BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

DOCTORAL THESIS

Brno, 2019 Ing. Pavol Harár

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

AUDIO CLASSIFICATION WITH DEEP LEARNING ON LIMITED DATA SETS
KLASIFIKACE AUDIA HLUBOKÝM UČENÍM S LIMITOVANÝMI ZDROJI DAT

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE
Ing. Pavol Harár

SUPERVISOR
ŠKOLITEL
Ing. Jiří Mekyska, Ph.D.

BRNO 2019

ABSTRACT

Standard procedures of dysphonia diagnosis by a clinical speech therapist have their downsides, mainly because the process is very subjective. Recently, an automatic objective analysis of a speaker's condition gained in popularity. Researchers successfully based their methods on various machine learning algorithms and handcrafted features. These methods, unfortunately, are not directly scalable to other voice disorders and the process of feature engineering is laborious and thus financially and talent expensive. Based on the previous successes, a deep learning approach might help to ease the problems with scalability and generalization, but an obstacle is a limited amount of training data. This is a common denominator in almost all systems for automated medical data analysis. The main aim of this work is to research new approaches to deep-learning-based predictive modeling using limited audio data sets, focusing especially on voice pathology assessment. This work is the first to experiment with deep learning in this field and on so far the largest combined database of dysphonic voices, which was created in this work. It provides a thorough examination of publicly available data sources and identifies their limitations. It describes the design of novel time-frequency representations based on Gabor transform and it presents a new class of loss functions, that yield target representations beneficial for learning. In numerical experiments, it demonstrates improvements in the performance of convolutional neural networks trained on limited audio data sets using the augmented target loss function and the newly proposed time-frequency representations, namely Gabor and Mel scattering.

KEYWORDS

deep learning, voice pathologies, Gabor scattering, limited data, audio

ABSTRAKT

Standardní postupy diagnózy dysfonie klinickým logopedem mají své nevýhody, především tu, že je tento proces velmi subjektivní. Nicméně v poslední době získala popularitu automatická objektivní analýza stavu mluvčího. Vědci úspěšně založili své metody na různých algoritmech strojového učení a ručně vytvořených příznacích. Tyto metody nejsou bohužel přímo škálovatelné na jiné poruchy hlasu, samotný proces tvorby příznaků je pracný a také náročný z hlediska financí a talentu. Na základě předchozích úspěchů může přístup založený na hlubokém učení pomoci překlenout některé problémy se škálovatelností a generalizací, nicméně překážkou je omezené množství trénovacích dat. Jedná se o společný jmenovatel téměř ve všech systémech pro automatizovanou analýzu medicínských dat. Hlavním cílem této práce je výzkum nových přístupů prediktivního modelování založeného na hlubokém učení využívající omezené sady zvukových dat, se zaměřením zejména na hodnocení patologických hlasů. Tato práce je první, která experimentuje s hlubokým učením v této oblasti, a to na dosud největší kombinované databázi dysfonických hlasů, která byla v rámci této práce vytvořena. Předkládá důkladný průzkum veřejně dostupných zdrojů dat a identifikuje jejich limitace. Popisuje návrh nových časově-frekvenčních reprezentací založených na Gaborově transformaci a představuje novou třídu chybových funkcí, které přinášejí reprezentace výstupů prospěšné pro učení. V numerických experimentech demonstruje zlepšení výkonu konvolučních neuronových sítí trénovaných na omezených zvukových datových sadách pomocí tzv. "augmented target loss function" a navržených časově-frekvenčních reprezentací "Gabor" a "Mel scattering".

KLÍČOVÁ SLOVA

hluboké učení, patologie hlasu, Gabor scattering, limitovaná data, zvuk

HARÁR, Pavol. *Audio classification with deep learning on limited data sets*. Brno, 2019, 159 p. Doctoral thesis. Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications. Advised by Ing. Jiří Mekyska, Ph.D.

DECLARATION

I declare that I have written the Doctoral Thesis titled "Audio classification with deep learning on limited data sets" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Doctoral Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

DEDICATION

I dedicate this work to the three most important women in my life, my dearest mother Katka, who gave me life, my beloved wife Monika, who gave me the reason to live it and Ivanka, who showed me how. Thank you.

ACKNOWLEDGEMENT

I would like to extend my deepest gratitude to my father for his love and support, to my son Hugo for greeting me joyfully when I return home from work and to all my family for putting up with my nerdy nature.

