Stefan Kahl

Identifying Birds by Sound:
Large-scale Acoustic Event Recognition for Avian Activity Monitoring

Wissenschaftliche Schriftenreihe

Dissertationen der Medieninformatik

Band 10

Prof. Dr. Maximilian Eibl (Hrsg.)

Stefan Kahl

Identifying Birds by Sound: Large-scale Acoustic Event Recognition for Avian Activity Monitoring

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Universitätsverlag Chemnitz
2020

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über http://www.dnb.de abrufbar.

Titelgrafik: Eugene Brennan - EugenesDIYDen / pixabay.com
Satz/Layout: Stefan Kahl

Technische Universität Chemnitz/Universitätsbibliothek
Universitätsverlag Chemnitz
09107 Chemnitz
https://www.tu-chemnitz.de/ub/univerlag

readbox unipress
in der readbox publishing GmbH
Rheinische Straße 171
44147 Dortmund
https://www.readbox.net/unipress

ISSN 2195-2574 print - ISSN 2195-2582 online
ISBN 978-3-96100-110-1
http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa2-369869

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Identifying Birds by Sound: Large-scale Acoustic Event Recognition for Avian Activity Monitoring

Dissertation
zur Erlangung des akademischen Grades

Dr. rer. nat.

Herr Stefan Kahl, M.Sc.
geboren am 23.08.1984 in Lutherstadt Wittenberg

Fakultät für Informatik an der
Technischen Universität Chemnitz

Tag der Einreichung: 05. September 2019
Tag der Verteidigung: 13. Dezember 2019

Gutachter:

Prof. Dr. Maximilian Eibl
Technische Universität Chemnitz

Prof. Dr. Marc Ritter
Hochschule Mittweida

"If it looks like a duck, and quacks like a duck, we have at least to consider the possibility that we have a small aquatic bird of the family anatidae on our hands."

Douglas Adams, Dirk Gently's Holistic Detective Agency [Adams, 1987].

Abstract

Birds are omnipresent and often reveal their presence through their vocalizations. They respond to environmental changes over many spatial scales and are thus ideal indicator species to monitor ecosystem health across various lifeforms. Automated observation of avian vocal activity and species diversity can be a transformative tool for ornithologists, conservation biologists, and bird watchers to assist in long-term monitoring of critical environmental niches. Digital sound transformation is commonly used when studying bird sounds. Since the inception of the sound spectrograph, spectrograms play a significant role in avian research. We can assume that visual representations of bird sounds contain valuable information on species identity, rendering spectrograms a particularly suitable representation. Deep artificial neural networks have surpassed traditional classifiers in the field of visual recognition and acoustic event classification. Still, deep neural networks require expert knowledge to design, train, and test powerful models. With this constraint and the requirements of future applications in mind, an extensive research platform for automated avian activity monitoring was developed: BirdNET. An unprecedented amount of training, validation, and test data was used to assess the overall system performance on more then 3,900 hours of field recordings covering 987 classes and almost 300 hours of fully annotated soundscapes containing almost 80,000 vocalizations. The resulting benchmark system yields state-of-the-art scores across various acoustic domains and was used to develop expert tools and public demonstrators that can help to advance the democratization of scientific progress and future conservation efforts.

Acknowledgements

At this point I would like to thank my supervisors Maximilian Eibl, Marc Ritter, Holger Klinck, and Danny Kowerko who gave me the funding, professional advice, freedom, and opportunities without which this dissertation would not have been possible. I would also like to thank Karolin and Holger Klinck for their incredible support, hospitality, kindness, and friendship, which made my work on this thesis so exciting and fun. This project and the development of BirdNET would not have been possible without the help of many incredibly smart people who have contributed their time, knowledge, and engineering efforts. To name but a few, Shyam Madhusudhana, David Winiarski, Nick Gershfeld (all Cornell Lab of Ornithology), and David Mann (loggerhead.com) helped build sophisticated prototypes that run BirdNET. Incredible effort was made by Alec Hopping, Cullen Hanks, Jay McGowan, Matt Young, Randy Little, Russ Charif, and Sarah Dzielski (all Cornell Lab of Ornithology) to annotate soundscape recordings. I am grateful to Willem-Pier Vellinga (Xeno-canto) and Jessie Barry (Macaulay Library) who provided me with tens of thousands of training recordings. Thanks to Alexis Joly (INRIA), Hervé Goëau (CIRAD), Fabian-Robert Stöter (INRIA), Ivan Eggel (HES-SO), and Henning Müller (HES-SO) who helped me to organize the 2019 edition of BirdCLEF. I would like to thank Tom Denton, Rif A. Saurous, and Dick Lyon of Google for their advice, ideas, and help in improving the accuracy of BirdNET. Many thanks to Jan Schlüter (OFAI, Université de Toulon) for his help in the development of tailor-made functions for the detection of birdsong for Lasagne and Theano. My wonderful colleagues at Chemnitz University of Technology have been very supportive over the years and I am deeply grateful for their advice, help, ideas, collaboration, and friendship. And, of course, none of this would have ever been possible without the loving support of my family. Thank you!

This project was kindly supported by Jake Holshuh (Cornell class of '69) and the Arthur Vining Davis Foundations. The European Union and the European Social Fund for Germany (ESF) partially funded this research. This work was also partially funded by the German Federal Ministry of Education and Research in the program of Entrepreneurial Regions InnoProfileTransfer in the project group localizeIT (funding code 03IPT608X).

Contents

1 Motivation and Contributions 1

1.1 Avian ecology . . . 1
1.2 Democratizing deep neural networks . . . 3
1.3 Methodology and outline . . . 4
1.4 Results . . . 6

2 Theory on Bird Biology and Audio Signal Processing 11

2.1 On the relationship of humans and birds . . . 12
2.2 Avian vocal behavior . . . 13
  2.2.1 Studying bird sounds . . . 14
  2.2.2 Production and perception of sounds in birds . . . 17
  2.2.3 Vocal development and repertoires . . . 26
  2.2.4 Song variation in space and time . . . 31
  2.2.5 Function of bird song and singing behavior . . . 34
2.3 Avian ecology . . . 36
  2.3.1 Habitats, abundance, and migration . . . 37
  2.3.2 Conservation biology and citizen science . . . 39
2.4 Sound digitization and representation . . . 44
  2.4.1 Digital sound recording . . . 44
  2.4.2 Spectrogram computation . . . 47
  2.4.3 Adaption to avian acoustic monitoring . . . 55
2.5 Summary . . . 58

3 Acoustic Events and Deep Learning 61

3.1 Acoustic event recognition . . . 61
3.2 Deep artificial neural networks . . . 64
  3.2.1 Concepts and topologies . . . 64

3.2.2 Evolution of CNN 73
3.3 Evaluation campaigns for avian acoustics 76
3.3.1 BirdCLEF 77
3.3.2 DCASE Bird Audio Detection 81
3.4 Summary 84

4 System Architecture 85
4.1 Design Decisions 86
4.1.1 Extensive functionality 87
4.1.2 Detailed configuration 90
4.1.3 Domain-agnostic workflow 91
4.1.4 Reproducibility and transparency 91
4.1.5 Distribution and applications 92
4.2 Concepts and workflows 94
4.2.1 Components 94
4.2.2 Training 96
4.2.3 Evaluation 106
4.2.4 Distribution 111
4.3 Implementation details 112
4.3.1 Audio and image processing 113
4.3.2 Deep learning frameworks 114
4.3.3 Web tools and services 116
4.4 Summary 117

5 Experiments 119
5.1 Goals and main focus 120
5.2 Experimental investigations 120
5.2.1 Acquisition and composition of data 121
5.2.2 Experimental setup 125
5.2.3 Spectrogram computation 129
5.2.4 Architecture design 133
5.2.5 Topologies and training regimes 141
5.2.6 Mobile architectures 152
5.2.7 Benchmark system 157
5.3 Results 166

5.4 Summary 181

6 Applications 183
6.1 Live stream analysis 183
6.1.1 Abundance and vocal activity 185
6.1.2 Temperature and vocal activity 186
6.2 SWIFT soundscape analysis 189
6.2.1 AMTiC 189
6.2.2 SWAMP 193
6.3 Smartphone app 197
6.3.1 Scope 198
6.3.2 Design 198
6.3.3 Distribution 200
6.3.4 Reception 201
6.3.5 Results 202
6.4 HaikuBox 208
6.5 Summary 209

7 Conclusion and Future Work 211

Bibliography XIX

A Year-round Vocal Activity A1

B AMTiC Detection Plots A5

C SWAMP Spatial Maps A10

D Species-specific Results A16

Listen to bird sounds online:

While reading, you will come across several examples of bird sounds that I considered useful for the narrative of this thesis. I decided to create a web page that lets you listen to those examples instead of just looking at spectrograms or transcriptions. Whenever you see a speaker icon, you can scan the QR code above to listen to the corresponding bird sounds online at:

https://birdnet.cornell.edu/samples

1. Motivation and Contributions

1.1 Avian ecology 1
1.2 Democratizing deep neural networks 3
1.3 Methodology and outline 4
1.4 Results 6

The assessment of ecosystem health through long-term monitoring of avian activity is a cornerstone of conservation biology [Gregory et al., 2004, p. 17]. The availability of high-end recording equipment led to vast archives of audio content collected by an active community of bird watchers around the globe. Including the public in the process of habitat monitoring is very likely to succeed due to the interest that birds spark in many people. Yet, not everyone can identify birds by sight or sound which would require new tools that not only record and analyze but also teach the process of bird identification by eye and ear.

1.1. Avian ecology

In avian ecology, point counts are of particular importance to conduct surveys regarding avian activity and diversity (see Section 2.3). In that field, autonomous recording units (ARU) mark the starting point of automation. The costs of mobile recorders, which can last several days or weeks, were drastically reduced over the past few years[1]. With an increase in automation comes the need for computerized analysis. Large amounts of data and hundreds of recorded soundscapes require fast and reliable processing—methods from the field of (deep) machine learning might be able to provide that.

Identifying birds (in this case by sound) is not a trivial task. The diversity of avian vocalizations poses a considerable challenge to any automated recognition

---

[1] The SWIFT recording unit of the Cornell Lab of Ornithology costs $250 in small series production and contains high-quality, durable recording equipment.

system. Aside from technical constraints like sensitivity, noise floor, polar patterns, or size, automated bird sound detection has to deal with a vast variety of avian vocalizations. One of the most frequently asked questions about such a system is whether it would be like "*Shazam for birds*". Considering what fuels the song recognition capabilities of *Shazam*, the answer must be 'No, it's more complicated than that'. Recognition technology in the field of audio fingerprinting relies on matching short audio chunks with entries in a database to derive similarity scores [Weare, 2006], [Chandrasekhar et al., 2011]. This way, audio files can be identified quickly, mostly independent of recording quality or length of the recording. However, this technology has two major drawbacks: The queried audio file has to exist in the database, and the file must not change over time to guarantee high recognition performance.

Fingerprinting has only very limited application for bird song recognition. The avian vocal tract enables birds to emit sounds of great variety and complexity. Avian auditory physiology suggests that birds (especially true songbirds, oscines) are able to learn, mimic, or invent vocalizations (see Section 2.2.2). Aside from that, birds adapt to environmental niches by altering their vocal output. We can assume that variation in space and time will lead to an ever-changing repertoire of songs and calls. If each individual of an oscine species learns vocalizations based on territorial neighbors, habitat, or time of the year, we have to assume that fingerprinting does not provide the technological foundation to develop a robust recognition system. An example might help to grasp the outreach of this assumption: When presented with a never before aired *Beatles* song, a song that seemed forgotten, a recognition system based on fingerprints (like *Shazam*) would almost certainly not be able to identify it. It might not even be able to recognize the band, whereas the casual listener would probably make the right guess. In birds, local dialects, mimicry, or vast song repertoires of multiple hundreds of songs would require to record every possible bird sound in a database. To some extend, the same acoustic features used for fingerprinting (mostly MFCC, see Section 3.3.1), would suffice for a number of species that are not able to learn, but they would most likely not suffice to grasp the vast variety of bird sounds in general. The ability to derive learned embeddings based on statistical features of large and diverse input value distributions and the potential to generalize on unseen samples might render deep neural networks an applicable choice for bird sound recognition.

1.2. Democratizing deep neural networks

Processing large amounts of audio-visual data has led to a paradigm shift in machine learning. With the emergence of deep artificial neural networks (DNN)[2], classic programming was almost entirely replaced by more generic approaches in that domain. Today, extremely complex DNN solve a vast amount of tasks, often with uncanny performance. And yet, we are still far away from true computer intelligence, despite the fact that artificial neural networks were designed with human cognition in mind. The past years have seen a tremendous hype evolving around topics of so-called 'AI'. Some of this excitement might even be justified. Countless smart-devices that we use every day have already been taken over by a significant number of DNN solving tasks like face recognition, text completion, or speech synthesis [Ignatov et al., 2018]. Hardware accelerated processing was key for the success of DNN in the early 2010s, and it is key to many applications today.

Often purely based on statistics, DNN depend on hundreds, sometimes even thousands of examples to learn a generic representation of objects, texts, scenes or other complex value distributions. Additionally, designing, training, and testing a neural network requires expert (domain) knowledge and intuition due to the holistic nature of implementation details and hyperparameters. Despite the vast output of publications in this field of research, the process of democratizing this technology might come to an abrupt halt in the near future. The need for data and computing power led to some disturbing developments in the recent past. A striking negative example is AutoAugment by Google Brain [Cubuk et al., 2018]. In this paper, Cubuk et al. present an adaptive way of generating augmented samples for training that are ideal for the task and yield new state-of-the-art results across a number of popular benchmark datasets. Yet, 15,000 GPU hours were needed to optimize this form of data augmentation for the ImageNet dataset alone. This kind of computing power is available to very few global players only and the results are not applicable for reproduction. In early 2019, OpenAI (founded by entrepreneur Elon Musk) developed language models that were able to perform almost human-like text generation [Radford et al., 2019]. In contradiction to its name, OpenAI decided not to release the code and model for public inspection "*[d]ue to concerns about large language models being used to generate deceptive, biased, or abusive language at scale [...]*"[3]. Many scientists objected this strategy, arguing "*[...] that deceptive*

---

[2] A rather delayed inception which vastly accelerated in 2012 with AlexNet. See Section 3.2

[3] https://openai.com/blog/better-language-models/, 2019-08-13

technologies lose most of their powers if the public is broadly aware of the potential for manipulation."[4]

The central motivation for this dissertation lies in the idea that computer science provides (open source) tools that help people to solve (complex) tasks in new ways. But those tools are only truly powerful when their design involves the people who are going to use them—like Frederick P. Brooks, Jr. proposed in his ACM Allen Newell Award acceptance lecture at SIGGRAPH [Brooks Jr, 1996]:

> *"If the computer scientist is a toolsmith, and if our delight is to fashion power tools and amplifiers for minds, we must partner with those who will use our tools, those whose intelligences we hope to amplify."*

Training and applying deep neural networks to new task domains can help to achieve just that. The field of bioacoustics often relies on the analysis of large amounts of collected data. The goal is "*[...] to collect and interpret sounds in nature by developing and applying innovative conservation technologies across multiple ecological scales to inspire and inform conservation of wildlife and habitats.*"[5] Current environmental changes (like global warming) amplify the need for transformative tools for many research groups alike. Two essential questions evolve from that: How can the field of deep learning contribute to bioacoustics? How can large-scale data analysis provide new insights that might help to cope with environmental issues of our time? In this thesis, I will try to answer both questions with emphasize on avian ecology.

1.3. Methodology and outline

As part of my research on avian vocalizations, I will provide basic insights into the avian vocal tract and auditory system in Chapter 2. I will shed light on song learning and imitations, local dialects, and repertoires but will solely focus on aspects that have an explicit implication on automated bird sound recognition. Some dimensions—like song tutoring—have only limited impact, whereas others demand task-specific solutions. The methodology of this thesis follows this principle whenever a theoretical overview is presented. The application of methods from the field of deep learning for visual recognition to the domain of acoustic event identification might appear to be somewhat unjustified when considering audio data as purely

---

[4] https://thegradient.pub/openai-please-open-source-your-language-model/, 2019-08-13

[5] Mission statement of the Bioacoustics Research Program of the Cornell Lab or Ornithology

sequential. However, one of the most important tools in avian research is the visualization of audio signals in form of spectrograms [Kroodsma, 2005, p. 2].

We can assume that birds encode their species identity in their vocalizations. This means that we can also assume that species identity is encoded in spectrograms as well. Still, it remains unclear *how* species identity is encoded, which leads to the following question: What are the spectrogram characteristics that ensure high applicability for the recognition of bird vocalizations? I will explore the process of spectrogram generation with respect to avian vocal behavior in Section 2.4.2.

Using DNN to identify bird sounds in audio data is not entirely new. In fact, two major evaluation campaigns focus on this topic since 2014 (see Section 3.3). The progress in this domain has led to huge leaps in performance over the past editions. Similar to the results of the ILSVRC competition, the arrival of deep learning quickly transformed the entire field of avian acoustics. Today, almost every proposed system in this field relies on DNN. Due to this, I will summarize recent advances and introduce the basic concepts of deep learning in Chapter 3. The focus of this part will once again be on task-specific considerations, relevant technologies, and high-level concepts only.

Extensive experimentation is often key to success and ensures general applicability. The high number of variables in a deep learning system requires vast domain knowledge that renders most implementations inaccessible. Modern frameworks like Lasagne, Keras, PyTorch, TensorFlow, or MXNet provide vast functionality to develop such a system on a high abstraction level. In Chapter 4, I will propose a software research platform that combines this functionality with task-agnostic, high-level workflows to design, train, validate, and apply DNN for a variety of tasks: BirdNET. Chapter 4 will focus on the core aspects of this platform and introduce components, implementation details, and interfaces. BirdNET will serve as the primary tool for evaluation with unprecedented collections of audio data and state-of-the-art DNN models.

The design of the experimental studies in Chapter 5 evolves around the ability of DNN to generalize on unseen samples despite a high number of classes with significant intra-class heterogeneity. The investigation of core strategies and components should provide generalizable insights into the process of developing a robust system for automated bird sound recognition. The proposed approach involves the acquisition of large training, validation, and test data that represents real-world use cases, establishing a baseline setup to test certain hypotheses that concern spectrogram extraction, architectural designs, as well as various DNN topologies and their corresponding training regimes, and eventually the application of the experimentally validated system to investigate the influence of avian vocal diversity on the overall results. The established benchmark system will be used to confirm basic avian behavioral patterns during a number of application scenarios.

Handling extremely large amounts of data is the core of BirdNET's application to real-world monitoring scenarios. I will present four different demos and prototypes that employ the benchmark system to detect avian vocalizations in diverse soundscape data in Chapter 6. First, an online live stream demo analyzes year-round recordings from Sapsucker Woods, Ithaca, USA recorded by an outdoor microphone. Placed near a bird feeder and a pond, the microphone provides diverse soundscapes that reveal different levels of correlation between abundance, temperature, and vocal activity. Secondly, two monitoring scenarios are presented that use ARU (i.e. the aforementioned SWIFT recording unit) and assess biodiversity at monitoring sites in Germany and the USA. Both scenarios provide insights into (spatio-) temporal behavioral patterns and strength and weaknesses of the proposed benchmark system. Thirdly, I will explore recent observations that were made by users of a BirdNET smartphone app. This application focuses on teaching birding by ear while at the same time involving the public in the process of avian monitoring. Democratizing deep learning technologies to develop tools for a wide audience is one of the core aspects of this thesis. The smartphone app provides evidence that this approach indeed leads to increased user engagement. Finally, a fully automated monitoring station—consisting of a solar powered Raspberry Pi and a fly-through bird feeder—provides an overview of how BirdNET could help close the gap between dense but spatially confined recording arrays and highly distributed but non-continuous mobile recording devices (i.e. smartphones). The so-called HaikuBox will be distributed in the USA so that schools and other educational institutions can maintain an autonomous monitoring station that provides year-round observation data.

1.4. Results

BirdNET is a versatile research platform and follows in the footsteps of AMOPA ([Ritter, 2014]) and the Xtrieval Web Lab ([Wilhelm-Stein, 2016]). It provides an ecosystem of components that can be used to form processing chains to build applications for an expanding list of scenarios. And yet, BirdNET itself is an expert tool that requires expert knowledge to configure and use. Despite its high abstraction level of functionality and centralized fine-grained settings, public demonstrators and prototypes are its core strength. Task-agnostic workflows and fast processing of audio data with deep learning techniques render BirdNET a valuable experimentation platform to derive (mostly) generalizeable results.

Building upon that, the experimental design in this thesis focused on certain hypotheses that address main aspects of avian acoustic event recognition. In this sense, most of the presented results are transferable to other implementations or monitoring tasks. Fair experimental conditions for each run and comparable, reproducible results were primary concerns. Due to that, the contribution to the field of avian activity monitoring is two-fold: First, derived results by hypothesis confirmation and extensive experimentation provide a sound foundation to quickly build powerful recognition systems for long-term, large-scale analysis scenarios. Secondly, the ability to develop applications based on flexible composition of core functionality and high-level programmable interfaces allows to adjust to new use cases (mostly) independent of available hardware or task requirements.

The experimental investigation in this thesis focused on spectrogram computation, architecture design of DNN, deep and shallow topologies, implicit and explicit regularization, cost-sensitive learning, and model efficiency when computational resources are limited (see Chapter 5 for more details). The proposed workflow is task-agnostic and provides detailed insights into how current state-of-the-art technologies from the domain of visual recognition can help to solve complex acoustic monitoring tasks. All formulated hypotheses were confirmed or partially confirmed; the most notable results imply that

- Spectrograms that visualize longer chunks of audio contain more valuable information and thus result in better classification performance.
- High temporal resolution of input spectrograms (short frame length) improves the classification performance.
- Multi-label classification with mixup training increases the overall performance across all tasks.

- Deeper topologies (more layers) do not necessarily perform better than wider topologies (more filters).
- Deeper topologies do outperform shallow layouts when computational resources are limited.
- Except for oversampling, cost-sensitive learning does not improve the overall classification performance.

Starting with a benchmark system that resembled current best practices and domain knowledge from previous experience and publications in the field, incremental improvements were employed and validated. The evaluation featured a number of different complementary metrics (see Section 4.2.3) and diverse test data from various domains (see Section 5.2.1). The resulting benchmark system yields state-of-the-art scores across all validation domains, especially when compared to the scores achieved during the 2019 BirdCLEF challenge[6]. With an increase of 15.4% over the best single model that did not use validation samples for training, the proposed training regime and DNN design appear to be competitive considering the difficulty of the task. The benchmark system also revealed that

- Deep neural networks are data hungry and require large numbers of training data (in this particular case up to 750 per class).
- Signal quality of training samples significantly affects the overall classification quality and manual pre-processing might be worth the effort in some cases.
- Task-specific designs and training regimes outperform standard architectures from other campaigns and task domains.

The investigation of species-specific scores revealed that number and quality of training samples significantly impact the classification results (see Figure 5.7). Additionally, bird species diversity plays an important role, especially for species that incorporate heterospecific material into their vocalizations or those that are likely to be confused with similar classes. However, we can conclude that no single signifier implies if the overall classification performance of a species is going to be applicable. Depending on the target use case, a number of considerations have to be taken into account (e.g. computational resources, degree of manual interference, availability of

---

[6] Which were not officially published at the time of writing and are part of the annual CLEF lab proceedings.

clean training samples). Additionally, the lack of a 'gold standard' might correlate structural deficiencies of the approach with the lack of clean and time-stamped labels. Despite the extremely heterogeneous validation data (which for the most part was only weakly linked to the training data), the focus on fully automated data processing, and the renunciation of expert sample curating, the experimental results may not reflect real-world performance. It remains possible (and plausible) that empirical assessments of application performance might reveal proper overall performance despite low category scores. For most classes, we can assume that the derived scores are representative, which indicates strengths and weaknesses of the approach (for a detailed assessment of class performance see Appendix D).

In order to shed some light on the automated (i.e. learned) extraction of high-level features by deep neural networks, class activation maps were generated. This approach resulted in remarkably detailed visualizations of important vocal features. Although the generated activation maps might not reveal how birds encode species identity, they imply that certain parts of every bird vocalization are of high significance to be identifiable by the proposed recognition system. Most notably, redundant elements (as in trills) compensate information loss (e.g. Red-winged Blackbird and Wood Thrush), re-occurring elements suffice for species identification (e.g. Common Chaffinch and White-crowned Sparrow), and gaps between notes and the duration of single elements help to identify similar sounding species (e.g. Black-capped Chickadee and Tufted Titmouse). Sometimes, only a small frequency band or portion of a vocalization encode species identity (e.g. Blue Jay and Common Buzzard). For more details see Section 5.3.

Other major contributions to the field of avian acoustics and activity monitoring include high-fidelity applications for real-time audio stream analysis, large-scale soundscape processing for selected monitoring stations and recorder arrays, mobile learning tools (i.e. smartphone apps), and fully-autonomous, solar powered monitoring stations (see Chapter 6). Data visualization of results reveals spatio-temporal patterns of avian behavior and species diversity that indicate the vast potential of the proposed technology for a broad range of tasks in the domain of ecosystem health assessment. BirdNET might indeed have the potential to transform the field of bioacoustics through novel expert tools and public involvement.

2. Theory on Bird Biology and Audio Signal Processing

2.1 On the relationship of humans and birds **12**
2.2 Avian vocal behavior **13**
  2.2.1 Studying bird sounds 14
  2.2.2 Production and perception of sounds in birds 17
  2.2.3 Vocal development and repertoires 26
  2.2.4 Song variation in space and time 31
  2.2.5 Function of bird song and singing behavior 34
2.3 Avian ecology **36**
  2.3.1 Habitats, abundance, and migration 37
  2.3.2 Conservation biology and citizen science 39
2.4 Sound digitization and representation **44**
  2.4.1 Digital sound recording 44
  2.4.2 Spectrogram computation 47
  2.4.3 Adaption to avian acoustic monitoring 55
2.5 Summary **58**

This chapter aims at providing most of the theoretical background needed to follow the work of this thesis. First, I will introduce some central ideas covering the biological aspects of bird sound, conservation biology, and relevant citizen science projects. Secondly, a brief introduction to audio signal processing will cover the main aspects of digital sound representations and their adaption to human perception. The intention of this chapter is to provide an overview rather than giving in-depth insights into both fields, solely focusing on aspects that are relevant for an automated bird sound recognition system. Therefore, some methods and aspects are left out or shortened with remarks to further introductory literature.

2.1. On the relationship of humans and birds

Birds play a significant role in our lives. Most of us hear or see birds almost daily, birds are a common source of sound for humans. Many consider the songs and calls of birds relaxing but have also experienced the cacophony of a busy dawn chorus in the early hours of a springs's day. This comes as no surprise. Birds vocalize in the hearing range of humans and—for the most part—share our daily routine of wake and sleep. Avian life often depends on daylight, and many birds are busy from dawn to dusk and silent during the night.

Many people have a distinct relationship with birds that involves childhood memories and the desire for recreation as an adult. Most households maintain a bird feeder during the winter times to prevent backyard birds from starving but also to cherish the beauty and elegance of local bird species.

In contrast, identifying birds by sight or sound cannot be considered general knowledge, even for some very common species. When asked, most people note that they would wish to be able to identify birds by sound but consider this a complicated task with a steep learning curve. Some people even feel guilty for not knowing more about their environment, especially if it is about such a common environmental aspect like the songs of birds.

Furthermore, conservation biology became one of the most important scientific fields in the past years and the interest is ever-growing. Birds face many challenges because of human-induced environmental issues that include the destruction of breeding and overwintering habitats, global warming, noise and light pollution, pesticides, and large buildings. Computer science should be understood as an interface between research, citizen science, and public education. If researchers can create tools that help the public to learn more about their environment, people might regain a strong attachment to nature and thus be more open to protecting it.

Additionally, reliable identification of bird species with the help of a computer system would be a transformative tool for ornithologists, conservation biologists, and birders. Even though members of these groups are often organized in large communities, the need for participation of the public is vital for some of the most crucial projects like long-term bird counts to assess changes in ecosystems and habitats.

2.2. Avian vocal behavior

Communication between members of one bird species is not just limited to sound. However, the channels of vision and sound are more developed in birds, in stark contrast to mammals, where the olfactory system is often significantly more important for communication. For many applications of bird identification and observation, sound is also the primary source of information, especially if we consider the difficulties of identifying birds by sight over a long distance, in dense forests, or during migration that happens at night [Byers and Kroodsma, 2016, p. 382].

Avian vocalizations are extremely diverse and song is often the most complex utterance. Most of the roughly 10,000 bird species in the world produce sounds—either with their vocal tract or non-vocally using other body parts. *Passeriformes* is the largest order of birds and contains the suborder *Passeri* (oscines) which members are commonly called (true) songbirds [Lovette, 2016, pp. 51-59]. The variety of avian vocalizations is greatest in songbirds and most of them are capable to develop learned vocalizations. Independent of their taxonomic order or capability to learn, imitate, mimic, or invent, birds produce sounds built from single elements that form notes, phrases, series, warbles, trills, and—eventually—song [Pieplow, 2017, p. 8].

Avian vocalizations serve different social functions and can be divided into songs and calls—a concept that I will explore in this section. Avian vocal activity is mostly linked to annual changes in seasons that affect breeding cycles and migration. Variation in space often results in distinct local dialects that often restrict communication between individuals of different populations. [Byers and Kroodsma, 2016, pp. 382-392] All of those aspects influence the development of an automated bird sound recognition system. Therefore, this section focuses on the history of bird studies, the production and perception of bird sounds, vocal development and repertoires, song variations, and the function of bird song with emphasis on North American and European species.

2.2.1. Studying bird sounds

Reasons to study birds—and bird sounds in particular—are manifold. From a scientific standpoint, birds are ideal subjects to observe because of their omnipresence. Additionally, birds perform important ecological functions in almost every habitat. They often form the top of the food chain and thus incorporate changes on lower levels. They mainly feed on insects, anurans, fish, mammals, seeds and nectar and thus are impacted by almost every life-form in their habitat. Birds also often occupy environmental niches that are highly endangered by climate change and pollution. The observation of bird abundance in a habitat can tell us a lot about the current state of the ecosystem.

Although the study of bird song is comparatively recent—advances in recording technologies were not made before the 20th century—some work was done before that. One of the earliest episodes that supposedly ignited the science of bird song learning is reported in *'Bird song: biological themes and variations'* by Catchpole and Slater and dates back to 1773 [Catchpole and Slater, 2008, p. 2]:

> *"The Hon. Daines Barrington [...] established the existence of song learning, for example, because he heard the song of a wren emanating from a house he was passing and, knowing how difficult such birds were to keep in captivity, knocked on the door out of curiosity, only to discover that the singer was a captive goldfinch. Presumably this bird had been exposed to wren song at some stage and had picked it up."*

During that time, the study in bird song was limited in depth mostly due to the lack of analytical equipment [Catchpole and Slater, 2008, p. 3]. Today, modern recording, storage, and—most importantly—visualization technology allows researchers to gain fine-grained, highly detailed insights into auditory bird communication.

The inception of the sound spectrograph that was developed during World War II—and its availability for researchers in the 1950's—is considered the birth hour of modern science of bird song according to [Marler, 2004, pp. 2-10]. Moreover, the transformation of sound into images has been widely adapted in the field of bird sound research since the publication of William Thorpe's study on song learning in chaffinches [Thorpe, 1954] that almost exclusively relied on visualizations of songs made using a sound spectrograph. The images created using this (or a similar) technology are commonly referred to as sonograms or spectrograms.

I will also include some visualizations of bird songs and calls throughout this thesis to illustrate details and differences. Computer technology allows us to generate different visual abstractions of audio signals; the details of this process will be elaborated in Section 2.4. For reasons of consistency, all visualizations that represent sound in time and frequency will be called spectrograms further on.

The visualization of bird sounds is well established and widely used. It is considered one of the most convenient modalities when analyzing bird vocalizations [Kroodsma, 2005, p. 2]. However, the transcription of bird songs remains ambiguous and often non-intuitive—despite the fact that sounds are commonly used to identify birds. According to Nathan Pieplow, expert birders tend to detect and identify ten times more species by ear than by sight [Pieplow, 2017, p. 1]. Nonetheless, most field guides focus on visual features like size, shape and plumage colors to help citizen scientists to identify birds. On top of that, those field guides use a wide variety of vocabulary to transcribe bird sounds and often do not contain the verbalization of all song variations.

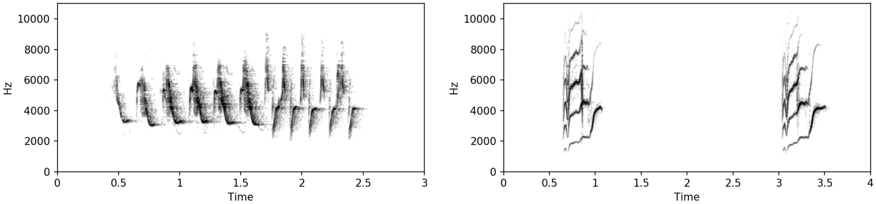
The following transcriptions of song and calls of the American Goldfinch (*Spinus tristis*) were taken from some of the most popular field guides for North America:

Peterson Field Guide to Birds [Peterson, 2010, p. 332]: Song clear, light, canary-like. In undulating flight, each dip is punctuated by *ti-DEE-di-di* or *per-chik-o-ree* or *po-ta-to-chip*.

Sibley Birds East [Sibley, 2016, p. 420]: Song high, musical, rapidly repeated phrases *toWEE toWEE toWEE tweer tweer tweer ti ti ti ti*; may suggest buntings but less stereotyped; fading at end. Call thin, wiry *toweeeowee* or *tweeee*; also a soft *tihoo* and variations. Flight call a soft, whistled, descending series of *ti di di di*.

National Geographic Field Guide to Birds [Dunn and Alderfer, 2017, p. 442]: Song is a lively series of trills, twitters, and *swee* notes. Distinctive flight call, *per-chik-o-ree*.

Learning to identify birds by their songs and calls according to those field guides can be quite challenging. It is clearly a hard task to memorize all those descriptions without a reference or years of experience. Furthermore, listening to a bird sound in the field and then trying to find the corresponding species according to those descriptions seems to be impossible. Figure 2.1 illustrates song and calls of the American Goldfinch as spectrogram for comparison with the aforementioned transcriptions.



(a) Song (🔊 1) (b) Calls (🔊 2)

Figure 2.1.: Vocalizations of the American Goldfinch (*Spinus tristis*). According to Sibley Birds East, the song depicted on the left can best be transcribed as *toWEE toWEE toWEE toWEE toWEE toWEE ti ti ti ti ti*, whereas both calls on the right can be verbalized as *toweeeowee toweeeowee.*

We can see that the actual signal is richer than any of the transcriptions ever can be. This is not an entirely new discovery, in fact, as Peter Marler reports, skilled musicians have tried to transcribe the notes of bird songs using musical characters for quite some time [Marler, 2004, p. 3]. Most notably, French bird watcher and composer Olivier Messiaen (1908–1992) is well-known for his transcriptions of bird songs and stylized songs from the Wood Thrush or the Baltimore Oriole [Fallon, 2007]. The vocal abilities of birds however are far greater than any musical instrument can accomplish, forcing musicians to adapt the transcription in pitch and tempo rendering them unrecognizable to ornithologists.

In his impressive *Field Guide to Bird Sounds of Eastern North America*, Nathan Pieplow presents a novel approach to break down bird sounds into basic elements and then using those elements to describe bird songs and calls in a uniform way [Pieplow, 2017]. The resulting visual index—the core reference feature of this book, intended to help birders identify unfamiliar bird species by ear—covers 83 densely written pages for Eastern North American species alone. Even the attempt to combine spectrograms and recordings of birds like in *'The Sound Approach to birding'* [Constantine, 2006] requires intense learning sessions and is often not practical to casually learn birding by ear.

In conclusion, identifying birds by sound is a convenient way as the vocal-auditory system of birds is highly developed and often used for communication. Additionally, spectrograms can be used to analyze bird vocalizations. On the other hand, teaching someone to identify birds by sound is challenging. Computer assisted tools can help to ease the process of identification for researchers, birders, and citizen scientists. However, in order to decide what such a system needs to look like, we have to investigate more aspects of bird vocalizations.

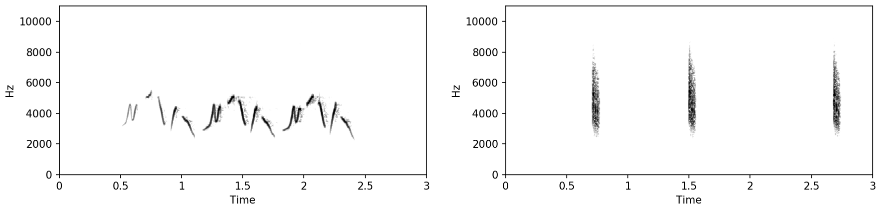
2.2.2. Production and perception of sounds in birds

Birds are true masters when it comes to singing. The astonishing number of notes as well as the range in pitch and tempo are unmatched. Especially oscine passerine birds (or true songbirds) are known for their rich repertoires of songs and calls—some of them are life-long learners or even mimics. Thanks to their specialized vocal tract, songbirds are able to emit sounds with high precision and complexity. However, bird sounds do not just include songs. I will explore the avian vocal tract, different types of bird sounds—there is a number of distinct non-vocal sounds—and avian physiology in this section.

Vocal sounds

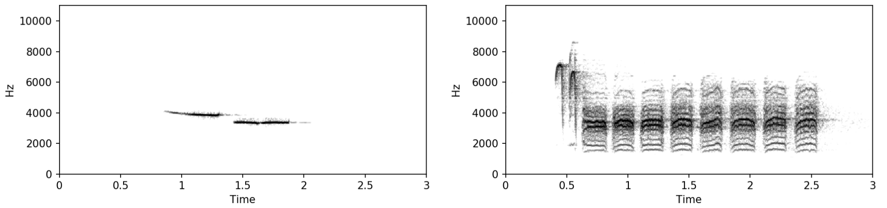
Vocal sounds are not limited to songbirds. Almost every bird species emits sound to defend a territory, attract mates, warn about predators, or mock and mimic. Based on complexity, length, or function, vocal sounds of birds can be divided into songs and calls. Although the distinction between song and call is not always clear-cut, a given vocalization can usually be classified as either of both. Generally, a call is considered less complex but richer in function, whereas a song oftentimes is louder and more complex but mainly used to attract mates or compete with members of the same sex. [Byers and Kroodsma, 2016, pp. 360-365] The differences between songs and calls for two species are illustrated in Figure 2.2: The Common Yellowthroat (*Geothlypis trichas*) with a complex song and simple calls, as well as the Black-capped Chickadee (*Poecile atricapillus*) with a rather simple song and more complex calls.

The syrinx is the sound producing organ in birds and the equivalent of the human voice box or larynx. Catchpole and Slater provide a detailed overview of the most important biological aspects of bird vocalizations and introduce the research that was done to unravel avian sound production [Catchpole and Slater, 2008, pp. 20-28]. The complexity of the syrinx varies in different bird species and is greatest in



(a) Common Yellowthroat song (🔊 3)

(b) Common Yellowthroat calls (🔊 4)



(c) Black-capped Chickadee song (🔊 5)

(d) Black-capped Chickadee call (🔊 6)

Figure 2.2.: Songs and calls of two North American bird species—Common Yellowthroat and Black-capped Chickadee.

songbirds. In contrast to the human larynx, the syrinx consists of two chambers. Songbirds can vibrate paired tissues called labia (see Figure 2.3, Lateral and medial labium) in each of them [Byers and Kroodsma, 2016, p. 380]. This is an interesting fact, as it allows birds to produce two-voiced sounds (biphonation) at different pitches and thus very complex musical structures with a high tempo. Additionally, the syrinx of birds is positioned closer to the lungs at the junction of the two bronchi [Catchpole and Slater, 2008, p. 23]. The sounds produced with both chambers are then mixed when they pass the trachea. This fascinating effect can be observed in a number of birds, especially the Wood Thrush (*Hylocichla mustelina*) that creates one of the most complex sounds a bird can make with its impressive trills at the end of each song (🔊 7).

However, there are physical limitations to the complexity and tempo of bird vocalizations. John Brackenbury provides a detailed description of sound modulations that birds are capable of in Volume 1 of *'Acoustic Communication in Birds'*

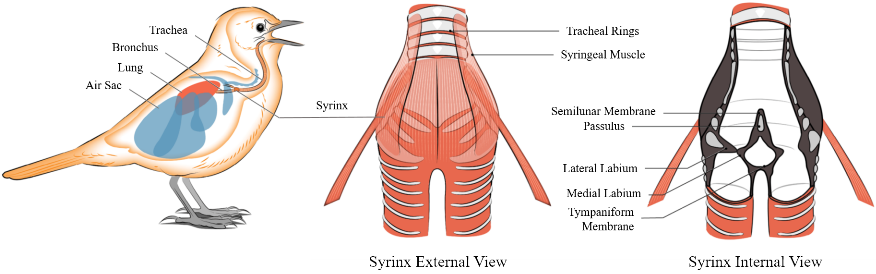
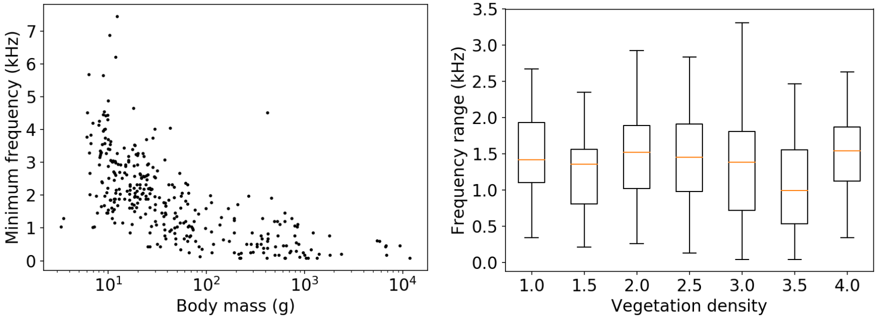


Figure 2.3.: Vocal tract of songbirds: External and internal view of the syrinx. Illustrations by Andrew Leach. Adopted from Dr. Rod Suthers, PhD, The Suthers Laboratory.

[Brackenbury, 1982, pp. 66-70]. Mainly, modulations of the produced sounds are achieved by stretching or retracting the neck, breathing patterns, and by changes in the width of the opened bill. It was also shown that a trade-off between bandwidth and repetition rate exists, meaning that complex utterances can only be repeated at a lower rate [Podos, 1997].

Birds vocalize in different frequency ranges. Again, a number of (physical) constraints determines the pitch at which a sound is produced and transmitted. According to Catchpole and Slater, several considerations have to be taken into account [Catchpole and Slater, 2008, pp. 86-92]. First, the size of the sound-producing mechanism required to emit low-frequency sounds is often limited by the size of the bird itself. Therefore, small birds (like most songbirds) are not capable of low-pitched sound production (see Figure 2.4a). Additionally, a number of environmental factors influence the transmission of audio signals and thus generally require a specific frequency range for ideal transmission. In open habitats, the most prominent environmental influences are temperature and humidity. Sound travels faster in warm air, high humidity enhances transmission, and atmospheric turbulence might scatter the signal. In forests, distortions might also occur when objects (like dense vegetation) scatter the path of transmission. The height at which birds vocalize above ground, the distance to the intended receiver, and the amount of background noise also play an important role and might alter the signal. [Wiley and Richards, 1982, pp. 147-151], [Halfwerk et al., 2011]



(a) Minimum frequency and body mass

(b) Frequency range and vegetation density

Figure 2.4.: The supplementary data provided by [Hu and Cardoso, 2009] shows a distinct correlation between body mass and the minimum frequency—only larger birds are able to vocalize in a lower frequency range. The correlation between vegetation density and the frequency range of bird vocalizations in the same study is not as distinct as the acoustic adaptation hypothesis might suggest. Only a slight decrease can be observed when comparing open (1.0-2.0), semi-closed (2.0-3.0), and closed (3.0-4.0) habitats.

The acoustic adaptation hypothesis proposes that the physical structure of bird sounds correlates with differences in habitat acoustics. According to this hypothesis, bird sounds with lower frequencies, narrower frequency ranges, and longer inter-element intervals should occur more frequently in densely vegetated habitats. [Morton, 1975] However, data of recent studies (see Figure 2.4b) suggests that habitat structure only weakly predicts the acoustical properties of bird songs [Hu and Cardoso, 2009]. This observation is backed by studies on mocking calls [Billings, 2018] or meta-analyses that identify other significant factors like energetic costs of bird vocalizations, the exposure to predators, or fitness of a population [Boncoraglio and Saino, 2007]. Additionally, the study of acoustic ecology suggests that birds—along mammals and insects—occupy ‘acoustic niches’ to reduce the effect of overlap between different species in frequency and time. It is assumed that the audio bio-spectrum of a habitat is kept intact by alternating vocalizations of different individuals. [Krause, 1993]

The acoustic adaption and niche hypotheses also apply to human altered environments. Some bird species like the Great Tit (*Parus major*) are known to vocalize with higher minimum frequencies in dense and noisy urban areas to avoid overlap with low-frequency noise [Slabbekoorn and Peet, 2003]. Independently of their suitability to predict the physical structure of bird vocalizations, both hypotheses indicate that frequency shifts in bird sounds occur depending on environmental factors. This is an important modality and should be accounted for when training an acoustic recognition system—either by dataset adaption (e.g. data augmentation) or shift invariant audiovisual features (see Chapter 3 for more details).

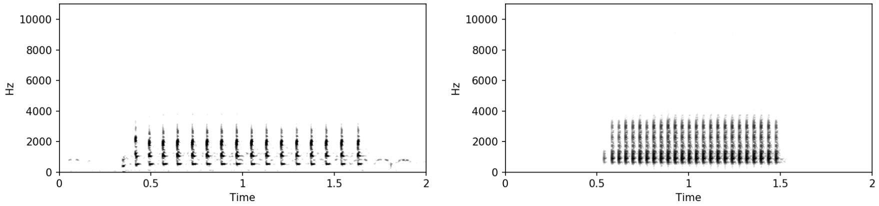
Non-vocal sounds

The vocal sounds produced by birds can be used to identify species or even individuals. Additionally, a variety of non-vocal sounds are also distinct for some species and help birders to decide which species they are hearing. Some of those sounds are even very common and the casual listener would probably recognize them instantly. However, not all non-vocal sounds are suitable for identification, e.g. splashing of water might reveal the presence of an aquatic bird, but the sound alone is most likely not sufficient to identify a species.

According to Nathan Pieplow's *'Field Guide to Bird Sounds of Eastern North America'* [Pieplow, 2017, pp. 4-5], we can discriminate five categories of characteristic non-vocal sounds:

Beating the bill against a hard surface: Common in woodpeckers that hit their bill against hard surfaces like trees or metal plates that yield loud and long-lasting sounds when struck repeatedly [Byers and Kroodsma, 2016]. Woodpecker drumming mainly occurs during the breeding season and probably functions as territory defense and courtship. Woodpecker species have characteristic drum patterns that vary in speed and are easily discernible in spectrograms (see Figure 2.5).

Snapping the bill shut: The White Stork (*Ciconia ciconia*) probably produces the most widely known sound of bill-clattering (🔊 10). The sound is created by quickly beating the mandibles together and can be heard over long distances [Cramp and Simmons, 1977, p. 334]. Other species like owls, flycatchers and gnatcatchers use bill snapping in close-range aggressive displays [Pieplow, 2017, p. 5].



(a) Downy Woodpecker (🔊 8)

(b) Hairy Woodpecker (🔊 9)

Figure 2.5.: Characteristic non-vocal drum patterns of two woodpecker species. Species-specific drum rates can be used for identification.

Clapping the wings together: This kind of sound production generally happens during flight when species like Long-eared and Short-eared Owls or pigeons clap their wings together above or below their body [Pieplow, 2017, p. 5].

Moving feathers through air: When in flight, the feathers of most birds produce sound when moving through the air. However, some species are specialized in sound production using their feathers [Pieplow, 2017, p. 5]. One of the most prominent examples are hummingbirds that emit characteristic buzzing sounds during flight. Additionally, male hummingbirds are known to use their wings and tails to produce a diversity of sonations when diving towards a perched female [Clark and Feo, 2008], e.g. Anna's Hummingbird (*Calypte anna*) produces a mechanical 'chirp' when diving that precedes the normal song (🔊 11).

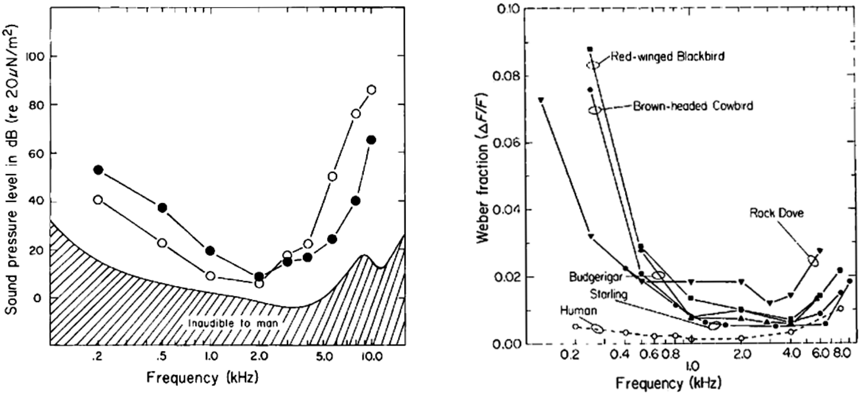
Inflating body cavities with air: Despite the fact that all birds have air sacs as part of their respiratory system, only a few species have modified air sacs that are used in sound production. An inflated air sac can serve as resonating chamber or can be used to explosively release air, which results in popping sounds. [Pieplow, 2017, p. 5] The American Bittern (*Botaurus lentiginosus*) is known to produce bizarre sounds inflating the esophagus (🔊 12).

Avian auditory physiology

Although all vertebrates share the same structural organization of the main brain regions, the capabilities of bird brains can be divided into two groups. Only parrots, hummingbirds, and songbirds have the necessary forebrain anatomy develop learned vocalization. Other bird species lack song learning capabilities and only use basal brain structures to produce innate and genetically inherited vocalizations. [Jarvis, 2004, p. 226] Neurobiologists discovered two main vocal pathways in birdbrains; one that controls the production of songs (motor pathway) and another that is more involved in the mechanics of song learning (learning pathway) [Catchpole and Slater, 2008, pp. 36-37]. Both pathways are part of a highly technical field of research that rapidly advances. However, I will solely focus on the perception and auditory capabilities of birds with respect to the three most important dimensions: Frequency discrimination, intensity discrimination, and temporal discrimination [Dooling, 1982, pp. 102-110]. All three dimensions are interconnected with the brain structure and the main auditory pathway but are primarily important because of their significance for audio signal processing. Once we know how birds perceive vocalizations of other individuals, we can decide on optimal parameter settings for technical representations of bird sounds.

Frequency discrimination: Pitch is considered an important characteristic of many communication signals—not just in birds but also for humans. Frequency is an important cue for song recognition in many species [Dooling, 2004, p. 215]. The audiogram is the most basic measure of hearing (Figure 2.6a). On average, birds hear best between 1 and 5 kHz with limits of the auditory space available for vocal communication from about 500 Hz to 6 kHz [Dooling, 2004, pp. 207-209]. However, as discussed earlier, constraints in sound production and transmission limit the ability of birds to vocalize using the full auditory space. Considering this, the audiogram is of limited value to explore the relationship between auditory capability and natural behavior [Dooling, 1982, p. 102]. Frequency discrimination thresholds provide a more realistic measure of auditory capability. Birds are highly sensitive to frequency changes and—according to studies for five species—can discriminate a 1% change in frequency with the highest sensitivity between 1 and 4 kHz (Figure 2.6b). However, humans are significantly more sensitive to changes in a broader frequency range.

Intensity discrimination: Birds often vocalize over longer distances to communicate with the intended receiver. As the signal travels across a diverse landscape with various obstacles (such as dense vegetation), it is often altered before it reaches the recipient. The intensity of the broadcasted signal is probably impacted the most in long-range communication. Therefore, intensity can be considered an unreliable



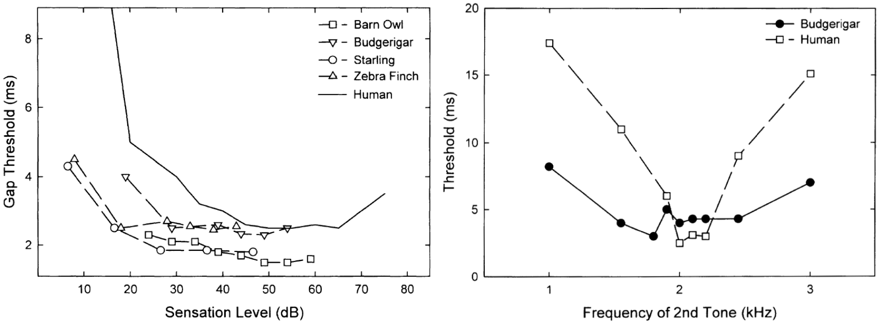
(a) Audiogram

(b) Frequency discrimination thresholds

Figure 2.6.: The audiogram (left) illustrates the auditory space for oscine (black) and non-oscine birds (white) [Dooling, 1982, p. 97]. Birds are most sensitive from 1-5 kHz. Frequency discrimination thresholds (right) are a more realistic measure of auditory capability. A Weber fraction of 0.01 to 0.02 translates to a 1% change in frequency. [Dooling, 1982, pp. 102-103] Birds are most sensitive between 1-4 kHz, humans are capable of detecting even smaller changes in frequency.

acoustic dimension to encode species or individual identity, motivational state, or even distance [Dooling et al., 2000, p. 330]. However, Richards and Wiley suggest that repetitive amplitude modulation (like in trills of bird songs) allows enough redundancy to minimize the effects of amplitude fluctuations and reverberations on long-range acoustic communication [Richards and Wiley, 1980]. Despite that, the difference in intensity between two successive sounds has to exceed 3 dB to be detectable by birds. [Dooling, 1982, p. 104] Again, humans are more sensitive in that domain and normally can detect 1 dB change in intensity [Dooling, 2004, p. 215]. Birds are on par with other vertebrates like cats, mice and rats.

Temporal discrimination: Birds show impressive hearing abilities compared to humans considering their inferior anatomy of the inner ear—but humans remain more versatile and perform better in frequency and intensity discrimination. It appears that there is only one specific domain where birds outperform the human



(a) Minimally detectable gap in noise (b) Minimally detectable gap using tonal stimuli of different frequency

Figure 2.7.: Temporal integration measures for humans and birds as shown in [Dooling et al., 2000, pp. 337-338]. The gap detection threshold in noise ranges from 2-4 ms for birds that are also less affected by low intensity sounds than humans (left). The difference is even more significant when two consecutive tonal markers differ in frequency (right). Both measures assume that the studied bird species are representative for the majority of birds.

auditory system: Temporal discrimination. This ability can mainly be measured in two ways, namely maximum temporal integration and minimum temporal integration. The relation between the detection of a sound and its duration is usually referred to as maximum temporal integration. It is a measure of the ability to sum acoustical energy over time. [Dooling et al., 2000, p. 335] In that domain, birds perform almost as well as humans. A sound should be at least 200 ms in duration to maximize audibility [Dooling, 1982, p. 105]. Birds can also discriminate changes in duration of about 10 to 20%, which is similar to what humans can achieve [Dooling, 1982, p. 196]. In terms of the minimum integration time that birds can achieve, we have to consider two modalities of gap detection measurements. First, if we look at two consecutive sounds (tonal markers) that are similar, gap detection abilities of birds are again on par with that of humans (and other mammals) and usually range from 2 to 4 ms [Dooling et al., 2000, p. 336]. However, if we consider consecutive sounds that differ in frequency, birds are significantly less affected in their ability to distinguish the gap between two tones [Dooling et al., 2000, p.

337]. Figure 2.7 illustrates the differences in the ability of birds and humans to discriminate between temporal changes in an acoustic signal.

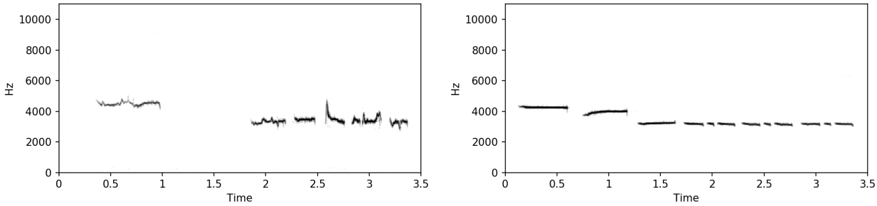
Available studies only focus on a few bird species and we have to assume that other species might not fall into that spectrum. The results however seem to be widely accepted and Robert J. Dooling can be considered one of the most prolific authors on that subject. Therefore, we can summarize that the auditory system of birds does not outperform human hearing abilities, except for temporal integration. Birds are most sensitive to changes in frequency between 1 and 4 kHz, can detect a 3 dB change in intensity and have superior temporal integration of 2 to 4 ms for consecutive tones of different frequency.

2.2.3. Vocal development and repertoires

Compared to all bird sounds and vocalizations, songs are especially prominent. Unlike most other birds, oscine passerines learn many components of their songs, which therefore can be very complex. The process of vocal development for a bird starts as nestling, continues after fledging, and is perfected during adulthood. In this section, I will explore the different dimensions of song learning according to Michael Beecher: When a song is learned, how many songs a bird can learn, if it needs tutoring, as well as copying fidelity and the degree of canalization [Beecher, 2008].

When songs are learned

Among all birds, songbirds are especially diverse with about 4,600 of the world's 10,000 species. Most songbirds have a sensitive period during which they learn their songs and perfect their repertoire. The length of that period can range from only a few weeks up to the entire lifespan in so-called open-ended learners. [Byers and Kroodsma, 2016, pp. 370-371] Experiments usually include the tutoring of captive birds to determine when they are sensitive to song learning. To do that, researchers often use playback to simulate an individual of the same species. One of the most comprehensive studies of song learning in birds (in that case Song Sparrows) was reported by Beecher in 2008 and suggests that learning phases last until the bird's first breeding season [Beecher, 2008]. However, sensible phases show significant variation across species and sometimes consist of more than one peak of high sensitivity [Hultsch and Todt, 2004]. The example in Figure 2.8 shows the



(a) Practice (plastic) song (🔊 13)

(b) Crystallized song (🔊 14)

Figure 2.8.: Comparison of two White-throated Sparrow songs (same individual). After song learning is completed, adult males sing a series of pure tones (b) whereas young birds struggle with the precise control of the syrinx and throat muscles (a).

plastic and crystallized song of a White-throated Sparrow (*Zonotrichia albicollis*) to illustrate how song learning changes complex vocalizations. It is important to account for plastic songs when monitoring a habitat, especially a few weeks into the breeding season. It will be interesting to see how well acoustic recognition systems can identify young birds that do not have a crystallized song.

How many songs a bird learns

Most bird species learn to sing in the early weeks of their life. Some species memorize and imitate large repertoires, while others simply learn to control their syrinx and throat muscles to sing their innate song. Although many bird songs are based on some common features that seem to be genetically encoded for a species, some birds only develop one song with not much variation across all individuals. Especially suboscine birds like the Eastern Phoebe (*Sayornis phoebe*) or flycatchers of the *Empidonax* genus develop their innate song even when captured, isolated or surgically deafened [Kroodsma, 1988, Kroodsma and Konishi, 1991]. This is particularly interesting, since acoustically isolated songbirds usually develop abnormal songs. Moreover, the songs of early-deafened birds tend to be even more abnormal [Marler and Sherman, 1983].

The number of songs that one individual can learn varies across species and ranges from less then ten to above 100 or even further for some species (see Table 2.1).

Table 2.1.: Estimated repertoire sizes for selected species as summarized in [Catchpole and Slater, 2008, p. 205]. The Brown Thrasher is one of the most versatile birds with long song sequences and a tremendously large repertoire (🔊 15).

Species	Repertoire size
Ovenbird (*Seiurus aurocapilla*)	1
White-crowned Sparrow (*Zonotrichia leucophrys*)	1
Common Chaffinch (*Fringilla coelebs*)	1-6
Great Tit (*Parus major*)	2-8
Hermit Thrush (*Catharus guttatus*)	6-12
Song Sparrow (*Melospiza melodia*)	7-11
European Starling (*Sturnus vulgaris*)	15-70
Marsh Wren (*Cistothorus palustris*)	33-162
Northern Mockingbird (*Mimus polyglottos*)	53-150
Common Nightingale (*Luscinia megarhynchos*)	160-231
Song Thrush (*Turdus philomelos*)	138-219
Brown Thrasher (*Toxostoma rufum*)	1500+

However, the Brown Thrasher (*Toxostoma rufum*) seems to be more versatile than any other species. Donald Kroodsma reports the study of one individual that sang more than 1800 different songs over the course of two hours [Kroodsma, 2005, p. 196]. Even profound mimics like the Northern Mockingbird (*Mimus polyglottos*) are known to have only up to 150 songs per individual. This is even more impressive considering that each individual develops its own repertoire. However, evidence suggests that not all songs are memorized. Some of them seem to be improvised on the spot, others are copies of another territorial male [Kroodsma, 2005, p. 199].

Some species, like the Blue Jay (*Cyanocitta cristata*) or American Crow (*Corvus brachyrhynchos*), are part of the *Passeriformes* order but do not develop stereotypical songs. Despite the lack of 'real' songs, those birds often have vast repertoires of calls and are highly variable [Kroodsma, 2005, pp. 179-191].

Repertoire size itself is hard to comprehend, because not all songs are distinct and may contain repeating elements in varying order. Additionally, repertoires can be extremely large and thus hard to quantify. Most of the time, they can only be estimated using an average repetition rate over the course of a long recording [Catchpole and Slater, 2008, pp. 204-208]. This approach is extremely time-consuming due to the manual comparison of spectrograms and therefore, only a few individuals get studied [Kroodsma, 2005, p. 192].

Vocal development in the other orders of birds has been less well studied. Parrots are known for their ability to imitate human speech, hummingbirds are known to develop abnormal songs in isolation, and species like doves or pigeons develop vocalizations that seem to be encoded in the genes without any imitation or tutoring. [Byers and Kroodsma, 2016, pp. 378-379]

Song tutoring

Imitation of adult bird song plays a significant role in the vocal development of many bird species. However, the implications for a computerized bird sound recognition system are limited. In contrast to the duration of song learning and the eventual size of the repertoire of an individual, the fact that birds learn from a tutor cannot be incorporated into a dataset easily. Additionally, even if a dataset would reflect the song tutoring within a population of birds, a resulting recognition system would still have to focus on common song features to identify individuals of the same species from another population.

Learning from a tutor can be considered a combination of social interaction between individuals and the sensitive phase of young learner [Byers and Kroodsma, 2016, pp. 372-376]. While some species (mostly oscine passerines) develop abnormal songs when isolated, other species do not require a tutor to develop crystallized songs. The parents of a young male—especially the father—can be considered as one of the most influential tutors of a young bird. However, dialects in bird song are common among the majority of songbirds, which indicates that the father might provide an important learning experience, but juvenile birds tend to acquire songs that are needed at their own breeding location [Kroodsma, 2004, p. 121]. It is believed that young birds hatch with a rough ‘template’ of their own species’ song and only memorize vocalizations that match that template, thus learning song almost exclusively from individuals of the same species [Catchpole and Slater, 2008, pp. 52-53].

Copying fidelity

Developmental programs of bird song include species that copy songs by precise imitation, species that vary in the importance of imitation, and species that almost exclusively rely on invention [Byers and Kroodsma, 2016, p. 373]. Again, isolated individuals of some species develop abnormal songs if their learning process depends on tutoring. For example, this was shown for Swamp Sparrows (*Melospiza georgiana*) and Song Sparrows (*Melospiza melodia*) that still showed distinct differences in their repertoires despite isolation but were not able to learn crystallized songs [Marler and Sherman, 1985]. Other species like the Gray Catbird (*Dumetella carolinensis*) are known to develop rich repertoires in isolation that feature songs that are indifferent from songs developed by tutored individuals [Kroodsma et al., 1997]. As discussed earlier, the Brown Thrasher is known to invent entire sequences of songs on the spot. It remains unclear whether tutoring is still required in order to invent vocalizations [Beecher, 2008].

Degree of canalization

The ability to mimic allows some species to incorporate astonishing amounts of heterospecific material into their repertoire of songs. Mimicry significantly adds to the diversity of a repertoire and might help to attract mates with an impressive succession of songs. However, the functional significance of mimicry remains uncertain [Catchpole and Slater, 2008, p. 75]. One of the most accomplished mimics is the Northern Mockingbird (*Mimus polyglottos*). Male mockingbirds sing well over 100 songs and a considerable number of them are copies of other species' vocalizations [Byers and Kroodsma, 2016, p. 399]. Mockingbirds often tend to vocalize in a long sequence of songs, permanently switching from one tonality to another (🔊 16).

Of all mimics, the Superb Lyrebird (*Menura novaehollandiae*) seems to be the most elaborate, with an uncanny ability to incorporate not just natural but also technical sounds into its song. Especially individuals in captivity tend to mimic car alarms, human speech, camera shutters or even toy sounds (🔊 17). Studies of European Starlings (*Sturnus vulgaris*) documented high-fidelity mimicry of tens of different sounds, including calls of owls, gulls and ducks as well as various environmental sounds like the 'meow' of cats or a squeaky door [Hausberger et al., 1991]. Other species known for their mimicry include the Lawrence's Thrush (*Turdus lawrencii*), the Marsh

Warbler (*Acrocephalus palustris*), and even Blue Jays (*Cyanocitta cristata*) that imitate calls of raptors like hawks. Although other species are known to mimic occasionally, it is believed that this occurs as maladaptive side effect of song learning and the males who learn the wrong songs usually are not able to attract mates [Byers and Kroodsma, 2016, p. 399].

2.2.4. Song variation in space and time

Most songbirds have a remarkable repertoire of songs and oscine passerines develop and perfect their repertoire during various developmental programs. We already know that song learning in birds occurs as part of the adaption to a new environment with new territorial neighbors and potential mates. Tutoring plays an important role in song learning and we can assume that imitation limits the incorporation of heterospecific material into the repertoire for the majority of species. That being said, we also have to assume that vocal bird sounds differ depending on population and habitat (variation in space). Additionally, variations in bird song occur from generation to generation and even over the course of a day (variation in time), when males alter their song sequences or phrases depending e.g. on the presence of other aggressive males, potential mates, or predators.

Variation in time

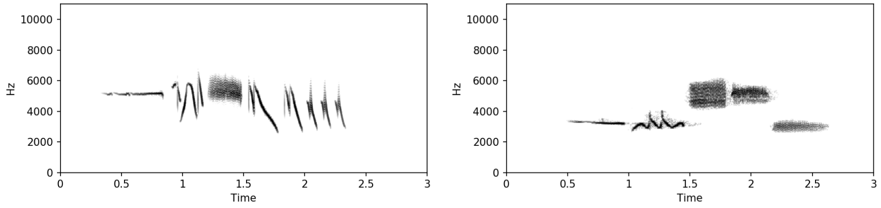
Birdwatchers are able to identify species by their songs, which is useful to not only distinguish closely related species with similar plumage but also when sight is limited. In avian systematics, song is of particular use at the species level and if songs can indicate species identity, we have to assume that song enables birds to recognize their own species and even individuals. [Catchpole and Slater, 2008, p. 149]. Experiments with Ovenbirds (*Seiurus aurocapilla*) showed that a male can distinguish familiar songs of a neighbor from (playback) sounds of an intruder [Weeden and Falls, 1959]. It is also known that (the extensively studied) White-throated Sparrows (*Zonotrichia albicollis*) respond more aggressively to a stranger's song depending on the playback location, which indicates that these sparrows recognize that certain songs belong to a particular individual [Falls and Brooks, 1975]. This is of importance for any automated bird sound recognition system as it adds to the diversity of a potential dataset and we have to account for individual variations of a species song (see Figure 2.9).

The invariant feature hypothesis suggests that those features of song that vary least and are relatively constant between individuals are most likely reliable for species recognition [Catchpole and Slater, 2008, p. 153]. Douglas Nelson found that alterations in frequency of the entire song are more important for species recognition since they vary less than other cues like number of phrases, trill-note duration, and inter-note interval. Frequency changes greater than two standard deviations relative to the mean resulted in significantly weaker responses during his experiments. [Nelson, 1988] Therefore, we can conclude that an automated recognition system needs to focus on semantically meaningful (high-level) and shift-invariant features of bird song for species identification.

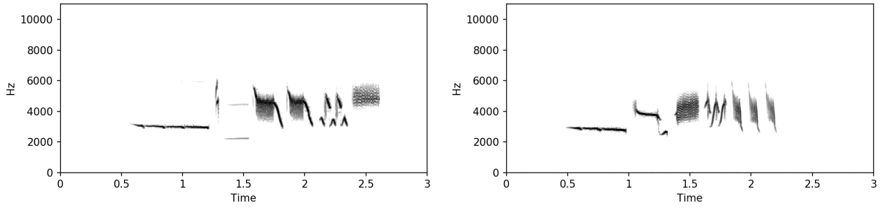
Variation in space

Song sharing within a population includes the sharing of entire repertoires or not a single element at all. Variations occur in patterns (also called dialects) that affect the distribution of song types and song elements among birds within an area. According to Catchpole and Slater, we can distinguish between micro- and macrogeographic variations that sometimes even occur within sharp boundaries. [Catchpole and Slater, 2008, pp. 242-254] Donald Kroodsma reports an episode in which he observed more than a thousand White-crowned Sparrows, but only six different songs along a 30-mile coastline in California that had very distinct boundaries between populations [Kroodsma, 2005, pp. 44-55]. Distinct macrogeographic variations in the U.S. most prominently occur in March Wrens (*Cistothorus palustris*), for which the songs of western individuals (that occur west of the Missouri River all the way to the Pacific) are very different from those of their eastern relatives (that occur east to the atlantic) [Byers and Kroodsma, 2016, p. 387]. Again, a sharp boundary of habitats that prevents western and eastern individuals from mating can be observed. This even leads to the suggestion that we might face different species because of significant geographic variations in song [Kroodsma, 2005, p. 134].

However, birds tend to construct their songs from a limited range of elements (see Figure 2.9) and variations between bird populations mostly occur through element permutations. Variations in element types are much less apparent. Yet, in some



(a) White-crowned Sparrow 1 (🔊 18) (b) White-crowned Sparrow 2 (🔊 19)



(c) White-crowned Sparrow 3 (🔊 20) (d) White-crowned Sparrow 4 (🔊 21)

Figure 2.9.: Songs of four individuals of the White-crowned Sparrow (*Zonotrichia leucophrys*). Males in the field usually restrict themselves to a single song type [Chilton and Lein, 1996]. Common elements like the long introductory whistle, the characteristic trills and buzzes occur in all four song types but vary in pitch. The inter-note and inter-phrase intervals appear to be similar among all four individuals. Recordings in this example where acquired in different locations across North America and variations in song patterns are most likely the result of regional dialects.

species, different populations do not share a single element that reoccurs in all individuals rendering dialect characteristics hard to define [Catchpole and Slater, 2008, p. 246].

Why do spatial variations occur? Different hypotheses have been formulated, including the matching habitat hypothesis [Hansen, 1979] that is somewhat similar to the acoustic adaption and niche hypotheses, the hypothesis of genetic adaption to form distinct populations and eventually new species [Nottebohm, 1972], the hypothesis of social adaption to territorial neighbors [Catchpole and Slater, 2008, pp. 260-261], or even the hypothesis that song variations occur as functionless byproduct of song learning [Andrew, 1962].

In any case, the fact that song pattern variations occur based on location but mostly affect the permutation of song elements, is of great significance for automated species identification systems. Classification algorithms cannot rely on song features (or even acoustic fingerprints) of individuals but instead need to incorporate re-occurring patterns across all individuals of one species. It remains to be seen how intra-species song diversity affects the need for extensive training data.

2.2.5. Function of bird song and singing behavior

Most investigations of bird vocalizations have focused on the songs of songbirds. Although females are known to sing in many species, male bird song appears to be more prominent throughout the year. [Byers and Kroodsma, 2016, p. 392] Changes in vocal production based on season and daytime play a significant role in automated bird species recognition. I will explore the most prominent aspects of variations in song production in this section.

