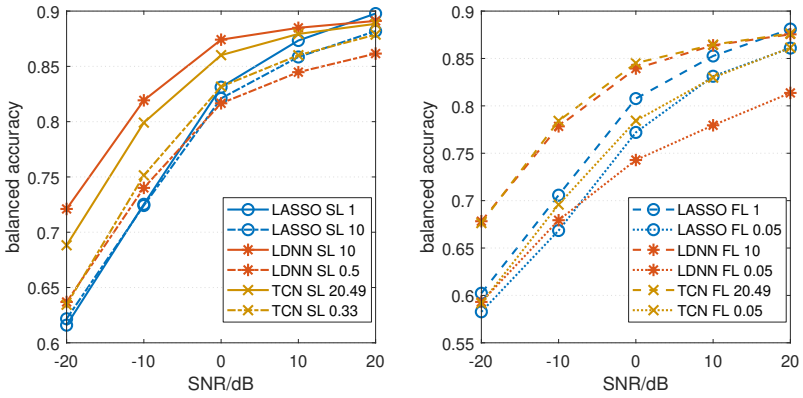
To investigate the factors influencing performance, three main scene configuration parameters were varied systematically across scenes: the SNR between target and distractor sources, the number of distractor sources, and the scene mode.

7.3.1 *SNR*

The performance over different SNRs is presented in Fig. 7.4. For all SNRs, the same azimuth configurations are aggregated, hence the SNR is the only parameter varied. For each model type, the best and the worst models' performances are plotted. The legend indicates the respective temporal context sizes.

Very clearly it can be observed that the influence of temporal context is larger for lower SNRs: particularly for situations with lower target than distractor(s) energy, the difference between worst and best models increases up to about 0.11. Comparing performances "horizontally", the best models have a detection advantage of about 10 dB. For SNRs above 10 dB, performances almost converge. For both labeling modes, the worst LDNN models are a bit different in that they are not able to catch up in performance for higher SNRs — their performance seems to be "globally" worse.

Intuitively, it is reasonable that for harder listening situations, it would be advantageous to be able to listen longer and integrate information to come to a conclusion. One part contributing to this may



(a) Segment-based labels (b) Frame-based labels

Figure 7.4: Balanced accuracies depending on model type, temporal context, and SNR. Depicting performances of each model class' best and worst model over SNR. Performances are averaged over all respective test scenes, all test files, and all classes. Lines display arithmetic means.

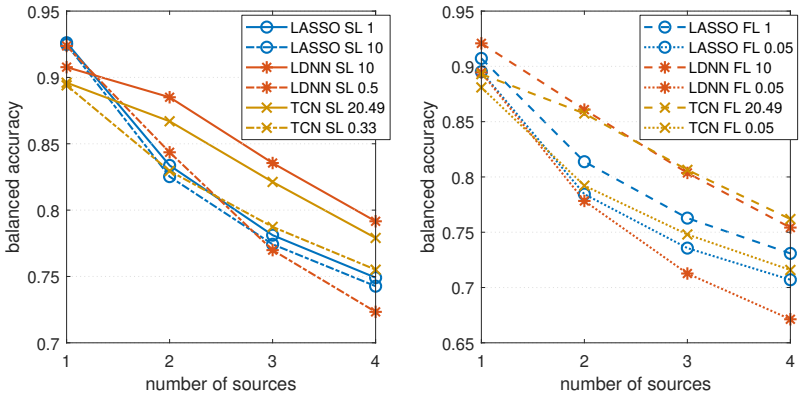
be "glimpsing" (Cooke 2006) – focusing recognition on segments less impeded by distractor noise, if it varies over time –, which is known behavior for humans when recognizing speech in noisy situations.

Interestingly, the best LASSO models reach the highest performance of all models for SNR of 20 dB. Two explanations seem possible: for almost "clean" conditions and hence less advantage for the temporal-context-integrating models (the DNNs), the spectral features (which the LASSO models use) come into play again and give them an edge. Or the DNN models optimized their feature extraction more with respect to the difficult scenes, and this came at the cost of slightly reduced performance for unimpeded sound events.

7.3.2 *Number of sources*

Fig. 7.5 presents the performance of the models over different number of co-occurring sources. Similar behavior as regarding the SNR can be noted: for easy scenes (only one active source), model performances are much closer to each other — increased temporal context is not necessary/helpful[4]. For more difficult scenes with simultaneously ac-

4 Guillaume et al. (2004) studied the time needed for humans recognizing isolated sound events — results showed that for a lot of sound events, as short as below 150 ms was enough, and no sound they tested needed more than 675 ms (averaged across subjects).



(a) Segment-based labels (b) Frame-based labels

Figure 7.5: Balanced accuracies depending on model type, temporal context, and number of sources. Depicting performances of each model class' best and worst model over number of sources. Performances are averaged over all respective test scenes, all test files, and all classes. Lines display arithmetic means.

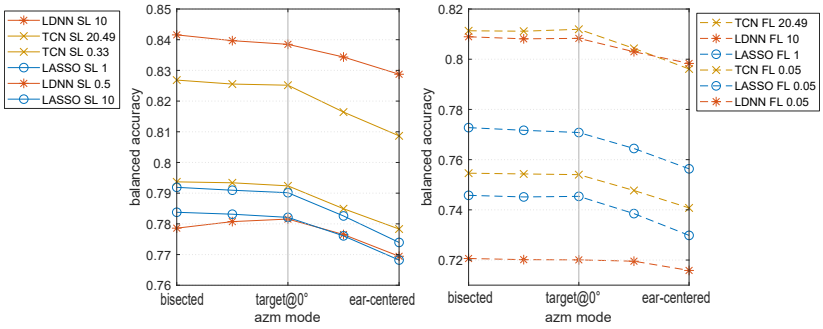
tive sources, the performance differences increase. However, there is less clear gradation of this performance difference; between 2, 3, and 4 sources, the difference between worst and best model does not change very much. Also, the difference is lower than for difficult SNRs, with about 0.07 in BAC that the 10 s LDNN can catch up on.

Again, for the segment-based labeling mode and the easiest situation (one source), the strongest model turned out to be the LASSO, and the explanation given there is equally applicable here.

The LDNN with 0.5 s temporal context (segment-based) exhibits, when compared to the plotting over SNR, curious behavior: while for easy SNR, it was worst and for difficult SNR it was bad, its performance on one-source scenes is, together with the LASSO, highest. It seems that this model specialized on situations with only one source active and is perturbed even by slight noise introduced through additional sources.

7.3.3 *Sources position*

The performances of the models depending on source positions are presented in Fig. 7.6, showing performances of the best and worst models over scene mode. Two conclusions can be drawn: (i) the influ-



(a) Segment-based labels (b) Frame-based labels

Figure 7.6: Balanced accuracies depending on model and azimuth mode (cf. Section 7.1.1). Performances are averaged over test scenes with different SNR and number of sources, all test files, and all classes. Depicting performances of each model class' best and worst model over scene mode.

ence of temporal context is not dependent on the source positions — performance differences are (almost) constant over azimuth modes. (ii) Detection performances are worse when sources are distributed around the ear on one side of the head only. After the findings presented in Section 5.3, this was expected, but the effect found for the models developed here was smaller. This likely can be accounted to the extension of the multi-conditional training up to four co-occurring sources, which makes specialization to favorable source distributions more difficult than in the two-source case.

7.4 DISCUSSION OF DNN RESULTS

Conclusions on performance differences between TCNs and LDNNs should be drawn conservatively — it does look like TCNs may deal better with short temporal contexts, and that LDNNs may be able to extract more relevant information from long temporal contexts. But while the differences between models' performances are significant, the random search in hyperparameter optimization does introduce chance, and insufficient number of samplings could result in arbitrary advantages for the best found combinations of the one or the other model.

As mentioned in Section 7.1.4, the hyperparameter optimizations for both the LDNN and TCN models were performed under computational constraints (particularly the available GPU hardware), leading to an under-sampling of the various parameters. Hence, it is possible that

more search would find hyperparameter combinations that made TCN models perform as well as LDNN on long contexts, and LDNN perform as well as TCN on short contexts. Of course it is also possible that LDNN simply are more efficient at learning from long contexts (10 s correspond to 1000 frames – quite a lot of time steps) than TCN.

Unfortunately, due to the resource-wise limitations, it was also not possible to also train a 20 s LDNN model, so whether performance would increase further – as is the case for TCN –, is unknown. In general, additional models with even longer context sizes should have been trained, since up to 20 s, TCN performance is still increasing.

Concerning the TCN models, a problem-specific convolution dilation sequence (in contrast to exponentially increasing with the layer number) or feature map size (instead of using the same for all layers) might lead to improved performance particularly for large temporal context lengths. Instead of the disregarded weight normalization other techniques for optimization stabilization such as batch normalization (Ioffe and Szegedy 2015) or layer normalization (Ba et al. 2016) could be included.

Concerning the LDNN models, individual number of neurons for different layers and applying LayerNorm (an LSTM batchnorm alternative) might improve performance. The batch sizes and learning rates (usually highly dependent on each other) could have been optimized rather than fixing values.

With both LDNN and TCN, a potential performance gain was missed by ignoring locality/invariances along the feature dimensions of frequency and modulation frequency (for amplitude modulation spectrograms). It would be possible (and likely profitable) to use two- and three-dimensional convolutional kernels for the TCNs (Espi et al. (2015) and Piczak (2015a) used 2D-kernels, for instance), and additional one- and two-dimensional convolutional layers with the LDNNs as described in Xingjian et al. (2015). The same was discussed in Huy Phan et al. (2016). The sound type-specific results presented above (Section 7.2.1) indeed indicate a shortcoming in this regard for several classes.

An advantage of TCN over LSTM is its increased parallelization, allowing for faster training, since the convolution kernel can slide independently along the temporal dimension while the LSTM graph traversing is an inherently sequential operation. Apart from the better performance in the presented experiments, the LSTM has the benefit of not needing to optimize the used context size individually per sound type — the TCN architecture showed more dependence in this aspect, cf. Section 7.2.1.

It has to be mentioned that the performance gains observed by including longer temporal context certainly are only possible for sound events that either inherently are as long, or are repeated over the time, as is the case for the constructed scene-instances in this study (cf. Section 2.2.1). The first naturally is not the case for all events (but for a substantial amount, it is), the latter occurs regularly, but not always. Concerning the sound types tested in this study, only for knock, scream, and crash, doubts on the realism of both assumptions could be cast[5].

7.5 SUMMARY

In this chapter,

- the sequence modeling capabilities of LSTM and its recent competitor TCN have been confirmed for sound event detection; both outperformed LASSO models by a significant margin.
- it has been shown that models able (and allowed) to integrate information over long time can increase detection performances considerably particularly for acoustically demanding polyphonic situations with distracting sources exhibiting higher energy than target sources.
- perceptually-motivated smooth segment-based labeling has been compared with "instantaneous" frame-based labeling, and the latter shown to be the more difficult problem, but qualitatively not different.
- it was demonstrated that multi-conditional modeling smoothly extends beyond two-source scenes by incorporating up to four simultaneous sources and to other model types; random multi-conditional training scene parameter sampling produces effective sound event detectors. The impact of spatial source distribution found before is still there, but attenuated.

Concluding, building sound event detection models for continuous realistic polyphonic scenes should employ architectures able to exploit long-range temporal context information like LSTM or TCN for reaching maximum performance. Training should be conducted multi-conditionally and over sufficiently long scenes of at least 10 s length.

5 On the other hand, depending on the situation, if a knock or scream would not be heard, they may be repeated, of course.

Part III

ROBUST SPATIAL SOUND EVENT DETECTION

If there is a crying baby and a fire, we want to know *where* the baby is.

8

MULTI-CONDITIONAL SOUND EVENT DETECTION ON SPATIAL STREAMS

This chapter is based on Ivo Trowitzsch, Christopher Schymura, et al. 2019. "Joining Sound Event Detection and Localization Through Spatial Segregation." In review for IEEE/ACM Transactions on Audio, Speech, and Language Processing, *arXiv preprint arXiv:1904.00055*.

Two key issues in computational auditory scene analysis (CASA) are (a) detecting sound events and their types within that stream, which was addressed in Part ii of this thesis, and (b) localizing the corresponding sources emitting the sounds, denoted sound source localization (SSL). In this chapter, the combination of the two is investigated: *joint sound event localization and detection (SELD)*. For comprehensive understanding of acoustic (or any) scenes, it is not only necessary to know what is there and where there is something, but instead to know *what is where*.

There are four different fundamental approaches to joining sound event detection and source localization:

1 – TEMPORAL CORRELATION Associating type and location of sounds through temporal correlation. This however is not possible for multiple sounds starting at the same time; and difficult for moving sources. If sources move (temporarily) to the same location, tracking gets lost.

2 – SOUND-TYPE MASKED SSL Attending to streams related to individual sound events. This "focus" can be created through masking the input such that a particular sound known to be active is "passed through" to SSL, and other sounds or noise are suppressed. Such masking is feasible in time-frequency domain if sound events exhibit specific frequency signatures. The subsequent localization then produces locations associable to these events. However, not all sound event classes exhibit coherent (and narrow) frequency signatures – for instance "alarm" is more of a semantic class, and can range from electronic beeps to fire bells; or "piano" ranges from very low to high frequencies. Also,

the approach is likely to fail for co-occuring sound events with similar frequency patterns.

3 – SPATIALLY MASKED SED Attending to streams related to individual source locations. This implies masking of the input such that only sound from a particular direction is passed through to sound event detection (SED), and sound from other directions is suppressed. Such masking is technically doable in time domain through beam-forming, or in time-frequency domain through spatial segregation, attributing individual time-frequency-bins to particular directions. Sound events detected on the spatial streams are then associated with a location. Efficient masking depends on spatial separation of sources and hence dissimilar spatial cues.

4 – JOINT SELD Building models that by construction detect localized sound events. Such models do not localize and detect separately or subsequently (as in approaches 2 and 3), but instead produce joint attributes from the start. While – because of the implicit combination of approaches 1-3 – this approach should in principle be the most powerful, it also requires the most powerful model, more difficult to train and to understand.

Approach *3*, detecting sound events on spatially segregated streams, was chosen to be followed in this study. The employed spatial segregation model, computing softmasks in time-frequency space as similarly described in Harishkumar and Rajavel (2014), Kolossa and Orglmeister (2004), and Ma et al. (2018), serves as a processing step for associating auditory features later used for sound event detection with specific sound source locations.

Compared to approach *2*, sound event detection on spatial streams has the advantage of enabling localized identification of multiple sources of the same type or with similar frequency ranges active. Compared to approach *4*, this approach is feasible also with less powerful models classes, faster to train, and easier to understand. Furthermore, systems following approach *3* are modular, which enables work and research on the individual components.

The spatially masked auditory features are used for sound event detection in the introduced scheme of multi-conditional training (Section 3.2.2 and Chapters 5 and 7). Although in Chapter 7 SED models based on deep neural network (DNNs) have been built with superior performance, for this study lasso models were used again, because they are very easy and fast to train and test and have shown to produce decent performance with the employed features. The focus was to

rather perform and provide extensive tests and qualitative analysis of the system instead of demonstrating the best performance possible.

The system as presented here is a proposition of how to join sound event detection and localization, as well as how to analyze and measure performance of such a system. Fig. 8.2 depicts the system and its information flow.

In this chapter, Section 8.1 firstly describes particularities and extensions to Chapters 2 and 3 with respect to data and methods concerning this chapter. Study results are presented then starting in Section 8.2, which elaborates on overall performance and the principal practicability of the method; Section 8.3 analyzes dependence on acoustic scene configurations (signal-to-noise ratio (SNR), number of sources, and spatial distribution of sources). In Sections 8.4 and 8.5, the influence of perturbation of information about number of active sources and their locations fed to the stream segregation model is investigated.

8.1 METHODS AND DATA

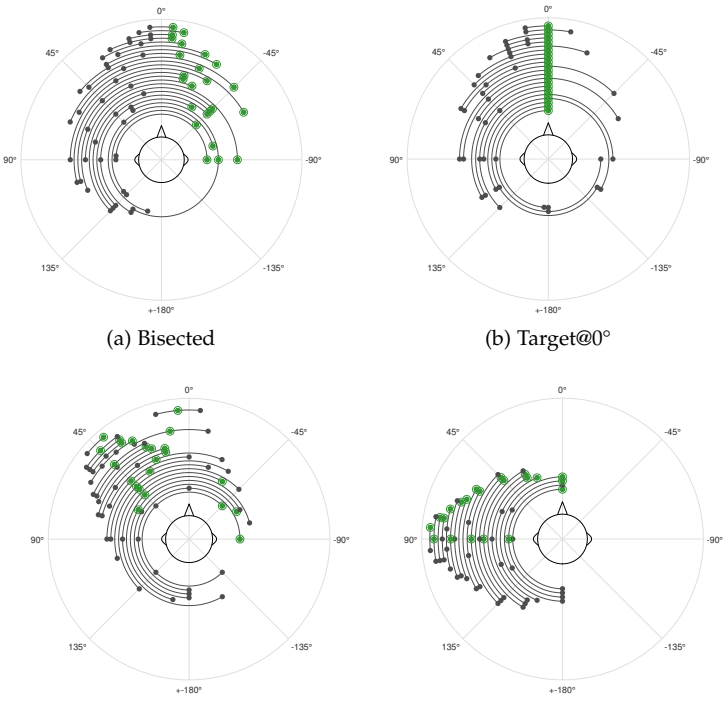
The basic methods and data introduced in Chapters 2 and 3 are also used here. Anything specific to the experiments described in this chapter in this regard is elaborated on in the following sections. Additionally to the description of the auditory scenes rendered for training and testing, particularly the inclusion of the spatial stream segregation model into the sound event detection process, which implies some important changes and amendments to methodology used in the chapters before, is explained.

8.1.1 *Auditory Scenes*

The original sound data for synthesizing auditory scenes was taken from the NIGENS database, described in Section 2.1. For methodology and terms regarding the scene rendering, confer Section 2.2.

For training the detection models, the set defined in Section 7.1.1 was used. The testing set defined in there was extended significantly for higher coverage of spatial source distributions: *468 scenes* were defined with the following parameters varied:

- the number of sources (one to four)
- the SNRs between target and distractor sources (−20 dB, −10 dB, 0 dB, +10 dB, +20 dB)
- the azimuth difference between sources (0°, 10°, 20°, 45°, 60°, 90°, 120°, 180°)



(a) Bisected (b) Target@0°

(c) Front-left (d) Ear-centered

Figure 8.1: Test scene configurations (only scenes with at least two sources), sorted into the different *scene modes*. Black filled circles depict distractor sources, target sources (green) are highlight by an enclosing open circle. Each scene is indicated by one circle fragment. The head is at the center.

- the "scene mode" (depicted in Fig. 8.1):
  1. *bisecting*: the nose (0° azimuth) points between target and distractor source(s)
  2. *target@o*: the nose points towards the target source
  3. *front-left*: sources are mainly between 0° and 90°; they are not bisected and targets are not at 0°, and they are not symmetric around the ear
  4. *ear-centered*: sources are distributed in the left hemisphere symmetrically around the ear (90°)
- the position of the target among the sources: either at one end, or (only for three-source scenes[1]) at the center.

1 Only for three-source scenes to save computation time.

All together, 96 different azimuth configurations are used among the 468 scenes. Exact definitions of training and test scenes can be found in Trowitzsch (2019).

8.1.2 *Spatial stream segregation model*

A spatial stream segregation model developed in the Two!Ears project and described in Ma et al. (2016) and Trowitzsch, Schymura, et al. (2019) was used.

Blocks of interaural time-differences (ITDs) and interaural level-differences (ILDs) in time-frequency-representation (cf. Section 2.3), as well as the estimated number of active sources with corresponding locations serve as inputs to the segregation model. It produces a softmask weighting factor for the i-th source at time-step k and frequency-channel l according to

$$m_{kl}^{(i)} = \frac{p(\tilde{\boldsymbol{y}}_{kl} \,|\, \boldsymbol{g}(\phi_i),\, \boldsymbol{R}_l)}{\sum_{j=1}^{M} p(\tilde{\boldsymbol{y}}_{kl} \,|\, \boldsymbol{g}(\phi_j),\, \boldsymbol{R}_l)}, \tag{8.1}$$

given a set of M estimated azimuthal source locations $\{\phi_i\}_{i=1}^{M}$ and an observation model for the binaural observations $\tilde{\boldsymbol{y}}_{kl} = \begin{bmatrix} \tau_{kl} & \delta_{kl} \end{bmatrix}^T$ (ITD (τ_{kl}) and ILD (δ_{kl}) cues at time frame k and frequency channel l).