I am extremely grateful to my supervisor Jiří Mekyska for his guidance and help throughout my studies, for his friendship, and for motivating me when I needed it most. It would not have been possible to do all this work without his friendly and expert advice. I am also very grateful to my co-supervisor Monika Dörfler for help and expertise in applied math and harmonic analysis and for always being friendly and kind. Special thanks to Zoltán Galáž for his relentless support, incredible fun during our studies and also for motivating me to even start a PhD. I would like to thank prof. Zdeněk Smékal and prof. Jiří Mišurec for being honest and direct with me in all situations and for the generous financial support. Many thanks to Jesus B. Alonso-Hernandez who helped me make the first steps with my research. Great thanks go to Roswitha Bammer, Anna Breger and all other co-authors for their dedication and hard work. I had a great pleasure to directly work with outstanding researchers, to whom I would like to express my sincere thanks, namely Dennis Elbrächter, Julius Berner, prof. Philipp Grohs and Jan Schlüter. Thanks should also go to Pavel Rajmic for sharing his network of collaborators and suggesting possible research institutions for an internship. I also wish to thank Lukáš Vrábel for his quick support regarding state-of-the-art techniques. I need to thank Radim Burget for introducing me to the concept of neural networks and to Václav Uher, Lukáš Povoda and Jan Mašek for answering a plethora of questions when I was still new to Linux and programming. Very special thanks go to Vojtěch Zvončák for shared night shifts fixing broken GPU servers. I can not thank enough to Jana Nosková, Jitka Šichová, Petr Číka, and prof. Jaroslav Koton for patient guidance through the bureaucratic system as academia can be. Especially, I am thankful to Jordy Timo van Velthoven for suggesting to write a cumulative dissertation which saved my sanity. I am also very happy that I had the pleasure to meet other colleagues, with whom I did not work directly, but who also helped and inspired me during our lunch or coffee discussions or during workshops and conferences. The list would be too long to write out, and desperately incomplete, but in general this thanks goes to all at the fifth and seventh floors of the T12 building of the Department of Telecommunications of the Brno University of Technology, to all at the fifth floor of the Faculty of Mathematics of the University of Vienna, and to the teams of BDALab, NuHAG, IDeTIC, ARI, and OFAI research groups and institutions. I would like to gratefully acknowledge the effort of the editors and reviewers of our publications and my thesis, as well as the open-source software developers and maintainers, for donating their time to science. Finally, I cannot thank enough to Lukáš Ševčík for inspiring me, back in 2010, to switch fields completely and to start anew at a tech university.

Brno ....

author's signature

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

ACKNOWLEDGEMENT

Research described in this Doctoral Thesis has been implemented in the laboratories supported by the SIX project; reg. no. CZ.1.05/2.1.00/03.0072, operational program Výzkum a vývoj pro inovace.

Brno ....

author's signature

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

Contents

Preamble

Publications

Appendix

Preamble

1 Introduction

> *"The potential benefits are huge; everything that civilisation has to offer is a product of human intelligence; we cannot predict what we might achieve when this intelligence is magnified by the tools that AI may provide, but the eradication of war, disease, and poverty would be high on anyone's list. Success in creating AI would be the biggest event in human history. Unfortunately, it might also be the last, unless we learn how to avoid the risks."*
>
> *Stephen Hawking, Stuart Russell, Max Tegmark & Frank Wilczek, 2014*

As a society, we should not try to stand in the way of technological advances. I believe, there is no way of stopping what we have already been once able to imagine. However, we also should not completely surrender to anything that just becomes convenient. Instead, we should use the new technology with caution, without locking ourselves and other species on the planet unwarily out of other options. While keeping such an open, but a watchful mindset, we should focus on finding ways to take advantage of our innovations to ease the suffering where we see possible.

During my doctoral studies, I have tried to focus my research work in that manner – to embrace new ideas and further their development towards a wholesome application. It culminates in this thesis in the form of a cumulative dissertation, which comprises a certain portion of the published works produced by my coauthors and me. It gives a brief introduction to each of the relevant topics and describes the genesis of the presented ideas. Furthermore, it provides a story line contextually linking and summarizing the individual papers. Finally, the work as a whole is discussed and concluded.

From a methodological point of view, the central idea of this scientific endeavor is an exploration of the learning capabilities of deep neural networks trained with audio data, particularly in sequence classification. From the application perspective, we explore and address the domain-specific challenges which emerge in the analysis of pathological voices.