Seasonal variations and breeding cycle

The ‘dual function’-theory suggests that male bird song serves two primary functions: Defense of a territory and mate attraction [Catchpole and Slater, 2008, p. 114]. Empirical evidence correlates seasonal song production and the breeding cycle of birds. Different studies support this assumption and show that seasonal song production spikes with the start of breeding activity and then sharply declines over the next few weeks [Catchpole, 1973]. Experiments on mate removal also demonstrated this strong correlation. Males tend to increase their song production significantly when female mates are removed and return to a normal level when the mates return. [Krebs, 1981], [Otter and Ratcliffe, 1993] Song always occurs throughout the year but reaches its peak in spring when resident birds and migrants occupy and defend their territories [Catchpole and Slater, 2008, p. 114]. This annual cycle is of great importance for automated bird sound classification as the arrival of migrants in their breeding grounds poses a hard challenge due to the high levels of song production. Additionally, since song production decreases after the breeding season, and some migrants might even be detectable for only a few weeks throughout the year, the recognition system has to adapt to changes in species and song diversity. We will see in Section 2.3 how metadata can help to predict species abundance based on time and location.

Daily variations and dawn chorus

Song production does not only vary over the course of a year but also during the day. Most day-active birds are silent throughout the night and start the day with a burst of songs roughly one hour before sunrise [Byers and Kroodsma, 2016, p. 396]. More and more species join this so-called ‘dawn chorus’ in a rather predictable sequence. It is believed that the ability to see strongly correlates with the beginning of song production at sunrise. Birds with larger eyes and thus better ability to see in dim light tend to start song production earlier. [Thomas et al., 2002] This even applies for tropical birds that start singing based on height above ground—sunlight first penetrates the forest canopy and later reaches the ground [Berg et al., 2006]. The dawn chorus can be considered the most important time of the day for species detection based on song. However, it is also the most challenging: Song overlap during this cacophony poses one of the most difficult challenges in signal processing. Additionally, birds tend to sing faster songs with shorter intervals of silence during the dawn chorus compared to the rest of the day. Some species even use entirely different repertoires at dawn. [Byers and Kroodsma, 2016, p. 397] The reasons for this behavior are manifold and range from social interaction with neighbors to the inability to hunt due to the lack of sunlight and thus the tendency to instead use the first light of the day to sing. The diurnal rhythm of song and the relationship with other behaviors is not yet fully understood. [Catchpole and Slater, 2008, p. 130]

Considering the variations of song production over time, it remains questionable whether birds avoid overlap with other individual’s vocalizations. We already know that birds use counter-singing to communicate with territorial neighbors (see Section 2.2.4) and occupy acoustic niches (see Section 2.2.2); on top of that, birds do in fact avoid competition when singing. Several studies have shown that birds adjust their song output in relation to other species and other individuals by changing the rhythm of their vocalizations in terms of song length and patterns of silence between songs in asynchronous cycles. [Catchpole and Slater, 2008, pp. 136-138]

Another noteworthy dimension of bird sounds are flight songs and flight calls. Vocalizing during flight can be considered physically demanding. However, many shore-

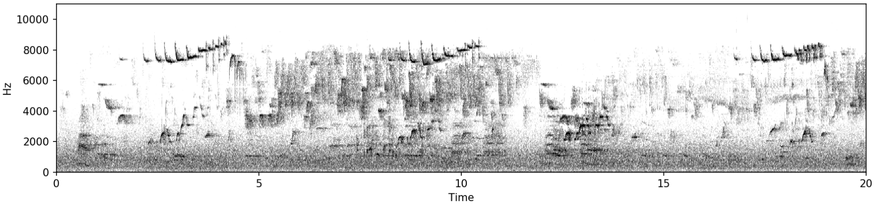


Figure 2.10.: Visualization of dawn chorus vocalizations recorded in June in Alaska, United States (🔊 22). The high levels of song production and the significant number of overlapping vocalizations in frequency and time pose a hard challenge for any bird sound recognition system.

birds produce long and complex songs in flight, European Sky Larks even tend to rise into the sky to sing long whistle notes before descending back into their territory. [Byers and Kroodsma, 2016, p. 400] Monitoring of nocturnal bird migration is almost always limited to non-visual observation and thus often relies on flight calls—short, often high-pitched species-specific vocalizations given during sustained flight [Farnsworth, 2005]. The automated detection of flight calls for species identification during the night can be seen as a distinct—almost independent—area of research within the complex of automated bird sound recognition because of the relatively rare nature of those acoustic events [Lostanlen et al., 2018].

2.3. Avian ecology

Seasonal changes in bird species abundance, diversity, and composition in a local habitat are strongly linked to avian migratory patterns and thus the result of adaptations to variations in the environment. Migratory bird species take two-way trips between wintering and breeding sites annually (migration). Even non-migratory birds depart from their hatching grounds to find new breeding locations (dispersal). [Winkler et al., 2016, pp. 453-454] It is vital for any avian monitoring system to account for these two types of movements.

Selecting representative vertebrates as indicator species to monitoring environmental changes in habitats across all lifeforms is a common technique in conservation biology. Birds are ideal indicators as they have been shown to respond to various environmental changes over many spatial scales. They usually reveal their presence, and their abundance is influenced by nature and configuration of surrounding habitats. [Carignan and Villard, 2002] Additionally, geographic distribution, local abundance, and habitat specialization all influence the vulnerability to extincting, particularly in highly specialized species with small populations and narrow range [Fitzpatrick and Rodewald, 2016, p. 590].

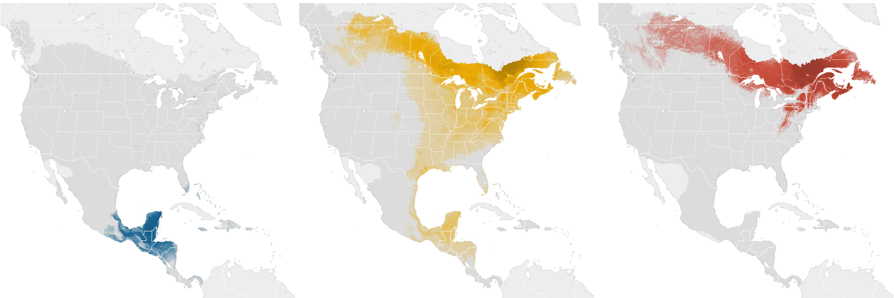
I will explore some of the most critical aspects of bird migration and dispersal in this section. I will also shed some light on recent conservation efforts and citizen science projects in support of conservation biology.

2.3.1. Habitats, abundance, and migration

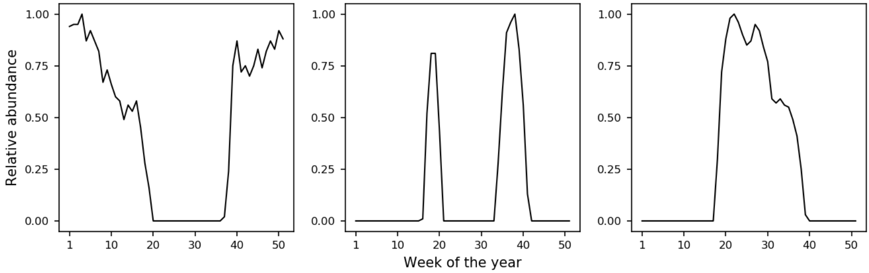
One of the most exhaustive resource to study avian movements is ‘eBird Status and Trends’ with its maps, charts, and animations. Available online, it provides unprecedented depth of information for 107 North American species in four key areas: Abundance, population trends, habitat association, and range. [Fink et al., 2018] Based on observations of citizen scientists and predictions derived from those, the data collection lively illustrates the macrogeographic scale of bird migration (see Figure 2.11).

The migratory range of birds and seasonal variations in abundance provide us the clues of when and where to look for certain species. For example, monitoring efforts in Kentucky, USA, have to account for relatively short peaks in species diversity when migrating birds pass through. Apparently, avian movements, time of the year, and location are strongly linked with the diversity of avian vocalizations. We can expect migratory (and also dispersal) patterns to significantly influence vocal production in a given region. On top of that, knowing which species are to expect based on location and time, helps to identify pivotal environments to protect from some of the most critical threats to bird populations: Habitat loss, habitat fragmentation, introduced predators, pollution, introduced diseases, and other human-induced stressors [Fitzpatrick and Rodewald, 2016, pp. 593-603].

Habitat attraction and avoidance also play a significant role in species composition in avian communities. Competing bird species are partitioning limited resources such as food and territorial niches for breeding [MacArthur, 1958], [Koenig, 2016,



(a) Non-breeding (b) Pre-breeding (c) Breeding



(d) Normalized relative abundance in Guatemala, Kentucky (USA), and Quebec (CA)

Figure 2.11.: Seasonal variations in abundance of the Magnolia Warbler (*Setophaga magnolia*). Migration starts in May, when the birds leave their winter habitats in Central America, ranges across the Eastern United States and ends in June, when the breeding grounds in Canada are reached. Starting in September, the warblers migrate again, and reach the winter habitats in late October. Maps and data provided by [Fink et al., 2018].

pp. 518-522]. Vegetation structure can change the diversity of bird communities and the number of species a certain habitat can contain is limited based on its configuration [Vickery et al., 1995]. Yet, differences on species diversity and composition occur on a rather microgeographic scale, rendering preferences on habitat structure a hard to comprehend dimension of bird migration. For example, the Magnolia Warbler preferably occupies evergreen broadleaf forest in its Central American winter habitats. When migrating north, it can be found in deciduous broadleaf forest, woody savannas, and urban environments. On its breeding grounds, it prefers mixed forest. [Fink et al., 2018] On a local scale, this list is not exhaustive and we have to assume that individuals also occur in different habitats, although they usually avoid their characteristics (e.g. large water bodies or closed shrublands). In order to orchestrate conservation efforts, the preservation of preferred habitats is key to protect endangered species and to maintain their reproductive success [Fitzpatrick and Rodewald, 2016, p. 583].

For now, macrogeographic migration patterns present us a more reliable picture: The chance of encountering a Magnolia Warbler in the western states of the U.S. year-round or during the winter month in Canada is slim, and the conclusion we can draw from that for an automated recognition system is of greater significance. Predicting species abundance, diversity and composition based on time and (macrogeographic) location will greatly impact the reliability of the proposed recognition system of this thesis (see Chapters 5 and 6).

2.3.2. Conservation biology and citizen science

Monitoring populations is one of the most important approaches to assess ecosystems in terms of conservation priority—especially in regions with high overall biological diversity. Monitoring indicator species, such as birds, can help to identify early warning signs that indicate habitat changes that are likely to affect many other species [Fitzpatrick and Rodewald, 2016, pp. 607-608]. Commonly, two main methods—survey and census—are used to assess the species composition of an area. Since census would require researchers to count every individual and thus limits investigations to only a few species, surveys are easier to conduct and provide a sufficient assessment for most tasks. However, surveys might also be incomplete due to the question of when, where, and for how long to count individuals or species. [Bibby et al., 2004, pp. 1-3] Of all the available counting techniques, I will focus on point counts to estimate relative abundance and population trends during a survey. They are of particular interest for automation efforts and widely used in conservation biology.

Point counts

When bird watchers or researchers decide to observe an area and assess bird species diversity by counting, they often encounter difficulties. The trade-off between time available and the number of points sampled concerns the question of how to maximize the probability of detections. The need for standardized methods of bird counts for different geographic regions, habitats, seasons, and levels of abundance is apparent. [Lynch, 1995] In an effort to formulate those standards, Ralph et al. organized a workshop during which biologists suggested important dimensions of bird count methodology [Ralph et al., 1995]. Some of those suggestions are of particular interest because they serve as strong arguments for an automation of the counting process:

Setup of monitoring stations: According to Ralph et al., point count stations should be systematically located and placed to avoid boundaries between habitat types with a minimum distance between stations of 250 meters (820 feet). An increased number of independent sampling stations is considered more reliable than repeatedly counting at a smaller number of stations.

Count period at each station: The amount of time spent at a monitoring station is a compromise between the acquisition of accurate data and the effort of sampling a larger number of stations. The proposed observation time is 5 minutes when the time of travel between monitoring sites is less than 15 minutes and 10 minutes when the time for travel is greater than 15 minutes.

Time periods and weather conditions: The detectability of bird species varies depending on season and time of day. The suggestion is to conduct point counts when the detection rate of the species being studied is most stable. Additionally, Ralph et al. note that birds should not be surveyed when it is raining, during heavy fog, or windy conditions.

Observer training: Differences between observer skills have a significant impact on the success of point counts. The ability to identify birds by sight and sound usually takes several years to develop. Ralph et al. state that the training of non-experts is almost certainly too time-consuming to be feasible when all species that occur at a location should be counted.

Involving the public to conduct point counts has certain advantages. First, the number of monitoring stations increases significantly. Secondly, the covered time span expands to year-round observations. Thirdly, local bird watchers are supposedly skilled enough to conduct reliable point counts without further training. Two of the most prominent campaigns to involve citizen scientists for bird counts are eBird checklist observations [Sullivan et al., 2009] and the annual 'Stunde der Gartenvögel' of the Naturschutzbund Germany (NABU) [1]. In both cases, bird watchers are asked to provide bird counts at their preferred location over the course of 5-60 minutes. Still, observations are limited in time and thus might not provide a complete assessment. Additionally, monitoring a specific habitat or cryptic and endangered species still requires systematic observations by experts.

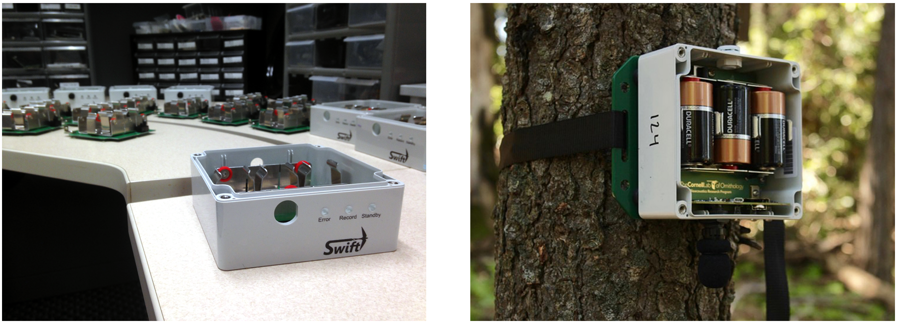
Acoustic monitoring

Point counts require bird watchers to identify species either by sight or sound. The overall habitat structure plays a significant role in point counts and may limit the possibility to see individuals that occur in an area. In fact, birds are easier to find and to detect in open habitats—even if they are silent. Yet, most detections in closed forest can only be made by ear. [Bibby et al., 2004, p. 3] Considering this, acoustic monitoring using autonomous recording units (ARU) allows researchers to conduct point counts in almost any densely vegetated habitat. A recent survey study—published by Shonfield & Bayne in 2017—notes that ARU have become a widely used sampling tool in ecological research and monitoring over the past decade [Shonfield and Bayne, 2017]. Advances in hard- and software allows manufacturers to produce weatherproof recording units at low costs. One prominent example of a widely used ARU is the SWIFT recording unit provided by the Cornell Lab of Ornithology (Figure 2.12). This particular unit is currently being used to gather continuous audio recordings from various locations around Ithaca, New York in the United States—the experimental foundation of this thesis (see Chapter 5).

According to Shonfield & Bayne, using ARU to conduct point counts has certain advantages, including repeated sampling across spatial and temporal scales, reduced observer bias due to peer review, reduced field time as result of continuous observation, and a permanent record of the survey for further analysis and storage [Shonfield and Bayne, 2017]. Still, those advantages come at the cost of increased workload due to manual labeling of countless hours of recordings. The difficulty of processing large amounts of data is one of the main reasons why human-conducted

---

[1] https://www.nabu.de/tiere-und-pflanzen/aktionen-und-projekte/stunde-der-gartenvoegel/



(a) SWIFT recorder assembly line (b) SWIFT recorder in the field

Figure 2.12.: Autonomous recording units are a widely used sampling tool in ecological research. The SWIFT recorder provided by the Bioacoustics Research Program (BRP) of the Cornell Lab of Ornithology[2] allows to record up to 30 consecutive days of audio. Optimizing the assembly of these weatherproof ARU reduces the costs to \$250 per unit. SWIFT recorders will provide the majority of audio recordings used for experiments and evaluation throughout this thesis. Images provided by the BRP.

point counts still are the superior tool of habitat assessment. It is the primary goal of this thesis to provide conservation biologists, ornithologists, and citizen scientists with tools that help to increase the effectiveness of autonomous recording units for avian activity monitoring.

Public data acquisition

Identifying birds by ear often requires expert knowledge and years of experience (see Section 2.2.1). The training of an automated recognition system usually requires large amounts of data—audio recordings of bird vocalizations. Again, involving the public can help to acquire recordings of bird species across the globe. Community projects like xeno-canto.org[3] or eBird[4] and professional collections like the

---

[3] https://www.xeno-canto.org

[4] https://ebird.org

outstanding Macaulay Library[5] provide a vast amount of open source data on bird vocalizations, occurrences, and other metadata.

Xeno-canto: Founded in 2005 by Bob Planqué and Willem-Pier Vellinga, xeno-canto.org evolved into one of the largest collections of sounds of wild birds from all across the world. The website aims to popularize bird sounds, to improve their accessibility, as well as to increase the knowledge of bird sounds in general. Xeno-canto is open for public contribution and recordings are shared using various Creative Commons licenses. The collection itself features more than 400,000 recordings of over 10,000 species totaling for more than 7,000 hours of audio data. [Xeno-canto, 2019]

eBird: The Cornell Lab of Ornithology and the National Audubon Society launched eBird in 2002 to engage a vast network of citizen-scientists to report bird observations using standardized protocols [Sullivan et al., 2009]. Since then, eBird grew into the world's largest biodiversity-related citizen science project. Bird watchers around the world contribute over 100 million bird sightings each year. When submitting an observation, birders have to answer a number of questions and sightings are cross-checked using a list of likely species based on date and region. eBird supports the scientific community by opening its collection to researchers through tools, applications, and programming interfaces. [eBird, 2019]

Macaulay Library: In 1929, Arthur Allen and Paul Kellogg made the very first recordings of wild birds in Ithaca, NY, United States. This marks the birth hour of the Cornell Library of Natural Sounds. Due to rapidly advancing recording technologies, the collection evolved into the largest scientific archive of natural history audio, video, and photographs over the course of the following decades. The early 2000's mark the inception of the digital era that was followed by a rapid phase of expansion. To honor their contribution to the Library of Natural Sounds, the collection was named the Linda and William Macaulay Library of Natural Sounds. As of today, the collection consists of more than ten million photos, over 400,000 audio recordings, and almost 60,000 videos. [Macaulay, 2019]

The importance of the aforementioned libraries and archives can not be overestimated. Public involvement in the collection of media and metadata is vital to advance avian research. Fortunately, the birding community is highly active and dedicated to record, share, and analyze vast amounts of observations. Other initiatives like the *British Library Sound Archive*, the *Tierstimmenarchiv Berlin* or the

---

[5] https://www.macaulaylibrary.org

Australian National Wildlife Collection provide access to high quality collections of natural sounds. Thanks to those efforts, the acquisition of training data for an automated bird sound recognition system will most likely result in a large and heterogeneous dataset of high overall quality.

2.4. Sound digitization and representation

Since the late 1920's, when Allen and Kellogg recorded their first birds, recording equipment has evolved into sophisticated tools of sound capturing. The new millennium brought advanced digital devices that replaced common tape recorders. Today, computer technology allows us to capture, process, and analyze digitized sounds of birds with ease. Since the arrival of the smartphone, almost every adult owns a compact recording device—the technological variety of sound recorders is ever expanding. This variety has implications for automated sound processing systems. Professional recording gear used by birders and scientists differs greatly from the microphones and recorders used in ARU or smartphones. In this section, I will explore those differences, elaborate on spectrogram generation and the adaption of digitized sound representations to the avian vocal and auditory range.

2.4.1. Digital sound recording

The study of bird song is an important tool for every ornithologist and capturing bird sounds remains one of the most important tasks in field research (see Section 2.2.1). Birders and scientists go through great lengths to capture birds in the wild. (Semi-) professional recordings provided by the Macaulay Library or Xeno-canto are the results of those efforts. As stated earlier, the vast archive of both collections can be used to train an automated bird sound recognition system. However, the application of such a system will very likely be affected by a domain shift from unidirectional recordings as training data towards omnidirectional recordings as test data. Considering this, it is important to account for the differences of both domains.

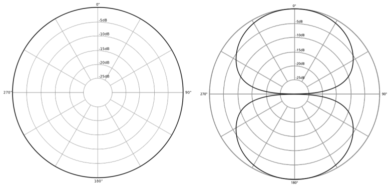
Unidirectional recordings

The reasons to record bird vocalizations are manifold. Typically, a permanent record of those sounds is useful to analyze song structures, the number of songs a male sings, to study behavioral patterns and the interaction with territorial neighbors or even to conduct playback experiments. Additionally, recordings of bird sounds can be used to teach birders how to identify birds by sound or to preserve a record of endangered species. Whatever the reason, recording equipment should satisfy two main criteria: It should provide high quality recordings for further lab analysis and should be light enough to carry out in the field. [Catchpole and Slater, 2008, p. 12]

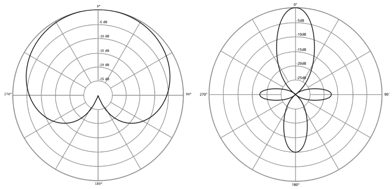
Mainly, two types of microphones are used for this task: Highly directional 'shotgun' microphones and recording systems that include a parabolic reflector (sometimes called 'dish') [Kroodsma, 2005, p. 404]. Humans and birds mostly share the same vocal and auditory range, thus commercially available recording systems often fulfill the requirement of decent quality. The type of microphone that is suitable for the task of bird sound recording is often determined by the polar response pattern (see Figure 2.13). When the source of the sound is close enough to the mic (like it is in most scenarios where human speech is recorded in a studio), a cardioid polar pattern is most likely the ideal choice. Since birds often perch in the canopy or flee when humans approach, shotgun patterns are more effective as they allow recordists to point the microphone at the distant sound source. However, parabolic reflectors in combination with omnidirectional microphones are far more effective at capturing distant or soft sound. The parabola reflects incoming sounds towards the center, where the mic is mounted, favouring higher frequencies. Parabolas become ineffective below frequencies at which the wavelength of a signal is equal to the diameter of the reflector and thus might alter the recorded vocalization to a notable extend. [Kroodsma, 2005, pp. 405-406] Despite that, parabolic reflectors are widely used to record bird sounds since birds tend to vocalize using higher frequencies than humans and parabolas also shield the microphone from background noise.

Omnidirectional recordings

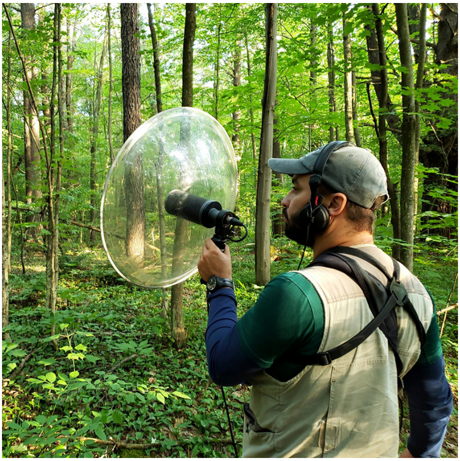
The characteristics of a microphone—like polar response pattern, sensitivity, or noise floor—have a significant impact on the overall quality of a recording. Yet, the choice of microphone for a bird sound recording task is often limited due to



(a) Omnidirectional (b) Bidirectional



(c) Cardioid (d) Shotgun



(e) Professional recording gear

Figure 2.13.: Different microphone polar patterns (a-d). Cardioid patterns are most commonly used in the vast majority of studio recording scenarios. Shotgun microphones can be used to capture distant sounds by pointing at the source. Parabolic dishes in combination with omnidirectional microphones capture high frequencies over long distances and thus are ideal for bird sound recordings. Professional recording gear often includes high quality headphones and digital recorders. Authors: Galak76 (a, b, d), Nicoguaro (c), Jay McGowan (e)

certain constraints—two of which are important to consider for long-term acoustic monitoring: The recording scenario and the available space in a compact recording system. One of the goals of this thesis is to develop an automated bird sound recognition system for mobile devices such as ARU (point counts) or smartphones (single species identification). Directional microphones are not a good option for acoustic monitoring since they have to be pointed at a sound source by hand. Omnidirectional microphones are a better choice as they will likely capture a bird independently from its position. Acoustic recorders—like the SWIFT recorder—mostly use small omnidirectional microphones pointed downwards to capture the surrounding soundscape. Equally capturing sounds from all directions comes at the costs of less detailed recordings of distant sounds and high ambient noise levels.

Both impairments considerably impact the automated recognition and the shift in recording domains has to be accounted for during data augmentation.

Today, smartphones are a convenient tool to capture bird sounds. Since smartphones are handheld devices, manually pointing at the sound source could improve the recording quality. However, due to their compact size, modern smartphones often contain very small microelectro-mechanical system (MEMS) microphones [Bogue, 2013]. Due to their small and flat form factor, MEMS microphones are often omnidirectional, but the type of housing influences the directivity [Lewis, 2011]. It is almost impossible to predict the type of microphone and its characteristics due to the heterogeneous ecosystems of smartphone manufactures and devices. The recording quality is heavily dependent on built-in microphones, device casings and hardware pre-processing steps to reduce background noise. Due to this fact, an acoustic classification system has to account for a vast variety of alterations of recorded sounds.

2.4.2. Spectrogram computation

Spectrograms are widely used in the study of bird sounds and provide detailed visual clues that help to analyze avian vocalizations. Their level of detail is far greater than any textual transcription of bird sounds and almost every publication in the field of avian communication features spectrograms to visualize distinct features of bird song. Species identity is known to be encoded in vocal and non-vocal bird sounds and therefore will likely be represented in visualizations of those sounds. Spectrograms can be considered one of the most important digital representations of sound in avian research. Thus, visual representations of sound in the time and frequency domain will be the building blocks of the machine learning approach to automated bird sound recognition described in this thesis. Spectrogram computation however, leaves us with certain degrees of freedom, which I will explore in this section.

Sampling rate

Sound waves travel through a medium like air or water and can be measured as local variation in pressure, or local movement of the medium. Every device that is capable of measuring these local variations can be used to capture a *waveform* that represents the deviation of pressure from normal pressure over time. [Schlüter, 2017,

p. 46] When digitizing a recorded waveform, the continuous-time signal has to be discretized using samples spaced at certain intervals. The *Nyquist-Shannon sampling theorem* specifies an important condition for this process: A continuous-time signal can be exactly reconstructed from its samples if the *sampling rate* exceeds twice the signal bandwidth [Lyon, 2017, p. 134]. This means that an audio signal with a maximum frequency of 1000 Hz has to be sampled with a rate of at least 2000 Hz to be exactly reconstructable. Oversampling in form of a small safety margin helps to prevent the loss of information. Sampling at 2000 Hz results in 2000 discrete data points for each second.

In practice, the choice of the best sampling rate for recorded bird sounds is affected by some constraints. Usually, we are limited to sampling rates of 22.05 kHz, 44.1 kHz or 48 kHz. A sampling rate of 22.05 kHz would be more than enough for bird vocalizations since the vocal range of birds is usually limited between 500 Hz and 6 kHz. However, most devices capable of sound recording natively support sampling at 44,100 kHz—a de facto standard set by Sony for practical reasons when recording to analog video cassette tapes. The SWIFT field recorder uses a sampling rate of 48 kHz, another commonly used rate in modern audiovisual entertainment (e.g. DVD and Blu-ray). Choosing the most practical sampling rate based on native device capabilities avoids costly re-sampling.

Frame length and overlap

When computing a spectrogram, the discrete-time sequence of samples is broken up into periodically overlapping frames (or windows) of length N. Spectral information is then extracted by applying the discrete Fourier transform (DFT) to each frame. Following the notation of [Heinzel et al., 2002], the DFT transforms a vector of N complex numbers $x_k, k = 0 \dots N-1$ into a vector of N complex numbers $y_m, m = 0 \dots N-1$ with

$$y_m = \sum_{k=0}^{N-1} x_k \exp\left(-2\pi i \frac{mk}{N} \right), \quad m = 0 \dots N-1 \tag{2.1}$$

The computational costs of this transformation are $O(N^2)$. Most scientific programming libraries use a more efficient Fast Fourier Transform (FFT) implementation proposed by Cooley and Tukey [Cooley and Tukey, 1965] that only needs $N \log_2 N$

operations but requires N to be a power of 2. Some libraries implement various forms of the FFT, which use very efficient forms of the Cooley and Tukey approach, allowing for a significant reduction of computations for arbitrary input lengths.

Choosing an inapt frame length can lead to *leakage*—discontinuity between samples. Therefore, the time series of samples x_j is multiplied with a window function w_j to remove discontinuities. For a DFT of length N, the window function is defined by a vector of real numbers $w_j, j = 0 \ldots N - 1$. Many window functions exist, most of them are smooth 'bell-shaped' curves. One of the most widely used window functions for spectrogram computation is the Hann window, defined as:

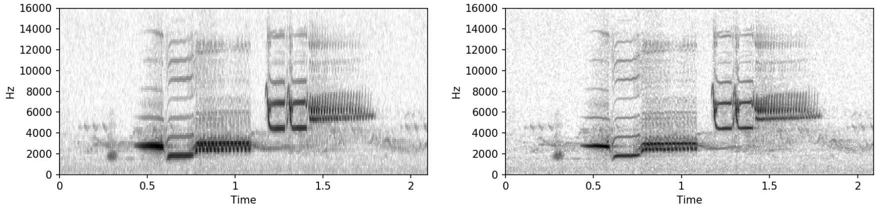
$$w_j = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi \cdot j}{N}\right)\right], \quad j = 0 \ldots N - 1 \tag{2.2}$$

Again, this notation follows Heinzel et al., who also state that the overlap between consecutive frames should account for the window width (narrow windows need more overlap) and give equal weight to all data. Heinzel et al. propose to use a Hann window with 50% overlap between consecutive frames for spectrogram computation. [Heinzel et al., 2002]

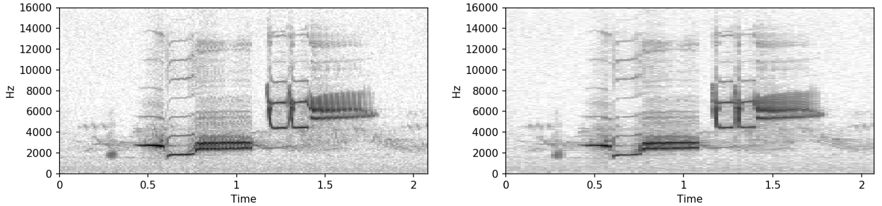
However, choosing the best frame length and overlap for a specific task can be challenging and often depends on the applied scenario. Larger windows result in higher frequency resolution, smaller windows provide more temporal details (Figure 2.14). For bird sounds, temporal resolution appears to be more important than frequency resolution (see Section 2.2.2).

The complex result of the short-time Fourier transform (STFT) for a single frame is added to a matrix that contains magnitude and phase for each sample in time and frequency. Computing the squared magnitude of this matrix results in the (power) spectrogram.

The shape of the output spectrogram is important for further processing and has to be taken into account when choosing values for frame length and overlap. Low input resolution is desirable to reduce computational costs, but the level of detail has to match the requirements of the target use case.



(a) frame length N = 256, hop size H = 128 (b) frame length N = 512, hop size H = 256



(c) frame length N = 1024, hop size H = 512 (d) frame length N = 2048, hop size H = 1024

Figure 2.14.: Different frame lengths and hop sizes using a Hann window for a Wood Thrush song. While short frames (a, b) result in high temporal resolution and blurred frequency representations, longer windows (c, d) show significant temporal blurring and sharp details along the frequency axis. For this sample, temporal resolution is more important than frequency resolution.

The resulting width S_w and height S_h of the output spectrogram S can be derived from the values for sampling rate f_s, duration t, frame length N, and hop size $H = N - overlap$ as follows:

$$S_w = \left\lfloor \frac{f_s \cdot t}{H} \right\rfloor - 1, \quad S_h = \left\lfloor \frac{N}{2} \right\rfloor + 1 \tag{2.3}$$

According to that, a spectrogram for a one-second signal sampled at 48 kHz using a frame length of 512 samples and 50% overlap (hop size = 256) has an output resolution of 186 × 257 ($S_w \times S_h$) data points (or pixels). Instead of a fixed number of samples, some implementations use milliseconds as length of a frame.

Frequency scaling

The human perception of pitch is not linear. Using a linear frequency scale when computing a spectrogram does not account for the logarithmic pitch perception of higher frequencies. Additionally, a linear frequency scale might overemphasize high frequencies. [Lyon, 2017, pp. 88-89] Humans and birds are limited in their auditory range and a compression of those high frequencies could reduce computational costs during further processing. Experimental pitch comparison formed the basis for a better scaling.

Stevens, Volkmann, and Newman proposed the *mel scale*, a log-like scale of pitch, which maps Hertz to Mels [Stevens et al., 1937]. The mel scale is one of the most frequently used perceptional scalings. Based on the assumption that an offset of 700 Hz marks the division between near linear and near logarithmic perception [Lyon, 2017, p. 88], the definition of the mel-frequency scale is

$$f_{scaled} = 2595 \log_{10} \left(1 + \frac{f}{700}\right). \tag{2.4}$$

The mel scale maps 1000 Hz to 1000 mels, a scale factor of 2595 provides near linear scaling for low frequencies and near logarithmic scaling for higher frequencies above 1 kHz (Figure 2.15). However, evidence suggests that the mel scale is an inaccurate reflection of pitch perception and it often faces criticism, especially for its high break frequency. Numerous other break frequencies have been proposed, e.g. Glasberg and Moore estimate f_{break} at 228.8 Hz based on tone-in-noise experiments [Glasberg and Moore, 1990] or Fant, who proposes a break frequency of 1000 Hz, which provides a more accurate approximation for higher frequencies [Fant, 1968].

From an engineering perspective, high break frequencies reduce the number of low-frequency channels needed to compress a linear frequency scale. Considering the general form

$$f_{scaled} = A \log_{10} \left(1 + \frac{f}{f_{break}}\right) \tag{2.5}$$

where A is an arbitrary scaling constant, different values for f_{break} might yield a more appropriate scaling for different scenarios. Birds are most sensitive to changes in frequency between 1 and 4 kHz, the auditory space of birds ranges from 500 Hz

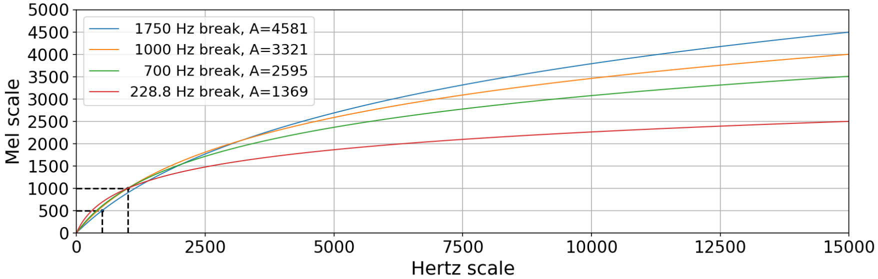


Figure 2.15.: Different auditory frequency scales. The mel scale proposed by Stevens, Volkmann, and Newman uses a break frequency of 700 Hz, while Fant proposed a break frequency of 1000 Hz. Glasberg and Moore estimate a break frequency of 228.8 Hz based on tone-in-noise masking experiments. All three curves provide approximately linear scaling until 1000 Hz and a logarithmic scaling for higher frequencies. Birds vocalize with higher pitch than humans. Therefore, a break frequency of 1750 Hz and approximate linear scaling until 500 Hz emphasizes higher frequencies, which provides better frequency scaling despite the lack of strong auditory evidence.

to 6 kHz. Birds usually vocalize with higher frequencies than humans due to their limited size. Therefore, using a high break frequency of 1750 Hz and a scale factor A of 4581—that maps 500 Hz to 500 mels—provides a frequency compression that accounts for avian auditory physiology. Yet, this assumption is purely based on technical considerations and lacks strong auditory evidence.

Magnitude scaling

The linear scale power spectrogram—as result of the STFT with squared magnitudes—does not reflect subtle changes in intensity due to its wide margin between values (Figure 2.16a). To counter this, and to provide a more detailed representation of intensity changes, different magnitude scales have been proposed. Some of them address the human auditory system, some of them are purely technically motivated. For the recognition of bird sounds in field recordings, three main types of magnitude scaling appear applicable.

Logarithmic scaling: The most widely used scale to reflect changes in intensity in acoustic signals is the decibel (dB) scale. As a logarithmic scale, it is defined as

$$S_{dB}(t, f) = 10 \log_{10}\left(\frac{S(t, f)}{r}\right) \tag{2.6}$$

for each value of a power spectrogram S in time t and frequency f. The reference power r typically refers to the maximum value of S or 1. Some implementations of the dB scale allow to normalize the scaled output S_{dB} to a maximum value with

$$S_{dB}(t, f)' = max(S_{dB}(t, f), S_{dB}(t, f) - m) \tag{2.7}$$

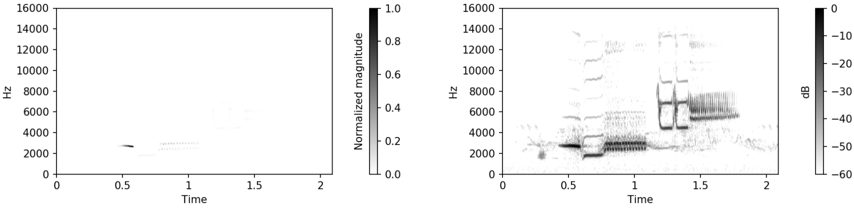
whereas m defines a dB threshold, e.g. of 60 dB (Figure 2.16b). Despite its common application for acoustic signals, the dB scale is not an accurate perceptive scale. Human perception of loudness is not equally sensitive for all frequencies [Lyon, 2017, pp. 50-53]. Based on the avian auditory physiology, the same applies for birds. Considering this, logarithmic scaling of squared magnitudes can be seen as yet another technical convenience.

Nonlinear scaling: Scaling magnitudes depending on the use case can help to yield representations that lead to better results during further processing. For bird sounds, Schlüter applied a nonlinear magnitude transformation (Figure 2.16c) for the classification of 1,500 South American bird species during the 2018 BirdCLEF challenge (also see Section 3.3.1).

The proposed transformation with a trainable parameter a for every time-frequency bin x is defined as

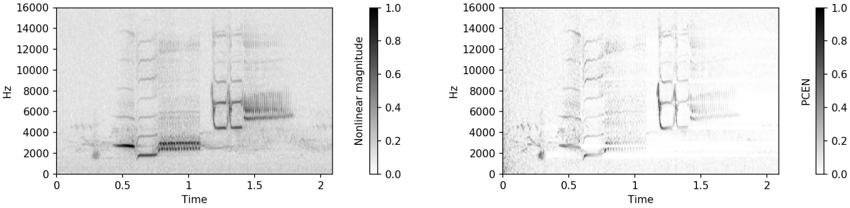
$$y = x^{\sigma(a)}, \quad with \quad \sigma(a) = 1/(1 + \exp(-a)) \tag{2.8}$$

and also amplifies low magnitudes but preserves more subtle details as a dB scaling. Schlüter proposed values for a ranging from -1.2 to -1.7, based on his 2018 BirdCLEF experiments. [Schlüter, 2018] Since empirical evidence verifies the suitability of this transformation for a bird sound recognition task, the application of this approach might improve the classification performance on faint bird vocalizations in soundscape recordings despite the lack of heavy noise suppression.



(a) Power spectrogram

(b) Logarithmic scaling



(c) Nonlinear scaling

(d) PCEN

Figure 2.16.: Different magnitude scales. The power spectrogram (a) with simply squared magnitudes does not reflect subtle changes in intensity. Typically, the decibel scale (b) is applied to achieve logarithmic scaling for changes in intensity. Schlüter proposed a nonlinear scale that preserves fine changes in intensity but might result in a noisy representation (c). Wang et al. proposed per-channel energy normalization (d), which adaptively suppresses noise but creates heavy artefacts for shorter chunks of audio. All scales were normalized to preserve comparability.

Adaptive scaling: The most sophisticated approach of magnitude scaling—that also has been applied in a bird sound recognition scenario—was proposed by Wang et al. to enhance far field human speech recognition: Per-channel energy normalization (PCEN). According to Wang et al., PCEN uses adaptive gain control to dynamically compress magnitudes instead of static compression (like in log or nonlinear scales). [Wang et al., 2017] The approach was successfully applied by Lostanlen for the detection of flight calls in noisy environments with slightly adapted parameter settings [Lostanlen et al., 2019], [Lostanlen et al., 2018]. However, the approach seems to perform worse on short chunks of audio and produces heavy artefacts (Figure 2.16d) compared to other magnitude transformations—even when parameter settings were trained [Schlüter, 2018].

The avian auditory system appears to be insensitive to changes of 3 dB or less. Additionally, intensity is a fairly inaccurate encoding of information in long-range communication. Choosing the appropriate magnitude transformation might be more important for the separation of overlapping signals of different individuals, or the separation of foreground and background, than it might be for species classification.

2.4.3. Adaption to avian acoustic monitoring

Preserving as much information as possible while maintaining a good overall compression in the final visual representation of a acoustic signal is vital for further processing. Again, finding the best configurations for the three most important domains of spectrogram computation is key. I will propose parameters that are a compromise of technical feasibility and auditory evidence for temporal resolution, frequency compression and magnitude scaling.

Temporal resolution

Considering the fact that temporal resolution might be the most important dimension of spectrogram computation, choosing the right parameters is critical . However, practical limitations often constrain the ideal selection. One of the most critical constraints is the actual output resolution of each spectrogram. The reason for this is purely technical: The input size of any image processing algorithm has to be as small as possible to ensure high throughput and low computational costs. Mainly, there are two ways to restrict the temporal output resolution of the STFT. First, limiting the length of audio that is represented as spectrogram to only a few seconds and secondly, adjusting the window size and overlap of the STFT to provide a maximum of useful details and a minimum of frames.

We already know that bird songs typically are longer and more complex than calls. Additionally, species identity is known to be encoded in song. Therefore, the appropriate duration of an audio signal that is visualized in a spectrogram has to reflect the expected duration of a bird vocalization to avoid cropping. Not every part of a song

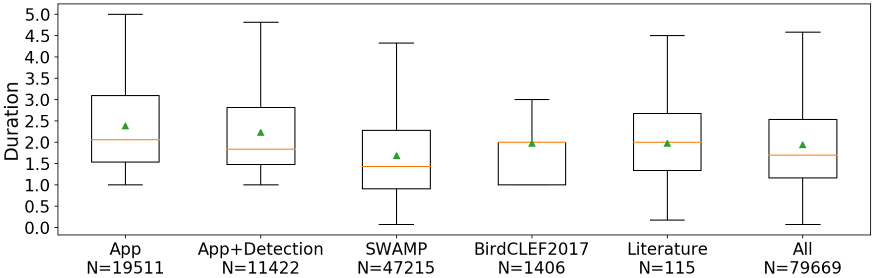


Figure 2.17.: Empirical results for the length of bird song. Users of the smartphone app were asked to isolate bird vocalizations by drawing an interval on the screen. Of those selections, roughly 60% contained a detectable bird species. The SWAMP dataset consists of 15 fully annotated days of soundscapes and features over 47,000 expert annotations of more than 80 North American species. The 2017 BirdCLEF dataset contains expert annotated soundscapes and over 50 audible species from South America. Only a few publications contain quantifiable measures of the length of bird vocalizations but also feature European species. Only intervals with a duration shorter than five seconds were considered.

is equally important (e.g. trills contain redundant information), but it is not feasible to predict which part is sufficient to identify a species. Therefore, the entire vocalization should be present in a spectrogram. Unfortunately, only a few publications mention quantifiable durations of bird songs across multiple species. As a result, I will rely on empirical data extracted from various sources to determine the average length of bird vocalizations across species (Figure 2.17). The empirical data contains almost 80,000 human selections (expert annotations and non-expert interval selections) and more than 100 published song durations extracted from [Beletsky, 1989], [Marler and Isaac, 1960], [Irwin, 2000], and [Dobson and Lemon, 1975]. The analysis of the data reveals that the average length of bird vocalizations across hundreds of species is 1.94 seconds. Thus, limiting the length of audio chunks depicted in a spectrogram to 2-3 seconds will ensure that almost every bird vocalizations is represented in its entirety.

Limiting the number of samples used to generate a spectrogram allows us to decrease the window length of the STFT to gain a high temporal resolution. Still, a temporal resolution in gap detection between two consecutive tones of differing frequency of 2 to 4 ms (which is typical for birds) requires a window length of only 64 (2.6 ms) or 128 samples (5.3 ms). At a sampling rate of 48 kHz, both would lead to a unfeasible number of frames and low frequency resolution. A window length of 512 samples (10.7 ms) appears to be a good compromise between output resolution and auditory evidence. With an overlap of 50% (256 samples), each frame corresponds to a time step of 5.3 ms.

Frequency scaling

Birds vocalize in a limited frequency range and the avian auditory system is not equally sensitive to all frequencies. The minimum and dominant frequencies of almost every bird sound lie above 200 Hz and below 10 kHz [Hu and Cardoso, 2009]. Therefore, restricting the visible frequency range in a spectrogram to values between 150 Hz and 15 kHz most likely represents the vast majority of bird vocalizations and reduces the amount of data needed to process this representation. Cutting off frequencies below the Nyquist threshold (which would be at 24 kHz if we sample at 48 kHz) can lead to leakage when applying a frequency compression. Thus, the use of band pass filters helps to flatten the critical frequency band between 150 Hz and 15 kHz and to avoid hard cut-offs. Typically, Butterworth filters [Butterworth, 1930] are used to avoid leakage at cut-off frequencies. Compressing the remaining frequencies to achieve a scaling that reflects the avian auditory system using a mel-like scale will likely result in a dense representation with maximum information preservation in the frequency domain.

Magnitude scaling

Highly directional recordings of bird vocalizations often include clearly audible sounds with fine details. Most publicly available sound files contain the primary bird sound with high intensity. However, omnidirectional recorders produce noisy recordings that blend ambient noise and bird sounds. The application of a magnitude scaling has to reflect on that. Considering the works of Schlüter, using a nonlinear magnitude scale seems to be the most appropriate choice for bird sound recognition in noisy environments despite the fact that this kind of compression is neither adaptive nor widely used.

Proposed workflow

Finally, the proposed workflow of spectrogram computation for avian acoustic monitoring can be outlined by the following steps:

- Open an audio file or stream at 48 kHz sampling rate, re-sample if necessary.
- Split the audio signal into 2- or 3-second chunks.
- Perform the STFT with Hann windows of 10.7 ms length (512 samples), 50% overlap, and 512 frequency bins that result in a fixed target output width and height.
- Apply a bandpass filter at cut-off frequencies of 150 Hz and 15 kHz to avoid leakage.
- Perform frequency compression using a mel-like scale with 64 mel bands and a break frequency at 1750 Hz.
- Transform the magnitudes using a nonlinear scale.
- Normalize the output between 0 and 1.
- Save the final spectrogram as lossless image for further processing.

The entire workflow provides the foundation for baseline experiments, and will be subject of further evaluation in Chapter 5.

2.5. Summary

Avian vocalizations—especially bird song—are often highly complex and consist of a rapid succession of elements. Birds are capable to utter two-voiced sounds thanks to their sophisticated vocal tract. The evolution of song is complex and—for most passerine birds—requires extensive learning, imitation and even improvisation. The intra-species complexity of song repertoires is vast and the number of songs a single individual is capable to sing ranges from very few to multiple hundreds. Local dialects add to the sheer amount of species-specific vocalizations rendering the identification of birds based on their sounds a complex task with a steep learning curve.

Additionally, birds are important indicator species for environmental monitoring and habitat assessment. One of the most common approaches to survey the species diversity of an area are point counts—labor intensive tasks that require extensive expert knowledge. Automated recording units can assist in this task but require even more time to analyze due to their hour-long recordings. An automated bird sound recognition system would be a transformative tool to assist ornithologists, conservation biologists and citizen scientists in the assessment of avian species diversity and long-term monitoring of critical environmental niches.

The worldwide community of birders provides extensive data that can be used to train and assess such an automated system. However, the provided data differs from what can be expected in the field. Audio recordings are of high quality, recorded with (semi-) professional and highly directional equipment and observations based on checklists are rare for remote habitats. Still, the sheer amount of available data provides a great starting point for development despite the differences.

Digital tools of sound transformation are commonly used in avian research. Visualizations of bird vocalization in the form of spectrograms have a long lasting tradition in bird biology and are widely adapted to analyze audio recordings. We can assume that visual representations of bird sounds also contain valuable information on species identity, rendering spectrograms a particularly suitable representation of avian sounds. Still, it remains to be seen which configuration and parameter settings provide a maximum of encoded information at a minimum of computational costs.

3. Acoustic Events and Deep Learning

3.1 Acoustic event recognition **61**
3.2 Deep artificial neural networks **64**
  3.2.1 Concepts and topologies 64
  3.2.2 Evolution of CNN 73
3.3 Evaluation campaigns for avian acoustics **76**
  3.3.1 BirdCLEF 77
  3.3.2 DCASE Bird Audio Detection 81
3.4 Summary **84**

The extraction of semantic features and meaningful information from audio recordings is a lively field of research that mainly evolves around the automated recognition of human speech. The identification of a vast number of sounds and acoustic events has applications in many areas of our daily life. The automated detection and classification of bird vocalizations is one of those applications. In this chapter, I will explore the evolution of machine learning attempts to solve the many challenges in this field. I will briefly introduce the academic field of acoustic event recognition, shed some light on recent technological advances in deep learning and will lay out the progress that has been made by participants of the two major bird sound detection and classification campaigns. Again, this chapter aims at providing an overview and I will reference further introductory literature that provides a more in-depth look at central ideas.

3.1. Acoustic event recognition

In 1953, Colin Cherry started to empirically explore the behavioral concepts behind the human ability to selectively attend to the voice of one speaker in a mixture of different speech signals (the ‘cocktail party problem’, [Cherry, 1953]). The field of computational segregation of sound sources evolved around the assumption that humans perform an auditory scene analysis (ASA) to transform the sounds of different sources into separate mental representations [Bregman, 1994]. The segregation of overlapping acoustic signals is of particular importance for automated speech recognition (ASR) over long distances in diverse environments—especially considering the problems of reverberation and additive background noise [Barker et al., 2013]. But human speech is just one modality of information encoding and with the inception of smart devices that can 'hear' (e.g. smartphones or robots), other sound sources and events are of interest as they might carry information not present in speech [Stowell et al., 2015]. The detection and classification of acoustic events is a subarea of computational auditory scene analysis [Wang and Brown, 2006] that assigns labels to perceived sounds.

We can observe some terminology confusion in the literature concerning the definition of acoustic event classification (AEC) and acoustic event detection (AED). According to Temko et al., we can distinguish between both fields using the following definitions [Temko et al., 2006b]:

Acoustic event classification deals with events that occur in isolated audio segments, each of which actually contains an event. The goal is to classify those isolated events based on acoustic features.

Acoustic event detection combines the identification of timestamps of events in continuous audio streams and the classification of those detected events.

However, since AED is often used to describe the detection of only one class or highly abstract classes of acoustic events (e.g. 'animal sound' or 'human speech'), the identification of bird species based on sounds can be considered a combination of AEC and AED due to its high number of classes and vast intra-class heterogeneity. Furthermore, we have to consider the difference between sound emission and perception. We could argue that the perception of a sound has to reflect the auditory capabilities of the receiver—auditory physiology or microphone characteristics—whereas the emission of sound includes vocal capabilities as well as acoustics in open or closed environments. Ideally, computational attempts to identify acoustic events would be robust against variations in emission and perception. However, it is often easier to reflect on perceptual characteristics in controlled environments (e.g. when technical recorder characteristics are known). Therefore, I will use the following definition to refer to automated bird sound identification:

Acoustic event recognition involves both tasks of detecting an event in a noisy input stream of audio and classifying it using a fixed amount of fine-grained target classes that require high-level concepts of acoustic events.

Acoustic event recognition (AER) for automated bird species identification consists of two processing stages: First, the occurrence of a bird sound has to be detected in a noisy stream of audio across a variety of digital receivers (AED) and secondly, this sound has to be assigned to one particular bird species (AEC). Both tasks are equally important and will be combined to one single step of species recognition using classifiers that are capable to suppress false detections for non-events.

The overall process of acoustic event recognition is based on the extraction of features and their classification to distinguish between audible sounds. This process is often complemented with extensive pre-processing of audio data and post-processing of classification results. Yet, the extraction of suitable features is vital to the success, and their design is often based on the human auditory system, e.g. the widely used Mel Frequency Cepstral Coefficients (MFCC). Some authors report good overall results using MFCC to identify environmental results [Cowling and Sitte, 2003]. However, speech features are not necessarily suited for the detection and classification of generic acoustic events [Chu et al., 2009]. Time- and frequency-domain features are commonly used in computational scene recognition ([Peltonen et al., 2002], [Eronen et al., 2006]) and are often complemented with transformations to reduce the spatial dimension of the resulting feature vectors [Zhuang et al., 2008]. Common classifiers include Hidden Markov Models ([Chen et al., 2005], [Temko et al., 2006a]), Gaussian Mixture Models ([Raboshchuk et al., 2015]), and Support Vector Machines ([Chu et al., 2009]).

Since their recent inception, deep artificial neural networks (DNN) excelled the performance of 'traditional' classifiers in the domain of acoustic event recognition and are successfully applied to identify environmental sounds [Shu et al., 2018], [Salamon and Bello, 2017]. The capabilities of deep neural networks in the acoustic domain are highly competitive, sometimes even resulting in perfect predictions in controlled environments [Kahl et al., 2017a]. We can assume that those classifiers are well-suited to process large and divers collections of avian recordings. This assumption is backed by the ever increasing performance of deep neural networks in well-known evaluation campaigns. I will explore the evolution of DNN in recent years and their application to various bird identification tasks in the following two sections.

3.2. Deep artificial neural networks

Machine learning marks a paradigm shift from classic programming—where humans input rules and data to automatically generate answers—towards a more generic approach where humans input data and answers to generate the rules (supervised learning). In 1950, Alan Turing objected Ada Lovelace's comment on general-purpose computers which—according to her—never do anything really new. Turing stated that computers might very well be capable of learning and originality. [Turing, 1950] Until today, this assumption fuels the ongoing pursuit to achieve true computer intelligence. However, recent machine learning approaches that solve complex tasks are designed to find statistical structure in a given number of examples, which allows those systems to come up with rules that eventually lead to an automation of this task. Even though the name suggests that the human brain stood as example for the design of deep artificial neural networks—a specific group of machine learning approaches—we cannot assume that the human brain implements anything like the learning mechanisms used in modern deep learning—despite conceptual references to neurobiology. [Chollet, 2017a, pp. 4-8]

Still, deep artificial neural networks significantly expanded the capabilities of modern data processing approaches to generalize and can be perceived as milestone developments towards computer intelligence. When developing and applying those technologies to automatically solve complex tasks, we have to keep in mind that deep learning is based on statistics and often heavily biased by input data. So far, deep learning has achieved remarkable breakthroughs in image processing, speech recognition, or machine translation (see [Zhao et al., 2019], [Hinton et al., 2012a], [Wu et al., 2016]). In this section, I will introduce the basic terminology, topologies and recent advances in deep learning using artificial neural networks.

3.2.1. Concepts and topologies

One of the most comprehensive works on deep neural networks is the 2016 edition of *Deep Learning* by Goodfellow, Bengio, and Courville [Goodfellow et al., 2016]. I will follow the definitions and notations for most concepts introduced in this book. François Chollet provides a more practical introduction in *Deep Learning with Python* [Chollet, 2017a]. Yet, the scope of this section is to provide the necessary terminology needed to follow the chapters on system architecture and experiments rather than providing in-depth details. Both of the aforementioned books are highly recommended as further introductory literature.

Deep learning

Although the term *deep* might suggest that we are dealing with an approach of learnable *deep understanding*, it rather refers to successively stacked nonlinear operations that form a processing chain [Chollet, 2017a, p. 8]. One core concept of deep learning is to replace hand-crafted features that describe input data with an automatically learned solution. In so-called *end-to-end learning*, the machine learning algorithms do not rely on any hand-engineered descriptors and instead learn to map raw inputs to predictions autonomously. This conceptual pattern is not reliant on extensive knowledge in the task domain. Still, the design of deeply stacked nonlinear operations—deep artificial neural networks—requires expert knowledge in the deep learning domain, extensive experimentation, and careful composition of data. [Schlüter, 2017, p. 14]

Feedforward networks

Deep feedforward networks (or multilayer perceptrons) are considered the quintessential deep learning models. Following the notation of [Goodfellow et al., 2016, p. 163], the goal of a feedforward network is to approximate a function f^*, where—for a classifier—the assignment of $y = f^*(x)$ maps an input x to a category (or class) y. Feedforward networks learn the value of the parameter θ and define a mapping $y = f(x; \theta)$ that results in the best function approximation. This way, the flow of information is directional and passes through the function being evaluated from x towards the output y. Feedforward networks do not contain any feedback connections. In contrast, recurrent neural networks (RNN) do feed outputs of the model back into itself and thus form a separate class of deep models specifically well-suited to process sequential inputs. While RNN are widely used in natural language applications [He et al., 2019], feedforward networks form the basis of many commercial applications, especially in the domain of object recognition in images [Sandler et al., 2018].

Feedforward models are composed of many different functions connected to a chain. For example, a chain consisting of three (differentiable) functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, which can be called first, second, and third layer of a network, has the form

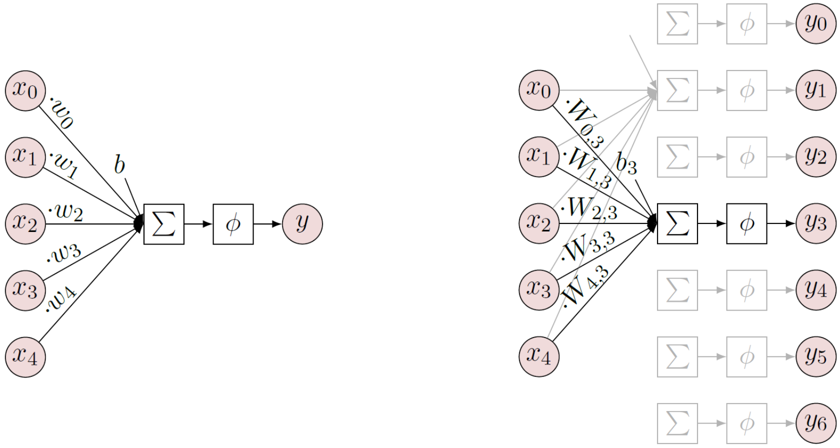
$$f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x))) \tag{3.1}$$

and thus a length (or depth) of three. The final layer of such a network is called output layer, intermediates are referred to as hidden layers. The given training data consists of noisy, approximate examples of $f^*(x)$ where each sample x is complemented with a label $y \approx f^*(x)$. The output layer must produce a value close to y, the behavior of all other layers is not explicitly specified by the training data. [Goodfellow et al., 2016, pp. 163-164]

Neuron activation

If we consider each network layer vector valued, we can interpret the role of each vector element analogous to a neuron (or unit), all of which act in parallel with each representing a vector-to-scalar function [Goodfellow et al., 2016, p. 164]. Conceptually derived from the signal transmission in the human brain, artificial neurons receive inputs from many other neurons and compute their own activation value. Values x_i of the incoming vector x are weighted by a weight vector w and offset by a bias b. The values of w and b are called the tunable parameters of a layer.

The mapping of all input values to an output scalar is computed using the weighted sum of the input values expressed as $\sum_i x_i \cdot w_i$ or the dot product $w^T x$. After adding the offset scalar b, the accumulated inputs are passed through the (often nonlinear) activation function $\phi(a)$ (see Figure 3.1).



(a) Visualization of $\phi(b + w^T x)$

(b) Visualization of $\phi(b + W^T x)$

Figure 3.1.: Neuron activation in feedforward networks. Values x_i of the incoming vector x are weighted by a weight vector w and offset by a bias b. The accumulated inputs are passed through the activation function $\phi(a)$ and are thus mapped to the neuron output y. Since all neurons of a feedforward layer share the same inputs x, we can express the vector of weighted sums as a matrix product $W^T x$. Visualizations shown in [Schlüter, 2017, p. 15]

In summary, the mapping of weighted inputs to outputs as the elemental function of feedforward networks is defined as:

$$y = f(x; \theta, \phi) = \phi\big(b + \sum_i w_i x_i\big) = \phi\big(b + w^T x\big) \tag{3.2}$$

In their most basic form, the neurons of each layer are fully connected to all neurons of the preceding layer. Since all neurons share the same input vector, the vector of weighted sums can also be expressed as matrix dot product $W^T x$. [Schlüter, 2017, p. 15-16], [Rey and Wender, 2011, p. 16-17]

Furthermore, we can derive the mapping from inputs x to labels y by an appropriate choice of the activation function of the output layer neurons—the transfer function $\phi(a)$. Mainly, two transfer functions are commonly used in classification scenarios. First, the logistic sigmoid function maps the inputs to a value between 0 and 1 and is defined as

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \tag{3.3}$$

where the resulting output scalars can be interpreted as probability that the input belongs to the target class. The sigmoid transfer function is the ideal choice for binary and multi-label classification tasks. The softmax function for each output unit i is defined as

$$s(a)_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} \tag{3.4}$$

and also assigns probability values between 0 and 1 to each value of the output vector. Since the sum of all output probabilities is 1, the softmax function is best used for categorical classification problems with 'one hot' labels.

The neurons of the hidden layers of a feedforward network usually all use the same nonlinear activation function, including sigmoid activation (logistic or hyperbolic tangent), or the widely adapted rectifier $r(a) = max(a, 0)$ [Glorot et al., 2011] and leaky rectifier $l(a) = max(a, 0.01a)$ activation [Maas et al., 2013]. Neurons with rectified activation are called 'rectified linear units' (short ReLU) and—along their derivates like PReLU, ELU, or SELU—are successfully applied in a vast number of scenarios [Ramachandran et al., 2017].

Training

Finding the best approximation of f^* is the goal of the learning algorithm. Stacking nonlinear functions (or layers) in a feedforward network allows to model nonlinear relations between inputs and outputs. While learning, we have to assess the error of the output y in comparison to the targets t using a so-called loss function $J(y, t)$. The choice of loss functions depends on the task and therefore on the choice of transfer function for the last-layer neurons (e.g. softmax or sigmoid). In binary classification, the binary cross-entropy

$$J(y;t) := -t\log(y) - (1-t)\log(1-y) \tag{3.5}$$

is a common choice. For categorical classification, the categorical cross-entropy

$$J(y;t) := -\sum_i t_i \log(y_i) \tag{3.6}$$

is commonly used to compute the network error [Schlüter, 2017, p. 22]. The goal of the learning algorithm is to minimize the output loss across all input samples. The fundamental concept of deep neural networks is to use the loss as feedback and to adjust the tunable weights of each unit accordingly. Computing the contribution of each parameter to the overall loss is done during *backpropagation* by applying the chain rule to compute the derivatives of complex, stacked functions. This process is also called *optimization*. Learning takes place when the adjustable weights of each unit are updated by changing their value by a small margin—the *learning rate*—according to the gradient of the loss. This process moves the loss towards a (local) minimum, thus leading to learned statistical representations of the input data. [Chollet, 2017a, pp. 46-52]

Convolutional layers

In fully-connected, dense feedforward networks, each unit of a layer is connected to all units of the preceding layer. Therefore, densely connected layers learn global patterns in their input feature space. In contrast, convolutional layers learn local patterns as they only connect a limited amount of preceding neurons to form an output. Local patterns can be recognized anywhere in the input data and provide significantly improved generalization capabilities. [Chollet, 2017a, p. 122] This is especially useful if neighboring values of the input vector are semantically linked. Convolutional neural networks (CNN, convnets) are the most widely used type of feedforward models in image processing. For images, the convolution operation converts input feature maps (3D inputs with *height* × *width* × *channels*) to output feature maps by transforming small patches (Figure 3.2). Taking advantage of the structural characteristics of an input can simplify the learning task when learned

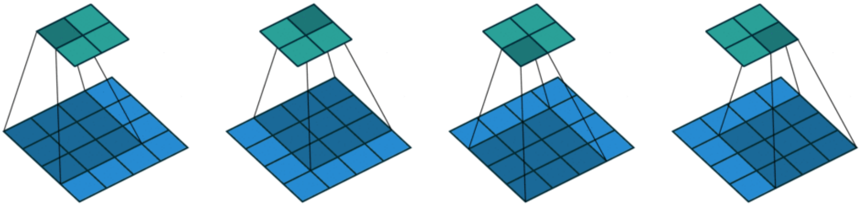


Figure 3.2.: Example of a convolution with kernel size 3x3. Convolutional layers learn local patterns as they only connect a limited amount of preceding neurons to form an output. In the depicted images, a 4x4 input matrix is transformed into a 2x2 output through local mapping. Images appear in [Dumoulin and Visin, 2016].

patterns provide better representations through exploitation of local value correlations. A fully-connected feedforward network neuron can be transformed into a convolutional unit with

$$c(X;\theta,\phi) = \phi\big(b + \sum_i X_i * W_i\big) \tag{3.7}$$

by replacing scalar inputs x_i with a matrix X_i, each scalar weight w_i with a matrix W_i (the so-called kernel) and scalar multiplication for vector values by a 2D-convolution operation (usually denoted with an asterisk) [Schlüter, 2017, p. 19]. Convolutional layers use the same activation functions for their units as densely connected layers.

Since the dimensions of each kernel are significantly smaller than the input, convnets have fewer learnable parameters than fully-connected networks. In addition, convolutional layers share kernel weights across all input values. Both characteristics provide convnets with two fundamental properties: Learned local patterns are translation invariant (equivariance) and CNN can learn spatial hierarchies of patterns as result of shape-altering intermediate pooling operations, padding, strides or through dilation with large receptive fields. [Chollet, 2017a, pp. 122-123], [Goodfellow et al., 2016, pp. 322-330]

Generalization

The recent success of deep neural networks derives from their ability to solve complex tasks (in computer vision). In fact, DNN are able to fit random labels to random noise for almost any kind of input data. Due to this, a number of difficulties arise when trying to force neural networks to generalize. Additionally, the term generalization itself is problematic—what do we mean when we say that a DNN generalizes well? Commonly, generalization means that a network trained to minimize the training loss is capable of achieving a similar performance on unseen validation samples. Still, training and validating data are often strongly linked since they originate from the same distribution. It remains questionable, whether we can assume that the performance on the validation data is representative of the performance on *any* unseen (real-world) samples. From an experimental point of view, validating the DNN performance using unseen but labeled samples is the only way of determining if one neural network generalizes better than others.

The capability to generalize is linked to the *effective capacity* of a neural network. Depending on the task, the number of parameters needed to fit the entire training data with zero loss is only depending on the complexity of the input value distribution. That implies, that any DNN with sufficient capacity can fit any label-data assignments, even random noise. [Zhang et al., 2016] Still, since every network with sufficient capacity will eventually memorize the training data, the best way of measuring the capability to generalize is through validation with unseen samples. If the training and validation loss diverge by a huge margin, we can assume that the DNN is focusing on semantically unlinked noise in the training data—an effect which is typically called *overfitting*.

Regularization

Avoiding overfitting and maintaining generalization capabilities is the main goal while training deep neural networks. A number of methods to achieve that goal have been proposed, some of them are very effective—although Zhang et al. (2016) state that it is often unclear why. Most approaches employ some form of *implicit* or *explicit regularization* for DNN with a large number of trainable parameters. Explicit regularization often affects the model capacity and typically includes domain-specific data augmentation (see Chapter 4), weight decay (or L2 penalty, [Goodfellow et al., 2016, pp. 115-116]) and dropout [Srivastava et al., 2014]. Implicit regularization methods commonly include early stopping of the training process and the well-established batch normalization [Ioffe and Szegedy, 2015]. However, the impact of each regularization method on the overall generalization capability has to be evaluated experimentally and strongly depends on the complexity of the task to solve. Regularization can significantly improve the performance of a deep neural network or having little impact at all. Additionally, the inapt combination of regularization methods might prevent the network from converging towards the minimal loss, thus requiring us to increase the model capacity.

Network design

One of the key considerations for a deep learning approach is determining the network architecture. The word *architecture* refers to the overall structure of a network [Goodfellow et al., 2016, p. 191]. According to the literature, the terms *architecture* and *topology* are interchangeable and both widely used. Still, I prefer to refer to the overall design of a neural network as architecture (e.g. feedforward, recurrent, convolutional) and to the specific succession of layers and their number of units as topology (e.g. shallow, deep, wide). The process of finding the best possible topology for a given task is often labor intensive and based on intuition. For some tasks, even one hidden layer might be sufficient, others often require very deep networks with up to several hundreds of layers (especially in image processing). The prototypical process of designing a suitable topology involves experimentation guided by monitoring the error function and incremental changes based on evaluation results. One of the most controversial design choices in deep learning evolves around the assumption that feedforward networks benefit from depth and not width. More specifically, increasing the number of learnable parameters without increasing the depth is said to be significantly less sufficient than an increased number of successive layers with an unaltered number of weights [Goodfellow et al., 2016, p. 197]. Yet, deep nets are considerably harder to train and require significantly more computational resources. Investigating the evolution of deep neural networks for image processing as part of the next section reveals that wide but shallow topologies perform better for some tasks.

3.2.2. Evolution of CNN

It is save to say that the annual *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC, [Russakovsky et al., 2015]) played a crucial role in the development and evolution of deep neural networks for computer vision. In fact, some of the most influential topologies, training methods and deep learning techniques have been proposed to solve the main task of this challenge: Identifying visual objects representing 1,000 different classes in real-world photographs. The competition itself started in 2010 and lasted until the year 2017, when the organizers decided that the (tremendously) complex task of visual object recognition can be considered as solved—with a top-5 error rate of 3.57%, human-level performance on this dataset has already been topped in 2015.

LeNet: Following Schmidhuber's timeline of deep learning [Schmidhuber, 2013], the inception of modern DNN started with Hubel and Wiesel's description of simple and complex cells in the visual cortex in 1962 [Hubel and Wiesel, 1962]. But the technical foundation was formed even before that with the description of the perceptron by Rosenblatt in 1958 [Rosenblatt, 1958]. The formulation and adaption of the backpropagation algorithm for neural networks during the 1970's and its eventual application for CNN by Yann LeCun in 1989 [LeCun et al., 1989] ignited the rapid development of more complex and more powerful deep neural networks. It was also LeCun et al. who presented the first, widely-adopted CNN architecture that was capable of classifying characters and handwritten digits with high precision [LeCun et al., 1998]. This shallow topology consisting of two convolutional, two pooling and three fully-connected layers was able to achieve an error rate of only 0.8% on 10,000 test images.

AlexNet: The milestone achievement—mostly considered as the breakthrough of deep learning—of winning the ImageNet competition in 2012 and beating the previous best top-1 error rate of 45.7% [Sánchez and Perronnin, 2011] by 8.2%, was accomplished by the SuperVision group including Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton [Krizhevsky et al., 2012]. The proposed topology consisted of five convolutional, three pooling and three fully-connected layers. The so-called AlexNet architecture follows the main design of LeNet but employs other influential design decisions like strided convolutions, ReLU neuron activation, and dropout. Until today, many CNN topologies follow the presented ideas and solve a wide variety of tasks with AlexNet-like designs. Subsequently, one year later, another

AlexNet-like architecture—ZFNet—won the ImageNet competition by adjusting the configuration of the original topology [Zeiler and Fergus, 2014]. More importantly, Zeiler and Fergus (hence the name ZFNet) established the first convenient way to visualize CNN features in their work, the reason why this topology is considered a major milestone in deep learning.

VGG: In 2014, Simonyan and Zisserman published their streamlined VGG architecture to compete in the ImageNet competition [Simonyan and Zisserman, 2014]. They did not outscore other contestants, but the topology of 16 weighted layers (VGG-16) remained a milestone achievement because of its lean topology that is easily reproducible. It also established the use of 3x3 filters instead of large kernels of 7x7 or even 11x11. Interestingly, Simonyan and Zisserman found that adding more layers and increasing the depth of the network did not improve the classification results. The VGG-16 version achieved a 8.8% error rate and significantly outperforms ZFNet. However, the VGG-19 version only achieved 9.0% error rate, leading to the assumption that depth is limited for streamlined topologies with stacked convolutions. The VGG topology is very wide, using a high number of filters and thus contains more than 13 million trainable parameters, which make it prone to overfitting for smaller datasets with less variance. Still, due to the use of stacked 3x3 convolutions instead of larger kernel sizes, the number of parameters is significantly smaller than in AlexNet that consists of more than 60 million tunable weights.