This observation model is implemented through generalized linear models (GLMs) (Dobson 2002), trained from anechoic binaural auditory scenes generated employing the head-related impulse response (HRIR) also used for SED model scene synthesis (see Section 2.2.2, Wierstorf et al. (2011)). White noise was used as source signals for this training.

An example of softmasks produced by Eq. (8.1) and the general information flow concerning the stream segregation stage and its embedding into the segregated detection system is depicted in Fig. 8.2.

8.1.3 *Detection model input*

Features for the models were constructed as described in Section 3.1.1.1. As base auditory representations, ratemaps, spectral features, and amplitude modulation spectrograms were used, described in Section 2.3. The three representations were split into overlapping blocks of 500 ms length, with a shift of 333 ms. The mean-channel feature set, as defined in Section 4.1.2, was constructed from these representations.

Segment-based labels as described in Section 3.1.2.1 were produced for each feature vector.

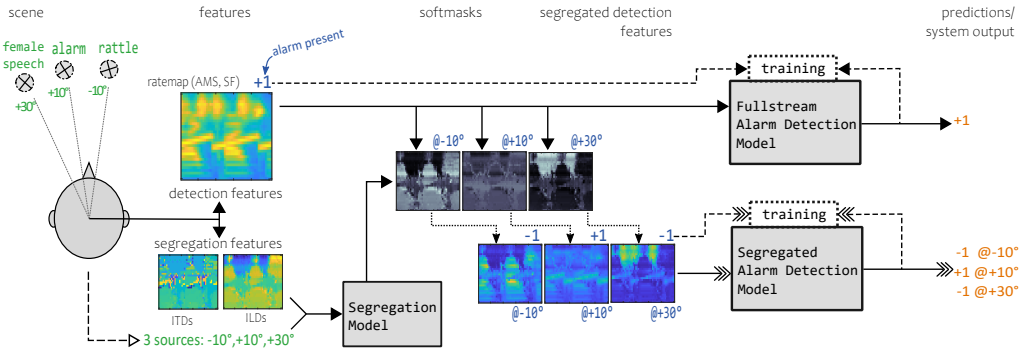


Figure 8.2: From binaural scenes to localized detections. Exemplary scene with three sources, at $-10°$ (emitting female speech), $+10°$ (alarm sound), and $+30°$ (rattling sound). From ear-signals, detection and segregation features are computed and cut into blocks of 500ms (amplitude modulation spectrograms (AMS) and spectral features (SF) omitted for clarity). Together with input about number of active sources and their azimuths (ground truth at training time, systematically perturbed or ground truth at testing time for the analysis; estimated or set values in a deployment system), the segregation model produces one softmask for each spatial stream, that is, for each azimuth. Each softmask gets applied to the detection features, such that one set of features is formed per stream. Labels about the presence of target sound events (alarm, in this case) are attached according to the associated azimuth. Segregated features are then input to the segregated detection models, which are trained to predict the presence of target events in the passed blocks (in the depicted example, all predictions are correct). Fullstream models, not part of the segregated detection but complementary and for comparison, get non-masked detection features and detect the presence of target events in the full mixture.

8.1.3.1 *Segregated detection model input*

Segregation into spatial streams takes place after the generation of the different auditory representations and their segmentation into blocks (see above), and before the construction of mean-channel feature vectors from the blocked auditory representations.

The segregation model produces a set of probabilistic time-frequency softmasks (see Section 8.1.2), with the number of masks corresponding to the number of currently active sources in the scene[2]. A source was defined active in a block if its mean energy in that block was above -40 dB of its whole scene-instance maximum energy.

2 Actually of the number of locations with active sources (active spatial streams) — two sources at the same location have to count as one.

These masks were applied (through multiplication) to the ratemaps and amplitude modulation spectrograms; spectral features were afterwards computed from the masked ratemaps. Hence, one set of masked representations per spatial stream was produced, such that one mean-channel feature vector per spatial stream could be generated. Fig. 8.2 summarizes the data processing steps.

Since each mask was generated based on a presumed location of active sound source(s), each mask is associated with this particular location. Ergo, each feature vector for the detection model is attributable to this location — and hence each detection on this feature vector, which is why in the following the output/activity of the segregated detection models is also called localized detection.

Both for training the detection models and testing their performance, labels indicating the presence or absence of target sound types are needed. If a block was labeled negative before segregation, all segregated blocks were labeled negative. If a block was labeled positive before segregation, the segregated feature vector associated to the location *closest* to the target source was labeled positive, and the others negative.

Note that through this labeling for the segregated detection, there emerge effectively *three* kinds of sample types: negative samples from mixture blocks in which the target sound event was not present, positive samples from mixture blocks in which the target sound event was present, *and negative* samples from mixture blocks in which the target sound event *was active*. The respective negatives in the following are sub-indexed npp or pp for "no-positive-present" or "positive-present".

8.1.4 *Model training*

Two types of models were trained: *fullstream* detection models, operating on the full (mixed) stream, and *segregated detection* models, operating on the segregated streams, features, and labels as described above. Apart from this difference in input and a difference in sample weighting (elaborated on below), both model types were trained identically, as described in Sections 3.2 to 3.4 and below.

For each of the thirteen target sound types defined by NIGENS, binary one-vs-all classifiers were trained with GLMNET multi-conditionally (cf. Section 3.2.2 and Chapter 5) using the 80 training scenes (see Section 8.1.1) for all models in this study. Models were trained on one training-test-set split, with the training sets consisting of 75 % of the sound files.

For each model training, 200 000[3] samples were sub-sampled across all scenes. The sub-sampling was done as described in Section 3.2.3; for segregated detection, the sample weights (Eq. (3.3)) had to be amended because of the differentiation of the two effective types of negative samples: the weights of positive samples $(x_i, y_i = +1)$ were set as before to

$$w_i = 1/(N_{L(i)=+1,nas(i)} \cdot N_{si(i)}) \,, \tag{8.2}$$

but the weights of negative samples $(x_i, y_i = -1)$ were set to

$$w_i = 1/(2 \cdot N_{LPP(i),nas(i)} \cdot N_{si(i)}) \tag{8.3}$$

where $LPP(i)$ indicates whether the sample was generated from a block in which a positive was present or not, respectively (see Section 8.1.3.1).

8.1.5 *Model testing*

While training was conducted multi-conditionally, tests were performed on individual scenes in order to conclude on relations between scene parameters and performances.

Blocks in which not all sources were *active* (since sources emit sounds that also exhibit silences), got removed to better reflect the influence of the number of sources. All remaining samples from the test set were used without further sub-sampling, amounting to about 12 million samples tested with each fullstream model and about 30 million with each segregated detection model.

As described in Section 8.1.2, the segregation model needs input on the number of active spatial streams and on their azimuths. For training, ground truth knowledge was used. For testing, three different modes were implemented:

1. Using ground truth for both data.
2. Using ground truth for the number of active streams, but perturbing the location information of those streams. This perturbation was conducted by adding random azimuth values drawn from a normal distribution with sigma of 5°, 10°, 20°, 45°, and 1000° to each block's location. The latter basically corresponds to drawing locations uniformly. Note that the sources' locations in the scenes were *not* changed, but *only the information about them* given as input to the segregation model, and consequentially the azimuth associated with a block.

3 As the Lasso model has few free parameters (number of features + 1), this amount was enough. Actually performance saturated even before.

Table 8.1: Spatially segregated detection measures & nomenclature overview

PURE DETECTION	
BAC_{sw}	Stream-wise balanced accuracy (used for training). Mean of $SENS_{sw}$ and $SPEC_{sw}$
$SENS_{sw}$	Stream-wise sensitivity. Positive detection rate
$SPEC_{sw}$	Mean of $SPEC_{pp}$ and $SPEC_{npp}$
$SPEC_{pp}$	Specificity (negative class accuracy) of blocks (streams) that do not contain a target event, but at times at which in *another* stream, the target event *is* active
$SPEC_{npp}$	Specificity of blocks that do not contain a target event, at times at which the target event is *inactive in all* streams
BAC_{tw}	Time-wise segregated detection models' balanced accuracy. Mean of DR_{tw} and $SPEC_{tw}$
DR_{tw}	Time-wise segregated detection models' detection rate; aggregated over streams.
$SPEC_{tw}$	Time-wise segregated detection models' specificity (negative class accuracy); aggregated over streams.
BAC_{fs}	Fullstream models' balanced accuracy. Mean of DR_{fs} and $SPEC_{fs}$
DR_{fs}	Fullstream models' detection rate (positive class accuracy).
$SPEC_{fs}$	Fullstream models' specificity (negative class accuracy).
LOCALIZED DETECTION (all conditioned on target events being active *and* detected)	
$BAPR$	Best-assignment-possible rate. Proportion of sound event detections in the best-available (azimuth-wise) stream, but in no other stream
NEP	Number of excess positive assignments — amount of streams with false positive event detections
$AzmErr$	Mean azimuth error. Averages the azimuth distance of all positive-assigned streams to the correct azimuth
Placement likelihood	Depicts the average proportions of event detections in streams depending on their distance to the event's correct azimuth

3. Using ground truth for the locations of active streams, but perturbing the data about the active source number. A uniformly drawn random number between -2 and $+2$ was added to the number of streams ground truth (thresholding downwards at 1). In case of a reduction, the respective number of locations handed to the segregation model was removed randomly. In case of an increase, locations drawn randomly from a uniform distribution between 0° and 360° were added to the segregation model input.

8.1.6 *Performance measurement and evaluation*

Training and testing performance was measured utilizing balanced accuracy (BAC) respectively its constituents sensitivity and specificity, cf. Section 3.3.2.

For segregated detection training, BAC had to be adjusted: with this method, there exist many negative samples of points in time without positive present (as with fullstream negative samples), but there also exist negative samples of points in time with a positive present in another spatial stream, see Sections 8.1.3.1 and 8.1.4. These, however, have a much lower proportion than the npp negatives and would, if not up-weighted, have minor influence on training. This, consequently, would result in worse localized detection performance, because of too low cost of not discriminating between target and distractor streams. Thus, BAC_{sw} (sw for stream-wise) was defined for segregated detection:

$$\begin{aligned} BAC_{sw} &:= 0.5 \cdot SENS + 0.5 \cdot SPEC_{sw}, \text{ with} \\ SPEC_{sw} &:= 0.5 \cdot SPEC_{pp} + 0.5 \cdot SPEC_{npp}. \end{aligned} \tag{8.4}$$

While BAC_{sw} summarizes performance in one number so that the models can be optimized, it is difficult to gain insight into the actual behavior of the models through that number. Two different aspects of segregated detection performance are interesting: time-wise detection performance, and localized detection performance.

8.1.6.1 *Time-wise detection evaluation*

To evaluate how well the system recognizes sound events irrespective of location and to compare performance to fullstream sound event detection models, time-wise measures were used, namely BAC_{tw}, mean of detection rate DR_{tw} and specificity $SPEC_{tw}$. To obtain these, the segregated detection models' predictions over streams were *aggregated* for each point in time: a positive prediction in any stream produces an aggregate positive prediction. Hence, an aggregate negative prediction is constituted only if all streams are predicted negative. It is obvious that this can lead to an increase of the number of true positives as well as of false positives (shown and discussed in Sections 8.2 and 8.3.2).

The subindex tw indicates time-wise aggregate segregated detection performance, fs indicates fullstream models' performance.

8.1.6.2 *Localized detection evaluation*

To evaluate how well the system assigns detected sound events to the localized streams, four measures were established that provide understanding of the behavior when a sound event is *present and detected*:

Table 8.2: Generalization: stream-wise performances on full training and test set, averaged over classes and all scenes

PERFORMANCE	TEST SET MEAN	TRAINING SET (CV) MEAN
BAC_{sw}	0.777	0.806
$SENS_{sw}$	0.775	n/a
$SPEC_{pp}$	0.649	n/a
$SPEC_{npp}$	0.837	n/a

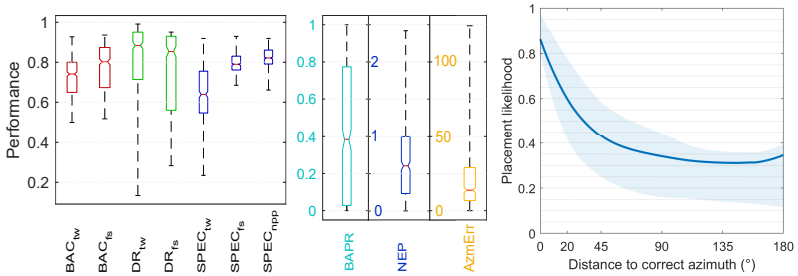
- The *placement likelihood* measures the average proportion samples are getting assigned positives, depending on their associated distance from the sound event's correct azimuth. Ideally, the placement likelihood would be 1 at the correct azimuth, and 0 everywhere else[4].
- The *best-assignment-possible rate (BAPR)* describes how often the system assigns a positive to the stream with associated location *closest* to the true azimuth, and *only* to this stream. For unimpaired source-count and location input (see Section 8.1.5), the closest stream is always the one with correct azimuth; for perturbed situations, it may well be a stream with azimuth distance greater than 0°.
- The *number of excess positive assignments (NEP)* indicates how many streams erroneously got assigned a positive. Ideally, this would be zero.
- The *mean azimuth error (AzmErr)* averages the distance of all positive-assigned streams to the correct azimuth.

Table 8.1 provides an overview over measures and nomenclature for easy reference.

8.2 EVALUATION: METHOD PRACTICABILITY

Training produces functional models, as Table 8.2 shows. BAC_{sw} on the test set (averaged such that scenes with 1,2,3,4 sources have equal weight, as in training) is only a bit below training performance and well above chance level. ($SENS_{sw}$, $SPEC_{pp}$ and $SPEC_{npp}$ constitute BAC_{sw}, cf. Section 8.1.6.) Since different sounds and different scenes are used in the test set compared to the training set, this performance demonstrates successful generalization of the models.

4 Only if the correct azimuth is actually always among the segregated streams, that is, for unperturbed data.



(a) Time-wise detection (b) Localized detection (c) Placement likelihood

Figure 8.3: Grand average (full test set, all test files, all classes) performances. Time-wise performances (a) are ignorant of location, providing detection performance *aggregated* over streams for segregated detection models, and comparing to fullstream models' detection performance on the full mix. Localized detection performances (b),(c) present measures regarding detection in the *correct* stream (that is, associated to the correct location). Box-plots indicate the 25th to 75th percentiles, the median and its 95 % confidence interval, whiskers depict the complete range of values. The placement likelihood plot (c) displays the arithmetic mean and, shaded, the 25th to 75th percentiles. Table 8.1 or Section 8.1.6 provide descriptions of the presented measures.

Disassembling the surrogate performance number BAC_{sw}, in Fig. 8.3a, time-wise performances (as introduced in Section 8.1.6.1) of segregated detection are given and compared to fullstream detection. While being in the same range, the fullstream models do exhibit higher balanced accuracy. This is due to a notably worse specificity of the segregated detection models (median of 0.66, i. e. one out of three times, without a sound event present, the system actually assigns a positive to one or more streams) for which the better detection rate (median of 0.9, i. e. nine out of ten times, when a sound event is present, it is also detected) can not make up. This has to be carefully interpreted (see Section 8.3.2), since the additionally depicted underlying *stream-wise* $SPEC_{npp}$ of the segregated detection models is actually even a bit higher than the fullstream's specificity.

The actual purpose of the segregated detection models is assigning sound events to the correct spatial stream. Figs. 8.3b and 8.3c show different indicators in this regard:

- Looking at the placement likelihood[5], a sound event placement is most likely in a stream at the correct azimuth. This likelihood quickly drops with increasing azimuth distance up to around 60°. The ideal system would produce a peak at 0° only, but the graph shows that the method produces assignments more likely to be close to the true azimuth than far from it.

- The best-assignment-possible rate (BAPR) is, in the median, about 40 %. That is, for the more difficult half of the scenes, between 0 % and 40 % of the event assignments are made to the correct stream (and only to it). For the easier half of the scenes, between 40 % and 100 % of the assignments are made to the correct stream (and only to it). The wide range indicates that scenes differ a lot in how well they can be segregated into localized streams; which is analyzed and discussed in Section 8.3.

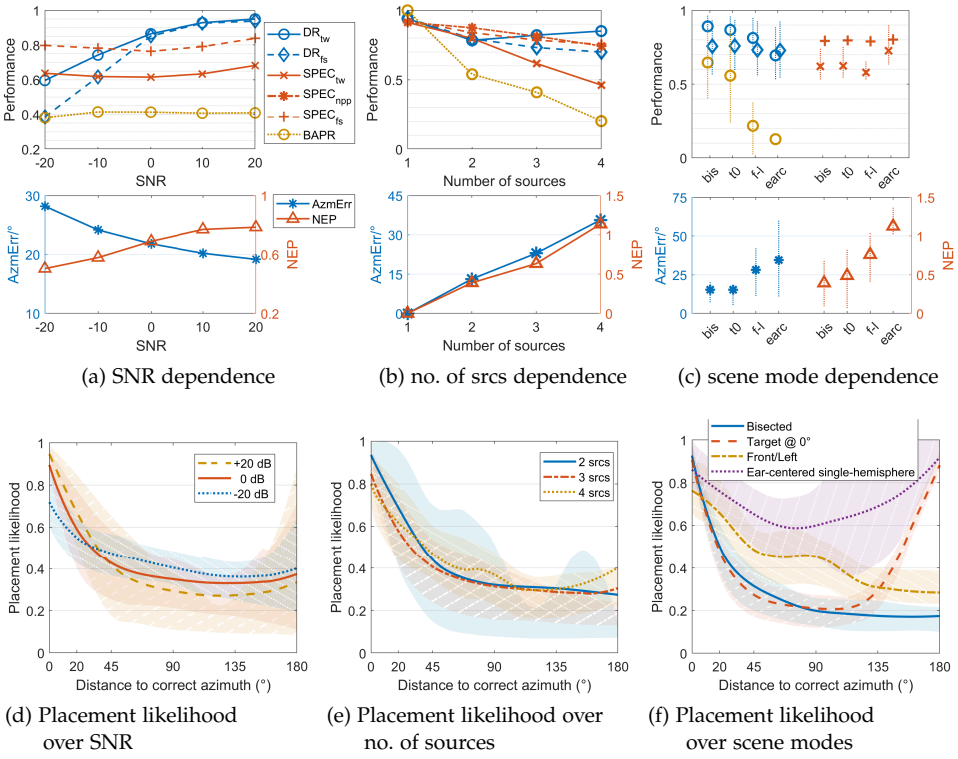
- The median azimuth error (AzmErr), giving the mean distance between the true sound event's azimuth and the azimuths of its assigned streams, is about 13° and ranges from 0° to 125°. This low average deviation is consistent with the placement likelihood plot, showing that most assignments are done close to the true azimuth.

- The median number of excess positive assignments (NEP) is 0.6. For about 25 % of the scenes, only one stream is assigned a positive (which is ideal), but for the larger part of scenes, assignments to more than one stream occur frequently. Looking at the azimuth error, at least these excess assignments usually happen to streams close to the true azimuth.

8.3 INFLUENCE OF SCENE CONFIGURATION

The presented all-scenes grand average results exhibit a very wide range of performance. To investigate the factors influencing performance, three main scene configuration parameters are varied systematically across scenes in the following: the SNR between target and distractor sources, the number of distractor sources, and the scene mode.

5 This graph reads like: if there was a stream located at 20° distance to the true sound source's azimuth, the mean proportion of blocks from this stream getting a positive sound event assignment would be 0.6.



(a) SNR dependence (b) no. of srcs dependence (c) scene mode dependence

(d) Placement likelihood over SNR (e) Placement likelihood over no. of sources (f) Placement likelihood over scene modes

Figure 8.4: Performances depending on SNR (a),(d), number of sources (b),(e), and scene mode (c),(f), averaged over all respective test scenes, test files, and classes. Line plots display arithmetic means and, shaded, 25th to 75th percentiles. Table 8.1 provides descriptions of the presented measures.

8.3.1 *SNR*

The performance of the system over different SNRs is presented in Figs. 8.4a and 8.4d. For all SNRs, the same scene configurations are aggregated, hence the SNR is the only parameter varied.

Detection rate and specificity show typical behavior — DR_{tw} dropping with SNR, $SPEC_{tw}$ remaining mostly constant. Notable are the differences between segregated detection and fullstream models: the offset between specificities remains the same, while the detection rate differs only for difficult SNRs, where the segregated detection models perform better.