This document is structured into three main parts, namely the Preamble, Publications, and Appendix. Those lines describing my ideas and points of view are written in the singular form of the first person, the rest, summarizing the joint effort of my coauthors and me, is written in the plural form of the first person.

In the following sections, I am introducing the relevant topics in a non-technical, colloquial way. The main aim is to provide a potential reader without sufficient background, an idea, where these topics originate. A more experienced reader with an understanding of deep learning, audio signal processing, and medical data analysis, whom this thesis assumes, may consider skipping the Introduction and continuing directly to Summary of the Publications. If this is not the case, I recommend further reading, at the end of each section.

1.1 Deep Learning

Artificial intelligence (AI) is a field of study in computer science (CS). Its definition is unfortunately not clear or straightforward, as it took Stuart Russell and Peter Norvig exactly 31 full pages in their book *Artificial Intelligence: A Modern Approach (2016)* [44], to introduce, define and summarize the concept along with its philosophical, mathematical and other cultural foundations. Colloquially, it refers to the ability of a machine to solve a task by imitating intelligent human behavior. It is often confused with artificial general intelligence (AGI), which inspired a multitude of science fiction authors because of the fascinating idea of a computer that would be equally intelligent to humans in every aspect. The term "Artificial intelligence" was coined by John McCarthy in 1955 [8], just about a year after the sad death of Alan Turing, the father of CS [4].

> *"It would be useful if computers could learn from experience and thus automatically improve the efficiency of their own programs during execution."*
>
> *Donald Michie, 1968*

Machine learning (ML), a subfield of AI, is also a term which refers to a particular set of algorithms, that enable the computers to learn from historical data i.e. experience, without being explicitly programmed. This is a paraphrased quote often attributed to Arthur Samuel, who is also considered having coined the term "Machine learning" back in 1959 [46]. According to Russell and Norvig, ML is a capability of a computer to adapt to new circumstances and to detect and extrapolate patterns [44]. A ML algorithm builds a mathematical model based on the set of training data, which provides an approximation of an unknown optimal solution of the task as measured by a performance metric.

"We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data."

Yann LeCun, Yoshua Bengio & Geoffrey Hinton, 2015

Deep learning (DL) is a subfield of ML concerned with artificial neural networks (ANN). ANNs are computation systems of interconnected artificial neurons, which very loosely model the biological neurons. ANNs have been developed since 1943, when McCulloch and Pitts, inspired by the study of the human brain modeled an electrical circuit of a simple neural network [35], and since Rosenblatt described a mathematical model of Perceptron in 1958 [43]. Nowadays, ANNs are usually described as directed graphs of nodes connected with edges and organized into layers.

The term "Deep learning" was introduced by Rina Dechter in 1986 [47]. The word "deep" refers to a subset of ANNs with a number of hidden layers (number of layers excluding input and output layer) bigger than one. Depending on how the nodes are linked, i.e. the topology, the deep neural network (DNN) is either feedforward or recurrent. Edges represent weights, which parameterize the model and are adjusted during the training of the network.

DNNs were not popular at first but became widely used with the increased availability of data and computing power. In the past years, they dramatically improved the state of the art in areas such as speech recognition, visual object recognition, robotics, bioinformatics, online advertising, search engines, and medical applications [30, 31], to name a few. In this work, we are mainly concerned with architectures composed of one or more of the following components: standard fully connected feedforward layers, convolutional layers as introduced in deep convolutional neural networks (CNN) and recurrent long short-term memory layers (LSTM) [29, 25, 45].

For further reading, please refer to the following books, which go into a great detail in each topic: *Artificial Intelligence: A Modern Approach (Russell & Norvig, 2016)* [44], *Pattern recognition and machine learning (Bishop, 2011)* [5], *Introduction to Machine Learning (Alpaydin, 2014)* [2], *Deep Learning (Goodfellow, Bengio & Courville, 2016)* [17]. For a quicker overview of DL, refer to the works of LeCun, Bengio & Hinton (2015) [30], Schmidhuber (2016) [47], Liu et al. (2017) [32] and Pouyanfar et al. (2018) [41].

1.2 Digital Audio Signal Processing

To process sound information with neural networks, it is necessary to transform the continuous acoustic physical phenomenon into its discrete, digital, computer-understandable representation, i.e. audio data. The field concerned with recording real-world signals like voice, music, etc., their further conversion and processing is called digital signal processing (DSP). In this work, we will be mainly interested in decisions of sampling rate and time-frequency representations of the sound, as well as psychoacoustics and we will study their impact on learning.