GoogLeNet/Inception: In the same year, Szegedy et al. attempted to construct a very deep architecture that consisted of 22 weighted layers [Szegedy et al., 2015]. The so-called GoogLeNet was able to set a new record of only 6.6% top-5 error rate in the object classification category. The authors successfully adopted 1x1 convolutions and the global average pooling operation from Lin et al. [Lin et al., 2013] to build very deep networks. GoogLeNet consists of successive 'Inception' modules that reduce the computational costs through dimensionality reduction (stacking 1x1 and 3x3 or 5x5 convolutions). The network also contains only one fully-connected layer for softmax classification preceded by global average pooling. As a result, GoogLeNet had only 6.7 million trainable parameters.

ResNet: The first successful attempt to top human performance on the ILSVRC dataset was presented by He et al. in 2015 [He et al., 2016a]. The described architecture consisted of up to 152 weighted layers—an incredible increase compared to previous attempts. To avoid the effects of vanishing or exploding gradients that can occur during backpropagation in very deep and narrow networks, the authors established so-called skip or shortcut connections to form 'Residual Blocks' consisting of two branches—one that contains convolutional layers and one that directly maps the input to the output of the block through element-wise addition. Error propagation can now skip the convolutional layers of a residual block, thus still reaching earlier layers of the network. An ensemble of multiple ResNets was able to achieve a top-5 error rate of 3.57% passing the human-level threshold of 5.1% set by Andrej Karpathy in 2014 [Karpathy, 2014]. In the following years, a number of additions and modifications to the ResNet architecture have been proposed. Most notably, Zagoruyko and Komodakis showed that 'Wide ResNets' with a reduced number of layers but increased number of filters outperform all previous versions [Zagoruyko and Komodakis, 2016]. Until today, deep residual networks remain the go-to architecture for many tasks due to their easy-to-implement topology and high classification performance.

DenseNet: Huang et al. built upon the idea of residual shortcut connections and promoted a deep architecture that passes collective knowledge from early layers to *every* succeeding layer [Huang et al., 2017]. These so-called densely connected networks concatenate outputs from dense blocks by stacking the channels of previous blocks. The width of the networks grows with every additional dense block, thus passing previously established features to deeper layers. The authors argue that the so-achieved diversified features have richer patterns. In fact, large DenseNets outperform a variety of ResNets on current benchmark datasets. Still, the implementation of those topologies is more complex and so ResNets often remain the preferred choice for many tasks.

MobileNets: In an attempt to further reduce the number of parameters and computational costs of deep neural networks, Howard et al. published MobileNets in 2017 [Howard et al., 2017]. The basic idea evolves around the observation, that depthwise separable convolutions perform on par with standard convolutional layers but significantly reduce the computational costs by limiting the number of input values that each filter processes [Chollet, 2017b]. Depthwise separable convolutions can be considered a special case of grouped convolutions (already introduced with AlexNet) where the number of groups equals the number of channels. The same idea was employed by Xie et al. to further improve the performance of residual networks [Xie et al., 2017]. In summary, recent developments and improvements generally aimed at maintaining the performance of well-established typologies and—at the same time—reducing the computational costs to make deep neural networks more applicable to real-world scenarios.

Born-again networks: We have to remind ourselves that top-performing systems presented to solve complex tasks during international competitions often use bagging to merge the predictions of an ensemble of networks. The ImageNet competition showed that network ensembles are very powerful and often outperform single nets. Still, those ensembles are not well-suited for real-time applications due to their incredibly high computational costs. Evolving around the idea of *knowledge distillation* presented in [Hinton et al., 2015], born-again networks employ student-teacher training where—instead of 'hard', one-hot labels—'soft' predictions of a teacher network are used to train the student [Furlanello et al., 2018]. The same approach can be used to distill large ensembles into one single network, which mostly maintains the overall performance and gains real-time applicability.

The investigation of recent advances in CNN topologies reveals techniques that are very efficient and widely adopted. However, the primary domain of the presented models is object classification in photographs. This raises the question whether these architectures are suitable to classify acoustic events. I will shift the focus of this investigation towards the recognition of avian vocalizations. Fortunately, two major evaluation campaigns evolved around this task. I will shed some light on recent developments within these campaigns in the following section.

3.3. Evaluation campaigns for avian acoustics

The comparability and reproducibility of scientific progress is key to advance a specific field of research. Unfortunately, many published works present experiments using proprietary sets of data. Establishing data collections that can be used to compare a scientific approach to the state-of-the-art helps to publish more transparently. The deep learning community mostly adopted major benchmark collections like the MNIST dataset of handwritten digits and characters [LeCun et al., 1998], the CIFAR-10 and CIFAR-100 tiny image dataset [Krizhevsky and Hinton, 2009], or the aforementioned ImageNET collection [Russakovsky et al., 2015]. The bioacoustics research community established multiple evaluation campaigns, each of which features a different dataset. Until today, no *de facto* standard has evolved and the campaigns are ever changing in terms of data and metrics. Yet, each annual edition led to various submissions of participating research groups presenting their approach for the same data collection— making those submissions comparable and progress (to some extend) measurable over consecutive editions. Due to the size of the provided training and test datasets, the number of included species, and participating research groups, the BirdCLEF and DCASE Bird Audio Detection challenge can be considered the most important evaluation campaigns to investigate the scientific progress in the field of avian sound recognition.

3.3.1. BirdCLEF

The *LifeCLEF Bird Detection Challenge* (BirdCLEF) launched in 2014 as part of the workshop program of the *Cross-Language Evaluation Forum* (CLEF). Before that, only three campaigns attempted the task of avian sound classification in audio recordings: The ICML4B Bird Challenge that was hosted on Kaggle and included 35 bird species and 90 test recordings [Glotin et al., 2013a], the 9th annual MLSP Competition featured 19 species and 645 recordings [Briggs et al., 2013], and the SABIOD and Biotope workshop at NIPS 2013 that included 1,000 recordings containing 87 species [Glotin et al., 2013b]. In the first edition, organizers challenged participants with a significantly larger dataset consisting of more than 14,000 recordings covering 501 bird species from South America [Goëau et al., 2014]. However, the recordings originated from Xeno-canto and mostly contained only one primary species (mono-species recordings). In contrast to other competitions, the task did not include omnidirectional recordings of soundscapes thus focusing entirely on species recognition in (mostly) high-quality recordings. Still, the total number of classes and audio files was unprecedented and posed a considerable challenge to a community that mainly experimented with 'traditional' features and classifiers. The organizers decided to use a ranking metric (mean average precision, MAP) to assess the performance of the submitted runs. Two scenarios were evaluated: First, the performance of the proposed systems including all annotated fore- and background species, and secondly, the performance considering only foreground (primary) species. It is noteworthy that audio files provided by citizen scientists on xeno-canto.org do only contain weak labels. Those labels state the species that are audible in one recording without any information on the timestamps. Additionally, some recordings might even contain false labels or might entirely be missing background annotations due to non-expert labeling.

The first edition of BirdCLEF in 2014 mainly saw approaches that included the extraction of low-level audio features using established frameworks like OpenSMILE ([Eyben et al., 2010]) or Marsyas ([Tzanetakis and Cook, 2000]). Most prominently, spectral features like MFCC were used in almost every attempt. Participants extracted features for segments of various length, but only Lasseck used probability estimations to decide whether a segment actually contains a valid bird vocalization [Lasseck, 2014]. All participating groups performed a dimension reduction on the extracted features to cope with large-scale data. The classification of segments and the eventual assignment of a label to each recording was done using support vector machines ([Martinez et al., 2014], [Leng et al., 2014]), decision trees and random forests ([Lasseck, 2014], [Stowell and Plumbley, 2014]), or nearest neighbor clustering methods ([Joly et al., 2014], [Northcott, 2014]). Only one participating research group decided to use a deep neural network to classify extracted features. However, the applied densely-connected network did perform significantly worse than most other classifiers [Koops et al., 2014]. The authors stated that the most plausible reason for the inferior performance was overfitting to the training data. Koops et al. also speculated that other network architectures might be more efficient and not equally prone to learn unrelated noise in training recordings.

In 2015—the second edition of BirdCLEF—the organizers decided to significantly raise the number of recordings in the dataset to more than 30,000. The dataset contained 999 South American bird species, almost doubling the previous number of classes [Goëau et al., 2015]. Interestingly, the performance of the proposed systems was on par with the results from 2014 despite the increase in complexity. Still, the large number of recordings forced participants to reduce the training data and the number of features—strongly implying the deficiencies of low-level audio feature classification for extremely large datasets. Again, MFCC were among the most frequently used features, the most successful classifiers were SVM ([Joly et al., 2015]) and decision trees ([Lasseck, 2015], [Stowell, 2015], [Meza et al., 2015]).

One of the most significant breakthroughs came in 2016 with the arrival of deep learning techniques for bird sound identification. The mono-species dataset of the third edition was unchanged, but the organizers introduced a new test dataset including soundscapes recorded by the Xeno-canto community [Goëau et al., 2016]. The performance of the proposed systems increased greatly, improving the mean average precision by more than 10% compared to previous editions. Sprengel et al. introduced a CNN classifier trained on extracted spectrograms that achieved a

Table 3.1.: Evolution of system performance on the BirdCLEF mono-species recording task based on mean average precision (MAP). In 2018, very deep topologies of CNN performed best, but well-designed shallow architectures were on par. This underlines the importance of task-specific optimization.

Year	Species	MAP	Classifier	Reported in
2014	501	0.453	Randomized decision trees	[Lasseck, 2014]
2015	999	0.454	Randomized decision trees	[Lasseck, 2015]
2016	999	0.555	5-layer CNN	[Sprengel et al., 2016]
2017	1,500	0.616	Deep Inception-v4 CNN	[Sevilla and Glotin, 2017]
2018	1,500	0.740	Deep Inception-v3 CNN	[Lasseck, 2018b]
2018	1,500	0.705	8-layer CNN	[Schlüter, 2018]

MAP of 0.555 including background species and 0.686 considering only foreground species [Sprengel et al., 2016]. The authors applied the classical scheme of image classification with deep neural networks to the domain of acoustic event recognition. The approach included the splitting of audio files into chunks, extracting mel scale spectrograms for each chunk and pre-filtering the training data by an elaborate signal-to-noise estimation based on morphological operations. Sprengel et al. applied data augmentation to all training samples consisting of pitch and time shifts as well as additional noise from rejected segments. The AlexNET-like 5-layer CNN was evaluated on a subset of 50 classes before the application to the entire dataset. As mentioned in the working notes, the authors were not able to successfully train a deeper neural network, confirming the observation that deeper typologies suffer from vanishing gradients. The next best system proposed by Lasseck achieved a MAP of 0.519 respectively 0.585 excluding background species [Lasseck, 2016]. Interestingly, Piczak also applied a deep convolutional neural network to solve the task but was not able to outperform Lassecks ‘traditional’ attempt [Piczak, 2016]. This result indicates that hand-crafted, well-designed features are key to successful avian sound classification. That assumption is backed by the observation that deep learning models showed inferior performance in the soundscape domain, at best achieving a MAP of 0.078 compared to 0.137 accomplished by Lasseck.

Still, the success of deep neural networks in the domain of sound identification led to the disappearance of MFCC, SVM and decision trees in all following editions.

All presented approaches in 2017 included CNN trained on extracted spectrograms [Goëau et al., 2017]. Again, the organizers decided to increase the number of recordings and classes to more than 48,000 audio files containing 1,500 South American species. The best performing system featured the Inception-v4 architecture trained on extracted spectrograms and achieved a mean average precision of 0.616 including background species, topping the previous benchmark by more than 6% despite the increase of recordings [Sevilla and Glotin, 2017]. This result confirms two main hypotheses: First, spectrograms are well-suited to encode species identity of birds and secondly, convolutional neural networks achieve a high detection quality for extremely large datasets. Our own approach—that closely followed Sprengel et al.—confirmed another interesting observation: Well-designed shallow (AlexNet-like) architectures perform on par with extremely deep models, achieving a MAP of 0.605 [Kahl et al., 2017a]. However, the detection accuracy for soundscape recordings remained low despite newly introduced recordings of higher quality from Peru and Colombia.

The 2018 edition of BirdCLEF saw another vast increase in the performance for Xeno-canto recordings. The dataset remained unchanged and the presented approaches all included deep neural networks [Goëau et al., 2018]. With his strong performance and a MAP of 0.740, Lasseck demonstrated the superiority of his segmentation and data augmentation approach for spectrograms [Lasseck, 2018b]. The author adopted his workflow from previous editions, replacing the classifier with a very deep Inception-v3 model. Lasseck evaluated various state-of-the-art topologies like Xception, ResNet152 or DenseNet but stated that none of those was able to achieve a better performance. Schlüter presented a shallow AlexNet-like architecture closely following the baseline system provided by us as organizers [Kahl et al., 2018a]. Still, his workflow of spectrogram extraction, model adaption and result pooling significantly outperformed our baseline approach [Schlüter, 2018]. His attempt of single-pass species prediction for arbitrary signal lengths included some interesting design choices that I will evaluate further throughout Chapter 5.

In an attempt to replace CNN classifiers by recurrent neural networks (more specifically long short-term memory networks, LSTM), Müller and Marti experimented with deep architectures trained on raw signal chunks [Müller and Marti, 2018]. The success of this method for speaker diarization ([Wang et al., 2018]) implied that this approach would be applicable for bird species identification. However, the authors were not able to train a system that achieved more than mediocre scores of 0.246

including background species. This outcome further cemented the superiority of deep convolutional neural networks in that domain.

The 2019 edition of BirdCLEF exclusively shifted the task of avian sound recognition towards soundscape analysis due to the lack of performance in this domain in previous editions. The dataset featured 15 fully-annotated days of continuous soundscapes containing 659 species and more than 80,000 labeled segments. My work as an organizer included the acquisition of training and test data and I will explore the dataset in Chapter 5 in more detail.

3.3.2. DCASE Bird Audio Detection

The IEEE AASP *Challenge on Detection and Classification of Acoustic Scenes and Events* (DCASE) was launched in 2016 to contribute to the increasing interest in computational auditory scene analysis [Mesaros et al., 2018]. Established challenges and workshops like the *Music Information Retrieval Evaluation eXchange* [Downie, 2008] or *TRECVid Multimedia Event Detection* [Awad et al., 2016] were not explicitly focused on environmental sounds and real-world acoustic scenes and were thus not suited to explore new methods of sound event classification. DCASE provided the ideal platform to promote a number of new competitions including the *Bird Audio Detection* (BAD) challenge that focused on avian sounds. Stowell et al. recognized the demand for automated bird sound identification systems to assess long-term monitoring data [Stowell et al., 2016] and launched the BAD track as part of DCASE 2017.

The challenge itself pursues a slightly different approach than BirdCLEF. As former participants, Stowell et al. focused on the mere detection of avian sounds instead of their classification. Yet, the task is extremely complex given the main constraint: Participants are asked to detect avian sounds in field recordings of an unknown domain. The shift of the acoustic domain between training and test data requires proposed systems to adapt to unseen environmental sound sources. The organizers provide four main datasets: First, the Chernobyl dataset collected in the Chernobyl Exclusion Zone [Wood and Beresford, 2016] containing more than 6,000 manually annotated items. Secondly, the crowdsourced WarblR dataset that contains 10,000 manually labeled 10-second recordings submitted by users of the WarblR smartphone app. Thirdly, the Freefield1010 dataset [Stowell and Plumbley, 2013] consisting of 7690 short audio clips selected from the Freesound online audio archive, and lastly, the PolandNFC dataset collected by the organizers [Pamuła et al., 2017] that contains 22 half-hour recordings that feature nocturnal flight calls. Datasets were split into segments, requiring participants to predict whether a segment contains an avian sound or not. Submitted runs were evaluated via leave-one-out cross-validation (train on one acoustic scene, apply to another) and the evaluation used the area under the ROC curve (AUC) measure as primary quality metric. [Stowell et al., 2018]

In the first edition, the superiority of deep neural networks was once again apparent. The 'traditional' baseline system provided by the organizers (proposed as part of the 2014 BirdCLEF challenge, [Stowell and Plumbley, 2014]) achieved 79% AUC, while Lasseck's systems (part of the 2016 BirdCLEF challenge, [Lasseck, 2016]) scored 84.2% AUC. Both systems were significantly outperformed by shallow, well-designed DNN architectures that even topped the performance of a very deep DenseNet [Pellegrini, 2017]. Cakir et al. proposed a neural net topology consisting of four convolutional and two recurrent layers [Cakir et al., 2017]. This network used convolution and pooling operations to reduce the input spectrograms in the frequency domain, passing the remaining time-series through gated recurrent units (GRU, [Cho et al., 2014]) and achieved an AUC score of 88.5%. Interestingly, this CNN-RNN fusion approach performs considerably well for the binary classification of acoustic scenes, implying that semantically enriched temporal information is sufficient to detect the presence and absence of birds.

Gill and Schlüter submitted the best performing system that scored 88.7% AUC and the proposed CNN topology contained four convolutional and three fully connected layers with only 370,000 trainable parameters [Grill and Schlüter, 2017]. The network design impressively demonstrates that task-specific layouts are capable of achieving state-of-the-art performance while maintaining practical applicability. The authors used mel scale spectrograms of fixed length (1,000 frames, 80 mel bins) to represent input recordings. The pooling scheme—to reduce the inputs of consecutive convolutional layers—accounts for the large temporal resolution and applies 3 × 1 max pooling for deeper layers, preserving decent frequency resolution to feed into fully-connected layers.

Consequently, the 2018 edition saw only minor improvements of the impressive scores of the first edition. This time, Lasseck applied the very deep Inception-v3 architecture to achieve the new best score of 89% AUC [Lasseck, 2018b]. The system and training scheme is closely related with Lassecks 2018 BirdCLEF submission.

Table 3.2.: Selected results of both editions of the bird audio detection challenge based on area under the curve scoring (AUC). The gap between various CNN topologies and decision trees is significantly smaller for the binary detection task of presence or absence of bird sounds. The results imply that task-specific network designs and training regimes are more efficient than elaborate architectures or domain adaption attempts.

Year	AUC	Classifier	Reported in
2017	0.842	Randomized decision trees	[Lasseck, 2016]
2017	0.885	4-layer CNN + 2-layer RNN	[Cakir et al., 2017]
2017	0.887	4-layer CNN + 3-layer FC	[Grill and Schlüter, 2017]
2018	0.788	Capsule Networks	[Vesperini et al., 2018]
2018	0.808	4-layer CNN + 3-layer FC	[Berger et al., 2018]
2018	0.890	Deep Inception-v3 CNN	[Lasseck, 2018a]

However, it remains questionable whether the elaborate segmentation of training data led to the improved score or if improvements are due to the complex CNN architecture.

One observation to back the argument that the influence of the applied CNN architecture is limited was made by Berger et al. who combined a different method of domain adaption with the CNN established by Gill and Schlüter [Berger et al., 2018]. The attempt yielded only 80.8% AUC score, significantly lacking state-of-the-art performance. In opposition to that, Vesperini et al. proposed a new deep neural network design employing so-called capsule networks (CapsNets) developed by Sabour et al. [Sabour et al., 2017]. This attempt resulted in even lower scores of 78.8% AUC despite the fact that CapsNet are designed to account for spatial hierarchies between visible objects. Considering both outcomes, it seems that CNN explicitly designed to solve a specific task also need a well-designed training regime to yield top performance. Additionally, it appears that the domain of acoustic event classification using spectrograms is less complex than object recognition in photographs and CNN architectures with a small computational footprint might already achieve competitive results when specifically adjusted to the task at hand.

3.4. Summary

Recent advances in the domain of acoustic event recognition impressively demonstrate that deep artificial neural networks outperform long-established classifiers like Gaussian mixture models, decision trees, or support vector machines. It is also apparent that low-level audio features do not suffice to represent large, complex datasets with multiple hundreds of classes. The frequent use of spectrograms to visualize avian vocalizations has explicit practical appeal in the domain of automated bird sound recognition. The evolution of deep neural networks advanced the field of object recognition in images beyond human-like performance. Yet, the domain of bird species identification in spectrograms is less complex despite its high intra-class diversity. Considering this, the design of neural network architectures, topologies, and training regimes that explicitly adapt to a specific task is crucial. The lively field of deep learning research is well-documented, well-explored but still requires expert knowledge to train and evaluate decent network designs. Additionally, dataset bias and lack of generalization are among the main concerns when applying neural networks to real-world use cases. In the following chapters, I will present a system design that accounts for recent developments in the deep learning domain, enables extensive evaluation on benchmark datasets and allows the distribution and application of trained models to a number of monitoring scenarios of avian activity.

4. System Architecture

4.1 Design Decisions **86**
  4.1.1 Extensive functionality 87
  4.1.2 Detailed configuration 90
  4.1.3 Domain-agnostic workflow 91
  4.1.4 Reproducibility and transparency 91
  4.1.5 Distribution and applications 92
4.2 Concepts and workflows **94**
  4.2.1 Components 94
  4.2.2 Training 96
  4.2.3 Evaluation 106
  4.2.4 Distribution 111
4.3 Implementation details **112**
  4.3.1 Audio and image processing 113
  4.3.2 Deep learning frameworks 114
  4.3.3 Web tools and services 116
4.4 Summary **117**

Since 2007, the Chair of Media Informatics at the Chemnitz University of Technology deals with the semantic enrichment of multimodal data. The main focus is on questions of human-computer interaction and the fusion of metadata of large heterogeneous datasets. During that time, an interdisciplinary research group developed two main frameworks: The Extensible Retrieval and Evaluation Framework (Xtrieval) dedicated to processing, indexing, and searching in large text corpora, and the Automated Moving Picture Annotator (AMOPA) designed to semantically enrich large audio-visual media collections. Both frameworks have been used to solve a variety of (scientific) tasks, and a considerable amount of research has been released (see also [Berger et al., 2015]).

Most notably, Kürsten published his approach to component-level evaluation in information retrieval using Xtrieval in 2012 [Kürsten, 2012]. Four years later, Wilhelm-Stein added the layer of applications to the system with his work on teaching the information retrieval process [Wilhelm-Stein, 2016]. Ritter presented a holistic attempt to extract high-level metadata from video footage of local TV stations using AMOPA in 2014 [Ritter, 2014]. All three approaches can be considered milestone developments that incorporated the entire workflow of dataset handling, training of an automated retrieval or classification system, and its application to real-world use cases and scenarios. I built upon the presented design decisions and ideas to develop a third system that combines component-level evaluation, a holistic workflow design, and applications to solve the complex task of bird species identification in audio recordings. Implicitly encoding the use case and underlying technology, the system is called *BirdNET*.

In contrast to common naming schemes that refer to a specific DNN design (like AlexNet, ResNet, or DenseNet), BirdNET is more than a single neural network architecture. BirdNET is a toolkit, a framework, and an infrastructure to train, evaluate, and distribute deep neural networks for acoustic monitoring of avian activity. Yet, the design of BirdNET is dedicated to research and thus not necessarily applicable on a consumer level. In this chapter, I will summarize my design decisions, the underlying concepts and workflows of training, evaluation and distribution, as well as some implementation details concerning third-party functionality.

4.1. Design Decisions

Wilhelm-Stein designed the *Xtrieval Web Lab* as a platform that allows users to perform retrieval experiments without programming knowledge [Wilhelm-Stein, 2016, pp. 127-129]. Built on recent web technologies, the Xtrieval Web Lab can be used to design and execute fine-grained experiments on a number of datasets. Users have full control over the succession of components, their configuration, and even the evaluation process. Similar to the fully configurable processing chains established by Ritter for AMOPA [Ritter, 2014, pp. 101-103], Wilhelm-Stein uses lanes that consist of an ordered set of components to model the process of input handling, data processing, and output distribution [Wilhelm-Stein, 2016, p. 136]. Following this scheme, BirdNET combines extensive functionality and a highly configurable, domain-agnostic workflow of components that helps to create reproducible results and task-specific applications.

4.1.1. Extensive functionality

Modern deep learning frameworks like Tensorflow[1] or Keras[2] provide functionality to not only build DNN but also to load and process input data, run experiments, visualize results, and deploy models to a variety of platforms. Both frameworks are meant to provide layers of abstraction that hide core functionality and guide researchers through the complex process of training deep neural networks. With that in mind, BirdNET was designed to cover all areas of deep learning, from data handling to training and evaluation towards model deployment and visualization.

Audio data handling: Loading and processing audio data is one of the key features to build a system for bird sound recognition. BirdNET can load and open files from a local storage or network resource, regardless of their encoding. Additionally, BirdNET allows to read from continuous audio streams over network (e.g. live streams of remote recording stations) or local system hardware (e.g. sound card and microphone). The audio processing component computes spectrograms from raw audio signals based on detailed settings and can be used as stand-alone library in other projects.

Image data handling: Since spectrograms are widely used to analyze avian vocalizations and have proven to be extremely valuable for automated bird species identification, BirdNET provides extensive image handling functionality. This includes the loading of 2D or 3D images from local storage, shape and value transformations like resizing, value or filter operations, and—most importantly—augmentation on sample and batch level. Augmentation includes almost every established method of cropping, value shifting, flipping, rotation, the addition of noise, as well as contrast, lightness, and hue transformations. Again, this component for image processing was designed as stand-alone library to provide a collection of image transformation techniques for other projects.

Multi-threading: Deep learning requires specialized hardware to efficiently compute weight updates of deep neural networks. In my research, consumer graphics

---

[1] https://www.tensorflow.org

[2] https://keras.io

cards (GPU) were used for batch forwarding and backpropagation. Training a DNN also requires a considerable amount of CPU time, especially for data handling operations like loading and augmentation. In an attempt to streamline the training process, BirdNET uses multiple threads to prepare batches of samples using the system's CPU while passing a single batch of images through the network on a GPU. A queue of readily prepared batches balances different processing speeds between CPU and GPU depending on model complexity or number of samples. This component is called *batch generator* and was specifically developed for BirdNET without multi-purpose application in mind.

Epoch training: Modern deep learning frameworks pass 4D tensors through convolutional neural networks. For images, the second, third, and fourth dimension represent channels, height, and width, the first dimension represents the number of samples or batch size. One batch (sometimes also called mini-batch) is a subset of training samples, the sum of all batches represents the entire training data. The process of passing one batch through the network, computing the loss and propagating it back through the network to update weights is called one *iteration*. A training *epoch* consists of all iterations needed to process every batch of the training data once. BirdNET employs batch training and—after each epoch—assesses the current training progress.

Online and offline evaluation: The assessment of the model performance on unseen data is done in two ways: Online evaluation takes place during training after each epoch with frozen weights, offline evaluation is done after the training process has finished and uses unseen audio files of mono-species recordings or soundscapes. Both assessment methods employ a set of complementary metrics that are used to determine the best overall result of the training.

Metrics: Evaluation campaigns have mainly established ranking metrics to assess the detection and classification performance of automated bird sound recognition systems. However, not every metric is suitable for every use case and thus BirdNET supports multiple evaluation measures. Those metrics include categorical and binary cross-entropy, top-1 accuracy, sample- and class-wise mean average precision, precision, recall, and various f-measures. Each metric has its specific strength and weakness, the combination of different assessments copes with dataset imbalances and bias as well as task-specific requirements. The metrics module of BirdNET allows the task-agnostic evaluation of automated classifiers.

Model deployment: BirdNET employs implicit regularization in form of early stopping. After each epoch, the online evaluation process determines the overall performance of the current model and saves a so-called *snapshot* of the model to local storage. After the training converges, the best overall snapshot is used for offline evaluation. In addition, the IO-component of BirdNET allows to preserve trained models along their input shape configuration for further usage. Models are stored as byte-wise data structure dumps, making the entire model easily reusable. Saved snapshots can be deployed to a number of applications, the IO-component handles all save and load functionality.

Model visualization: One main concern when training deep neural networks is the inability to comprehend why a specific model generalizes well and others don't. The term 'black box' is often used to describe complex model structures that lack transparency. The deep learning community has established certain methods to visualize and interpret weight configuration of DNN. Most notably, Olah et al. published a series of articles on how to understand the detection process when inferring test samples [Olah et al., 2017], [Olah et al., 2018], [Carter et al., 2019]. BirdNET implements some of those methods to raise the awareness of what the network actually learned during training. The employed methods include the visualization of weight and kernel activations, as well as occlusion and saliency maps for hidden layers.

Metadata handling: Time of the year and location are crucial aids when identifying birds in the field. Community projects like Xeno-canto or eBird provide vast amounts of metadata that can help to improve the automated detection performance. BirdNET is capable of incorporating those information into the training and evaluation process in form of labels, sample selectors, or post-filters for plausible predictions. All metadata is stored in unified text files on local storage and can be accessed during all stages.

Logging and statistics: Since BirdNET was primarily designed as research platform, logging of the training and evaluation process over time is one of the central functions of the system. Individual labels can be assigned to each experiment, thus making the results and statistics of each run traceable. Logging supports different output levels for general progress information, errors, and evaluation metrics. Log files are stored locally and can be accessed after each training cycle.

Applications: BirdNET is applicable to a number of scientific and real-world scenarios. Model storage and deployment supports the design of independent applications on various platforms like desktop computers, web browsers, smartphones or even low-power mobile recorders. Therefore, BirdNET also includes the infrastructure and interfaces to make models accessible using web technologies or stand-alone hardware. Applications include analysis systems for evaluation campaigns, the analysis of large amounts of field recordings, and the real-time analysis on (semi-) mobile devices. I will introduce those applications in more detail in Chapter 6.

Due to its extensive functionality, BirdNET has to be considered an expert tool that requires expert knowledge to operate. Although a number of applications and demos provide accessibility to a wide audience, the scope of the development did not account for consumer-level applicability or teaching purposes. The area of application is purely academic and serves the sole purpose of in-depth analysis and scientific exploration.

4.1.2. Detailed configuration

One truly remarkable code repository was released in 2015 by Ross Girshick as a very detailed complementary resource to the outstanding 'Faster R-CNN' tech report on arXiv [Ren et al., 2015]. The code base came with complete install instructions, pre-trained models, and example scripts to reproduce the published results[3]. Girshick allowed interested research groups to fully manipulate the entire detection and classification process and even train their own models. As a stand-out example of transparency in modern computer science research, the repository featured a central configuration file that provided access to the core functionality of the code base without having to search for crucial settings. In the following years, it became more and more common to not just release the experimental code base but also to refactor the code to make it more accessible. Today, almost every major contribution to the field of deep learning research is complemented with a code repository. Most evaluation campaigns require their participants to release the code they implemented to solve the challenge's tasks. In order to follow this tradition, a central configuration system that allows to manipulate every detail of the BirdNET training and evaluation workflow without having to find the corresponding portion of source code was implemented. This configuration system gives users full control over every aspect of dataset handling, spectrogram computation, data augmentation, model design and hyperparameters, training regime, regularization, evaluation data

---

[3] https://github.com/rbgirshick/py-faster-rcnn

and metrics, as well as the final deployment of trained models. Since the intended scope of BirdNET aims at scientific exploration, the entire workflow is dynamically configurable by editing a single file.

4.1.3. Domain-agnostic workflow

BirdNET is not just fully configurable but also allows to train on other than audio input data. Since all variables of input source and shape can be adjusted, adapting the classification workflow of BirdNET can be considered domain-agnostic. To simplify things even further, BirdNET uses folder names as one-hot labels. Due to that, researchers only have to swap datasets on local storage to train a DNN for a completely different usage scenario. This could include training a detection system for sounds of other animals like insects or mammals, switching domains towards ambient assisted living or even to the domain of image classification for photographs. BirdNET dynamically saves and loads trained models and allows to apply them for various use cases. The only requirement is a fitting input data shape, independent of its source. The system will automatically adjust to new input dimensions by reshaping the input tensors.

In the past, the domain-agnostic workflow of BirdNET was applied to contribute to a number of international evaluation campaigns like the TRECVid instance search challenge ([Kahl et al., 2016], [Kahl et al., 2017b], [Thomanek et al., 2018]) or BirdCLEF ([Kahl et al., 2017c], [Kahl et al., 2018a]). It also helped to solve some classification tasks from other acoustic domains [Kahl et al., 2017a]. Additionally, BirdNET was used to design a classification system for visual impairments of digitized (S)VHS tapes from local TV stations [Müller, 2018]. Still, the primary usage scenario is acoustic event recognition and future developments will be dedicated to that domain.

4.1.4. Reproducibility and transparency

The deterministic behavior of a scientific system is a cornerstone of reproducibility. Therefore, most approaches use fixed random seeds to eliminate result offsets due to unpredictable states. Randomization plays a significant role in deep learning and influences many aspects. Typically, a single random seed is used to initialize the system's randomizer at the beginning of each experiment. As a result, repeated runs of the exact same configuration will lead to the exact same results. Yet, using only one initialization process can lead to unexpected, inconsistent behavior as consequence of very insignificant changes to the randomized picking order. For example, removing only a single sample from the training data leads to changes in the randomized value sequence during the next experiment and thus to unwanted side-effects or even significantly different results. To counter this, BirdNET uses a single random seed but multiple initialization processes to ensure system consistency despite minor changes. The implemented initializers control sample selection, augmentation sequences, model initialization, batch shuffling, and validation sample selection and order. Consequently, changes in the composition of the training data no longer affect the initialization process of the model or the selection of validation samples due to independent randomizers. This way, model performance on different portions of the dataset can be evaluated—the observed effects are due only to changes in the sample selection.

With transparency in mind, I designed BirdNET to be comprehensible and reusable. The code base features extensive comments and documentation along refactored, modular implementations. The repository containing the source code for my BirdCLEF contributions is publicly available on GitHub[4] and will be expanded to feature web services and demos in the future.

4.1.5. Distribution and applications

Applying a scientific system that was developed to achieve good overall results on domain-specific training data to real-world use cases is a challenging task. On top of that, the achieved performance on a validation dataset (although independent) might not stand when the system is exposed to truly unseen samples. Application design and development are crucial parts of scientific exploration, but not many research groups ‘go the extra mile’ to open their system to the public. However, public demos are an excellent tool to communicate research progress and to gather feedback that can help to improve the system’s performance. Therefore, model deployment capabilities are a central aspect of BirdNET and account for three main design constraints: Platform independence, interfaces, and real-time processing.

Platform independence: BirdNET is applicable to a variety of stand-alone systems like mobile recorders, web servers, or desktop workstations. I tried to reduce

---

[4] https://github.com/kahst/BirdCLEF-Baseline

the number of code dependencies to avoid conflicts in versioning on different target platforms. Yet, some core functionality is provided by third-party libraries, most notably the deep learning back end (for more details see Section 4.3). As a result, BirdNET requires a(ny) Linux distribution to host the training and evaluation process. Since most applications were developed for access over internet, each client has to be capable of providing the resources for a web browser. Fortunately, many (mobile) devices run a Linux OS, making BirdNET available for Raspberry Pi, online cloud services, or smartphones.

Interfaces: BirdNET supports two kinds of interfaces: A programmable interface consisting of a set of methods and functions to instantiate a training or analysis process, and secondly, a RESTful API to allow access over the internet. Each interface is dedicated to different use cases and applications. For instance, running an audio analyzer on a Raspberry Pi requires to capture audio data from an input stream (preferably a microphone) and passing that data to the processing pipeline using pre-defined functions. Accessing BirdNET via smartphone also requires the recording of audio data and passing this data to the REST server that then starts the analysis automatically. (I will provide more detailed information on how to apply BirdNET in Chapter 6.)

Real-time processing: One of the most important requirements is the ability to process audio data in real-time. In the case of spectrogram analysis, 'real-time' means that the entire process of audio handling, DNN forward pass, and result computation has to be finished before the time represented in the spectrogram elapses. More specifically, the analysis of a three-second spectrogram must be finished in under three seconds to be considered 'real-time'. When running on specialized hardware—like in powerful, stationary workstations—processing a short signal chunks only takes some milliseconds. In contrast, on a low-power, (semi-) mobile platform like the Raspberry Pi, the efficient analysis of audio data is more challenging. BirdNET uses optimized workflows and caching to save time and computational resources. However, the design of the deep neural network significantly impacts the analysis speed. I will present a practical approach on how to reduce the size and computational requirements of deep neural networks in Section 5.2.7.

One of the core aspects during the development of BirdNET was its future distribution. The value of public demos and prototypes cannot be overestimated. Opening a system for citizen scientists and non-experts provides valuable insights into how people are expecting a system to work. Observing the usage of those demonstrators also provides clues on which aspects of the detection process are more important than others. Eventually, this process will lead to insights on how to cope with false detections, how to communicate the recognition process and how to build usable tools that perform well on domain-specific data despite some inefficiencies.

4.2. Concepts and workflows

The entire concept of BirdNET has been designed with two main objectives in mind: Scientific evaluation and application. The workflow for each task consists of task-specific modules that can be divided into three main groups: Utilities, core functionality, and interfaces. Additionally, certain external functionality provided by the host system and clients is needed to model the workflow of single processes. This section provides an overview of the interaction of constituents and their intra- and inter-component communication.

4.2.1. Components

BirdNET consists of components. Some of them serve low-level, stand-alone tasks, others combine functionality to provide high-level access to fundamental features. With transparency and reproducibility in mind, the interaction between groups of constituents was designed to be highly functional and domain-agnostic. Due to that, the implementation of each component leaves certain degrees of freedom in terms of dynamic configuration and usage. Still, each element serves a very specific task and plays a distinct role in the overall workflow (see Table 4.1).

The host system has to provide local storage for training and test data as well as intermediate results, logs, and model snapshots. BirdNET requires certain third-party libraries and frameworks, which have to be provided by the host system as well. Utility components are mostly stand-alone and can be combined to serve high-level tasks by providing dynamically configurable low-level functionality like spectrogram extraction, data augmentation, batch handling, model export, result evaluation, or the visualization of neuron activations. Core modules form processing chains of low-level components to build DNN, compile their static computational graphs including loss functions and optimizers, train models on large heterogeneous datasets, and evaluate their performance on a variety of test samples. Core functionality of

Table 4.1.: BirdNET components and their tasks. External functionality from host and client is needed to control and access the workflow. Utility modules provide task-specific functionality, which is then combined in core components of modeling, training, and testing. Interfaces provide access via programmable functions, a central configuration file and a RESTful API.

Group	Name	Task
Host	Storage	Training and test data, logs, and snapshots
	Packages	Third-party functionality
Utilities	Audio	File IO, spectrogram extraction
	Image	File IO, data augmentation
	Batch generator	Multi-threaded sample preparation
	IO	Model snapshot import and export
	Metrics	Result evaluation
	Visualize	Neuron activation and saliency maps
Core	Model	Dynamic DNN generation and graph compilation
	Train	Load data, train model, save snapshot
	Test	Load data, load snapshot, test model
Interfaces	Config	Centralized control over every aspect
	Functions	High-level, programmable access to components
	RESTful API	Access over internet
Client	Record	Capture raw audio
	Parse	Prepare analysis results
	Display	UI, visualizing detections

BirdNET can be accessed using a centralized configuration file, programmable, high-level functions or a RESTful API, which serves as linchpin for most applications and demos. The component-level workflow is based on community standards for deep learning systems and implements central ideas of AMOPA and the Xtrieval Web Lab. Individual components cover many aspects of their specific task including best practices and recent scientific advances in the field of acoustic event recognition.

4.2.2. Training

The training process of BirdNET is a core component that consists of several sub-tasks. For the recognition of acoustic events, those sub-tasks include data handling, spectrogram computation and pre-processing, data augmentation, and most importantly the optimization of a deep neural network. In this section, I will explore key functionalities of those tasks in more detail.

Data handling

In order to provide domain-agnostic functionality, all samples are locally stored on the host system. This way, users are able to easily investigate the data, look at samples and metadata, and can decide which data properties would influence the overall performance of a trained neural network. To streamline the data acquisition process, BirdNET requires images stored in folders that provide one-hot labels with their name. On top of that, metadata stored in a textual, human-readable format (in our case JSON) can be used to provide additional information on the contents of each sample.

At the beginning of each training cycle, BirdNET reads the dataset from local storage and decides which samples to use for training. The selection is based on user and task requirements and can be fully controlled using the centralized configuration file. The following properties of each sample are considered before the selection:

- **Origin**: Xeno-canto, Macaulay Library, eBird, AudioSet, or private collections; each source of audio recordings has its own characteristics that need to be taken into account.
- **Signal-to-noise ratio**: Estimated level of non-label sounds in the sample. The assessment is made during pre-processing of extracted spectrograms.
- **Rating**: Most community platforms provide user ratings to indicate the overall quality of a recording.
- **Bird seen**: If yes, the recordists states that she has actually seen the bird, which implies that the label has a high chance to be correct.

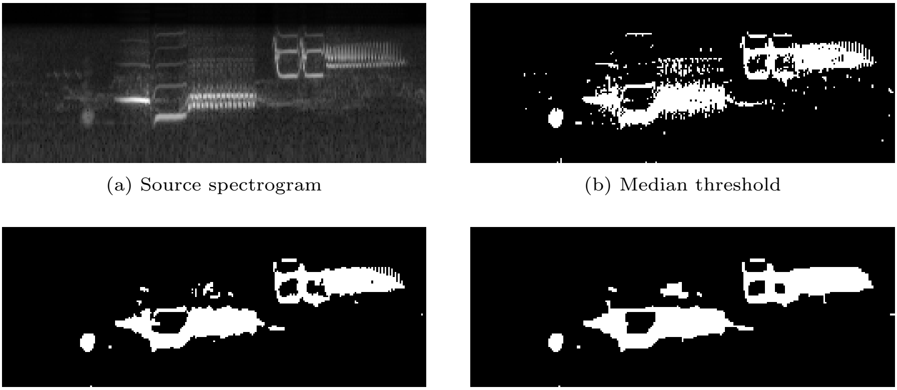
- **Background species**: Every recording we use for training only contains weak labels that do not state any temporal information. A high number of background species might lead to falsely labeled samples during the extraction of short audio chunks from a longer recording.
- **Samples per class**: Most training datasets have a significant class imbalance. To counter that, the number of samples per class can be limited or expanded before training.

The individual samples of each class are rated and ranked based on the above criteria. Next, a subset of samples is chosen to generate input data of maximum quality. Each input sample consists of a file path and corresponding label—the actual file handling happens during the batch generation stage. The sequence of input samples of the entire dataset is then randomized to minimize the bias of loss estimation and weight updates.

Spectrogram computation and pre-processing

When training on weak samples, the biggest challenge is to extract segments of audio signal that actually contain valuable information in regard to the assigned label. Considering bird song, this means that we have to account for segments of silence between successive vocalizations. This even becomes more apparent for sparse vocalizations like most calls. Some birds might even vocalize only a few times during a recording. In the meantime, background noise dilutes the label. Traditionally, the amount of actual signal in a sequence of values is estimated using one of the many measures of signal-to-noise ratio. Most signal processing libraries implement the most basic form: Mean divided by standard deviation. A number of other measures have been proposed accounting for deficiencies of the basic estimation. Still, for the recognition of acoustic events, especially bird sounds, task-specific methods need to be applied to distinguish between signal and noise.

Single elements (or notes) of a bird vocalization can span a number of time steps leaving spaces between frequency bins (e.g. slurs) or span a number of frequency bands but leaving temporal breaks in between (e.g. trills). In either case, a signal-to-noise measure has to treat columns and rows of a spectrogram separately. As mentioned in Section 2.4.3, spectrogram computation has to account for temporal integration in avian auditory physiology. Therefore, we can assume that fine-grained



(a) Source spectrogram

(b) Median threshold

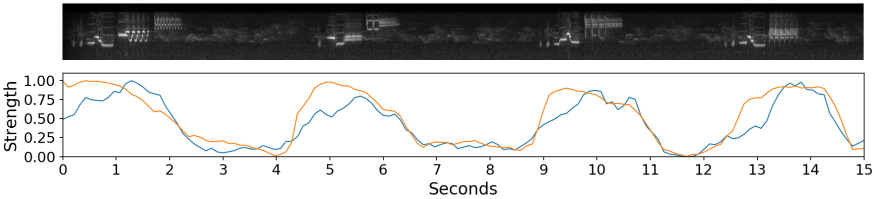
(c) Blur

(d) Morphological closing

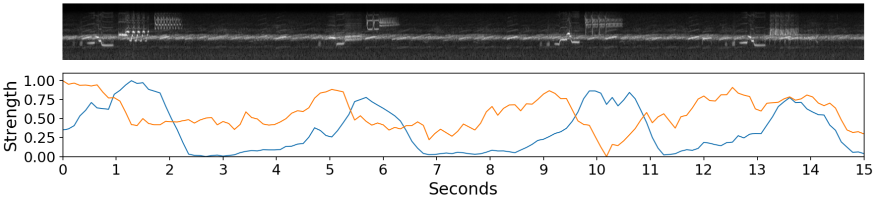
Figure 4.1.: Spectrogram pre-processing to estimate noise levels. Thresholding applies different values to frequency bins and time steps to account for rapidly uttered song elements (b). Blurring the result effectively eliminates single spots (c). Morphological closing merges single elements into larger units, indicating where larger portions of a song are (d). The number of remaining (white) pixels in the image indicate the signal strength in relation to non-song elements (black).

details are represented in each spectrogram. Applying a mel-like scale further emphasizes this effect in the frequency domain. As a result, a high variation in time and frequency indicates the presence of valuable signal information. The process of determining which pixel value of a spectrogram is part of a bird vocalization is depicted in Figure 4.1.

Sprengel et al. established this method with their participation in the 2016 BirdCLEF challenge [Sprengel et al., 2016]. Since then, a number of approaches in the same domain have successfully adopted this process [Chou and To, 2018], including our own BirdCLEF attempts [Kahl et al., 2017c], [Kahl et al., 2018a]. The method has proven to be robust against the most common impairments: The chorus of non-bird sounds (see Figure 4.2). This also applies to other constant sound sources like heavy wind, traffic or artifacts from mobile recorder defects. Compared to the basic method of signal-to-noise ratio estimation, the proposed method is mostly



(a) Mostly clean recording (🔊 7)



(b) Spring Peeper chorus overlay (🔊 23)

Figure 4.2.: Normalized signal strength estimation for two versions of a Wood Thrush recording. For the mostly clean version, the standard signal-to-noise ratio (mean divided by standard deviation, orange) highly correlates with the advanced estimation based on morphological features (blue). An (artificial) overlay of a Spring Peeper chorus—which is common for some regions in North America—significantly disturbs the standard measure. The advanced estimation remains mostly unaffected.

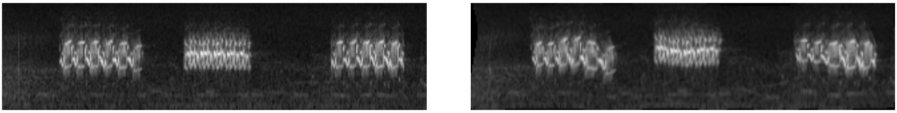
unaffected by heavy disturbance due to background sounds. This is important in order to maintain a high sample quality when extracting spectrograms from short audio chunks of longer recordings. Still, it remains unclear whether the contained signal actually represents a bird and if so, if it is the correct species indicated by the weak label. This distinction can only be made manually or by a sophisticated classifier—both methods are unfeasible for fast pre-processing of large amounts of audio data.

Data augmentation

In the domain of deep learning—a particularly ‘data-hungry’ domain—the need for training sample diversification has led to numerous transformation methods. Some of them, like random crop, rotation, or contrast changes also apply for the domain of acoustic event recognition—since we are dealing with images as well. However, domain-specific data augmentation is important to account for unforeseen variations in real-world samples as early as during training. In terms of bird sound identification, some methods have proven to be explicitly powerful. Those methods include the adaption to changes in pitch when birds adjust their vocal output according to a habitat. Additionally, the vast number of recording devices and environmental noise sources need to be represented in the training data. The shift in recording domains for high-quality, mono-species recordings and omnidirectional soundscapes is significant. The generalization ability of a trained DNN depends on the selection of training samples that represent the test data. Judging from recent publications, the following augmentations appear to be well-suited for this task:

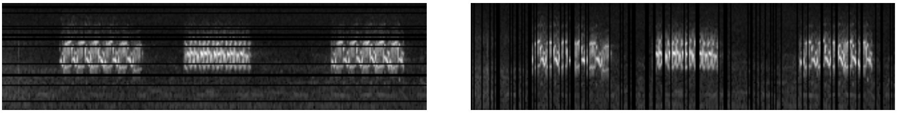
Vertical and horizontal stretch: Lasseck achieved state-of-the-art performance in the 2018 BirdCLEF and DCASE Bird audio detection challenge. In his attempt, he applied a number of data augmentation methods that account for the inner-species diversity of bird vocalizations [Lasseck, 2018b]. One of the best performing methods supposedly arose from the fact that birds change the pitch and tempo of their vocalizations on certain occasions. Lasseck decided to use vertical (frequency domain) and horizontal (time domain) stretches of randomly selected portions of the input spectrogram. This way, a specific selection of frequency bins or time steps is amplified (enlarged) and the resulting spectrogram is then trimmed back to the original shape.

Vertical and horizontal roll: Two years before that, Sprengel et al. implemented a similar but lossless version of Lasseck’s transformation [Sprengel et al., 2016]. In their attempt, pitch and time shifting is done using random vertical and horizontal roll. When rolling, the pixels of a spectrogram are shifted along one axis, elements that exceed the spectrograms width or height are added at the other end of the roll axis. Vertical roll accounts for pitch changes in bird sounds, horizontal roll simulated the process of short-time chunk selection from longer recordings. Both methods have proven to be fundamental to the success of bird sound identification systems.



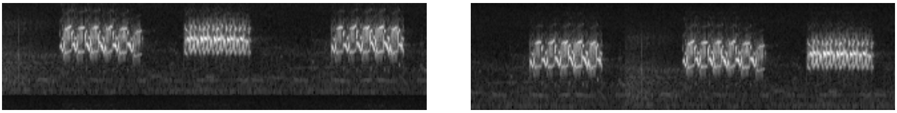
(a) Input spectrogram

(b) Elastic distortion



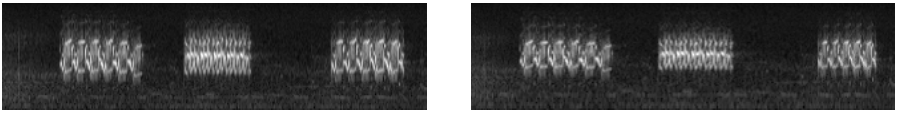
(c) Frequency masking

(d) Time masking



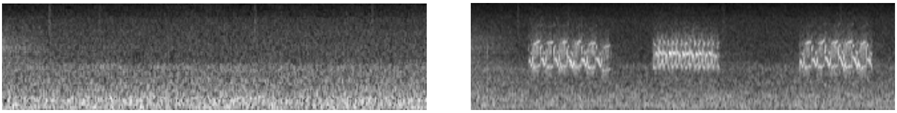
(e) Frequency roll

(f) Time roll



(g) Frequency stretch

(h) Time stretch



(i) Noise sample

(j) Noise sample addition

Figure 4.3.: Different domain-specific augmentation methods. Along common augmentations like random crop, contrast changes, or rotation, some methods are particularly useful for spectrogram transformation. Since the *acoustic adaption hypothesis* suggests that birds adapt to a habitat by altering the pitch of vocalizations, frequency stretch, roll or distortion have proven to significantly impact the detection quality. Other augmentation methods like the addition of noise samples, masking, or time roll and stretch account for impairments of recordings or other environmental sound sources. An (almost) infinite number of unseen samples can be created through the combination of the above methods.

Elastic distortion and warp: The combination of time and frequency shifts can lead to unrealistic distortions of bird sounds. Yet, deep neural networks often benefit from strong regularization—sometimes even heavily distorted input samples can be useful. One method of elastic distortion was applied by Simard et al. for the recognition of handwritten digits [Simard et al., 2003]. When applied to spectrograms, these local distortions lead to artifacts that mostly preserve the input signal but apply random shifts and stretches. A similar observation was made by Park et al. in 2019. The authors discovered that warping spectrograms along a given grid leads to strong recognition performance of human speech [Park et al., 2019].

Time and frequency masking: In the same article, Park et al. confirmed the applicability of spectrograms for the recognition of human speech. They also applied two other methods of data augmentation: Time and frequency masking. Arguably, not every element of an utterance is equally important for classification. The same applies to bird sounds, where trills repeatedly encode the same signal to counter reverberations. Considering this, masking (or dropout) of entire frequency bands or time steps can help to force a DNN to focus on semantically important features that are invariant to information loss.

Addition of noise: Environmental background noise is one of the main impairments of field recordings. Choruses of anurans or insects (like Spring Peepers or Crickets), heavy wind, rain, or even technical sounds like traffic, lawn mowers, or construction noise heavily impact the detection quality of an automated system. From an application perspective, each use case comes along with certain sound sources that need to be accounted for. When training a neural network, we are basically presented with two main strategies to avoid these distractions: Collecting ambient sound samples and using them to train a separate class of acoustic events, or adding these samples to existing spectrograms to force the network to ignore semantically unlinked information. One of the most powerful augmentation methods implements the second strategy: Noise sample addition. With the application of the advanced signal-to-noise estimation during the spectrogram extraction from training recordings, a number of samples will be rejected due to insufficient signal. Those samples are particularly well-suited to simulate different levels of ambient noise when added to a mostly clean spectrogram. Additionally, the addition of non-bird sounds results in new, ‘natural’ training samples. Our experiments as part of BirdCLEF have shown that this augmentation method significantly increases the overall detection performance [Kahl et al., 2017c], [Kahl et al., 2018a], [Kahl et al., 2018b].

Models and snapshots

When designing neural network architectures and topologies, we not only have to account for maximum performance but also the future use case when the DNN is applied. Certain constraints influence the overall network design. One of the most important constraints is the target platform. Not every DNN topology is suitable for all platforms, especially when it comes to (semi-) mobile hardware like the Raspberry Pi. Typically, reducing the capacity of a model to make it applicable to such hardware comes at the cost of accuracy. Therefore, the dynamic model generation of BirdNET allows to investigate a vast variety of topologies that are based on one architectural concept. During the process of finding the best model for a target platform, certain degrees of freedom can be adjusted. The best possible topology that still satisfies the requirements of the intended use case can be found through extensive experimentation. BirdNET supports two major DNN architectural designs and a number of variations of those.

AlexNet-like topologies: Shallow typologies with only a few layers have proven to yield strong performance for the detection and recognition of bird vocalizations [Grill and Schlüter, 2017], [Kahl et al., 2017c], [Schlüter, 2018]. Additionally, simple network designs are easier to train and considerably faster to evaluate due to the reduced training time. Task-specific design choices can be made in rapid succession, often leading to strong performance despite the lack of capacity. Additionally, the domain of acoustic event recognition in spectrograms is not as complex than other computer vision tasks. Considering this, it appears that the strong performance of shallow topologies is strongly linked with this circumstance. AlexNet-like architectures are well-suited for hardware platforms that do not contain specialized processors. Still, it became apparent that very deep networks outperform shallow networks by a small margin [Sevilla and Glotin, 2017]. Although achieving the best possible performance with those kinds of designs might not be possible, they are still highly applicable and thus integrated into BirdNET.

ResNet variations: Very deep architectures with multiple tens of layers achieve state-of-the-art performance for many scenarios. Despite the fact that unaltered topologies of those deep architectures can be successfully applied to the domain of bird identification [Lasseck, 2018b], task-specific layouts are needed to comply with design constraints like limited hardware resources. Of all deep network architectures, ResNets appear to be the most flexible. Building upon the initial ideas presented in [He et al., 2016a], a number of research groups have revisited the design. Most notably, He et al. proposed pre-activated blocks in [He et al., 2016b], Zagoruyko and Komodakis showed that ResNet derivatives with a high number of filters profit from dropout and thus need significantly less layers [Zagoruyko and Komodakis, 2016], and Xie et al. proposed residual blocks that derive from the Inception design [Xie et al., 2017]. Even DenseNets—as introduced by Huang et al.— can be seen as an extension of the initial ResNet architecture [Huang et al., 2017]. Residual networks are specifically robust against information loss. Huang et al. have shown that dropout of entire residual branches during training leads to better performance [Huang et al., 2016]. Veit et al. demonstrated that even the loss of multiple layers does not significantly affect the general detection performance [Veit et al., 2016]. Overall, the residual design allows to easily implement variations that can be strong additions to the initial layout. Still, the individual impact of each variation has to be explored through experimentation and is likely dependent on task-specific modalities. The available degrees of freedom are:

- **Model type**: Choosing the type of DNN for training affects the usage of almost every other degree of freedom. Two network architectures are available, the specific topology characteristics are dynamically build based on the following settings.
- **Nonlinearity**: Most common activation functions are supported, including the many variations of the rectified linear activation (ReLU) like leaky and very leaky ReLU activation, exponential linear activation (ELU), as well as identity mapping. The choice of activation function affects all layers except the input and output units.
- **Initialization scheme**: Random weight initialization also affects all layers equally. BirdNET uses He initialization ([He et al., 2015]) sampled from a normal distribution. The required gain factor is derived from the activation function and changes accordingly without the need for user input.
- **Number of filters**: The question whether wide or deep neural networks perform better can be investigated by setting different values for the number of filters (channels) in each layer. As input, a 1D vector is required, specifying the number of layers with its length and the amount of filters with its values. For AlexNet-like architectures a vector of *[32, 64, 128, 256]* would specify a neural network with four convolutional layers with twice the amount of filters

in each succeeding layer. For a ResNet, this vector would generate a DNN with four consecutive residual stacks (each with a later defined number of blocks) whereas each stack applies the same amount of channels to each block (and thus to each convolutional layer).

- **Kernel sizes**: Equivalent to the definition of layer count and amount of filters, the corresponding kernel sizes are specified by another 1D vector with entries of the form (h, w) that define height and width of the filters of each layer (or residual stack).
- **Number of groups**: Grouped convolutions were already introduced with AlexNet and can help to reduce the number of operations needed during a forward pass by limiting the number of channels that each group receives as input. The number of convolutional groups can be defined by yet another 1D vector that sets values for each convolutional layer (or residual stack).
- **Batch normalization**: One of the most important methods of implicit regularization is batch normalization. It can be activated for all convolutional layers of a neural network by setting a flag. Batch norm learns statistics of input batches over all channels and applies normalization by reducing covariance shift [Ioffe and Szegedy, 2015]. This method typically impacts the detection performance significantly but comes at the cost of slower training. Batch norm is applied before the activation of neurons and often works best with ReLU activations.
- **Dropout**: Lasagne provides three types of explicit regularization in the form of dropout: Random dropout of single neurons of a layer, dropout of locations (the same neuron in every channel) and dropout of entire channels. The type and probability of neuron deactivation during dropout is specified globally in BirdNET for all dropout layers.
- **ResNet K and N**: In their paper about wide ResNets, Zagoruyko and Komodakis introduced the two scaling factors K and N. Both impact the dynamic generation of any ResNet variation in BirdNET by multiplying the number of filters in each block (K) and by specifying the number of blocks in each stack (N). Both constants can be either specified as scalar or 1D vector that applies different values to each stack.

- **Global pooling**: Only fully convolutional neural networks can be generated with the aforementioned configurations. Recent DNN designs do not contain any fully-connected layers since they are prone to overfitting and greatly increase the number of weights. As an alternative, global pooling reduces all trailing dimensions beyond the second axis of incoming inputs before applying the final softmax or sigmoid activation. Schlüter introduced logistic mean exponential pooling in [Schlüter, 2018]—an addition specifically designed for the detection of multiple instances in one input sample. This appears to be an ideal choice for spectrograms that cover longer chunks of audio and might contain vocalizations of multiple species.

Models are then trained with randomized batches drawn from a subset of all available training samples. Learning rate decay ensures that each model converges towards the minimal loss, early stopping implicitly regularizes the training process. Several evaluation metrics are used to determine the best weight configuration, which is then serialized and locally stored for further examination and usage.

4.2.3. Evaluation

One of the key components of BirdNET is the evaluation module that assesses the performance of a trained net at certain stages. Two kinds of evaluation procedures are implemented: Online evaluation after each epoch at training time, and offline evaluation after the training has finished. During each procedure, different datasets and metrics are used to test the performance.

Datasets

As mentioned in Section 3.2.1, the assessment of the generalization capabilities of neural networks depends on test and validation data that contains unseen samples. Additionally, those datasets have to account for the expected use case of the model deployment. In order to cover different domains and applications while maintaining flexible experimentation, I decided to follow the widely-adopted scheme of training and validation splits that include samples that are most appropriate for the individual evaluation task.

Correlated validation set: Typically, when training deep neural networks, a large dataset is split into two folds: Training split and validation split. Since DNN require large amounts of data, a commonly used ratio is 9:1 for both splits. Each split covers all classes and has to contain enough samples to accurately represent the initial value distribution. Therefore, we can argue that the common (multi-fold) cross-validation scheme is not required in this case. Both portions of the training data can contain one single or multiple labels per sample allowing to assess binary or categorical error rates.

Uncorrelated validation set: In our specific domain, training samples represent short excerpts of longer recordings. Despite the fact that validation samples are never used to train, it is important to note that those samples are still highly correlated with the training data due to matching noise patterns from the original recordings. Due to this, BirdNET also uses uncorrelated validation data that contains samples that were solely extracted from validation recordings—entire audio files that are not part of the training data (but still might contain clues reflecting the recording equipment of bird watchers). Correlated and uncorrelated validation samples are used during online evaluation to monitor the performance of a DNN during training.

Mono-species recordings: Field recordings that only contain a single species are one of the target domains for BirdNET. In addition to the uncorrelated validation data, a portion of training recordings is used to serve as uncorrelated test data. Of all training examples, ten percent are exclusively used for offline evaluation after the training has finished. Due to the arbitrary length of those recordings, all scores are pooled (or bagged) to derive a single prediction. Evaluation metrics assess whether the pooled detection matches the initial (weak) label and—in some cases—the enlisted background species.

Soundscapes: Domain adaption capabilities of trained neural networks for acoustic event recognition are another important dimension of the evaluation process. Thanks to the effort of experienced annotators, it was possible to create a large soundscape dataset consisting of hour-long audio files from SWIFT recorders. The soundscape test data matches the BirdCLEF2019 dataset, thus making results comparable with other attempts. However, during the majority of experiments, only a (representative) fraction of the benchmark data will be used.

Metrics

The choice of metric significantly impacts the performance assessment of DNN. Depending on the task, we can choose from a variety of single- and multi-label metrics to account for certain constraints. Typically, each metric has its strengths and weaknesses. With this in mind, I decided to apply different, complementary metrics that fit different domains like single spectrograms, mono-species recordings and soundscapes, including benchmark metrics employed during the BirdCLEF and DCASE Bird detection challenge.

Cross-entropy: The most commonly used loss functions for classification tasks are the binary and categorical cross-entropy (see Section 3.2.1). Since training and validation data are (mostly) uncorrelated, a comparison between training and validation loss is fundamental to detect overfitting. Although explicit regularization like dropout affects the cross-entropy loss during training and not during validation, diverging loss values always indicate issues concerning the model capacity. BirdNET uses early stopping to prevent overfitting, the validation loss is used to decide which snapshot achieved the best performance. If the validation loss does not improve for more than five epochs, training is aborted.

Accuracy: In categorical classification using one-hot labels, accuracy is a common measure—despite its deficiencies. In our case, the top-1 accuracy for balanced validation datasets is one of the more strict metrics, requiring a DNN to detect the present class by assigning the highest score. The accuracy for targets t and predictions p as implemented in BirdNET is defined as

$$L_i = \mathbb{I}(t_i = argmax_c p_{i,c}) \tag{4.1}$$

where the prediction with the highest score for each sample has to match the index i of the one-hot label. This metric is omitted in multi-label scenarios due to the high number of true negatives.

Mean average precision: Ranking metrics are a better choice for tasks with multiple classes per sample. One of the most commonly used measures is the *mean average precision* (MAP). In contrast to the top-1 accuracy, the average precision reflects the rank of the desired labels among all predictions sorted by decreasing confidence. Since DNN use fixed output sizes that contain scores for each class, the average precision for one sample is defined as

$$AvgP = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{n_{rel}(S)} \tag{4.2}$$

where k is the rank in the list of returned species, n is the total number of returned species (equals the number of classes), $P(k)$ is the precision at cut-off k in the prediction vector, and $rel(k)$ is an indicator function based on binary labels that equals 1 if the item at rank k is a relevant species. Based on the ground truth, n_{rel} denotes the number of relevant species S for each sample with values from 0 to n. In real-world, multi-label scenarios, typical values for n_{rel} range between 1 and 5. The overall score reflects the mean across all samples, thus favouring imbalanced classes with more samples. But even in balanced datasets, the MAP is biased. Depending on the inner-class heterogeneity (e.g. repertoire size of one species), this metric implies good overall performance despite major deficiencies. However, the MAP is widely used and is one of the two main metrics at BirdCLEF, which makes results comparable.

Class-wise mean average precision: To counter a possible bias due to unbalanced data, the MAP metric can be adjusted to reflect classes and not samples (cMAP). This variation is of great importance to assess the real-world applicability of a trained DNN since it treats each class as equally challenging. Ranking metrics usually profit from large results lists that eventually contain every relevant species. Yet, for the analysis of soundscapes, particularly 'clean' results are needed. Therefore, detection systems have to apply a confidence threshold to avoid dilution of result lists. To account for this, the average precision for one class is defined as

$$AvgP(c) = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{n_{rel}(C)} \tag{4.3}$$

where the predictions of each class are ranked based on the confidence score and the indicator function $rel(k)$ denotes if the prediction at rank k is relevant. Acquiring the mean across all classes C with

$$cMAP = \frac{\sum_{c=1}^{C} AvgP(c)}{C} \tag{4.4}$$

results in a balanced measure of DNN detection quality that gives equal weight to each class independent from its initial bias (e.g. recording quality, number of samples, or species diversity). This measure is also the primary metric of the 2019

BirdCLEF challenge that explicitly focuses on continuous soundscapes that show high variance in the number of vocalizations for the contained bird species.

Precision, Recall, F-measures: Omnidirectional soundscape recordings are very challenging due to faint vocalizations and heavy background noise. Additionally, the output predictions for each soundscape segment have to be as accurate as possible. In many cases, only the highest scoring class can be considered. A convenient metric to assess the number of correct predictions compared to the number of mistakes in multi-label scenarios is the well-known *precision*, which is defined as

$$P = \frac{tp}{tp + fp} \tag{4.5}$$

and simply computes the ratio between true positives (tp) and false positives (fp). Precision is a strict measure that does not account for the rank of each prediction among all returned results. On top of that, the number of missed vocalizations is also of great importance. This measure is reflected in the *recall* metric that is defined as

$$R = \frac{tp}{tp + fn} \tag{4.6}$$

and accounts for the number of returned true positives compared to the number of missed vocalizations (false negatives, fn). Both metrics are strongly connected and sometimes contradict each other. Therefore, the harmonic mean of both metrics (F1-score) has been proposed. Considering the general form of the F-measure

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \tag{4.7}$$

with its scaling constant β, a shift towards precision—that favours high accuracy at the price of reduced recall—can be achieved by reducing β to 0.5 (which would be equal to the $F_{0.5}$-measure).

Area under ROC curve: The confidence of predictions plays a significant role in real-world use cases since results need to be dependable. Each F-measure only captures a specific snapshot that does not imply how sensitive the system is to changes in the minimum confidence threshold. The receiver operating characteristic (ROC) curve accounts for changes of this lower boundary. In compliance with the primary metric of the DCASE Bird detection challenge, the area under the ROC curve metric (AUC) is used to assess how distinct the distribution of confidence values is across all results. The ROC measures the true positive rate (or recall) against the false positive rate defined as

$$FPR = \frac{fp}{fp + tn} \tag{4.8}$$

where the number the of true negatives (tn) accounts for scores below a confidence threshold that actually do not contain a valid vocalization. Higher values for AUC indicate that a DNN can detect non-vocalizations and reflect that in the score. This is important considering long lasting segments of silence in soundscape recordings (e.g during the night).

Other implemented measures include the *top-1 mean confidence* to assess different pooling strategies and the *elapsed time per training epoch* to account for computationally expensive changes to the overall design of the DNN and training regime.

4.2.4. Distribution

Dependable results and fast computation are among the most important requirements that all application scenarios demand. Aside from that, a number of explicit variations to the overall detection process have to be made in order to cope with domain-specific constraints. Each scenario is based on different hardware, input data, and user-level interactions. Mainly, three distinctive application categories can be defined.

- **Large-scale analysis**: This involves the analysis of mono-species recordings and soundscapes. But only the latter has long-term applicability and would be the most transformative. The number of recordings acquired during the monitoring of a habitat over a longer period of time poses a significant challenge. Usually, high detection quality comes at the cost of computationally expensive processing. Although the analysis will be deployed to powerful workstations, fast processing of signal chunks is key to achieve a target processing time of one minute for one hour of audio data. On top of that, results need to be accurate with high F-measure and explicitly low false positive rate. Each input file will be split into short (overlapping) chunks to derive high-quality predictions despite high vocalization density. This is an explicit multi-label scenario.

- **Mobile recorders**: In an ideal setting, passive ARU would be replaced by active monitoring devices built with (semi-) mobile hardware. Since this scenario requires power-saving setups, the computational requirements of a deploy recognition system are at the forefront of design considerations. Yet, reducing model capacity to save resources often entails the lack of detection quality. I will focus on this use case during the experimental stage and explore technologies that allow the application of DNN for ARM architectures like the Raspberry Pi. Real-time processing of input spectrograms requires precise detections without the help of bagging or model ensembles.
- **Web and demo application**: The third application scenario evolves around the need for demos, prototypes and web applications to communicate scientific research. Fortunately, this domain is the most flexible in terms of detection quality and speed. Following the idea of *system-as-a-service*, access over internet does not require clients to execute computationally expensive operations. The expected amount of data is significantly less than during any large-scale analysis task and this allows longer execution times—real-time is not mandatory. However, this scenario requires task-specific error handling to communicate where and when the detection failed.

Each scenario has its specific constraints and requirements and thus needs explicitly adapted versions of BirdNET. Most importantly, model export has to comply with these circumstances and should make the deployment as easy as possible.

4.3. Implementation details

Publishing code repositories alongside scientific papers has become more an more common in the computer science community. Today, almost every major contribution to the field of deep learning is accompanied by an extensive, open source code base that enables interested research groups to investigate the presented results. Python is one of the most popular programming languages used in those repositories. The vast amount of third-party functionality makes it the ideal choice to rapidly advance scientific progress if distribution is not the primary goal. Additionally, every major deep learning framework has Python bindings.

Therefore, BirdNET's core functionality is also implemented in Python and I will introduce some of the most important additional libraries and frameworks in this section.

- **NumPy** *(https://www.numpy.org)*: This can be considered the most fundamental package for scientific computing in Python. The success of Python as programming language for scientific projects is largely due to NumPy's vast and extremely fast processing of N-dimensional inputs. Along with SciPy, NumPy covers most of the basic functionality of MATLAB but remains open source. BirdNET uses NumPy for all shape transformations of input data, prediction arrays, and basic image processing functionality.
- **Matplotlib** *(https://matplotlib.org)*: As another MATLAB-like Python library, Matplotlib provides 2D plotting functionality for scientific figures. Its functionality can be integrated with scientific workflows based on NumPy, rendering it a convenient tool to visually investigate the performance of an academic system. Additionally, all plots in this theses were created using Matplotlib.
- **Scikit-learn** *(https://scikit-learn.org)*: Measuring the overall performance of a computational system is one of the cornerstones of every scientific evaluation. Scikit-learn provides numerous, ready-to-use implementations of common evaluation metrics. Again, seamless integration into NumPy-based workflows is one of the key advantages of this library. In BirdNET, relevant metrics were implemented using Scikit-learn.

Additional functionality of BirdNET for smartphone apps or web applications is implemented in Java (Android) and JavaScript to support a high number of potential clients and platforms.

4.3.1. Audio and image processing

For the domain of acoustic event recognition, the processing of audio and image data has to be computationally inexpensive. Since BirdNET processes large amounts of input data in short periods of time (multiple times faster than 'real-time'), well-established third party libraries are used to cover those core functions.