The number of positively assigned streams increases with SNR, because the stronger target source dominates the time-frequency space

and more likely overrides ITDs and ILDs of the weak distractors. On the other side the mean azimuth error decreases with SNR, implying that — even though with more of them — the positive assignments at high SNRs are closer to the correct azimuth. The placement likelihood graph reflects this as well: up to about 30° azimuth distance, higher SNRs produce more (percentage-wise) assignments. Above about 30°, it reverses and higher SNRs of the target produce less assignments.

8.3.2 *Number of sources*

The performance of the system over different source counts is presented in Figs. 8.4b and 8.4e.

A clear negative correlation between number of sources and performance values can be observed, with the notable exception of the detection rate, which counter-intuitively increases slightly from two to four sources. For one and two sources, detection rate and specificity of segregated detection and fullstream models are very similar. The time-wise segregated detection $SPEC_{tw}$ however decreases for higher source counts much more strongly than $SPEC_{fs}$ — while the *block-wise* $SPEC_{npp}$ shows almost exactly the same behavior as $SPEC_{fs}$. This implies that the model's general ability to classify negatives is actually not lower than that of the fullstream models, and leads to the assumption that the reason for both the strong decrease in time-wise specificity as well as for the increase in detection rate is actually the *successful segregation* into streams — which eases detection of positives, be they true, or be they false, due to sound similarity. In a *mix* of active sources, any positive (true or false) is less likely detected (this is shown by the detection rate of the fullstream model), but the segregation (to a certain extent) un-mixes. Since all sounds apart from the target class sounds are emitted from all distractor sources, higher number of sources mean higher probability of (false) positive occurrences. Hence, the time-wise aggregation over streams produces an increase in detection rate through true or false positives, and a decrease in specificity through false positives. This is an effect deemed practically unavoidable. In order to re-balance performance between time-wise detection rate and specificity to increase precision, it may be an option to adjust the training performance measure (BAC_{sw}) such that the weight of specificity is increased beyond 0.5.

The indicators of localized detection performance, $BAPR$, $AzmErr$ and NEP, all show lower performance for higher number of sources. This is to be expected, since more sources imply more overlap in time-frequency-space and thus less distinct segregation masks. In the placement likelihood graph, this is difficult to observe, because the

means are very similar, but it can be noted by looking at the shaded indications of the 25th to 75th percentiles.

8.3.3 *Scene mode*

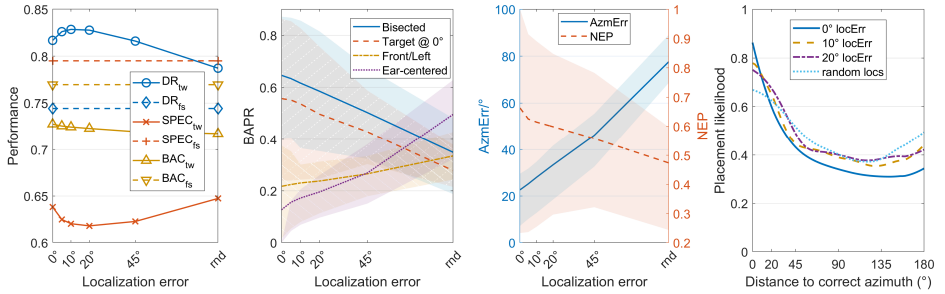
The performance of the system for the four different scene modes (cf. Section 8.1.1) are presented in Figs. 8.4c and 8.4f. A clear gradation can be observed, with the bisected and target@0 modes performing best, front-left scenes performing worse and ear-centered-single-hemisphere scenes performing by far worst. This holds for all performance indicators apart from specificity. Although for the fullstream models the scene mode is far less influential (since the spatial features are not used there), on a much lower scale the same pattern can be noted for the detection rate.

For scenes in bisected or target@0 modes, $BAPR$ is high and $AzmErr$ is low (about 60% of all cases with the optimal assignment, and around 15° mean azimuth error), and few excess positives are assigned.

Particularly the latter increase strongly for the other two modes (negatively correlating $BAPR$) due to the increased occurence of front-back-confusions. Front-back-confusions emerge because of the (approximate) front-back-symmetry of the head, which leads to similar spatial features for azimuths symmetric to the ear axis (Ma et al. 2017). The employed segregation model (Section 8.1.2) for this reason actually disregards differences between front and back at all, in favor of more robust segregation in the frontal hemisphere; any ear-symmetric scene hence must produce equal softmasks and result in the same classification of the symmetric streams.

The placement likelihood graph shows these effects very clearly. The bisected mode shows a curve close to the ideal, while the ear-centered curve demonstrates a severe lack of discrimination between locations for event assignments. Scenes with target at 0° show similar behavior as bisecting ones, but exhibit front-back-confusion approaching 180°.

While SNR and number of sources are unchangeable attributes of a given scene, the scene mode *is* changeable by head rotation. At least in a scene with sources changing positions slower than the head can turn, it should be possible to notably increase performance by choosing the head orientation such that the sources of interest are spread as wide as possible throughout the frontal hemisphere, optimally bisected. This is in accordance with results about dynamically improving localization performance in a binaural robot system (Ma et al. 2017).



(a) Time-wise performances (b) BAPR over scene modes (c) Segregated detection performances (d) Placement likelihood

Figure 8.5: Grand average (full test set, all test files, all classes) performances depending on strength of perturbation of location information fed into the segregation model. Localization error is given as standard deviation of the Gaussian perturbation added onto true azimuths, "rnd" standing for "random". Line plots display arithmetic means and, shaded, 25th to 75th percentiles. Table 8.1 or Section 8.1.6 provide descriptions of the presented measures.

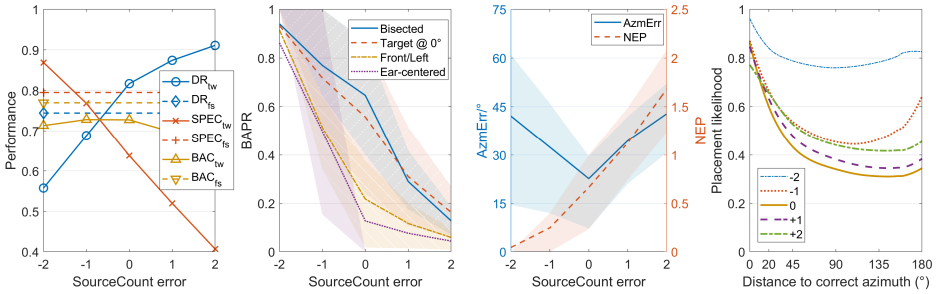
8.4 DETECTOR PERFORMANCE AND LOCALIZATION DEVIATION

The segregation model relies on knowledge of two scene configuration attributes: the number of active sources and the locations (azimuths) of those sources. For the results above, both model inputs have been fed with ground truth. Since a real system likely would not (always) produce correct information about these two aspects, experiments were conducted with systematically perturbed values.

This section analyzes the influence of perturbation of the location input. The locations fed into the segregation model have an added random Gaussian component (see Section 8.1.5) with different variances between 5° and 45°. Additionally, tests were performed with completely random azimuth input. For each individual variance, models were tested again on all test scenes and sound files and analyzed as before.

Fig. 8.5 shows the performances over localization error. Looking at the time-wise detection performance, it is notable that detection rate and specificity behave inversely, and both only change on a small scale (about ± 0.03).

The segregated detection performance indicators show stronger dependency on localization. The best-assignment-possible rate of the four different scene modes (cf. Sections 8.1.1 and 8.3.3) converge toward similar (low) values with increasing localization error — particularly the



(a) Time-wise performances (b) BAPR over scene modes (c) Segregated detection performances (d) Placement likelihood

Figure 8.6: Grand average (full test set, all test files, all classes) performances depending on strength of perturbation of source count information fed into the segregation model. Line plots display arithmetic means and, shaded, 25th to 75th percentiles. Table 8.1 or Section 8.1.6 provide descriptions of the presented measures.

two well-performing modes (bisecting and target@0) decrease strongly. Interestingly, with random localization, the ear-centered scene mode exhibits the best *BAPR*, standing out from the other three modes. This is because for target sources at 90°, which occur in this scene mode, the probability of any spatial stream with random location getting a similar mask is least (highest for 0°). This can not lead to head turning rules of course, because with random localization, a robot would not know how to position sources at the ear.

Since the performance order of the four modes remains stable up to very high localization errors, the head orientation guiding principle deduced in Section 8.3.3 stays valid, albeit with lower resulting performance gain.

The straight increase of *AzmErr* with localization error is logical — actually, the localization error does not even fully add to the system-inherent (at 0° localization error) azimuth error of about 22°.

8.5 DEPENDENCE ON NUMBER OF SOURCES ESTIMATION

After localization error, the impact of incorrect input of the number of active sources was analyzed. To this end, an error of ±2 was added to the source count and accordingly produced streams by the segregation model.

Fig. 8.6 shows performance over source count error — it is apparent that deviations from the correct number of streams bear strong

performance changes. Time-wise detection rate and specificity show anti-correlated behavior: for underestimation of number of sources, the detection rate degrades heavily, for overestimation of the source count, specificity drops even more.

Azimuth error and placement likelihood show that segregating into the wrong number of streams in both directions leads to worse localized detection performance. The azimuth error rises with any deviation: with too few streams, because the correct stream may be omitted, with too many streams, because segregation becomes more difficult and, as can be noted looking at NEP, because more excess positives are assigned.

The latter is also comprised in the strong decrease of $BAPR$ for source count overestimation — any case of excess positive assignment is not a best-possible assignment. The increase of $BAPR$ for negative source count error is no indicator of somehow better localized detection performance, but a mere logical consequence of the fact that with number of sources underestimation, scenes with two, respectively three, sources become segregated into one stream only, in which case the best-possible assignment is trivial.

Clearly the implication of these results is that the segregated detection system as proposed is dependent on an accurate estimation of number of active sources.

8.6 SUMMARY

In this chapter,

- a method to annotate binaural sound scenes with joint sound event type and location information by combining spatial segregation in time-frequency-space with robust sound event detection on the segregated streams in training and testing has been suggested and evaluated.

- it was demonstrated that this approach can produce localized sound type information under a broad range of auditory conditions. The localized detection performance depends particularly on the number of active sources in the scene, and on their spatial distribution. By turning the head such that the sources of interest are in the frontal hemisphere (and at best bisected by the nose), the system's performance in many situations can be increased strongly.

- it was found that proper estimation of the number of active spatial streams is a precondition of this approach; localization error does not influence segregated detection as heavily.

- a diverse set of test scenes has been defined for thorough study of performance behavior, together with a suitable training performance measure and several test performance indicators to enable capturing different qualitative aspects of the joint behavior of the combined models.

It can therefore be concluded that the proposed method for joint sound event localization and detection could be one core component of a binaural scene analysis system.

9

EMPLOYMENT IN THE TWO!EARS SYSTEM

All studies evaluated in the chapters before have been on simulations of *static* auditory scenes. One of the strengths (and goals) of the Two!Ears project was development of *active* auditory models, and the developed blackboard system hence supported models working with feedback loops and commanding of robot actions.

In Chapter 8, evaluation of the proposed sound event localization and detection (SELD) system was performed with ground truth information about source locations and number of sources. In the study detailed in the following, actual localization and number of sources estimation modules were employed as input to the spatial segregation system.

Furthermore, a new head turning module trying to maximize spatial separability was developed and is described below. Results presented below show that this head rotation strategy indeed increases segregated sound event detection performance significantly.

All in all, in this final study, the models developed in this thesis were employed in the Two!Ears blackboard system and active robust binaural computational auditory scene analysis was demonstrated.

In Section 9.1, the Two!Ears system and employed components as well as the acoustic scenes used are introduced, Section 9.2 presents the obtained results.

9.1 METHODS AND DATA

The study described in this chapter involves no model training, but only testing. Methods with respect to auditory data generation are used here as put in Chapter 2. The description of the training of the sound event detection (SED) and SELD models has been given in Chapter 8. In Sections 9.1.2 and 9.1.3, the test execution environment – the Two!Ears blackboard system – and its setup is described (so far, tests had been executed in the Auditory Machine Learning Training and Testing Pipeline (AMLTTP) (Appendix A)); Section 9.1.1 details the auditory scenes used for the testing.

Table 9.1: Scenes overview for model evaluation in Two!Ears development system. Cf. Fig. 9.2 for the layout of the ADREAM room. Fig. 9.1 depicts the azimuth configurations. "Num Srcs/act/pred" refers to the scenes' set-up number of sources, the actual mean active number of sources over time, and the mean predicted number of sources by the number of sources estimation module.

ROOM ACOUSTICS	NUM SRCS/ACT/PRED	SOURCE AZMS	SNR (dB)
anechoic	2/1.4/1.0	10°, 110°	0
	3/2.0/1.2	10°, 110°, −160°	
	4/2.7/1.4	10°, 110°, −160°, −45°	
anechoic	2/1.4/0.9	−5°, 55°	0
	3/2.0/1.1	−5°, 55°, 115°	
	4/2.7/1.3	−5°, 55°, 115°, 175°	
anechoic	2/1.4/1.0	180°, 150°	0
	3/2.0/1.2	180°, 150°, 120°	
	4/2.7/1.5	180°, 150°, 120°, 90°	
ADREAM room,	2/1.4/1.8	srcs@(pos4, pos2)	0
head@pos2	3/2.0/2.2	srcs@(pos4, pos2, pos1)	
	4/2.7/2.3	srcs@(pos4, pos2, pos1, pos3)	

9.1.1 *Auditory scenes*

Twelve auditory scenes were used in this study; four with two, three, and four sources each. Anechoic scenes rendered with the head-related impulse response (HRIR) from Wierstorf et al. (2011) as well as reverberant scenes rendered with the binaural room impulse response (BRIR) from Winter, Wierstorf, Podlubne, et al. (2016) are included (see Section 2.2.2 about binaural scene synthesis). Some scenes have sources distributed all around the head, some contain sources more densely packed. The average signal-to-noise ratio (SNR) between sources was always set to 0 dB. Table 9.1 and Fig. 9.1 detail the configurations of the auditory scenes used in this study.

Sounds from all classes of NIGENS (cf. Section 2.1) that had not been used for training of any of the included models, that is, only from the test sets, were used for scene rendering (245 files, 77 of which from the "general" sound class). Different to testing in the studies described in the chapters before, these files were randomly concatenated into four continuous streams (instead of the scene-instances produced so far for each original sound file). Random silences of 2 s to 10 s between concatenations were inserted to have the number of *active* sound sources vary. Prior to concatenation, sound files shorter than 10 s were looped until this threshold was reached. Each of the four streams, which were

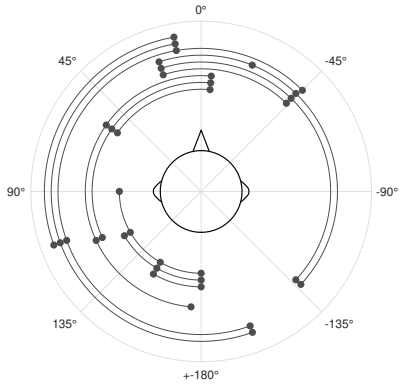


Figure 9.1: Two!Ears system employment test scenes. Black filled circles depict sources, each scene is indicated by one circle fragment. The head is at the center. For system configurations with head rotation modules, the above depiction is the starting situation.

subsequently used as input to the scene rendering, had a length of 1 h : 10 min.

9.1.2 Two!Ears *blackboard system*

The Two!Ears blackboard system was developed in the Two!Ears project basically as a middleware providing a platform for easy implementation of auditory modules in a dynamic binaural system; as similarly described for example in Ellis (1996) and Godsmark and Brown (1999). This system connects the acquisition of ear-signals with basic auditory data processing, and then mediates data between these basic stages and higher-level modules as well as between these higher-level modules. The system is dynamic, since modules can change basic auditory processing and command actions of the system host (real or simulated robot). It is described in detail in Ma et al. (2014, Ch. 3).

Two core modules are always part of the Two!Ears blackboard system:

- The binaural simulator (Winter, Wierstorf, and Trowitzsch 2016, Ch. 2.2) (or the robot interface, in case of robot deployment), which produces ear-signals from definitions of acoustic scenes (cf. Sections 2.2.1 and 2.2.2).
- The Auditory Front-end (AFE) (May, Decorsière, et al. 2015), which produces different auditory data representations of the ear-signals (cf. Section 2.3).

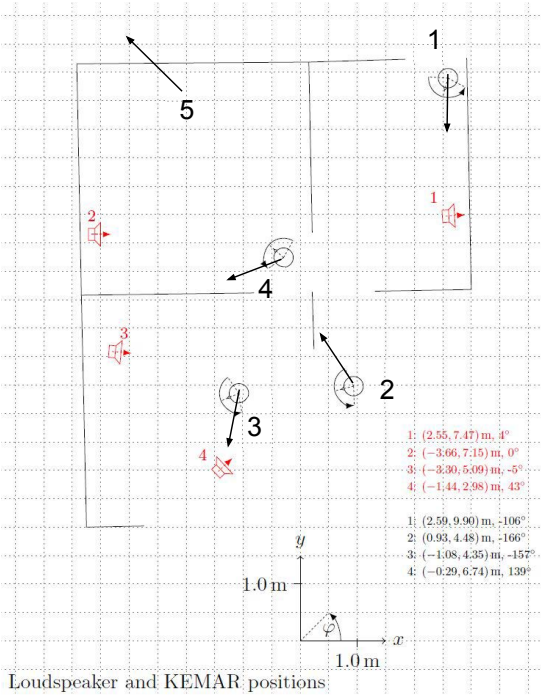


Figure 9.2: Layout of positions and orientations of the robot and the loud speakers in the ADREAM apartment (Winter, Wierstorf, Podlubne, et al. 2016) used during recordings. The arcs around the positions indicate the orientation ranges at which the BRIR were recorded.

During execution of the set-up blackboard systems, all data and results produced by models – i. e., source localizations, active source count estimations, sound event detections, etc. – get saved, enabling later analysis.

9.1.3 *Blackboard system setup*

To perform the investigation of spatially segregated SELD with actual source localization and different head rotation strategies, four different variants of the blackboard system were set-up in this study:

1. No head rotation — the head is fixed in the starting position of the scenes. Ground truth about number of currently active sources is known.

2. Random head rotation. The head is rotated arbitrarily irregardless of locations of currently active sources. Ground truth about number of currently active sources is known.

3. Maximum lateral distance head rotation, the head is rotated to maximize lateral separation of currently active sources. Ground truth about number of currently active sources is known.

4. Maximum lateral distance head rotation and number of currently active sources estimation.

The following components were thus employed in the blackboard system for execution of above setups:

- Sources localization module and localization confusion solver module as described in Ma et al. (2017) and Ma et al. (2016, Ch. 3.4.1). The underlying localization model is a deep neural network (DNN) architecture trained on anechoic data, and produces a likelihood of source activity over time for all azimuths based on interaural level-differences (ILDs) and left-right ear signal cross-correlation features. The localization confusion solver module integrates source location likelihood distributions prior and posterior to head rotations in order to eliminate phantom sources.

- (only setup 4) Number of active sources estimation module as described in Ma et al. (2016, Ch. 3.5.2), which was trained on auditory scenes from the ADREAM room (Winter, Wierstorf, Podlubne, et al. (2016), Fig. 9.2) and uses DUET features (Rickard 2007), interaural time-differences (ITDs), ILDs, and the source location likelihood distribution output of the source localization module.

- Spatial stream segregation module as described in Section 8.1.2 and Trowitzsch, Schymura, et al. (2019), taking ILDs, ITDs, the source location likelihood distribution output of the source localization module, and the number of active sources estimation output by the respective module or ground truth about the active source count as input.

- Spatially segregated SELD modules: as described in Chapter 8, one for each target sound class.

- Fullstream SED modules: as described in Chapter 8, one for each target sound class.

- (only setup 2) Random head rotation module. The name says it all — commands random head rotations (with a rotation speed of maximally $40° \, s^{-1}$).
- (setups 3 and 4) Maximum lateral distance head rotation module, described below in Section 9.1.3.1.

All of these modules processed consecutive 500 ms blocks of auditory data; blocks did not overlap in this study.