For a comprehensive introduction into these topics, please refer to the following books: *Discrete-Time Signal Processing (Oppenheim & Schafer, 2014)* [39], *Digital Audio Signal Processing (Zölzer, 2008)* [48], *Understanding Digital Signal Processing (Lyons, 2004)* [33], *Foundations of Time-Frequency Analysis (Gröchenig, 2001)* [18]. For a more concise merger introducing DL from the perspective of audio signal processing, refer to the paper by Purwins et al. (2019) [42].

1.3 Automatic Analysis of Medical Audio Data

According to an extensive survey in medical image analysis by Litjens et al. (2017) [31], medical images have been automatically analyzed as soon as it was possible to capture and load them into a computer. In the case of audio, researchers were first interested in using extralinguistic information to identify speakers, their age or gender. For speech emotion recognition, they have used paralinguistic information, and in the case of accent, dialect or speech recognition, the linguistic dimension has been studied. Just in the past years, the analysis of the speaker's condition gained in popularity, as Gómez-García, Moro-Velázquez & Godino-Llorente (2019) [16] explain in another great survey on automatic voice condition analysis (AVCA) systems. AVCA aims for an objective and automatic quantification of the degree to which a patient is impaired by a voice disorder. One of the main advantages of such analysis based on audio data is its relatively low cost, non-invasive nature and a possibility for continuous monitoring and in-cloud processing [36].

The fact that sparked my interest in this research direction was a link between hypokinetic dysarthria (HD) and Parkinson's disease (PD). HD is a motor speech disorder manifested in articulation, phonation, prosody, respiration, and faciokinesis, that occurs in up to 90 % of PD patients and is also considered one of the early markers of PD. For more information about HD and other disorders in PD, please refer to a thorough survey paper by Brabenec et al. (2017) [6]. Unfortunately, nowadays, it is still not possible to cure PD, but an early diagnosis can significantly improve patient's quality of life thanks to already available medication.

The standard procedure of HD diagnosis is carried out by a clinical speech therapist. Speech and voice of a patient are usually assessed using specific scales and questionnaires such as Frenchay dysarthria assessment [12] or 3F test [27]. This procedure still has its downsides though, mainly because the evaluations are very subjective. The human ear, even of a trained clinician, is not sensitive enough to capture slight changes in the patient's voice or speech, it is, therefore, hard to compare successive assessments for progression tracking, even from the same clinician [36].

Researchers thus started to work on automatic objective methods of HD analysis and proposed a variety of parameterization methods, to extract conventional or non-conventional features from the audio recordings of the patients' speech and voice. These were further utilized in predictive modeling using a variety of machine learning techniques [6] to infer an automatic evaluation of the patient's data. Successes of these methods are undeniable and encouraging, with strong advantages for clinicians who can use these methods as a supportive tool for their decisions. The main pros are objectivity and relatively good interpretability [11], which is important in this setting. Unfortunately, the approach is not directly scalable to voice disorders other than HD. The process of feature engineering is laborious and requires researchers with expertise in signal processing and machine learning as well as deep knowledge of the particular disorder and its underlying pathophysiological mechanisms. A model trained for one disorder will highly unlikely produce satisfactory predictions on data of another disorder. Even for the same disorder, the model's performance can differ greatly depending on the data acquisition conditions or labeling framework. For a more comprehensive list of factors affecting AVCA systems, refer to the work of Gómez-García et al. (2019) [16].

A DL approach might help to alleviate the problems with scalability and generalization, but and obstacle, as will be pointed out later, is a limited amount of available data, which is insufficient for today's DL models to fulfill their promises. The lack of data is a common denominator of almost all systems for automated medical data analysis.

> *"Deep Learning is getting really good on Big Data [...]. But Small Data is important too. [...] Hope more researchers work on Small Data – ML needs more innovations there."*
>
> *Andrew Ng, 2018*

Please, refer to the following articles for further information: *A Guide to Deep Learning in Healthcare (Esteva et al. 2019)* [13] provides a short, but comprehensive introduction to this topic, *A Survey on Deep Learning in Medical Image*

Analysis (Litjens et al. 2017) [31] provides an extensive survey regarding images, which is very relevant to this topic due to image-like properties of time-frequency representations of audio signals.