- **LibROSA** *(https://librosa.github.io/librosa/)*: This Python package is intended for the design of music information retrieval systems. It contains the building blocks of convenient audio processing and provides an easy-to-use, high-level API. However, LibROSA requires a number of external libraries and frameworks, which makes it incompatible with some platforms (e.g. Raspian, the most popular Raspberry Pi OS). Still, the process of opening (any) audio files and re-sampling them to the target sampling rate with LibROSA is easy and fast (since it is build upon FFMEG).
- **pyAudio** *(https://people.csail.mit.edu/hubert/pyaudio/)*: Some of LibROSA's functionality is based on pyAudio, which provides Python bindings for the I/O library PortAudio. pyAudio allows to open continuous streams of audio from almost any source and is thus well-suited for real-time processing of live audio data on almost any platform. I use pyAudio to read audio chunks from external microphones of mobile platforms like the Rasberry Pi.
- **SciPy** *(https://www.scipy.org)*: Countless scientific projects rely on SciPy, a Python-based ecosystem of open-source software for mathematics, science, and engineering. SciPy provides a high-level API for a wide variety of scientific functions including signal processing. In BirdNET, SciPy is used to apply bandpass filters to audio input signals to soften frequency cut-offs.
- **OpenCV** *(https://opencv.org)*: As one of the most important image processing libraries, OpenCV provides core functionality for image handling in BirdNET. This includes loading and saving of images, transformations like resizing and cropping, and almost every data augmentation. OpenCV contains Python bindings and is compatible with all major OS distributions.

Additionally, BirdNET uses custom mel-like filter banks that avoid 'hard-snapping' of frequency bins that LibROSA employs. This implementation also allows to adjust every parameter of the resulting frequency scale, especially the aforementioned break frequency and scaling constant.

4.3.2. Deep learning frameworks

The number of available deep learning frameworks is ever increasing. Today, TensorFlow and pyTorch can be considered the two most important collections of deep learning functionality. The scientific community quickly adopted to the evolution of those and other frameworks and some of the most acknowledged works were built with Caffe, Torch, Theano, CNTK or even Darknet. The choice of the underlying backend for deep learning applications depends primarily on personal or task-specific preferences, as most frameworks offer comparable functionalities.

- **Theano** *(https://github.com/Theano/Theano)*: Similar to most frameworks for modeling mathematical expressions in Python, Theano was launched in 2010 by Bergstra et al. as alternative to NumPy/SciPy that makes use of specialized hardware like GPU [Bergstra et al., 2010]. Consequently, most of Theano's functions are also contained in NumPy. Yet, due to the efficient compilation process and use of various optimization procedures for task-specific hardware, Theano provides a dramatic speed-up compared to execution on CPU [Theano Development Team, 2016]. Additionally, Theano supports symbolic differentiation of complex expressions—one of the building blocks of deep learning. Well before the inception of TensorFlow in 2015, Theano provided state-of-the art functionality and a well-maintained code base. Unfortunately, Theano was discontinued in 2017.
- **Lasagne** *(https://github.com/Lasagne/Lasagne)*: One reason for the early success of Theano in the field of deep learning was the quality of available high-level APIs. Most notably, Keras and Lasagne provided easy-to-use collections of neural network layers, optimizers and loss functions [Dieleman et al., 2015], [Chollet et al., 2015]. While Keras aimes at providing the complete integration of every aspect of training and testing of neural networks, Lasagne focuses on model generation. Its extensive documentation and clean code base make it accessible and controllable. Additionally, due to personal communication with Jan Schlüter—one of the core developers of Lasagne–BirdNET profited from custom implementations for the very specific domain of acoustic event recognition.

Similar to TensorFlow's layer API, Theano and Lasagne require researchers to implement a significant amount of supporting code, since they cover only core functionality. However, they also allow users to build processing chains that can be controlled in every detail. Therefore, Theano and Lasagne are ideal choices for BirdNET—which itself can be seen as another abstraction layer of Lasagne in the fashion of nolearn[5] by Daniel Nouri.

---

[5] https://github.com/dnouri/nolearn

4.3.3. Web tools and services

Web applications and services are becoming increasingly popular for making research prototypes accessible to a wide audience. In most cases, the actual scientific system relies on very specific hard- and software requirements. In contrast, web clients run on almost every platform, sometimes even supporting embedded systems without any need for specific resources. A number of libraries provide extensive functionality to provide user-friendly access to research prototypes with limited amount of implementation overhead.

- **Pickle** *(https://docs.python.org/2.7/library/pickle.html)*: Converting dynamic data structures and object hierarchies to byte streams can be done with Pickle. The library allows to serialize any given state of a dynamic Python object into a persistent representation (usually stored on hard drive). Pickle allows to conveniently export trained models and their weights along various additional configuration data for further usage. Preserving the best model snapshot and importing it into a number of BirdNET applications with the help of Pickle enables to deploy classifiers to new platforms without having to adjust detailed settings.
- **Bottle** *(https://bottlepy.org)*: Python provides distributed system functionality with its Web Server Gateway Interface (WSGI). Bottle is a convenient micro-framework with no external dependencies that wraps basic WSGI functionality. Bottle allows to create RESTful web applications that can be accessed over the internet or locally using any kind of web client. Compared to other widely-used frameworks like Django[6], Bottle is ideal for rapid prototyping of simple applications due to its streamlined API.
- **Twisted** *(https://twistedmatrix.com)*: One of the most important features of Bottle is the interchangeable back end. This means that a vast number of other web server implementations can be installed to handle API requests. Considering the load that is to be expected when deploying public web apps, asynchronous server architectures are ideal for handling traffic when multiple CPU cores are available. BirdNET uses Twisted as server back end in combination with Bottle. Twisted is a well-tested, event-driven networking engine that handles simultaneous API requests (especially file uploads) for all of BirdNET's online applications.

---

[6] https://www.djangoproject.com

- **Bootstrap** *(https://getbootstrap.com)*: Responsive web applications that are accessible with a vast number of clients are important to communicate advances in scientific research. Bootstrap is a convenient toolkit to develop such applications with ease. It provides functionality to quickly design and implement complex web pages with pre-defined components. Bootstrap is used for all web demos of BirdNET.
- **jQuery** *(https://jquery.com)*: This is one of the most popular libraries for element manipulation, event handling, and (most importantly) Ajax requests written in JavaScript. The interface extension jQuery UI provides high-level abstractions of basic interactions with web pages. Both libraries are used to handle user events for all web applications.
- **Chart.js** *(https://www.chartjs.org)*: Making model outputs accessible to a wide audience is key to communicate how the sound recognition system of BirdNET functions. Typically, predictions of neural networks consist of 1D output vectors that contain class probabilities. Chart.js provides a convenient and extensive API to visualize those confidence values and other stats related to the classification process for web applications.
- **Leaflet** *(https://leafletjs.com)*: Interactive maps are an important tool to investigate user submissions to BirdNET. Leaflet is one of the most popular libraries to build mobile-friendly maps based on various tile renderers. Leaflet supports certain levels of interaction based on user events and is fully customizable in terms of the overall look and feel. The library supports rapid prototyping with its high-level API.

BirdNET also uses a custom, real-time spectrogram viewer implementation based on JavaScript developed at the Bioacoustics Research Program of the Cornell Lab of Ornithology.

4.4. Summary

Based on the implementation of AMOPA and the Xtrieval Web Lab, BirdNET was built to provide modular processing chains of core components. Main design decisions include extensive functionality, detailed configuration, a domain-agnostic workflow, transparent and reproducible implementations, as well as an application-driven development process. The overall workflow employs detailed data handling, audio processing capabilities, extensive data augmentation, dynamic model design, and export. A number of third-party, open-source frameworks and libraries provide additional functionality for training, testing and distribution. Moreover, BirdNET is a research platform that allows to design and evaluate sophisticated training regimes of deep neural networks for acoustic event recognition. Yet, it is not the implementation, demos, or applications that are at the center of my attention throughout this thesis. In fact, each component of BirdNET is interchangeable or replaceable. BirdNET is a tool to explore methods, algorithms, and scenarios that are applicable to a wide variety of scientific problems. The next chapter will provide the experimental foundation for the future adoption of model architectures, training schemes, and task-specific applications and their transfer to any other deep learning framework or toolkit.

5. Experiments

5.1 Goals and main focus **120**
5.2 Experimental investigations **120**
  5.2.1 Acquisition and composition of data 121
  5.2.2 Experimental setup 125
  5.2.3 Spectrogram computation 129
  5.2.4 Architecture design 133
  5.2.5 Topologies and training regimes 141
  5.2.6 Mobile architectures 152
  5.2.7 Benchmark system 157
5.3 Results **166**
5.4 Summary **181**

This section provides an in-depth analysis of the aforementioned requirements, constraints, and recent advances in the domain of acoustic event recognition. It covers different aspects of data acquisition, spectrogram extraction, neural network architectures, training regimes, and applications. I will study the impact of core changes to the previously proposed system and their (task-specific) implications on the overall detection performance. Building on previous chapters, I will evaluate detailed workflow settings that explicitly evolve around the main aspects of large-scale, long-term avian acoustic monitoring. Eventually, I will investigate the performance of the resulting benchmark system that will also be applied to a number of real-world scenarios in Chapter 6.

5.1. Goals and main focus

Experimental evaluation is the standard way to assess the performance of deep neural networks. Usually, the ability to generalize is tested by training the neural network on a (often large) amount of training samples and testing the prediction quality on representative test data [Goodfellow et al., 2016, pp. 117-118]. In most cases, training and test data are strongly linked since they originate from the same value distribution. Still, the ability to achieve excellent results on truly new, unseen samples is one of the major advantages of DNN compared to traditional classifiers.

My goal is to study the impact of changes to core components of the overall workflow. Each scenario and use case has different requirements and constraints and some of the results might not be transferable to other tasks. However, my experimental efforts will focus on key elements of DNN performance to ensure mostly task-agnostic results. Therefore, I do not aim for complete evaluation of all possible configurations (which is unfeasible for most aspects) but instead concentrate my efforts on distinct alterations of data processing, DNN architecture, and training regime while minimizing side effects such as dataset bias or random variations.

5.2. Experimental investigations

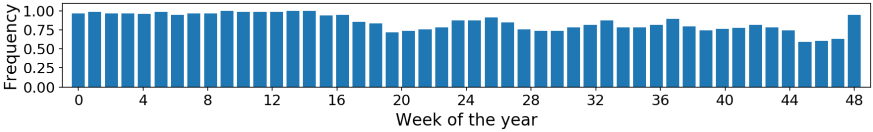
The design of the experimental studies in this thesis evolves around the ability of DNN to generalize on unseen samples despite a high number of classes with significant intra-class heterogeneity. I will test certain scenarios that focus on different domains of DNN applications with diverse data in mind. For the domain of acoustic event recognition for avian activity monitoring, the proposed approach first involves the acquisition of large training, validation and test datasets, which are mostly unlinked and represent real-world use cases. Secondly, a baseline setup to assess the initial system performance based on domain knowledge and assumptions derived from previous work will be established. Thirdly, the evaluation of different spectrogram extraction strategies, architectural designs, as well as various DNN topologies and their corresponding training regimes will build upon the baseline results. Next, I will propose and evaluate ways to reduce the model complexity for mobile applications. Finally, I will train a benchmark system, which is also subject of an in-depth analysis to decide on future improvements.

5.2.1. Acquisition and composition of data

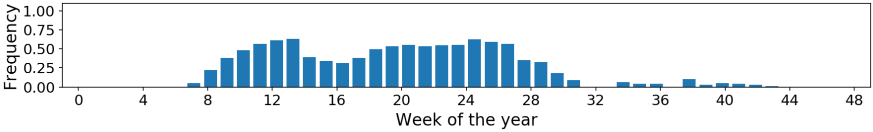
Selecting data samples that represent actual applications of deep neural networks is vital to ensure a good overall performance. Fortunately, the birding community provides vast archives of sound recordings for almost every bird on the planet. In this thesis, the focus is on North American and European species to allow real-world testing. Tremendous effort was put into the accumulation of audio files from both domains: Mono-species recordings and soundscapes. Expert annotators and consultants provided labels, bounding boxes and curated lists of birds for both continents. Additionally, metadata provided by citizen scientists was used to establish one of the largest datasets ever used in bird sound recognition.

The selection of bird species that would eventually form the contained classes of a trained neural network was based on two main sources of information. First, a curated list from expert ornithologists contained 595 species of North America that can be considered vocal. This list did not feature every possible species that might occur but included all common species that are likely to be encountered during monitoring scenarios. Secondly, European species were selected based on eBird frequency data. Therefore, the eBird API 1.1 was queried to generate class lists for grid cells, each with a size of 0.5 degree latitude and longitude. Every list contained occurrences based on submitted eBird checklists for every week of the year (reduced to four weeks per month, 48 weeks total, see Figure 5.1). The European class list included all birds that occur on at least 25% of all checklists (frequency) over at least four weeks per year. When the number of cells with that property exceeded 100, the species was selected as class. The resulting number of species for Europe is 555, which totals to an amount of 1,049 classes for both continents (due to some overlap). Again, the number of species reflects actual observations, which is crucial when considering future applications for smartphones or ARU.

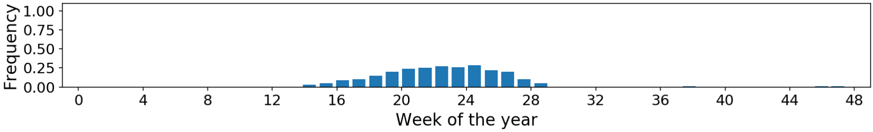
Based on these two lists of bird species, the web API of Xeno-canto and the Macaulay Library (ML archive and eBird) were queried to retrieve metadata and recordings. The Xeno-canto community collected hundreds of thousands of audio files, sometimes exceeding 1,000 recordings for a single species. To avoid future data biases due to imbalanced training data, I decided to limit the amount of retrieved recordings to 250 per species. Despite the fact that both lists contain birds that are frequently observed, each class is required to contain at least ten recordings. The reason for this were considerations of the experimental design where datasets are split into folds



(a) Great Tit (*Parus major*)



(b) Song Thrush (*Turdus philomelos*)



(c) European Serin (*Serinus serinus*)

Figure 5.1.: Relative frequency based on eBird checklist data for the city of Chemnitz, Germany. The Great Tit is a very common bird with high abundance year-round (a), the Song Thrush is partially migratory and has high relative abundance from March until August (b). The European Serin is not very common in this area but still fulfills the basic requirement for selection of at least four weeks with a frequency above 0.25 (c). Data provided by the eBird API 1.1

that dedicated 80% of the files for training, 10% for (online) validation, and 10% for (offline) testing. Since each fold needed to contain at least one sample per species with no overlap between training and validation data, ten files were considered the minimum amount of recordings. This restriction reduced the total number of classes to 984.

My previous work in this domain suggests that fallback classes, which contain non-events, are vital to the success of a recognition system. Based on those assumptions, various other data sources provided a variety of sound recordings that feature events

Table 5.1.: Different data sources that are used for training. Xeno-canto and the Macaulay Library provide vast archives of audio data for the 984 bird species of this project. Complementary, other acoustic events like insects, anurans, environmental, or technical sounds are also part of the dataset. Human vocal sounds form one additional class for training, so do non-bird animals. All other sound sources are merged into a non-event class.

Name	Classes	Files	Duration (h)	Size (GB)
Xeno-canto	984	118,882	1,798	156.3
ML archive	968	107,196	2,016	177
Non-birds	1 (83)	358	22	9.5
AudioSet	7 (16)	16,851	121	67.3
Freefield1010	1	5,755	16	5.1
WarblR	1	1,855	5.2	1.7
Combined	987	250,897	3,978.2	416.9

like vehicles, human speech, wind, rain, and other animal sounds. The Google AudioSet is one of the largest collections of human-labeled sounds that span a wide range of classes that are organized in an ontology [Gemmeke et al., 2017]. I selected 16 distinct events organized in seven classes that include human voice, whistles, and locomotion, insects, anurans, environmental, and technical sounds. As part of the DCASE Bird detection challenge, the Freefield1010 and WarblR datasets contain a high number of (unlabeled) non-events and account for 7,610 files derived from professional and semi-professional recordings. Sounds of other animals, especially insects and anurans that are common in North America are important to consider as sources of sound during our monitoring efforts in Ithaca, NY. Again, a curated list of other vocalizing animals was composed, the ML archive provided several hundreds of recordings that contain the most vocal non-bird species like Spring Peeper, American Bullfrogs, Chipmunks and Katydids. The list of recordings was extended with personal recordings contributed by Russ Charif and Mary Clapp.

The entire dataset features an unprecedented amount of recordings, almost 1,000 different classes of birds, and acoustic events with an accumulated run length of more than 3,978 hours. Every recording featured extensive metadata that was used to select high-quality recordings with supposedly correct labels in order to train an automated recognition system that is robust against unforeseen acoustic circumstances. Soundscapes with expert labels allow real-world testing and form comparable results. However, I did not use the entire dataset to train each iteration of the proposed DNN. Due to computational limitations, I decided to split the dataset into folds that served different purposes and sample compositions.

Sapsucker Woods 100, SSW100: This is the primary evaluation data split. It features the 100 most common species for the Sapsucker Woods bird sanctuary in Ithaca, New York and includes all 84 species that occur in the annotated soundscapes recordings. The training split consists of 36,072 recordings, the test split contains 3,966 randomly selected audio files. The dataset is imbalanced with a maximum amount of 500 files and a minimum amount of 56 per class. The dataset also features non-event recordings that were used as noise overlays during data augmentation.

BirdNET 1000, BN1000: The complete collection of all audio files contains more than 250,000 recordings from various (aforementioned) sources. The training data consists of 203,903 audio files that span 984 bird species. The test split contains at least one randomly selected recording for each species and a total of 22,175 files. This set also includes three non-event classes ‘Human’, ‘Non-Bird’, and ‘Noise’. This dataset is also imbalanced with a maximum of 500 recordings and a minimum of 10 recordings per species. Non-event classes contain up to 9,492 files.

BirdCLEF 2019 Test Soundscapes, BC2019: The test set of the LifeCLEF Bird recognition challenge consists of 335 fully-annotated soundscapes with a total duration of more than 280 hours. Of those soundscapes, 286 were recorded between March and July of 2017 in the Sapsucker Woods area in Ithaca, New York. In an incredible effort, expert annotators labeled more than 80,000 vocalizations that cover 84 bird species. During my experiments, the same ground truth (merged into five-second segments) and metrics as in BirdCLEF 2019 were used to assess the real-world performance of the final trained classifier.

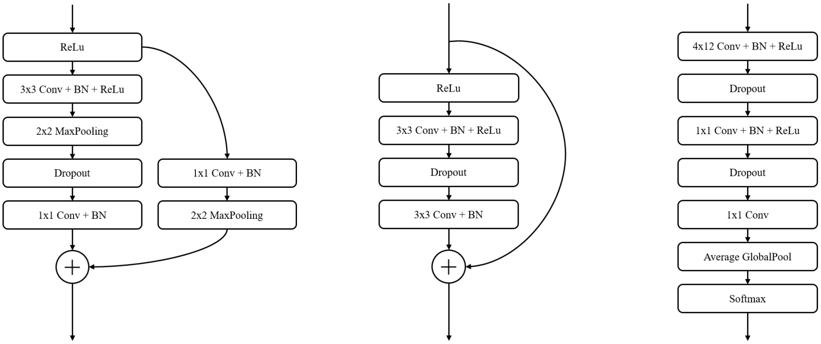
Dawn Chorus Soundscapes, DCSC: This is a representative collection of 24 soundscape recordings that include one hour before and one hour after sunrise of each of the fully-annotated days that are part of the BirdCLEF2019 test data. Each occurring species is contained in the SSW100 data split an thus recognizable by a trained classifier. Dawn chorus recordings are some of the most important soundscapes to survey the species diversity of a habitat and pose a considerable challenge due to a high number of vocalizations. For the sake of comparability, the ground truth used for this split also equaled the official BirdCLEF2019 annotations.

5.2.2. Experimental setup

Previous attempts in the domain of acoustic event recognition have led to some basic findings about the overall training and evaluation workflow. Since all hyperparameters (or settings) of each training cycle are interconnected, it is often impossible to identify those connections empirically. Therefore, I designed a baseline experiment, which featured preliminary hyperparameters, derived from previous experience. Each succeeding experiment built upon the results of this baseline attempt. This way, the best possible combination of hyperparameters would evolve over time. However, due to the holistic nature of these trials, the resulting workflow might still be improvable through further experimentation. In this section, I will define the variables, which were subject of investigation during the evaluation.

Evaluation mode: Due to the large amount of training data, I followed the scientific (de facto) standard of deep learning experiments. In contrast to x-fold cross-validation—which is common for traditional machine learning methods—each experiment consisted of three runs. Each run featured the exact same data source, architecture and training regime and was fully deterministic. However, randomness plays a significant role in deep learning. Therefore, each run used a different global random seed, which altered the order of samples, augmentations, and—most importantly—network initialization. The median score of those three runs served as the final assessment of the evaluated recognition system.

Audio processing: The computation of spectrograms from audio files has various degrees of freedom, which I discussed earlier. Not all of them were subject to investigation. I focused on two main constraints for spectrogram extraction: The DNN input shape and the average length of bird vocalizations. Empirically, the mean duration of a bird song ranges from two to three seconds (see Section 2.4.3, Figure 2.17). Longer spectrograms are more likely to contain a (weakly) labeled vocalization, shorter audio chunks provide a higher detection resolution. I investigated spectrograms that represented 2.0, 2.5, and 3.0 seconds of audio. For fair comparison, the resulting DNN input shape was kept consistent across all three variations. Considering the avian auditory system, the window length and overlap of each spectrogram were fixed in another series of experiments that lead to consistent temporal resolution but different input shapes. All other computational parameters (e.g. frequency and amplitude scaling) remained unaltered and followed the proposed workflow of Section 2.4.3.



(a) Downsampling block (b) Regular residual block (c) Classification branch

Figure 5.2.: Baseline design of DNN components. Based on the design of Wide ResNets, each residual block (b) contains two convolutional layers and one dropout layer followed by element-wise addition of the weighted and unweighted paths. Downsampling blocks (a) precede regular blocks and apply max pooling to reduce spatial dimensions and an additional convolutional layer in the shortcut branch to increase the amount of filters. All convolutional layers apply batch normalization (BN) before their ReLu activations. The classification branch (c) follows the design of Schlüter and reduces the input shape to a single dimension through average pooling followed by softmax activation [Schlüter, 2018].

DNN Architecture: The baseline experiment featured a modified version of the Wide ResNet architecture (Figure 5.2). Three core components form the succession of layers: First, a pre-processing stem transforms the original input spectrogram before it is passed through a series of residual stacks. Secondly, this sequence of residual stacks—consisting of downsampling and regular residual blocks— extracts features that are eventually passed through the third component, the classification branch. The initial design of the pre-processing branch is simple and contains a single 3x3 convolution with ReLu activation preceded by batch normalization. A 1x2 max pooling layer reduces the spatial dimension in the time domain. Residual blocks are identical with the ReLu pre-activated version of the design proposed in [Zagoruyko and Komodakis, 2016, Figure 1].

Table 5.2.: Baseline ResNet topology

Group	Name	Input shape	Output shape
Pre-processing	Conv + BN + ReLu	(1x64x384)	(8x64x384)
	Max pooling	(8x64x384)	(8x64x192)
ResStack 1	Downsampling block	(8x64x192)	(16x32x96)
	ResBlock	(16x32x96)	(16x32x96)
ResStack 2	Downsampling block	(16x32x96)	(32x16x48)
	ResBlock	(64x8x24)	(64x8x24)
ResStack 3	Downsampling block	(32x16x48)	(64x8x24)
	ResBlock	(64x8x24)	(64x8x24)
ResStack 4	Downsampling block	(64x8x24)	(128x4x12)
	ResBlock	(128x4x12)	(128x4x12)
Classification	Conv + BN + ReLu	(128x4x12)	(128x1x1)*
	Conv + BN + ReLu	(128x1x1)	(256x1x1)
	Conv + BN	(256x1x1)	(100x1x1)
	Global pooling	(100x1x1)	(100x1)
	Softmax	(100x1)	(100x1)

** this shape was altered to represent different time and frequency steps*

Downsampling blocks follow the original layout but employ max pooling instead of 2x2 strides. These blocks are also ReLu pre-activated and do not use a bottleneck convolution to increase the amount of filters. Instead, the number of filters is increased during the 3x3 convolution. The classification branch is derived from the design proposed in [Schlüter, 2018] but leads to just one prediction for the entire spectrogram to ensure fair evaluation of certain durations. Experiments concerning the overall architecture were supposed to shed light on the impact of topology changes for pre-processing, residual stacks and classification.

The resulting residual neural network consist of 110 total layers of which 24 are weighted (contain trainable parameters). The design is relatively shallow considering typical ResNets and cannot be considered very wide. With its 1.5 million parameters, the architecture has only limited capacity. However, baseline experiments featured only 100 classes and had to be as fair as possible for different input sources. Therefore, I chose an architecture that is not prone to overfitting and contains the necessary outline for further modifications.

Baseline training: Tuneable hyperparameters of the overall training process were chosen based on previous experience. Not all of them were subject to changes during further experiments. The success of a training regime often depends significantly on the interaction of numerous options. It remains questionable if the best possible combination can be found experimentally. The centralized configuration file of BirdNET contains 12 settings for spectrogram computation, 25 adjustable hyperparameters for the overall training process, 13 settings for DNN configuration, 17 data augmentation methods (each with at least 2 degrees of freedom) and 27 options for result post-processing. Since most settings are not simply binary but allow the choice of (almost) infinite assigned values, automated methods of hyperparameter optimization are not feasible considering the time needed to train a model. Due to this, grid search, random search, or even genetic algorithms are imperfect options. Following standard best practices, I decided to limit the investigation to essential settings based on their expected impact on the overall performance. For the baseline experiment, the following starting hyperparameters were chosen:

- No data augmentation
- 60 epochs with early stopping
- Batch size 32
- Constant learning rate of 0.001
- L2 weight regularization of 0.0001
- Adam optimizer
- Strictly balanced datasets
- 500 training samples per class
- 100 validation samples per class
- 5 test recordings per class
- 24 (dawn chorus) test soundscapes

In order to ensure fast training and to avoid bias due to imbalanced data splits, a subset of the SSW100 and DCSC datasets was used to train baseline classifiers on single labels. Frequency data derived from eBird checklists was used to post-filter soundscapes predictions based on date—a species had to occur on at least 2% of all checklists to be considered valid. This setup was not intended to achieve the highest possible scores. Instead, it was designed to ensure fair evaluation of different spectrogram computation modalities. All of the above hyperparameters remain unchanged during the investigation of audio visualizations. Under certain circumstances, other combinations might have favoured different input sources and provided better scores. For the sake of comparability, the baseline training regime was fixed.

5.2.3. Spectrogram computation

The first series of experiments evolved around the question whether high temporal resolution is key for automated bird species identification. Birds often have very complex vocalization with fine-grained temporal detail. For spectrogram computation, the temporal level of detail increases with shorter frames. However, shorter frames also increase the total number of frames for a fixed signal length. This impacts the input resolution of the resulting spectrogram. Considering future real-time applications, the input shape of the recognition system should be as small as possible. Additionally, the use of weak labels in the dataset might result in 'empty' samples when the extracted chunks are too short. On the other hand, short chunks provide better detection resolution in soundscapes. Longer chunks require wider input shapes due to more resulting frames with constant temporal resolution. Considering the average length of bird song, chunks with a duration between two and three seconds seem plausible. I decided to investigate two modes of spectrogram computation in terms of temporal resolution: First, a constant number of frames with varying frame length. Secondly, constant frame length but varying numbers of frames. Three hypotheses have been tested:

Hypothesis 1 *Spectrograms that visualize longer chunks of weakly labeled audio contain more valuable information and thus result in better classification performance despite lower temporal resolution.*

Hypothesis 2 *Spectrograms that visualize shorter chunks of weakly labeled audio contain less valuable information for successful training but provide better detection results in soundscapes.*

Hypothesis 3 *High temporal resolution (short frame length) improves the classification performance.*

Considering the investigation of song duration (Figure 2.17), I decided to evaluate signal chunks with a duration of 2.0, 2.5 and 3.0 seconds. To keep the resulting spectrogram shape (and thus the DNN input shape) constant, the frame length was adapted in a way that each spectrogram used Hann windows of varying length but constant overlap of 50%. Due to design constraints, the resulting spectrogram width had to be divisible by 8 to allow spatial reduction with square pooling sizes. Constant input shapes are crucial to ensure fair evaluation in which the DNN architecture does not affect the result due to a varying amount of trainable weights. The median score

Table 5.3.: Spectrogram computation experimental results (median out of three trials). All spectrograms use a mel-like scale with 64 bins to scale frequencies and 384 Hann windows with varying length and constant 50% overlap.

		SSW100 Val split		SSW100 Test split		DCSC	
D	FL	MAP	cMAP	MAP	cMAP	F0.5	AUC
2.0	10.7	0.555	0.49	**0.744**	**0.670**	0.160	**0.726**
2.5	13.0	0.576	0.514	0.736	0.656	**0.182**	0.724
3.0	15.6	**0.598**	**0.545**	0.737	0.661	0.140	0.709

D = duration in seconds, FL = frame length in milliseconds

of three experiments for each chunk duration was considered representative of the approach and used for comparison.

The results of the experimental evaluation of different chunk durations (see Table 5.3) confirmed the hypotheses 1 and 2. Longer signal chunks significantly improve the classification performance for single spectrogram predictions. This implies that three-second spectrograms are easier to train and lead to better classifiers. Shorter signal chunks lead to better performance in the specific domain of mono-species recordings and soundscape analysis. However, post-processing of predictions can ease the difference in classification performance through bagging of scores and overlapping intervals. Post-processing most likely cannot compensate insufficient training. Consequently, a spectrogram duration of three seconds appears to be the best choice for bird sound recognition.

Building on these results, hypothesis 3 was tested. Since a small DNN input resolution is desirable, variations in overlap between consecutive frames lead to different numbers of resulting frames while keeping the frame length constant. The investigation covered three computational modes of three-second spectrograms. Full-size spectrograms with 10.73 ms windows, 50% overlap and 576 frames, mid-sized spectrograms with 10.7 ms windows, 384 frames and 26.8% overlap, and finally, small spectrograms with only 192 frames, 26.5% overlap and a large window size of 20.6 ms (1024 samples).

Table 5.4.: Three-second spectrogram experimental results (median out of three trials). Reducing the frame length significantly improves the soundscape performance but only marginally affects single spectrogram predictions. Overlapping Hann windows by 50% does not improve the performance. The same applies for high input resolutions.

		SSW100 Val split		SSW100 Test split		DCSC	
NF	OL	MAP	cMAP	MAP	cMAP	F0.5	AUC
192	26.5	0.587	0.533	0.733	0.654	0.165	**0.738**
384	26.8	**0.599**	**0.545**	**0.743**	**0.671**	**0.191**	0.722
384	50.0	0.598	**0.545**	0.737	0.661	0.140	0.709
576	48.8	**0.599**	0.538	0.736	0.659	0.153	0.736

NF = Number of frames, OL = Overlap in percent

Changing the input resolution of a neural network requires to alter the topology. In order to keep most of the design consistent, I decided to adjust the receptive field of the classification branch. This way, larger input shapes provide more details to process and the number of network parameters changes accordingly—larger inputs most likely require more capacity. Additionally, larger inputs significantly increase the time per prediction. The experiments showed more than 35% longer training times per epoch for spectrograms with a high number of frames.

Investigating the results, hypothesis 3 was partially confirmed. The most notable increase in classification performance came in the soundscape domain (see Table 5.4). Shorter frame lengths improved the F0.5 measure significantly. In all other domains, scores are on par with other configurations. Very large frames (of 1024 samples) decreased the performance notably. This implies, that highly detailed spectrograms with short frames are the best choice for bird sound recognition in noisy environments. However, altering the input size of a neural network eventually leads to changes in the capacity and thus large inputs could lead to overfitting due to more model parameters. This effect cannot be entirely excluded although experiments provided fair conditions.

Frequency scaling is another important dimension of spectrogram computation. The proposed mel-like scale (see Section 2.4.2) emphasizes lower frequencies—which adapts to the avian vocal and auditory system. Still, reducing the frequency reso-

Table 5.5.: Three-second spectrogram frequency scaling experimental results (median out of three trials). More details in the frequency domain do not improve the classification performance. In fact, the decline is significant in the soundscape domain.

	SSW100 Val split		SSW100 Test split		DCSC	
MEL BINS	MAP	cMAP	MAP	cMAP	F0.5	AUC
48	0.597	**0.551**	0.724	0.660	0.168	0.713
64	**0.599**	0.545	**0.743**	**0.671**	**0.191**	**0.722**
80	0.592	0.537	0.740	0.651	0.186	0.686
96	0.582	0.522	0.735	0.662	0.154	0.718

lution of a spectrogram leads to information loss. Research suggests that temporal resolution is more important than frequency resolution for avian vocalizations (see Section 2.2.2). This assumption does not imply how an adequate frequency scaling should look like, and it remains questionable if higher frequency resolution in fact improves the overall performance. Therefore, another hypothesis can be formulated:

Hypothesis 4 *Higher frequency resolution (more than 64 mel bins) does not improve the classification performance.*

I decided to test hypothesis 4 experimentally through four different configurations that included a varying number of mel bins. Again, altering the number of frequency bins changed the input size of the DNN. During these experiments, the network architecture remained unchanged, only the classification branch was adjusted.

The results imply no significant difference in the scores (see Table 5.5). In fact, we can only observe a small performance decline with higher frequency resolution. Using 64 mel bins appears to be the best overall setting, at least in terms of soundscape performance. Smaller spectrograms almost perform on par—a finding that might help to reduce computational costs for mobile recorders in future trials. Therefore, hypothesis 4 was partially confirmed. Yet, evaluation results indicated that all tested spectrogram extraction schemes perform similar. The only truly significant change came with the increase in duration from two to three seconds per audio chunk. Other variables like the choice of mel scale, magnitude scaling or window functions were not evaluated due to their (expected) low impact factor. Three-second spectrograms with 64 mel bins and 384 (Hann windowed) frames of 10.7 ms length served as input source for all further experiments.

5.2.4. Architecture design

The next step of the evaluation process was dedicated to the investigation of different architecture designs of the neural network. Based on the initial setup, different versions of key components were tested. This included pre-processing stems, residual blocks and variations of the classification branch. Until this point, all experiments focused on the single label task of identifying one bird species per spectrogram. With real-world scenarios in mind, this constraint does no longer apply. Therefore, a multi-label training scheme was established for all further experiments. Following the idea of mixup training proposed in [Zhang et al., 2017], multi-label samples were created using two randomly added single-label spectrograms:

$$\begin{aligned} \hat{x} &= \lambda x_i + (1 - \lambda) x_j \\ \hat{y} &= y_i \vee y_j \end{aligned} \tag{5.1}$$

The scale factor λ in the range $[0.25, 0.75]$ randomly weights each of the two samples (x_i, x_j) to form a new spectrogram that contains two vocalizations. The two corresponding 'one-hot' label vectors (y_i, y_j) are combined by logical disjunction (boolean 'or') to form a multi-label vector. This process is repeated for randomly selected samples from each training batch until the number of labels per spectrogram reaches a pre-defined average. The maximum label count per sample is limited to 3.

This method of sample synthesis can be considered as data augmentation and acts as strong regularizer during training. The initial design of the DNN did not have the capacity to represent this increase in data complexity. Therefore, the first series of experiments concerned the number of filters needed to reflect changes in the input value distribution. First, the unaltered baseline network will be tested using sigmoid instead of softmax outputs, a slight decrease in performance is expected:

Hypothesis 5 *Single-label classification performance will decrease with the use of sigmoid instead of softmax activation in the classification branch.*

Table 5.6.: Multi-label experiments (median out of three trials). Synthesizing samples through random weighted addition significantly improves the classification performance due to strong regularization. Increasing the model capacity helps to map the complex input value distribution.

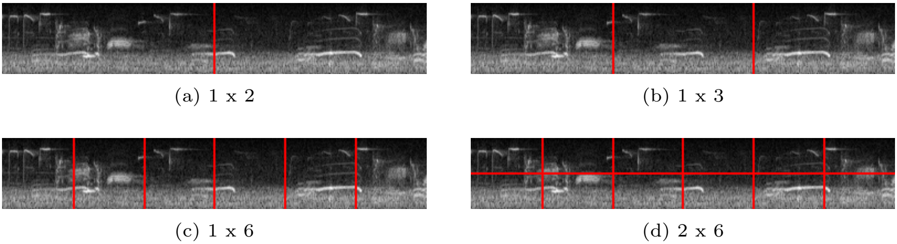
		SSW100 Val split		SSW100 Test split		DCSC	
LPS	K	MAP	cMAP	MAP	cMAP	F0.5	AUC
1.0	1	0.608	0.552	0.753	0.666	0.148	0.774
1.5	1	0.625	0.571	0.764	0.697	**0.191**	0.770
1.5	1.5	0.649	0.595	0.788	0.716	0.165	0.796
1.5	2	**0.653**	**0.604**	**0.794**	**0.721**	0.149	**0.801**

LPS = Labels per sample (average across one batch, max = 3)

Secondly, the multi-label synthesis was applied to the baseline architecture, which was expected to lead to slightly increased performance due to stronger regularization. Finally, the number of filters of the baseline architecture was raised to reflect on the increased data complexity. To achieve this, the scaling factor K proposed in [Zagoruyko and Komodakis, 2016] affected all convolutional layers from the preprocessing stem, residual stacks and classification branch with the exception of layers dedicated to reflect the number of classes. This change was expected to yield significantly better results due to a considerably higher number of trainable weights. Hypothesis 6 reflects these assumptions:

Hypothesis 6 *Multi-label classification with mixup training will increase the overall performance across all tasks.*

The results shown in Table 5.6 confirm both hypotheses. Training with augmented samples that contain one, two or three labels significantly improves the classification performance. Random weighted addition serves as strong regularization and thus affects single- and multi-labels tasks. Due to the increased complexity of the training data, an increased model capacity has notable impact when compared to the baseline model. However, the classification branch in its current form does only predict species probabilities for the entire spectrogram. Aside from regularizing effects, random weighted addition does not *per se* lead to better predictions. Short audio chunks provide a better temporal resolution and thus are capable of prediction species despite overlapping vocalizations. Changing the receptive field of the clas-



(a) 1 x 2

(b) 1 x 3

(c) 1 x 6

(d) 2 x 6

Figure 5.3.: Different class branch output shapes represent different time steps when correlating deep features to the input spectrogram. The depicted (synthesized) sample spectrogram contains three labels and is the result of random weighted addition. The final spectrogram represents a very busy acoustic scene. Dividing this scene into segments and predicting probabilities for all classes for each of those segments can help to improve multi-label classification performance.

sification branch simulates this short-chunk prediction process by passing different output shapes into the global pooling layer.

As a result, the class branch output shape represents abstract visual features that can be mapped to temporal (and frequency) steps in the input spectrogram. Due to this, we can assume that each segment of the output shape maps visual features of very short chunks of input audio (see Figure 5.3).

For a three-second spectrogram, an output shape of 1 x 6 contains predictions for each of the 100 classes every half second. With help of this arbitrary shape, we can now simulate independent predictions for short chunks of audio that provide decent resolution to grasp overlapping vocalizations. We can derive the following hypothesis from this:

Hypothesis 7 *Classification branch output shapes that represent short temporal steps improve the overall performance.*

To test this hypothesis, I decided to apply *log-mean-exponential* pooling as introduced in [Schlüter, 2018]. First mentioned by Pinheiro and Collobert, this pooling strategy aims to avoid vanishing scores that might occur due to average computation

Table 5.7.: Multi-label experiments (median out of three trials). Predicting species probabilities for short chunks of input audio increases the classification performance considerably. Three time steps (where each equals a step size of one second) appear to perform best. Smaller time steps do not increase the performance.

		SSW100 Val split		SSW100 Test split		DCSC	
COS	K	MAP	cMAP	MAP	cMAP	F0.5	AUC
(1, 1)	2	0.653	0.604	0.794	0.721	0.149	**0.801**
(1, 2)	2	0.675	0.631	0.804	0.738	0.168	0.792
(1, 3)	2	**0.679**	**0.635**	0.815	**0.751**	0.161	**0.801**
(1, 4)	2	0.674	0.630	0.813	0.746	**0.175**	0.794
(1, 6)	2	0.676	0.627	**0.817**	0.744	0.163	0.789

COS = Classification output shape (h, w), K = Filter multiplier

and preserves high confidences before passing the result through sigmoid activation [Pinheiro and Collobert, 2015]:

$$lme(y; a) = \frac{1}{a} \log \left(\frac{1}{T} \sum_{t=0}^{T-1} \exp(a \cdot y_t) \right) \tag{5.2}$$

For each time series y of local, short chunk predictions with scores $(y_0, y_1, ..., y_{T-1})$ for a single species, the sharpness factor a controls the behavior of this function. For $a \to \infty$, this function approximates the behavior of maximum pooling that only keeps the highest probability predicted for each class across all time steps. With $a \to 0$, the function approximates standard average pooling. Schlüter proposes a sharpness $a = 1$, previous experiments confirmed the suitability of this approach.

The results of the experiments with different output shapes as shown in Table 5.7 indicate that step-wise predictions affect the overall performance up until a certain threshold. This observation partially confirmed hypothesis 7. In the conducted experiments, this threshold is 3. More time steps did not help to improve the performance. Reasons for this outcome include the fact that short chunks might not include the entire vocalization in many cases. Despite the fact that overlapping sounds are problematic, one-second predictions appear to be the best choice. An additional series of experiments based on this outcome revealed that an added frequency step

Table 5.8.: Multi-label experiments (median out of three trials). Based on the assumption that three time steps perform best, some modifications were tested. Neither an added frequency step nor an increase in the sharpness factor *a* did improve the classification results. Higher sharpness seems to affect the soundscape performance—an observation that is worth considering for mobile recorders.

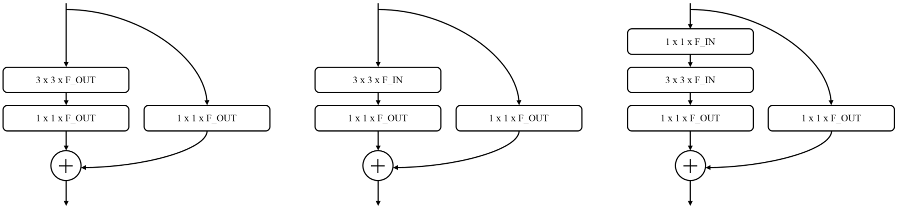
		SSW100 Val split		SSW100 Test split		DCSC	
COS	a	MAP	cMAP	MAP	cMAP	F0.5	AUC
(1, 3)	1	**0.679**	**0.635**	**0.815**	**0.751**	0.161	0.801
(2, 3)	1	0.669	0.626	0.799	0.742	0.166	**0.803**
(1, 3)	5	0.662	0.613	0.785	0.720	**0.187**	0.791

COS = Classification output shape (h, w), a = Sharpness

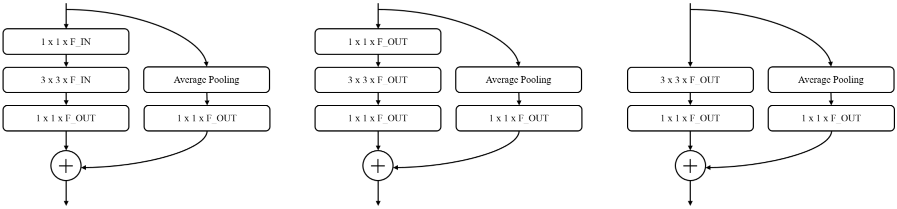
does not provide useful information (see Table 5.8). Due to the mel-like frequency scaling, vocalizations typically span the majority of the frequency band and therefore cannot be separated. Higher sharpness for the log-mean-exponential pooling function also does not improve the performance. This suggests that average-like pooling is better suited to represent the species distribution in multi-label spectrograms.

Downsample blocks

Features extracted in early layers form the foundation of a well-performing classification branch. In (almost every) classic DNN design, spatial dimensions decrease with depth, while the number of channels (filters) increases. Almost every proposed milestone architecture in the domain of visual object recognition uses such a layout. A higher number of channels comes at the cost of increased training time—something that is compensated by smaller inputs. Yet, altering spatial dimensions and filter count in a Wide ResNet architecture can introduce bottlenecks or unnecessary high computational costs. Therefore, a number of downsampling block designs have been proposed. The next series of experiments focused on the question, whether changes to the downsampling block—which handle both, spatial reduction and filter increase—can lower the computational costs while maintaining the overall performance.



(a) DS_1 (b) DS_2 (c) DS_3



(d) DS_4 (e) DS_5 (f) DS_6

Figure 5.4.: Different downsample block designs. Batch normalization and nonlinearites are omitted for clarity. Each block increases the number of filters (F_IN→F_OUT) and reduces the spatial dimensions through max pooling (strides in the original ResNet design). The baseline block (a) does not use a bottleneck layer and increases the number of filters in the 3x3 convolution. Design 2 (b) delays that increase until the 1x1 convolution. Block 3 (c) has a bottleneck layer, design 4 (d) precedes the shortcut convolution with average pooling. Designs 5 and 6 (e + f) combine previous variations.

I decided to evaluate six different designs derived from previous experience and state-of-the-art publications (Figure 5.4). Most notably, the baseline block (DS_1), which was used in past editions of BirdCLEF, is a simplified version of the original design (DS_3). ResNet tweaks recently proposed in [Xie et al., 2018] alter the original design to a more sophisticated layout (DS_4). Other tested versions contain single alterations of these designs. The results shown in Table 5.9 indicate that differences between downsample block variations are only minor. Surprisingly, the initial baseline design performed very well, even when compared to more recent approaches. Training times differed significantly and the main constraint for

Table 5.9.: Investigation of downsample block designs (median out of three trials). Only minor differences exist between the performance of different downsample block designs. Version 4 appears to be the best compromise of training speed and overall scores.

		SSW100 Val split		SSW100 Test split		DCSC	
D_ID	TPE	MAP	cMAP	MAP	cMAP	F0.5	AUC
DS_1	87	0.679	0.635	**0.815**	**0.751**	0.161	0.801
DS_2	**77**	0.662	0.614	0.801	0.730	**0.178**	0.799
DS_3	87	0.667	0.619	0.806	0.735	0.163	0.798
DS_4	79	0.676	0.633	0.807	0.733	0.165	**0.824**
DS_5	101	**0.684**	**0.643**	0.814	0.748	0.150	0.822
DS_6	79	0.676	0.637	0.808	0.745	0.151	0.795

D_ID = Design id, TPE = Time per epoch in seconds

the eventual selection reflected that. Although layout DS_5 performed best on the single spectrogram task, its increase in training duration renders it a non-optimal choice. Overall, DS_4 as mentioned by Xie et al. in their 'ResNET bag of tricks' appears to perform best considering time per epoch and overall scores. However, other training settings could increase the performance of a design when specifically adapted. Nonetheless, layout DS_4 was used for all further experiments. The layout of the regular residual block remained unchanged and reflected the wide dropout design proposed as best performing design in [Zagoruyko and Komodakis, 2016].

Pre-processing

Large receptive fields in early layers of a DNN have proven to be effective in the past. AlexNet used 11x11 kernels, ZFNet used 7x7 filters. The GoogleNet design used 5x5 kernels as largest filters but kept the 7x7 convolution as first layer. VGG-16 introduced stacked 3x3 convolutions to reduce the number of parameters while maintaining the size of the receptive field. Especially for tasks that involve high resolution inputs (like photographs) or require high resolution outputs (like segmentation masks), large receptive fields tend to increase the overall performance [Yu and Koltun, 2015]. However, it remains questionable if that also applies for

Table 5.10.: Investigation of pre-processing kernel sizes (median out of three trials). Large kernel sizes do not notably slow down training due to only one input channel. A 5x5 kernel size appears to perform best. Stacked 3x3 convolutions significantly increase the training time and slightly decrease the performance.

		SSW100 Val split		SSW100 Test split		DCSC	
KS	TPE	MAP	cMAP	MAP	cMAP	F0.5	AUC
3x3	**79**	0.676	0.633	0.807	0.733	0.165	**0.824**
5x5	**79**	**0.677**	**0.634**	**0.811**	**0.744**	0.176	0.816
9x9	80	0.664	0.619	0.795	0.735	0.164	0.816
7x7	**79**	0.661	0.615	0.797	0.734	0.158	0.821
5x5*	93	0.668	0.626	0.809	0.739	**0.179**	0.805

*KS = Kernel size, TPE = Time per epoch in seconds, *stacked 3x3 convolutions*

very dense, extremely specialized visual representations—such as spectrograms. I decided to test the following hypothesis in another series of experiments:

Hypothesis 8 *A large receptive field of the pre-processing stem improves the overall performance.*

First, different kernel sizes in the first layer of the pre-pocessing stem were tested to examine whether larger kernels actually improve the overall performance. According to previous advances and the fact that almost every very deep network design features stacked convolutions in its stem to resemble large kernels, the expected results should show better scores for larger filters. However, the experimental investigation revealed that this is not the case when spectrograms are used an input (Table 5.10). A kernel size of 5x5 consistently improved the performance compared to the initial 3x3 convolution, but larger filter sizes decreased classification scores significantly. Therefore, hypothesis 8 can only be partially confirmed.

Stacking 3x3 convolutions to achieve the same receptive field with less parameters is a common design scheme. The entire Inception architecture is built on the assumption that deeper networks profit from replacing costly convolutional operation with more effective successions of stacked layers [Szegedy et al., 2015]. Replacing a 5x5 convolution with two (stacked) 3x3 convolutions reduces the number of network parameters while increasing the depth. Yet, except for a slight increase in soundscape performance, this design choice did not yield better results than a single 5x5 convolution but significantly raised the time needed to train one epoch.

According to the results of the architectural investigations, all further experiments used 5x5 kernels in the pre-processing stem, DS_4 downsample block designs, regular Wide ResNet residual blocks, and three time steps with log-mean-exponential pooling in the classification branch. One of the most important insights of the conducted series of experiments revealed that all tested architectural changes perform similar with no version that significantly outperforms all other choices across all tested domains. This implies that changes to the training regime (which was fixed so far) might have a greater impact on the overall performance.

5.2.5. Topologies and training regimes

With the main DNN architecture established, the next investigation focused on network topologies and training regimes. Until now, hyperparameters like learning rate, batch size, or optimizer remained unchanged to provide fair evaluation. Additionally, no data augmentation was used. In this section, I will explore the results of deep and wide network topologies, different augmentation methods, as well as some variations of essential training hyperparameters.

This can be considered a critical stage of the experimental process since the sequence of the conducted tests might influence the outcome. To be more specific, testing deeper or wider networks without augmentation might lead to overfitting and non-conclusive results. However, the full potential of some augmentation methods might not be visible when trained on a shallow and narrow topology. I decided to evaluate different augmentation methods first, so that very powerful topologies do not suffer from overfitting. Task-specific augmentation is key to good overall scores and the methods that I evaluated experimentally reflect changes in frequency, time and magnitude (see Figure 4.3 in Section 4.2.2). The current DNN layout supposedly does not have the capacity to map the increased input distribution when samples are augmented. Therefore, the scale factor K was set from 2 to 2.5 in order to add more filters and thus more weights to the network. To ensure that this setting would lead to overfitting without any further regularization, another baseline experiment was conducted.

Table 5.11.: Investigation of augmentation methods (median out of three trials). The results indicate that noise samples are serving to adapt the acoustic domain, vertical stretch best simulates frequency shifts in bird song, and horizontal roll significantly diversifies the training sample selection.

	SSW100 Val split		SSW100 Test split		DCSC	
METHOD	MAP	cMAP	MAP	cMAP	F0.5	AUC
none	0.668	0.626	0.794	0.731	0.145	0.809
v_roll	0.683	0.644	**0.820**	0.752	0.182	0.792
h_roll	**0.690**	**0.651**	0.815	0.750	0.152	0.808
v_stretch	0.684	0.646	0.818	0.753	0.185	0.809
h_stretch	0.681	0.643	0.816	0.743	0.140	0.791
f_dropout	0.674	0.634	0.818	0.761	0.161	0.802
t_dropout	0.663	0.619	0.786	0.723	0.161	0.818
warp	0.677	0.636	0.815	0.750	0.154	**0.830**
noise	0.680	0.640	0.815	**0.763**	**0.224**	0.807

The results shown in Table 5.11 indeed suggest that scaling the width caused the DNN with increased capacity to overfit. The scores across all tasks dropped notably compared to previous trials. It remains unclear whether the capacity was sufficient to provide fair conditions for all augmentation methods that acted as unequally strong regularization. However, the effects that could be observed were still very conclusive. Only one augmentation method decreased the overall scores, the majority of approaches led to significantly better classification results.

One particular interesting observation implies that time-domain dropout decreases scores, while frequency domain dropout increases the performance. This very likely reflects the information density in both domains: Information along the frequency axis is somewhat redundant while the temporal resolution of the used spectrograms contains very dense data points. This supports the assumption that the selected computational approach of spectrogram generation is ideal in terms of available detail and resulting input resolution. Experiments with different numbers of mel bins (Table 5.5) revealed comparable performance when dropping 16 frequency bins.

The slightly reduced information density in the frequency domain allowed to simulate frequency shifts of vocalizing birds in different habitats through augmentation.

Those methods—especially vertical stretch—proved to be very effective. Horizontal roll preserves all the information in the time domain and still increases the input sample diversity—with notable effect. Additional noise samples are one of the most powerful augmentation methods, which was expected considering previous experience. This leads to the conclusion that vertical stretch best simulates shifts in pitch, horizontal roll emulates different sample selection strategies and noise samples cover the domain shift between mono-species recordings and soundscapes. I selected these three methods to serve as augmentation for all further experiments. During those, each method had a 50% chance to be selected as data augmentation that resulted in samples that contain none, one, two, or all three (with 12.5% probability) augmentations.

Depth vs. width

With strong regularization methods added, the next series of experiments concerned the omnipresent question whether deep networks provide better performance than wide topologies. The implemented ResNet architecture with its two scale factors K and N allows to easily tune both dimensions. Yet, the current baseline network was derived from the wide residual network design and thus supposedly favours wide but shallow layouts. Twelve different topologies were assessed (Table 5.12) to test the commonly formulated hypothesis:

Hypothesis 9 *Deeper topologies (more layers) perform better than wider topologies (more filters).*

Since the added data augmentation significantly increases the variance of the input value distribution, networks with high capacity (number of parameters) were expected to perform better. However, those topologies also needed significantly longer training durations and thus might not be worth the improved performance when the difference is too small.

The observable results (see Table 5.13) demonstrate that effect clearly. More capacity and thus longer, costly training only improves the performance until a certain threshold. The choice of the network topology has to reflect this constraint. Yet, the outcome of the investigation also shows that wider and deeper topologies do outperform the baseline design. In fact, the basic assumption derived by Zagoruyko et al. in their Wide ResNet paper [Zagoruyko and Komodakis, 2016] was exactly

Table 5.12.: Different network topologies.

ID	ResNet_K	ResNet_N	PARAMS	LAYERS	TPE
1	2	2	5,037,492	28 (121)	94
2	2	3	6,608,052	36 (157)	109
3	2	4	8,178,612	44 (193)	124
4	2	5	9,749,172	52 (229)	139
5	3	2	11,285,212	28 (121)	123
6	3	3	14,816,092	36 (157)	160
7	3	4	18,346,972	44 (193)	196
8	3	5	21,877,852	52 (229)	230
9	4	2	20,018,948	28 (121)	157
10	4	3	26,293,508	36 (157)	204
11	4	4	32,568,068	44 (193)	253
12	4	5	38,842,628	52 (229)	301

TPE = Time per epoch in seconds

confirmed: Increasing the scaling factor K and thus increasing the number of filters per layer does consistently improve the classification results independent of the depth of the network. However, the soundscape performance appears to be entirely linked to the depth of the DNN. This implies that Hypothesis 9 can be confirmed. The performance across all other tasks appears to be solely linked to the capacity of the network and significantly improves with higher numbers of parameters.

Although deeper topologies outperform shallow ones, there appears to be a limit until which depth actually benefits the experimental outcome. A depth of three blocks per residual stack consistently outperforms shallow stacks with only two residual blocks. After that, more depth increases the classification performance but not in every case—which unfortunately is a bit inconsistent to be reliant. However, this indicates that the Wide ResNet design effectively compensates a lack of depth with wider layers—which is exactly what was intended. Deeper topologies do not contribute to an increased performance for single label and mono-species recording tasks, but both are of high importance.

Considering this outcome, the choice of topology that was used in all further experiments had to reflect the overall task performance and the required computational

Table 5.13.: Investigation of network topologies (median out of three trials). More filters (and thus higher capacity) lead to consistently better results independent from the depth. The soundscape performance strongly correlates with the depth of the topology. Computationally expensive topologies do not necessarily provide better results.

	SSW100 Val split		SSW100 Test split		DCSC	
ID	MAP	cMAP	MAP	cMAP	F0.5	AUC
1	0.687	0.649	0.823	0.765	0.186	0.800
2	0.678	0.638	0.823	0.760	0.231	0.776
3	0.690	0.659	0.830	0.770	0.211	0.784
4	0.690	0.653	0.832	0.768	0.250	0.746
5	0.694	0.657	0.833	0.777	0.189	**0.815**
6	0.691	0.659	0.825	0.773	0.231	0.775
7	0.699	0.664	0.838	0.778	0.230	0.755
8	0.698	0.664	0.839	0.780	0.245	0.742
9	0.702	**0.671**	0.839	0.783	0.177	0.785
10	0.702	0.667	0.838	0.779	**0.265**	0.758
11	**0.703**	0.670	0.839	**0.786**	0.225	0.762
12	0.700	0.668	**0.840**	0.777	0.238	0.74

costs. Therefore, topology 10 appeared to be the best compromise. With a width of four and a depth of three, it achieved the best soundscape performance (the AUC score can be raised with a different choice of confidence threshold and is thus omitted in the decision process). It also closely matched the top performance across other tasks. With a training time of 204 seconds per epoch, the computational costs are acceptable, the network capacity leaves some room for more regularization, which will come with the use of more samples and more classes.

Dropout

Wide residual networks benefit from width because of additional regularization in the form of random dropout. This regularization method is widely used and powerful. Hinton and Srivastava argue that dropout consistently improves the performance of deep neural networks—an observation that is backed by countless publications in the field [Hinton et al., 2012b], [Srivastava et al., 2014]. 'Standard' dropout prevents activations from becoming strongly correlated, which would lead to overfitting. To counter this, single activations are dropped (zeroed) randomly with a certain probability at training time. Tompson et al. argue that spatial dropout—which drops entire channels instead of single neurons—leads to better regularization when spatial features exhibit strong correlation [Tompson et al., 2015]. For images, this is the case, the same might apply for spectrograms. The next series of experiments focuses on this hypothesis:

Hypothesis 10 *Spatial dropout improves the overall performance through better regularization for spatially correlated inputs.*

The effectiveness of dropout regularization is linked to the overall capacity of the network and the probability to drop activations or channels. Since both methods might require different dropout probabilities, a series of settings was tested (Table 5.14).

The results however are not conclusive. Spatial dropout significantly increases the recognition performance in the soundscape domain, but the performance drops considerably in all other tasks. This outcome only partially confirms Hypothesis 10. It appears that low dropout probabilities consistently perform worse in both test scenarios for mono-species recordings and soundscapes. Aside from that, evidence does not suggest that one method is far superior, but dropout in general increases the performance of a Wide ResNet (which confirms another assumption by the authors of the original proposal). However, spatial dropout requires more epochs to complete the training process due to stronger regularization. Considering this, the initial setting of 50% random dropout appeared to be the best choice for future experiments that use more samples to leave some capacity for more diverse input data.

Learning rate decay

For the sake of faster experimentation, I decided to use the ADAM optimizer, which converges considerably faster and automatically adapts the learning rate for each weight individually [Kingma and Ba, 2014]. Although it might not converge towards the optimal solution [Wilson et al., 2017], it eases the choice of the initial learning

Table 5.14.: Investigation of dropout regularization (median out of three trials). Spatial dropout achieves better scores in the soundscape domain but diminishes the performance across all other tasks. Random dropout with 50% probability per activation appears to be the best choice.

		SSW100 Val split		SSW100 Test split		DCSC	
TYPE	P	MAP	cMAP	MAP	cMAP	F0.5	AUC
R	0.125	**0.704**	**0.668**	0.837	0.776	0.207	0.742
R	0.250	0.700	0.667	0.835	**0.783**	0.261	**0.763**
R	0.500	0.702	0.667	**0.838**	0.779	0.265	0.758
S	0.125	0.693	0.656	0.828	0.774	0.256	0.742
S	0.250	0.693	0.660	0.832	0.772	0.273	0.729
S	0.500	0.686	0.647	0.827	0.768	**0.277**	0.733

P = Dropout probability, R = Random dropout, S = Spatial dropout

rate—a critical setting for fast training. Specifying an initial learning rate sets an upper bound for the adaptive weight updates for ADAM. Most optimizers benefit from a learning rate schedule that incrementally decays the learning rate while training. This might as well apply for ADAM. All previous experiments featured a constant learning rate, the next series tested commonly used schedules. In its simplest form, learning rate decay linearly reduces the update step size between a starting and end value. The most popular form of learning rate decay however is a step-wise reduction. The learning rate is multiplied with a pre-defined factor—usually 0.1 per step—at certain points during training. With that technique, steps need to be placed whenever the loss flattens—something that might be challenging to do. Adaptive detection of flattening loss or continuous decay could be better choices. Of the continuous scheduling methods, two variations stand out: Kingma and Ba used square root scheduling in their original proposal of the ADAM algorithm, Loshchilov and Hutter proposed an aggressive schedule that uses cosine annealing to decay the learning rate towards zero [Loshchilov and Hutter, 2016].

The results however indicate that the DNN is overfitting to the metric of a particular task due to delayed early stopping (see Table 5.15). The training loss decreases considerably more than the validation loss, which only helps two of the three tasks, the overall soundscape performance suffers. Therefore, a learning rate schedule should be applied for specific tasks only and was not employed during further experiments.

Table 5.15.: Investigation of learning rate schedules (median out of three trials). Continuous decay requires knowledge of the expected training progress. An adaptive schedule avoids that by adjusting the learning rate whenever the validation loss flattens for three epochs. However, soundscape performance appears to suffer significantly, which could be the result of overfitting due to delayed early stopping

	SSW100 Val split		SSW100 Test split		DCSC	
TYPE	MAP	cMAP	MAP	cMAP	F0.5	AUC
Constant	0.702	0.667	0.838	0.779	**0.265**	0.758
Linear	0.704	0.670	0.837	0.782	0.239	**0.781**
Step	0.707	0.672	0.848	0.785	0.234	0.772
Adaptive	**0.717**	**0.685**	**0.851**	**0.797**	0.218	0.767
Root	0.709	0.674	0.836	0.778	0.195	0.773
Cosine	0.712	0.679	0.841	0.785	0.221	0.766

However, of all learning rate decay schedules, the adaptive, step-wise version appears to outperform all other methods as it scales the learning rate by a factor of 0.5 whenever the validation loss flattens for three epochs and thus does not require manual adjustment. We can conclude that continuous decay is only of use when we know how well the model actually performs during training. The result of that would be an increased amount of experimentation—which is not very practical. Still, I used the adaptive scheduling method for model fine-tuning after training with a constant learning rate to boost the performance of the benchmark system.

Cost-sensitive learning

Modern deep learning frameworks with high-level API like Keras allow to address class imbalances in the training data. In reality, most classes of a datasets will consist of different numbers of samples. Basically, two strategies exist to counter class imbalances: Data-level approaches and algorithm-level methods. Addressing imbalanced amounts of samples per class on data level usually implies that classes have to be over- or undersampled in order to restore balance. Both methods are very popular and have proven to be effective for deep neural networks in the image recognition domain [Buda et al., 2018]. On an algorithmic level, class imbalances

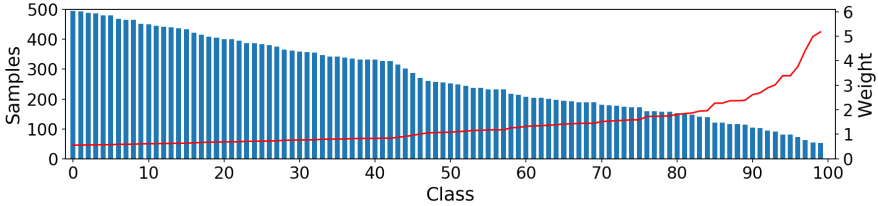


Figure 5.5.: Class weights based on number of samples as in Equation 5.3. The data splits used to investigate certain balancing strategies contained a random amount of samples per class (blue bars). Class weights (red line) multiply the loss for each class accordingly—rare classes gain more weight and force the model to optimize towards underrepresented classes.

can be addressed with penalties (costs) that serve as an addition to the employed loss function. A number of weight functions have been proposed and most of them resemble the share of a class in relation to the entire dataset. One of the simplest forms of class weights can be derived from

$$W_i = \frac{N}{C \cdot S_i} \tag{5.3}$$

where N is the total number of samples, C is the total number of classes and S_i specifies the number of samples for a specific class i. When all classes have an equal amount of samples, every class weight is 1. Imbalanced datasets reflect class probabilities based on sample count (see Figure 5.5). The above equation implements the ‘balanced’ mode of class weight computation of scikit-learn—a widely used method.

The vector of class weights can be applied to a DNN by adding an additional layer [Khan et al., 2017] or by altering the error measure. Commonly, the cost-sensitive penalty is applied to the loss function, e.g. by multiplying it with the vector of all class weights as implemented in TensorFlow. Aside from that, we can also change the loss function to reflect class imbalances. One of the most prominent examples is the *focal loss* proposed in [Lin et al., 2017]. Focal loss does not only reflect class imbalances, it also emphasizes samples that are hard to learn. For bird vocalizations, this sounds very promising since some birds have a high intra-species heterogeneity. However, this heterogeneity cannot be easily measured and it is thus very complex to find a quantifiable weight that could serve as penalty. The focal loss function shifts the attention to samples that were falsely classified during the previous training step and thus automatically establishes a difficulty measure.

Focal loss is typically aimed at datasets with extreme imbalances and contributes very effectively to improving the detection of rare events. In my experiments, I used the non-α balanced version proposed by Lin et al. with

$$FL(y) = -t(1-y)^{\gamma}\log(y) - (1-t)y^{\gamma}\log(1-y) \tag{5.4}$$

for targets t and predictions y. The added modulating factor $(1-y)^{\gamma}$ and the focusing parameter γ allow to down-weight (easy) correct classifications. Although this method favors underrepresented classes and hard-to-train samples, it might amplify the focus on unrelated noise since not all samples do actually contain a valid bird vocalization. Adjusting the valaue for γ allows to weaken the effect of the focal loss. The initial design proposed by Lin et al. was intended for extreme imbalances of 1:1000. Pre-tests revealed that a γ-value of 0.25 performs best for the current use case and was therefore chosen for this investigation.

The unbalanced dataset for this series of experiments was simulated by randomly limiting the number of samples per class to a (uniformly chosen) value between 50 and 500. For comparison, a run without any balancing method was added as the resulting dataset contains only half as many samples as previous, balanced collections. The validation and test data remained unaltered. The most interesting metric is the class-wise mean average precision. As balanced measure of classification performance, it reflects the overall performance across all classes independent of their amount of samples or difficulty level (vocal diversity). Therefore, we can derive the following hypothesis:

Hypothesis 11 *Cost-sensitive learning methods—that adapt to class imbalances in the dataset—increase the performance despite underrepresented classes.*

This hypothesis can be partially confirmed based on the results (Table 5.16). However, of all the investigated methods, only oversampling appears to have a significant effect. Additionally, some methods only increase the performance for one task.

Table 5.16.: Investigation of data balancing and cost-sensitive learning methods (median out of three trials). Oversampling slightly increases the performance for mono-species recordings, class weights significantly increase the soundscape performance. Focal loss does not help to improve the performance for unbalanced data. It drastically decreases the soundscape performance due to emphasis on unrelated noise in the training samples. Focal loss does slightly improve the results for balanced datasets but even then drastically reduces the soundscape performance.

		SSW100 Val split		SSW100 Test split		DCSC	
DATA	METHOD	MAP	cMAP	MAP	cMAP	F0.5	AUC
U	None	0.658	**0.618**	0.789	0.739	0.177	0.791
U	OS	**0.659**	0.611	**0.814**	**0.747**	0.180	0.767
U	OS/US	0.645	0.591	0.796	0.735	0.177	0.780
U	Weights	0.635	0.593	0.772	0.726	**0.216**	0.784
U	FL	0.656	0.617	0.781	0.732	0.164	**0.81**
B	None	0.702	0.667	**0.838**	0.779	**0.265**	0.758
B	FL	**0.707**	**0.673**	0.834	**0.783**	0.179	**0.767**

U = Unbalanced, B = Balanced, OS = Oversampling, US = Undersampling
FL = Focal loss

Class weights are best for soundscape performance, focal loss drastically decreases the scores in that domain. In fact, the focal loss trials suffer from overfitting to unrelated noise in the training data. Non-events that contain a bird species label are highly problematic for this kind of penalty. While focal loss slightly improves the performance for balanced data splits, it consistently produces a high number of false positives that do not contain a bird vocalization in soundscape recordings. Unfortunately, this outcome is anti-climatic. Established and very basic methods of dataset balancing outperform sophisticated approaches by a significant margin. Yet, as part of the final model fine-tuning, cost-sensitive penalties might lead to improved, task-specific performance and could be worth the application.

In conclusion, the experimental evaluation of topologies and training hyperparameters revealed that data augmentation has great impact on the overall performance. Sample diversity in all three spectrogram domains (time, frequency, magnitude) effectively improves the classification results through adaption to the target domain.

Very deep topologies outperform shallow designs by a significant margin. However, the baseline architecture that follows the wide residual network design of Zagoruyko and Komodakis benefits from increased width and dropout regularization. Learning rate decay and cost-sensitive learning provide task-specific improvements and were applied during fine-tuning of the final benchmark system (see Section 5.2.7).

5.2.6. Mobile architectures

Mobile DNN architectures are typically very limited by computational resources. Additionally, soundscape analysis relies on real-time capabilities that require DNN to process a spectrogram in less time than this spectrograms represents (e.g. a three-second spectrogram has to be processed in less than three seconds). For better temporal resolution, overlapping spectrograms are often desirable. In our scenario, a three-second spectrogram should be processed in less than two seconds to allow a one-second overlap. The target platform for all experiments in this section is a Raspberry Pi 3 A+[1]. On this device, the proposed workflow of audio processing and spectrogram extraction requires 250 milliseconds. Therefore, the maximum (practical) amount of time for one DNN prediction is 1,750 ms.

The first mobile architecture for examination was a traditional AlexNet-like design with 8 layers and simply stacked 3x3 convolutions (Table 5.17). The pre-processing stem and classification branch were exactly the same as in previous experiments since their design has been experimentally validated. The overall network layout is simple but provides sufficient capacity combined with real-time execution on a Raspberry Pi. Theano and Lasagne are not optimized for ARM architectures and simply run in CPU mode. Other frameworks like TensorFlow Lite are specifically designed for the Raspberry Pi and might provide better performance. For this series of experiments, these circumstances were omitted.

Of all tested versions, a scaling factor K of 3 appears to perform best (see Table 5.18). All tested variations are real-time capable by definition, but $K = 4$ is not practical when an overlap of one second should be achieved. The results show that more parameters do not increase the performance beyond a certain threshold. Shallow, AlexNet-like architectures are significantly inferior to fully optimized wide residual networks, even when compared to variations with the same amount of trainable weights (see Table 5.13; with $K = 4$ the network has 12,690,480 parameters).

---

[1] see https://en.wikipedia.org/wiki/Raspberry_Pi for more details and hardware specs

Table 5.17.: AlexNet-like mobile topology based on previous design decisions. Shallow designs have proven to be effective in the past. With less than 1 million parameters, the 8-layers model is real-time capable when applied to a Raspberry Pi.