9.1.3.1 *Maximum lateral distance head rotation*

Analysis of localized detection's dependence on spatial distribution of sources in Section 8.3.3 led to the conclusion that best performance can be achieved by positioning the head such that sources are spread as wide as possible throughout the frontal hemisphere. However, the employed spatial segregation model (Trowitzsch, Schymura, et al. 2019) does not differentiate between frontal and dorsal hemisphere, and uses an observation model mapping azimuthal locations to ITDs and ILDs through a generalized linear model (GLM) of sine functions. Therefore, it is suggested here that the head be oriented to maximize *lateral distances* between sources, irregardless of whether they are in the frontal or dorsal hemisphere.

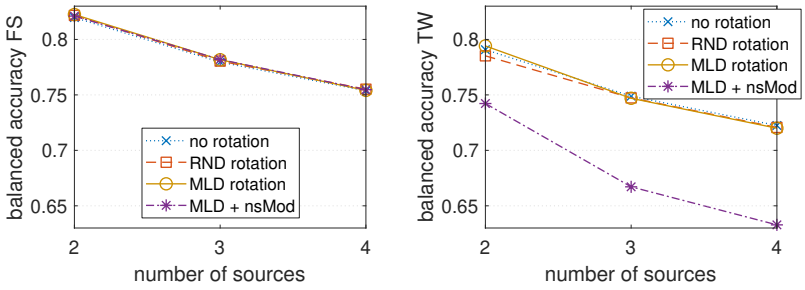
To test this method, which shall be called maximum lateral distance (MLD) head rotation in the following, a TWO!EARS blackboard system module implementing this rule was developed. This module is executed after every new block of data and subsequent estimation of sources location distribution, i. e. in the case of the described experiments, every 500 ms.

Since true azimuths of sources are unknown, the sources location probability distribution over azimuths p_L as estimated by the localization module then is the basis for calculation of lateral distances: for each possible head orientation, calculate lateral distance $latDist$ with respect to p_L as well as their mean longitudinal position $\overline{longPos}$ as

$$latDist = \sum_{a_1=1}^{360} \sum_{a_2=1}^{360} |sin(a_1) - sin(a_2)| \cdot \min(p_L(a_1), p_L(a_2)) \tag{9.1}$$

$$\overline{longPos} = \sum_{a=1}^{360} p_L(a) \cdot \cos(a). \tag{9.2}$$

The maximum lateral distance (MLD) orientation is then defined as the orientation maximizing $latDist$; for equal $latDist$, it is the orientation additionally maximizing $\overline{longPos}$ (i. e. orientations with sources in the front are preferred over orientations with sources in the back).



(a) BAC_{fs} (b) BAC_{tw}

Figure 9.3: Fullstream and aggregated time-wise balanced accuracies in Two!Ears system employment over number of sources with different system components (without head rotation module, with random head rotation module, with MLD head rotation module, with MLD head rotation and number of sources estimation module). Line plots depict averages over all respective scenes and 13 target sound classes.

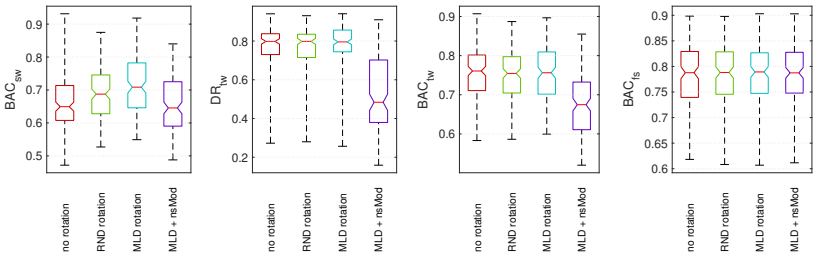
The head is then rotated towards the MLD orientation, but with a rotation speed of maximally $40° \, s^{-1}$. Since the used localization module seems to be particularly prone to front-back confusions for sources at azimuths of 0° or 180°, as a heuristic, the module avoids such positioning and instead slightly shifts a bit.

9.2 RESULTS

For each of the above described blackboard system configurations (Section 9.1.3), for each of the above described scenes (Section 9.1.1), the saved detection model output was individually evaluated and compiled into the performance measures introduced in the previous chapters (Sections 3.3.2 and 8.1.6). These results were aggregated (i) into grand averages over all classes and scenes and (ii) into averages over classes and scenes with the same number of sources, i. e. scenes with two, three, or four sources.

9.2.1 *Pure detection performances*

Figs. 9.3, 9.4c and 9.4d show balanced accuracies (BACs) of the fullstream and segregated detection models. Box-plots indicate the 25th to 75th percentiles, the median and its 95 % confidence interval, whiskers depict the complete range of values. Line plots indicate arithmetic means.



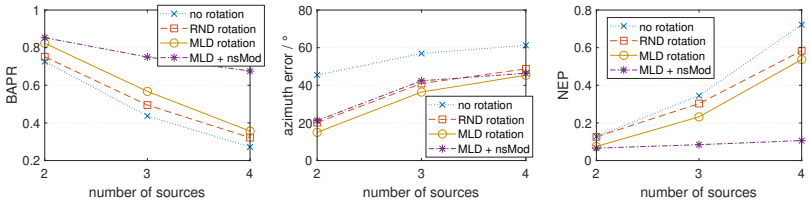
(a) BAC_{sw} (b) DR_{tw} (c) BAC_{tw} (d) BAC_{fs}

Figure 9.4: Average localized detection performance measures in Two!Ears system employment with different system components (without head rotation module, with random head rotation module, with MLD head rotation module, with MLD head rotation and number of sources estimation module). Boxplots pool performances over all scenes and 13 target sound classes.

Firstly, these results show that the models efficiently work not only in the testing environment of AMLTTP, but also in the continuously streaming "live" environment of the Two!Ears blackboard system. Secondly, it is observable that performances of the fullstream model are completely unaffected by the different head rotation strategies and blackboard setups. This is more or less to be expected; more, because the fullstream models in principle are independent (because of the good generalization across azimuth configurations) of head orientation, and less, because evaluation in Section 8.3.3 had predicted small advantages for azimuth configurations with sources distributed in the frontal hemisphere, which the MLD head rotation strategy enforces. However, the advantage was very small, and the MLD rotation due to restricted head rotation speed (see Section 9.1.3.1) does not have the head at optimal orientation always.

Looking at the pure detection BACs over number of sources, behavior is as reported in Section 8.3.2, with the time-wise segregated detection performances being a bit lower than the fullstream performances.

Notable is the significantly lower time-wise detection performance with the number of sources model active (setup 4, Section 9.1.3), which, inspecting Fig. 9.4b, can be traced back to a very low detection rate for this setup. In Table 9.1, the average predicted number of sources are reported next to the true average active number of sources — it is apparent that the used model is strongly underestimating the active source count for the anechoic scenes (it was trained on ADREAM scenes). In Section 8.5, a strong degradation of detection rate was



(a) *BAPR* (b) *AzmErr* (c) *NEP*

Figure 9.5: Localized detection performance measures in Two!Ears system employment over number of sources with different system components (without head rotation module, with random head rotation module, with MLD head rotation module, with MLD head rotation and number of sources estimation module). Line plots depict averages over all respective scenes and 13 target sound classes.

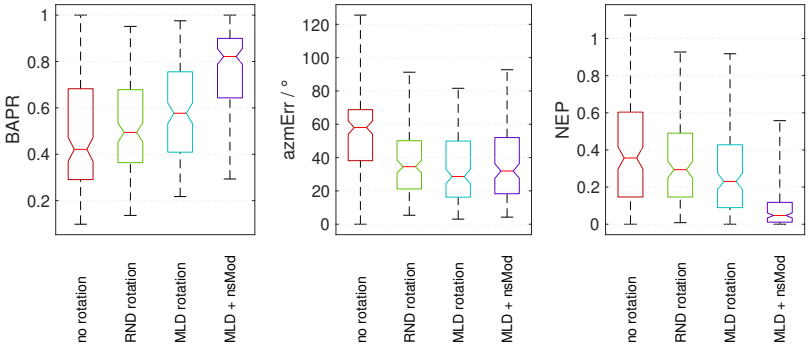
predicted for underestimation of number of sources, but the observed loss is even higher, and not compensated by increased specificity. This may be due to the fact that in Section 8.5, source locations were deleted randomly for source count underestimation, but in actual application, the azimuths with lowest localization likelihood are omitted. These probably have lower energy and are less likely detected in a "wrong" spatial stream than more energetic sources.

9.2.2 *Localized detection performances*

The stream-wise BAC was used as a localized detection performance surrogate for training of the segregated detection models in Chapter 8 and Section 8.1.4. While it does not allow insight into specific behavior of the models, it does provide an overall performance measurement.

Fig. 9.4a displays the grand average stream-wise BAC for the four different system setups and head rotation strategies. The worst performance was achieved by setup 4, using number of sources estimation instead of ground truth; the low detection rate of this setup was already discussed above.

With respect to the different head rotation strategies, BAC_{sw} shows clearly a significant increase going from no rotation over random rotation to MLD rotation. Figs. 9.5b and 9.6b show that for the mean azimuth error, the dependency is even stronger, specifically going from no rotation to random rotation. The large difference between these two modes – although there would be no reason to assume that the segregated detection performs significantly stronger with random head orientation than with fixed head – indicates the large gain in localiza-



(a) *BAPR* (b) *AzmErr* (c) *NEP*

Figure 9.6: Average localized detection performance measures in Two!Ears system employment with different system components (without head rotation module, with random head rotation module, with MLD head rotation module, with MLD head rotation and number of sources estimation module). Boxplots pool performances over all scenes and 13 target sound classes.

tion accuracy introduced through integration of pre- and post-rotation localizations of the confusion solver employed (cf. Section 9.1.3, and Ma et al. (2017))[1]. With MLD rotation, the mean azimuth error can be further reduced in the median by about 8°, which is in the predicted range of the evaluation in Section 8.3.3.

The behavior of best-assignment-possible-rate (BAPR) (Figs. 9.5a and 9.6a) and number of excess positives (NEP) (Figs. 9.5c and 9.6c) similarly reflect the advantage of the MLD rotation. The BAPR in the median is increased by about 10 %, the median NEP is reduced from about 0.28 to 0.22. These performance differences show for scenes with two, three, or four sources (almost) equally.

As observed and discussed already in Section 8.5, the underestimation of the active source count model leads to high BAPR and low NEP, which is simply a consequence of the subsequent segregation into fewer spatial streams.

9.3 SUMMARY

In this chapter,

1 Comparing Fig. 9.5b and Fig. 8.5c, an average localization error of somewhere between 20° to 45° could be inferred with the random head rotation module active.

- it was demonstrated that the fullstream and segregated detection models efficiently detect sound events in a dynamic online binaural system with actual sound source localization instead of ground truth.
- it was validated that performances in the Two!Ears blackboard system were well predicted in the evaluation of tests performed in the (static) AMLTTP in Chapter 8.
- it was suggested a new head rotation strategy that maximizes lateral source separation, thereby optimizes spatial segregability, and which significantly increased localized detection performance.

Concluding, it was shown that the models developed in this thesis indeed robustly detect sound events both on full- and spatially segregated streams and constitute efficient components of a binaural system performing computational auditory scene analysis.

Part IV

DISCUSSION AND CONCLUSION

For what it's worth.

10

DISCUSSION

Many aspects of the presented work have been reasoned about already throughout the analyses in Parts ii and iii and placed into existing research in Chapter 1. In the following sections, several aspects are continued, topics spanning across the individual chapters are discussed, and further related literature is reviewed.

10.1 BUILDING ROBUST SOUND EVENT DETECTION MODELS

The development and analysis of polyphonic sound event detection (SED) by and large started to gain attention through the Detection and Classification of Acoustic Scenes and Events (DCASE) 2013 and 2016 challenges (A. Mesaros et al. 2018; Dan Stowell et al. 2015). Unfortunately, the level of polyphony was very low; to an extent making it hard to call scenes polyphonic[1], at least for the 2016 synthetic task — consequently, Lafay et al. (2017) in their analysis of results have not found performance differences between monophonic and polyphonic scenes.

The challenge tasks on real audio (DCASE 2016 and 2017) were much harder, with mean polyphony of 2.53 (excluding silences for the calculation). Results accordingly were much worse, leading A. Mesaros et al. (2018) and Annamaria Mesaros et al. (2019) to the conclusion that polyphonic SED on real audio still is difficult to tackle and data sets for training were still too small.

This thesis is the first to address robust sound event detection in polyphonic scenes with up to four co-occurring sources, and the first to present thorough analyses about the effects of the different dimensions of polyphonic sound scenes and systematic solutions about polyphonic model building.

1 At the hardest level, 55 sound events with average duration of about 1.5 s were distributed randomly along 120 s.

10.1.1 *Data*

One contribution of the work done in the course of this thesis was the creation and public deployment of the NIGENS database (Section 2.1, Trowitzsch, Taghia, et al. (2019b)). It is tailored to the task of synthesizing complex acoustic scenes through providing over 1000 sound files with isolated events and frame-level labeling, something which to this extent was unavailable so far (see Section 2.1.1). Particularly for realistic spatial scene creation, such a data set is indispensable: labeling of recorded real spatial audio with ground truth about sound types and source locations would be of prohibitive effort, and scene synthesis the only realistic viable option to obtain well-defined acoustic scenes of specific complexity.

Admittedly, real recordings of complete scenes would always be the gold standard; and for pure SED without localization, there exist the DCASE real audio data sets. For the multi-conditional training approach followed here and for the fine-grained analyses (which both rely on the capability to synthesize scenes along acoustic scene dimensions), these data sets were not suitable (although they even are real binaural data). But it remains desirable to additionally train (with adapted procedures) and test on these data sets as a comparison.

Using "general" sounds (cf. Section 2.1.2) instead of only target sounds is a clear distinction from other works, since the general class has high overlap with all target classes, and thus renders model definition (realistically) more difficult compared to a situation where class boundaries only have to be found against (usually limited amounts of) other target classes (with less overlap).

A shortcoming is the neglect of diffuse background noise in this work. While the focus intentionally was sound event detection in situations with simultaneously active distinct sources, in real life there are frequently situations with noise-like background without specific events, like noise from rain, or wind going through trees in a forest, or thousands of cars in the vicinity. In defense, there are noise-sounds like these in the general class, and there is the "fire" class, which exhibits the mentioned features as well. However, they were not used as diffuse (non-spatial) sounds, but emitted from point sources. Adding diffuse background noise to the proposed training and testing would be easy, and add an interesting additional dimension for analysis[2].

2 As a side-note and as described in Section 1.5.1, in most existing literature, it is the other way around: only background noise and no distinct disturbing other sound events.

10.1.1.1 *Unlabeled data*

While strongly labeled data with event on- and offset annotations is rare, there is an abundance of unlabeled or weakly labeled data available on platforms like YouTube or Freesound. Exploiting this wealth has a lot of potential for semi-supervised approaches, pre-training, or representation learning, which can be followed by supervised learning on labeled data. A few such approaches are described in the following.

In a very large-scale setup, Aytar et al. (2016) trained a convolutional neural network (CNN) on raw waveform from 2 million unlabeled Flickr videos, supervised by *visual* recognition networks. The resulting network (which they called "SoundNet") built a new internal representation for the sound input, which subsequently could be used for supervised sound classification training. Their results on DCASE and ESC data suggested superior performance compared to other models without the SoundNet pre-training[3].

Z. Zhang et al. (2017) have proposed to use a recurrent neural network (RNN) encoder-decoder trained on predicting audio sequences, subsequently extracted so-called bottleneck features from the RNN, and trained audio classification models on top of it. Again, results suggest strong performance, but unfortunately, tests only have been performed on individual sounds, instead of polyphonic situations.

Another pre-trained public deep neural network (DNN) model making available deep feature representations (learned from 100 million videos), called "VGGish", was released by Google (Hershey et al. 2017).

After unlabeled sound data, the second most available sound data is *weakly labeled* data, that is, audio clips with global tags without notion of when individual sound events occur. To make use of this data anyway for SED, Kong et al. (2019) have proposed a system consisting of two models, one CNN trained to segregate the time-frequency (TF)-space, and another one to identify the masked sound events. Their results indicate that this could be a valuable approach.

Because of its increasing importance, weakly labeled SED is now part of the DCASE series (Annamaria Mesaros et al. 2019).

10.1.2 *Multi-conditional modeling and Robustness*

The results presented in this thesis demonstrate that robust polyphonic sound event detection models efficiently can be built in data-driven ways, even with conventional models like the Least Absolute Shrinkage and Selection Operator (LASSO). This is in line with the general trend

3 But other results (Shan and Ren 2018) of models built on SoundNet implied worse performance than time-frequency energy-based counterparts.

towards data-driven over manual design, as also noticed by A. Mesaros et al. (2018) in their review of the DCASE 2016 challenge.

Multi-conditional modeling has been the means of choice in this work, and the such-trained models have proven to be robust at the dimensions of signal-to-noise ratio (SNR), number of sources and azimuth configuration, plus – separately – room acoustics. A study modeling multi-conditionally across all four dimensions, and showing simultaneous generalization should be done additionally.

That the robustness in polyphonic acoustic scenes obtained through multi-conditional training is fundamental, could be demonstrated by training models for many qualitatively different sound classes, with three different model types operating on two different kinds of representation, and two different labeling methods; all performing qualitatively similar with respect to the acoustic conditions.

For sound event detection, no other work compares multi-conditional and single-conditional models on matching and non-matching conditions[4]. However, for speech recognition, Yin et al. (2015) have presented results very similar to the ones presented here in Chapters 4 and 5. They demonstrated that training on all noisy and clean data together resulted in optimal performance across different (a priori unknown) conditions without deterioration compared to the matching single-conditional models; validating the evaluations here. Despite the facts that an extended SNR range, additional variability through azimuths of sources, less predictable noise (distinct versus background noise), and a less powerful model type was used here, the same generalization effects were generated.

Martin-Morato et al. (2018) recently investigated the robustness of features extracted from SoundNet, a DNN trained on millions of sound from video clips (Aytar et al. 2016) with respect to background noise and reverberation, similar to the analysis in Sections 4.2 and 6.2. Their results indicate severe drops in performance when testing in reverberant conditions, and very strong influence of background noise, even already for SNRs above 0. This accordance with results shown here confirms the assumption that model architecture and highly adapted features are not the main factors in robustness across acoustic conditions, but rather the training scheme. The conclusion presented in Martin-Morato et al. (2018) – that SoundNet's deep representation needs to be increased to get robust – is not shared here. In light of the outcomes provided, models using SoundNet's (or any other valuable) representation rather

4 A few do, but only monophonic with diffuse background noise, and with single-conditional models only trained for one (clean) condition, e. g. Dennis et al. (2012), Wu et al. (2018), and Haomin Zhang et al. (2015).

should still be trained multi-conditionally to avoid overfitting onto the acoustic condition.

Generally, it seems a promising approach to train multi-conditional models on labeled datasets like NIGENS or ESC (Piczak 2015b) with the deep representations (like SoundNet or variants tailored to SED with higher temporal resolution, Wang and Metze (2017)) obtained through unsupervised training from the wealth of available unlabeled data, see Section 10.1.1.1 above.

DATA AUGMENTATION Since the prevalence of deep learning accompanied with the need for lots of data in order to prevent overfitting, data augmentation is being used frequently; in DCASE Community (2019), there are basically no contributions without any form of data augmentation. Often, variants of time stretching or frequency shifting are employed, which mostly show small effects (e. g. Piczak (2015a) and Salamon and Bello (2017)). So-called "mixup" or "between-class" augmentation (Jeong and Lim 2018; Tokozume et al. 2017; Wei et al. 2018), borrowed from image recognition (Hongyi Zhang et al. 2017), is closer to the proposed multi-conditional training, since it also involves superposition of different training samples.

Inoue et al. (2018) and Takahashi et al. (2016) propose "Equalized Mixture Data Augmentation" and show a strong performance increase (over not using it with the same system). As with the here employed multi-conditional training, different sounds are mixed to increase sample variety. In contrast, they propose to mix *inside* each sound class; the intention was not to gain robustness with respect to acoustic conditions, but robustness with respect to intra-class variance. This approach could be well combinable with acoustical multi-conditional training.

10.1.3 *Model types*

The choice of LASSO (Section 3.4) as an SED classifier is uncommon, frequently used classifiers (before DNNs) for sound identification have been particularly support vector machines (SVMs), Gaussian mixture models (GMMs), and hidden markov models (HMMs) (Sharan and Moir 2016; Stiefelhagen et al. 2007; Dan Stowell et al. 2015). This choice was made for the following reasons: (i) the LASSO is very fast and easy to train with a very efficient tool (GLMNET), (ia) it is much faster to train than any non-linear method, (ib) linear models commonly all perform about similarly, (ii) it is extremely easy and fast in application, which can be useful with low performance hardware as in hearing aids, for example, but also was an issue with the Two!Ears robotic platform, (iii) it can efficiently deal with very high-dimensional data,

also with highly correlated variables, and (iv) it provides indications about variable importance through its L_1-regularization.