1.4 Objectives

From the perspective of the superordinate analysis of this dissertation, here I retrospectively delineate the main objective of this work, which is to **research new approaches to DL based predictive modeling using limited audio data sets, with a special focus on voice pathology assessment**. This main aim along with its sub aims will be later discussed in section Concluding Discussion. More specifically this dissertation aims to:

Aim 1: Explore the specifics of medical audio data analysis with DL
This constitutes conducting first experiments directly with the raw waveform in a search for an end to end system of voice pathology detection, which would map raw waveforms to the corresponding targets. Such experiments should also show the specific nature of the data and how to handle them with DL while determining the caveats.

Aim 2: Identify prospective DNN architectures w.r.t. AVCA systems
We plan to test popular DNN building blocks used in CV and in time-series analysis, namely CNN and LSTM, expecting automatic feature extraction.

Aim 3: Review available data sources and their limitations
More specifically to review their previous uses, identify which speech tasks they comprise, what is the distribution of healthy vs. dysphonic samples, what is the distribution of pathology types recorded and to propose an approach of combining the databases.

Aim 4: Clarify which input and target representations are useful
Specifically, to train models using raw waveforms and standard time-frequency representations, and compare the performances with handcrafted speech features. Moreover to identify, which other input modalities, such as gender, age, a grade of dysphonia, etc. affect the modeling capabilities and to suggest possibilities of redefining the task by changing the targets.

Aim 5: Propose countermeasures to high data demand
More precisely, to research and propose novel input and target data representations, which would benefit training on limited data sets.

2 Summary of the Publications

The main body of this work consists of five selected publications done during my doctoral studies. This section gives a short overview of their order, how the articles are contextually linked and how each of the preceding work and other events, like research visits, influenced the research direction and topic of the whole thesis. This timeline is presented in Table 2.1. The Publications are presented in versions of accepted or submitted manuscripts, their templates are unified, but contents are unchanged, apart from the numbering of tables, figures, equations and theorems, which may not fully reflect the official version.

Before I started to work on these articles, I did some prior work, where I was exploring the idea of DL and its application to time-series and audio data. In a paper entitled *Speech Emotion Recognition with Deep Learning (Harar, Burget & Duta, 2017)* [22], we have successfully used a CNN for automatic speech feature extraction and classification into one of three classes, i.e. emotional states – angry, neutral, sad.

After I was exposed to work and ideas of Mekyska and Galaz at the *Brain Diseases Analysis Laboratory*, I started to work on the utilization of DL in AVCA systems to avoid the "manual" feature engineering. Shortly after, I made a research visit to the University of Las Palmas de Gran Canaria, where Assoc. Prof. Jesús B. Alonso-Hernández generously provided his experience and further guidance.

Based on this cross-fertilization of ideas, a preliminary study entitled *Voice Pathology Detection using Deep Learning* [20] was published and presented at *International Conference and Workshop on Bioinspired Intelligence (IWOBI)* in July 2017. To the best of our knowledge, this was the first work in the world that studied the use of DL to solve this type of a problem. The objective of this study was to clarify, whether the use of DNN based on a combination of CNN and LSTM, applied to raw input audio signal, would prove itself worthy of further exploration for voice pathology detection. This work was chosen to be extended for a special issue in the journal *Neural Computing and Applications (IF 4.664, Q2 in AI)* and was once again presented at *Systematic Approaches to Deep Learning Methods for Audio* workshop in Vienna in September 2017.

The extended version with title *Towards Robust Voice Pathology Detection* [24] contains an extensive survey of previously published works, presents experiments conducted on four databases, namely Arabic Voice Pathology Database (AVPD) [37, 38], Massachusetts Eye and Ear Infirmary Voice Disorders Database (MEEI) [34], Príncipe de Asturias Database (PDA) [15] and SVD. Furthermore, it compares performances of ML and DL models trained using raw audio signal, spectral and cepstral time-frequency representations, and conventional handcrafted features. Also

Table 2.1: Timeline

- **Prior work**
  First published experiments with CNNs for audio sequence classification applied to speech emotion recognition.

- **University of Las Palmas de Gran Canaria (IDeTIC)**
  Acquired new data, exchanged ideas, and received guidance in the research of pathological voices from the machine learning perspective from Assoc. Prof. Jesús B. Alonso-Hernández.

- **Voice Pathology Detection Using Deep Learning**
- **Towards Robust Voice Pathology Detection**
  In-depth analysis of the state of the art and available data sets. Identified the main issues and conducted cross-database experiments.

- **University of Vienna (NuHAG)**
  Collaboration and supervision from Dr. Monika Dörfler in applied math and harmonic analysis. Strong focus on the fundamentals of neural networks and audio time-frequency representations.