Group	Name	Input shape	Output shape
Pre-processing	Conv + BN + ReLu	(1x64x384)	(8x64x192)
Conv 1	Conv + BN + ReLu Dropout	(8x64x192)	(16x32x96)
Conv 2	Conv + BN + ReLu Dropout	(16x32x96)	(32x16x48)
Conv 3	Conv + BN + ReLu Dropout	(32x16x48)	(64x8x24)
Conv 4	Conv + BN + ReLu Dropout	(64x8x24)	(128x4x12)
Classification	Conv + BN + ReLu	(128x4x12)	(128x1x1)
	Conv + BN + ReLu	(128x1x1)	(256x1x1)
	Conv + BN	(256x1x1)	(100x1x1)
	Global pooling	(100x1x1)	(100x1)
	Sigmoid	(100x1)	(100x1)

Total number of layers: 33 (8 weighted), Parameters: 814,776

However, the soundscape performance is surprisingly competitive, which might be due to the (sometimes inaptly) fixed confidence threshold but mainly implies that the employed training regime is very effective independent of the underlying DNN architecture. Considering computational limitations and extremely fast execution on a ARM CPU, the performance is still very competitive.

A central question that arises from the observed scores is whether deeper (residual) architectures perform even better. More layers require more computational resources, but some designs are still real-time capable due to some specific design choices. In the past, different mobile model architectures that achieve competitive results in the domain of visual object recognition have been proposed (most notably in [Howard et al., 2017], [Sandler et al., 2018], and [Tan and Le, 2019]). Based on residual blocks and their alterations, mobile networks employ grouped convolutions that parallelize feature extraction pathways by limiting the amount of channels that

Table 5.18.: Investigation of mobile AlexNet-like topologies (median out of three trials). A high number of filters per layer and thus more effective capacity significantly improves the performance. Surprisingly, the regularizing effect of insufficient capacity (underfitting) appears to increase the soundscape performance. Considering the inexpensive model design, the results are very competitive across all tasks.

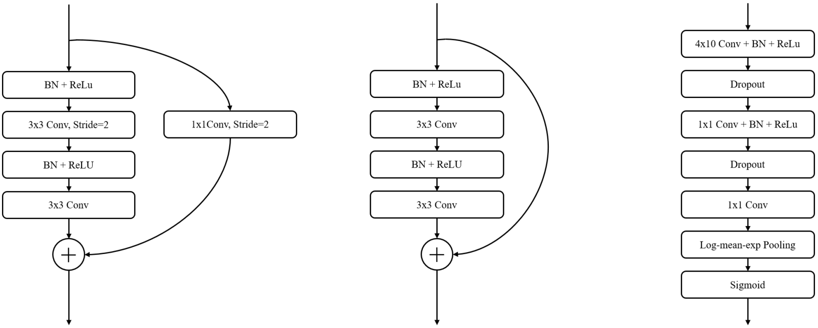
		SSW100 Val split		SSW100 Test split		DCSC	
K	TPP	MAP	cMAP	MAP	cMAP	F0.5	AUC
1	0.49	0.610	0.535	0.767	0.677	0.208	0.753
2	1.02	0.653	0.592	0.811	0.742	**0.254**	0.752
3	1.78	**0.657**	**0.595**	0.811	0.739	0.210	**0.768**
4	2.92	0.654	0.591	**0.814**	**0.745**	0.207	0.748

TPP = Time per prediction in seconds (incl. audio processing)

each convolution processes. In the most extreme case, the amount of convolutional groups matches the number of incoming channels, which is called *depthwise separable convolution* [Kaiser et al., 2017]. An increased number of groups significantly reduces the computational costs of a model while mostly maintaining the overall performance.

The design of residual blocks and stacks for the next series of experiments strongly resembled the classical (but ReLU pre-activated) design without any further changes [He et al., 2016b, Fig. 1b]. The pre-processing stage and classification branch remained unchanged and incorporated experimentally proven design decisions (see Figure 5.6). The architecture uses strided convolutions instead of max pooling to spatially downsize inputs. This change was expected to yield slightly worse results but significantly faster processing. Again, the scaling factors K and N were used to increase the number of residual blocks per stack and the number of filters per layer. In the most basic form (with $N = 1$), this layout already has 16 weighted layers, which is a considerable increase compared to shallow, AlexNet-like designs of previous experiments.

As observed in Section 5.2.5, the expected results should show an increase in overall performance (especially for soundscapes) for deeper topologies. Two hypotheses can be derived from the aforementioned considerations:



(a) Downsample block (b) Basic residual block (c) Classification branch

Figure 5.6.: Baseline design of mobile ResNet components. Residual block follow the original (ReLU pre-activated) design of He et al. Pre-processing stem and classification branch incorporate experimentally derived design decisions. Instead of max pooling, strided convolutions are used to spatially downsize incoming features.

Hypothesis 12 *Residual neural network designs outperform shallow AlexNet-like architectures.*

Hypothesis 13 *Deeper topologies (with more convolutional layers) outperform shallow layouts, even when computational resources are limited.*

In order to preserve real-time capabilities with one-second overlap, the processing of a three-second chunk of audio has to be finished in under two seconds. Whenever the execution time exceeded this limit, grouped convolutions were used to reduce the computational costs. Since convolutional groups were expected to yield slightly worse results, the number of groups was kept as low as possible.

The experimental results of this investigation shown in Table 5.19 strongly confirm hypothesis 12. Residual model designs outperform shallow architectures across all tasks. This applies even when the number of parameters—and thus the network capacity—is comparable. Residual layouts are more flexible and provide more degrees of freedom to adjust execution time and performance. Depth plays a crucial role but does not lead to better results without an increase in capacity. This only

Table 5.19.: Investigation of mobile ResNet topologies (median out of three trials). Residual neural networks once again outperform AlexNet-like layouts. In contrast to previous investigations however, depth does only increase the overall performance when the number of parameters is raised accordingly. When computational resources are limited, wider topologies outperform deeper designs. Grouped convolutions consistently reduce the classification scores.

			SSW100 Val split		SSW100 Test split		DCSC	
K	N	TPP	MAP	cMAP	MAP	cMAP	F0.5	AUC
1	1	0.80	0.638	0.578	0.781	0.714	0.238	0.721
1	2	1.36	0.654	0.609	0.813	0.747	0.240	0.728
1	3	1.87	0.659	0.610	0.808	0.735	0.242	0.740
1	4	2.42	0.651	0.606	0.800	0.736	0.244	0.747
1*	5	1.68	0.640	0.586	0.781	0.714	0.235	0.765
1.5	1	1.39	0.656	0.602	0.809	0.746	**0.246**	0.756
1.5	2	2.64	0.671	0.626	0.824	0.764	0.245	0.762
1.5*	3	1.46	0.638	0.581	0.782	0.724	0.226	0.764
2	1	2.08	**0.675**	**0.630**	**0.826**	**0.768**	0.244	**0.769**
3*	1	1.99	0.665	0.611	0.811	0.752	0.243	0.790

TPP = Time per prediction in seconds (incl. audio processing)

** Used grouped convolutions*

partially confirms hypothesis 13. With constant scaling factor K (and thus a constant number of filters per layer), an increased number of layers and thus more depth does not yield better results when grouped convolutions have to be applied to reduce execution times. The lack of capacity diminishes the performance of more layers.

The same observation was consistently made in other trials. Whenever grouped convolutions are needed, the performance drops independent of the depth of the model. This circumstance also limited the performance of wider designs rendering convolutional groups a non-optimal choice for the target use case. Overall, a depth of $N = 1$ and width of $K = 2$ appears to perform best. The execution times of this setting are still acceptable and only slightly exceed the previously set limit.

In conclusion, mobile network design perform well on a variety of sound recognition tasks. Despite limited computational resources, the proposed architectures achieve competitive scores with only slightly decreased performance compared to previous investigations. Three main insights can be derived from that: First, residual network designs reach top performance and can be comfortably adjusted to a variety of tasks and target platforms. Secondly, higher scores come at the cost of significantly increased execution times. Efficient models that preserve real-time capability on (semi-) mobile devices obtain competitive scores that can only be outperformed by extremely costly network designs. Thirdly, the established training regime—especially the proposed augmentation method—appears to highly impact the overall performance. Changes to the employed training regime significantly affect the classification results and consistently provide better scores than any architecture alteration. This observation is in line with recent advances made during the BirdCLEF challenge and implies that a number of different model architectures perform equally good when training parameters are chosen carefully.

5.2.7. Benchmark system

During the training of the final benchmark system, the aforementioned techniques, architectures and topologies were combined to learn representations of nearly 1,000 bird species. According to the experimental results in Table 5.13, scaling factors of $K = 4$ and $N = 3$ appeared to be the best choice for the benchmark model. A number of training iterations was conducted to estimate the influence of different amounts of samples on the overall performance. First, the model was trained with 100 samples per class for a few epochs to initialize the network. After that, the number of samples was steadily increased after convergence. Previously trained snapshots were used to initialize succeeding trials. Pre-training models allows to transfer knowledge that has already been learned onto the new task. This scheme drastically reduces the time needed for training. With each new start of a training process, the learning rate and dropout probability were reduced but kept constant across all epochs. In order to counter class imbalances, slight oversampling was employed. Samples of classes that did not contain enough training spectrograms were repeatedly added to the dataset until the number of samples reached 10% of the desired amount. This way, underrepresented classes did gain more weight during the training but did not benefit overfitting due to excessive repetition of samples. Three non-event classes were added to the dataset: *Human*, *Non-Bird* and

Noise. To avoid future classification of non-bird sounds and ambient noise, the label vectors of those two classes only contained zeros. This way, the model was forced to suppress high class probabilities for non-event samples. Human vocal sounds are an important dimension for public demonstrators and were thus part of the training and test data.

With a random selection of max. 3,000 samples per class, the entire training data (BN1000, see Section 5.2.1) split contained 1,727,234 three-second spectrograms. The validation data was kept mostly balanced to avoid biased error measures. In total, the BN1000 validation split contained 87,764 samples (max. 100 per class). Mono-species recordings were used to evaluate the overall model performance on a number of unseen audio files. Again, the test data was kept mostly balanced and contained a total amount of 2,868 recordings (max. three per class). 24 dawn chorus recordings were used to examine the soundscape performance of each model. For the sake of better comparability, this portion of the test data remained unchanged.

The effectiveness of a DNN can be estimated by limiting the amount of training samples. For future applications, it is desirable to avoid the need for extremely high sample counts per class due to the potential lack of recordings for rare or endangered species. With 1,000 samples per class (3,000 seconds of audio for three-second spectrograms), the proposed network design already achieves competitive scores (see Table 5.20). Adding more samples to the training data slightly improves the scores but eventually decreases the performance due to significant class imbalances. The most common bird species are represented by a vast amount of recordings and high number of annotated vocalizations in the soundscape data—which is therefore highly biased. Yet, for real-world use cases, the same imbalance will occur when monitoring avian diversity without the focus on endangered or rare species. Models trained on imbalanced data hold applicable value due to this circumstance.

Born-again networks do not outperform their teacher models in the mono-species task but significantly increase the soundscape performance when the confidence threshold is adjusted (model 4 was chosen due to the highest soundscape scores, see Table 5.21). Interestingly, 'soft' labels appear to increase the ability to grasp uncertain or faint vocalizations. The born-again student model outperforms all other networks across almost every metric in the soundscape domain.

Soundscape performance is one of the most important aspects when considering real-world monitoring scenarios. In fact, the single-model performance has to be

Table 5.20.: Benchmark experiments for 984 species, 87,764 validation spectrograms, 2,868 mono-species recordings and 24 soundscapes. An increase in training samples slightly improves the overall performance. A restart of the training process with more samples and pre-trained model weights appears to result in different gradient minima and thus occasionally decreased performance. Born-again networks do not outperform models trained on binary labels but significantly increase the soundscape performance (see Table 5.21). All models used moderate oversampling to counter class imbalances.

		BN1000 Val split		BN1000 Test split		DCSC	
ID	SAMPLES	MAP	cMAP	MAP	cMAP	F0.5	AUC
1	1000	0.637	0.596	0.766	0.729	0.319	0.593
2	1500	0.647	0.601	0.769	0.735	0.275	**0.601**
3	2000	**0.655**	**0.611**	**0.772**	**0.739**	0.294	0.580
4	2500	0.646	0.600	0.767	0.731	0.314	0.570
5	3000	0.651	0.607	0.771	0.735	**0.323**	0.582
4**	2500	0.652	0.606	0.765	0.727	0.322	0.553

*** trained as born-again network*

maximized to avoid costly ensemble strategies. A number of hyperparameters can be tuned to improve the overall scores for the DCSC dataset. Most prominently, the confidence threshold applied to distinguish between bird vocalizations and non-events has to be adjusted to provide objective scoring. The F0.5 measure is the most important metric for this task and threshold optimization aims at maximizing this score.

As part of the BirdCLEF 2019 soundscape evaluation campaign, Lasseck (MfN, Museum für Naturkunde, Berlin) submitted two single-model runs that achieved state-of-the-art performance on the BC2019 test data[2]. One of his models was trained using the validation data provided by the organizers. Considering only dawn chorus recordings, those two runs achieved a F0.5 score of 0.243 and 0.412 when validation samples were used (see Table 5.21).

---

[2] Results are part of the CLEF 2019 working notes collection. At the time of writing, this collection has not been officially published.

Table 5.21.: Investigation of single-model soundscape performance. Model 4 provides the best performance of all models trained on binary targets. The born-again version of this network significantly improves the scores across almost every metric. This implies that 'soft' labels enable a DNN to grasp faint or distorted vocalizations.

		DCSC					
ID	CT	P	R	MAP	cMAP	F0.5	AUC
MfN 1	0.52	0.356	0.148	0.148	0.107	0.243	0.668
MfN 2*	0.43	0.452	0.349	0.354	0.205	0.412	0.646
1	0.08	0.425	0.235	0.234	0.140	0.335	0.634
2	0.05	0.388	0.231	0.230	0.157	0.316	**0.657**
3	0.07	0.433	0.229	0.231	0.159	0.334	0.612
4	0.06	0.456	0.251	0.252	0.153	0.359	0.611
5	0.07	0.442	0.236	0.236	0.150	0.342	0.623
4**	0.06	**0.495**	**0.272**	**0.276**	**0.172**	**0.389**	0.605

ID = Model identifier according to BirdCLEF 2019 submissions and Table 5.20
CT = Best confidence threshold (according to the F0.5 measure)
*P = Precision, R = Recall, * used validation data for training*
*** trained as born-again network*

Interestingly, the best performing BirdNET model archives a F0.5 measure of 0.389 and significantly outperforms MfN 1 (which only featured 659 species). However, using validation data to fine-tune a trained model (MfN 2) drastically improves the scores. Two explanations for this observation have to be considered: First, the shift in acoustic domain between mono-species recordings and soundscapes can be overcome with samples of the expected target domain. Due to the high efforts needed to collect and annotate this data, this approach appears to be non-optimal. However, the increase in performance is considerably high and manually labeling soundscape data might be worth the effort. A second explanation evolves around the assumption that evaluation campaigns often reward overfitting to the employed metric. In this case, the provided validation data consisted of three fully annotated days and thus a very representative portion of the entire dataset (test data consisted of 12 days). Assuming that the contained vocalizations covered most of the value distribution of the test data, training with validation samples can be considered training with test samples and thus overfitting to the test data. Future investigations have to address this uncanny increase in performance to determine which of the two explanations is the most plausible.

Other performance enhancing strategies to increase the overall scores on the soundscape data involve the bagging of scores through exponential pooling and the adjustment of the sigmoid activation function that converts network outputs into class probabilities. Again, the experimental focus was on improving the F0.5 measure for the DCSC data (optimizing other metrics can be done using the same approach and mostly depends on the target use case). Additionally, the mentioned methods can be used to influence the behavior during application and help to adjust prediction probabilities depending on the use case, when the quality of input recordings differs from training data. Due to this, scores achieved during those experiments are (to some extend) the result of overfitting to the soundscape data and metrics.

The 2019 BirdCLEF evaluation system requires to predict bird species for 5-second intervals. With an overlap of one second, the number of analyzed spectrograms per time interval for current recognition system increases. Additionally, higher temporal resolution helps to cope with overlapping vocalizations—especially during the dawn chorus. Thus, a slight increase in performance can be observed (see Table 5.22). However, overlapping spectrograms and increasing the number of predictions comes at the cost of increased analysis time. For noncritical use cases, this change might be worth the additional computational costs.

Suppressing low class probabilities through other than average pooling is also very effective—especially for faint vocalizations and uncertain predictions. When the amount of test samples for a specific prediction interval is increased (e.g. by overlapping spectrograms), class probabilities p can be effectively pooled with:

$$p_i = (s \cdot y_i)^2 \tag{5.5}$$

The adjustable factor s linearly scales the class predictions y for each class i before squaring the resulting scores and thus emphasizes class probabilities until a certain threshold. A scaling factor of 2 improves all scores > 0.5, a scaling factor of 8 increases all scores > 0.125. In terms of maximized F0.5 measure, $s = 2$ performs best. Higher values for s however significantly improve the cMAP metric.

Table 5.22.: Investigation of enhanced single-model soundscape performance. Depending on the use case, techniques to optimize towards specific metrics can be applied. Adjusting spectrogram overlap, bagging of scores and sigmoid activation sensitivity can help to increase the performance for specific applications. Low AUC scores are due to strict elimination of false positives below an F0.5-optimal confidence threshold.

	DCSC					
METHOD	P	R	MAP	cMAP	F0.5	AUC
MfN 1	0.356	0.148	0.148	0.107	0.243	0.668
MfN 2*	0.452	0.349	0.354	0.205	0.412	0.646
NONE	**0.495**	0.272	0.276	0.172	0.389	0.605
OVERLAP	0.482	0.29	0.293	0.178	**0.395**	0.596
EXP POOL	0.490	0.278	0.284	0.210	0.391	0.605
SIGMOID	0.469	0.289	0.292	0.178	0.388	**0.629**
ALL	0.474	**0.294**	**0.300**	**0.218**	0.394	0.597

*P = Precision, R = Recall, * used validation data for training*

Pooling (or bagging) of scores to merge multiple predictions into one can also be very effective when applied to ensembles. Although ensembles of trained networks often require a vast amount of computational resources, the increase in performance might be worth considering. Again, due to potential overfitting to the evaluation metric, this method is particularly popular across all evaluation campaigns.

The sigmoid activation function to convert class activations y into probability scores p is defined as

$$p_i = \frac{1}{1 + \exp(-s \cdot y_i)} \tag{5.6}$$

with a scaling factor s of 1.0 in the standard implementation. Lowering this scaling factor leads to higher scores for uncertain class predictions, higher values of s require higher class activations to produce the same score. Adjusting s is one of the most convenient strategies to increase recall or prediction according to the desired application at the time of inference. With $s = 1.125$, the F0.5 measure is highest for the DCSC data. When all of the aforementioned approaches are combined, a scaling factor of 0.95 performs best.

The combination of all enhancing strategies archives a F0.5 score of 0.394, which outperforms Lasseck's approach (MfN 1) by 15.1% and almost reaches the single-model performance when validation data is used for training (MfN 2). Considering the uncertain reasons for the drastically improved scores of Lasseck's second run, the experimentally derived scores resemble state-of-the-art performance. However, the F0.5 measure was not the primary metric of the BirdCLEF challenge and the submitted runs are thus not entirely optimized towards this measure. The margin in performance is still significant. Especially when considering the higher number of classes and custom network architecture (Lasseck uses the Inception-v3 model, [Szegedy et al., 2016]). Ensemble methods for DNN are very popular and typically lead to slightly improved scores when optimized for campaign-specific metrics [Lasseck, 2018b], [Schlüter, 2018]. Due to the high costs of processing soundscapes with model ensembles, this strategy was not further pursued. The established (single-network) baseline models will be used to investigate more aspects of the domain of bird sound recognition in Section 5.3.

Mobile benchmark experiments

Deep architectures for mobile applications achieve competitive scores if optimized for a specific task. When the number of classes is limited, these models perform almost on par with large, very deep topologies—especially in the soundscape domain (see Table 5.19). However, increasing the number of classes and thus the complexity of the input data might lead to drastically reduced scores due to insufficient network capacity. It might not be the ideal choice to force a mobile net to learn features of a large amount of classes. When applied to a specific habitat, only a certain number of bird species is relevant for this location. However, for the sake of comparability, all 984 species and three non-event classes were part of the mobile benchmark experiments.

The validation and test data remained unchanged compared to previous trials, dawn chorus soundscapes were used to evaluate real-world performance. The training regime also remained unaltered to provide fair conditions for all trials. During the experiments, the number of training samples was steadily increased, pre-trained models were used to initialize each new run. Again, dropout probability and learning

Table 5.23.: Benchmark experiments for mobile architectures. An increase in training samples leads to higher class imbalance but improves the scores until a certain threshold. A split size of 2,500 training samples appears to perform best. Mobile models perform significantly worse than large topologies. Considering the computational constraints, the results are still competitive. Knowledge distillation helps to improve the soundscape performance and can be considered a valuable addition to the overall training procedure.

		BN1000 Val split		BN1000 Test split		DCSC	
ID	SAMPLES	MAP	cMAP	MAP	cMAP	F0.5	AUC
1	1000	0.575	0.482	0.690	0.619	0.271	**0.673**
2	1500	0.573	0.485	0.690	0.630	0.285	0.644
3	2000	0.581	0.493	0.695	0.632	0.268	0.654
4	2500	**0.584**	**0.498**	0.697	0.631	0.283	0.664
5	3000	0.582	0.497	0.694	**0.638**	0.265	0.656
6	1000*	0.574	0.489	0.699	0.634	0.325	0.652
7	2500*	0.574	0.492	**0.700**	**0.638**	**0.330**	0.625

** trained with knowledge distillation*

rate were reduced with each start of a new training process and kept constant across all epochs.

The experimental results mostly reflect the outcome of previous trials: Increasing the number of samples does increase the overall scores until a certain threshold (Table 5.23). A training split size of 2,500 samples per class appears to perform best. Yet, the margin between scores of the two benchmark models (non-mobile vs. mobile) is significant. The amount of classes in the benchmark trial clearly demonstrated that mobile networks suffer greatly when computational constraints apply. Reducing the input data complexity based on local environmental characteristics is recommended.

Knowledge distillation supposedly helps to increase the overall performance despite the lack of network capacity. Similar to training a born-again network, knowledge distillation (or model distillation) uses the predictions of a teacher model to present targets for training of a student network. These 'soft' targets resemble the un-

Table 5.24.: Investigation of single-model soundscape performance for mobile architectures. Again, more training samples do not necessarily lead to better scores. Knowledge distillation along overlap, pooling and sigmoid variations provide significantly improved performance. The best single mobile model outperforms MfN1 by a significant margin despite its limited computational footprint.

		DCSC					
ID	CT	P	R	MAP	cMAP	F0.5	AUC
MfN 1	0.52	0.356	0.148	0.148	0.107	0.243	0.668
MfN 2*	0.43	0.452	0.349	0.354	0.205	0.412	0.646
1	0.10	0.395	0.17	0.168	0.108	0.274	0.685
2	0.08	0.379	0.197	0.193	0.117	0.290	0.676
3	0.09	0.351	0.195	0.189	0.116	0.277	**0.689**
4	0.12	0.379	0.197	0.193	0.116	0.290	0.674
5	0.09	0.366	0.195	0.188	0.119	0.283	0.679
6	0.14	0.453	0.209	0.208	0.109	0.326	0.657
7	0.11	**0.472**	0.217	**0.225**	0.118	0.339	0.652
7**	012	0.461	**0.227**	**0.225**	**0.135**	**0.344**	0.625

ID = Model identifier according to BirdCLEF 2019 submissions and Table 5.23
CT = Best confidence threshold (according to the F0.5 measure)
*P = Precision, R = Recall, * used validation data for training*
*** used overlap, pooling and sigmoid variation at test time*

certainty of predictions with low confidence—which is especially important when weak labels lead to ‘empty’ samples. The employed teacher model in this series of experiments was derived from the best performing single model of previous trials. Based on the results in Table 5.21, model 4 was selected. The resulting scores of this training process show that knowledge distillation does not boost the scores for clean samples of mono-species recordings. However, the soundscape performance drastically improves through training with uncertain predictions. It appears that this domain profits from a networks ability to cope with distorted samples—which is learned by imitating the teacher model.

Soundscape performance of mobile networks increases even further when the aforementioned post-prediction techniques are applied (see Table 5.24). Training with more samples does not automatically lead to better scores—which is consistent with previous observations and backs the assumption that the employed training regime is very effective even for limited amounts of samples per class. Again, post-processing of predictions through bagging of scores, increased sensitivity due to adjusted sigmoid activation and overlapping predictions improve the F.05 measure to 0.344. This score is substantially worse when compared to large models but still outperforms Lasseck's approach by 10.1%—an impressive margin considering the small computational footprint of this network design. Ensembles of models might improve the overall performance but are omitted in this investigation due to the lack of real-world applicability.

In conclusion, the proposed DNN designs provide state-of-the-art performance, even when the amount of training samples is limited. The employed training regime appears to be very effective and—along with task-specific optimizations—leads to unprecedented scores in the soundscape domain. Knowledge distillation is a valuable addition to train born-again networks and mobile architectures that lack capacity. Training samples that originate from the target domain might increase the performance even further and could be worth the labor intensive annotation of soundscape data.

5.3. Results

With the two best performing models established, the final experiments included the entire BirdNET 1000 test data and all 2019 BirdCLEF soundscapes from North America. In total, the test data contained 22,960 mono-species recordings and 286 soundscapes (covering 12 days). In both trials, BirdNET achieves top performance considering previous attempts of BirdCLEF participants (see Tables 5.25 and 3.1). Across all samples, the mean average precision is 0.791 with an AUC of 0.974. These scores are slightly better than during previous investigations due to the increased class imbalance. The class-wise mean average precision provides a more balanced performance estimation but still indicates good overall classification quality.

Table 5.25.: Final mono-species results for the best single models. The final test dataset contained more than 20,000 recordings with weak labels. BirdNET achieves state-of-the-art performance on an unprecedented amount of samples including background species. Scores of the mobile version drop considerably but are still very competitive.

	BN1000			
MODEL	TOP-1 ACC	MAP	cMAP	AUC
BirdNET	0.777	0.791	0.694	0.974
BirdNET Pi	0.699	0.728	0.580	0.969

ACC = Accuracy

Due to weakly labeled samples and incomplete notation of background species, the real-world performance is very likely to be better than the scores indicate. The lack of a ‘gold standard’ prevents a fully objective result estimation. Yet, considering the high number of classes, vast intra-class heterogeneity and diverse test recordings, the current recognition quality is most likely applicable for a variety of use cases in avian activity monitoring. Although mono-species results are not entirely comparable (BirdCLEF 2018 featured more classes but manually curated data; top scores were achieved by ensembles), we can conclude that deep neural networks are able to extract high quality features from extremely complex input data to recognize birds in field recordings.

Soundscapes are of particular interest and the transfer of knowledge derived from high quality recordings to noisy soundscape data appears to be a very practical approach when manual interference is not desirable. Despite the shift in acoustic domains, BirdNET achieves strong results with a final F0.5-measure of 0.414 (Table 5.26). Again, mobile performance drops considerably but remains applicable—especially when considering the possibility of real-time processing. The best single model outperforms Lasseck’s approach MfN 1 by 15.4% and is almost on par with MfN 2 that used validation samples for training. Additionally, the best mobile model outperforms MfN 1 by 9.1% despite significantly more classes and cost-efficient architecture.

Table 5.26.: Final soundscape results for the best single models. The overall performance of all models slightly increases on the entire 2019 BirdCLEF Soundscape test data. BirdNET significantly outperforms MfN1 and is on par with MfN 2 without the use of validation samples.

		BC2019					
MODEL	CT	P	R	MAP	cMAP	F0.5	AUC
MfN 1	0.52	0.335	0.180	0.186	0.135	0.260	0.645
MfN 2*	0.44	0.451	0.360	0.371	0.228	0.416	0.627
BirdNET	0.04	0.449	0.358	0.359	0.228	0.414	0.584
BirdNET Pi	0.12	0.419	0.265	0.262	0.139	0.351	0.637

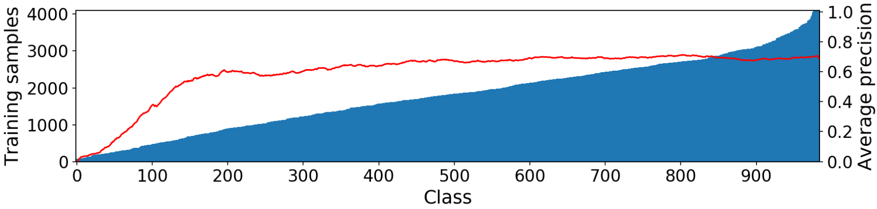
CT = Best confidence threshold (according to the F0.5 measure)
*P = Precision, R = Recall, * used validation data for training*

Some interesting questions concerning the overall classification performance arise form the investigation of the benchmark scores:

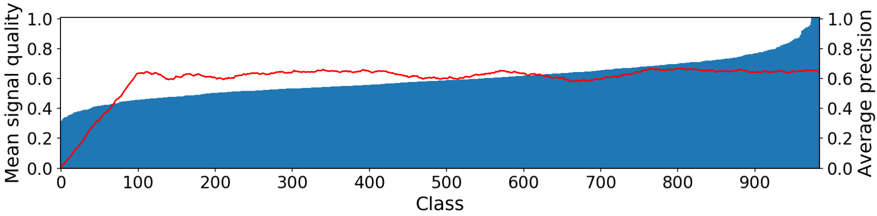
- To which extend is the overall performance affected by low signal quality in training recordings?
- How do weak and noisy labels affect the achieved scores?
- If at all, how does species diversity affect the overall performance?
- How well does the model perform for extremely diverse species?

Finding the answers to these questions would need a ‘gold standard’ of correctly labeled recordings for every species in the dataset. Due to the lack of timestamps, the available validation data often contains false labels, non-events and prominent background species. However, we have to cope with these circumstances and it is assumed that all validation samples have a correct label for the following investigation. The entire validation dataset contains 208,610 spectrograms and spans 985 classes (all bird species + human vocal sounds). An extensive listing of class-specific results can be found in Appendix D.

In total, only 15 species were classified with an AUC score below 0.7, only 34 with an AUC score below 0.8. Considering the vast amount of classes, this result is very promising and demonstrates the effectiveness of the proposed system. The AUC



(a) Number of training samples



(b) Mean signal quality

Figure 5.7.: Factors that impact the overall performance. Class-wise average precision (red, smoothed with moving average) consistently improves with the number of available training samples (a). Signal quality of training samples (b) drastically decreases the classification performance until a threshold of 0.4 (noise measure based on morphological features, see Figure 4.1).

measure can be seen as a very effective assessment of intra-class detection quality and implies that very certain predictions mostly contain the correct event. When ranking the validation samples based on class scores, the average precision of each class provides another good look at the classification quality for each individual class. In this domain, only 58 species show an average precision below 0.3, only 184 species are classified with an average precision below 0.5—above this threshold, the results can be considered real-world applicable. Interestingly, the worst performing species contain Owls, Doves, Sparrows, Sandpiper, Finches and others—a diverse mix of genera but no species that can be considered widely spread or very common. Yet, this also applies for the top-performing classes, raising the question whether genus and abundance are signifiers for classification quality.

After close investigation, two main factors appear to impact the overall detection performance most: The number of available training samples and the signal quality among those samples (see Figure 5.7). The class-wise average precision is strongly affected when the number of available training samples is below 750. This implies two major drawbacks of the presented approach: First, deep neural networks are data hungry and require many samples to train a classifier from scratch. Secondly, rare species that feature only a few recordings on Xeno-canto or in the Macaulay Library might not be reliably detectable due to this circumstance. Additionally, the average precision is affected when the recording quality is low. Based on the proposed signal-to-noise measure (that uses morphological features to determine the signal strength), an average signal quality below 0.4 significantly impacts the class-wise performance. For those classes, manual selection of training samples might help to improve the scores while maintaining a high level of automation.

Still, the quality of the training data alone does not suffice to predict how well a certain species can be detected by the proposed system. Another important dimension is species diversity—mostly in terms of repertoire size (see Table 5.27). The examination of class-specific results for selected species often leads to a somewhat ambivalent picture: The Ovenbird (*Seiurus aurocapilla*), as a rather simple species, has poor training data quality and overall poor classification performance. Yet, the Hermit Thrush (*Catharus guttatus*)—another fairly simple species—has also poor quality training data but performs best among all investigated species. Species that are known to incorporate hetero-specific material into their vocalizations like the European Starling (*Sturnus vulgaris*) and Northern Mockingbird (*Mimus polyglottos*) imply that imitation is a significant challenge for automated recognition systems. Both classes show poor performance independent of training data quality. For extremely diverse species with vast repertoires, confusion with other species might affect the scores more than any other dimension. When good quality training data is available, repertoire size does not affect the classification scores—the model achieves high scores for Common Nightingale (*Luscinia megarhynchos*) and Brown Thrasher (*Toxostoma rufum*). However, the Song Thrush (*Turdus philomelos*) probably suffers from confusion with the Eurasian Blackbird (*Turdus merula*), which also can be observed when analyzing soundscape data. Species diversity with a (supposedly) high number of false labels might amplify this effect.

Table 5.27.: Correlation between species diversity, sample count, signal quality, and overall scores. Repertoire size alone is no conclusive indication for recognition quality. Noisy training data affects the overall performance and mostly leads to decreased performance for any species diversity. Birds that imitate (European Starling, Northern Mockingbird) show unsatisfactory performance despite good quality training data.

		BN1000 Training and validation data			
SPECIES	RS	TS	SQ	AP	F0.5
Ovenbird	1	*2326*	0.580	0.503	0.519
White-crowned Sparrow	1	3059	*0.519*	0.633	0.644
Common Chaffinch	1-6	3360	0.651	0.580	0.604
Great Tit	2-8	3769	0.554	0.611	0.554
Hermit Thrush	6-12	3071	0.520	**0.875**	**0.866**
Song Sparrow	7-11	3176	0.593	0.542	0.599
European Starling	15-70	3934	0.707	0.509	0.426
Marsh Wren	33-162	3288	0.727	0.790	0.740
Northern Mockingbird	53-150	3833	0.609	*0.443*	*0.352*
Common Nightingale	160-231	**4241**	**0.785**	0.745	0.737
Song Thrush	138-219	3816	0.751	0.541	0.425
Brown Thrasher	1500+	3508	0.726	0.764	0.706

RS = Repertoire size, TS = Training samples, SQ = Mean signal quality
AP = Average precision, F0.5 = Maximum F0.5 through optimized confidence

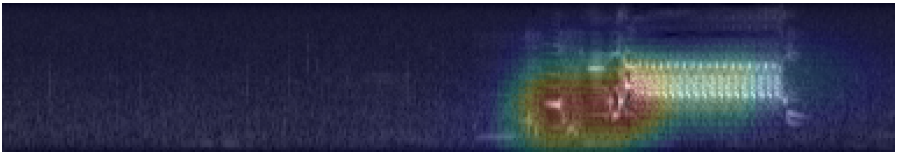
Local dialects play a significant role in bird song recognition and repertoire size is only one dimension of species diversity. One core aspect of song variation through dialects is the permutation of re-occurring elements (see Section 2.2.4). Additionally, song complexity plays an important roll as well. Two-voiced sounds or fast sequences of trill notes, frequency range and similarity between species have to be considered. We already know that species identity is encoded in bird songs (and calls). Statistically speaking, features that can be used to identify a certain species should occur in every vocalization. Deep neural networks heavily rely on those features and are very good in identifying and utilizing re-occurring patterns in avian vocalizations to recognize species. However, deep neural networks are known to be extremely difficult to interpret and are said to be ‘black boxes’.

Of all approaches to shed light on the decision-making of DNN, class activation maps (CAM) appear to be the most conclusive when trying to identify the most important parts of a bird vocalization. Other visualization methods like guided backpropagation ([Springenberg et al., 2014]) or deconvolution ([Zeiler and Fergus, 2014]) yield high resolution outputs but are not class-discriminative.

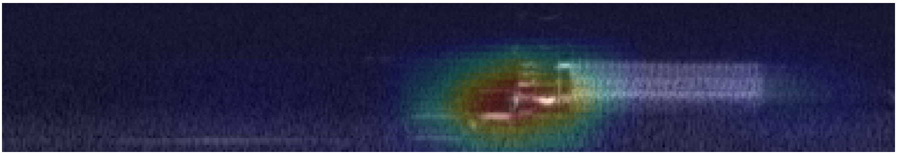
First proposed in [Zhou et al., 2016], CAM highlight the most important parts of an image in terms of class-specific scores. CAM can be derived by weighting convolutional layer activations (as proposed by Zhou et al.), weighted gradients ([Selvaraju et al., 2017]) or simply by occluding certain parts of the input image and observing the scores (occlusion mapping, [Zeiler and Fergus, 2014])—which is independent of the network architecture but still very discriminative.

When occluding the input spectrogram, a sliding window successively occludes every part of the image. In the occluded area, all values are set to zero. Despite the fact that this might introduce an unwanted bias that might force the net to react unpredictably (similar to adversarial inputs, [Szegedy et al., 2013], [Kurakin et al., 2016]), zero values are common artifacts in the training data due to pre-processed recordings. The proposed model appears to be able to cope with such impairments. However, whenever an important region is occluded, the class score of the primary species should drop. Lower scores after occlusion imply high importance for the particular region. Figures 5.8 to 5.15 provide insights into the most important parts of bird vocalizations for selected species that were identified with high confidence.

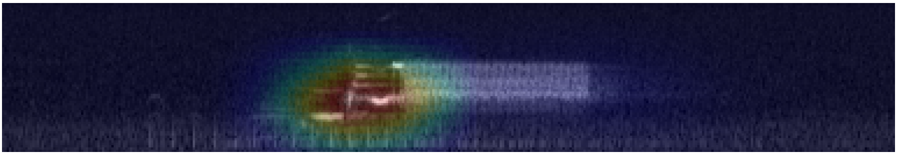
Although the depicted activation maps might not reveal how birds encode species identity, they imply that certain parts of every bird vocalization are of high significance to be identifiable by the proposed recognition system. Most notably, redundant elements (as in trills) compensate information loss (e.g. Red-winged Blackbird and Wood Thrush), re-occurring elements suffice for species identification (e.g. Common Chaffinch and White-crowned Sparrow), and gaps between notes and the duration of single elements help to identify similar sounding species (e.g. Black-capped Chickadee and Tufted Titmouse). Sometimes, only a small frequency band or portion of a vocalization encode species identity (e.g. Blue Jay and Common Buzzard).



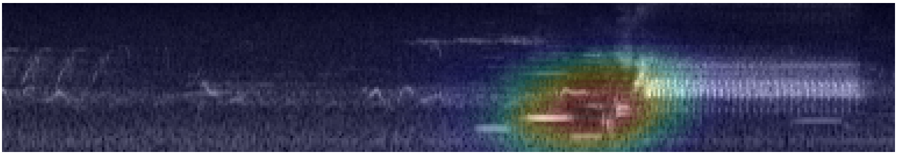
(a) c = 0.98



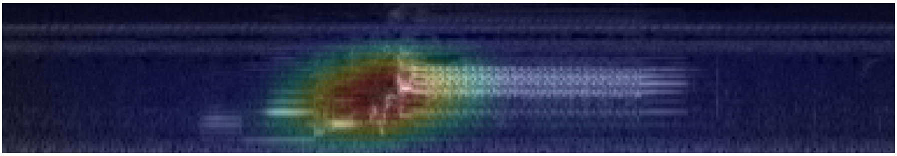
(b) c = 0.95



(c) c = 0.95

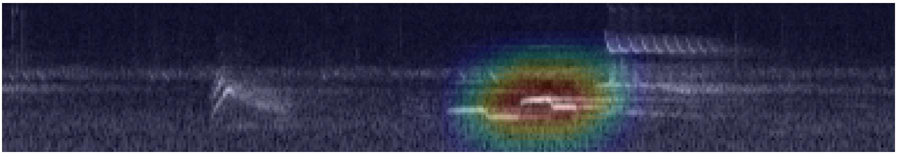


(d) c = 0.88

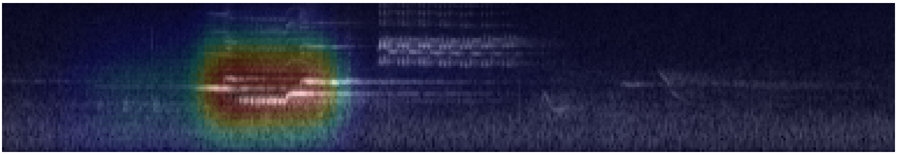


(e) c = 0.88

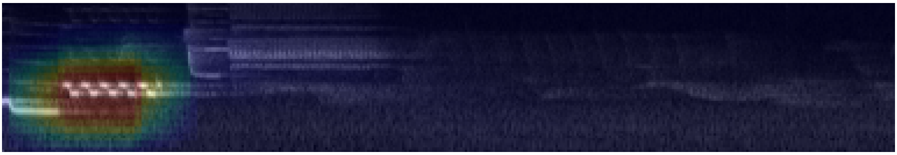
Figure 5.8.: Class activation maps of Red-winged Blackbird (*Agelaius phoeniceus*) vocalizations with confidence *c*. Introductory notes are consistently more important for species identification than the characteristic trill that appears to provide enough redundancy to compensate information loss. (🔊 24)



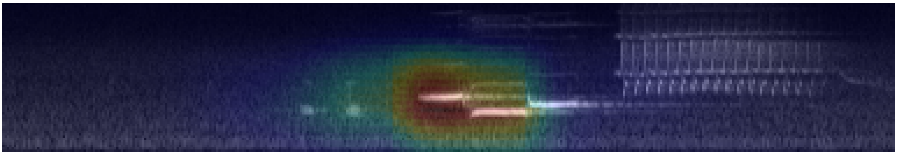
(a) c = 0.99



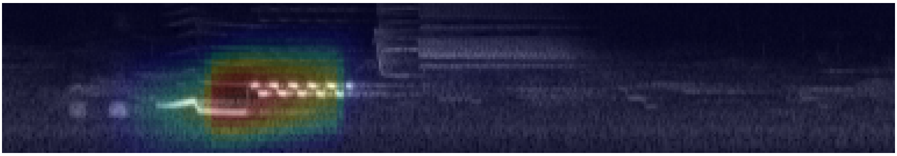
(b) c = 0.99



(c) c = 0.99

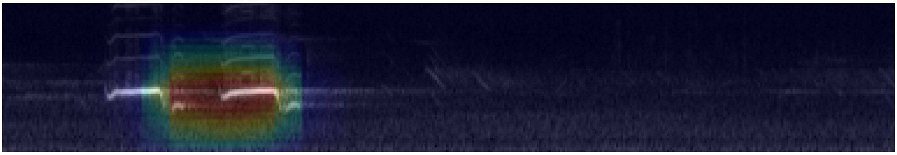


(d) c = 0.99

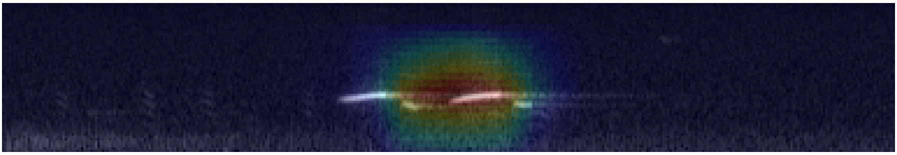


(e) c = 0.99

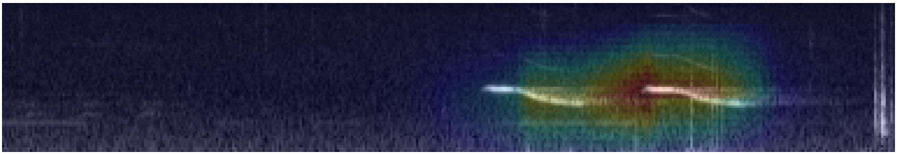
Figure 5.9.: Class activation maps of Wood Thrush (*Hylocichla mustelina*) vocalizations with confidence *c*. This species is one of the most prominent examples for two-voiced sounds with its characteristic trills at the end of each song. Despite this re-occurring pattern, introductory notes consistently lead to higher class activation. (🔊 25)



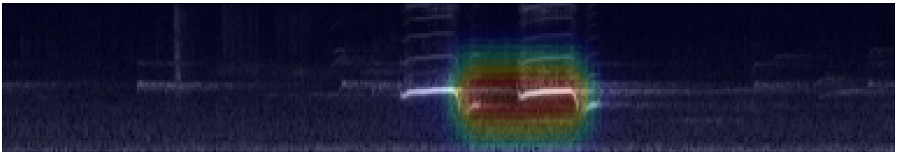
(a) c = 0.99



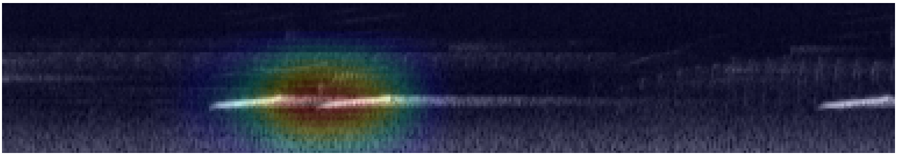
(b) c = 0.99



(c) c = 0.99

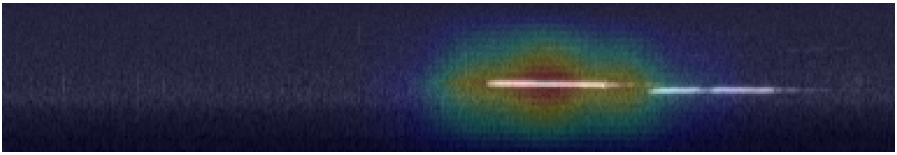


(d) c = 0.99

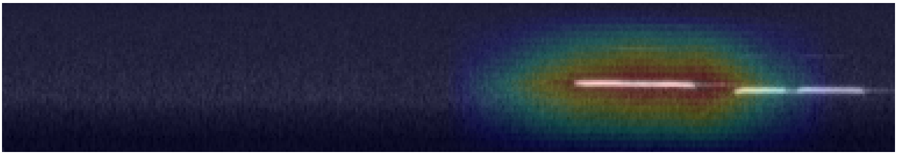


(e) c = 0.98

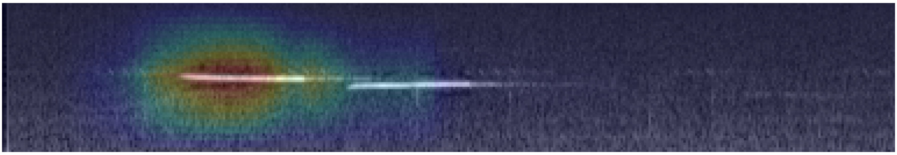
Figure 5.10.: Class activation maps of Tufted Titmouse (*Baeolophus bicolor*) vocalizations with confidence *c*. The gap between two consecutive tones appears to be the discriminating feature for species identification. The introductory note alone does not suffice to distinguish the Tufted Titmouse from other species like the Black-capped Chickadee (*Poecile atricapillus*) that utter similar tonal sequences. (🔊 26)



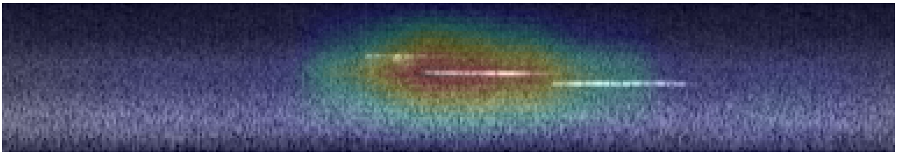
(a) c = 0.99



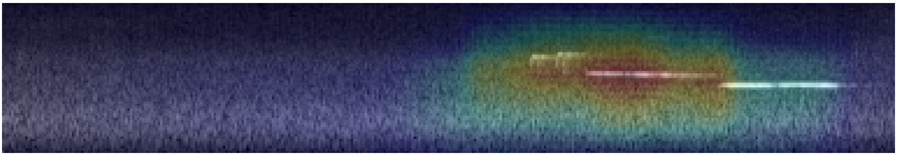
(b) c = 0.99



(c) c = 0.98

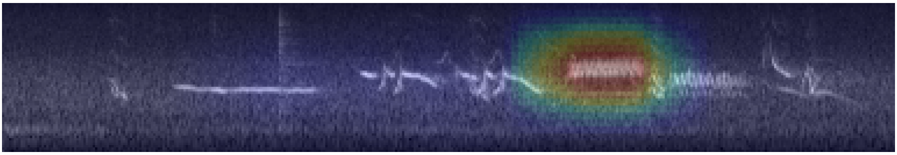


(d) c = 0.91

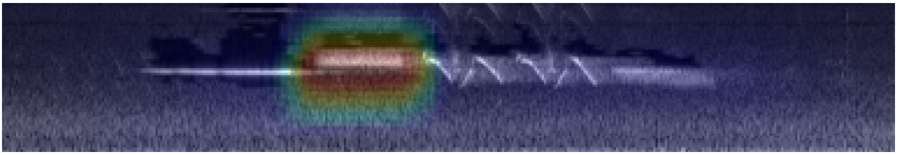


(e) c = 0.89

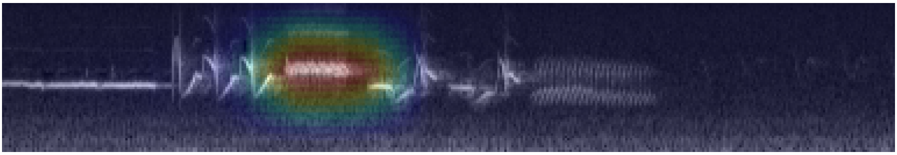
Figure 5.11.: Class activation maps of Black-capped Chickadee (*Poecile atricapillus*) vocalizations with confidence *c*. Although the species-specific vocalizations are similar to some variations of the Tufted Titmouse song (see Figure 5.10), class activations show a distinct focus on the first note while the gap between notes does not appear to be important for species identification. (🔊 27)



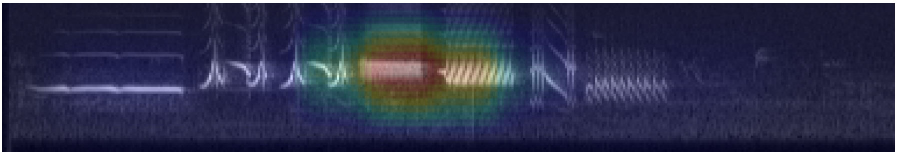
(a) c = 0.99



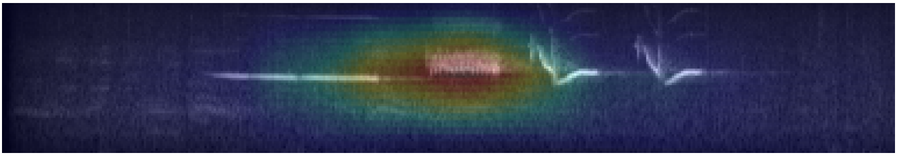
(b) c = 0.99



(c) c = 0.99

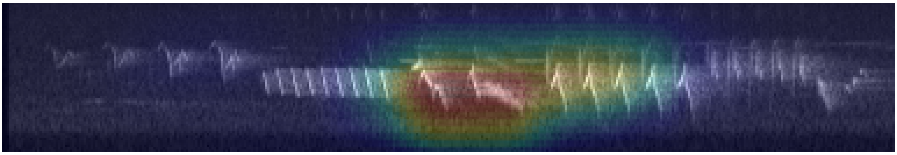


(d) c = 0.99

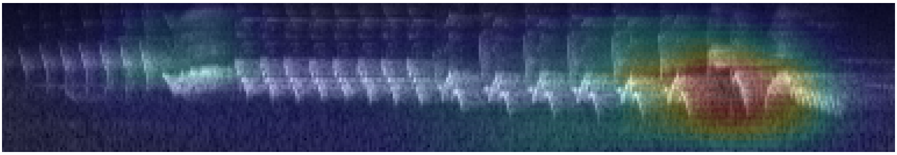


(e) c = 0.83

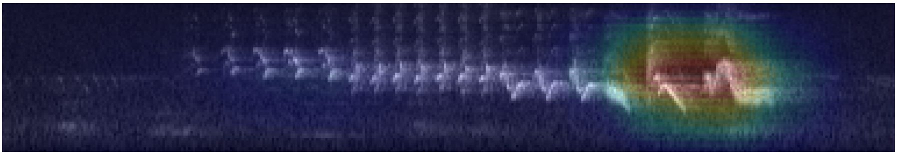
Figure 5.12.: Class activation maps of White-crowned Sparrow (*Zonotrichia leucophrys*) vocalizations with confidence *c*. Although an individual in the field only utters a single song, permutations of song elements lead to a vast number of regional dialects. Re-occurring elements are key to identify this species. Other elements appear to contain redundant or insufficient information for identification. (🔊 28)



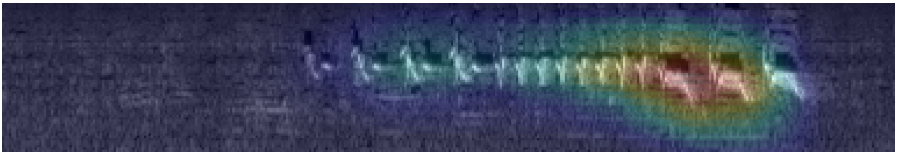
(a) c = 0.99



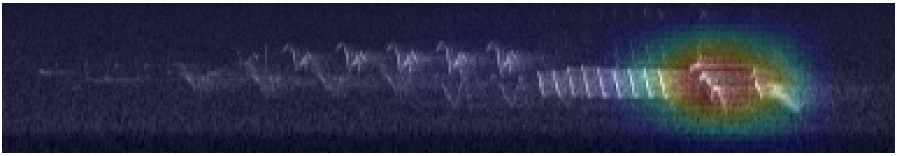
(b) c = 0.99



(c) c = 0.99

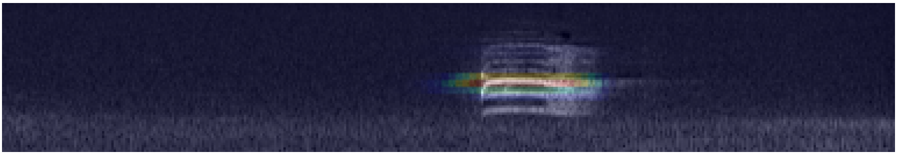


(d) c = 0.98

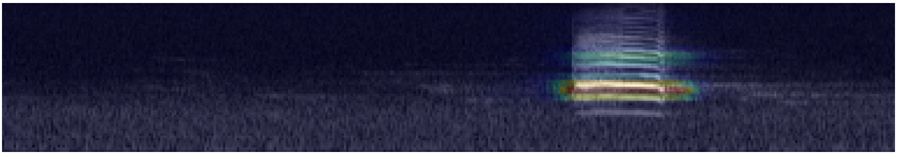


(e) c = 0.98

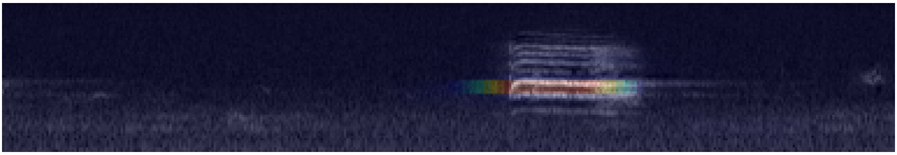
Figure 5.13.: Class activation maps of Common Chaffinch (*Fringilla coelebs*) vocalizations with confidence *c*. Again, re-occurring elements are key for identification. Despite distinct regional dialects between populations of this species, common patterns in song lead to high class activation independent of their location in the song sequence. (🔊 29)



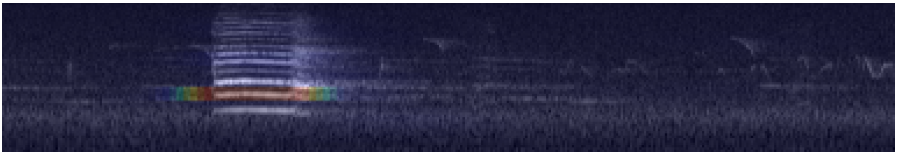
(a) c = 0.99



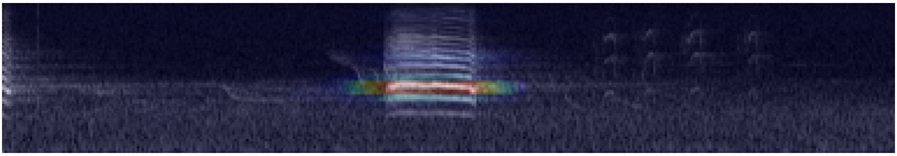
(b) c = 0.99



(c) c = 0.99

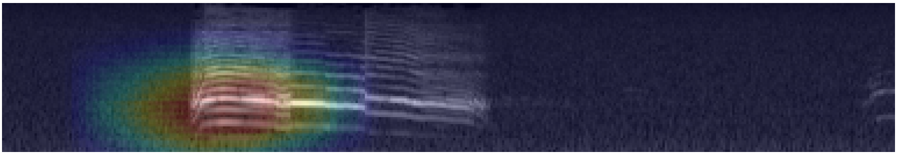


(d) c = 0.99

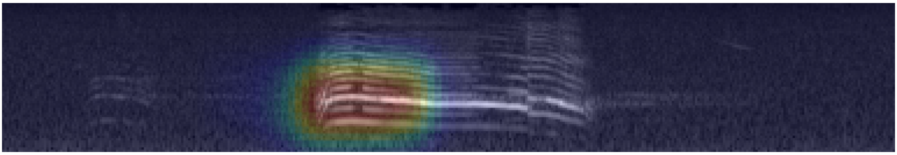


(e) c = 0.99

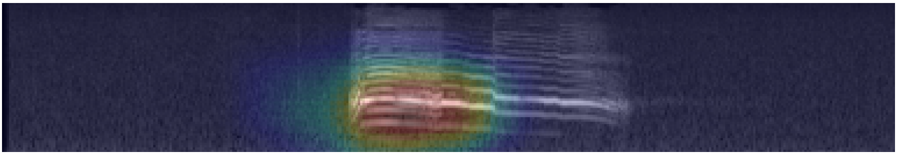
Figure 5.14.: Class activation maps of Blue Jay (*Cyanocitta cristata*) vocalizations with confidence *c*. The nasal 'jeer' of this species is characteristic and commonly heard. Only a single frequency band appears to encode species identity and thus leads to significant drops in confidence when occluded. (🔊 30)



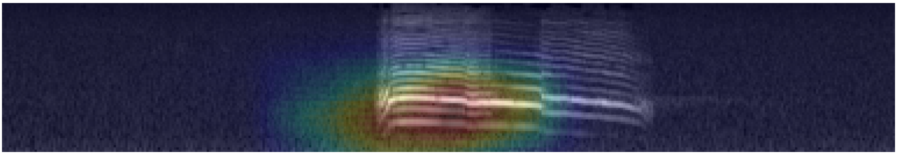
(a) c = 0.99



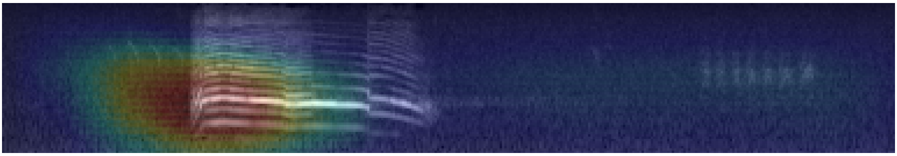
(b) c = 0.99



(c) c = 0.99



(d) c = 0.97



(e) c = 0.88

Figure 5.15.: Class activation maps of Common Buzzard (*Buteo buteo*) vocalizations with confidence *c*. The nasal upslur at the beginning of each vocalization appears to be the most discriminating feature. Other parts of the utterance contain redundant information that does not lead to increased class activation. (🔊 31)

5.4. Summary

The design of the experimental studies in this chapter evolved around the ability of DNN to generalize on unseen samples despite a high number of classes with significant intra-class heterogeneity. A number of hypothesis were tested on an unprecedented amount of training, validation, and test data that consisted of more than 3,900 hours of field recordings covering 987 classes and almost 300 hours of fully annotated soundscapes containing almost 80,000 vocalizations. Considering constraints of real-world applications, two main DNN designs were tested: A Wide ResNet architecture with elaborate residual blocks for powerful workstations as well as a simplified ResNet variation following the original design for mobile platforms like the Raspberry Pi.

The investigation focused on spectrogram computation, architecture design, deep and shallow topologies, implicit and explicit regularization, cost-sensitive learning, and model efficiency when computational resources are limited. All formulated hypotheses were confirmed or partially confirmed; the most notable results imply that

- Spectrograms that visualize longer chunks of audio contain more valuable information and thus result in better classification performance.
- High temporal resolution of input spectrograms (short frame length) improves the classification performance.
- Multi-label classification with mixup training increases the overall performance across all tasks.
- Deeper topologies (more layers) do not necessarily perform better than wider topologies (more filters).
- Deeper topologies do outperform shallow layouts when computational resources are limited.
- Except for oversampling, cost-sensitive learning does not improve the overall scores.

The resulting benchmark system yields state-of-the-art scores across all domains, especially when compared to the scores achieved during the 2019 BirdCLEF challenge. With an increase of 15.4% over the best single model that did not use validation samples for training, the proposed training regime and DNN design appear to be competitive considering the difficulty of the task. The benchmark system also revealed that

- Deep neural networks are data hungry and require large numbers of training data (in this particular case up to 750 per class).
- Signal quality of training samples significantly affects the overall classification quality.
- Task-specific designs and training regimes outperform standard architectures from other campaigns.

The investigation of species-specific scores revealed that number and quality of training samples significantly impact the classification results. Additionally, species diversity plays an important role, especially for species that incorporate heterospecific material into their vocalizations or those that are likely to be confused with similar classes. For species identification, re-occurring elements and patterns provide important clues to recognize vocalizations despite very distinct regional dialects.

6. Applications

6.1 Live stream analysis **183**
  6.1.1 Abundance and vocal activity 185
  6.1.2 Temperature and vocal activity 186
6.2 SWIFT soundscape analysis **189**
  6.2.1 AMTiC 189
  6.2.2 SWAMP 193
6.3 Smartphone app **197**
  6.3.1 Scope 198
  6.3.2 Design 198
  6.3.3 Distribution 200
  6.3.4 Reception 201
  6.3.5 Results 202
6.4 HaikuBox **208**
6.5 Summary **209**

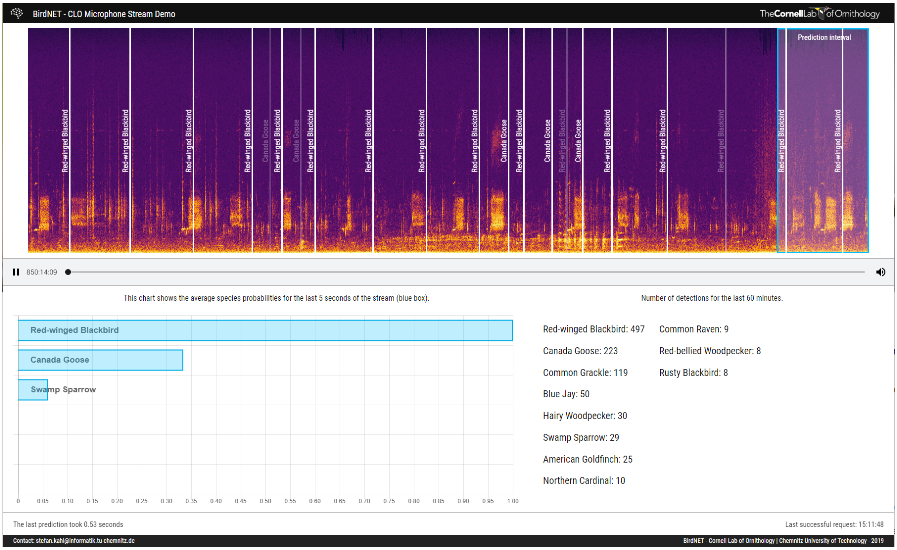
Applications are an important corner stone of scientific communication with the public. Optimizing the behavior of the developed system for each application helps to establish strategies to cope with unforeseen constraints of real-world data. Due to its modular nature, BirdNET assembles different processing chains for each application. Each of those chains consists of task-specific basic components. The highly-automated model export and import allow to rapidly develop new application scenarios for real-time analysis, autonomous recording stations, and mobile apps. I will present four of them in this chapter.

6.1. Live stream analysis

The BirdNET web demo for live stream audio data was one of the first applications to ever employ the automated recognition system[1]. The initial version was released

---

[1] https://birdnet.cornell.edu/live/



BirdNET - CLO Microphone Stream Demo

This chart shows the average species probabilities for the last 5 seconds of the stream (blue box).

Number of detections for the last 60 minutes.

Red-winged Blackbird: 497
Canada Goose: 223
Common Grackle: 119
Blue Jay: 50
Hairy Woodpecker: 30
Swamp Sparrow: 29
American Goldfinch: 25
Northern Cardinal: 10
Common Raven: 9
Red-bellied Woodpecker: 8
Rusty Blackbird: 8

The last prediction took 0.53 seconds

Last successful request: 15:11:48

Contact: stefan.kahl@informatik.tu-chemnitz.de

BirdNET - Cornell Lab of Ornithology | Chemnitz University of Technology - 2019

Figure 6.1.: Live stream demo screenshot. The client visualizes the audio stream as spectrogram and places markers to indicate estimated timestamps of bird vocalizations. The bar chart contains species probabilities for the prediction interval, the species counter displays the number of detections of the last 60 minutes.

as early as November 2017 and immediately sparked the interest of bird enthusiasts and researchers alike. Intended for large screens, the demo was part of numerous presentations of the technology. It helped to analyze changes to the underlying system for real-world applications—especially over the course of changing seasons. Located in Ithaca, NY, USA, the weatherproof microphone used for this demo continuously records ambient sound with high sensitivity (and thus high range). Next to the microphone sits the bird feeding station used for the *'Cornell Lab FeederWatch Cam at Sapsucker Woods'*[2]. As a consequence, bird vocalizations are often the result of social interaction between individuals visiting the feeder. Additionally, a large water body that is home to a vast number of aquatic birds, is also in hearing range.

---

[2] https://youtu.be/5x01AdCuLnk

The species diversity of the stream, its high number of audible social calls, and especially its high recording range render the stream a particular challenging sound source. For the analysis, the input audio signal is buffered and read after each second to run the analysis with high resolution (in this case: three-second spectrograms with two-second overlap). Predictions are made for a total of five seconds, scores are bagged through exponential pooling. Since the analysis of the stream is independent from any visualization, the analysis server stores the current result vector filled with species probabilities on hard drive.

Online access to the data is provided through the BirdNET web API. The client (in most cases a web browser) renders the live stream data and visualizes the signal as continuous spectrogram (see Figure 6.1). Every 700 ms, the client requests the current analysis data and also plots charts and label markers. Those markers represent estimates of the location of every detected vocalization in the stream's spectrogram. Due to the independence of both stream-accessing methods, markers might show a slight drift compared to the actual vocalization. In addition to the probability visualization of each detection interval, a list of species that were detected over the course of the last 60 minutes is shown.

6.1.1. Abundance and vocal activity

This list demonstrates one of the central use cases of any stream analysis: Diversity estimation and avian (vocal) activity. Some soundscapes recorded by the streaming microphone are extremely complex and often lead to missed vocalizations. However, species diversity estimation does not require the system to detect every single vocalization. The most important measure is species presence—which can be estimated through accumulated detections. On top of that, avian vocal activity provides an interesting perspective on avian social behavior and abundance. Investigating analysis data from live stream audio over the course of an entire year reveals insights into how environmental factors like season, weather, and—most notably—migration might impact the vocal behavior of birds.

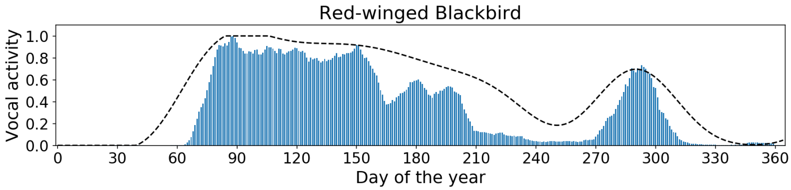
Figure 6.2 visualizes one of the most apparent aspects of avian vocal activity estimation: The correlation between abundance and vocal activity. Analysis results of 365 days of monitoring data reveal that the vocal activity of some species highly correlates with the abundance reported by bird watchers (through eBird checklists).

This applies for both, migratory and stationary species. The measured vocal activity highly depends on environmental factors and monitoring location.

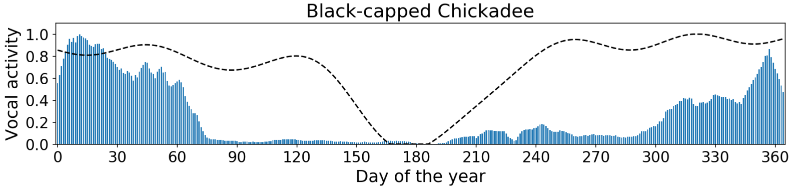
Placed near a bird feeder, the live stream microphone records vocal interaction of bird species and their rivalries. This is apparent for one of the most abundant species of Eastern North America: The Black-capped Chickadee (*Poecile atricapillus*) that appears to be extremely vocal during the winter feeding season. To some extent, this seems plausible. Black-capped Chickadees are known to encode information about predators, territorial males or alarming situations with a varying number of *dee* notes in their calls—a behavior that is apparently often provoked around bird feeders between November and February. Due to that, observed and measured activity often diverge notably. As another example of weak correlation between abundance and vocal activity, the Great Crested Flycatcher (*Myiarchus crinitus*) appears to vocalize mostly during breading season and remains mostly silent otherwise. Despite the fact that this species has an estimated presence between April and September, detections in the live stream data occur only between May and June. It remains questionable whether this behavior is entirely linked to breeding, but the observation suggests notable effects of mating and hatching on the overall vocal activity. Other species like the Red-winged Blackbird (*Agelaius phoeniceus*) or the Yellow-rumped Warbler (*Setophaga coronata*) show much less difference between observed abundance and recorded vocal activity.

6.1.2. Temperature and vocal activity

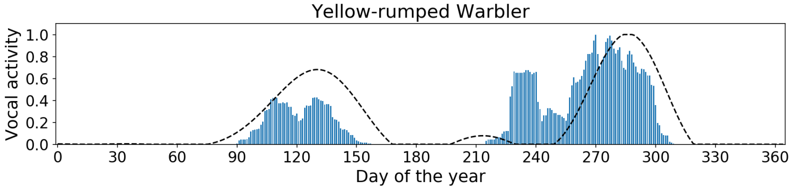
The high sensitivity and thus high range of the live stream microphone allows to investigate even more aspects of avian vocal behavior. Birds are known to adjust their vocal activity based on weather conditions [Robbins, 1981]. Low temperatures often lead to reduced activity [Garson and Hunter JR, 1979], [Hasan and Badri, 2016]. Despite the bias of a nearby bird feeder, this effect is observable with significant correlation in the live stream data, especially for months from October to March, when the effects of migration are minimal (see Figure 6.3).



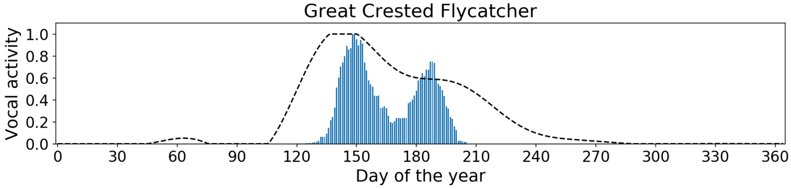
(a) Year-round, strong correlation



(b) Year-round, weak correlation



(c) Migratory, strong correlation



(d) Migratory, weak correlation

Figure 6.2.: Normalized relative vocal activity (blue bars) and normalized eBird checklist frequency (dashed line). The results of one year of live stream analysis show interesting vocal activity patterns when compared to the observed abundance. See Appendix A for more charts.