Certainly, the LASSO was a limit to performance in the presented investigations (as Chapter 7 – expectably – showed). However, non-exhaustive tests with nonlinear SVM (rbf kernels) and random forests had not improved performance, which led to the supposition that basic nonlinear information in the data was already extracted in the feature creation (Section 3.1.1.1).

Generally, the aim of this thesis was to rather perform and provide extensive tests and qualitative analyses of how to fundamentally tackle sound event detection, instead of demonstrating the highest performance possible. Definitely, using a potent, up-to-date model class, with the available computational resources a lot of the presented work would not have been possible to conduct. But to be honest: it was a (welcome) surprise that the LASSO models were able to perform as well on multi-conditional data across such wide ranges of acoustic scenes. This can only be interpreted highly in favor of the extracted perceptually motivated features.

Of course, algorithms providing temporal modeling capabilities were expected to increase performance, as information about sound identity is to a good extent encoded temporally. DNNs were the natural choice in this regard, as they have taken over from HMMs for several years now, particularly CNNs and RNNs (A. Mesaros et al. 2018; Annamaria Mesaros et al. 2019; Purwins et al. 2019; Xia et al. 2019).

10.1.4 *Features*

Two fundamental auditory representations are used in this thesis: ratemaps (and derived from it, spectral features) and amplitude modulation spectrograms (AMSs) (see Section 2.3).

Ratemaps belong to the group of mel-scaled frequency-filtered energy representations. In the latest three DCASE SED challenges (DCASE Community 2019; A. Mesaros et al. 2018; Annamaria Mesaros et al. 2019), mel-scaled energy representations are both the most used features as well as the winning features. They have thus taken over from the long time more popular Mel-frequency cepstral coefficients (MFCCs).

AMS, in contrast to the speech recognition domain[5], seem not to be used a lot in SED. This is surprising, since (a) evidence of its importance in human hearing is clear (Gygi et al. 2004; Luo et al. 2006; Shannon et al. 1995; Smith et al. 2002), (b) it is reasonable to assume that overlapping sound events in the same frequency band may still be separable to a

5 For example in T. May and T. Dau (2013), Mitra et al. (2014), and Moritz et al. (2015)

certain extent through their different amplitude modulation, and (c) the LASSO models trained here confirm these points by making strong use particularly of the AMS-derived features (identified by means of the L_1-regularized model coefficients; not shown here).

Some publications exist promoting spectro-temporal Gabor filterbanks, argued to capture not only temporal and spectral, but joint spectro-temporal modulations (Schröder et al. 2015). In a short experiment with ratemap, AMS, and spectro-temporal Gabor features, indeed a minimally higher performance compared to the feature set without the Gabor features was found. Unfortunately, a check whether the Gabor features could have actually replaced the AMS features, i. e., whether AMS in presence of spectro-temporal Gabor modulation features still conveyed unique information, was not done here.

Temporal information is summarized by computing statistical moments over time similar to Nogueira (2016), Nogueira et al. (2013), and H. Phan et al. (2015), and complemented with spectral summaries similar to Geiger et al. (2013), Marchi et al. (2016), and Nogueira et al. (2013).

The established model building pipeline would facilitate more experiments with respect to features, for example investigating the effect of a Dual-resonance non-linear filter bank (Meddis et al. 2001) or automatic gain control (Lyon 2017), both implying better (and more human-like) handling of signals with varying energy level.

Even more biologically-inspired (Smith and Lewicki 2006) features, namely spiking neural representations, are gaining popularity: Scholler and Purwins (2011), Wu et al. (2018), and Q. Yu et al. (2019) have utilized such features for audio classification, and Wu et al. (2018) and Q. Yu et al. (2019) additionally demonstrated multi-conditional training and very good performance with these features. Whether their results hold for more complex acoustic scenes than they investigated (monophonic + diffuse background noise), would be interesting.

The power of DNNs allows using raw waveform as input features; in DCASE 2017, a few challenge entries did so, and achieved decent, but lower performance compared to the systems using log-mel energies (Annamaria Mesaros et al. 2019). Dai et al. (2017) suggest that DNNs models indeed have to be very deep, that is, have many layers, to be able to efficiently perform SED on raw audio data. They find a model performing well – similar to a model operating on log-mel energies – with 18 layers. Most SED-systems still operate on spectro-temporal representations instead of raw audio (Xia et al. 2019). However, in principle, deep learning systems can learn feature representations tailored to the task. An intermediate approach could be to let DNNs operate on individual gammatone-filtered raw audio or inner hair-cell signals, alleviating

the network from the task of learning frequency-selective filtering, but leaving it with the possibility to find finer, task- and frequency-specific representations from the filtered audio. In an interesting work in a similar manner, Çakir and Virtanen (2018) started training a DNN with a weight-initialization such that the network already had in-built time-frequency representations, but during training could re-learn to whatever function most beneficial. However, results turned out to be worse compared to handcrafted time-frequency representations.

FREQUENCIES 80 Hz to 8 kHz is the frequency range employed in this work. This range was chosen because (a) it is a common choice (Sharan and Moir 2016), (b) with increased range comes either decreased resolution or increased number of filters (hence increased computational effort), and particularly because (c) it allows usage of a sample rate of 16 kHz, which means considerably faster processing and considerably less memory/disk space needs both in the binaural scene synthesis and in the auditory processing.

But is this enough? The persistence spectra in Fig. 2.1 reveal content beyond 8 kHz. Humans can hear up to about 20 kHz. Gygi et al. (2004) showed that for humans the most informative frequency range for identification of environmental sounds is 1200 Hz to 2400 Hz, and that the information gain through including the range from 2400 Hz to 4800 Hz was small. On the other hand, 70 % of the sounds were still identifiable when only using information above 8 kHz — therefore at least, one could conclude that the high frequencies offer redundant information which would be helpful in polyphonic situations. Whether and how much this extended frequency range would provide information gain about co-occurring sound events in this context, should be tested. Results in Çakir and Virtanen (2018) and Jeong and Lim (2018) indicate an advantage of higher sample rates and thus higher frequency range. Also, the leading models in DCASE Community (2019) were working with at least 32 kHz sample rate.

10.1.4.1 *Binaural features*

The whole modeling presented here was done using binaural data from two microphones in a dummy head's ears. Experiments with different schemes for combining the binaural features were performed: either superposition or concatenation of the representations from both channels was applied (Section 4.1.2).

Over the last few years, the use of multi-channel features has been suggested for improving audio scene classification and SED (Adavanne et al. 2017; Adavanne, Politis, and Virtanen 2018; Adavanne and Virta-

nen 2017; H. Phan et al. 2015; Xu, Kong, Huang, et al. 2017), facilitating recognition of overlapping events.

Similarly, in the performed experiments on single-conditional iso- and cross-performances (Chapter 4) clear advantages were found for models employing (concatenated) two-channel features over models employing mean-channel features — but only, if training was performed with data from scenes with the same or very similar source locations as in testing. For mismatched source locations, the mean-channel features performed stronger. Interestingly, even with the mean-channel multi-conditional models, source distributions could be identified that increase or decrease detection performance. An investigation of these dependencies had not been conducted before.

10.1.5 *Temporal context*

Audio classification usually was based on features extracted from time windows of multiple seconds in order to sufficiently capture non-stationary and temporal characteristics (e. g. Valero and Alias (2012)). It has been shown that time windows can be reduced to one second without significant loss in performance (Khunarsal et al. 2013). For "online" sound event detection in a robotic system, this may still be too long; hence in this work, the time windows for the LASSO models were even further reduced to 0.5 s to build responsive detectors.

In order to make use of longer temporal contextual information and investigate the benefit from it, RNNs and CNNs, which have the fundamental capability to aggregate information over long time, and still be responsive, were trained. They were compared to several variants of above mentioned LASSO models, with different (also multiple) time windows (Chapter 7).

Even when training RNNs, in literature often rather short fixed length audio sequences are employed (e. g. Jeong and Lim (2018)), which then often implies truncation of sequences. In realistic and/or highly disturbed/noisy scenes, this bears potential loss in performance due to not using all available information. Takahashi et al. (2016) showed an improvement of (single-source) SED performance with DNNs for increasing input time context. For their system, performance seemed to saturate between about 1 s to 3 s. However, investigations with respect to the influence of temporal context in polyphonic situations had not been published so far. Huy Phan et al. (2018) have investigated the influence of temporal context length for audio *scene* classification with CNN and RNN, and found considerable scene-dependent performance profits for increased test signal lengths up to 30 s.

The results presented demonstrate significant gain in demanding auditory situations from using temporal context sizes up to the maximum tested, 20 s (the limit may thus be even higher). This was only possible for the DNN models, all tested variants of LASSO models failed to benefit from context longer than one second.

Considering that in this work the concept of *semantic* realism in the analyzed acoustic scenes was completely ignored, these results are assumed to actually underestimate the benefit from longer temporal context information. While the acoustic scenes here were rendered from randomly drawn sound events independent from each other, in reality, sound events of course very often are not independent. It seems likely that it would be easier to identify sound events if they occurred in an acoustic scene in which they usually do (Huang et al. 2018; Niessen et al. 2008). This should be the case even more so, when models have access to longer time context, because longer time context would mean more semantic context.

However, interestingly, experiments with humans in this regard seem to not have provided clear results yet. In Gygi and Shafiro (2011), sounds embedded in *incongruent* scenes were slightly better identified than in congruent scenes. Contrary, Risley et al. (2012) have shown a positive effect of semantic context in the perception of environmental sounds.

10.2 EVALUATING SOUND EVENT DETECTION MODELS

The choice of loss function for the model optimization and performance measure for model selection and generalization evaluation is a substantial factor for the success of machine learning – it *defines* what's success and what not.

The following subsections discuss the avoidance of the most-commonly performance metric in SED, the F-score, and the utilization of the balanced accuracy (BAC) instead.

10.2.1 *Why not F-score*

As elaborated in Section 3.3.2.3, the F-score is dependent on the ratio of negative to positive samples. So, F-scores do not only make a statement about the quality of models' predictions, but they are statements about the quality of models' predictions on data with a particular ratio of negatives to positives: hence F-scores are not comparable for different data distributions.

However – unless very careful pre-selection is performed –, in the binomial one-vs-all multi-label setup employed in this thesis (and with most others for SED), usually data distributions *do* differ, and this not only between different experiments and studies, but already between the different sound classes in the same experiment, because samples of different sound classes rarely occur in balanced amounts[6]. The F-scores of models for different sound types thus are not comparable, even if tested on the same mixtures.

Based on above reasoning, the F-score is reasonably used only when optimizing model performance on a particular data set/distribution is of interest, but less so, if interpretation of algorithm performance or comparison of modeling approaches are goals, because the values have no direct interpretation and are rarely comparable. Similarly, if chances are that proportions of positives and negatives in application of models differ from training, model assessment via F-score is problematic. (Hand (2006) argues that such differences are more likely than not.)

Unfortunately, flawed comparisons based on F-scores obtained from different distributions are common practice. For example, the evaluation of the DCASE polyphonic challenges (Lafay et al. 2017; Dan Stowell et al. 2015) in this regard was conducted methodologically erroneous. On same-length scenes, F-scores on different densities of events are not comparable. Scenes with higher density of sound events by definition have lower r_{NP} (cf. Eq. (3.9)), and models operating on them consequentially (unless the false positive rate is 0) have higher F-score (cf. Eq. (3.8)) — without actually getting better. However, the level of polyphony in the challenge was controlled by event density; scenes got more difficult because the event densities were increased and more polyphonic overlap occurred. So the problem there interpretation-wise was even doubled: models could achieve higher F-scores without getting better — on more difficult scenes. Similarly, in A. Mesaros et al. (2018) F-scores between DCASE 2016 task 2 (synthetic scenes) and task 3 (real audio) were compared — but because the data sets are different, conclusions drawn from such comparisons are questionable.

ERROR RATE In DCASE and elsewhere commonly used additionally to the F-score, it shares exactly the same problems, because it does not take into account the true negatives either. A description can be found in Annamaria Mesaros et al. (2016).

6 Even less so for mixtures of sounds.

10.2.2 *Why balanced accuracy*

Because of above reasons, and since producing robust – particularly with respect to conditions differing from training, including different distributions – classifiers was the goal of this work, choosing BAC (see Section 3.3.2.1) as performance measure seemed more appropriate. BAC is closer as a metric to an interpretation of "informedness" of a classifier. The same line of argumentation can be found in Powers (2011) promoting Bookmaker Informedness (also known as Youden's J Statistic (Youden 1950)), which is a scaled equivalent of BAC.

However, the choice of performance measure in this work, both for training and testing, is legitimate subject for debate as well. To come to a really informed decision, first of all would need a clear definition of the problem with respect to cost of misallocation of classifiers, which was not done here. Instead through usage of balanced accuracy on data with small share of positives, implicitly higher cost was allocated to false negatives than to false positives, which is to a certain extent arbitrary.

So for one thing, it is debatable whether BAC was the optimal choice. Frequently, the Matthews correlation coefficient is suggested as the most informative single score binary classification measure, least impacted by class imbalance (Chicco 2017; Powers 2011), so this might have been an alternative. Alternatively, Hand (2009) and Hand and Anagnostopoulos (2014) propose the "H-measure" (and strongly argue against the commonly used AUC measure) for situations in which misallocation cost is not known a-priori, so maybe this would have been the most appropriate measure.

For another thing, using $BAC2$ (Section 3.3.2.2) for training and then BAC for testing is questionable. $BAC2$ does work to prefer a configuration of equal sensitivity and specificity, but because of this it also rates $SENS = 0.8$, $SPEC = 0.8$ over $SENS = 0.9$, $SPEC = 0.72$, which would feature higher BAC. No systematic investigation upon the effect of using $BAC2$ in training has been done; this should be caught up upon, or BAC should be used in training as well.

10.2.3 *Macro-averaging*

As explained in Section 3.3.3, class-average performance numbers are used throughout this thesis. Annamaria Mesaros et al. (2019) in their report on the DCASE 2017 task 3 illustrate why, and fuel the criticism of common performance evaluation: the F-score was used in the challenge class-independently (micro-averaged), leading to the situation that the top-performing systems were actually unable to recognize a single

instance for three out of six classes. They may have been the top-performing systems, but with that metric, they have not shown that they are able to detect a whole spectrum of sound events. Contributions across the different DCASE events commonly failed to detect several small (with respect to number of occurrences) sound types, which may be attributed among other reasons to the described problems in the definitions of task metrics (and thus optimization targets).

10.2.4 *Event continuity*

One aspect completely disregarded in the performed evaluations is whether predicted sound events were continuous from onset to offset (evaluation metrics called "event-based" in Annamaria Mesaros et al. (2016)). Instead, evaluation was on percentages of correctly detected blocks or frames, irregardless of any mistaken pauses in-between (evaluation metrics called "segment-based" in Annamaria Mesaros et al. (2016)). With sufficiently high detection rates, probably some running average or alike could solve most small discontinuities (which is what many systems do, e. g. Cakir et al. (2015b)), and whether they pose a problem at all, will be task-dependent.

When using a further validation model classifying the whole event subsequent to primary detection, as proposed in Huy Phan et al. (2017) to reduce false positives, producing "continuous" events would be of increased importance.

10.3 ROBUST SPATIAL SOUND EVENT DETECTION

Joint sound event localization and detection (SELD) is a new field (at least with respect to machines performing it [7]), unsurprising, considering that even SED does not have a very long history. Publications tackling it are hard to find, and then often do not describe true SELD systems, because identities and locations are predicted side by side (Butko et al. 2011; Lopatka et al. 2016), not solving the problem of how to associate found locations and identities.

In this thesis, an approach for binding the prediction of the two modalities through spatial segregation has been developed and analyzed (Chapter 8). It could be shown that SED and sound source localization (SSL) can be joined efficiently with this method in a modular system, and that robust performance can be achieved through multi-conditional training. Analyses with respect to different acoustic

7 but, as far as the author of this thesis understands from the literature known to him, it is also not yet fully understood with humans (Bizley and Cohen 2013)

conditions have been presented, particularly regarding the influence of spatial source distribution around the head, which is something that has not been done before in this context. Strong impact of the true source locations on the system's performance has been found. In a binaural robotic system, source locations are subject to the head orientation, hence it was proposed to turn the head such that favorable relative source positions can be achieved. In Chapter 9, successful application of this approach was demonstrated.

The most similar method was published in May et al. (2012), performing speech detection and localization bound through spatial segregation, plus speaker identification. Additionally to the restriction on speech, different training paradigms were used.

Firstly, their system processed whole sentences, rather than the 0.5 s-blocks used for the localized sound event detection here. Using longer segments certainly would be an easy way to increase performance: sound events become more identifiable (cf. Chapter 7) and streams more segregable, because the strength of superposition varies over time and longer blocks include more frames with the individual sources standing out ("glimpsing", also discussed in Section 7.3.1). However, when considering longer temporal contexts for spatially segregated detection, it gets less likely that scenes stay static, i. e., that sources don't move. This then makes segregating into coherent streams and running SED models like the DNNs introduced in Chapter 7 over long time on them a more challenging problem, because some sort of tracking would need to be implemented.

Secondly, the system presented in May et al. (2012) was trained on clean (single-source) data, and spatial masks were applied only during testing, together with a missing data approach (Cooke et al. (2001), one of the most widely used missing/unreliable auditory data approaches). This is a clear contrast to the system proposed here, for which robustness is learned through training multi-conditionally in polyphonic scenes with spatial masks already applied. If one wants, it is a difference of manual versus data-driven design. The results presented in May et al. (2012) show a strong degradation of the final speaker identification for SNR even above 0 dB, while the performances of the herein developed system only degrade very slightly for this SNR range.

Proper estimation of the number of active (spatial) streams is a precondition of the here proposed approach; deviation of more than one leads to very strong performance degradation. This was shown both in the systematic testing environment of Auditory Machine Learning Training and Testing Pipeline (AMLTTP) (Section 8.5) and in the online continuous environment of the Two!Ears simulation framework

(Section 9.2), employing a number-of-sources estimation model with insufficient accuracy[8].

Localization error, on the other hand, does not influence segregated detection as heavily. Even with large errors, the system does not break down, but propagates the input error under mild impairment of assignment precision to the output (Section 8.4).

Adavanne, Politis, Nikunen, et al. (2018) recently presented a fully joint SELD system based on a convolutional recurrent neural network (CRNN) model which produces sound event information over time and, for each sound type modeled, continuous output about the direction of arrival. Their proposed system looks promising in that it combines all information into one powerful (DNN) model that consequentially can make use of auditory cues about sound types and source locations simultaneously rather than sequentially. Furthermore, it is not needing a priori information about the number of active sources. On the other side of the coin, only one source emitting the same sound type can be localized.

As part of the contribution with respect to SELD of this thesis, measures for the quantification of localized sound event detection success have been developed and presented. The solutions obtained include two metrics that similarly have been developed independently in Adavanne, Politis, Nikunen, et al. (2018): what is called here mean azimuth error (azmErr), is called there "DOAerror", and what is called here "number of excess positives" (NEP) and time-wise specificity and detection rate, is summarized in their work through the "frame recall", which is the percentage of frames with correct number of out-put active source locations. The "placement likelihood" presented in this work is unique and adds more fine-grained information about the system's behavior.

As elaborated above already (Section 10.2), the choice of training performance measure has a crucial impact on system performance in any machine learning model. For training the SELD models, a single-number metric is needed. Compared to employing standard BAC, that can not distinguish between negative samples with or without positive in another stream (cf. Section 8.1.6.2) and hence would result in models largely unable to assign events to only the correct stream, the modified BAC_{sw} was proposed. While it has shown to produce functioning models, there may as well be more suitable measures; in particular, BAC_{sw} does not impose cost on azimuth distance to the correct location

8 The development of an efficient model for the estimation of the number of active sources was out of scope in this thesis, but would be another very interesting problem to be solved for an integrated CASA system.

of positive assignments[9]. This would be an interesting point for further research.