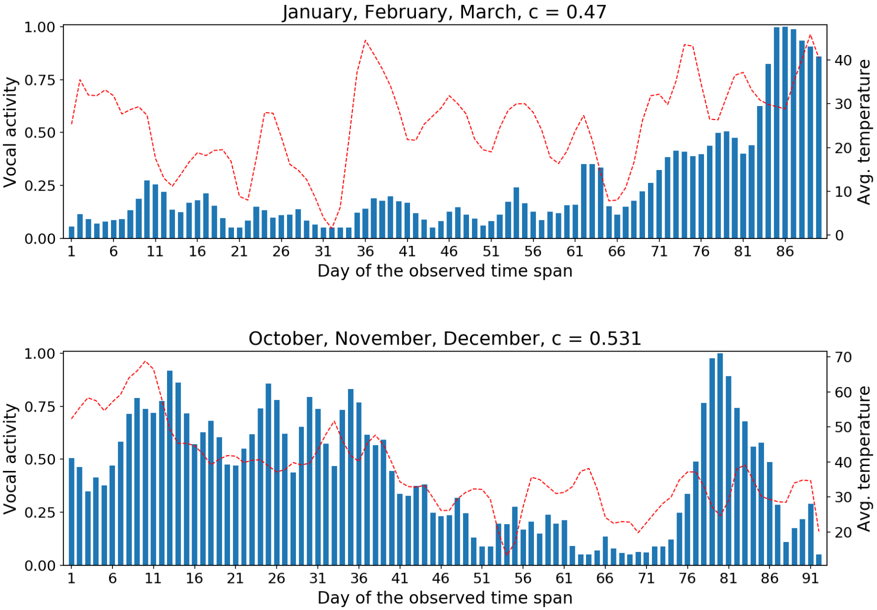


Figure 6.3.: Normalized relative vocal activity (blue bars) and average temperature in °F (red line). Avian vocal activity often strongly correlates (c, estimated with Pearson's r) with daily temperature variations.

Although the analysis data of live stream audio might contain false positive detections and a strong bias towards species that regularly visit bird feeders, automatically detected vocalizations indicate behavioral patterns that cannot easily be observed manually. Despite its primary use for presentation purposes, the live stream demo allows to investigate avian vocal activity in continuous audio streams. In the future, this application might help ornithologists to discover behavioral patterns linked to the social functions of bird vocalizations. Without any manual interference, we are now able to proof basic concepts of avian ecology and vocal behavior. Building on that, spatio-temporal patterns might emerge when investigating the results of multiple recording stations.

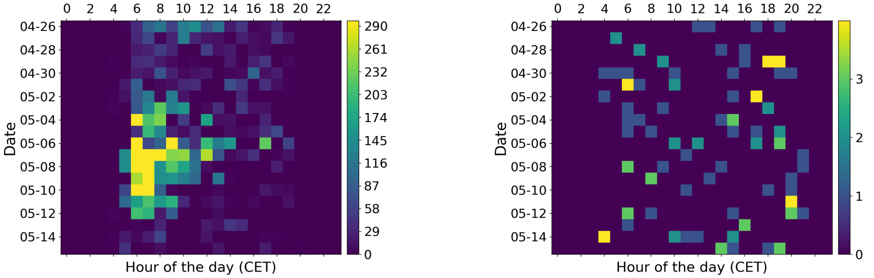
6.2. SWIFT soundscape analysis

The assessment of avian activity using point count surveys is an important aspect of avian ecology (see Section 2.3.2). Autonomous recording units are widely used but often require labor intensive, manual analysis, which renders this approach mostly unfeasible for long-term monitoring scenarios. BirdNET allows to quickly process large amounts of audio data to extract bird vocalizations. Since the survey scenario in avian ecology does not require to detect every single bird sound, the current detection rates are applicable (despite relatively low recall). Still, maintaining a larger number of recording stations requires on-site maintenance to swap batteries of SD cards. Reducing the number of recording units is less labor intensive but still provides insights into aspects of avian activity, especially the assessment diversity.

6.2.1. AMTiC

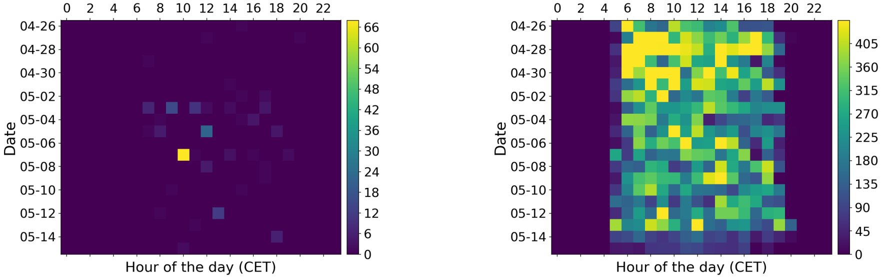
In 2019, SWIFT recording units were installed at the Tierpark Chemnitz (the local zoo) to estimate the number of free ranging bird species that occur in some of the many different (micro-) habitats (AMTiC - **A**coustic **M**onitoring at **Ti**erpark **C**hemnitz). Two primary locations were selected: First, a densely overgrown area surrounded by shrubs, large trees, and native hoofed animal enclosures (A1). Secondly, a location next to a shallow creek and pond, which are mainly inhabited by (captive) native aquatic birds but also attract a number of songbirds that prefer small water bodies (A2). Both locations often show very different avian profiles due to the distinct habitats they represent (see Figure 6.4 and Appendix B). Other recording locations focused on captive, tropical birds and are thus omitted here.

The results imply that the choice of location is crucial and generates a significant bias in species diversity estimation. During this monitoring scenario, the Common Chiffchaff (*Phylloscopus collybita*) has high detection rates at site A1 and is almost non-present at site A2 when analysing the audio recordings collected between April 26th and May 15th—three weeks of high avian activity in Central Europe. It appears likely that individuals at site A1 used the dense vegetation to establish a nest and breed, which significantly influences the amount of registered vocalizations. At site A2, a flock of House Sparrows (*Passer domesticus*) used the nearby maintenance buildings to feed on seeds, fruits, and grass prepared for other animals—a behavior that is typically displayed in zoos.



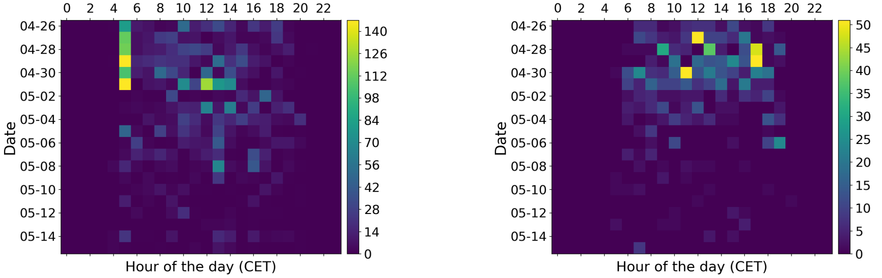
(a) Common Chiffchaff A1

(b) Common Chiffchaff A2



(c) House Sparrow A1

(d) House Sparrow A2



(e) Eurasian Blue Tit A1

(f) Eurasian Blue Tit A2

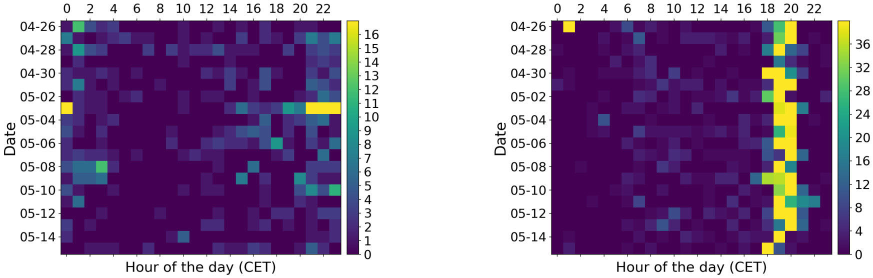
Figure 6.4.: Vocal activity of selected species recorded in Chemnitz. Recording station AMTiC 1 (A1) was surrounded by dense vegetation, AMTiC 2 (A2) was next to a shallow creek and pond.

Other species—like the Eurasian Blue Tit (*Cyanistes caeruleus*)—are equally present in the recordings of both monitoring stations and thus provide a more objective picture of vocal behavior over time. Most notably, Blue Tits appear to limit their vocal output with the end of April and are only rarely detected after that. The reasons for this observation might be natural (e.g. reduced mating and breeding activity) or technical when other species with high vocal activity mask the Blue Tits songs and calls. Increasing the number of recordings stations might help to investigate this aspect of avian behavior more closely.

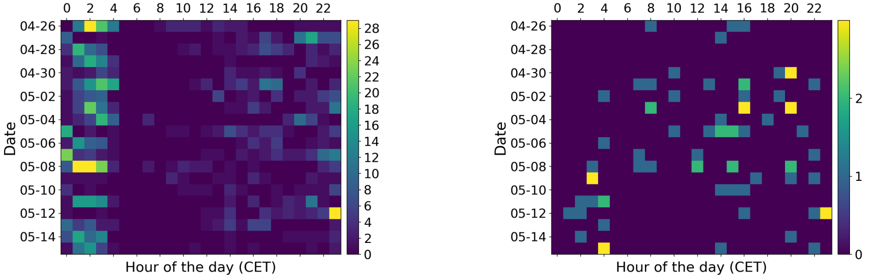
Comparing the detected vocal activity across multiple SWIFT recorders also reveals falsely detected species (see Figure 6.5). Depending on the monitoring location and the presence of nearby noise sources, BirdNET tends to consistently detect certain species that are not present. The patterns that occur in data plots over a selected period of time often show continuous detections with no clear distinction between day and night and a relatively low detection rate in general. Birds that vocalize in a lower frequency range (e.g. Owls and Doves) are often prone to produce false positives. During the AMTiC project, the Eurasian Eagle-Owl (*Bubo bubo*) and Tawny Owl (*Strix aluco*) were detected due to the noise of distant vehicles that emitted sounds similar to those of Owls with their tires (🔊 32). Although it can not be excluded that these species were present during the observed time, the results are most likely not applicable for further investigation.

Additionally, a group of captive Greater flamingos (*Phoenicopterus roseus*) appeared to be very vocal and audible in both recorders, which led to false positives of the Whooper Swan (*Cygnus cygnus*). Flamingos are not represented in the current version of BirdNET since they mostly occur in captivity. The confusion of native and non-native species has to be considered when analyzing monitoring data and still requires a fair amount of manual interference and post-processing to eliminate unwanted false positives. However, false detection rates are surprisingly low and the results contain valuable insights into the avian activity of certain habitats.

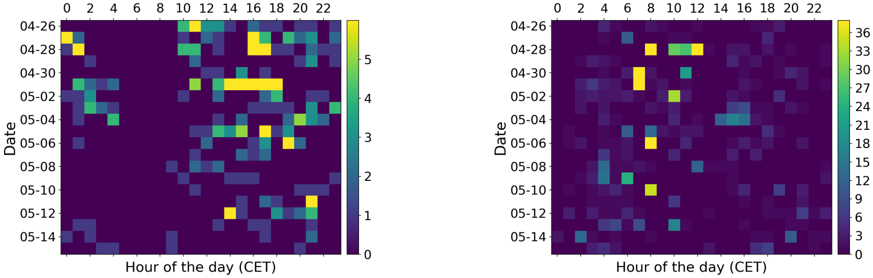
Additionally, a number of temporal patterns emerge when analyzing the detection results. This includes clearly visibly daily cycles of activity between dawn and dusk, as well as changes in vocal output over the course of several weeks and—most importantly—breeding season and migration. A single recording unit suffices and already displays distinct activity patterns for most species.



(a) Eurasian Eagle-Owl A1 (b) Eurasian Eagle-Owl A2



(c) Tawny Owl A1 (d) Tawny Owl A2



(e) Whooper Swan A1 (f) Whooper Swan A2

Figure 6.5.: Examples for common false detections in soundscape data. Most predators and some aquatic birds are falsely detected when triggered by ambient noise or unknown bird species.

6.2.2. SWAMP

Activity patterns are usually amplified when multiple tens of SWIFT recorders form an array to closely monitor an area through point count surveys. Since 2017, the Sapsucker Woods Bird Sanctuary in Ithaca, NY, USA is object of acoustic investigation to assess changes in species abundance over a long period of time (SWAMP - **S**apsucker **W**oods **A**coustic **M**onitoring **P**roject, see Figure 6.6).

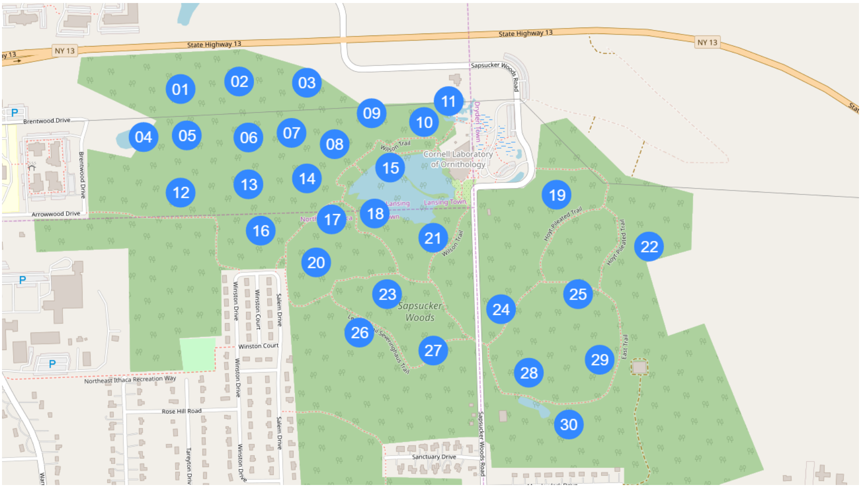


Figure 6.6.: SWAMP recording sites. An array consisting of 30 SWIFT recorders continuously monitors an area that approximately spans 0.6 x 0.9 mi (900 x 1,500 m) and generates more than 100 GB of audio data per day. Maps generated with Leaflet and OpenStreetMap tiles.

Depending on the use case, soundscape analysis of recorded audio reveals spatio-temporal patterns that are almost impossible to achieve without automated analysis. Some of the most interesting aspects of vocal behavior are changes in the vocal output over the course of a day. Due to different habitat structures, the vocal output at different recording sites is likely to vary significantly. The analysis of soundscapes recorded with the SWAMP array in May 2017 reveals exactly this: Species abundance and vocal activity differ depending on location and time of the day, which is visible in a number of consecutive heat maps representing the total amount of hourly detections for every recording unit (Figures 6.7 and 6.8).

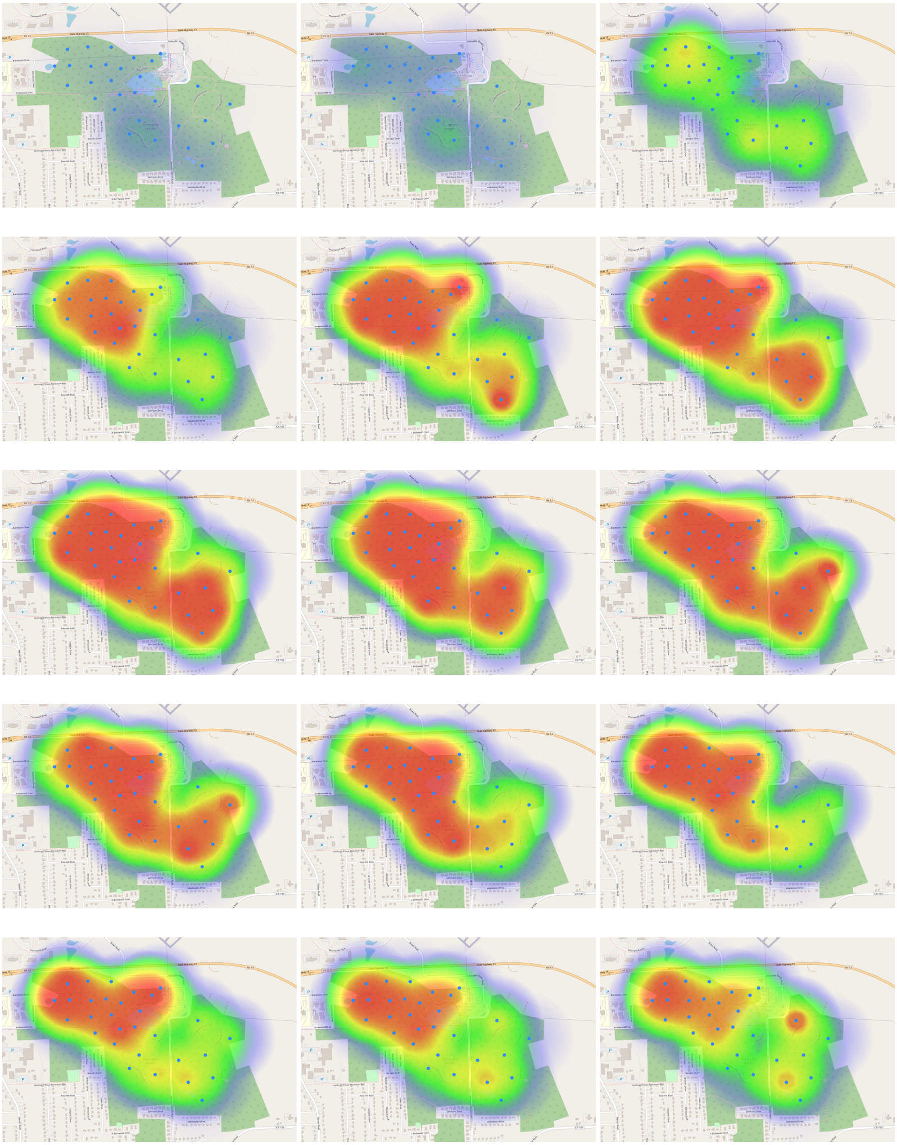


Figure 6.7.: Spatio-temporal patterns of hours 1 to 15.

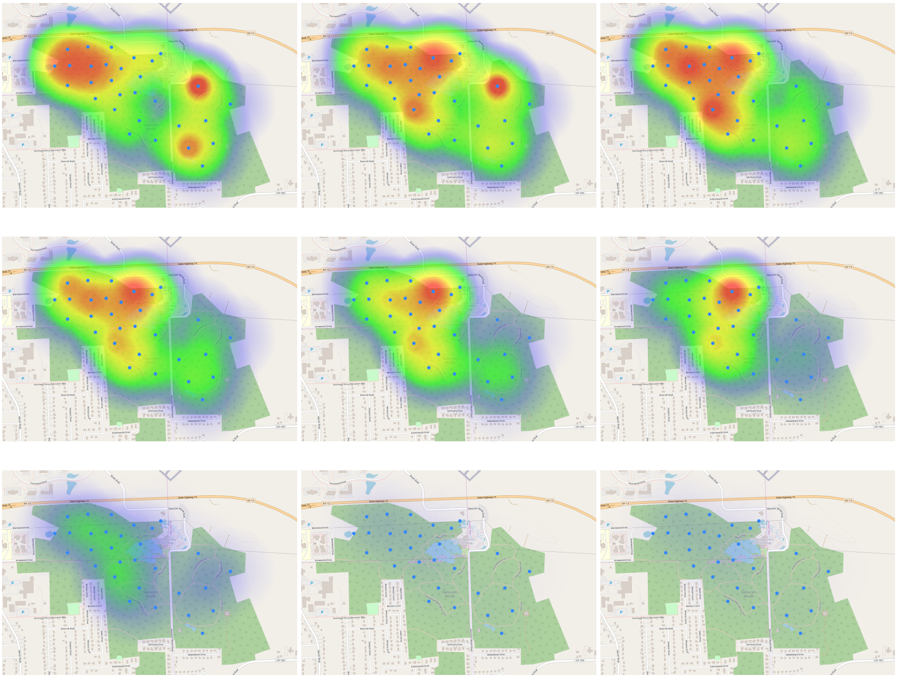
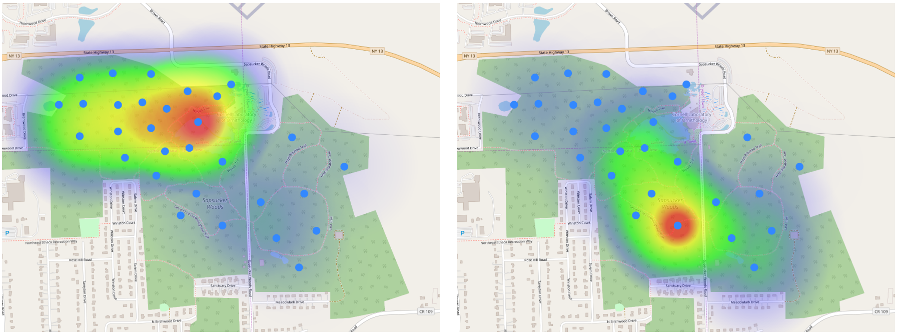


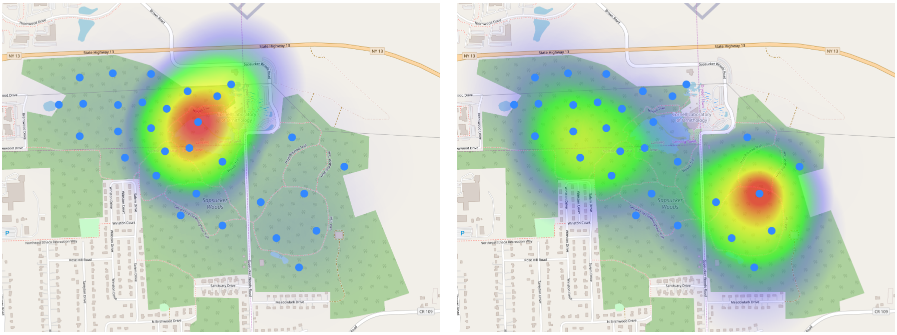
Figure 6.8.: Spatio-temporal patterns of hours 16 to 24. Avian vocal activity increases until 4 a.m., peaks between 6 a.m. and 10 a.m., and decreases until 9 p.m. with only minor activity during the night. Average across the month of May 2017, all times in EDT (UTC-4).

Although this insight is not completely new, the resolution and scale at which these results can be derived is unprecedented. Over the course of a day in May, the (normalized) average hourly vocal activity is greatest at the recording locations 1-20. Despite the nearby highway, avian activity in the Sapsucker Woods appears to be more dependent on (micro-) habitat composition.

Stand-out recording locations include unit 4, 11, 15, and 30 that are located near water bodies with high numbers of aquatic birds (mostly Mallards and Canada Geese), as well as unit 19 with high Blue Jay activity. SWIFT units 21, 23, 24, 26, and 27 recorded visibly less vocalizations—probably due to the fact that this area spans a



(a) Black-capped Chickadee (b) Wood Thrush



(c) Canada Goose (d) Great Crested Flycatcher

Figure 6.9.: Spatial maps of normalized absolute species abundance in May 2017. Most species only occur at specific locations throughout the Sapsucker Woods area. Some recorders do not pick up a single vocalization of a particular species—which implies that habitat preferences limit the range even within a relatively small observed area. For more maps see Appendix C.

net of trails that are regularly used by birdwatchers. Yet, these patterns indicate a correlation between species abundance and environmental factors, but they do not necessarily imply the specific causality of changes in vocal behavior. We can only assume that the presence of humans and their structures are avoided by some species. Dense vegetation often offers more breeding locations and better protection against predators and is therefore favoured by many species—a circumstance that is also visible in the observational data.

Specific habitat characteristics and species-specific preferences become even more apparent when investigating spatial maps of species distribution within the Sapsucker Woods area (see Figure 6.9). These maps are generated by accumulating every detection for each species at each site. The resulting maps indicate the normalized vocal activity in May 2017 and often show a very limited range despite the relatively small observed area. Black-capped Chickadees appear to prefer remote forests covered by site 1-18, the Great Crested Flycatcher also occurs East of Sapsucker Woods Road (site 25), the Wood Thrush prominently occurs alongside the Wilson Trail (site 27), and Canada Geese (unsurprisingly) prefer larger water bodies (site 15). Again, the analysis data implies a correlation between habitat and vocal activity. It does not necessarily indicate species abundance and it allows only limited reasoning about causalities. However, an increased number of monitoring stations allows to derive spatio-temporal patterns that gain more details with every additional recorder.

6.3. Smartphone app

Mobile applications for avian activity monitoring can be a transformative tool to assess species diversity on a global scale over a long period of time. From a scientific point of view, public involvement to gain vast amounts of observational data can provide otherwise unattainable perspectives and insights into some of the most complex avian behavioral patterns. The success of eBird with millions of submitted checklists and observations, sound snippets, and photos supports this assumption. Additionally, public participation and scientific communication could open the domain of machine learning to a completely new audience. Bird watchers, citizen scientists, and an interested public are likely to adapt new technologies that address environmental issues. Smart devices that run mobile applications are widely spread and often provide an advanced technology platform that is well-suited for this purpose.

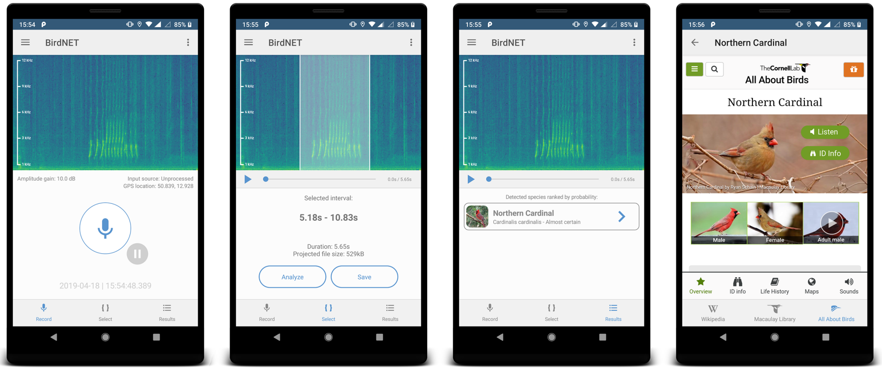
6.3.1. Scope

The BirdNET smartphone app is primarily intended to serve as a learning tool. It targets bird watchers and beginners alike but mainly focuses on providing knowledge about local bird species. With the combination of sound visualization, interpretability, and added details about birds from various knowledge bases, the BirdNET app intends to teach the process of birding by ear. Yet, it does not contain any tutorials and specific insights into the vast diversity of bird vocalizations. Learning to identify birds by ear is guided solely by automated recognition. Public involvement in the assessment of species abundance and vocal activity is simply achieved by the intrinsic motivation of learning. Whereas eBird checklists require (expert) knowledge about local bird species, an automated recognition system only requires basic reasoning. This often eases the difficulty of providing reliable observational data to contribute to the environmental efforts by researchers.

When viewed as recording station, each mobile device becomes a valuable addition to the vast grid of conducted point counts. Despite the occasional use of the app to identify a bird species, long-term data series might emerge due to immersive usability. It appears plausible that those data series can help to gain new insights into species abundance, diversity, and vocal activity. The latter becomes one of the most interesting aspects of large-scale observation. Since the automated detection is limited to sound, an assessment of species presence reflects how often and how prominently a species is audible. In contrast to other point count surveys that rely on both modalities—sight and sound—the retrieved observation data allows to exclusively correlate vocal activity with other environmental factors like habitat characteristics, weather, or migration. We can assume that users of the smartphone app tend to be very selective in the sounds they analyze. This supposedly amplifies the effect of vocal presence and might shift the focus to uncommon or very vocal species.

6.3.2. Design

The disproportionate importance of spectrograms for avian research was at the center of the design process for the BirdNET smartphone app. Visualizing sounds is an important cornerstone of interpretability. Therefore, one key element of the interface design is a real-time (rolling) spectrogram view that instantly visualizes



Amplitude gain: 10.0 dB

Input source: Unprocessed
GPS location: 50.839, 12.928

Selected interval:
5.18s - 10.83s

Duration: 5.65s
Projected file size: 529kB

(a) Record (b) Select (c) Analysis (d) Details

Figure 6.10.: BirdNET app screenshots. The interface serves as a visual metaphor of a portable recording device and guides users through the process of recording, selection, and analysis. Additional knowledge bases like Wikipedia and AllAboutBirds.org provide more information on the identified species.

sounds recorded by the mobile device (see Figure 6.3.2). The entire interface serves as a visual metaphor for a portable recording device with its rather technical appearance. User interaction is guided through a simple process of recording, selection, and analysis that is repeated for every new identification. Observing the spectrogram view provides indications about the overall recording quality and helps to isolate the sounds that should be analyzed. This process avoids a major drawback of acoustic recognition—low signal quality—by eliminating unwanted noise and overlapping vocalizations. Additionally, the visual perception of bird sounds helps to reason whether a detection is plausible or not. It might also reduce the frustration when the app is not confident about the identified species. This process is supported with verbalizations of confidence scores through terms like ‘almost certain’, ‘probable’, ‘highly uncertain’ or even ‘wild guess’.

The presentation of results allows to further investigate the identified species. Included resources contain web content derived from Wikipedia, the Macaulay Library, and AllAboutBirds.org (for North American species only). The additional content includes textual descriptions, sound samples, and pictures that are intended to support the learning process. Observations are recorded on a web server that also handles the load of the analysis process. On-device detection would be technically feasible, a centralized approach however has certain advantages. First, updating models and thus providing better results does not rely on user interaction. Secondly, an increase of computational resources allows to apply larger and deeper networks, and finally, the reduced technical requirements of a client allow the app to be ported to a wide range of devices. The BirdNET web API handles the request/response workload and distributes incoming observations across several analysis workers. Again, single model performance is key to quickly process a potentially large number of requests.

6.3.3. Distribution

Starting with a rough prototype, the development of the app began in 2017 and only featured a 'Record' button that accessed the device microphone when clicked. Following the design of very common recording scenarios like voice messages, the recording duration was determined by the time the button was pressed. First results were unsatisfying. Despite good overall performance in previous evaluations, the detection rate was extremely low and unreliable. Due to that, the process of recording was complemented with a real-time spectrogram view that then allowed to isolate bird vocalizations. After a testing stage with a selected group of frequent users, the app was officially released to the Google Play Store in September 2018. Without any advertisement or paid content, the number of active users steadily grew only through specific, topic-related search in the Play Store. At the end of 2018, a total number of 1.414 active users had installed the app[3]. These numbers began to rise quickly and surpassed 100,000 active installs on May, 27th. At the end of July 2019, more than 250,000 users had BirdNET installed on their mobile device. BirdNET is free of charge and available for Android devices only.

---

[3] Active users (or active installs) mark the amount of devices that currently have installed the app. Retention rates for this app are approximately 70% over the entire live-span.

6.3.4. Reception

With an average rating of 4.344 based on 843 reviews[4], the BirdNET smartphone app significantly outperforms similar apps in the same category. Mobile song ID applications include *BirdGenie: ID Birds by Song* by Princeton University Press with a rating of 1.4 across 21 reviews, and *Song Sleuth: Auto Bird Song ID w/ David Sibley* by Wildlife Acoustics, Inc. with a rating of 2.2 across 13 reviews. *Naturblick* by the Museum für Naturkunde Berlin also features an automated sound recognition component and is rated at 3.9 across 341 reviews. Sunbird Images distributes a mobile app for bird identification that also contains acoustic recognition called *Bird Song Id: Auto Recognition* that received an average rating of 4.1 across 381 reviews.

Most notably, textual feedback for BirdNET notes a wide variety of aspects that were intended during the development of the app. Some of the more extensive reviews reflect on the use of the app to learn about bird vocalizations, the visual selection of sounds, the response to non-events (e.g. human sounds), and raised awareness for bird encounters:

> *First birdsong recognition app that really works. Most birds in our backyard spend most of their time in trees, heard but not seen, and* ***it's great to learn who's singing****.*
>
> *Simple and brilliant. Helped me narrow a sound down to a chiffchaff -* ***a sound I have been wondering about for years.*** *THANKS!!!!*
>
> *Works pretty well. I tried it on some birds that i already recognized and it got it right. Nice interface in that you can just keep it recording and then* ***visually select the sound bite you want*** *to analyze.*
>
> *It's a mistake to take BirdNet on a walk if you're out for exercise!* ***You're too tempted to stop and identify birds****. The app is fascinating and seems to work very well.*
>
> *Does an amazing job at listening to audio and choosing what it thinks the noise could be (****Including humans making bird noises!****)*

---

[4] All scores were retrieved on 2019-07-31

> *I've been using this app for a couple of weeks now, sooooo* ***useful and fun!*** *It clearly identifies birds, enabling for me to see and photograph ones that* ***I'd never have spotted without this app****. Love love it!*

Additionally, the app has sparked national and international media attention, which underlines the appeal of an automated recognition and learning tool. User feedback was used to improve the app by adding features like a detailed observation overview or by adjusting the visual interface to provide a better user experience and acoustic guidance for visually impaired users. The app also supports local common names in 12 languages and a localized interface in English, German, French, Czech, and Dutch.

6.3.5. Results

Between September 2018 and July 2019, BirdNET users submitted more than 2.9 million recordings with a total duration of 6,021 hours containing more than 2 million observations[5]. Due to the lack of an iOS version and the popularity of Android outside the USA, European users submitted more than twice as many observations than users in North America. From months April to July 2019, the observation density is sufficient to generate spatio-temporal patterns of avian activity on a continental scale. Yet, the dataset contains strong biases due to some circumstances that have to be taken into account:

- The app is available in selected countries only including Central and Western Europe, as well as the United States and Canada.
- People in densely populated areas use the app more frequently and thus submit more observations.
- Predictions are post-filtered according to eBird checklist frequencies.
- Over time, more people start using the app, which steadily increases the number of observations.
- People might not submit an observation of a species they were able to previously identify with the app.

---

[5] A valid observation has to contain GPS coordinates, a confidence score of more than 1.5% and must not feature a non-event class.

- The number of observations significantly varies with time (e.g. on weekends) and weather conditions (e.g. summer heat waves).
- Observations might contain false positives due to class confusion that leads to more detections for the confused species.

Considering these limitations, analyzing the submitted data requires to eliminate or at least to reduce some of the most apparent biases. Relative values that are normalized over time present a more objective view but are still highly dependent on the number of users in an area. Data interpolation and moving averages can help to eliminate outliers and gaps that arise from spatio-temporal variations. Two main categories of avian activity can be investigated using BirdNET submission: Vocal activity over time and location, as well as avian diversity on a large scale.

Vocal activity

Estimating the vocal activity of a bird species based on observations submitted via smartphone app might indicate the overall abundance on a global (or at least continental) scale. Vocal activity itself is linked to avian behavioral patterns, especially breeding and migration. The absolute number of detections of a species indicates how prominently the bird is perceived. Assuming that smartphone users tend to analyze (and thus submit) observations when they occur very frequently, the amount of detected vocalizations in relation to the total number of all recognized songs and calls might imply how much a species stands out—which can be due to very high vocal activity. When visualized over time (see Figure 6.11) for Europe, species like the Eurasian Wren (*Troglodytes troglodytes*) and Common Nightingale (*Luscinia megarhynchos*) show peaks in their (perceived) vocal activity with up to 12% of all submitted vocalizations.

With multiple tens of thousands of detections, these two species are an example of birds with complex, clearly audible vocalizations that are often conspicuous and not easily identifiable by the average listener. When encountered, smartphone users are apparently very likely to submit a vocalization. Due to this, we can observe changes in vocal activity over time with relatively dense data points independent from the number of users. Despite the fact that vocal activity plots imply migration patterns, we cannot necessarily conclude when and where the activity was highest. Again, analyzing spatio-temporal patterns (Figures 6.12 and 6.13) provides more detail.

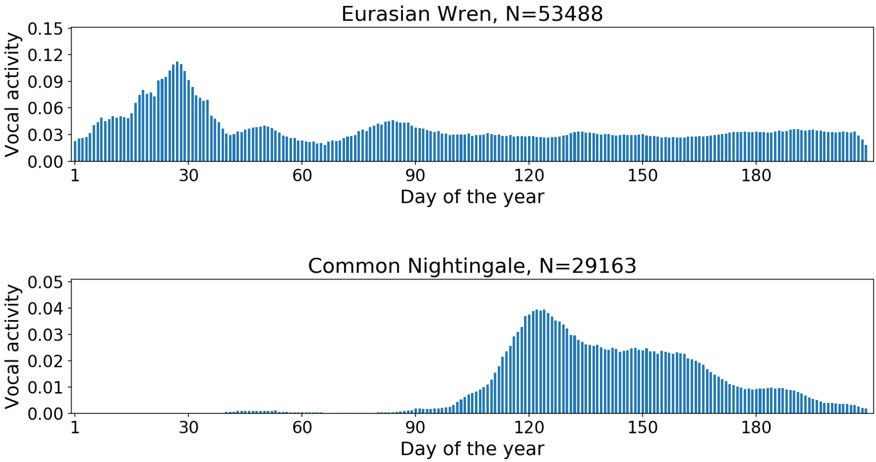
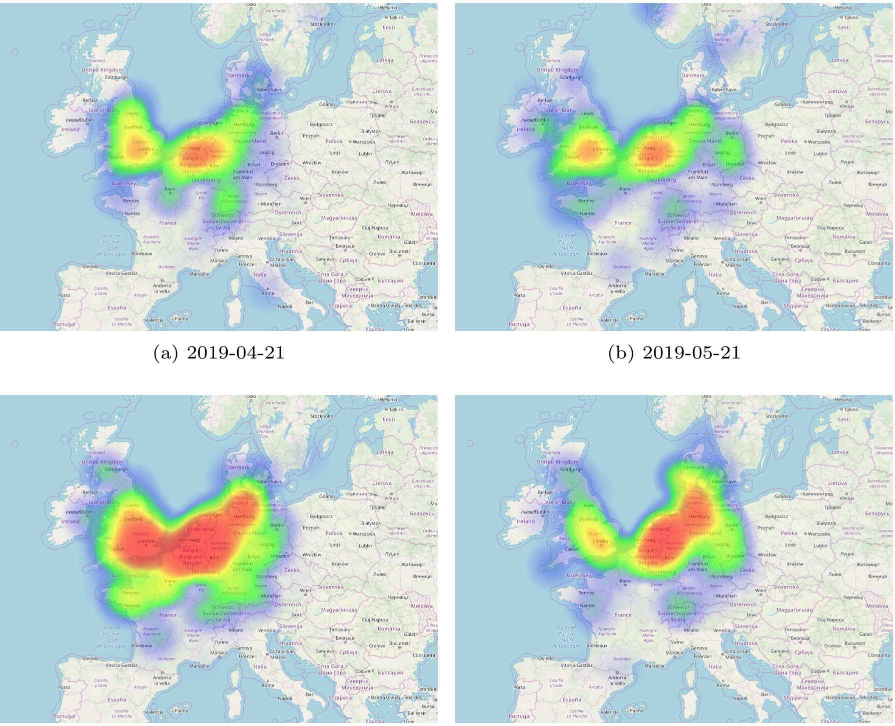


Figure 6.11.: Relative vocal activity in Europe between January and July 2019 for two very conspicuous species.

These patterns reveal two main insights: First, spatial abundance changes over time and does not necessarily reflect the relative vocal activity. Both visualizations have to be treated complementary. The occurrence of the Eurasian Wren peaks in June, whereas the highest relative vocal activity was measured in January when many migrating birds are missing. Secondly, vocal activity can be used to assess species distribution. The spatial abundance of the Common Nightingale peaks in May with distinct distribution patterns and is significantly reduced after that. This pattern matches the relative vocal activity and both dimensions provide valuable insights into habitat preferences across Europe. Still, we have to assume that both visualizations would profit from an increased number of observations and lack considerable amounts of details in the current form. Monitoring submissions over several years might reveal more conclusive patterns for even more species. The number of users during the time of spring migration in 2019 was not sufficient to assess movements with a satisfying level of detail and additional observations are needed.



(a) 2019-04-21

(b) 2019-05-21

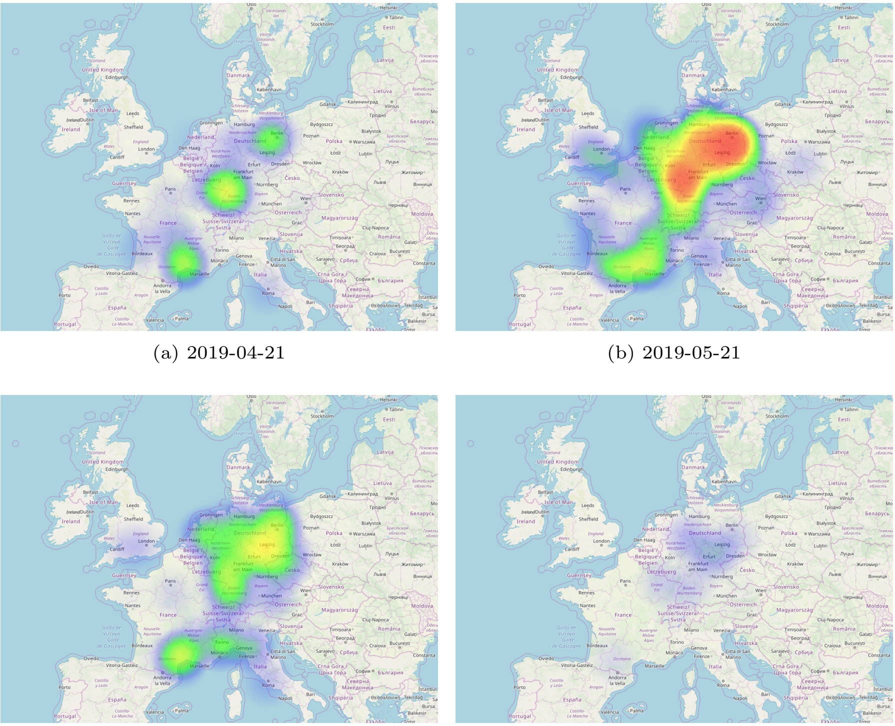
(c) 2019-06-20

(d) 2019-07-20

Figure 6.12.: Spatio-temporal patterns of relative, normalized vocal activity of the Eurasian Wren.

Avian diversity

In addition to the vocal activity of single species, the diversity of birds for an area is another important dimension of avian ecology. On a continental scale, avian diversity indicates behavioral patterns that are mostly linked to migration. To some extend, this measure also implies species composition when stationary and migratory species are somewhat predictable. Again, spatio-temporal patterns can be derived from large-scale data over long periods of time (see Figure 6.14). Yet, the spatial scale of species diversity does not necessarily allow to draw conclusions

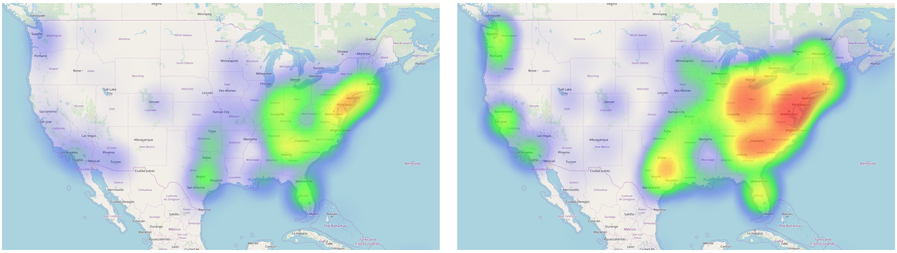


(a) 2019-04-21 (b) 2019-05-21

(c) 2019-06-20 (d) 2019-07-20

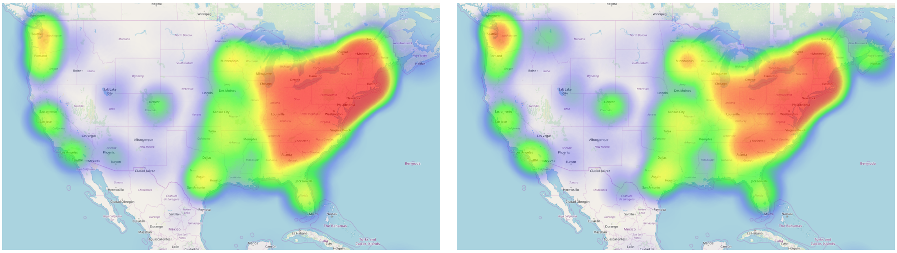
Figure 6.13.: Spatio-temporal patterns of relative, normalized vocal activity of the Common Nightingale.

whether specific habitats are more attractive than others. The data points provided by users of the BirdNET smartphone app contain strong biases and lack the required density—at least for the first half of 2019. A strong emphasis on densely populated areas leads to a strong correlation between urban environments and the expected species diversity for a specific location. The mid-west of the United States apparently lacks a sufficient number of submissions. Interpolation and weighting of observations could resolve this issue, still, only long-term data will provide the required details.



(a) Week 15, 240 species

(b) Week 18, 317 species



(c) Week 21, 346 species

(d) Week 24, 367 species

Figure 6.14.: Spatio-temporal patterns of normalized species diversity in the USA. A strong bias towards densely populated areas results in strong correlation between avian diversity and urban environments. The effects of peak migration (usually between weeks 17-20) are still clearly visible.

Temporal patterns provide a more objective look at changes in avian diversity. Between the end of April and the first weeks of May (weeks 17-20), migration usually reaches its peak across North America. Despite a correlation between the number of users and the number of detected species, the effect of migration on the diversity of species is clearly visible in these patterns. Again, urban areas profit from a high number of data points, but the overall diversity index remains high even for some rural areas. In addition to eBird checklist data, those results could provide a valuable dimension of real-time species diversity assessments.

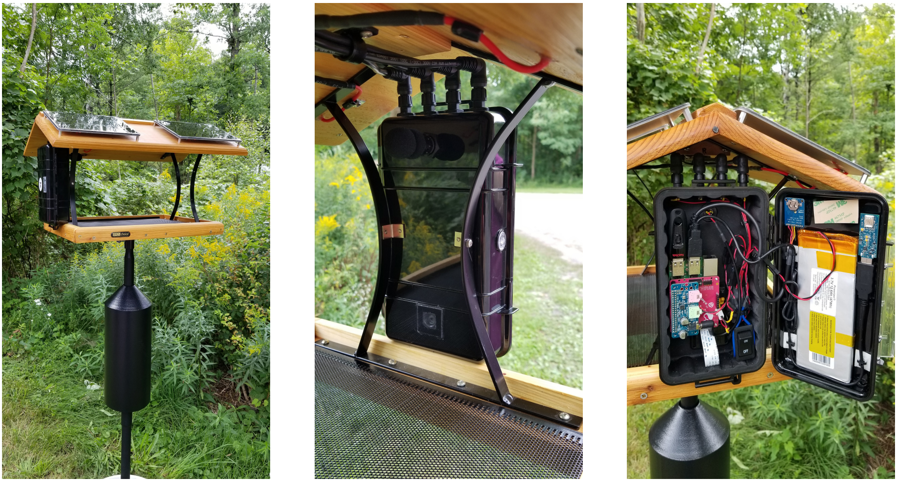
Outlook

Since its release in September 2018, the BirdNET smartphone app has sparked the interest of many bird watchers in North America and Europe. Due to a high number of active users, the app provides a significant amount of observational data that can be used to analyze avian behavioral patterns that are linked to vocal activity. Due to its nature as a learning tool, users of the app do not necessarily document every encounter and only submit vocalizations that are conspicuous and not easily recognizable by the average listener. Strong biases towards the number of users, urban areas and species selection based on personal interests, often lead to unwanted correlations that disturb the occurring patterns. However, the app has already proven to be a valuable tool for highly distributed, long-term monitoring of avian activity. Over time, more and more users will start using the app and provide data that can be used to study annual effects of migration, habitat changes, and climate change—among others.

6.4. HaikuBox

As an autonomous recording and analysis station, the HaikuBox prototype is intended to serve as link between SWIFT recorders and the BirdNET smartphone app. When distributed across the United States at selected locations, HaikuBoxes will provide daily assessments of species occurrences based on real-time analysis. Due to the fixed position and non-selective audio analysis, spatio-temporal data is expected to compensate some of the drawbacks of mobile apps and recorder arrays that lack either range or resolution. The on-device execution of the bird sound recognition significantly reduces the maintenance overhead. Recorded audio data is analyzed 'on the fly' and results are stored in textual form, which is periodically sent to a central server. The entire station is solar powered, a Raspberry Pi provides the computing platform. Yet, the power consumption of the mobile computer is too high to allow full-day assessments. Therefore, each HaikuBox station only records one hour before and one hour after sunrise—during the dawn chorus. The remainder of daylight is used to charge the Lithium battery.

The fly-through bird feeder serves two main purposes: First, it attracts birds year-round and provokes vocalizations due to social interaction of individuals. Secondly, we can assume that people are more likely to maintain a recording station when they



(a) Feeder unit (b) Microphone and camera (c) Interior

Figure 6.15.: HaikuBox prototype attached to a fly-through bird feeder that attracts local bird species and provokes a high number of vocalizations due to social interaction. On-device analysis using mobile DNN architectures reduces the maintenance overhead of swapping SD cards, solar panels charge the Lithium battery.

can observe backyard birds feeding. The production of a small series of HaikuBoxes will start with the end of 2019 after a testing stage that will provide insights into the applicability of mobile DNN architectures for this use case. Again, the dual use as learning tool and scientific monitoring station aims at democratizing the process of bird watching to sensitize a wide audience to the effects of environmental changes on avian activity.

6.5. Summary

Public demos, prototypes, and applications allow bird watchers and scientists to gain insights into behavioral patterns linked to avian activity. In contrast to traditional point counts, those applications provide data at large scale—spatially and temporally. Correlating effects of environmental changes to migration, breeding, social interaction, diversity, and activity can be observed due to dense data points that originate from recording arrays (AMTiC, SWAMP), highly-distributed mobile recorders (smartphones), or single-mic stations (HaikuBox, live stream microphones). Processing large-scale data comes at the cost of significantly increased computing power. BirdNET provides applicable and fast deep neural networks for avian activity monitoring for all those scenarios. Involving the public in the process of data acquisition sparks the interest of many users that are eager to submit observations when using designated learning tools. These tools combine both, scientific assessments and democratized technology to reach a wide audience. The applications presented in this chapter mark the starting point of public services that allow users to record, analyze, and submit observations independent of their profession. BirdNET will help to expand these efforts by including more species, continents, and task-specific applications in the near future.

7. Conclusion and Future Work

Birds are meaningful to many people and are a common source of sound for humans. Many households maintain a bird feeder during the winter to prevent backyard birds from starving and to cherish the beauty and elegance of local species. Birds are omnipresent, often reveal their presence through their vocalizations, and they respond to various environmental changes over many spatial scales. Therefore, they are ideal indicator species to monitor environmental changes in habitats across all lifeforms and to identify early warning signs that indicate habitat changes (see Section 2.3). Automated observation of avian activity to assess vocal activity and species diversity can be a transformative tool for ornithologists, conservation biologists, and bird watchers to assist in long-term monitoring of critical environmental niches.

Communication between individuals of birds is not limited to sound, but the avian vocal tract and auditory system are highly developed. For many applications of bird identification and observation, sound is the primary source of information. However, avian vocalizations are highly complex and often consist of rapid successions of elements and notes. Most birds emit sound to communicate—either with their vocal tract or non-vocally using other body parts (see Section 2.2.2). *Passeriformes* is the largest order of birds and contains the suborder of oscine passerines—true songbirds. The evolution of song in oscines is complex and requires extensive learning, imitation, and even improvisation. The intra-species heterogeneity of song repertoires is vast with species being able to sing multiple hundreds of songs per individual. Variations in time and space add to that diversity and render the automated identification of avian vocalizations an extremely difficult task (see Section 2.2.4).

Digital sound transformation is commonly used when studying bird sounds. Since the inception of the sound spectrograph, spectrograms play a significant role in avian research (see Section 2.2.1). We can assume that visual representations of bird sounds contain valuable information on species identity, rendering spectrograms a particularly suitable representation. The worldwide community of bird watchers provides vast archives of digital sound recordings for almost every bird species on the planet. These recordings are mostly of high quality and were recorded with (semi-)

professional and often highly directional equipment (see Section 2.3.2). Autonomous recordings units to monitor ecological niches and habitats mostly use omnidirectional microphones, which usually leads to significant levels of ambient noise. Adapting to this shift in acoustic domains is critical when training an automated recognition system based on publicly available audio data.

Deep artificial neural networks (DNN) have surpassed traditional classifiers like Gaussian mixture models, decision trees or support vector machines in the field of acoustic event recognition. This progress has transformed the the two largest evaluation campaigns for avian sound identification and led to very competitive results in that domain (see Section 3.3). Still, deep neural networks require expert knowledge to design, train, and test powerful models—a process that is often guided by intuition due to the holistic nature of hyperparameter tuning. Additionally, dataset bias and lack of generalization are among the main concerns when applying neural networks to real-world use cases (see Section 3.2.1). With these constraints and the requirements of future applications in mind, an extensive toolkit for automated avian activity monitoring was developed: BirdNET.

The implementation of this toolkit was based on a number of distinct design decisions including extensive functionality, detailed configuration, a domain-agnostic workflow, transparent and reproducible implementations, as well as an application-driven development process. BirdNET employs the overall workflow of detailed data handling, audio processing capabilities, extensive data augmentation, and dynamic model design and export. Due to those features, BirdNET is a research platform that allows to design and evaluate sophisticated training regimes of deep neural networks for acoustic event recognition (see Chapter 4).

The experimental studies in this dissertation evolved around the ability of DNN to generalize on unseen samples despite a high number of classes with significant intra-class heterogeneity. The investigation focused on key components and computational processes under fair conditions in order to obtain largely generalizable results (see Chapter 5). An unprecedented amount of training, validation, and test data was used to assess the overall system performance on more then 3,900 hours of field recordings covering 987 classes and almost 300 hours of fully annotated soundscapes containing almost 80,000 vocalizations. The resulting benchmark system yields state-of-the-art scores across all domains, especially when compared to recent advances during the 2019 BirdCLEF challenge. With an increase of 15.4% over the best single model designed for the 2019 soundscape evaluation dataset that did not use validation samples for training, the proposed training regime and DNN design appear to be competitive considering the difficulty of the task (see Section 5.2.7).

Avian activity monitoring often requires labor intensive interference to conduct point counts or to analyze large amounts of audio data. BirdNET can not only help to reduce the amount of manual work in that domain, it also provides new, valuable insights into some of the most basic avian behavioral patterns (see Chapter 6). The assessment of avian vocal activity and diversity gains an important dimension of investigation with spatio-temporal visualizations of observational data acquired through long-term monitoring using autonomous recorders or massive amounts of submissions using a smartphone app. Future developments of BirdNET will focus on expanding that dimension by providing fast and reliable acoustic recognition for various monitoring scenarios.

Future work

BirdNET is already advanced and provides good overall performance for many use cases. However, soundscape analysis still poses a significant challenge with scores that leave considerable room for improvements. Future progress will focus on enhance those scores to expand BirdNET to a variety of monitoring scenarios. The system's road map for the next few years includes the following developments:

Training with less samples: The current results presented in this dissertation imply that deep neural networks are indeed extremely data hungry and require vast amounts of training samples. Additionally, the overall quality of those samples has to be very high to prevent decreased performance. Both circumstances are somewhat unsatisfying. Pre-processing of noisy samples could lead to improved performance due to the elimination of falsely labeled data. This process would either require manual interference—which is not desirable—or an automated assessment. In both cases, the amount of training samples will decrease. Considering the difficulties of obtaining audio recordings of rare or endangered species, this limitation holds considerable weight. Certain strategies to cope with these difficulties have been proposed. Most notably, triplet loss introduced in [Weinberger and Saul, 2009] helps to learn feature embeddings in DNN so that similar data points are closer to each other. Focusing on similarities rather than categories might help to cope with weak labels and an ever expanding number of classes.

The effectiveness of this approach has been demonstrated in the domain of person identification ([Hermans et al., 2017]) but was also successfully applied to scenarios in the domain of bioacoustics [Thakur et al., 2019]. The potentially reduced need for large amounts of training samples using this technique is very promising.

Source separation: Dawn chorus recordings often suffer from a cacophony of sounds that overlap. In the current state, BirdNET is capable to resolve some overlapping scenarios. Yet, this mostly comes as the result of increased sensitivity, which itself often leads to high numbers of false positives. Separating sound sources has become a valuable tool in the field of audio analysis [Ewert et al., 2014], [Rivet et al., 2014]. Ambient recording with two or more microphones can ease the difficulties when distinguishing different individuals of birds. The effect of masked vocalizations might be reducible when processing multi-channel recordings. The second generation of SWIFT recorders will support stereophonic monitoring and the current analysis workflow of BirdNET already supports other than one-channel inputs. The domain shift between mono-species, directional field recordings, and soundscapes persists in this scenario, however, stereo training samples can be synthesized from mono-channel audio [Orban, 1970].

Visual attention: In 2017, Vaswani et al. proposed that *Attention is all you need* [Vaswani et al., 2017]. Intended for sequence transduction, the approach demonstrated that shifting the attention of a trained DNN to salient parts of the input data can help to achieve significantly better results. Complementary to that, the concept of visual attention has been applied to solve a variety of tasks in the past (e.g. [Xu et al., 2015]). This even applies for the domain of acoustic event recognition—especially bird sounds—as part of the BirdCLEF 2017 and 2018 challenge [Sevilla and Glotin, 2017], [Schlüter, 2018]. Still, attention mechanisms in the domain of image recognition are not widely used. Implementing visual attention would require some modifications to the code base of BirdNET but could potentially lead to increased performance for extremely noisy data.

Switching back ends: Since the discontinuation of Theano in 2017, other deep learning frameworks like PyTorch and TensorFlow became increasingly popular. Both frameworks are widely adopted in the scientific community and provide vast functionality without the need to implement complex training workflows. Most importantly, the ability of those frameworks to port trained models to different target platforms becomes a central advantage compared to older toolkits.

Today, model export natively supports mobile devices such as smartphones, Raspberry Pi (and other ARM architectures), web browsers, and even embedded systems. In the future, tensor processing units (TPU) will accelerate the processes of training and inference—specialized software however is required. Therefore, future versions of BirdNET will be based on TensorFlow due to the large community and vast ecosystem of functionalities.

Expanding the API: Providing researchers with transformative tools for automated avian monitoring is one of the main goals of BirdNET. With its applications and demos, the presented system is capable of processing large amounts of data in real-time. Still, soundscape recordings are often hour-long and need considerable amounts of computational resources when processed. In the future, BirdNET will include a cloud computing infrastructure that allows to distribute the audio analysis workflow across multiple workstations. Similar to the eBird web API, soundscape analysis will be available to the scientific community through an expanded web interface. With *BirdNET as a service*, more researchers will be able to use the tools created as part of this dissertation.

Improving the smartphone app: The data analysis in Section 6.3 showed that the density of submitted observations did not yet suffice to derive spatio-temporal patterns that indicate migrational movements. The smartphone app will gain value for the scientific community as a tool that provides insights into avian behavioral patterns on a global scale with an increasing number of users. In order to achieve that, the app will be maintained and expanded in terms of functionality, supported devices, and user experience. Future developments will include an iOS version that hopefully leads to broader usage across North America. On-device recognition is already technically feasible and will eventually find its way into the app.

Adjusting to South America: The focus of this dissertation was on North American and European species only. The assessment of the system's performance is easier when mobile recorders can be maintained regularly and local bird watchers provide valuable insights into their work. The current monitoring scenario is already extremely complex and some of the remaining issues (e.g. overlap in soundscapes) have to be resolved before applying BirdNET to acoustic monitoring scenarios as part of conservational efforts. Tropical environments are highly endangered and often contain multiple thousands of bird species. Acoustic monitoring in such environments requires portable analysis devices that are extremely power efficient, waterproof, and can communicate over long distances.

The computational constraints that those devices employ contradict the increased number of species and vast avian diversity in those areas. Future versions of BirdNET will adjust to those circumstances and might be deployed in the South American rain forest in the next few years.

The comprehensive evaluation of deep learning techniques for avian activity monitoring in this thesis only marks the starting point of an expanded ecosystem of functionalities and applications for the field of bioacoustics. Due to its domain-agnostic workflow, BirdNET is not just limited to birds but can be applied to almost any acoustic monitoring scenario. Most prominently, that includes the detection of marine mammals, insects, or fish in highly endangered habitats. Advances in soft- and hardware will help to provide highly automated tools to cope with environmental issues of our future.

Bibliography

[Adams, 1987] Adams, D. (1987). *Dirk Gently's Holistic Detective Agency.* Simon and Schuster.

[Andrew, 1962] Andrew, R. J. (1962). Evolution of intelligence and vocal mimicking. *Science*, 137(3530):585–589.

[Awad et al., 2016] Awad, G., Snoek, C. G., Smeaton, A. F., and Quénot, G. (2016). Trecvid semantic indexing of video: A 6-year retrospective. *ITE Transactions on Media Technology and Applications*, 4(3):187–208.

[Barker et al., 2013] Barker, J., Vincent, E., Ma, N., Christensen, H., and Green, P. (2013). The pascal chime speech separation and recognition challenge. *Computer Speech & Language*, 27(3):621–633.

[Beecher, 2008] Beecher, M. D. (2008). Function and mechanisms of song learning in song sparrows. *Advances in the Study of Behavior*, 38:167–225.

[Beletsky, 1989] Beletsky, L. (1989). Communication and the cadence of birdsong. *American Midland Naturalist*, pages 298–306.

[Berg et al., 2006] Berg, K. S., Brumfield, R. T., and Apanius, V. (2006). Phylogenetic and ecological determinants of the neotropical dawn chorus. *Proceedings of the Royal Society of London B: Biological Sciences*, 273(1589):999–1005.

[Berger et al., 2015] Berger, A., Eibl, M., Heinich, S., Herms, R., Kahl, S., Kürsten, J., Kurze, A., Manthey, R., Rickert, M., Ritter, M., et al. (2015). ValidAX-Validierung der Frameworks AMOPA und XTRIEVAL. *Chemnitzer Informatik-Berichte.*

[Berger et al., 2018] Berger, F., Freillinger, W., Primus, P., and Reisinger, W. (2018). Bird audio detection - DCASE 2018. Technical report, DCASE2018 Challenge.

[Bergstra et al., 2010] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy).* Oral Presentation.

[Bibby et al., 2004] Bibby, C. J. et al. (2004). *Bird ecology and conservation: A handbook of techniques*, volume 1, chapter Bird diversity survey methods. Oxford University Press.

[Billings, 2018] Billings, A. C. (2018). The low-frequency acoustic structure of mobbing calls differs across habitat types in three passerine families. *Animal Behaviour*, 138:39–49.

[Bogue, 2013] Bogue, R. (2013). Recent developments in MEMS sensors: A review of applications, markets and technologies. *Sensor Review*, 33(4):300–304.

[Boncoraglio and Saino, 2007] Boncoraglio, G. and Saino, N. (2007). Habitat structure and the evolution of bird song: A meta-analysis of the evidence for the acoustic adaptation hypothesis. *Functional Ecology*, 21(1):134–142.

[Brackenbury, 1982] Brackenbury, J. H. (1982). *Acoustic Communication in Birds - Volume 1 - Production, Perception, and Design Features of Sound*, chapter The Structural Basis of Voice Production and Its Relationship to Sound Characteristics. Academic Press.

[Bregman, 1994] Bregman, A. S. (1994). *Auditory scene analysis: The perceptual organization of sound.* MIT press.

[Briggs et al., 2013] Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S. F., Hadley, A., Betts, M., Fern, X. Z., et al. (2013). The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *2013 IEEE international workshop on machine learning for signal processing (MLSP)*, pages 1–8. IEEE.

[Brooks Jr, 1996] Brooks Jr, F. P. (1996). The computer scientist as toolsmith II. *Communications of the ACM*, 39(3):61–68.

[Buda et al., 2018] Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

[Butterworth, 1930] Butterworth, S. (1930). On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541.

[Byers and Kroodsma, 2016] Byers, B. E. and Kroodsma, D. E. (2016). *Handbook of bird biology*, chapter Avian Vocal Behavior. John Wiley & Sons.

[Cakir et al., 2017] Cakir, E., Adavanne, S., Parascandolo, G., Drossos, K., and Virtanen, T. (2017). Convolutional recurrent neural networks for bird audio detection. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1744–1748. IEEE.

[Carignan and Villard, 2002] Carignan, V. and Villard, M.-A. (2002). Selecting indicator species to monitor ecological integrity: A review. *Environmental monitoring and assessment*, 78(1):45–61.

[Carter et al., 2019] Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. (2019). Activation atlas. *Distill*. https://distill.pub/2019/activation-atlas.

[Catchpole, 1973] Catchpole, C. K. (1973). The functions of advertising song in the sedge warbler (Acrocephalus schoenobaenus) and the reed warbler (A. scirpaceus). *Behaviour*, 46(3):300–319.

[Catchpole and Slater, 2008] Catchpole, C. K. and Slater, P. J. (2008). *Bird song: Biological Themes and Variations (2nd edition)*. Cambridge University Press.

[Chandrasekhar et al., 2011] Chandrasekhar, V., Sharifi, M., and Ross, D. (2011). Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications. In *12th International Society for Music Information Retrieval Conference (ISMIR)*.

[Chen et al., 2005] Chen, J., Kam, A. H., Zhang, J., Liu, N., and Shue, L. (2005). Bathroom activity monitoring based on sound. In *International Conference on Pervasive Computing*, pages 47–61. Springer.

[Cherry, 1953] Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979.

[Chilton and Lein, 1996] Chilton, G. and Lein, M. R. (1996). Song repertoires of puget sound white-crowned sparrows zonotrichia leucophrys pugetensis. *Journal of Avian Biology*, pages 31–40.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[Chollet, 2017a] Chollet, F. (2017a). *Deep learning with Python*. Manning Publications Company.

[Chollet, 2017b] Chollet, F. (2017b). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

[Chollet et al., 2015] Chollet, F. et al. (2015). Keras. `https://keras.io`.

[Chou and To, 2018] Chou, J. and To, C.-H. (2018). Cocktail party problem for bird sounds. *CS230, Stanford University*.

[Chu et al., 2009] Chu, S., Narayanan, S., and Kuo, C.-C. J. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1142–1158.

[Clark and Feo, 2008] Clark, C. J. and Feo, T. J. (2008). The Anna's hummingbird chirps with its tail: A new mechanism of sonation in birds. *Proceedings of the Royal Society of London B: Biological Sciences*, 275(1637):955–962.

[Constantine, 2006] Constantine, M. (2006). *The sound approach to birding: A guide to understanding bird sound*. The Sound Approach.

[Cooley and Tukey, 1965] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

[Cowling and Sitte, 2003] Cowling, M. and Sitte, R. (2003). Comparison of techniques for environmental sound recognition. *Pattern recognition letters*, 24(15):2895–2907.

[Cramp and Simmons, 1977] Cramp, S. and Simmons, K. (1977). *Handbook of the Birds of Europe, the Middle East and North Africa: Ostrich to Ducks*. Birds of the Western Palearctic I-VI : Ostrich to Ducks. Oxford University Press.

[Cubuk et al., 2018] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). AutoAugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

[Dieleman et al., 2015] Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S. K., Nouri, D., et al. (2015). Lasagne: First release. DOI: 10.5281/zenodo.27878.

[Dobson and Lemon, 1975] Dobson, C. W. and Lemon, R. E. (1975). Re-examination of monotony threshold hypothesis in bird song. *Nature*, 257(5522):126.

[Dooling, 1982] Dooling, R. J. (1982). *Acoustic Communication in Birds - Volume 1 - Production, Perception, and Design Features of Sound*, chapter Auditory Perception in Birds. Academic Press.

[Dooling, 2004] Dooling, R. J. (2004). *Nature's Music*, chapter Audition: Can birds hear everything they sing? Elsevier.

[Dooling et al., 2000] Dooling, R. J., Lohr, B., and Dent, M. L. (2000). *Hearing in birds and reptiles*. Springer.

[Downie, 2008] Downie, J. S. (2008). The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255.

[Dumoulin and Visin, 2016] Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.

[Dunn and Alderfer, 2017] Dunn, J. L. and Alderfer, J. (2017). *Field guide to the birds of North America*. National Geographic Books.

[eBird, 2019] eBird (2019). Engaging birders in science and conservation. `https://ebird.org/about`. Accessed: 2019-02-14.

[Eronen et al., 2006] Eronen, A. J., Peltonen, V. T., Tuomi, J. T., Klapuri, A. P., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. (2006). Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329.

[Ewert et al., 2014] Ewert, S., Pardo, B., Müller, M., and Plumbley, M. D. (2014). Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124.

[Eyben et al., 2010] Eyben, F., Wöllmer, M., and Schuller, B. (2010). OpenSMILE: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM.

[Fallon, 2007] Fallon, R. (2007). *OLIVIER MESSIAEN: Music, Art and Literature*, chapter The Record of Realism in Messiaen's Bird Style. Ashgate.

[Falls and Brooks, 1975] Falls, J. B. and Brooks, R. J. (1975). Individual recognition by song in white-throated sparrows. II. Effects of location. *Canadian Journal of Zoology*, 53(10):1412–1420.

[Fant, 1968] Fant, G. (1968). Analysis and synthesis of speech processes. *Manual of phonetics*, 2:173–277.

[Farnsworth, 2005] Farnsworth, A. (2005). Flight calls and their value for future ornithological studies and conservation research. *The Auk*, 122(3):733–746.

[Fink et al., 2018] Fink, D., Auer, T., Johnston, A., Strimas-Mackey, M., Iliff, M., and Kelling, S. (2018). eBird Status and Trends. Version: November 2018.

[Fitzpatrick and Rodewald, 2016] Fitzpatrick, J. W. and Rodewald, A. D. (2016). *Handbook of bird biology*, chapter Bird Conservation. John Wiley & Sons.

[Furlanello et al., 2018] Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born again neural networks. *arXiv preprint arXiv:1805.04770*.

[Garson and Hunter JR, 1979] Garson, P. J. and Hunter JR, M. L. (1979). Effects of temperature and time of year on the singing behaviour of wrens Troglodytes troglodytes and great tits Parus major. *Ibis*, 121(4):481–487.

[Gemmeke et al., 2017] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). AudioSet: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE.

[Glasberg and Moore, 1990] Glasberg, B. R. and Moore, B. C. (1990). Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1-2):103–138.

[Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

[Glotin et al., 2013a] Glotin, H., Clark, C., LeCun, Y., Dugan, P., Halkias, X., and Sueur, J. (2013a). Proc. 1st workshop on machine learning for bioacoustics - ICML4B. USA. ICML int. Conf. http://sabiod.univ-tln.fr.

[Glotin et al., 2013b] Glotin, H., LeCun, Y., Artières, T., Mallat, S., Tchernichovski, O., and Halkias, X. (2013b). Proc. neural information processing scaled for bioacoustics, from neurons to big data. USA. NIPS Int. Conf. http://sabiod.org/nips4b.

[Goëau et al., 2016] Goëau, H., Glotin, H., Vellinga, W.-P., Planqué, R., and Joly, A. (2016). LifeCLEF bird identification task 2016: The arrival of deep learning. In *CLEF (Working Notes)*, volume 1609, pages 440–449.

[Goëau et al., 2017] Goëau, H., Glotin, H., Vellinga, W.-P., Planqué, R., and Joly, A. (2017). LifeCLEF bird identification task 2017. In *CLEF (Working Notes)*, volume 1866.

[Goëau et al., 2014] Goëau, H., Glotin, H., Vellinga, W.-P., Planqué, R., Rauber, A., and Joly, A. (2014). LifeCLEF bird identification task 2014. In *CLEF (Working Notes)*, volume 1180, pages 585–597.

[Goëau et al., 2015] Goëau, H., Glotin, H., Vellinga, W.-P., Planqué, R., Rauber, A., and Joly, A. (2015). LifeCLEF bird identification task 2015. In *CLEF (Working Notes)*, volume 1391.

[Goëau et al., 2018] Goëau, H., Kahl, S., Glotin, H., Vellinga, W.-P., Planqué, R., and Joly, A. (2018). Overview of BirdCLEF 2018: Monospecies vs. sundscape bird identification. In *CLEF (Working Notes)*, volume 2125.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Gregory et al., 2004] Gregory, R. D., Gibbons, D. W., and Donald, P. F. (2004). *Bird ecology and conservation: A handbook of techniques*, volume 1, chapter Bird census and survey techniques. Oxford University Press.

[Grill and Schlüter, 2017] Grill, T. and Schlüter, J. (2017). Two convolutional neural networks for bird detection in audio signals. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1764–1768. IEEE.

[Halfwerk et al., 2011] Halfwerk, W., Holleman, L. J., Slabbekoorn, H., et al. (2011). Negative impact of traffic noise on avian reproductive success. *Journal of applied Ecology*, 48(1):210–219.

[Hansen, 1979] Hansen, P. (1979). Vocal learning: Its role in adapting sound structures to long-distance propagation, and a hypothesis on its evolution. *Animal Behaviour*.

[Hasan and Badri, 2016] Hasan, N. M. and Badri, M. (2016). Effect of ambient temperature on dawn chorus of house sparrows. *Environment and Ecology Research*, 4(3):161–168.

[Hausberger et al., 1991] Hausberger, M., Jenkins, P. F., and Keene, J. (1991). Species-specificity and mimicry in bird song: Are they paradoxes? A reevaluation of song mimicry in the European starling. *Behaviour*, 117(1):53–81.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[He et al., 2016a] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[He et al., 2016b] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

[He et al., 2019] He, Y., Sainath, T. N., Prabhavalkar, R., McGraw, I., Alvarez, R., Zhao, D., Rybach, D., Kannan, A., Wu, Y., Pang, R., et al. (2019). Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6381–6385. IEEE.

[Heinzel et al., 2002] Heinzel, G., Rüdiger, A., and Schilling, R. (2002). Spectrum and spectral density estimation by the discrete fourier transform (dft), including a comprehensive list of window functions and some new at-top windows.

[Hermans et al., 2017] Hermans, A., Beyer, L., and Leibe, B. (2017). In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

[Hinton et al., 2012a] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012a). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[Hinton et al., 2012b] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[Hu and Cardoso, 2009] Hu, Y. and Cardoso, G. C. (2009). Are bird species that vocalize at higher frequencies preadapted to inhabit noisy urban areas? *Behavioral Ecology*, 20(6):1268–1273.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Huang et al., 2016] Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

[Hubel and Wiesel, 1962] Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

[Hultsch and Todt, 2004] Hultsch, H. and Todt, D. (2004). *Nature's Music*, chapter Learning to sing. Elsevier.

[Ignatov et al., 2018] Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., and Van Gool, L. (2018). AI benchmark: Running deep neural networks on Android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) 2018*, pages 288–314.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[Irwin, 2000] Irwin, D. E. (2000). Song variation in an avian ring species. *Evolution*, 54(3):998–1010.

[Jarvis, 2004] Jarvis, E. D. (2004). *Nature's Music*, chapter Brains and birdsong. Elsevier.

[Joly et al., 2014] Joly, A., Champ, J., and Buisson, O. (2014). Instance-based bird species identification with undiscriminant features pruning. In *CLEF (Working Notes)*, volume 1180, pages 625–633.

[Joly et al., 2015] Joly, A., Leveau, V., Champ, J., and Buisson, O. (2015). Shared nearest neighbors match kernel for bird songs identification-LifeCLEF 2015 challenge. In *CLEF (Working Notes)*, volume 1391.

[Kahl et al., 2017a] Kahl, S., Hussein, H., Fabian, E., Schloßhauer, J., Thangaraju, E., Kowerko, D., and Eibl, M. (2017a). Acoustic event classification using convolutional neural networks. *INFORMATIK 2017*.

[Kahl et al., 2017b] Kahl, S., Richter, D., Roschke, C., Heinzig, M., Kowerko, D., Eibl, M., and Ritter, M. (2017b). Technische Universität Chemnitz and Hochschule Mittweida at TRECVID Instance Search 2017. In *Proceedings of TRECVID Workshop*.

[Kahl et al., 2016] Kahl, S., Roschke, C., Rickert, M., Richter, D., Zywietz, A., Hussein, H., Manthey, R., Heinzig, M., Kowerko, D., Eibl, M., et al. (2016). Technische Universität Chemnitz at TRECVID Instance Search 2016. In *Proceedings of TRECVID Workshop*.

[Kahl et al., 2017c] Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M., and Eibl, M. (2017c). Large-scale bird sound classification using convolutional neural networks. In *CLEF (Working Notes)*, volume 1866.

[Kahl et al., 2018a] Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., and Eibl, M. (2018a). A baseline for largescale bird species identification in field recordings. In *CLEF (Working Notes)*, volume 2125.

[Kahl et al., 2018b] Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., and Eibl, M. (2018b). Recognizing birds from sound-the 2018 BirdCLEF baseline system. *arXiv preprint arXiv:1804.07177*.

[Kaiser et al., 2017] Kaiser, L., Gomez, A. N., and Chollet, F. (2017). Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.

[Karpathy, 2014] Karpathy, A. (2014). What I learned from competing against a ConvNet on ImageNet. *http://karpathy. github. io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet*, 2. Accessed: 2019-03-05.

[Khan et al., 2017] Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., and Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Koenig, 2016] Koenig, W. D. (2016). *Handbook of bird biology*, chapter Ecology of Bird Populations. John Wiley & Sons.

[Koops et al., 2014] Koops, H. V., Van Balen, J., and Wiering, F. (2014). A deep neural network approach to the lifeclef 2014 bird task. In *CLEF (Working Notes)*, volume 1180, pages 634–642.

[Krause, 1993] Krause, B. L. (1993). The niche hypothesis: A virtual symphony of animal sounds, the origins of musical expression and the health of habitats. *The Soundscape Newsletter*, 6:6–10.

[Krebs, 1981] Krebs, J. R. (1981). Effect of removal of mate on the singing behaviour of great tits. *Anim. Behav*, 29:635–637.

[Krizhevsky and Hinton, 2009] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kroodsma, 1988] Kroodsma, D. E. (1988). *Evolution and learning*, chapter Contrasting styles of song development and their consequences among passerine birds. Lawrence Erlbaum Associates, Inc.

[Kroodsma, 2004] Kroodsma, D. E. (2004). *Nature's Music*, chapter The diversity and plasticity of birdsong. Elsevier.

[Kroodsma, 2005] Kroodsma, D. E. (2005). *The singing life of birds: The art and science of listening to birdsong*. Houghton Mifflin Harcourt.

[Kroodsma et al., 1997] Kroodsma, D. E., Houlihan, P. W., Fallon, P. A., and Wells, J. A. (1997). Song development by grey catbirds. *Animal behaviour*, 54(2):457–464.

[Kroodsma and Konishi, 1991] Kroodsma, D. E. and Konishi, M. (1991). A suboscine bird (Eastern phoebe, Sayornis phoebe) develops normal song without auditory feedback. *Animal Behaviour*, 42(3):477–487.

[Kurakin et al., 2016] Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

[Kürsten, 2012] Kürsten, J. (2012). *A generic approach to component-level evaluation in information retrieval*. PhD thesis, Chemnitz University of Technology.

[Lasseck, 2014] Lasseck, M. (2014). Large-scale identification of birds in audio recordings. In *CLEF (Working Notes)*, volume 1180, pages 643–653.

[Lasseck, 2015] Lasseck, M. (2015). Improved automatic bird identification through decision tree based feature selection and bagging. In *CLEF (Working Notes)*, volume 1391.

[Lasseck, 2016] Lasseck, M. (2016). Improving bird identification using multiresolution template matching and feature selection during training. In *CLEF (Working Notes)*, volume 1609, pages 490–501.

[Lasseck, 2018a] Lasseck, M. (2018a). Acoustic bird detection with deep convolutional neural networks. Technical report, DCASE2018 Challenge.

[Lasseck, 2018b] Lasseck, M. (2018b). Audio-based bird species identification with deep convolutional neural networks. In *CLEF (Working Notes)*, volume 2125.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Leng et al., 2014] Leng, Y. R., Dennis, J. W., and Dat, T. H. (2014). Bird classification using ensemble classifiers. In *CLEF (Working Notes)*, volume 1180, pages 654–661.

[Lewis, 2011] Lewis, J. (2011). Microphone specifications explained. *AN-1112, Analog Devices*.

[Lin et al., 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

[Lin et al., 2017] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

[Loshchilov and Hutter, 2016] Loshchilov, I. and Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[Lostanlen et al., 2019] Lostanlen, V., Salamon, J., Cartwright, M., McFee, B., Farnsworth, A., Kelling, S., and Bello, J. P. (2019). Per-channel energy normalization: Why and how. *IEEE Signal Processing Letters*, 26(1):39–43.

[Lostanlen et al., 2018] Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., and Bello, J. P. (2018). Birdvox-full-night: A dataset and benchmark for avian flight call detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 266–270. IEEE.

[Lovette, 2016] Lovette, I. J. (2016). *Handbook of bird biology*, chapter Avian Diversity and Classification. John Wiley & Sons.

[Lynch, 1995] Lynch, J. F. (1995). *Monitoring bird populations by point counts*, volume 149, chapter Effects of point count duration, time-of-day, and aural stimuli on detectability of migratory and resident bird species in Quintana Roo, Mexico, pages 1–6. US Department of Agriculture, Forest Service, Pacific Southwest Research Station.

[Lyon, 2017] Lyon, R. F. (2017). *Human and machine hearing*. Cambridge University Press.

[Maas et al., 2013] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, page 3.

[MacArthur, 1958] MacArthur, R. H. (1958). Population ecology of some warblers of northeastern coniferous forests. *Ecology*, 39(4):599–619.

[Macaulay, 2019] Macaulay (2019). The world's premier scientific archive of natural history audio, video, and photographs. https://www.macaulaylibrary.org/about/history/. Accessed: 2019-02-14.

[Marler, 2004] Marler, P. (2004). *Nature's Music*, chapter Science and birdsong: The good old days. Elsevier.

[Marler and Isaac, 1960] Marler, P. and Isaac, D. (1960). Physical analysis of a simple bird song as exemplified by the chipping sparrow. *The Condor*, 62(2):124–135.

[Marler and Sherman, 1983] Marler, P. and Sherman, V. (1983). Song structure without auditory feedback: Emendations of the auditory template hypothesis. *Journal of Neuroscience*, 3(3):517–531.

[Marler and Sherman, 1985] Marler, P. and Sherman, V. (1985). Innate differences in singing behaviour of sparrows reared in isolation from adult conspecific song. *Animal Behaviour*, 33(1):57–71.

[Martinez et al., 2014] Martinez, R., Silva, L., Olvera, T. E. V., Fuentes, G., and Ruíz, I. V. M. (2014). SVM candidates and sparse representation for bird identification. In *CLEF (Working Notes)*, volume 1180, pages 662–669.

[Mesaros et al., 2018] Mesaros, A., Heittola, T., Benetos, E., Foster, P., Lagrange, M., Virtanen, T., and Plumbley, M. D. (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.

[Meza et al., 2015] Meza, I., Espino-Gamez, A., Solano, F., and Villarreal, E. (2015). A fast baseline system for large scale bird identification. In *CLEF (Working Notes)*, volume 1391.

[Morton, 1975] Morton, E. S. (1975). Ecological sources of selection on avian sounds. *The American Naturalist*, 109(965):17–34.

[Müller and Marti, 2018] Müller, L. and Marti, M. (2018). Bird sound classification using a bidirectional LSTM. In *CLEF (Working Notes)*, volume 2125.

[Müller, 2018] Müller, S. (2018). *Systematisierung und Identifizierung von Störquellen und Störerscheinungen in zeithistorischen Videodokumenten am Beispiel digitalisierter Videobestände sächsischer Lokalfernsehsender*. PhD thesis, Chemnitz University of Technology.

[Nelson, 1988] Nelson, D. A. (1988). Feature weighting in species song recognition by the field sparrow (Spizella pusilla). *Behaviour*, 106(1):158–181.

[Northcott, 2014] Northcott, J. (2014). Participation of group SCS to LifeCLEF bird identification challenge 2014. In *CLEF (Working Notes)*, volume 1180, pages 670–672.

[Nottebohm, 1972] Nottebohm, F. (1972). The origins of vocal learning. *The American Naturalist*, 106(947):116–140.

[Olah et al., 2017] Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*. DOI: 10.23915/distill.00007.

[Olah et al., 2018] Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*. DOI: 10.23915/distill.00010.

[Orban, 1970] Orban, R. (1970). A rational technique for synthesizing pseudo-stereo from monophonic sources. *Journal of the Audio Engineering Society*, 18(2):157–164.

[Otter and Ratcliffe, 1993] Otter, K. and Ratcliffe, L. (1993). Changes in singing behavior of male black-capped chickadees (Parus atricapillus) following mate removal. *Behavioral Ecology and Sociobiology*, 33(6):409–414.

[Pamuła et al., 2017] Pamuła, H., Kłaczyński, M., Remisiewicz, M., Wszołek, W., and Stowell, D. (2017). Adaptation of deep learning methods to nocturnal bird audio monitoring. *Bird Migration Research Foundation.*

[Park et al., 2019] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779.*

[Pellegrini, 2017] Pellegrini, T. (2017). Densely connected CNNs for bird audio detection. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1734–1738. IEEE.

[Peltonen et al., 2002] Peltonen, V., Tuomi, J., Klapuri, A., Huopaniemi, J., and Sorsa, T. (2002). Computational auditory scene recognition. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–1941. IEEE.

[Peterson, 2010] Peterson, R. T. (2010). *Field Guide to Birds of Eastern and Central North America.* Houghton Mifflin Harcourt.

[Piczak, 2016] Piczak, K. J. (2016). Recognizing bird species in audio recordings using deep convolutional neural networks. In *CLEF (Working Notes)*, volume 1609, pages 534–543.

[Pieplow, 2017] Pieplow, N. (2017). *Field Guide to Bird Sounds of Eastern North America.* Houghton Mifflin Harcourt.

[Pinheiro and Collobert, 2015] Pinheiro, P. O. and Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721.

[Podos, 1997] Podos, J. (1997). A performance constraint on the evolution of trilled vocalizations in a songbird family (Passeriformes: Emberizidae). *Evolution*, 51(2):537–551.

[Raboshchuk et al., 2015] Raboshchuk, G., Jančovič, P., Nadeu, C., Lilja, A. P., Kóküer, M., Mahamud, B. M., and Veciana, A. R. d. (2015). Automatic detection of equipment alarms in a neonatal intensive care unit environment: A knowledge-based approach. In *Sixteenth Annual Conference of the International Speech Communication Association.*

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

[Ralph et al., 1995] Ralph, C. J., Droege, S., and Sauer, J. R. (1995). *Monitoring bird populations by point counts*, volume 149, chapter Managing and monitoring birds using point counts: Standards and applications, pages 161–168. US Department of Agriculture, Forest Service, Pacific Southwest Research Station.

[Ramachandran et al., 2017] Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

[Rey and Wender, 2011] Rey, G. D. and Wender, K. F. (2011). *Neuronale Netze-Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung. 2. Auflage*. Verlag Hans Huber.

[Richards and Wiley, 1980] Richards, D. G. and Wiley, R. H. (1980). Reverberations and amplitude fluctuations in the propagation of sound in a forest: Implications for animal communication. *The American Naturalist*, 115(3):381–399.

[Ritter, 2014] Ritter, M. (2014). *Optimierung von Algorithmen zur Videoanalyse: Ein Analyseframework für die Anforderungen lokaler Fernsehsender*. PhD thesis, Chemnitz University of Technology.

[Rivet et al., 2014] Rivet, B., Wang, W., Naqvi, S. M., and Chambers, J. A. (2014). Audiovisual speech source separation: An overview of key methodologies. *IEEE Signal Processing Magazine*, 31(3):125–134.

[Robbins, 1981] Robbins, C. S. (1981). Bird activity levels related to weather. *Studies in avian biology*, 6:301–310.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). ImageNet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[Sabour et al., 2017] Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

[Salamon and Bello, 2017] Salamon, J. and Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283.

[Sánchez and Perronnin, 2011] Sánchez, J. and Perronnin, F. (2011). High-dimensional signature compression for large-scale image classification. In *CVPR 2011*, pages 1665–1672. IEEE.

[Sandler et al., 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[Schlüter, 2017] Schlüter, J. (2017). *Deep learning for event detection, sequence labelling and similarity estimation in music signals*. PhD thesis, Johannes Kepler University Linz.

[Schlüter, 2018] Schlüter, J. (2018). Bird identification from timestamped, geotagged audio recordings. In *CLEF (Working Notes)*, volume 2125.

[Schmidhuber, 2013] Schmidhuber, J. (2013). My first deep learning system of 1991+ deep learning timeline 1962-2013. *arXiv preprint arXiv:1312.5548*.

[Selvaraju et al., 2017] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

[Sevilla and Glotin, 2017] Sevilla, A. and Glotin, H. (2017). Audio bird classification with Inception-v4 extended with time and time-frequency attention mechanisms. In *CLEF (Working Notes)*, volume 1866.

[Shonfield and Bayne, 2017] Shonfield, J. and Bayne, E. (2017). Autonomous recording units in avian ecological research: Current use and future applications. *Avian Conservation and Ecology*, 12(1).

[Shu et al., 2018] Shu, H., Song, Y., and Zhou, H. (2018). Time-frequency performance study on urban sound classification with convolutional neural network. In *TENCON 2018-2018 IEEE Region 10 Conference*, pages 1713–1717. IEEE.

[Sibley, 2016] Sibley, D. A. (2016). *Sibley Birds East.* Alfred A. Knopf.

[Simard et al., 2003] Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Slabbekoorn and Peet, 2003] Slabbekoorn, H. and Peet, M. (2003). Ecology: Birds sing at a higher pitch in urban noise. *Nature*, 424(6946):267.

[Sprengel et al., 2016] Sprengel, E., Jaggi, M., Kilcher, Y., and Hofmann, T. (2016). Audio based bird species identification using deep learning techniques. In *CLEF (Working Notes)*, volume 1609, pages 547–559.

[Springenberg et al., 2014] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Stevens et al., 1937] Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190.

[Stowell, 2015] Stowell, D. (2015). BirdCLEF 2015 submission: Unsupervised feature learning from audio. In *CLEF (Working Notes)*, volume 1391.

[Stowell et al., 2015] Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., and Plumbley, M. D. (2015). Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746.

[Stowell and Plumbley, 2013] Stowell, D. and Plumbley, M. D. (2013). An open dataset for research on audio field recording archives: Freefield1010. *arXiv preprint arXiv:1309.5275.*

[Stowell and Plumbley, 2014] Stowell, D. and Plumbley, M. D. (2014). Audio-only bird classification using unsupervised feature learning. In *CLEF (Working Notes)*, volume 1180, pages 673–684.

[Stowell et al., 2016] Stowell, D., Wood, M., Stylianou, Y., and Glotin, H. (2016). Bird detection in audio: A survey and a challenge. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

[Stowell et al., 2018] Stowell, D., Wood, M. D., Pamuła, H., Stylianou, Y., and Glotin, H. (2018). Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge. *Methods in Ecology and Evolution.*

[Sullivan et al., 2009] Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., and Kelling, S. (2009). eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[Szegedy et al., 2013] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199.*

[Tan and Le, 2019] Tan, M. and Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946.*

[Temko et al., 2006a] Temko, A., Malkin, R., Zieger, C., Macho, D., Nadeu, C., and Omologo, M. (2006a). Acoustic event detection and classification in smart-room environments: Evaluation of chil project systems. *Cough*, 65(48):5.

[Temko et al., 2006b] Temko, A., Malkin, R., Zieger, C., Macho, D., Nadeu, C., and Omologo, M. (2006b). Clear evaluation of acoustic event detection and classification systems. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 311–322. Springer.

[Thakur et al., 2019] Thakur, A., Thapar, D., Rajan, P., and Nigam, A. (2019). Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss. *The Journal of the Acoustical Society of America*, 146(1):534–547.

[Theano Development Team, 2016] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[Thomanek et al., 2018] Thomanek, R., Roschke, C., Manthey, R., Platte, B., Rolletschke, T., Heinzig, M., Vodel, M., Kowerko, D., Kahl, S., Zimmer, F., et al. (2018). University of Applied Sciences Mittweida and Chemnitz University of Technology at TRECVID 2018. In *Proceedings of TRECVID Workshop*.

[Thomas et al., 2002] Thomas, R. J., Széskely, T., Cuthill, I. C., Harper, D. G., Newson, S. E., Frayling, T. D., and Wallis, P. D. (2002). Eye size in birds and the timing of song at dawn. *Proceedings of the Royal Society of London B: Biological Sciences*, 269(1493):831–837.

[Thorpe, 1954] Thorpe, W. H. (1954). The process of song-learning in the chaffinch as studied by means of the sound spectrograph. *Nature*, 173(4402):465.

[Tompson et al., 2015] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. (2015). Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656.

[Turing, 1950] Turing, A. M. (1950). I.—Computing Machinery and Intelligence. *Mind*, LIX(236):433–460.

[Tzanetakis and Cook, 2000] Tzanetakis, G. and Cook, P. (2000). Marsyas: A framework for audio analysis. *Organised sound*, 4(3):169–175.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[Veit et al., 2016] Veit, A., Wilber, M. J., and Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pages 550–558.

[Vesperini et al., 2018] Vesperini, F., Gabrielli, L., Principi, E., and Squartini, S. (2018). A capsule neural networks based approach for bird audio detection. Technical report, DCASE2018 Challenge.

[Vickery et al., 1995] Vickery, P. D., Herkert, J. R., Knopf, F. L., Ruth, J., and Keller, C. E. (1995). Grassland birds: An overview of threats and recommended management strategies. In *Strategies for Bird Conservation: The Partners in Flight Planning Process, proceedings of the third Partners in Flight Workshop*, pages 74–77.

[Wang and Brown, 2006] Wang, D. and Brown, G. J. (2006). *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press.

[Wang et al., 2018] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., and Moreno, I. L. (2018). Speaker diarization with LSTM. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243. IEEE.

[Wang et al., 2017] Wang, Y., Getreuer, P., Hughes, T., Lyon, R. F., and Saurous, R. A. (2017). Trainable frontend for robust and far-field keyword spotting. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5670–5674. IEEE.

[Weare, 2006] Weare, C. B. (2006). Audio fingerprinting. US Patent 7,080,253.

[Weeden and Falls, 1959] Weeden, J. S. and Falls, J. B. (1959). Differential responses of male ovenbirds to recorded songs of neighboring and more distant individuals. *The Auk*, pages 343–351.

[Weinberger and Saul, 2009] Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244.

[Wiley and Richards, 1982] Wiley, H. R. and Richards, D. G. (1982). *Acoustic Communication in Birds - Volume 1 - Production, Perception, and Design Features of Sound*, chapter Adaptations for Acoustic Communication in Birds: Sound Transmission and Signal Detection. Academic Press.

[Wilhelm-Stein, 2016] Wilhelm-Stein, T. (2016). *Information Retrieval in der Lehre.* PhD thesis, Chemnitz University of Technology.

[Wilson et al., 2017] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158.

[Winkler et al., 2016] Winkler, D. W., Shamoun-Baranes, J., and Piersma, T. (2016). *Handbook of bird biology*, chapter Avian Migration and Dispersal. John Wiley & Sons.

[Wood and Beresford, 2016] Wood, M. and Beresford, N. (2016). The wildlife of Chernobyl: 30 years without man. *Biologist*, 63(2):16–19.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Xeno-canto, 2019] Xeno-canto (2019). Sharing bird sounds from around the world. https://www.xeno-canto.org/about/xeno-canto. Accessed: 2019-02-10.

[Xie et al., 2018] Xie, J., He, T., Zhang, Z., Zhang, H., Zhang, Z., and Li, M. (2018). Bag of tricks for image classification with convolutional neural networks. *arXiv preprint arXiv:1812.01187*.

[Xie et al., 2017] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500.

[Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

[Yu and Koltun, 2015] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122.*

[Zagoruyko and Komodakis, 2016] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146.*

[Zeiler and Fergus, 2014] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

[Zhang et al., 2016] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530.*

[Zhang et al., 2017] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412.*

[Zhao et al., 2019] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems.*

[Zhou et al., 2016] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.

[Zhuang et al., 2008] Zhuang, X., Zhou, X., Huang, T. S., and Hasegawa-Johnson, M. (2008). Feature analysis and selection for acoustic event detection. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 17–20. IEEE.

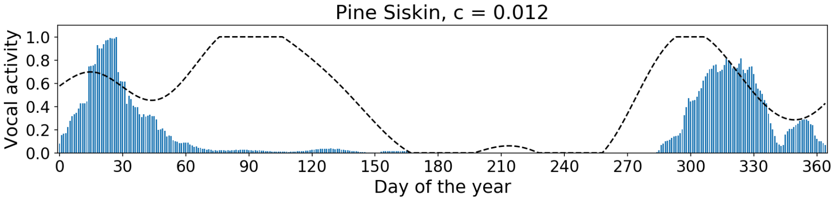
Appendix

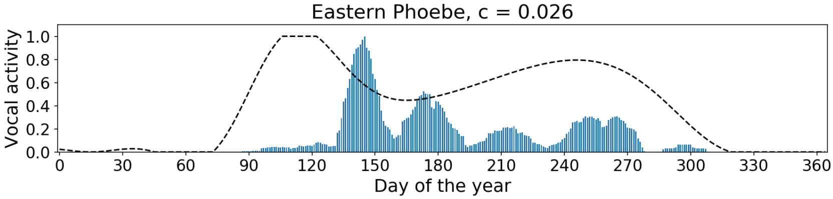
A. Year-round Vocal Activity

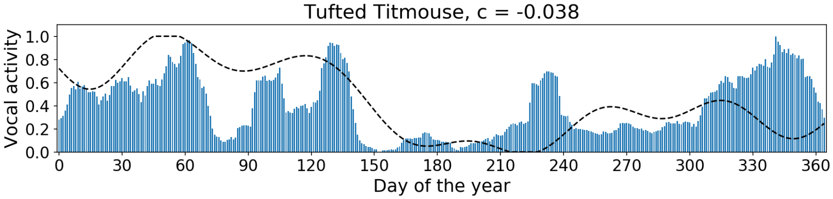
Visualizations of the correlation between relative normalized vocal activity and normalized eBird frequency for one year recorded between 2018-07-20 and 2019-07-19. Detections (blue bars) in live stream data represent the normalized percentage measured against all vocalizations of each of the 365 days. Abundance (dashed line) is represented by the smoothed, weekly frequency based on percentage of eBird checklists that contain this species for Ithaca, NY, USA.

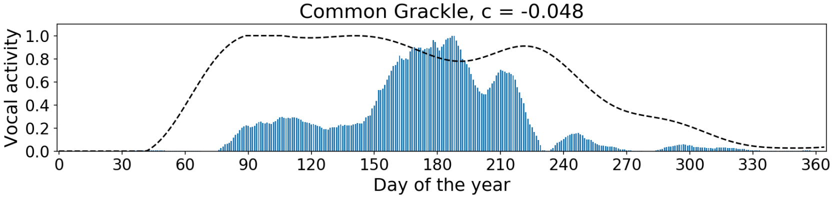
The Pearson correlation coefficient was used to measure the correlation between the two data series.

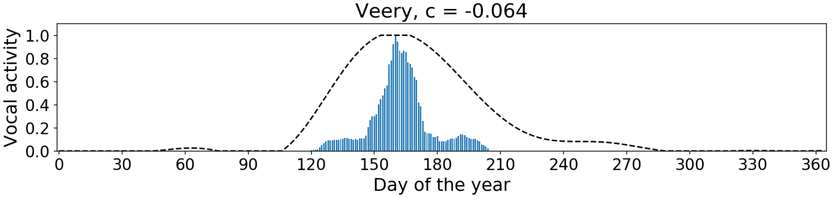
Visualizations sorted by absolute correlation c.

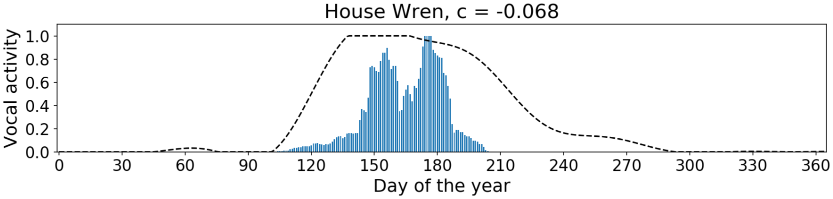


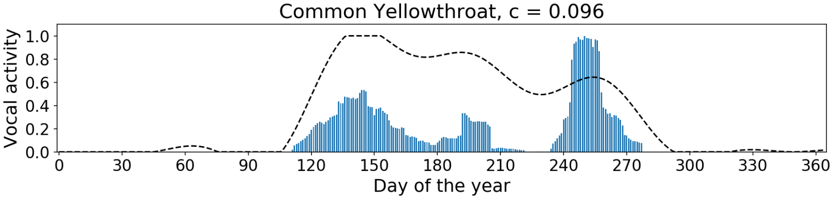


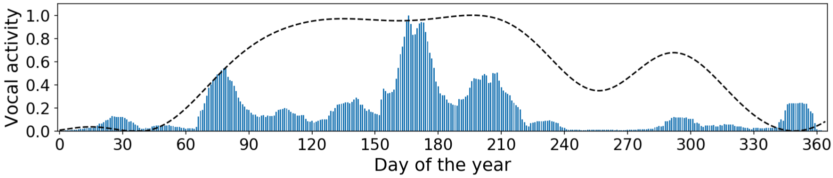

Tufted Titmouse, c = -0.038

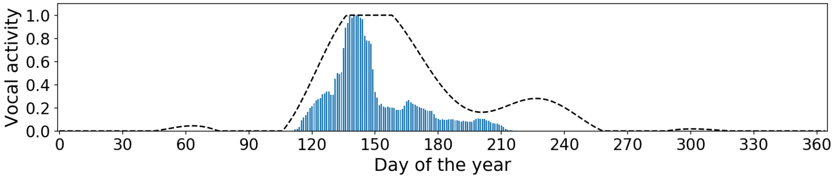

Common Grackle, c = -0.048

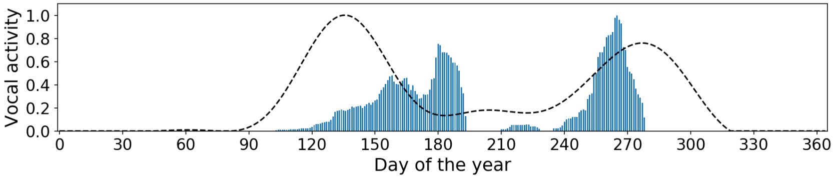

Veery, c = -0.064

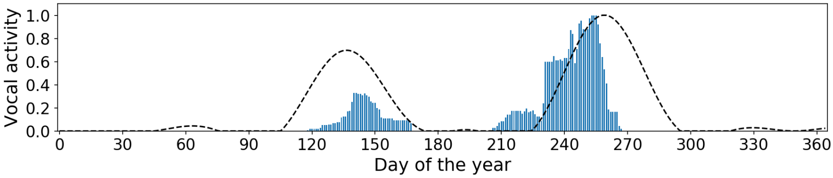

House Wren, c = -0.068

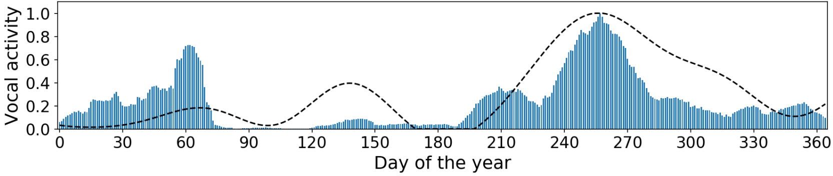

Common Yellowthroat, c = 0.096

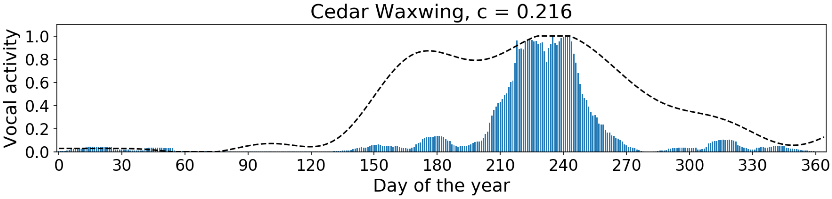

American Robin, c = -0.112

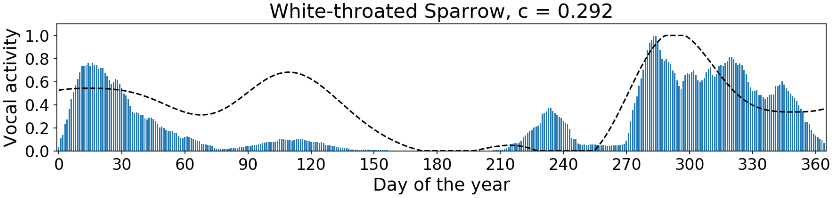

Baltimore Oriole, c = -0.128

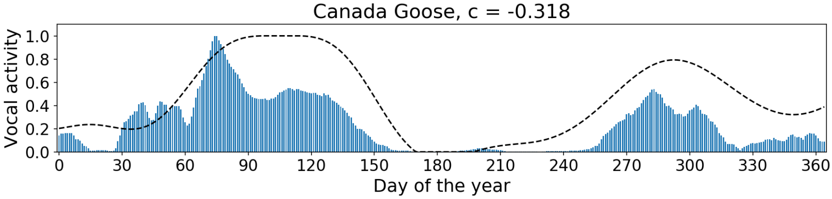

Blue-headed Vireo, c = 0.131

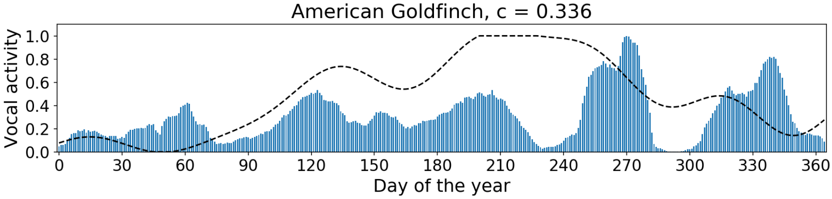

Black-throated Green Warbler, c = 0.169

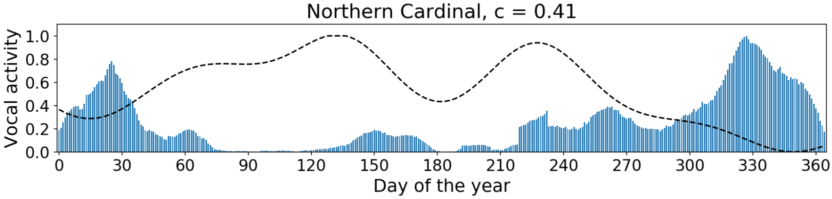

Blue Jay, c = 0.176


Cedar Waxwing, c = 0.216
Vocal activity
1.0
0.8
0.6
0.4
0.2
0.0
0
30
60
90
120
150
180
210
240
270
300
330
360
Day of the year


White-throated Sparrow, c = 0.292
Vocal activity
1.0
0.8
0.6
0.4
0.2
0.0
0
30
60
90
120
150
180
210
240
270
300
330
360
Day of the year

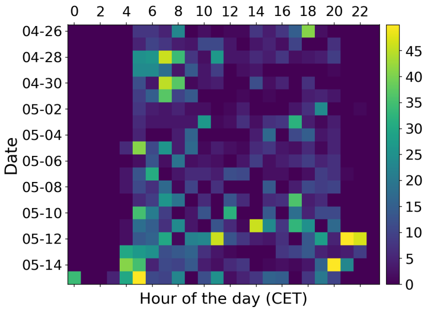

Canada Goose, c = -0.318
Vocal activity
1.0
0.8
0.6
0.4
0.2
0.0
0
30
60
90
120
150
180
210
240
270
300
330
360
Day of the year


American Goldfinch, c = 0.336
Vocal activity
1.0
0.8
0.6
0.4
0.2
0.0
0
30
60
90
120
150
180
210
240
270
300
330
360
Day of the year

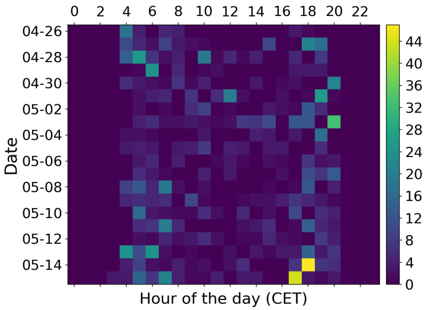

Northern Cardinal, c = 0.41
Vocal activity
1.0
0.8
0.6
0.4
0.2
0.0
0
30
60
90
120
150
180
210
240
270
300
330
360
Day of the year

B. AMTiC Detection Plots

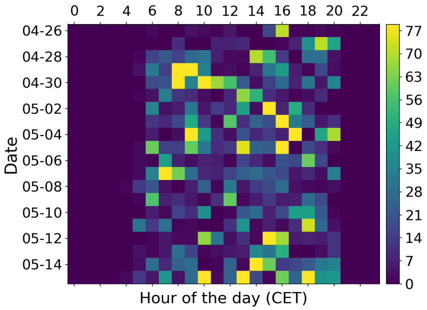
Result visualization of bird species detections from soundscapes recorded between 2019-04-26 and 2019-05-15 at Tierpark Chemnitz. Plots show cumulative detections per hour and location (AMTiC Recorder 1, A1 and AMTiC Recorder 2, A2) and changes in species abundance over time. All times in Central European (Summer) Time (CET, UTC+2).



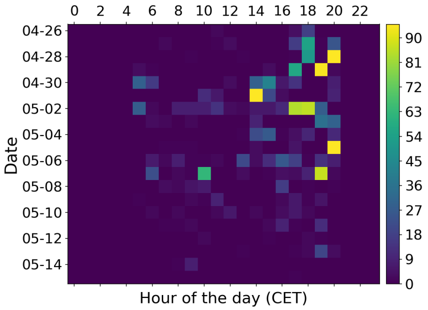
(a) Carrion Crow A1



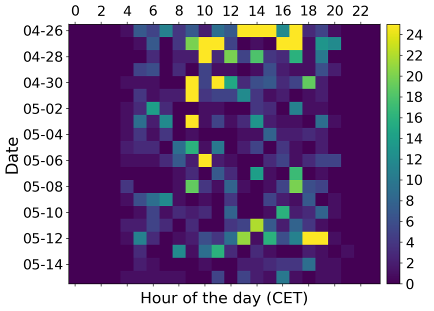
(b) Carrion Crow A2



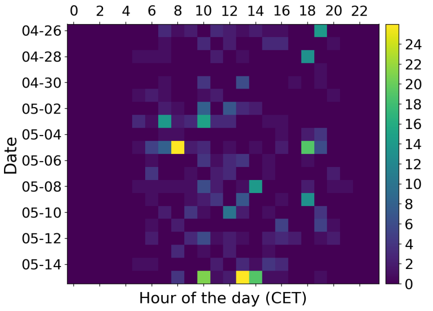
(c) Common Chaffinch A1



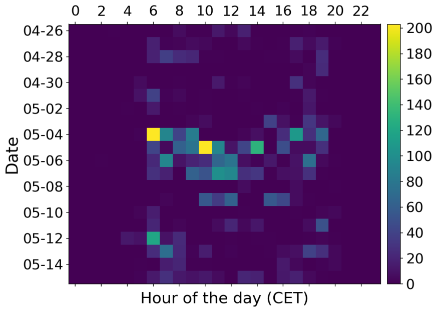
(d) Common Chaffinch A2



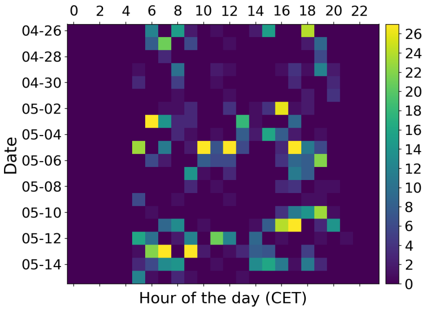
(a) Common Firecrest A1



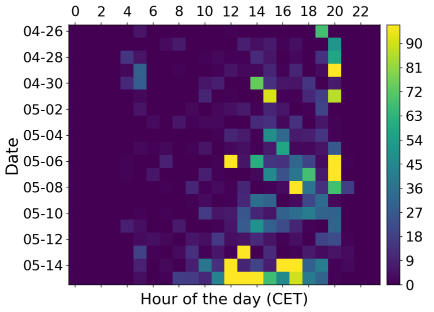
(b) Common Firecrest A2



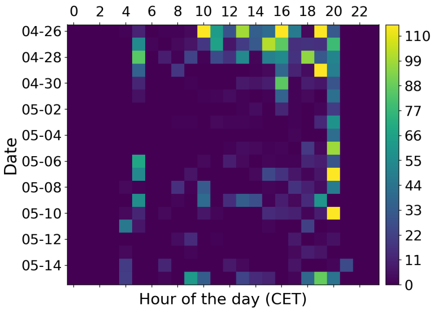
(c) Common Wood-Pigeon A1



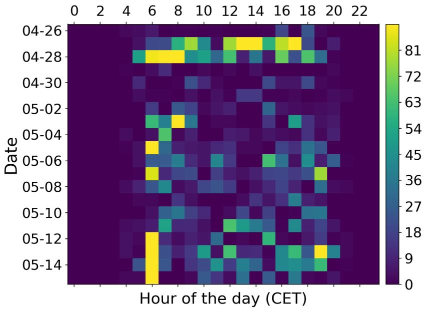
(d) Common Wood-Pigeon A2



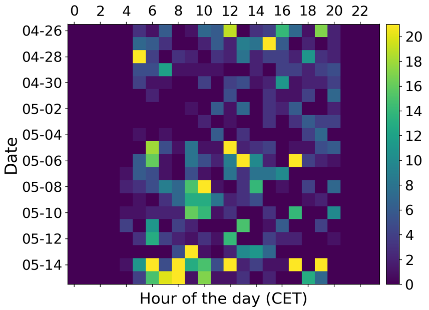
(e) Eurasian Blackbird A1



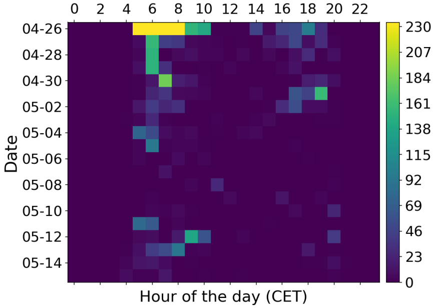
(f) Eurasian Blackbird A2



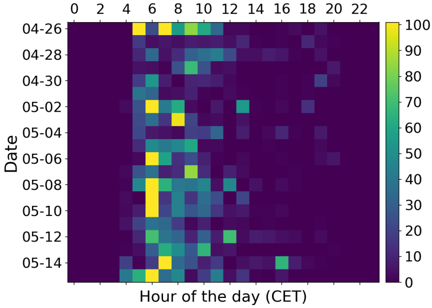
(a) Eurasian Blackcap A1



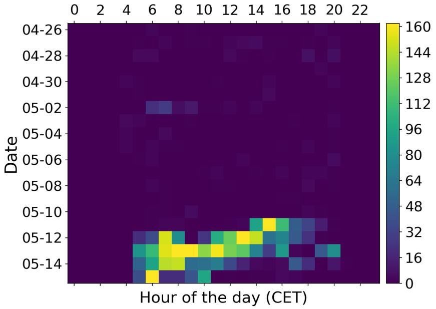
(b) Eurasian Blackcap A2



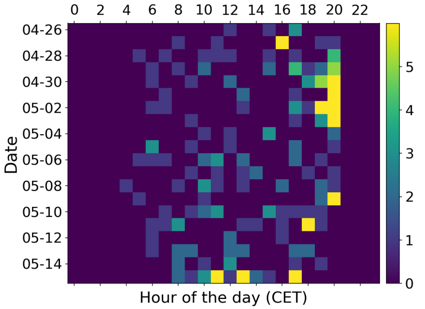
(c) Eurasian Wren A1



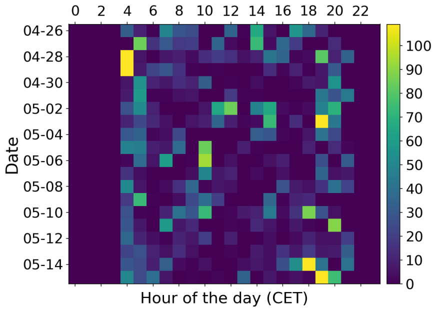
(d) Eurasian Wren A2



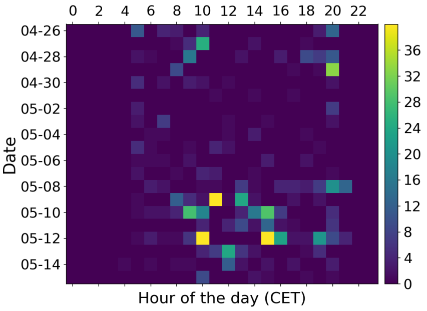
(e) European Pied Flycatcher A1



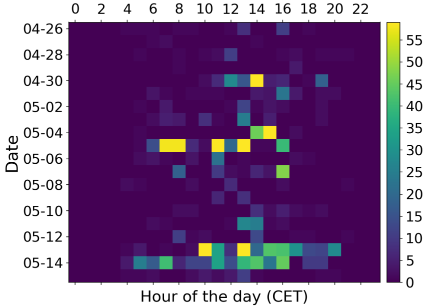
(f) European Pied Flycatcher A2



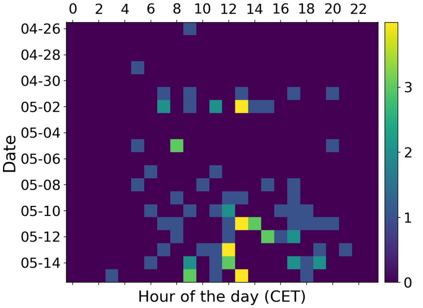
(a) European Robin A1



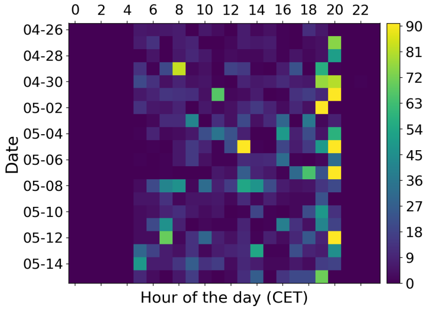
(b) European Robin A2



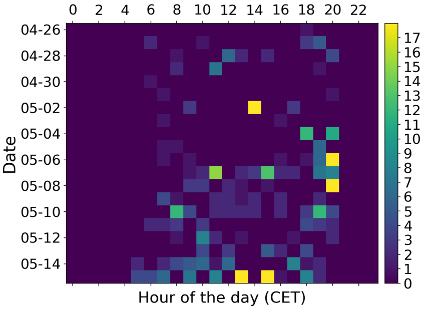
(c) European Serin A1



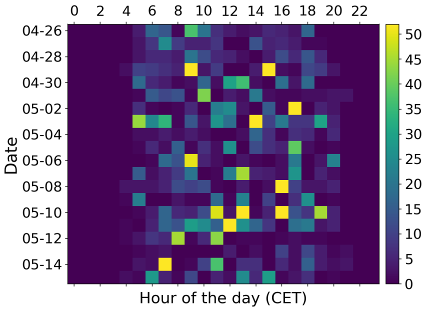
(d) European Serin A2



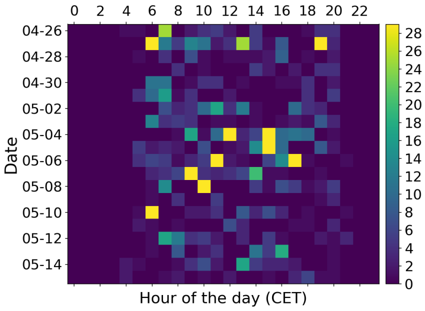
(e) Great Spotted Woodpecker A1



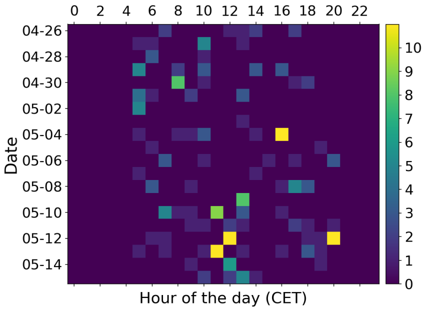
(f) Great Spotted Woodpecker A2



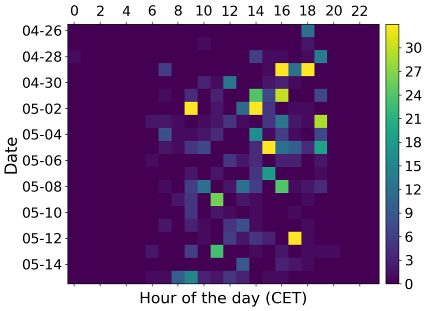
(a) Great Tit A1



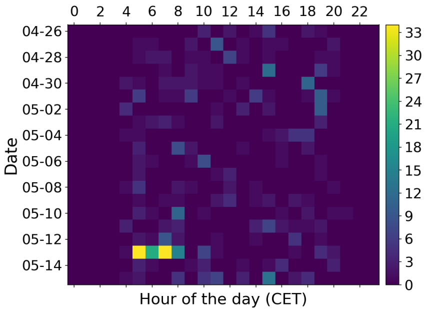
(b) Great Tit A2



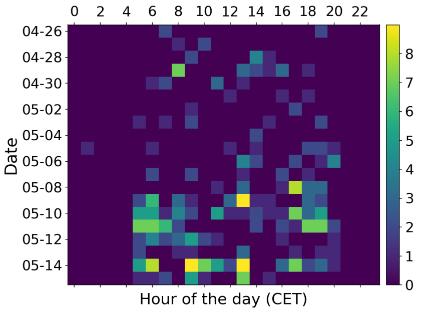
(c) Long-tailed Tit A1



(d) Long-tailed Tit A2



(e) Spotted Flycatcher A1

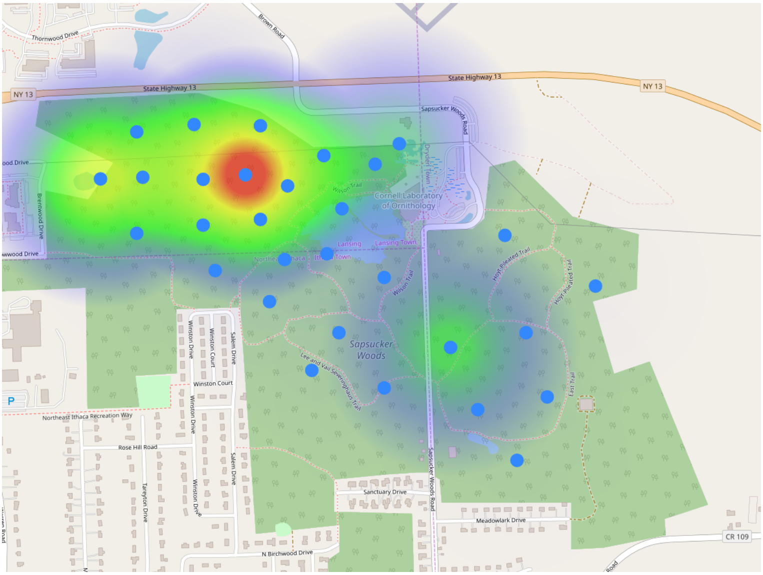


(f) Spotted Flycatcher A2

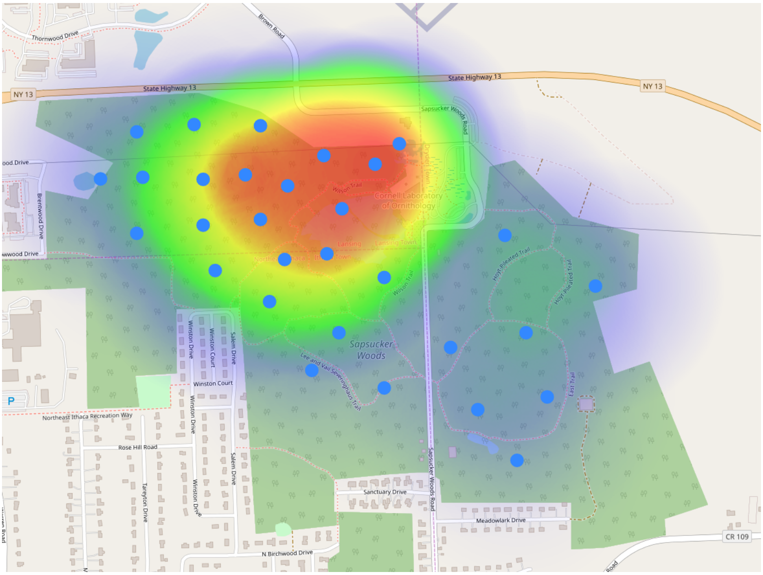
C. SWAMP Spatial Maps

Spatial maps visualizing normalized absolute vocal activity in May 2017 across all recorders of the SWAMP monitoring array in Sapsucker Woods Ithaca, NY, USA. Colors represent high activity (red), medium activity (green) and low activity (blue).

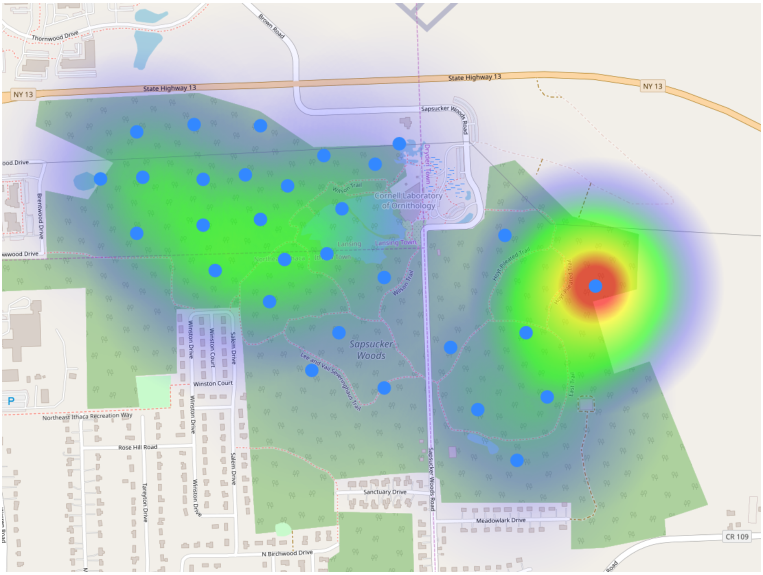
All maps were generated with Leaflet and OpenStreetMap tiles.



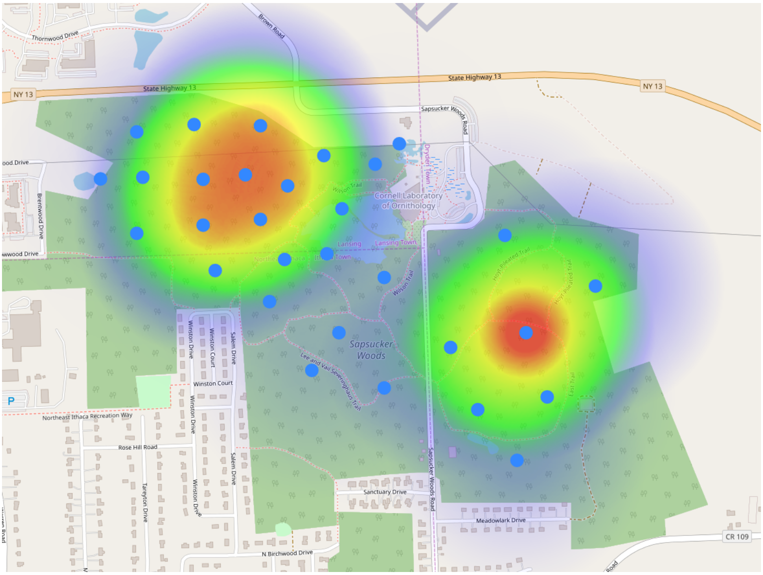
(a) Gray Catbird



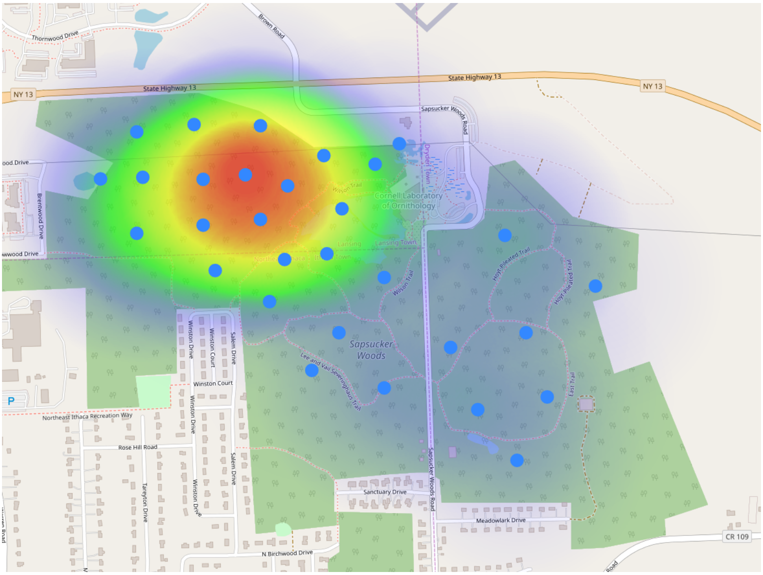
(a) American Goldfinch



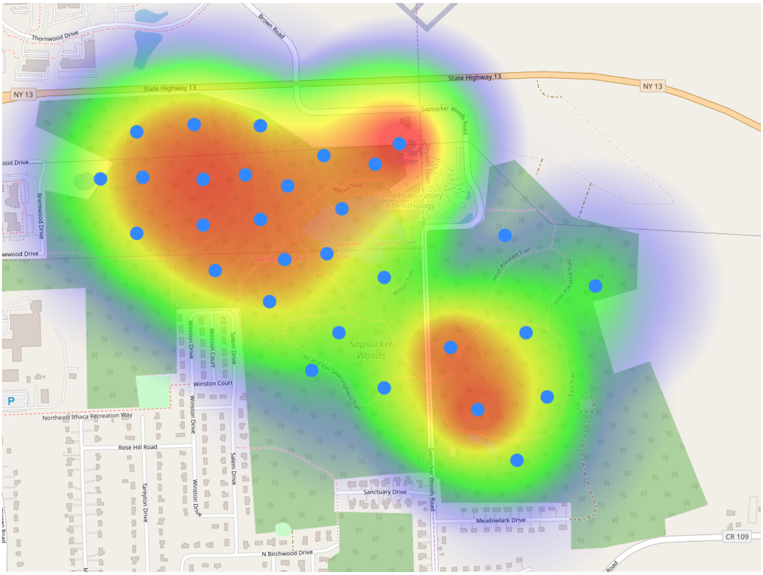
(b) Tufted Titmouse



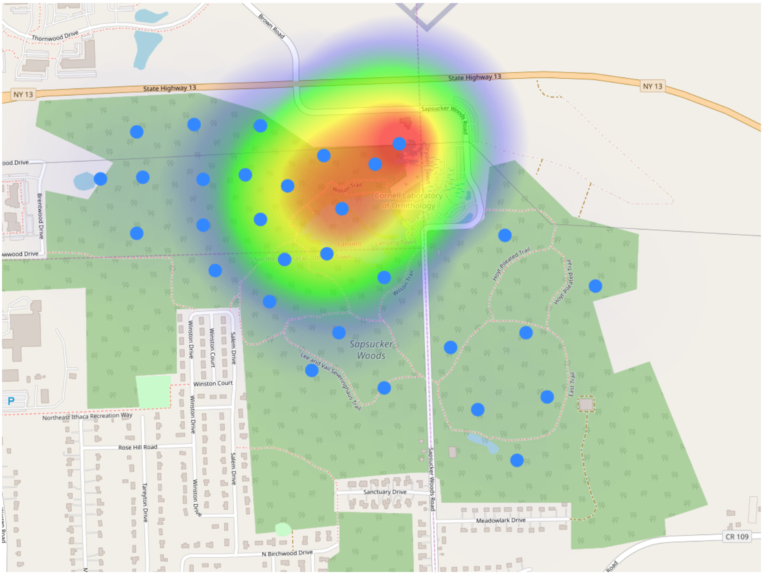
(a) Ovenbird



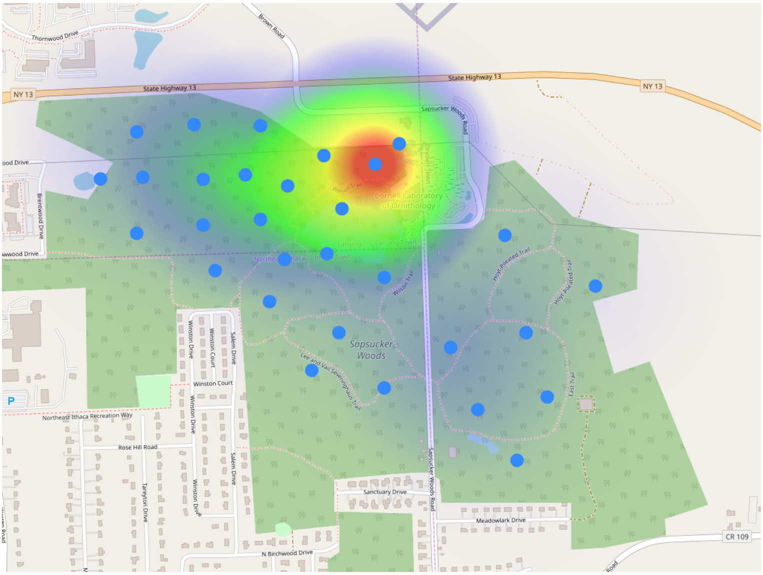
(b) White-throated Sparrow



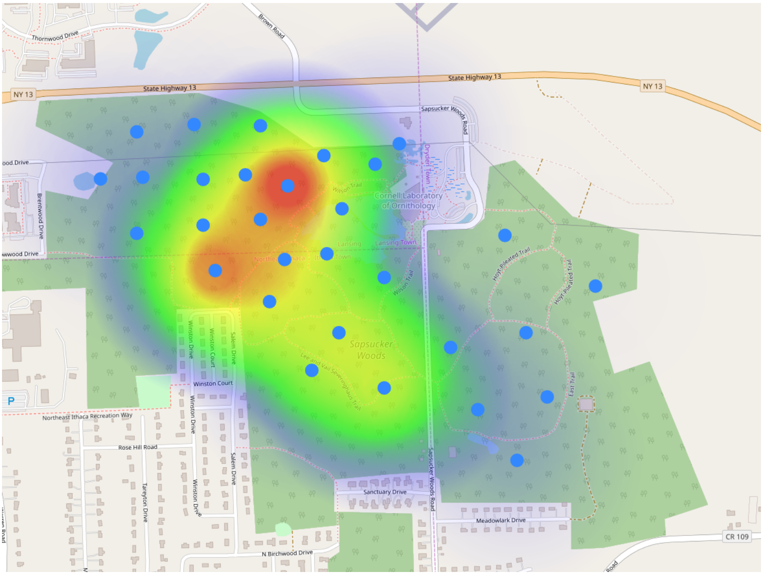
(a) Mourning Dove



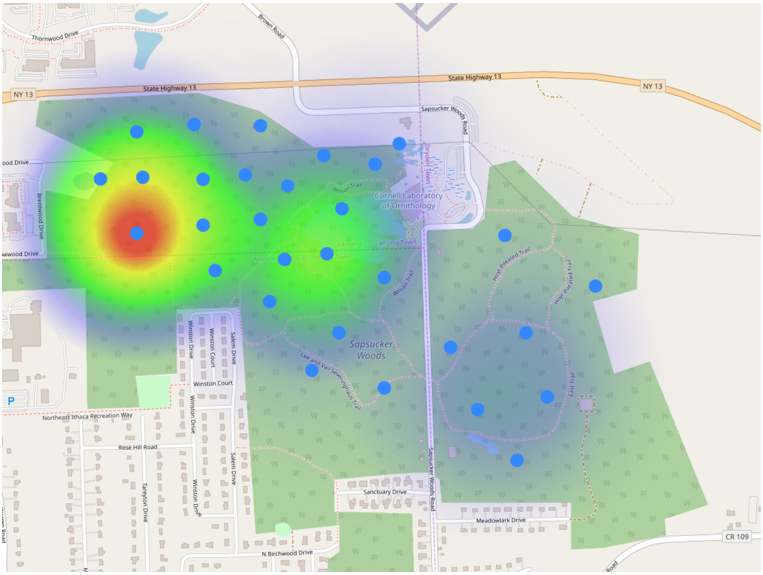
(b) Swamp Sparrow



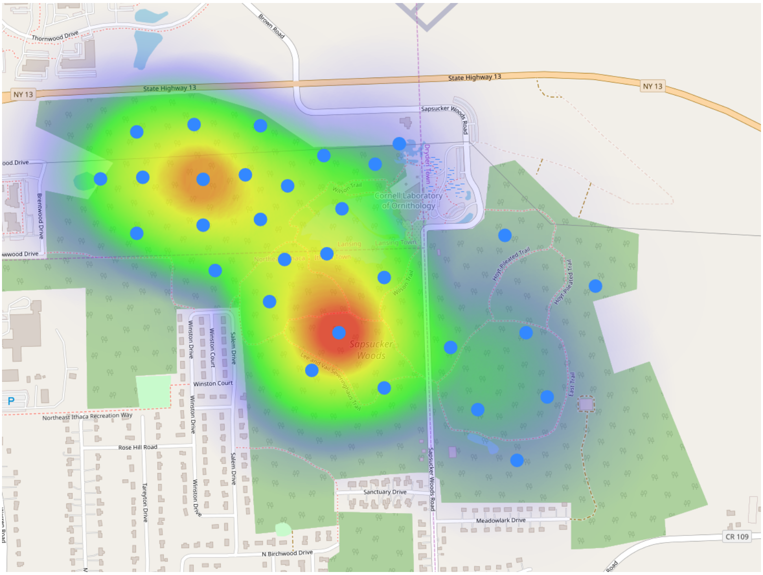
(a) Northern Cardinal



(b) Blue-headed Vireo



(a) Downy Woodpecker



(b) Hairy Woodpecker

D. Species-specific Results

List of supported species and their individual class results based on the best single model. The results reflect the scores achieved on all validation samples independent of their noise level. Despite the lack of a ‘gold standard’, class comparisons in detection quality allow to estimate which species are expected to perform best.