Looking at human auditory scene analysis (ASA), the approach of segregating sounds only by means of spatial cues and treating individual time-frequency bins independently (as is proposed here), certainly falls short. As A. S. Bregman (1993) reasons, it is more likely that spatial cues are used together with rules about sound regularities. Hence, it is reasonable to assume that the performance of the proposed SELD system could profit significantly from a more sophisticated segregation model that takes into account spatio-spectro-temporal context. This is what a fully joint model like the one presented in Adavanne, Politis, Nikunen, et al. (2018) does *implicitly*. However, it would be equally interesting to do such context-sensitive explicit segregation in the sequential modular approach proposed here, particularly when also being interested in human-like ASA.

The importance (and impact on performance) of such a segregation model would undoubtedly increase in reverberant conditions. The analysis presented does not include reverberant acoustic scenes; and the investigations with respect to impact of reverberation on fullstream models (Chapter 6) may not translate to the joint detection and localization, since reverberation particularly perturbs spatial cues. This is a shortcoming and should be caught up on, but favoring depth in anechoic analysis over adding reverberant analysis was a compromise necessary to keep the investigation feasible.

As closing words for this section: the field of SELD is due to see increased activity and advances soon — the DCASE 2019 challenge for the first time includes a task about this problem. The work presented here about spatial sound event detection hopefully can also help to start off systematic research on this topic.

9 Other than discriminating between correct or incorrect stream

11

SUMMARY AND CONTRIBUTION

The following sections summarize the work done in the course of this thesis and its contributions to the fields of sound event detection, joint sound event localization and detection, and computational auditory scene analysis.

THE NIGENS DATABASE Systematic polyphonic binaural sound event detection (SED) modeling and testing across many different acoustic scenes is feasible only through simulated (synthesized) scenes. The NIGENS general sounds database (Section 2.1) was tailored to this task. It features 1017 wav-files containing isolated sound events distributed among 14 different sound classes plus a "general" sound class consisting of a very wide variety of sounds not contained in the other event classes. Frame-level event on- and offset annotations make sure that scene synthesizers can generate consistent polyphonic scenes with scene-wide annotations and training algorithms work with precise targets. The database was made available under Creative Commons Attribution Non Commercial 4.0 license.

THE AUDITORY MACHINE LEARNING TRAINING AND TESTING PIPELINE To get from sound files to complex synthesized acoustic scenes to ear-signals, from ear-signals to auditory representations to features and labels for machine learning algorithms, from training data to proper model training by many different algorithms, taking heed of appropriate performance measurement and model selection, from trained models to tested models to evaluated models, a comprehensive pipeline is necessary. The Auditory Machine Learning Training and Testing Pipeline (AMLTTP) (Appendix A) is this pipeline (and more), implemented for consistent development of auditory models of all kinds. Basically everything presented in this thesis was created through the AMLTTP. It has been published as public-domain open-source code.

ANALYSIS OF ACOUSTIC SCENE CONFIGURATION INFLUENCE The acoustic performance factors of polyphonic SED have been thoroughly analyzed. It was shown that sound event detection models trained on one-source scenes degrade strongly in cases when a distractor source is present simultaneously emitting sound events even of much lower energy (10 dB), but by superimposing highly variable general sounds at training time, models can learn to focus on the target class in the presence of distracting sounds even with much stronger energy (−20 dB) (Chapter 4). Compared to existing publications, the range of investigated conditions and severity of disturbances has been considerably increased, with the number of co-occurring sources up to 4 (Chapter 7) and the energy ratios between co-occurring target and distractor sources down to −20 dB.

Particular focus was put on investigation of performance behavior in mismatched conditions, i. e., when models have been trained under different acoustic conditions than they are used in. It was shown that it is not enough to train with polyphonic scenes (but better than training on clean scenes, which is frequently done), because model performances anyway strongly depend on how similar the training conditions were to test conditions. In particular, deviations in mean signal-to-noise ratio (SNR) lead to distinct performance drops.

The provided results provide indication of what to expect in these various conditions, minimum baselines for what is possible to reach, and guidelines of what has to be considered.

BINAURAL SOUND EVENT DETECTION All signals used for training and testing the sound event detection models in this work have been binaural ear-signals from a (simulated) robot dummy head (Section 2.2). Compared to most SED research that is conducted on single-channel data, this adds the (so far completely untreated) dimension of source locations to acoustic scenes; furthermore the question how to build features from the two channels.

Addressing the latter, channel-average and channel-concatenated features have been evaluated (Chapter 4). The mean-channel feature set handled deviations of the training azimuth configuration more tolerantly than the two-channel set, which instead showed significantly better performance if training and testing configurations were close enough. The investigation of the effect of azimuth configuration and head orientation on SED performance was conducted throughout all parts of this work, and it was found that for all models tested, there is an optimal head orientation relative to the sources, at which detection performance is maximized (Sections 5.3, 7.3.3 and 8.3.3).

That these results can lead to the development of robotic systems that actively maximize performance through head rotation, has been demonstrated in Chapter 9.

DEVELOPMENT OF POLYPHONIC MULTI-CONDITIONAL MODELS How to produce robust models in a data-driven way through multi-conditional training – and analyzing their robustness – is the core of this thesis. Robust classifiers should be as independent as possible of acoustic testing conditions, which in real-world applications can change quickly.

It was possible to show that robustness can be achieved using this method with auditory-inspired features (Sections 2.3 and 3.1.1) even with simple linear classifiers (Sections 3.2 and 3.4) far from the learning power of deep neural network (DNNs). Multi-conditional training produces robust sound event detection models able to generalize across a wide range of acoustic conditions (Chapters 5 to 8), analyses comparing multi-conditional with single-conditional iso- and cross-test performances have been conducted (Chapters 5 and 6). The multi-conditional models clearly outperformed single-conditional models tested at a priori unknown configurations. Even better, they reached almost optimal (single-conditional condition-matched) performance for most configurations. Only single-conditional models at matched conditions using two-channel features could outperform the multi-conditional models, which naturally can not specialize to particular source locations.

Furthermore, it was demonstrated that multi-conditional modeling smoothly extends to other model types like long short-term memory (LSTM) or temporal convolutional network (TCN) (Chapter 7). One conclusion of this thesis therefore is that multi-conditionally trained models are a good choice for practical binaural applications (Chapter 9): only a single model needs to be trained for each target sound class, inferring the conditions a priori is not required, the resulting models are robust with respect to conditions and even reach a close-to-optimal performance.

ANALYSIS OF INFLUENCE OF ROOM ACOUSTICS The influence of room acoustics and training-test-mismatches in this aspect on sound event detection has been investigated for the first time.

The presented results (Chapter 6) show that single-conditional SED models specialize to the room and head acoustics they are trained on, and hence are sensitive to deviation at test time from these; but they are less prone to mismatch performance degradation if they were trained under "difficult" room acoustics. It was found that single-conditional models can perform adequately if they were trained on data from the

room they are applied in later, and that they do not need to be trained on data with the exact same position in the room. However, it could be demonstrated that models trained multi-conditionally across room acoustics generalize very well across and also to other room acoustics.

ROBUST PERFORMANCE MEASUREMENT In sound event detection, the most widely used performance measure, the F-score, brings with it a lot of problems outside of training for a specific, a priori known, data distribution. The most widely used way of dealing with class-specific performances measured, micro-averaging, brings with it misinterpretation with respect to general sound event detection potential of algorithms. Hence, in this work, the aspect of correct and robust performance measurement (Sections 3.2.3 and 3.3) and its importance is illuminated and contrasted to common practice (Section 10.2).

INFLUENCE OF TEMPORAL CONTEXT Sound events in realistic acoustic scenes often occur over longer periods of time and not only once. Therefore, an analysis on how sound event detection performance increases with the size of temporal context accessible to the models has been presented in this work (Chapter 7). It was shown that performance increases particularly for complex acoustic scenes and demanding auditory conditions and that this is very sound class-specific.

The extent to which context can be accessed by simple linear models through statistically summarized temporal features (Section 3.1.1.1) was compared with the extent to which models able to learn temporal integration themselves (here, DNNs, Section 3.5) can profit from. Evaluation showed that the difference is quite considerable: while the architectures able to exploit long-range temporal context information, LSTM and TCN, were able to exploit up to (the maximally tested) 20 s context and continuously increased performance, the simple models peaked at 1 s at lower performance. Hence, training should be conducted over sufficiently long scenes.

While both LSTM and its convolution-based competitor TCN have shown their sequence modeling power, LSTM produced the maximum performance achieved.

Related to questions of temporal context is the definition of training targets (and for evaluation, testing targets). Two different modes of defining training targets have been investigated here, one more "physically" motivated and supposedly more exact/instantaneous (without any notion of temporal context, Section 3.1.2.2), and one more perceptually motivated with a "smoothed" (temporally integrated, and hence a bit lagged) interpretation (Section 3.1.2.1). While the first showed to be the more difficult problem, qualitatively with respect to behavior

over acoustic conditions, they were not different (Chapter 7). What one choses hence can be decided based on the needs of the application.

SPATIALLY SEGREGATED SOUND EVENT DETECTION Aside "pure" sound event detection, a substantial part of this thesis was devoted to *spatially segregated* sound event detection (Chapter 8), which was suggested and evaluated as a method to annotate sound scenes with joint sound event type and location information in a binaural system. The proposed method combines spatial masking in time-frequency-space with sound event detection on the segregated streams, enabling formation of coherent auditory objects with location and sound event type associated.

Along with the proposed system, the general problem of joint sound event localization and detection (SELD), which only very recently started to get treated, has been introduced and discussed. Performance measures for quantification and qualification of localized sound event detection success were developed and presented (Section 8.1.6).

Evaluation showed that sound event detection and sound source localization can be joined efficiently through spatial masking in a modular system. Analyses with respect to the different acoustic conditions showed robust performance under a broad range of conditions. The influence of spatial source distribution has been particularly investigated due to its increased importance with respect to spatial segregation based on inter-aural signal differences.

The presented analysis demonstrates that this approach can produce localized sound type information and could be one core component of a binaural scene analysis system. The localized detection performance depends particularly on the number of active sources in the scene, and on their spatial distribution. By turning the head such that the sources of interest are laterally separated (and at best bisected by the nose), the system's performance in many situations can be increased strongly. It was found that proper estimation of the number of active spatial streams is a precondition of this approach, while localization error does not influence segregated detection as heavily.

A diverse set of test scenes for thorough study of the behavior and conditions of good performance was defined. It can serve, together with the proposed performance measures, as testbed and benchmarks for alike systems with different components and other approaches to the problem, and of course particularly for spatial segregation and sound event detection models.

CASA WITH THE DEVELOPED MODELS IN THE TWO!EARS SYSTEM Finally, binaural computational auditory scene analysis was conducted

with the developed components employed in the Two!Ears robotic simulation platform (Chapter 9). It was demonstrated that the fullstream and segregated detection models efficiently detect sound events in a dynamic online binaural system with actual sound source localization instead of location ground truth. Results asserted that performances were well-predicted in the evaluation of tests performed in the (static) AMLTTP in Chapter 8.

A new head rotation strategy was proposed that maximizes lateral source separation (Section 9.1.3.1) and thereby optimizes spatial segregability. Indeed, utilizing this strategy in the dynamic Two!Ears system resulted in significantly increased localized detection performance.

Concluding, it was showed that the models developed in this thesis indeed robustly detect sound events both on full- and spatially segregated streams and constitute efficient components of a binaural system performing computational auditory scene analysis.

Part V

APPENDIX

A

THE AUDITORY MACHINE LEARNING TRAINING AND TESTING PIPELINE

This chapter is based on Ning Ma et al. 2016. *Deliverable 3.5: Report on Evaluation of the Two!Ears Expert System.* Final report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D3.5_evaluation_of_expert_system.pdf.
Parts of the following text have been published in that project report already; but also completely written by me.

All scene generation, data processing, model training and model testing was done using the open-source Auditory Machine Learning Training and Testing Pipeline (AMLTTP) (Trowitzsch, Kashef, et al. 2019), which wraps all steps described in Chapters 2 to 8. A considerable amount of time has been spent on developing, testing and improving this tool — without it, it would not have been possible to conduct the studies described in this thesis.

AMLTTP is particularly suited for sound event detection (SED) model training; among other features, it enables straight-forward generation of complex spatial polyphonic sound scenes together with polyphonic annotations, from databases with isolated sounds like NIGENS.

However, it is not limited to SED model training, rather, AMLTTP is an object-oriented (Matlab) framework for building and evaluating all kinds of models for auditory sound object annotation and assigning attributes to them. It consists of two major parts: (i) a data generation engine, and (ii) a model training and testing back-end; both parts can be broken down into further sub-stages and components. While the pipeline is designed with flexibility in mind and is extendable to new target attributes, data features, or model and training algorithms, it so far serves the specific purpose of training and evaluation of block-based auditory object-*type*, object-*location*, and *number-of-sources* classifiers using data from simulated auditory scenes generated within the same framework. It is tightly coupled with the Two!Ears system and its components.

A.1 DATA GENERATION

First and very important, AMLTTP contains a large-scale data generator including acoustic scene synthesis. "Large-scale", because AMLTTP not only generates data, it also efficiently *manages* huge amounts of data at all stages, keeping track of everything already created, enabling resume and reuse at all levels[1].

AUDITORY SCENE SIMULATION What has been described in Section 2.2, was executed by the AMLTTP scene synthesizer. Ear-signals are produced from audio files using the binaural simulator (Winter, Wierstorf, and Trowitzsch 2016, Ch. 2.2) from the TWO!EARS system. This can be done under various conditions, set up through a configuration object specifying the following (and more):

- An arbitrary number of sources point-sources with configurable positions relative to the head can be created.
- The "head", i. e. the head-related impulse response (HRIR) used, can be exchanged. By default, it is defined as a KEMAR head.
- Sources can be set up to emit specific audio files or white noise.
- Sources can be set up to playback one audio file and then mute, or loop over this audio file, or playback audio files from a set in random order, for a defined duration.
- Ear-signal-level average SNRs between sources can be fixed. They are defined as the ratio of powers between the individual sources' ear-signals.
- Simulated reverberation (shoe-box room model) can be defined, or
- binaural room impulse responses (BRIRs) can be used instead of HRIRs. Multi-speaker BRIRs are supported to allow for setting up multiple sources.

The scene synthesizer is very automatable and thus enables efficient generation of large amounts of scenes.

1 In the course of this thesis, AMLTTP temporarily managed up to 40 TB data consisting of several hundreds of different scenes with a variety of auditory representations and plenty of different feature and label sets.

COMPUTATION OF AUDITORY REPRESENTATIONS After ear-signals have been generated, stage two basically wraps the Two!Ears Auditory Front-end (AFE) (May, Decorsière, et al. 2015) for automated and efficient computation of base auditory representations (Section 2.3) like ratemaps, amplitude modulation spectrogram (AMS) or interaural time-differences (ITDs) — the whole range of representations the AFE can compute is usable.

FEATURES CONSTRUCTION Stage three performs feature construction (Section 3.1.1) from the auditory representations, executed by so-called `FeatureCreators`.

`FeatureCreators` inherit from a common base interface and are modules to be implemented by the user of the AMLTTP[2]. They construct data vectors in a form that is suitable for the model to be trained — for example, as described in Sections 3.1.1.1 and 3.1.1.2 for time-invariant segment-based features or frame-based features. To be able to later evaluate models on a feature-level, e. g. for dimensionality reduction, `FeatureCreators` automatically produce a detailed description of each feature dimension. Masks can be used to incorporate results of feature selection methods and only train or test with selected dimensions.

LABEL CREATION Similar to the generation of feature vectors described in the section above, the corresponding target vectors are produced by `LabelCreators` to produce labels as for example described in Section 3.1.2. These are later used in the supervised training of models, and serve as ground truth in the testing of models. Analogous to `FeatureCreators`, `LabelCreators` implement a common interface for quickly creating new labelers describing different object attributes. Any of these object attributes are derived from scene-wide annotations at different levels of abstraction (e. g. active sources, source energies, etc.) which were produced in the scene generation stage and are passed through the pipeline. `LabelCreators` for producing *sound type*, *source location* and *number of sources* targets are already available and can be used 'right out of the box', including combinations thereof. Binomial (Section 3.2.1), multinomial, or regression targets can be produced, and combined to form multivariate labels.

COUPLING WITH THE TWO!EARS SYSTEM The AMLTTP and the Two!Ears blackboard system are tightly coupled in two central aspects: AMLTTP-trained models can be used from within the blackboard system, and blackboard system knowledge sources can be used from within the AMLTTP.

2 Hence any feature set imaginable can be implemented.

Existing Two!Ears system knowledge sources and the models that come packaged with them can get included as "data processors" into the AMLTTP through a wrapper interface. This enables incorporation of other models' hypotheses about the ear-signals into the data generation process. Models that use other models' outputs can be built through this feature. As an example, the spatial-segregation module of the Two!Ears system has been included through this mechanism into AMLTTP and this way enabled training on spatially segregated auditory data (Section 8.1.3).

UTILITY A lot of work has gone into making the AMLTTP both easy and effective to use. While the first is accomplished through a clean high-level interface, the latter is enabled to a large extent through the following two points:

1. All products from intermediate stages, such as the generated ear-signals, or features produced by the AFE, are saved together with their corresponding configurations inside a caching system. Since these stages can be very time-costly for large data sets: repeating trainings with a different model, resuming a process after a crash, etc., are made possible without having to recompute everything again. Also, data are saved in such a way that they will be re-combined automatically whenever parts of a configuration have already been computed before. This saves a lot of computation time and helps assigning the produced data to many different experiments' configurations and reproduce experiments more easily.

2. The AMLTTP can be run concurrently on many processes and/or machines, whilst working from the same data. This is secured through file semaphores so that processes don't interfere with one another when operating on the same configuration in the same stage.

STANDALONE USAGE The data generation facilities of the AMLTTP can be used with or without subsequent model training and testing. All data generated can be easily exported to mat-files for usage in other environments, as was for example proceeded for the training of deep neural network (DNNs) (Chapter 7) with Python and Tensorflow.

A.2 MODEL TRAINING AND TESTING

The other core part of AMLTTP is the model training and testing back-end, working seamlessly together with the data generation pipeline.

PROPER MODEL BUILDING BACK-END A lot of machine learning best practices and utilities for proper training and testing are integrated into AMLTTP. Among them are:

- Easy system for data set splitting into training, validation and test sets from re-combinable "folds". Wrapper `modelTrainers` (see below) provide implemented (parallel executing) cross-validation (Section 3.3.1) and hyperparameter selection for any kind of model algorithm.
- In-built feature standardization (Section 3.1.1.3).
- Flexible and user-extendable sample weighting system enabling sample-specific cost/loss, as described in Sections 3.2.3 and 8.1.4 for the herein developed models.
- Several implemented performance measures (Section 3.3.2) for model selection and model testing, including balanced accuracy (BAC) and F-score. A common base interface enables users to implement the measure of their choice.
- Powerful model evaluation support through associating individual model predictions for short segments with the respective scene annotations.

MODEL TRAINING ALGORITHMS Consistent with the concept of modularity and extendability, common interfaces exist for models and their corresponding `modelTrainers`. Any algorithm for model construction can be used, and any model type can be constructed and tested. A class inheriting from these interfaces can implement its own technique and can be plugged into AMLTTP. Currently implemented are:

- L_1-regularized, L_2-regularized, or elastic net logistic (Section 3.4), Gaussian, or Poisson regression,
- random forest,
- and support vector machines (SVMs).

While DNNs are state-of-the-art for much model building nowadays, (i) Matlab unfortunately is not state-of-the-art for DNNs, (ii) of course it is anyway completely possible and easy to implement DNN `modelTrainers` with Matlab, and (iii) this thesis has shown that "conventional" models still can achieve very good performances.

A.3 USE-CASES

There have been many use-cases for the AMLTTP in this thesis:

- Building of single-conditional sound event detection models, and testing them in matched and mis-matched conditions (Chapters 4 and 6).
- Training of multi-conditional sound event detection models, and testing them on wide ranges of conditions on robustness (Chapters 5 and 6).
- Generation of multi-conditional data for training and testing of DNN sound event detection models on long acoustic scenes (Chapter 7).
- Development of a joint sound event localization and detection (SELD) system by training multi-conditional sound event detection models on spatially segregated data (Chapter 8).
- Development of models employable as components in the Two!Ears binaural system for computational auditory scene analysis (CASA) (Chapter 9). Models created by the AMLTTP can be plugged into the blackboard system and be used there without any modification or interface adjustments. They automatically employ the right `FeatureCreator` and feed the system with their hypotheses about auditory object attributes. Regardless of whether the ear-signals that are fed into the blackboard system are produced using a binaural simulator or acquired from actual microphones, features are produced in exactly the same manner as in the AMLTTP.