S2N = Signal-to-noise-ratio based on morphological features, higher is better
TS = Total amount of training samples
AP = Average precision across all validation samples
AUC = Area under ROC Curve across all validation samples
F0.5 = Optimized F0.5-measure across all validation samples
CT = Confidence threshold to achieve the optimal F0.5

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Abert’s Towhee	0.506	1515	0.621	0.965	0.619	0.13
Acadian Flycatcher	0.38	2093	0.635	0.989	0.716	0.25
Acorn Woodpecker	0.72	3145	0.805	0.992	0.848	0.15
African Blue Tit	0.62	1613	0.709	0.985	0.602	0.12
African Reed Warbler	0.698	1657	0.714	0.993	0.589	0.21
Alder Flycatcher	0.449	2287	0.56	0.98	0.642	0.1
Aleutian Tern	0.919	690	0.924	0.999	0.903	0.07
Alexandrine Parakeet	0.722	848	0.595	0.952	0.627	0.12
Algerian Nuthatch	0.941	1031	0.739	0.981	0.807	0.15
Allen’s Hummingbird	0.391	523	0.679	0.987	0.564	0.07
Alpine Accentor	0.503	472	0.68	0.984	0.403	0.04
Alpine Swift	0.358	564	0.775	0.966	0.77	0.13
Altamira Oriole	0.587	1616	0.473	0.981	0.403	0.09
American Avocet	0.763	1437	0.564	0.952	0.584	0.1
American Bittern	0.555	1362	0.438	0.92	0.457	0.12
American Coot	0.67	2282	0.657	0.971	0.67	0.24
American Crow	0.573	2641	0.696	0.974	0.69	0.13
American Dipper	0.336	752	0.583	0.956	0.593	0.21
American Golden-Plover	0.614	1884	0.654	0.989	0.701	0.21

SPECIES	S2N	TS	AP	AUC	F0.5	CT
American Goldfinch	0.533	2720	0.708	0.97	0.724	0.19
American Kestrel	0.603	1519	0.692	0.959	0.732	0.16
American Oystercatcher	0.518	1086	0.679	0.972	0.605	0.17
American Pipit	0.553	1370	0.751	0.989	0.758	0.09
American Redstart	0.539	2774	0.6	0.986	0.584	0.15
American Robin	0.585	3661	0.715	0.979	0.681	0.13
American Three-toed Woodpecker	0.645	1943	0.481	0.986	0.399	0.14
American Tree Sparrow	0.549	2498	0.633	0.981	0.678	0.21
American Wigeon	0.561	1429	0.516	0.962	0.464	0.09
American Woodcock	0.356	2501	0.794	0.966	0.806	0.13
Anhinga	0.709	1224	0.478	0.935	0.436	0.1
Anna's Hummingbird	0.535	1843	0.655	0.967	0.644	0.21
Antillean Nighthawk	0.621	1688	0.692	0.968	0.69	0.1
Aplomado Falcon	0.839	897	0.436	0.834	0.468	0.07
Aquatic Warbler	0.696	1945	0.794	0.991	0.837	0.13
Arctic Loon	0.657	753	0.564	0.933	0.545	0.27
Arctic Tern	0.705	2576	0.718	0.989	0.763	0.18
Arctic Warbler	0.517	2674	0.83	0.987	0.856	0.09
Arizona Woodpecker	0.62	1023	0.476	0.938	0.601	0.11
Ash-throated Flycatcher	0.512	3273	0.738	0.983	0.779	0.16
Asian Desert Warbler	0.598	136	0.198	0.711	0.235	0.1
Atlantic Puffin	0.445	1131	0.759	0.953	0.698	0.07
Atlas Flycatcher	0.745	223	0.594	0.983	0.469	0.1
Audouin's Gull	0.735	249	0.672	0.988	0.315	0.03
Audubon's Oriole	0.71	1284	0.55	0.943	0.612	0.1
Azure Tit	0.726	738	0.638	0.95	0.57	0.08
Bachman's Sparrow	0.538	1873	0.691	0.988	0.682	0.13
Baird's Sandpiper	0.685	1062	0.59	0.982	0.629	0.21
Baird's Sparrow	0.67	2382	0.818	0.997	0.691	0.21
Bald Eagle	0.492	790	0.526	0.921	0.558	0.14
Baltimore Oriole	0.508	3406	0.58	0.986	0.546	0.17
Band-rumped Storm-Petrel	0.981	82	0.396	0.971	0.324	0.03
Band-tailed Pigeon	0.515	1965	0.457	0.931	0.516	0.1
Bank Swallow	0.575	1872	0.834	0.993	0.854	0.13
Bar-headed Goose	0.932	249	0.713	0.932	0.63	0.1
Bar-tailed Godwit	0.831	1070	0.469	0.947	0.525	0.13
Bar-tailed Lark	0.479	119	0.198	0.848	0.12	0.04
Barbary Partridge	0.435	97	0.157	0.706	0.097	0.04
Barn Owl	0.674	1816	0.705	0.981	0.686	0.11
Barn Swallow	0.772	2468	0.827	0.983	0.835	0.15
Barnacle Goose	0.551	1587	0.883	0.998	0.818	0.11

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Barred Owl	0.496	1965	0.607	0.971	0.609	0.11
Barred Warbler	0.684	2439	0.592	0.993	0.51	0.11
Bay-breasted Warbler	0.501	880	0.397	0.969	0.4	0.15
Bearded Reedling	0.431	1503	0.81	0.989	0.849	0.12
Bell's Sparrow	0.541	1852	0.754	0.984	0.727	0.11
Bell's Vireo	0.673	2734	0.701	0.982	0.685	0.12
Belted Kingfisher	0.591	1571	0.732	0.968	0.776	0.14
Bendire's Thrasher	0.812	1619	0.642	0.969	0.771	0.18
Bewick's Wren	0.708	2945	0.752	0.987	0.724	0.11
Bicknell's Thrush	0.55	2695	0.686	0.986	0.725	0.13
Black Francolin	0.652	1238	0.399	0.898	0.493	0.21
Black Grouse	0.701	2204	0.932	0.998	0.909	0.17
Black Guillemot	0.434	571	0.559	0.935	0.513	0.1
Black Kite	0.558	1398	0.522	0.958	0.595	0.19
Black Lark	0.722	94	0.016	0.831	0.003	0.01
Black Oystercatcher	0.644	3115	0.884	0.996	0.872	0.15
Black Phoebe	0.444	2451	0.78	0.984	0.8	0.1
Black Rail	0.514	2187	0.681	0.985	0.713	0.15
Black Redstart	0.654	3177	0.534	0.94	0.559	0.26
Black Rosy-Finch	0.381	451	0.559	0.975	0.42	0.06
Black Skimmer	0.732	1681	0.861	0.996	0.858	0.16
Black Stork	0.8	626	0.589	0.984	0.575	0.12
Black Swan	0.745	216	0.258	0.881	0.2	0.04
Black Tern	0.873	2506	0.813	0.998	0.832	0.15
Black Turnstone	0.733	815	0.947	0.999	0.842	0.01
Black Vulture	0.458	1183	0.369	0.903	0.311	0.05
Black Wheatear	0.597	915	1.0	1.0	0.692	0.02
Black Woodpecker	0.631	2258	0.631	0.988	0.664	0.2
Black-and-white Warbler	0.528	2301	0.518	0.983	0.429	0.09
Black-backed Woodpecker	0.568	2327	0.537	0.99	0.485	0.12
Black-bellied Plover	0.435	1300	0.656	0.976	0.664	0.18
Black-bellied Sandgrouse	0.723	228	1.0	1.0	1.0	0.01
Black-bellied Whistling-Duck	0.553	1333	0.792	0.99	0.771	0.12
Black-billed Cuckoo	0.579	1347	0.621	0.96	0.67	0.13
Black-billed Magpie	0.708	2066	0.597	0.94	0.623	0.16
Black-capped Chickadee	0.531	3129	0.792	0.988	0.751	0.13
Black-capped Vireo	0.869	2270	0.673	0.969	0.717	0.11
Black-chinned Hummingbird	0.508	1097	0.679	0.98	0.638	0.17
Black-chinned Sparrow	0.582	1668	0.639	0.976	0.703	0.12
Black-crested Titmouse	0.608	2108	0.684	0.976	0.656	0.19
Black-crowned Night-Heron	0.461	1801	0.52	0.958	0.516	0.19

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Black-crowned Tchagra	0.612	1690	0.671	0.975	0.69	0.1
Black-eared Wheatear	0.487	959	0.439	0.962	0.385	0.12
Black-footed Albatross	0.672	1726	0.931	0.999	0.92	0.13
Black-headed Bunting	0.603	1098	0.707	0.976	0.676	0.1
Black-headed Grosbeak	0.593	2975	0.711	0.988	0.705	0.19
Black-headed Gull	0.679	2458	0.793	0.996	0.731	0.19
Black-legged Kittiwake	0.383	2695	0.782	0.995	0.727	0.27
Black-necked Stilt	0.841	2774	0.935	0.999	0.871	0.17
Black-shouldered Kite	0.56	1613	0.854	0.991	0.846	0.1
Black-tailed Gnatcatcher	0.776	2433	0.838	0.998	0.785	0.15
Black-tailed Godwit	0.867	1884	0.855	0.995	0.794	0.15
Black-throated Blue Warbler	0.546	1811	0.364	0.976	0.333	0.11
Black-throated Gray Warbler	0.536	2471	0.457	0.975	0.495	0.27
Black-throated Green Warbler	0.565	2684	0.574	0.989	0.621	0.25
Black-throated Sparrow	0.627	3025	0.722	0.988	0.768	0.27
Black-whiskered Vireo	0.592	2754	0.817	0.991	0.839	0.14
Black-winged Pratincole	0.636	80	0.032	0.802	0.028	0.13
Black-winged Stilt	0.695	2688	0.948	0.999	0.915	0.15
Blackburnian Warbler	0.515	1857	0.525	0.99	0.446	0.09
Blackpoll Warbler	0.497	2787	0.506	0.974	0.624	0.22
Blue Grosbeak	0.528	2694	0.595	0.982	0.663	0.21
Blue Jay	0.645	2247	0.631	0.963	0.602	0.16
Blue Rock-Thrush	0.53	1685	0.503	0.969	0.413	0.09
Blue-cheeked Bee-eater	0.488	553	0.631	0.939	0.702	0.11
Blue-gray Gnatcatcher	0.672	2674	0.681	0.975	0.708	0.16
Blue-headed Vireo	0.518	2867	0.828	0.995	0.8	0.22
Blue-throated Hummingbird	0.468	1972	0.64	0.971	0.688	0.12
Blue-winged Teal	0.687	1505	0.727	0.977	0.795	0.14
Blue-winged Warbler	0.554	2210	0.43	0.982	0.441	0.27
Bluethroat	0.637	3448	0.507	0.97	0.371	0.07
Blyth’s Reed Warbler	0.691	3862	0.744	0.994	0.656	0.14
Boat-tailed Grackle	0.649	1514	0.635	0.979	0.578	0.13
Bobolink	0.783	3103	0.822	0.996	0.789	0.14
Bohemian Waxwing	0.461	2262	0.878	0.986	0.883	0.1
Bonaparte’s Gull	0.702	1476	0.666	0.958	0.661	0.14
Booted Eagle	0.379	378	0.387	0.764	0.27	0.03
Booted Warbler	0.72	913	0.522	0.962	0.425	0.12
Boreal Chickadee	0.698	2212	0.744	0.987	0.753	0.14
Boreal Owl	0.501	2682	0.891	0.995	0.912	0.19
Botteri’s Sparrow	0.563	2754	0.718	0.988	0.74	0.13
Brambling	0.468	1890	0.852	0.995	0.865	0.11

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Brant	0.581	1777	0.693	0.994	0.656	0.12
Brewer's Blackbird	0.627	2239	0.789	0.991	0.729	0.11
Brewer's Sparrow	0.762	3471	0.649	0.986	0.657	0.32
Bridled Titmouse	0.744	2008	0.616	0.973	0.672	0.11
Broad-billed Hummingbird	0.545	574	0.576	0.895	0.525	0.17
Broad-billed Sandpiper	0.471	175	0.013	0.778	0.0	0.0
Broad-tailed Hummingbird	0.506	1527	0.691	0.964	0.711	0.1
Broad-winged Hawk	0.422	2026	0.431	0.958	0.434	0.18
Brown Creeper	0.419	2484	0.649	0.983	0.54	0.05
Brown Fish-Owl	0.373	236	0.008	0.878	0.0	0.0
Brown Jay	1.006	1975	0.833	0.994	0.851	0.12
Brown Pelican	0.87	680	0.469	0.954	0.402	0.19
Brown Thrasher	0.726	3508	0.763	0.993	0.705	0.19
Brown-capped Rosy-Finch	0.607	1267	0.699	0.996	0.532	0.11
Brown-crested Flycatcher	0.578	4152	0.723	0.988	0.747	0.17
Brown-headed Cowbird	0.535	2766	0.447	0.959	0.527	0.19
Brown-headed Nuthatch	0.579	1989	0.744	0.974	0.721	0.09
Brown-necked Raven	0.753	228	0.555	0.945	0.531	0.15
Buff-bellied Hummingbird	0.465	511	0.9	0.997	0.964	0.04
Buff-breasted Flycatcher	0.508	1877	0.73	0.989	0.76	0.18
Bufflehead	0.565	1012	0.15	0.928	0.109	0.05
Bullock's Oriole	0.663	3390	0.572	0.971	0.634	0.21
Burrowing Owl	0.607	1591	0.587	0.955	0.681	0.19
Bushtit	0.477	1913	0.863	0.988	0.826	0.16
Cackling Goose	0.709	1484	0.695	0.992	0.577	0.14
Cactus Wren	0.684	2874	0.64	0.982	0.666	0.13
Calandra Lark	0.86	2479	0.855	0.999	0.818	0.14
California Gnatcatcher	0.836	1428	0.815	0.991	0.735	0.1
California Gull	0.742	1402	0.535	0.919	0.459	0.11
California Quail	0.591	2545	0.646	0.979	0.681	0.18
California Scrub-Jay	0.69	1757	0.751	0.968	0.738	0.13
California Thrasher	0.718	3433	0.677	0.991	0.628	0.13
California Towhee	0.499	2078	0.754	0.982	0.768	0.07
Calliope Hummingbird	0.796	990	0.443	0.949	0.364	0.07
Canada Goose	0.697	3064	0.748	0.983	0.715	0.16
Canada Jay	0.631	2398	0.473	0.981	0.436	0.16
Canada Warbler	0.594	1742	0.575	0.985	0.643	0.16
Canyon Towhee	0.627	3134	0.555	0.981	0.619	0.16
Canyon Wren	0.559	2505	0.707	0.977	0.754	0.13
Cape May Warbler	0.477	2148	0.568	0.982	0.562	0.2
Carolina Chickadee	0.539	2558	0.747	0.991	0.772	0.21

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Carolina Wren	0.669	2821	0.586	0.98	0.563	0.12
Carrion Crow	0.688	1756	0.533	0.968	0.461	0.15
Caspian Gull	0.81	726	0.527	0.991	0.407	0.11
Caspian Tern	0.63	1991	0.72	0.983	0.716	0.15
Cassia Crossbill	0.479	536	0.764	0.985	0.735	0.09
Cassin's Finch	0.633	3135	0.629	0.99	0.603	0.14
Cassin's Kingbird	0.67	2541	0.696	0.984	0.743	0.13
Cassin's Sparrow	0.541	2447	0.667	0.991	0.667	0.16
Cassin's Vireo	0.567	2620	0.79	0.994	0.693	0.19
Cattle Egret	0.479	1773	0.749	0.993	0.669	0.12
Cave Swallow	0.621	1218	0.609	0.897	0.597	0.06
Cedar Waxwing	0.461	2217	0.872	0.992	0.903	0.11
Cerulean Warbler	0.67	2852	0.707	0.994	0.644	0.12
Cetti's Warbler	0.636	1628	0.646	0.98	0.647	0.12
Chestnut-backed Chickadee	0.507	2867	0.803	0.985	0.807	0.13
Chestnut-bellied Sandgrouse	0.951	283	0.505	0.864	0.6	0.08
Chestnut-collared Longspur	0.651	1473	0.517	0.978	0.52	0.25
Chestnut-sided Warbler	0.609	1915	0.42	0.982	0.371	0.08
Chihuahuan Raven	0.577	927	0.485	0.883	0.425	0.08
Chimney Swift	0.424	1110	0.669	0.919	0.731	0.09
Chipping Sparrow	0.567	2695	0.666	0.975	0.638	0.15
Chuck-will's-widow	0.55	2341	0.949	0.999	0.944	0.07
Chukar	0.669	770	0.433	0.925	0.436	0.12
Cinereous Bunting	0.589	363	0.462	0.862	0.383	0.08
Cirl Bunting	0.581	2578	0.601	0.983	0.686	0.24
Citril Finch	0.569	531	0.712	0.975	0.625	0.11
Citrine Wagtail	0.394	806	0.772	0.964	0.69	0.14
Clapper Rail	0.588	2113	0.565	0.983	0.484	0.12
Clark's Nutcracker	0.798	1999	0.748	0.996	0.753	0.13
Clay-colored Sparrow	0.629	2834	0.503	0.983	0.484	0.2
Clay-colored Thrush	0.567	3594	0.646	0.973	0.672	0.15
Cliff Swallow	0.686	1669	0.559	0.888	0.643	0.3
Coal Tit	0.661	3195	0.751	0.995	0.726	0.24
Collared Flycatcher	0.545	2766	0.816	0.996	0.804	0.18
Collared Pratincole	0.601	637	0.692	0.953	0.678	0.1
Common Black Hawk	0.547	760	0.274	0.836	0.368	0.11
Common Bulbul	0.653	3062	0.673	0.986	0.629	0.15
Common Buzzard	0.588	2267	0.717	0.984	0.71	0.21
Common Chaffinch	0.651	3360	0.579	0.982	0.603	0.11
Common Chiffchaff	0.477	2925	0.852	0.993	0.847	0.19
Common Crane	0.776	2577	0.855	0.993	0.794	0.1

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Common Cuckoo	0.618	2971	0.768	0.987	0.836	0.17
Common Eider	0.538	1472	0.838	0.995	0.845	0.16
Common Firecrest	0.464	2656	0.639	0.991	0.598	0.12
Common Gallinule	0.621	1901	0.701	0.985	0.629	0.2
Common Goldeneye	0.529	1101	0.417	0.907	0.498	0.11
Common Grackle	0.547	2923	0.723	0.989	0.703	0.12
Common Grasshopper-Warbler	0.52	3134	0.889	0.996	0.904	0.12
Common Greenshank	0.512	1593	0.868	0.997	0.832	0.22
Common Ground-Dove	0.54	1950	0.792	0.989	0.817	0.16
Common House-Martin	0.598	2539	0.716	0.969	0.679	0.16
Common Kingfisher	0.389	1196	0.824	0.979	0.842	0.09
Common Loon	0.671	2720	0.568	0.962	0.648	0.2
Common Merganser	0.519	1050	0.436	0.864	0.489	0.1
Common Murre	0.375	2096	0.645	0.986	0.482	0.08
Common Myna	0.622	2535	0.433	0.934	0.479	0.22
Common Nighthawk	0.447	2244	0.816	0.984	0.829	0.09
Common Nightingale	0.785	4241	0.744	0.993	0.736	0.18
Common Pauraque	0.431	3411	0.96	0.999	0.933	0.05
Common Pochard	0.554	553	0.875	0.994	0.711	0.09
Common Poorwill	0.332	2375	0.914	0.998	0.881	0.09
Common Quail	0.524	2240	0.651	0.984	0.646	0.08
Common Raven	0.621	2617	0.658	0.96	0.674	0.12
Common Redpoll	0.559	2665	0.849	0.993	0.676	0.14
Common Redshank	0.637	1868	0.794	0.986	0.749	0.26
Common Redstart	0.688	3596	0.666	0.99	0.675	0.22
Common Ringed Plover	0.467	1359	0.674	0.963	0.684	0.13
Common Rosefinch	0.534	3050	0.755	0.987	0.787	0.13
Common Sandpiper	0.496	1873	0.76	0.974	0.824	0.22
Common Scoter	0.266	373	0.661	0.973	0.629	0.12
Common Shelduck	0.7	1387	0.828	0.992	0.789	0.16
Common Snipe	0.605	1959	0.65	0.969	0.604	0.09
Common Swift	0.518	2033	0.861	0.991	0.887	0.17
Common Tern	0.721	2473	0.836	0.991	0.819	0.19
Common Waxbill	0.488	1555	0.839	0.985	0.862	0.1
Common Wood-Pigeon	0.515	2123	0.665	0.984	0.656	0.21
Common Yellowthroat	0.534	2908	0.449	0.958	0.483	0.16
Connecticut Warbler	0.614	2247	0.395	0.967	0.44	0.16
Cooper's Hawk	0.525	1035	0.653	0.972	0.684	0.1
Cordilleran Flycatcher	0.423	1736	0.699	0.981	0.67	0.1
Corn Bunting	0.689	3053	0.694	0.988	0.635	0.13
Corn Crake	1.054	2780	0.809	0.979	0.832	0.18

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Corsican Finch	0.448	195	0.571	0.937	0.456	0.05
Cory’s Shearwater	0.58	324	0.874	0.981	0.86	0.14
Costa’s Hummingbird	0.624	569	0.659	0.913	0.692	0.11
Couch’s Kingbird	0.541	2065	0.788	0.989	0.86	0.16
Crested Caracara	0.638	855	0.398	0.896	0.547	0.15
Crested Lark	0.48	2200	0.608	0.985	0.574	0.23
Crested Tit	0.648	2376	0.843	0.998	0.847	0.18
Cretzschmar’s Bunting	0.412	331	0.563	0.93	0.533	0.12
Crissal Thrasher	0.682	2561	0.742	0.971	0.624	0.18
Crowned Sandgrouse	0.533	72	0.884	0.993	0.843	0.21
Curlew Sandpiper	0.434	113	0.522	0.893	0.365	0.05
Curve-billed Thrasher	0.67	3651	0.708	0.994	0.701	0.31
Dark-eyed Junco	0.571	3115	0.739	0.992	0.676	0.15
Dartford Warbler	0.719	2244	0.958	0.999	0.908	0.09
Dead Sea Sparrow	0.79	274	0.548	0.947	0.587	0.09
Demoiselle Crane	0.588	191	0.981	0.999	0.743	0.03
Desert Finch	0.422	123	0.028	0.895	0.0	0.0
Desert Lark	0.486	125	0.434	0.975	0.272	0.03
Desert Sparrow	0.679	324	0.009	0.795	0.0	0.0
Desert Wheatear	0.514	475	0.32	0.846	0.389	0.12
Dickcissel	0.603	2360	0.788	0.996	0.783	0.14
Double-crested Cormorant	0.562	1303	0.509	0.969	0.433	0.13
Downy Woodpecker	0.557	2278	0.702	0.968	0.675	0.19
Dunlin	0.53	1493	0.547	0.984	0.488	0.1
Dunnock	0.625	2814	0.717	0.988	0.742	0.2
Dupont’s Lark	0.574	1040	0.88	0.999	0.84	0.18
Dusky Flycatcher	0.416	1671	0.712	0.977	0.679	0.07
Dusky Grouse	0.455	804	0.505	0.972	0.476	0.08
Dusky Warbler	0.424	2199	0.627	0.968	0.637	0.14
Dusky-capped Flycatcher	0.471	2766	0.602	0.982	0.672	0.21
Eared Grebe	0.487	683	0.684	0.959	0.657	0.06
Eastern Bluebird	0.546	2713	0.717	0.987	0.776	0.15
Eastern Bonelli’s Warbler	0.617	367	0.801	0.986	0.829	0.16
Eastern Kingbird	0.564	2629	0.701	0.968	0.704	0.08
Eastern Meadowlark	0.467	2875	0.642	0.984	0.699	0.16
Eastern Olivaceous Warbler	0.735	1883	0.649	0.975	0.628	0.17
Eastern Orphean Warbler	0.648	1005	0.51	0.98	0.398	0.12
Eastern Phoebe	0.517	2716	0.771	0.984	0.818	0.18
Eastern Rock Nuthatch	0.518	304	0.337	0.957	0.184	0.03
Eastern Screech-Owl	0.326	1921	0.834	0.995	0.846	0.27
Eastern Towhee	0.527	2884	0.577	0.979	0.578	0.17

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Eastern Whip-poor-will	0.676	2745	0.832	0.993	0.863	0.1
Eastern Wood-Pewee	0.374	2518	0.654	0.986	0.733	0.16
Eastern Yellow Wagtail	0.507	762	0.47	0.888	0.536	0.2
Egyptian Goose	0.71	1300	0.67	0.967	0.705	0.17
Elegant Tern	0.57	1029	0.607	0.955	0.693	0.16
Elegant Trogon	0.654	2979	0.56	0.976	0.622	0.1
Eleonora's Falcon	0.317	36	0.008	0.779	0.0	0.0
Elf Owl	0.576	2502	0.789	0.991	0.807	0.2
Emperor Goose	0.777	317	0.381	0.971	0.291	0.09
Eurasian Blackbird	0.741	3703	0.658	0.992	0.626	0.21
Eurasian Blackcap	0.748	3682	0.595	0.987	0.552	0.23
Eurasian Blue Tit	0.711	2820	0.645	0.983	0.684	0.19
Eurasian Bullfinch	0.424	2339	0.742	0.996	0.671	0.12
Eurasian Collared-Dove	0.582	2684	0.708	0.952	0.713	0.14
Eurasian Coot	0.6	1735	0.639	0.976	0.609	0.14
Eurasian Crag-Martin	0.396	926	0.394	0.967	0.219	0.02
Eurasian Curlew	0.634	2009	0.86	0.996	0.832	0.16
Eurasian Dotterel	0.297	273	0.35	0.898	0.318	0.06
Eurasian Eagle-Owl	0.381	2305	0.629	0.976	0.673	0.14
Eurasian Golden Oriole	0.577	3420	0.56	0.98	0.593	0.12
Eurasian Green Woodpecker	0.609	1775	0.456	0.953	0.493	0.18
Eurasian Griffon	0.503	266	0.005	0.665	0.0	0.0
Eurasian Hobby	0.596	1195	0.739	0.987	0.777	0.07
Eurasian Hoopoe	0.545	2610	0.758	0.982	0.807	0.15
Eurasian Jackdaw	0.582	2084	0.877	0.99	0.866	0.16
Eurasian Jay	0.729	2872	0.45	0.965	0.411	0.12
Eurasian Kestrel	0.584	1760	0.709	0.972	0.698	0.18
Eurasian Linnet	0.634	2921	0.857	0.994	0.858	0.21
Eurasian Magpie	0.592	2326	0.746	0.986	0.758	0.17
Eurasian Marsh-Harrier	0.438	1307	0.634	0.933	0.57	0.09
Eurasian Moorhen	0.53	1220	0.709	0.971	0.694	0.16
Eurasian Nightjar	0.419	2289	0.888	0.995	0.898	0.09
Eurasian Nutcracker	0.698	1649	0.573	0.975	0.561	0.15
Eurasian Nuthatch	0.634	2870	0.693	0.991	0.753	0.2
Eurasian Oystercatcher	0.548	2113	0.828	0.984	0.806	0.21
Eurasian Penduline-Tit	0.354	1379	0.855	0.993	0.838	0.07
Eurasian Pygmy-Owl	0.497	2320	0.73	0.979	0.732	0.22
Eurasian Reed Warbler	0.948	3764	0.801	0.994	0.723	0.28
Eurasian River Warbler	0.539	3047	0.98	0.999	0.919	0.04
Eurasian Scops-Owl	0.342	2543	0.934	0.998	0.929	0.13
Eurasian Siskin	0.642	2574	0.739	0.972	0.735	0.1

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Eurasian Skylark	0.767	3290	0.843	0.992	0.824	0.21
Eurasian Sparrowhawk	0.509	786	0.374	0.898	0.405	0.17
Eurasian Spoonbill	0.626	136	0.058	0.856	0.069	0.04
Eurasian Thick-knee	0.39	1100	0.889	0.998	0.798	0.09
Eurasian Three-toed Woodpecker	0.68	2676	0.705	0.993	0.54	0.12
Eurasian Tree Sparrow	0.681	2686	0.822	0.992	0.723	0.09
Eurasian Treecreeper	0.478	2478	0.712	0.994	0.631	0.25
Eurasian Wigeon	0.502	1358	0.716	0.971	0.688	0.12
Eurasian Woodcock	0.43	704	0.775	0.977	0.811	0.13
Eurasian Wren	0.711	3476	0.674	0.982	0.693	0.18
Eurasian Wryneck	0.618	2718	0.511	0.967	0.633	0.2
European Bee-eater	0.416	2282	0.839	0.994	0.894	0.13
European Golden-Plover	0.46	1364	0.769	0.989	0.756	0.17
European Goldfinch	0.659	3349	0.825	0.995	0.779	0.08
European Greenfinch	0.674	3237	0.778	0.986	0.782	0.2
European Honey-buzzard	0.447	520	0.569	0.975	0.436	0.15
European Pied Flycatcher	0.657	2776	0.692	0.987	0.704	0.18
European Robin	0.695	3842	0.702	0.989	0.661	0.17
European Roller	0.599	881	0.681	0.983	0.573	0.1
European Serin	0.666	3231	0.862	0.993	0.854	0.12
European Starling	0.707	3934	0.509	0.959	0.426	0.14
European Stonechat	0.476	2288	0.772	0.993	0.74	0.2
European Storm-Petrel	0.548	2049	0.958	0.998	0.944	0.08
European Turtle-Dove	0.582	1793	0.766	0.986	0.773	0.07
Evening Grosbeak	0.48	2524	0.867	0.996	0.879	0.12
Ferruginous Duck	0.464	72	0.023	0.806	0.0	0.0
Field Sparrow	0.457	1886	0.508	0.963	0.495	0.12
Fieldfare	0.642	2723	0.809	0.984	0.801	0.11
Finsch's Wheatear	0.579	79	0.019	0.939	0.0	0.0
Fire-fronted Serin	0.712	237	0.835	0.972	0.741	0.05
Fish Crow	0.51	1572	0.718	0.97	0.701	0.07
Flammulated Owl	0.284	2116	0.808	0.992	0.827	0.14
Florida Scrub-Jay	0.722	1554	0.773	0.981	0.847	0.13
Forster's Tern	0.675	1137	0.568	0.968	0.595	0.25
Fox Sparrow	0.505	2704	0.525	0.973	0.513	0.17
Franklin's Gull	0.925	1201	0.683	0.951	0.707	0.09
Fulvous Chatterer	0.59	162	0.002	0.467	0.0	0.0
Fulvous Whistling-Duck	0.728	559	0.52	0.912	0.566	0.11
Gadwall	0.572	1601	0.546	0.959	0.619	0.15
Gambel's Quail	0.639	2690	0.637	0.973	0.658	0.15
Garden Warbler	0.951	3995	0.718	0.993	0.658	0.29

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Garganey	0.567	497	0.598	0.95	0.573	0.11
Gila Woodpecker	0.741	1696	0.603	0.966	0.7	0.19
Gilded Flicker	0.513	936	0.684	0.959	0.615	0.07
Glaucous Gull	0.932	1085	0.548	0.988	0.46	0.08
Glaucous-winged Gull	0.695	1520	0.723	0.991	0.584	0.1
Glossy Ibis	0.769	659	0.435	0.903	0.387	0.15
Goldcrest	0.498	3040	0.772	0.993	0.714	0.15
Golden Eagle	0.646	898	0.763	0.963	0.78	0.15
Golden-cheeked Warbler	0.64	3516	0.805	0.995	0.836	0.18
Golden-crowned Kinglet	0.363	1725	0.663	0.986	0.536	0.1
Golden-crowned Sparrow	0.521	2545	0.581	0.976	0.65	0.16
Golden-fronted Woodpecker	0.626	1819	0.573	0.957	0.608	0.11
Golden-winged Warbler	0.633	2743	0.437	0.956	0.421	0.11
Grace’s Warbler	0.637	2563	0.485	0.98	0.501	0.12
Graceful Prinia	0.586	877	0.458	0.899	0.322	0.07
Grasshopper Sparrow	0.603	3044	0.546	0.98	0.514	0.14
Gray Catbird	0.679	3591	0.722	0.991	0.701	0.15
Gray Flycatcher	0.571	3077	0.661	0.992	0.69	0.15
Gray Hawk	0.584	1345	0.406	0.936	0.482	0.13
Gray Heron	0.56	1723	0.549	0.949	0.562	0.18
Gray Kingbird	0.539	2181	0.637	0.966	0.725	0.12
Gray Partridge	0.608	2171	0.784	0.985	0.754	0.12
Gray Vireo	0.601	1469	0.849	0.974	0.855	0.18
Gray Wagtail	0.355	1596	0.84	0.995	0.825	0.06
Gray-cheeked Thrush	0.573	981	0.558	0.926	0.528	0.09
Gray-crowned Rosy-Finch	0.572	1074	0.649	0.985	0.646	0.17
Gray-headed Chickadee	0.775	767	0.889	0.997	0.87	0.13
Gray-headed Swamphen	0.703	293	0.21	0.968	0.121	0.02
Gray-headed Woodpecker	0.517	2322	0.51	0.981	0.475	0.11
Gray-necked Bunting	0.534	303	0.44	0.846	0.4	0.04
Graylag Goose	0.792	1980	0.836	0.996	0.749	0.18
Great Bittern	0.479	1797	0.66	0.973	0.612	0.11
Great Black-backed Gull	0.876	1934	0.728	0.99	0.659	0.13
Great Blue Heron	0.703	1739	0.752	0.985	0.75	0.21
Great Cormorant	0.556	1059	0.572	0.958	0.529	0.14
Great Crested Flycatcher	0.437	2677	0.711	0.991	0.765	0.14
Great Crested Grebe	0.605	1025	0.748	0.982	0.69	0.14
Great Egret	0.675	1781	0.612	0.97	0.623	0.28
Great Gray Owl	0.607	1576	0.493	0.939	0.482	0.14
Great Gray Shrike	0.527	1909	0.718	0.989	0.63	0.17
Great Horned Owl	0.364	2405	0.665	0.978	0.716	0.13

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Great Kiskadee	0.709	3051	0.749	0.985	0.769	0.09
Great Reed Warbler	0.835	3717	0.838	0.994	0.832	0.21
Great Skua	0.558	455	0.392	0.95	0.478	0.07
Great Snipe	0.625	1471	0.912	0.998	0.904	0.09
Great Spotted Cuckoo	0.736	722	0.545	0.912	0.532	0.07
Great Spotted Woodpecker	0.639	2970	0.637	0.988	0.587	0.2
Great Tit	0.553	3769	0.61	0.985	0.554	0.1
Great White Pelican	0.646	190	0.317	0.96	0.125	0.03
Great-tailed Grackle	0.56	2838	0.484	0.956	0.47	0.22
Greater Flamingo	0.608	1344	0.781	0.995	0.773	0.16
Greater Hoopoe-Lark	0.466	375	0.646	0.947	0.743	0.09
Greater Pewee	0.518	2190	0.776	0.99	0.709	0.09
Greater Prairie-Chicken	0.866	2137	0.963	0.999	0.94	0.06
Greater Roadrunner	0.577	1758	0.315	0.926	0.327	0.11
Greater Sage-Grouse	0.718	1838	0.944	0.999	0.948	0.05
Greater Scaup	0.717	830	0.621	0.995	0.612	0.1
Greater Short-toed Lark	0.54	1787	0.748	0.993	0.701	0.1
Greater Spotted Eagle	0.552	630	0.816	0.991	0.726	0.14
Greater White-fronted Goose	0.74	2142	0.762	0.982	0.704	0.15
Greater Whitethroat	0.747	3366	0.643	0.986	0.665	0.22
Greater Yellowlegs	0.672	1976	0.732	0.981	0.725	0.09
Green Heron	0.523	1253	0.465	0.95	0.51	0.16
Green Jay	0.634	2129	0.776	0.99	0.823	0.15
Green Kingfisher	0.475	1031	0.459	0.862	0.434	0.07
Green Parakeet	0.793	1025	0.372	0.876	0.452	0.1
Green Sandpiper	0.475	1248	0.724	0.963	0.647	0.07
Green Warbler	0.478	1270	0.831	0.993	0.736	0.08
Green-tailed Towhee	0.708	3555	0.542	0.983	0.609	0.25
Green-winged Teal	0.458	1914	0.594	0.97	0.656	0.18
Greenish Warbler	0.555	2897	0.627	0.976	0.629	0.18
Groove-billed Ani	0.567	1709	0.504	0.963	0.587	0.15
Gull-billed Tern	0.739	1130	0.643	0.976	0.566	0.09
Hairy Woodpecker	0.482	2560	0.673	0.984	0.68	0.16
Hammond's Flycatcher	0.469	2559	0.698	0.991	0.747	0.15
Harlequin Duck	0.609	376	0.534	0.945	0.629	0.17
Harris's Hawk	0.637	1001	0.413	0.892	0.46	0.18
Harris's Sparrow	0.562	1194	0.583	0.938	0.557	0.13
Hawfinch	0.391	1801	0.764	0.992	0.744	0.11
Hazel Grouse	0.413	1867	0.66	0.982	0.706	0.14
Heermann's Gull	0.424	503	0.757	0.981	0.798	0.09
Henslow's Sparrow	0.425	2013	0.638	0.984	0.651	0.19

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Hepatic Tanager	0.591	2604	0.685	0.992	0.668	0.1
Hermit Thrush	0.52	3071	0.856	0.993	0.866	0.19
Hermit Warbler	0.61	2253	0.541	0.989	0.38	0.07
Herring Gull	0.762	2802	0.683	0.991	0.577	0.25
Hoary Redpoll	0.824	1163	0.749	0.999	0.593	0.11
Hooded Crow	0.698	1616	0.454	0.971	0.463	0.12
Hooded Merganser	0.53	675	0.429	0.86	0.49	0.09
Hooded Oriole	0.594	2934	0.595	0.959	0.588	0.2
Hooded Warbler	0.498	2567	0.559	0.988	0.581	0.22
Horned Grebe	0.869	909	0.279	0.853	0.271	0.11
Horned Lark	0.518	2651	0.764	0.988	0.719	0.09
House Bunting	0.644	731	0.739	0.975	0.68	0.09
House Finch	0.53	2795	0.764	0.978	0.739	0.11
House Sparrow	0.665	3583	0.728	0.99	0.61	0.18
House Wren	0.737	3455	0.576	0.972	0.634	0.28
Hudsonian Godwit	0.695	716	0.171	0.863	0.107	0.04
Hutton's Vireo	0.537	2768	0.78	0.995	0.777	0.19
Iberian Chiffchaff	0.481	2528	0.691	0.995	0.712	0.16
Iberian Magpie	0.539	1516	0.876	0.997	0.855	0.07
Icterine Warbler	0.924	3601	0.66	0.989	0.63	0.19
Inca Dove	0.491	1834	0.602	0.926	0.62	0.1
Indigo Bunting	0.535	2526	0.558	0.979	0.564	0.1
Isabelline Shrike	0.712	186	0.066	0.785	0.057	0.01
Isabelline Wheatear	0.787	1345	0.652	0.973	0.632	0.14
Island Scrub-Jay	0.554	978	0.799	0.987	0.814	0.06
Italian Sparrow	0.721	1013	0.393	0.984	0.338	0.06
Jack Snipe	0.346	193	0.263	0.948	0.214	0.06
Juniper Titmouse	0.769	1856	0.657	0.982	0.586	0.05
Kentish Plover	0.426	545	0.769	0.983	0.726	0.09
Kentucky Warbler	0.55	2721	0.587	0.989	0.545	0.25
Killdeer	0.536	2246	0.836	0.981	0.83	0.13
King Eider	0.615	816	0.8	0.999	0.62	0.05
King Rail	0.575	2159	0.713	0.985	0.67	0.25
Kirtland's Warbler	0.676	2663	0.751	0.995	0.8	0.14
Krueper's Nuthatch	0.555	420	0.474	0.854	0.45	0.13
Ladder-backed Woodpecker	0.475	1848	0.522	0.969	0.593	0.24
Lanceolated Warbler	0.521	977	0.855	0.978	0.853	0.09
Lapland Longspur	0.49	1523	0.541	0.954	0.529	0.18
Lark Bunting	0.802	2354	0.783	0.993	0.804	0.19
Lark Sparrow	0.631	2962	0.611	0.982	0.633	0.13
Laughing Dove	0.582	1594	0.747	0.979	0.792	0.13

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Laughing Gull	0.832	2190	0.814	0.981	0.718	0.1
Lawrence's Goldfinch	0.687	1923	0.836	0.991	0.808	0.14
Lazuli Bunting	0.617	2753	0.495	0.976	0.513	0.16
LeConte's Sparrow	0.547	1848	0.691	0.989	0.648	0.21
LeConte's Thrasher	0.891	2110	0.745	0.993	0.801	0.19
Leach's Storm-Petrel	0.813	3486	0.911	0.998	0.898	0.11
Least Bittern	0.501	1639	0.445	0.955	0.356	0.1
Least Flycatcher	0.485	2537	0.876	0.997	0.836	0.09
Least Grebe	0.556	957	0.588	0.963	0.566	0.2
Least Sandpiper	0.593	1088	0.621	0.959	0.639	0.1
Least Tern	0.584	1661	0.778	0.969	0.725	0.09
Lesser Black-backed Gull	0.793	1509	0.704	0.995	0.541	0.12
Lesser Goldfinch	0.516	3283	0.668	0.979	0.633	0.11
Lesser Gray Shrike	0.55	268	0.024	0.734	0.013	0.02
Lesser Kestrel	0.581	1571	0.883	0.997	0.866	0.11
Lesser Nighthawk	0.515	1346	0.709	0.95	0.67	0.11
Lesser Prairie-Chicken	0.945	1663	0.801	0.997	0.712	0.14
Lesser Redpoll	0.609	595	0.485	0.985	0.422	0.05
Lesser Short-toed Lark	0.695	1211	0.646	0.97	0.629	0.15
Lesser Spotted Eagle	0.602	737	0.796	0.982	0.665	0.08
Lesser Spotted Woodpecker	0.732	2315	0.513	0.978	0.516	0.17
Lesser White-fronted Goose	0.527	54	1.0	1.0	0.375	0.02
Lesser Whitethroat	0.661	2836	0.504	0.983	0.494	0.15
Lesser Yellowlegs	0.631	2301	0.768	0.991	0.722	0.12
Levaillant's Woodpecker	0.689	217	0.3	0.775	0.379	0.16
Lewis's Woodpecker	0.764	1570	0.614	0.96	0.636	0.1
Limpkin	0.73	2705	0.805	0.991	0.793	0.09
Lincoln's Sparrow	0.548	2575	0.486	0.979	0.455	0.17
Little Bittern	0.415	1222	0.898	0.999	0.782	0.1
Little Blue Heron	0.731	547	0.46	0.957	0.33	0.09
Little Bunting	0.506	745	0.537	0.952	0.542	0.13
Little Bustard	0.399	379	0.362	0.957	0.334	0.09
Little Crake	0.511	1702	0.638	0.971	0.552	0.08
Little Egret	0.732	1313	0.502	0.96	0.48	0.09
Little Grebe	0.594	1666	0.522	0.965	0.505	0.09
Little Gull	0.81	397	0.631	0.941	0.562	0.08
Little Owl	0.554	2658	0.781	0.992	0.783	0.14
Little Ringed Plover	0.516	1830	0.718	0.975	0.714	0.12
Little Stint	0.62	361	0.329	0.906	0.221	0.09
Little Swift	0.613	1248	0.939	0.995	0.94	0.15
Little Tern	0.603	1361	0.902	0.996	0.888	0.06

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Loggerhead Shrike	0.486	2701	0.758	0.99	0.683	0.08
Long-billed Curlew	0.756	1464	0.584	0.978	0.569	0.17
Long-billed Dowitcher	0.556	1309	0.45	0.971	0.458	0.17
Long-billed Thrasher	0.822	2170	0.699	0.985	0.709	0.29
Long-eared Owl	0.528	3249	0.725	0.984	0.73	0.21
Long-legged Buzzard	0.598	89	0.01	0.77	0.0	0.0
Long-tailed Duck	0.682	1494	0.705	0.988	0.652	0.08
Long-tailed Jaeger	0.858	1575	0.861	0.997	0.872	0.16
Long-tailed Tit	0.513	3069	0.884	0.994	0.875	0.12
Louisiana Waterthrush	0.507	1886	0.622	0.983	0.645	0.07
Lucy’s Warbler	0.644	2357	0.541	0.973	0.539	0.09
MacGillivray’s Warbler	0.577	2855	0.634	0.987	0.674	0.16
Magnificent Frigatebird	0.513	1382	0.955	0.999	0.948	0.08
Magnolia Warbler	0.535	2244	0.508	0.98	0.567	0.24
Mallard	0.667	2490	0.746	0.976	0.79	0.24
Mandarin Duck	0.472	427	0.086	0.682	0.057	0.02
Mangrove Cuckoo	0.614	730	0.386	0.855	0.458	0.12
Manx Shearwater	0.47	1954	0.845	0.969	0.824	0.13
Marbled Godwit	1.028	1473	0.795	0.975	0.729	0.1
Marbled Murrelet	0.501	2468	0.795	0.993	0.808	0.09
Marmora’s Warbler	0.644	584	0.739	0.914	0.721	0.11
Marsh Sandpiper	0.667	443	0.888	0.999	0.487	0.05
Marsh Tit	0.645	2912	0.687	0.989	0.688	0.13
Marsh Warbler	0.96	4059	0.637	0.978	0.579	0.15
Marsh Wren	0.726	3288	0.789	0.986	0.739	0.09
Masked Shrike	0.534	318	0.241	0.939	0.165	0.14
McCown’s Longspur	0.705	1694	0.773	0.995	0.743	0.13
Meadow Pipit	0.453	1666	0.841	0.993	0.793	0.1
Mediterranean Gull	0.641	435	0.626	0.866	0.678	0.1
Melodious Warbler	0.768	3179	0.678	0.992	0.664	0.2
Menetries’s Warbler	0.769	617	0.581	0.959	0.408	0.1
Merlin	0.566	1744	0.58	0.929	0.64	0.17
Mew Gull	0.836	2945	0.719	0.977	0.681	0.12
Mexican Chickadee	0.678	1649	0.744	0.984	0.734	0.06
Mexican Jay	0.863	1970	0.858	0.993	0.883	0.14
Mexican Whip-poor-will	0.526	2587	0.938	0.997	0.942	0.06
Middle Spotted Woodpecker	0.58	2315	0.771	0.995	0.814	0.13
Mississippi Kite	0.362	483	0.688	0.954	0.72	0.14
Mistle Thrush	0.628	3473	0.799	0.996	0.784	0.17
Moltoni’s Warbler	0.788	860	0.697	0.979	0.625	0.08
Monk Parakeet	0.705	1923	0.799	0.985	0.836	0.15

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Montagu’s Harrier	0.618	479	0.732	0.996	0.591	0.06
Montezuma Quail	0.521	1169	0.459	0.953	0.381	0.05
Mountain Bluebird	0.76	1790	0.683	0.987	0.678	0.11
Mountain Chickadee	0.603	2673	0.761	0.987	0.763	0.12
Mountain Chiffchaff	0.516	442	1.0	1.0	0.9	0.01
Mountain Quail	0.558	2746	0.674	0.99	0.7	0.14
Mourning Dove	0.492	2225	0.522	0.967	0.525	0.08
Mourning Warbler	0.531	2253	0.483	0.982	0.487	0.15
Moussier’s Redstart	0.53	183	0.213	0.84	0.13	0.02
Moustached Warbler	0.637	1670	0.677	0.995	0.568	0.16
Mute Swan	0.502	1483	0.37	0.911	0.409	0.14
Namaqua Dove	0.578	300	0.004	0.546	0.0	0.0
Nashville Warbler	0.575	2512	0.536	0.984	0.538	0.17
Nelson’s Sparrow	0.613	1597	0.529	0.894	0.496	0.08
Neotropic Cormorant	0.544	934	0.802	0.988	0.867	0.13
Northern Bald Ibis	0.585	296	0.652	0.98	0.5	0.04
Northern Beardless-Tyrannulet	0.544	1827	0.67	0.978	0.747	0.11
Northern Bobwhite	0.49	1348	0.53	0.985	0.501	0.1
Northern Cardinal	0.584	3215	0.656	0.986	0.578	0.09
Northern Flicker	0.512	2431	0.464	0.946	0.486	0.14
Northern Fulmar	0.459	1171	0.923	0.999	0.795	0.09
Northern Gannet	0.381	2141	0.975	0.999	0.948	0.18
Northern Goshawk	0.613	2413	0.467	0.975	0.547	0.2
Northern Harrier	0.665	991	0.395	0.825	0.477	0.17
Northern Hawk Owl	0.693	1479	0.51	0.965	0.45	0.09
Northern Lapwing	0.798	2149	0.778	0.988	0.812	0.16
Northern Mockingbird	0.609	3833	0.443	0.962	0.351	0.15
Northern Parula	0.548	2307	0.559	0.993	0.568	0.13
Northern Pintail	0.613	874	0.427	0.96	0.439	0.09
Northern Pygmy-Owl	0.384	1683	0.816	0.988	0.774	0.11
Northern Rough-winged Swallow	0.566	1728	0.673	0.985	0.72	0.13
Northern Saw-whet Owl	0.281	2096	0.778	0.992	0.786	0.15
Northern Shoveler	0.608	1035	0.587	0.964	0.667	0.19
Northern Shrike	0.542	1238	0.548	0.981	0.438	0.14
Northern Waterthrush	0.54	2729	0.655	0.983	0.663	0.11
Northern Wheatear	0.596	2354	0.689	0.987	0.67	0.25
Northwestern Crow	0.715	2128	0.66	0.947	0.679	0.22
Nuttall’s Woodpecker	0.589	1351	0.357	0.947	0.331	0.07
Oak Titmouse	0.646	2765	0.764	0.992	0.795	0.27
Olive Sparrow	0.647	2241	0.711	0.988	0.786	0.16
Olive Warbler	0.581	2210	0.407	0.945	0.42	0.1

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Olive-backed Pipit	0.547	1698	0.72	0.977	0.75	0.13
Olive-sided Flycatcher	0.505	2842	0.534	0.977	0.597	0.12
Olive-tree Warbler	0.702	198	0.454	0.849	0.507	0.06
Orange-crowned Warbler	0.551	2983	0.455	0.966	0.509	0.11
Orchard Oriole	0.525	2414	0.543	0.975	0.512	0.13
Ortolan Bunting	0.484	2355	0.651	0.984	0.641	0.19
Osprey	0.453	2112	0.766	0.985	0.752	0.16
Ovenbird	0.579	2326	0.502	0.982	0.518	0.12
Pacific Golden-Plover	0.61	1074	0.665	0.983	0.695	0.11
Pacific Loon	0.842	895	0.439	0.895	0.449	0.1
Pacific Wren	0.527	2085	0.783	0.988	0.829	0.21
Pacific-slope Flycatcher	0.449	2791	0.676	0.99	0.67	0.22
Paddyfield Warbler	0.689	1504	0.515	0.982	0.479	0.18
Painted Bunting	0.531	2224	0.639	0.98	0.665	0.16
Painted Redstart	0.518	2742	0.596	0.971	0.682	0.19
Pale Rockfinch	0.731	255	0.889	0.989	0.888	0.04
Pallas's Gull	0.719	49	0.769	0.983	0.67	0.09
Pallas's Leaf Warbler	0.609	1549	0.864	0.994	0.877	0.22
Pallid Harrier	0.412	131	0.001	0.191	0.0	0.0
Pallid Scops-Owl	0.469	255	0.001	0.266	0.0	0.0
Pallid Swift	0.596	591	0.865	0.984	0.811	0.1
Palm Warbler	0.526	1495	0.309	0.954	0.318	0.14
Parasitic Jaeger	0.83	1023	0.636	0.983	0.657	0.12
Parrot Crossbill	0.464	1407	0.769	0.963	0.675	0.11
Pechora Pipit	0.759	208	0.155	0.967	0.092	0.01
Pectoral Sandpiper	0.604	884	0.434	0.969	0.43	0.1
Peregrine Falcon	0.533	1624	0.775	0.986	0.818	0.13
Phainopepla	0.566	2225	0.678	0.985	0.693	0.13
Pharaoh Eagle-Owl	0.34	191	0.147	0.993	0.081	0.02
Philadelphia Vireo	0.559	2421	0.663	0.988	0.645	0.12
Pied Avocet	0.586	2401	0.811	0.996	0.751	0.15
Pied Kingfisher	0.6	627	0.51	0.881	0.517	0.09
Pied Wheatear	0.586	517	0.483	0.959	0.393	0.07
Pied-billed Grebe	0.643	1786	0.63	0.97	0.63	0.11
Pigeon Guillemot	0.568	1229	0.542	0.974	0.561	0.08
Pileated Woodpecker	0.587	2243	0.709	0.991	0.677	0.14
Pin-tailed Sandgrouse	0.751	282	0.847	0.995	0.789	0.06
Pine Grosbeak	0.516	2071	0.406	0.932	0.421	0.16
Pine Siskin	0.551	2736	0.778	0.98	0.713	0.09
Pine Warbler	0.505	2046	0.532	0.981	0.598	0.19
Pink-footed Goose	0.604	907	0.714	0.969	0.695	0.17

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Pinyon Jay	0.797	1886	0.914	0.997	0.901	0.11
Piping Plover	0.451	627	0.609	0.968	0.611	0.15
Plain Chachalaca	0.706	2590	0.763	0.992	0.79	0.11
Plumbeous Vireo	0.549	3361	0.661	0.99	0.571	0.16
Prairie Warbler	0.472	1891	0.459	0.977	0.498	0.13
Prothonotary Warbler	0.554	1850	0.695	0.993	0.698	0.15
Purple Finch	0.481	2756	0.507	0.978	0.513	0.2
Purple Gallinule	0.576	1696	0.805	0.998	0.641	0.09
Purple Heron	0.462	420	0.076	0.637	0.109	0.05
Purple Martin	0.66	2667	0.863	0.988	0.863	0.14
Purple Sandpiper	0.546	938	0.269	0.951	0.176	0.08
Pygmy Nuthatch	0.55	2423	0.749	0.984	0.843	0.13
Pyrrhuloxia	0.618	2461	0.567	0.972	0.572	0.2
Radde's Accentor	0.635	41	0.146	0.715	0.133	0.02
Razorbill	0.387	1774	0.947	0.999	0.746	0.05
Red Crossbill	0.494	2731	0.779	0.99	0.749	0.21
Red Junglefowl	0.66	1634	0.518	0.968	0.496	0.08
Red Kite	0.56	491	0.714	0.975	0.645	0.1
Red Knot	0.709	1964	0.695	0.993	0.61	0.14
Red Phalarope	0.719	1379	0.739	0.989	0.732	0.08
Red-backed Shrike	0.54	2228	0.53	0.981	0.453	0.12
Red-bellied Woodpecker	0.522	2162	0.575	0.986	0.62	0.19
Red-billed Chough	0.662	1474	0.743	0.984	0.777	0.12
Red-billed Firefinch	0.645	472	0.271	0.711	0.386	0.11
Red-billed Pigeon	0.564	1615	0.582	0.969	0.486	0.12
Red-breasted Flycatcher	0.6	2999	0.729	0.975	0.693	0.15
Red-breasted Merganser	0.575	414	0.028	0.814	0.0	0.0
Red-breasted Nuthatch	0.642	3036	0.874	0.996	0.867	0.21
Red-breasted Sapsucker	0.697	1726	0.277	0.953	0.227	0.06
Red-cockaded Woodpecker	0.504	2368	0.781	0.993	0.806	0.17
Red-crested Pochard	0.342	206	0.751	0.979	0.642	0.04
Red-crowned Parrot	0.681	976	0.934	0.998	0.915	0.05
Red-eyed Vireo	0.586	3737	0.847	0.997	0.779	0.2
Red-faced Warbler	0.633	1973	0.699	0.99	0.663	0.15
Red-flanked Bluetail	0.48	780	0.803	0.985	0.849	0.25
Red-footed Falcon	0.662	273	0.446	0.789	0.545	0.06
Red-headed Bunting	0.571	512	0.47	0.946	0.366	0.11
Red-headed Woodpecker	0.549	2596	0.709	0.988	0.711	0.11
Red-legged Kittiwake	0.376	1646	0.974	0.999	0.923	0.08
Red-legged Partridge	0.706	1930	0.836	0.993	0.832	0.13
Red-naped Sapsucker	0.71	1678	0.402	0.982	0.377	0.07

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Red-necked Grebe	0.776	1243	0.732	0.986	0.681	0.09
Red-necked Nightjar	0.393	851	0.913	0.998	0.897	0.05
Red-necked Phalarope	0.711	1618	0.694	0.971	0.669	0.06
Red-rumped Swallow	0.575	1380	0.667	0.968	0.704	0.11
Red-rumped Wheatear	0.418	78	0.295	0.739	0.317	0.12
Red-shouldered Hawk	0.652	1682	0.634	0.971	0.622	0.1
Red-tailed Hawk	0.591	1386	0.483	0.951	0.505	0.12
Red-tailed Shrike	0.424	147	0.007	0.592	0.0	0.0
Red-throated Loon	0.66	1937	0.624	0.986	0.69	0.21
Red-throated Pipit	0.45	592	0.566	0.934	0.597	0.08
Red-wattled Lapwing	0.796	1519	0.691	0.962	0.729	0.15
Red-whiskered Bulbul	0.539	3061	0.674	0.983	0.67	0.24
Red-winged Blackbird	0.592	3396	0.648	0.981	0.587	0.1
Redhead	0.603	796	0.315	0.975	0.23	0.11
Redwing	0.645	2561	0.74	0.986	0.721	0.12
Reed Bunting	0.543	2551	0.711	0.988	0.706	0.15
Richard’s Pipit	0.444	525	0.837	0.984	0.779	0.09
Ridgway’s Rail	0.541	1237	0.608	0.99	0.497	0.1
Ring Ouzel	0.509	2083	0.695	0.987	0.698	0.17
Ring-billed Gull	0.697	1842	0.774	0.982	0.699	0.12
Ring-necked Duck	0.653	1703	0.754	0.994	0.636	0.14
Ring-necked Pheasant	0.595	2144	0.608	0.968	0.664	0.19
Ringed Kingfisher	0.629	1961	0.621	0.975	0.696	0.16
Rivoli’s Hummingbird	0.474	1288	0.862	0.998	0.813	0.04
Rock Bunting	0.527	1809	0.712	0.967	0.779	0.24
Rock Martin	0.627	89	0.007	0.652	0.0	0.0
Rock Partridge	0.519	312	0.696	0.927	0.692	0.11
Rock Pigeon	0.448	1540	0.688	0.98	0.723	0.15
Rock Pipit	0.503	621	0.851	0.988	0.79	0.12
Rock Ptarmigan	0.645	1327	0.336	0.952	0.35	0.12
Rock Sandpiper	0.72	979	0.539	0.966	0.41	0.13
Rock Sparrow	0.682	1403	0.757	0.986	0.762	0.07
Rock Wren	0.583	3039	0.623	0.978	0.701	0.19
Rook	0.583	2734	0.805	0.994	0.755	0.18
Rose-breasted Grosbeak	0.542	2717	0.618	0.985	0.618	0.13
Rose-ringed Parakeet	0.621	2351	0.721	0.978	0.731	0.16
Roseate Spoonbill	0.543	920	0.891	0.999	0.783	0.07
Roseate Tern	0.525	976	0.706	0.99	0.638	0.11
Ross’s Goose	0.631	945	0.789	0.998	0.698	0.09
Ross’s Gull	1.143	699	0.162	0.768	0.176	0.01
Rosy Starling	0.549	799	0.611	0.939	0.645	0.1

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Rough-legged Hawk	0.827	1248	0.618	0.981	0.547	0.09
Royal Tern	0.56	1603	0.822	0.994	0.755	0.13
Ruby-crowned Kinglet	0.535	2449	0.806	0.991	0.823	0.13
Ruby-throated Hummingbird	0.392	1307	0.566	0.951	0.57	0.14
Ruddy Duck	0.63	1176	0.471	0.967	0.462	0.13
Ruddy Shelduck	0.654	1224	0.769	0.966	0.702	0.11
Ruddy Turnstone	0.568	1179	0.514	0.969	0.483	0.12
Rueppell's Warbler	0.694	370	0.893	0.999	0.742	0.06
Ruff	0.413	99	0.004	0.727	0.0	0.0
Ruffed Grouse	0.55	1899	0.507	0.982	0.588	0.11
Rufous Hummingbird	0.478	956	0.534	0.913	0.469	0.07
Rufous-crowned Sparrow	0.643	3010	0.511	0.973	0.658	0.17
Rufous-tailed Rock-Thrush	0.431	638	0.382	0.909	0.4	0.14
Rufous-tailed Scrub-Robin	0.666	2501	0.691	0.989	0.71	0.22
Rufous-winged Sparrow	0.608	2282	0.787	0.993	0.818	0.13
Rustic Bunting	0.523	1017	0.871	0.996	0.806	0.05
Rusty Blackbird	0.454	2118	0.869	0.996	0.856	0.1
Sabine's Gull	0.735	798	0.569	0.975	0.598	0.15
Sage Thrasher	0.824	3000	0.593	0.99	0.572	0.19
Sagebrush Sparrow	0.648	1850	0.686	0.994	0.665	0.15
Sanderling	0.513	1187	0.59	0.956	0.607	0.12
Sandhill Crane	0.791	2735	0.863	0.992	0.873	0.23
Sandwich Tern	0.6	1183	0.726	0.984	0.698	0.18
Sardinian Warbler	0.694	2710	0.727	0.986	0.707	0.24
Savannah Sparrow	0.539	2659	0.725	0.988	0.705	0.16
Savi's Warbler	0.407	2440	0.861	0.999	0.817	0.15
Say's Phoebe	0.476	1979	0.707	0.972	0.761	0.08
Scaled Quail	0.553	1364	0.67	0.977	0.679	0.13
Scarlet Tanager	0.542	3275	0.656	0.985	0.715	0.23
Scissor-tailed Flycatcher	0.682	1281	0.562	0.91	0.613	0.14
Scott's Oriole	0.559	2992	0.49	0.969	0.501	0.23
Scrub Warbler	0.758	364	0.309	0.866	0.277	0.08
Seaside Sparrow	0.614	2371	0.631	0.986	0.623	0.19
Sedge Warbler	0.943	3700	0.85	0.995	0.839	0.28
Sedge Wren	0.669	3365	0.671	0.982	0.6	0.11
Semicollared Flycatcher	0.42	993	0.302	0.946	0.196	0.05
Semipalmated Plover	0.609	1866	0.745	0.988	0.782	0.16
Semipalmated Sandpiper	0.699	1913	0.581	0.971	0.545	0.13
Sharp-shinned Hawk	0.571	1437	0.144	0.812	0.192	0.11
Sharp-tailed Grouse	0.759	2518	0.782	0.992	0.821	0.2
Short-billed Dowitcher	0.652	758	0.38	0.956	0.279	0.13

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Short-eared Owl	0.629	1345	0.382	0.964	0.406	0.09
Short-tailed Hawk	0.495	938	0.328	0.949	0.311	0.07
Short-toed Snake-Eagle	0.409	293	0.273	0.731	0.328	0.14
Short-toed Treecreeper	0.447	2287	0.661	0.982	0.65	0.08
Siberian Jay	0.553	693	0.377	0.944	0.208	0.05
Siberian Rubythroat	0.728	2019	0.653	0.97	0.672	0.25
Siberian Stonechat	0.529	524	0.504	0.959	0.431	0.08
Slender-billed Gull	0.674	1093	0.696	0.994	0.622	0.08
Smith's Longspur	0.521	1465	0.553	0.991	0.628	0.08
Snail Kite	0.587	901	0.376	0.876	0.409	0.19
Snow Bunting	0.593	1823	0.615	0.975	0.628	0.13
Snow Goose	0.628	2494	0.818	0.993	0.802	0.25
Snowy Egret	0.725	1135	0.53	0.981	0.422	0.17
Snowy Owl	0.623	1235	0.548	0.962	0.332	0.04
Snowy Plover	0.516	974	0.37	0.923	0.451	0.15
Solitary Sandpiper	0.499	1289	0.609	0.923	0.573	0.1
Sombre Tit	0.584	431	0.145	0.843	0.072	0.04
Song Sparrow	0.593	3176	0.542	0.977	0.599	0.28
Song Thrush	0.751	3816	0.541	0.985	0.425	0.18
Sooty Grouse	0.59	1114	0.379	0.963	0.416	0.16
Sooty Shearwater	0.593	337	0.75	0.95	0.468	0.04
Sooty Tern	0.657	1134	0.813	0.988	0.789	0.09
Sora	0.592	2185	0.766	0.993	0.805	0.24
South Polar Skua	0.812	218	0.01	0.875	0.0	0.0
Spanish Sparrow	0.591	1822	0.753	0.981	0.702	0.12
Spectacled Warbler	0.624	1312	0.611	0.924	0.587	0.15
Spotless Starling	0.561	2446	0.575	0.979	0.443	0.13
Spotted Crake	0.589	2240	0.858	0.991	0.855	0.14
Spotted Flycatcher	0.533	2320	0.767	0.984	0.786	0.2
Spotted Owl	0.537	2170	0.67	0.985	0.644	0.07
Spotted Redshank	0.559	553	0.563	0.933	0.495	0.07
Spotted Sandgrouse	0.518	721	0.875	0.999	0.857	0.05
Spotted Sandpiper	0.347	682	0.456	0.831	0.456	0.07
Spotted Towhee	0.658	3163	0.614	0.979	0.59	0.17
Sprague's Pipit	0.764	1605	0.675	0.992	0.614	0.11
Spruce Grouse	0.776	1096	0.625	0.992	0.715	0.13
Spur-winged Lapwing	0.714	559	0.795	0.994	0.72	0.06
Squacco Heron	0.688	198	0.004	0.534	0.0	0.0
Steller's Jay	0.727	3022	0.696	0.984	0.722	0.14
Stilt Sandpiper	0.897	845	0.57	0.962	0.54	0.15
Stock Dove	0.518	1305	0.579	0.986	0.515	0.07

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Subalpine Warbler	0.711	2580	0.751	0.993	0.7	0.22
Sulphur-bellied Flycatcher	0.668	2748	0.665	0.971	0.746	0.16
Summer Tanager	0.469	2973	0.74	0.991	0.807	0.21
Surfbird	0.691	1195	0.924	0.998	0.942	0.06
Swainson's Hawk	0.76	981	0.369	0.899	0.272	0.06
Swainson's Thrush	0.59	2441	0.728	0.983	0.718	0.17
Swainson's Warbler	0.552	2424	0.597	0.991	0.625	0.2
Swallow-tailed Kite	0.537	917	0.603	0.966	0.644	0.08
Swamp Sparrow	0.541	1875	0.666	0.982	0.626	0.12
Sykes's Warbler	0.667	1202	0.698	0.966	0.661	0.18
Syrian Woodpecker	0.492	619	0.455	0.939	0.441	0.13
Taiga Bean-Goose	0.533	370	0.903	0.997	0.781	0.17
Tawny Owl	0.618	3142	0.783	0.991	0.783	0.13
Tawny Pipit	0.488	920	0.518	0.926	0.611	0.23
Temminck's Stint	0.691	454	0.645	0.968	0.529	0.09
Tennessee Warbler	0.599	2151	0.709	0.991	0.733	0.21
Terek Sandpiper	0.463	546	0.444	0.937	0.444	0.09
Thekla's Lark	0.593	1764	0.777	0.993	0.664	0.23
Thick-billed Kingbird	0.67	2197	0.643	0.985	0.662	0.15
Thick-billed Murre	0.441	449	0.927	0.999	0.879	0.25
Thrush Nightingale	0.851	3881	0.834	0.997	0.854	0.28
Townsend's Solitaire	0.563	2610	0.715	0.99	0.69	0.13
Townsend's Warbler	0.488	2402	0.533	0.977	0.522	0.16
Tree Pipit	0.634	3065	0.64	0.989	0.689	0.21
Tree Swallow	0.685	3038	0.863	0.995	0.87	0.17
Tricolored Blackbird	0.599	1671	0.797	0.975	0.726	0.1
Tricolored Heron	0.726	941	0.617	0.993	0.454	0.12
Tristram's Warbler	0.57	304	0.58	0.919	0.467	0.09
Tropical Kingbird	0.49	2464	0.733	0.975	0.763	0.13
Tropical Parula	0.584	3019	0.424	0.969	0.477	0.19
Trumpeter Finch	0.487	190	0.027	0.752	0.005	0.01
Trumpeter Swan	0.679	1800	0.662	0.973	0.763	0.27
Tufted Duck	0.71	788	0.718	0.976	0.584	0.08
Tufted Titmouse	0.477	3124	0.66	0.985	0.623	0.11
Tundra Bean-Goose	0.618	183	0.244	0.99	0.255	0.06
Tundra Swan	0.594	2433	0.765	0.994	0.691	0.07
Twite	0.661	1138	0.985	0.999	0.933	0.06
Upcher's Warbler	0.783	477	0.015	0.857	0.0	0.0
Upland Sandpiper	0.655	1428	0.624	0.979	0.725	0.19
Ural Owl	0.449	1527	0.745	0.993	0.675	0.17
Varied Bunting	0.684	1600	0.606	0.964	0.713	0.19

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Varied Thrush	0.474	2507	0.508	0.977	0.502	0.07
Vaux's Swift	0.455	691	0.697	0.966	0.789	0.08
Veery	0.51	2109	0.759	0.994	0.819	0.22
Verdin	0.488	2610	0.71	0.985	0.706	0.1
Vermilion Flycatcher	0.536	1517	0.492	0.961	0.65	0.15
Vesper Sparrow	0.599	2202	0.595	0.982	0.599	0.23
Violet-green Swallow	0.595	2141	0.747	0.977	0.805	0.19
Virginia Rail	0.519	1982	0.48	0.962	0.472	0.16
Virginia's Warbler	0.534	1841	0.477	0.977	0.534	0.16
Wallcreeper	0.434	99	0.001	0.166	0.0	0.0
Wandering Tattler	0.457	488	0.713	0.952	0.74	0.06
Warbling Vireo	0.621	3518	0.681	0.987	0.725	0.13
Water Pipit	0.4	972	0.669	0.984	0.5	0.05
Water Rail	0.545	2094	0.637	0.967	0.672	0.31
Western Bluebird	0.666	1986	0.689	0.959	0.687	0.04
Western Bonelli's Warbler	0.535	2628	0.659	0.991	0.716	0.16
Western Capercaillie	0.593	1767	0.819	0.993	0.877	0.15
Western Grebe	0.65	1284	0.703	0.973	0.734	0.11
Western Gull	0.68	1353	0.575	0.985	0.47	0.16
Western Kingbird	0.598	1749	0.792	0.993	0.837	0.17
Western Meadowlark	0.533	3211	0.598	0.985	0.616	0.16
Western Olivaceous Warbler	0.637	1479	0.485	0.974	0.488	0.1
Western Orphean Warbler	0.582	2021	0.59	0.985	0.64	0.16
Western Rock Nuthatch	0.648	696	0.713	0.978	0.647	0.1
Western Sandpiper	0.849	923	0.482	0.986	0.368	0.08
Western Screech-Owl	0.404	2400	0.885	0.997	0.881	0.1
Western Swamphen	0.597	539	0.669	0.961	0.637	0.1
Western Tanager	0.571	3214	0.599	0.981	0.679	0.18
Western Wood-Pewee	0.515	3421	0.572	0.964	0.599	0.11
Western Yellow Wagtail	0.463	1886	0.762	0.99	0.718	0.15
Whimbrel	0.556	1961	0.655	0.974	0.669	0.11
Whinchat	0.585	3269	0.694	0.987	0.637	0.14
Whiskered Screech-Owl	0.391	1714	0.797	0.991	0.815	0.14
Whiskered Tern	0.48	1190	0.695	0.972	0.678	0.13
White Ibis	0.591	1190	0.69	0.987	0.552	0.09
White Stork	0.578	458	0.35	0.74	0.428	0.16
White Wagtail	0.472	2037	0.668	0.969	0.713	0.18
White-backed Woodpecker	0.534	2127	0.408	0.959	0.421	0.22
White-breasted Nuthatch	0.533	3069	0.743	0.984	0.779	0.19
White-crowned Sparrow	0.519	3059	0.633	0.975	0.644	0.18
White-crowned Wheatear	0.616	308	0.003	0.585	0.0	0.0

SPECIES	S2N	TS	AP	AUC	F0.5	CT
White-eyed Vireo	0.626	2902	0.687	0.988	0.701	0.13
White-faced Ibis	0.682	956	0.555	0.953	0.633	0.13
White-headed Woodpecker	0.77	1910	0.533	0.993	0.593	0.16
White-rumped Sandpiper	0.55	754	0.538	0.971	0.446	0.12
White-spectacled Bulbul	0.521	575	0.595	0.975	0.494	0.11
White-tailed Eagle	0.668	760	0.75	0.975	0.796	0.12
White-tailed Kite	0.597	1126	0.583	0.949	0.603	0.13
White-tailed Lapwing	0.69	222	0.483	0.857	0.558	0.16
White-tailed Ptarmigan	0.612	1117	0.621	0.962	0.677	0.05
White-throated Dipper	0.312	1016	0.667	0.979	0.652	0.12
White-throated Kingfisher	0.577	1840	0.759	0.98	0.697	0.07
White-throated Robin	0.62	213	0.003	0.543	0.0	0.0
White-throated Sparrow	0.425	3234	0.693	0.983	0.684	0.19
White-throated Swift	0.567	1119	0.539	0.909	0.655	0.09
White-tipped Dove	0.477	3568	0.548	0.969	0.563	0.2
White-winged Crossbill	0.547	2674	0.741	0.988	0.746	0.14
White-winged Dove	0.475	1777	0.589	0.982	0.608	0.17
White-winged Lark	0.735	305	0.782	0.998	0.707	0.19
White-winged Snowfinch	0.633	438	0.136	0.615	0.094	0.02
White-winged Tern	0.723	806	0.901	0.978	0.852	0.05
Whooper Swan	0.735	2397	0.901	0.993	0.843	0.08
Wild Turkey	0.558	1825	0.377	0.932	0.395	0.1
Willet	0.741	2254	0.741	0.992	0.765	0.24
Williamson's Sapsucker	0.672	2318	0.585	0.987	0.495	0.12
Willow Flycatcher	0.508	2782	0.642	0.976	0.673	0.16
Willow Ptarmigan	0.63	1861	0.575	0.978	0.597	0.15
Willow Tit	0.66	2651	0.665	0.985	0.673	0.22
Willow Warbler	0.638	3544	0.594	0.983	0.596	0.17
Wilson's Plover	0.543	1540	0.614	0.988	0.671	0.1
Wilson's Snipe	0.655	2473	0.696	0.983	0.719	0.21
Wilson's Warbler	0.523	2638	0.529	0.974	0.587	0.16
Winter Wren	0.486	2375	0.778	0.992	0.791	0.16
Wood Duck	0.582	2046	0.648	0.971	0.607	0.07
Wood Lark	0.517	3102	0.885	0.998	0.854	0.14
Wood Sandpiper	0.525	1560	0.735	0.967	0.709	0.1
Wood Stork	0.471	1310	0.39	0.883	0.401	0.07
Wood Thrush	0.626	3754	0.862	0.997	0.878	0.18
Wood Warbler	0.641	3531	0.74	0.995	0.816	0.24
Woodchat Shrike	0.659	1616	0.546	0.983	0.399	0.12
Woodhouse's Scrub-Jay	0.703	1390	0.367	0.851	0.514	0.22
Worm-eating Warbler	0.579	2462	0.584	0.993	0.549	0.17

SPECIES	S2N	TS	AP	AUC	F0.5	CT
Wrentit	0.535	1118	0.793	0.985	0.778	0.1
Yellow Rail	0.595	1370	0.948	0.999	0.915	0.03
Yellow Warbler	0.487	1863	0.582	0.979	0.569	0.18
Yellow-bellied Flycatcher	0.456	2604	0.745	0.993	0.767	0.09
Yellow-bellied Sapsucker	0.618	2286	0.403	0.959	0.396	0.25
Yellow-billed Chough	0.67	679	0.926	0.999	0.925	0.05
Yellow-billed Cuckoo	0.472	1802	0.524	0.968	0.591	0.19
Yellow-billed Magpie	0.621	973	0.708	0.956	0.633	0.14
Yellow-breasted Chat	0.543	3741	0.559	0.981	0.455	0.14
Yellow-browed Warbler	0.447	2308	0.891	0.992	0.901	0.06
Yellow-crowned Night-Heron	0.468	249	0.045	0.907	0.002	0.01
Yellow-eyed Junco	0.64	2542	0.494	0.976	0.391	0.05
Yellow-headed Blackbird	0.774	2992	0.647	0.987	0.665	0.21
Yellow-legged Gull	0.763	1638	0.77	0.994	0.638	0.14
Yellow-rumped Warbler	0.458	2312	0.557	0.972	0.541	0.11
Yellow-throated Vireo	0.53	3063	0.787	0.994	0.767	0.12
Yellow-throated Warbler	0.458	2002	0.566	0.971	0.567	0.09
Yellowhammer	0.551	3042	0.74	0.992	0.789	0.15
Zitting Cisticola	0.456	2432	0.841	0.988	0.861	0.14
Zone-tailed Hawk	0.633	768	0.402	0.916	0.483	0.2

Dissertationen der Medieninformatik

1. Kürsten, Jens (2012)
   A Generic Approach to Component-Level Evaluation in Information Retrieval
   ISBN 978-3-941003-68-2
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-96344

2. Berger, Arne (2014)
   Prototypen im Interaktionsdesign: Klassifizierung der Dimensionen von Entwurfsartefakten zur Optimierung der Kooperation von Design und Informatik
   ISBN 978-3-944640-00-6
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-126344

3. Ritter, Marc (2015)
   Optimierung von Algorithmen zur Videoanalyse: Ein Analyseframework für die Anforderungen lokaler Fernsehsender
   ISBN 978-3-944640-09-9
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-133517

4. Kurze, Albrecht (2016)
   Modellierung des QoS-QoE-Zusammenhangs für mobile Dienste und empirische Bestimmung in einem Netzemulations-Testbed
   ISBN 978-3-944640-60-0
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-195066

5. Wilhelm-Stein, Thomas (2016)
   Information Retrieval in der Lehre: Unterstützung des Erwerbs von Praxiswissen zu Information Retrieval Komponenten mittels realer Experimente und Spielemechaniken
   ISBN 978-3-944640-82-2
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-199778

6. Schneider, Anke (2017)
   Farbeinflussfaktoren zur emotionalen Bildwirkung und ihre Bedeutung für das Retrieval von Tourismusbildern
   ISBN 978-3-96100-002-9
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-209553

7. Rickert, Markus (2017)
   Inhaltsbasierte Analyse und Segmentierung narrativer, audiovisueller Medien
   ISBN 978-3-96100-029-6
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-226724

8. Müller, Stefanie (2018)
   Systematisierung und Identifizierung von Störquellen und Störerscheinungen in zeithistorischen Videodokumenten am Beispiel digitalisierter Videobestände sächsischer Lokalfernsehsender
   ISBN 978-3-96100-052-4
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-214115

9. Herms, Robert (2019)
   Effective Speech Features for Cognitive Load Assessment: Classification and Regression
   ISBN 978-3-96100-087-6
   https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-333464

10. Kahl, Stefan (2020)
    Identifying Birds by Sound: Large-scale Acoustic Event Recognition for Avian Activity Monitoring
    ISBN 978-3-96100-110-1
    http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa2-369869