Presented in Two!Ears project reports, there have also been:

- Training of a model estimating the number of active sound sources (Ma et al. 2016, Ch. 3.5.2).
- Generation of data for a fully joint SELD system based on DNNs (Ma et al. 2016, Ch. 3.4.4).
- Analyses of feature importance for sound event detection models based on L_1-regularized logistic regression models (Ma et al. 2015, Ch. 3).

And of course, there would be countless other projects that one could conduct through the means of the AMLTTP.

BIBLIOGRAPHY

Adavanne, Sharath, Pasi Pertila, and Tuomas Virtanen. 2017. "Sound event detection using spatial features and convolutional recurrent neural network." In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Adavanne, Sharath, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. 2018. "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks." *IEEE Journal of Selected Topics in Signal Processing*.

Adavanne, Sharath, Archontis Politis, and Tuomas Virtanen. 2018. "Multichannel sound event detection using 3D convolutional neural networks for learning inter-channel features." In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Adavanne, Sharath, and Tuomas Virtanen. 2017. "A report on sound event detection with different binaural features." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. http://arxiv.org/abs/1710.02997.

Alıías, Francesc, Joan Socoró, and Xavier Sevillano. 2016. "A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds." *Applied Sciences* 6 (5).

Aytar, Yusuf, Carl Vondrick, and Antonio Torralba. 2016. "Soundnet: Learning sound representations from unlabeled video." In *Advances in neural information processing systems*, 892–900.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. "Layer Normalization." *arXiv preprint arXiv:1607.06450*.

Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. 2018. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling." *arXiv preprint arXiv:1803.01271*.

Bardeli, Rolf, Daniel Wolff, Frank Kurth, Martina Koch, K-H Tauchert, and K-H Frommolt. 2010. "Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring." *Pattern Recognition Letters* 31 (12): 1524–1534.

Barker, J.P., M.P. Cooke, and D.P.W. Ellis. 2005. "Decoding speech in the presence of other sources." *Speech Communication* 45 (1): 5–25.

Benetos, Emmanouil, Grégoire Lafay, Mathieu Lagrange, and Mark D Plumbley. 2016. "Detection of overlapping acoustic events using a temporally-constrained probabilistic model." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6450–6454.

Bergstra, James, and Yoshua Bengio. 2012. "Random search for hyper-parameter optimization." *Journal of Machine Learning Research* 13 (02): 281–305.

Bizley, Jennifer K, and Yale E Cohen. 2013. "The what, where and how of auditory-object perception." *Nature Reviews Neuroscience* 14 (10): 693.

Blauert, Jens, Sylvain Argentieri, Guy Brown, Gabriel Bustamante, Benjamin Cohen-L'Hyver, Patrick Danès, Bruno Gas, et al. 2014. *Deliverable 4.1, Part B: Consolidated Literature Survey.* Report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D4.1_feedback-loop_selection_and_listing.pdf.

Bregman, A.S. 2008. "Auditory Scene Analysis." In *The Senses: A Comprehensive Reference*, edited by Richard H. Masland, Thomas D. Albright, Thomas D. Albright, Richard H. Masland, Peter Dallos, Donata Oertel, Stuart Firestein, et al., 861–870. Academic Press.

Bregman, Albert S. 1993. "Auditory scene analysis: Listening in complex environments." In *Thinking in sound: The cognitive psychology of human audition*, edited by Stephen Ed McAdams and Emmanuel Ed Bigand, 10–36. Clarendon Press/Oxford University Press.

Bregman, Albert S. 1994. *Auditory scene analysis: The perceptual organization of sound.* MIT press.

Brown, Guy J, and Martin Cooke. 1994. "Computational auditory scene analysis." *Computer Speech & Language* 8 (4): 297–336.

Bustamante, Gabriel, Patrick Danés, Thomas Forgue, and Ariel Podlubne. 2016. "Towards information-based feedback control for binaural active localization." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6325–6329. IEEE.

Butko, Taras, Fran González Pla, Carlos Segura, Climent Nadeu, and Javier Hernando. 2011. "Two-source acoustic event detection and localization: Online implementation in a smart-room." In *2011 19th European signal processing conference (EUSIPCO)*, 1317–1321. IEEE.

Cakir, Emre, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. 2015a. "Multi-label vs. combined single-label sound event detection with deep neural networks." In *2015 23rd European signal processing conference (EUSIPCO)*, 2551–2555. IEEE.

Cakir, Emre, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. 2015b. "Polyphonic sound event detection using multi label deep neural networks." In *2015 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Cakir, Emre, and Tuomas Virtanen. 2017. "Convolutional Recurrent Neural Networks for Rare Sound Event Detection." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 1–5. 11.

Çakir, Emre, and Tuomas Virtanen. 2018. "End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input." In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Chakraborty, Rupayan, and Climent Nadeu. 2014. "Sound-model-based acoustic source localization using distributed microphone arrays." In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 619–623. IEEE.

Cherry, Colin. 1957. "On human communication; a review, a survey, and a criticism."

Chetlur, Sharan, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. 2014. "cuDNN: Efficient Primitives for Deep Learning." *arXiv:1410.0759 [cs]*.

Chicco, Davide. 2017. "Ten quick tips for machine learning in computational biology." *BioData mining* 10 (1): 35.

Choi, Inkyu, Kisoo Kwon, Soo Hyun Bae, and Nam Soo Kim. 2016. "DNN-based sound event detection with exemplar-based approach for noise reduction." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 16–19.

Chollet, François, et al. 2015. *Keras*. https://keras.io.

Cooke, Martin. 2006. "A glimpsing model of speech perception in noise." *The Journal of the Acoustical Society of America* 119 (3): 1562–1573.

Cooke, Martin, Jon Barker, Stuart Cunningham, and Xu Shao. 2006. "An audio-visual corpus for speech perception and automatic speech recognition." *The Journal of the Acoustical Society of America* 120 (5): 2421–2424.

Cooke, Martin, Phil Green, Ljubomir Josifovski, and Ascension Vizinho. 2001. "Robust automatic speech recognition with missing and unreliable acoustic data." *Speech communication* 34 (3): 267–285.

Dai, Wei, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. 2017. "Very deep convolutional neural networks for raw waveforms." In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 421–425. IEEE.

David, H. A., and H. N. Nagaraja. 2003. *Order Statistics.* 3rd ed. Wiley.

DCASE Community. 2019. "DCASE 2018 task2 – General-purpose audio tagging of Freesound content with AudioSet labels." April 13. http://dcase.community/challenge2018/task-general-purpose-audio-tagging-results.

Dennis, Jonathan, Huy Dat Tran, and Eng Siong Chng. 2012. "Image feature representation of the subband power distribution for robust sound event classification." *IEEE Transactions on Audio, Speech, and Language Processing* 21 (2): 367–377.

Dennis, Jonathan, Huy Dat Tran, and Haizhou Li. 2010. "Spectrogram image feature for sound event classification in mismatched conditions." *IEEE signal processing letters* 18 (2): 130–133.

Dikow, Jan. 2018. "Analyse des Einflusses von Räumlichkeit auf die Robustheit maschineller Geräuscherkennung in binauralen Hörszenen." Master thesis, Technische Universität Berlin.

Dobson, Annette J. 2002. *An introduction to generalized linear models.* 2nd ed. Chapman & Hall/CRC Boca Raton.

Downey, Allen B. 2016. *Think DSP: digital signal processing in Python.* O'Reilly Media, Inc.

Dufaux, A., L. Besacier, M. Ansorge, and F. Pellandini. 2000. "Automatic sound detection and recognition for noisy environment." In *2010 18th European Signal Processing Conference (EUSIPCO)*, 1–4.

Ellis, Daniel PW. 1996. "Prediction-driven computational auditory scene analysis." Phd, Columbia University.

Erbes, Vera, Matthias Geier, Stefan Weinzierl, and Sascha Spors. 2015. "Database of single-channel and binaural room impulse responses of a 64-channel loudspeaker array." In *Audio Engineering Society Convention 138.*

Espi, Miquel, Masakiyo Fujimoto, Keisuke Kinoshita, and Tomohiro Nakatani. 2015. "Exploiting spectro-temporal locality in deep learning based acoustic event detection." *Eurasip Journal on Audio, Speech, and Music Processing,* no. 1.

Ewert, Stephan D, and Torsten Dau. 2000. "Characterizing frequency selectivity for envelope fluctuations." *The Journal of the Acoustical Society of America* 108 (3): 1181–1196.

Fonseca, Eduardo, Jordi Pons Puig, Xavier Favory, Frederic Font Corbera, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. 2017. "Freesound datasets: a platform for the creation of open audio datasets." In *Proceedings of the 18th ISMIR Conference,* edited by X Hu, SJ Cunningham, D Turnbull, and Z Duan. International Society for Music Information Retrieval (ISMIR).

Font, Frederic, Gerard Roma, and Xavier Serra. 2013. "Freesound technical demo." In *Proceedings of the 21st ACM international conference on Multimedia,* 411–412. ACM.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. "Regularization paths for generalized linear models via coordinate descent." *Journal of statistical software* 33 (1): 1.

Gal, Yarin, and Zoubin Ghahramani. 2016. "A theoretically grounded application of dropout in recurrent neural networks." In *Advances in neural information processing systems,* 1019–1027.

Gannot, Sharon, Emmanuel Vincent, Shmulik Markovich-Golan, and Alexey Ozerov. 2017. "A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (4): 692–730.

Garofolo, J. S., L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. 1993. "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1." *NASA STI/Recon Technical Report N* 93.

Gaver, William W. 1993a. "How do we hear in the world? Explorations in ecological acoustics." *Ecological psychology* 5 (4): 285–313.

Gaver, William W. 1993b. "What in the world do we hear?: An ecological approach to auditory event perception." *Ecological psychology* 5 (1): 1–29.

Geiger, J. T., B. Schuller, and G. Rigoll. 2013. "Large-scale audio feature extraction and SVM for acoustic scene classification." In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–4.

Gemmeke, Jort F, Lode Vuegen, Peter Karsmakers, Bart Vanrumste, et al. 2013. "An exemplar-based NMF approach to audio event detection." In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–4. IEEE.

Giannoulis, P., G. Potamianos, A. Katsamanis, and P. Maragos. 2014. "Multi-microphone fusion for detection of speech and acoustic events in smart spaces." In *2014 22nd European Signal Processing Conference (EUSIPCO)*, 2375–2379.

Girard, Didier A. 1998. "Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression." *The Annals of Statistics* 26 (1): 315–334.

Glasberg, Brian R, and Brian CJ Moore. 1990. "Derivation of auditory filter shapes from notched-noise data." *Hearing research* 47 (1-2): 103–138.

Glorot, Xavier, and Yoshua Bengio. 2010. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Godsmark, Darryl, and Guy J Brown. 1999. "A blackboard architecture for computational auditory scene analysis." *Speech communication* 27 (3-4): 351–366.

Greff, Klaus, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2017. "LSTM: A search space odyssey." *IEEE transactions on neural networks and learning systems* 28 (10): 2222–2232.

Grobler, CJ, CP Kruger, BJ Silva, and GP Hancke. 2017. "Sound based localization and identification in industrial environments." In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, 6119–6124. IEEE.

Guillaume, Anne, C Drake, Claude Blancard, V Chastres, and L Pellieux. 2004. "How long does it take to identify everyday sounds." In *ICAD 2004-Tenth Meeting of the International Conference on Auditory Display*. Georgia Institute of Technology.

Gygi, Brian, and Deborah Ann Hall. 2016. "Background sounds and hearing-aid users: A scoping review." *International journal of audiology* 55 (1): 1–10.

Gygi, Brian, Gary R Kidd, and Charles S Watson. 2004. "Spectral-temporal factors in the identification of environmental sounds." *The Journal of the Acoustical Society of America* 115 (3): 1252–1265.

Gygi, Brian, and Valeriy Shafiro. 2007. "Environmental sound research as it stands today." In *Proceedings of Meetings on Acoustics 153ASA*, 1:050002. 1. ASA.

Gygi, Brian, and Valeriy Shafiro. 2011. "The incongruency advantage for environmental sounds presented in natural auditory scenes." *Journal of Experimental Psychology: Human Perception and Performance* 37 (2): 551.

Hand, David J. 2006. "Classifier Technology and the Illusion of Progress." *Statistical Science* 21 (1): 1–14.

Hand, David J. 2009. "Measuring classifier performance: a coherent alternative to the area under the ROC curve." *Machine learning* 77 (1): 103–123.

Hand, David J, and Christoforos Anagnostopoulos. 2014. "A better Beta for the H measure of classification performance." *Pattern Recognition Letters* 40:41–46.

Harishkumar, N., and R. Rajavel. 2014. "Monaural speech separation system based on optimum soft mask." In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 1–4.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. Springer.

Hayashi, Tomoki, Shinji Watanabe, Tomoki Toda, Takaaki Hori, Jonathan Le Roux, and Kazuya Takeda. 2016. "Bidirectional LSTM-HMM hybrid system for polyphonic sound event detection." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 35–39.

Hayashi, Tomoki, Shinji Watanabe, Tomoki Toda, Takaaki Hori, Jonathan Le Roux, and Kazuya Takeda. 2017. "Duration-controlled LSTM for polyphonic sound event detection." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (11): 2059–2070.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, Weipeng, Petr Motlicek, and Jean-Marc Odobez. 2018. "Joint Localization and Classification of Multiple Sound Sources Using a Multi-task Neural Network." In *Proc. Interspeech 2018*, 312–316.

Heittola, Toni, Annamaria Mesaros, Tuomas Virtanen, and Antti Eronen. 2011. "Sound event detection in multisource environments using source separation." In *CHiME-2011 Workshop on Machine Listening in Multisource Environments.*

Hershey, Shawn, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, et al. 2017. "CNN Architectures for Large-Scale Audio Classification." In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Hertel, Lars, Huy Phan, and Alfred Mertins. 2016. "Comparing time and frequency domain for audio event recognition using deep learning." In *2016 International Joint Conference on Neural Networks (IJCNN)*, 3407–3411. IEEE.

Hirvonen, Toni. 2015. "Classification of Spatial Audio Location and Content Using Convolutional Neural Networks." In *Audio Engineering Society Convention 138.*

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long short-term memory." *Neural computation* 9 (8): 1735–1780.

Hosking, Jonathan RM. 1990. "L-moments: analysis and estimation of distributions using linear combinations of order statistics." *Journal of the Royal Statistical Society. Series B (Methodological):* 105–124.

Hsiao, Roger, Jeff Ma, William Hartmann, Martin Karafiát, František Grézl, Lukáš Burget, Igor Szöke, Jan Honza Černockỳ, Shinji Watanabe, Zhuo Chen, et al. 2015. "Robust speech recognition in unknown reverberant and noisy conditions." In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 533–538.

Huang, G., T. Heittola, and T. Virtanen. 2018. "Using Sequential Information in Polyphonic Sound Event Detection." In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 291–295.

Huzaifah, Muhammad. 2017. "Comparison of time-frequency representations for environmental sound classification using convolutional neural networks." *arXiv preprint arXiv:1706.07156*.

"IEEE DCASE 2016 Challenge." 2016. http://www.cs.tut.fi/sgn/arg/dcase2016.

Inoue, Tadanobu, Phongtharin Vinayavekhin, Shiqiang Wang, David Wood, Nancy Greco, and Ryuki Tachibana. 2018. *Domestic activities classification based on cnn using shuffling and mixing data augmentation.* Technical report. DCASE 2018 Challenge.

Ioffe, Sergey, and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *International Conference on Machine Learning*, 448–456.

Jensen, Kristoffer, and Tue Haste Andersen. 2003. "Real-time beat estimationusing feature extraction." In *International Symposium on Computer Music Modeling and Retrieval*, 13–22. Springer.

Jeong, Il-Young, Subin Lee, Yoonchang Han, and Kyogu Lee. 2017. "Audio event detection using multiple-input convolutional neural network." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*.

Jeong, Il-Young, and Hyungui Lim. 2018. "Audio tagging system using densely connected convolutional networks." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 197–201.

Jeub, Marco, Magnus Schafer, and Peter Vary. 2009. "A binaural room impulse response database for the evaluation of dereverberation algorithms." In *2009 IEEE 16th International Conference on Digital Signal Processing*, 1–5.

Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. 2015. "An empirical exploration of recurrent network architectures." In *International Conference on Machine Learning*, 2342–2350.

Kalchbrenner, Nal, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. "Neural Machine Translation in Linear Time." *arXiv preprint arXiv:1610.10099*.

Khunarsal, Peerapol, Chidchanok Lursinsap, and Thanapant Raicharoen. 2013. "Very short time environmental sound classification based on spectrogram pattern matching." *Information Sciences* 243:57–74.

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv preprint arXiv:1412.6980.*

Klapuri, Anssi. 1999. "Sound onset detection by applying psychoacoustic knowledge." In *1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 6:3089–3092. IEEE.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. 2017. "A study on data augmentation of reverberant speech for robust speech recognition." In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 5220–5224.

Kolossa, Dorothea, and Reinhold Orglmeister. 2004. "Nonlinear postprocessing for blind speech separation." In *International Conference on Independent Component Analysis and Signal Separation,* 832–839. Springer.

Komatsu, Tatsuya, Takahiro Toizumi, Reishi Kondo, and Yuzo Senda. 2016. "Acoustic event detection method using semi-supervised non-negative matrix factorization with a mixture of local dictionaries." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016),* 45–49.

Kong, Qiuqiang, Yong Xu, Iwona Sobieraj, Wenwu Wang, and Mark D. Plumbley. 2019. "Sound Event Detection and Time–Frequency Segmentation from Weakly Labelled Data." *IEEE/ACM Transactions on Audio, Speech and Language Processing* 27 (4): 777–787.

Kryter, Karl D. 2013. *The effects of noise on man.* Elsevier.

Lafay, Grégoire, Emmanouil Benetos, and Mathieu Lagrange. 2017. "Sound event detection in synthetic audio: analysis of the DCASE 2016 task results." In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA),* 11–15.

Lasseck, Mario. 2018. *Acoustic Bird Detection With Deep Convolutional Neural Networks.* Technical report. DCASE2018 Challenge.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep learning." *nature* 521 (7553).

Lerch, Alexander. 2012. *An introduction to audio content analysis: Applications in signal processing and music informatics.* John Wiley & Sons.

Li, Bo, Tara N. Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, et al. 2017. "Acoustic modeling for Google home." In *Proc. Interspeech 2017*, 399–403.

Lim, Hyungui, and Jeongsoo Park. 2017. "Rare sound event detection using 1D convolutional recurrent neural networks." In *Detection and Classification of Acoustic Scenes and Events 2017*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. "Fully convolutional networks for semantic segmentation." In *CVPR*.

Lopatka, K., J. Kotus, and A. Czyzewski. 2016. "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations." *Multimedia Tools and Applications* 75 (17): 10407–10439.

López-Pacheco, Marıía Guadalupe, Luis Pastor Sánchez-Fernández, Herón Molina-Lozano, and Luis Alejandro Sánchez-Pérez. 2016. "Predominant environmental noise classification over sound mixing based on source-specific dictionary." *Applied Acoustics* 112:171–180.

Luo, Huan, Yadong Wang, David Poeppel, and Jonathan Z. Simon. 2006. "Concurrent Encoding of Frequency and Amplitude Modulation in Human Auditory Cortex: MEG Evidence." *Journal of Neurophysiology* 96 (5): 2712–2723.

Lyon, Richard F. 2010. "Machine hearing: An emerging field [exploratory dsp]." *IEEE signal processing magazine* 27 (5): 131–139.

Lyon, Richard F. 2017. *Human and machine hearing*. Cambridge University Press.

Ma, Ning, Jose A. Gonzalez, and Guy J. Brown. 2018. "Robust Binaural Localization of a Target Sound Source by Combining Spectral Source Models and Deep Neural Networks." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26 (11): 2122–2131.

Ma, Ning, Tobias May, and Guy J. Brown. 2017. "Exploiting Deep Neural Networks and Head Movements for Robust Binaural Localization of Multiple Sources in Reverberant Environments." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (12): 2444–2453.

Ma, Ning, Ivo Trowitzsch, Youssef Kashef, Johannes Mohr, Klaus Obermayer, Christopher Schymura, Dorothea Kolossa, et al. 2016. *Deliverable 3.5: Report on Evaluation of the Two!Ears Expert System*. Final report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D3.5_evaluation_of_expert_system.pdf.

Ma, Ning, Ivo Trowitzsch, Johannes Mohr, Christopher Schymura, Dorothea Kolossa, Thomas Walther, Hagen Wierstorf, Tobias May, Guy Brown, and Patrick Danès. 2015. *Deliverable 3.4: Progress report on feature selection and semantic labelling.* Proress report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D3.4_progress_report_on_feature_selection_and_semantic_labelling.pdf.

Ma, Ning, Ivo Trowitzsch, Jalil Taghia, Christopher Schymura, Dorothea Kolossa, Thomas Walther, Hagen Wierstorf, Tobias May, and Guy Brown. 2014. *Deliverable 3.2: Software Architecture.* Progress report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D3.2_progress_report_on_software_architecture.pdf.

Marchi, Erik, Dario Tonelli, Xinzhou Xu, Fabien Ringeval, Jun Deng, and Björn Schuller. 2016. "The Up System for The 2016 DCASE Challenge Using Deep Recurrent Neural Network and Multiscale Kernel Subspace Learning." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016).*

Martin-Morato, I., M. Cobos, and F. J. Ferri. 2018. "On the Robustness of Deep Features for Audio Event Classification in Adverse Environments." In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, 562–566.

May, T., and T. Dau. 2013. "Environment-aware ideal binary mask estimation using monaural cues." In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–4.

May, Tobias, and Torsten Dau. 2014. "Computational speech segregation based on an auditory-inspired modulation analysis." *The Journal of the Acoustical Society of America* 136 (6): 3350–3359.

May, Tobias, Remi Decorsière, Chungeun Kim, and Armin Kohlrausch. 2015. *Deliverable 2.3: The Auditory Front-End Framework.* User manual. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D2.3_extension_of_the_binaural_model_and_integration_of_monaural_and_binaural_models_in_software_package.pdf.

May, Tobias, Ning Ma, and Guy J Brown. 2015. "Robust localisation of multiple speakers exploiting head movements and multi-conditional training of binaural cues." In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2679–2683. IEEE.

May, Tobias, Steven van de Par, and Armin Kohlrausch. 2012. "A binaural scene analyzer for joint localization and recognition of speakers in the presence of interfering noise sources and reverberation." *IEEE Transactions on Audio, Speech, and Language Processing* 20 (7): 2016–2030.

May, Tobias, Steven van de Par, and Armin Kohlrausch. 2013. "Binaural localization and detection of speakers in complex acoustic scenes." In *The technology of binaural listening,* 397–425. Springer.

McLoughlin, Ian, Haomin Zhang, Zhipeng Xie, Yan Song, Wei Xiao, and Huy Phan. 2017. "Continuous robust sound event classification using time-frequency features and deep learning." *PloS one* 12 (9).

Meddis, Ray, Lowel P O'Mard, and Enrique A Lopez-Poveda. 2001. "A computational algorithm for computing nonlinear auditory frequency selectivity." *The Journal of the Acoustical Society of America* 109 (6): 2852–2861.

Merimaa, Juha, Timo Peltonen, and Tapio Lokki. 2005. *Concert Hall Impulse Responses Pori, Finland: Reference.*

Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. 2017. "Regularizing and Optimizing LSTM Language Models." *arXiv preprint arXiv:1708.02182.*

Mesaros, A., T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley. 2018. "Detection and Classification of Acoustic Scenes and Events: Outcome of the DCASE 2016 Challenge." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26 (2): 379–393.

Mesaros, A., T. Heittola, A. Eronen, and T. Virtanen. 2010. "Acoustic event detection in real life recordings." In *2010 18th European Signal Processing Conference (EUSIPCO),* 1267–1271.

Mesaros, Annamaria, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. 2019. "Sound event detection in the DCASE 2017 Challenge." In press, *IEEE/ACM Transactions on Audio, Speech, and Language Processing.*

Mesaros, Annamaria, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. 2017. "DCASE 2017 challenge setup: Tasks, datasets and baseline system." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017).*

Mesaros, Annamaria, Toni Heittola, and Tuomas Virtanen. 2016. "Metrics for Polyphonic Sound Event Detection." *Applied Sciences* 6 (162): 162.

Misra, Hemant, Shajith Ikbal, Hervé Bourlard, and Hynek Hermansky. 2004. "Spectral entropy based feature for robust ASR." In *2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Mitra, Vikramjit, Wen Wang, Horacio Franco, Yun Lei, Chris Bartels, and Martin Graciarena. 2014. "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions." In *Proc. Interspeech 2014.*

Moritz, Niko, Jörn Anemüller, and Birger Kollmeier. 2015. "An Auditory Inspired Amplitude Modulation Filter Bank for Robust Feature Extraction in Automatic Speech Recognition." *IEEE/ACM Transactions on Audio, Speech and Language Processing* 23 (11): 1926–1937.

Nadiri, O., and B. Rafaely. 2014. "Localization of Multiple Speakers Under High Reverberation Using a Spherical Microphone Array and the Direct-path Dominance Test." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22 (10): 1494–1505.

Ng, Andrew Y. 2004. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance." In *Proceedings of the 21st international conference on Machine learning*, 78. ACM.

Niessen, Maria E, Leendert van Maanen, and Tjeerd C Andringa. 2008. "Disambiguating Sounds through Context." In *2008 IEEE International Conference on Semantic Computing*, 88–95.

Nogueira, Waldo. 2016. "Sound Scene Identification Based on Monaural and Binaural Features." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016).*

Nogueira, Waldo, Gerard Roma, and Perfecto Herrera. 2013. "Sound scene identification based on MFCC, binaural features and a support vector machine classifier." *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events.*

Oord, Aäron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. "WaveNet: A Generative Model for Raw Audio." In *9th ISCA Speech Synthesis Workshop*, 125–125.

Parascandolo, Giambattista, Heikki Huttunen, and Tuomas Virtanen. 2016. "Recurrent neural networks for polyphonic sound event detection in real life recordings." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6440–6444.

Patterson, Roy D, and John Holdsworth. 1996. "A functional model of neural activity patterns and auditory images." *Advances in speech, hearing and language processing* 3 (Part B): 547–563.

Phan, H., M. Maass, L. Hertel, R. Mazur, and A. Mertins. 2015. "A multi-channel fusion framework for audio event detection." In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–5.

Phan, Huy, Oliver Y Chén, Philipp Koch, Lam Pham, Ian McLoughlin, Alfred Mertins, and Maarten De Vos. 2018. "Beyond Equal-Length Snippets: How Long is Sufficient to Recognize an Audio Scene?" Accepted to 2019 AES Conference on Audio Forensics, *arXiv preprint arXiv:1811.01095*.

Phan, Huy, Lars Hertel, Marco Maass, and Alfred Mertins. 2016. "Robust audio event recognition with 1-max pooling convolutional neural networks." In *Proc. Interspeech 2016*, 3653–3657. 1.

Phan, Huy, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, Ian McLoughlin, and Alfred Mertins. 2017. "What makes audio event detection harder than classification?" In *2017 25th European Signal Processing Conference (EUSIPCO)*, 2739–2743. IEEE.

Piczak, Karol J. 2015a. "Environmental sound classification with convolutional neural networks." In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE.

Piczak, Karol J. 2015b. "ESC: Dataset for environmental sound classification." In *Proceedings of the 23rd ACM international conference on Multimedia*, 1015–1018. ACM.

Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018). 2018. Tampere University of Technology. Laboratory of Signal Processing.

Powers, D. M. H. 2011. "Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation." *Journal of Machine Learning Technologies* 2 (1): 37–63.

Purwins, Hendrik, Bo Li, Tuomas Virtanen, Jan Schluter, Shuo-Yiin Chang, and Tara N Sainath. 2019. "Deep Learning for Audio Signal Processing." *IEEE Journal of Selected Topics in Signal Processing*.

Qian, J., T. Hastie, J. Friedman, R. Tibshirani, and N. Simon. 2013. "Glmnet for Matlab." http://gts.sourceforge.net/.

Raake, Alexander, Jens Blauert, Jonas Braasch, Guy Brown, Patrick Danès, Thorsten Dau, Bruno Gas, et al. 2014. *Two!Ears – Integral interactive model of auditory perception and experience.*

Rabaoui, Asma, Manuel Davy, Stéphane Rossignol, and Noureddine Ellouze. 2008. "Using one-class SVMs and wavelets for audio surveillance." *IEEE Transactions on information forensics and security* 3 (4): 763–775.

Rajnoha, J. 2009. "Multi-condition training for unknown environment adaptation in robust asr under real conditions." *Acta Polytechnica* 49 (2).

Rickard, Scott. 2007. "The DUET blind source separation algorithm." In *Blind speech separation,* edited by Shoji Makino, Hiroshi Sawada, and Te-Won Lee, 217–241. Springer.

Risley, Robert, Valeriy Shafiro, Stanley Sheft, Adam Balser, and Brian Gygi. 2012. "The role of context in the perception of environmental sounds." In *Proceedings of Meetings on Acoustics 163ASA,* vol. 15. 1.

Rosenthal, David F, and Hiroshi G Okuno. 1998. *Computational auditory scene analysis.* Lawrence Erlbaum Associates Publishers.

Saeidi, Rahim, Pejman Mowlaee, Tomi Kinnunen, Zheng-Hua Tan, Mads Graesboll Christensen, Soren Holdt Jensen, and Pasi Franti. 2010. "Signal-to-signal ratio independent speaker identification for co-channel speech signals." In *2010 20th International Conference on Pattern Recognition (ICPR),* 4565–4568. IEEE.

Sainath, Tara N., and Bo Li. 2016. "Modeling Time-Frequency Patterns with LSTM vs. Convolutional Architectures for LVCSR Tasks." In *Interspeech 2016,* 813–817.

Salamon, J., C. Jacoby, and J. P. Bello. 2014. "A Dataset and Taxonomy for Urban Sound Research." In *22nd ACM International Conference on Multimedia,* 1041–1044.

Salamon, Justin, and Juan Pablo Bello. 2017. "Deep convolutional neural networks and data augmentation for environmental sound classification." *IEEE Signal Processing Letters* 24 (3): 279–283.

Salimans, Tim, and Diederik P Kingma. 2016. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks." In *Advances in Neural Information Processing Systems,* 901–909.

Saxe, Andrew M., James L. McClelland, and Surya Ganguli. 2013. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks." *arXiv preprint arXiv:1312.6120.*

Schädler, Marc René, Bernd T. Meyer, and Birger Kollmeier. 2012. "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition." *The Journal of the Acoustical Society of America* 131 (5): 4134–4151.

Scholler, S., and H. Purwins. 2011. "Sparse Approximations for Drum Sound Classification." *IEEE Journal of Selected Topics in Signal Processing* 5 (5): 933–940.

Schröder, Jens, Stefan Goetze, and Jörn Anemüller. 2015. "Spectro-temporal Gabor filterbank features for acoustic event detection." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23 (12): 2198–2208.

Schröder, Jens, Niko Moritz, Marc René Schädler, Benjamin Cauchi, Kamil Adiloglu, Jörn Anemüller, Simon Doclo, Birger Kollmeier, and Stefan Goetze. 2013. "On the use of spectro-temporal features for the IEEE AASP challenge 'detection and classification of acoustic scenes and events'." In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–4. IEEE.

Serizel, Romain, Victor Bisot, Slim Essid, and Gaël Richard. 2018. "Acoustic Features for Environmental Sound Analysis." In *Computational Analysis of Sound Scenes and Events*, edited by Tuomas Virtanen, Mark D. Plumbley, and Dan Ellis, 71–101. Springer International Publishing.

Shan, Siyuan, and Yi Ren. 2018. *Automatic Audio Tagging With 1d And 2d Convolutional Neural Networks.* Technical report. DCASE2018 Challenge.

Shannon, Robert V, Fan-Gang Zeng, Vivek Kamath, John Wygonski, and Michael Ekelid. 1995. "Speech recognition with primarily temporal cues." *Science* 270 (5234): 303–304.

Sharan, Roneel V, and Tom J Moir. 2016. "An overview of applications and advancements in automatic sound recognition." *Neurocomputing* 200:22–34.

Smith, Evan C, and Michael S Lewicki. 2006. "Efficient auditory coding." *Nature* 439 (7079): 978.

Smith, Zachary M, Bertrand Delgutte, and Andrew J Oxenham. 2002. "Chimaeric sounds reveal dichotomies in auditory perception." *Nature* 416 (6876): 87.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15:1929–1958.

Stiefelhagen, Rainer, Keni Bernardin, Rachel Bowers, R Travis Rose, Martial Michel, and John Garofolo. 2007. "The CLEAR 2007 evaluation." In *Multimodal Technologies for Perception of Humans*, 3–34.

"StockMusic.com." 2014. https://www.stockmusic.com.

Stowell, D., Y. Stylianou, M. Wood, H. Pamuła, and H. Glotin. 2018. "Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge." *Methods in Ecology and Evolution*. https://arxiv.org/abs/1807.05812].

Stowell, Dan, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. 2015. "Detection and classification of acoustic scenes and events." *IEEE Transactions on Multimedia* 17 (10): 1733–1746.

Sumitani, S., R. Suzuki, N. Chiba, S. Matsubayashi, T. Arita, K. Nakadai, and H. G. Okuno. 2019. "An Integrated Framework for Field Recording, Localization, Classification and Annotation of Birdsongs Using Robot Audition Techniques — Harkbird 2.0." In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8246–8250.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. "Sequence to Sequence Learning with Neural Networks." In *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 3104–3112. Curran Associates, Inc.

Takahashi, Naoya, Michael Gygli, Beat Pfister, and Luc Van Gool. 2016. "Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition." In *Proc. Interspeech 2016*, 2982–2986.

Tibshirani, Robert. 1996. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society. Series B (Methodological)*: 267–288.

Tokozume, Yuji, Yoshitaka Ushiku, and Tatsuya Harada. 2017. "Learning from between-class examples for deep sound recognition." *arXiv preprint arXiv:1711.10282*.

Tompson, Jonathan, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. "Efficient object localization using convolutional networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 648–656.

Trowitzsch, Ivo. 2019. "Supplementary materials to TASLP: SELD through spatial segregation." https://github.com/nigroup/Supplementaries-to-TASLP-SELD-Spatial-Segregation.

Trowitzsch, Ivo, Youssef Kashef, and Klaus Obermayer. 2019. "Auditory Machine Learning Training and Testing Pipeline: AMLTTP v3.0." https://doi.org/10.5281/zenodo.2575086.

Trowitzsch, Ivo, Johannes Mohr, Youssef Kashef, and Klaus Obermayer. 2017. "Robust Detection of Environmental Sounds in Binaural Auditory Scenes." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (6): 1344–1356.

Trowitzsch, Ivo, Christopher Schymura, Dorothea Kolossa, and Klaus Obermayer. 2019. "Joining Sound Event Detection and Localization Through Spatial Segregation." In review for IEEE/ACM Transactions on Audio, Speech, and Language Processing, *arXiv preprint arXiv:1904.00055*.

Trowitzsch, Ivo, Jalil Taghia, Youssef Kashef, and Klaus Obermayer. 2019a. "NIGENS general sound events database." https://doi.org/10.5281/zenodo.2535878.

Trowitzsch, Ivo, Jalil Taghia, Youssef Kashef, and Klaus Obermayer. 2019b. *The NIGENS General Sound Events Database.* Technical report. Technische Universität Berlin. eprint: arXiv:1902.08314.

Two!Ears Team. 2018. "Two!Ears Auditory Model 1.5." https://doi.org/10.5281/zenodo.1458420.

Valero, Xavier, and Francesc Alias. 2012. "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification." *IEEE Transactions on Multimedia* 14 (6): 1684–1689.

Wan, Li, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. "Regularization of Neural Networks using DropConnect." In *International Conference on Machine Learning*, 1058–1066.

Wang, DeLiang, and Guy J Brown. 2006. *Computational auditory scene analysis: Principles, algorithms, and applications.* Wiley-IEEE press.

Wang, Yun, and Florian Metze. 2017. "A Transfer Learning Based Feature Extractor for Polyphonic Sound Event Detection Using Connectionist Temporal Classification." In *Proc. Interspeech 2017.*

Wei, Shengyun, Kele Xu, Dezhi Wang, Feifan Liao, Huaimin Wang, and Qiuqiang Kong. 2018. "Sample mixed-based data augmentation for domestic audio tagging." In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 93–97.

Weinzierl, Stefan. 2008. *Handbuch der Audiotechnik.* Springer Science & Business Media.

Wierstorf, Hagen, Matthias Geier, and Sascha Spors. 2011. "A Free Database of Head Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances." In *Audio Engineering Society Convention 130.*

Wightman, Frederic L., and Doris J. Kistler. 1992. "The dominant role of low-frequency interaural time differences in sound localization." *The Journal of the Acoustical Society of America* 91 (3): 1648–1661.

Winter, Fiete, Hagen Wierstorf, Ariel Podlubne, Thomas Forgue, Jérome Manhès, Matthieu Herrb, Sascha Spors, Alexander Raake, and Patrick Danès. 2016. "Database of Binaural Room Impulse Responses of an Apartment-Like Environment." In *Audio Engineering Society Convention 140.*

Winter, Fiete, Hagen Wierstorf, and Ivo Trowitzsch. 2016. *Deliverable 1.3: Final database of audio-visual scenarios.* Final report. Two!Ears Project. http://twoears.eu/wp-content/uploads/deliverables/D1.3_final_database_of_audio-visual_scenarios.pdf.

Wu, Jibin, Yansong Chua, Malu Zhang, Haizhou Li, and Kay Chen Tan. 2018. "A spiking neural network framework for robust sound classification." *Frontiers in neuroscience* 12.

Xia, Xianjun, Roberto Togneri, Ferdous Sohel, Yuanjun Zhao, and Defeng Huang. 2019. "A Survey: Neural Network-Based Deep Learning for Acoustic Event Detection." *Circuits, Systems, and Signal Processing.*

Xingjian, SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In *Advances in neural information processing systems*, 802–810.

Xu, Yong, Qiuqiang Kong, Qiang Huang, Wenwu Wang, and Mark D. Plumbley. 2017. "Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging." In *2017 International Joint Conference on Neural Networks (IJCNN)*, 3461–3466.

Xu, Yong, Qiuqiang Kong, Wenwu Wang, and Mark D. Plumbley. 2017. *Surrey-CVSSP System for DCASE2017 Challenge Task4.* Technical report. DCASE2017 Challenge.

Yamakawa, Nobuhide, Tetsuro Kitahara, Toru Takahashi, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2010. "Effects of modelling within- and between-frame temporal variations in power spectra on non-verbal sound recognition." In *Proc. Interspeech 2010*, 2342–2345.

Yin, Shi, Chao Liu, Zhiyong Zhang, Yiye Lin, Dong Wang, Javier Tejedor, Thomas Fang Zheng, and Yinguo Li. 2015. "Noisy training for deep neural networks in speech recognition." *EURASIP Journal on Audio, Speech, and Music Processing* 2015 (1): 2.

Youden, William J. 1950. "Index for rating diagnostic tests." *Cancer* 3 (1): 32–35.

Yu, Hong, Achintya Sarkar, Dennis Alexander Lehmann Thomsen, Zheng-Hua Tan, Zhanyu Ma, and Jun Guo. 2016. "Effect of multi-condition training and speech enhancement methods on spoofing detection." In *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, 1–5. IEEE.

Yu, Qiang, Yanli Yao, Longbiao Wang, Huajin Tang, Jianwu Dang, and Kay Chen Tan. 2019. "Robust Environmental Sound Recognition with Sparse Key-point Encoding and Efficient Multi-spike Learning." *arXiv preprint arXiv:1902.01094.*

Zhang, Haomin, Ian McLoughlin, and Yan Song. 2015. "Robust sound event recognition using convolutional neural networks." In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 559–563. IEEE.

Zhang, Hongyi, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412.*

Zhang, Zixing, Ding Liu, Jing Han, and Björn Schuller. 2017. "Learning audio sequence representations for acoustic event classification." *arXiv preprint arXiv:1707.08729.*

Zhuang, Xiaodan, Xi Zhou, Mark A Hasegawa-Johnson, and Thomas S Huang. 2010. "Real-world acoustic event detection." *Pattern Recognition Letters* 31 (12): 1543–1551.

DECLARATION

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of Technische Universität's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Berlin, June 2019

Ivo Trowitzsch

This document was typeset with LaTeX, using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić, adapted to my needs.

Thank you very much for reading. Feedback will be appreciated.