- **On Orthogonal Projections for Dimension Reduction ...**
  Numerical experiments with augmented target loss function emphasizing important characteristics by beneficial representations of the target space.

- **Gabor Frames and Deep Scattering Networks in Audio ...**
- **Improving Machine Hearing on Limited Data Sets**
  Proposed and developed a software library for Gabor scattering and Mel scattering. Addressed the issue of insufficient amounts of data.

- **Future work**
  Combining the findings and applying them to voice pathology data.

Legend: – Journal article, – Conference paper, – Research visit

includes experiments with DenseNet [26] DNN architecture. It points out the limitations of the available data, the definition of the task and approach and suggests future work to alleviate the summarized problems.

In 2018, I have been awarded a grant for the mobility of researchers and thanks to the previously mentioned workshop in Vienna, I was given the opportunity to continue my research as a part of Numerical Harmonic Analysis Group (NuHAG) at the Faculty of Mathematics of the University of Vienna. This research visit under the supervision of Dr. Monika Dörfler radically changed my view on the problems at hand. I was invited to collaborate on multiple interesting fundamental research topics, for which I have conducted numerical experiments in music information retrieval setting, helped to design and implemented proposed algorithms, and created software libraries. In all the following articles, we have taken advantage of CNNs which were originally proposed for computer vision (CV), in predictive modeling with audio data. The reason is that standard FFT-based signal processing methods allowed exploiting advances in CV in the audio analysis by converting the raw audio waveforms into image-like representations (e.g. spectrograms).

A collaboration with the Department of Ophthalmology of the Medical University of Vienna led to an article accepted in the *Journal of Mathematical Imaging and Vision (IF 1.603, Q1 in CV)* titled *On Orthogonal Projections for Dimension Reduction and Applications in Augmented Target Loss Functions for Learning Problems* [7]. In this article, we studied the use of orthogonal projections on high-dimensional input and target data in learning frameworks and we introduced a general framework of augmented target loss functions (AT). These loss functions integrate additional information via transformations and projections of the target data. In two supervised learning problems, clinical image segmentation and music information classification, the application of our proposed AT increased the accuracy.

From the perspective of time-frequency analysis, in the paper *Gabor Frames and Deep Scattering Networks in Audio Processing* [3], we introduced Gabor scattering, a feature extractor based on Gabor frames and Mallat's scattering transform. Based on the provided theory, we have implemented the Gabor-scattering software library for Python programming language [19]. Furthermore, with numerical experiments, we showed, that the invariances encoded by the Gabor scattering transform lead to higher performance in comparison with just using Gabor transform, especially when few training samples are available.

As a next natural step, we included a human perceptual scale, which led to an extension of the Gabor scattering to a Mel scattering representation. The aforementioned software library was extended to cover both Gabor and Mel scattering. In the paper *Improving Machine Hearing on Limited Data Sets* [21] we investigated

how input and target representations interplay with the amount of training data in a music information retrieval setting. We compared the standard mel-spectrogram inputs with a newly proposed Mel scattering. Furthermore, we investigated the impact of additional target data representations by using the AT which incorporates unused available information. We observed that all proposed methods outperformed the standard mel-spectrogram representation when using a limited data set.

3 Concluding Discussion

To conclude this dissertation as a whole, the following section sums up the conclusions of the publications and is structured in such a way it tries to address the objectives in order of appearance in the section Objectives.

In the frame of **Aim 1** and **Aim 2**, we have hoped for an end to end system of voice pathology detection, which would map raw waveforms to the corresponding targets. The objective of the paper Voice Pathology Detection Using Deep Learning was to carry out a preliminary study which would clarify whether the use of the DNN model, especially combination of convolutional and LSTM layers would prove itself worthy of further exploration in case of voice pathology detection problem using only raw recordings of sustained vowel /a/. The examined method achieved 71.36 % accuracy on validation data and 68.08 % accuracy on testing data. It is important to note, that we did not restrict the classification to a subset of pathologies and we used all 71 present in the database.

We conclude that the main advantage of the DL approach with CNN is the automatic feature extraction, as opposed to the previously proposed methods. It saves a great amount of time and expertise in the area of the problem being solved. We found out, that the main disadvantage is the amount of data needed to train the model. The SVD database used in this experiment is extensive in numbers of persons recorded, but there are not enough samples of healthy persons in comparison with the number of samples of pathological patients. Also, the distribution of individual pathologies is extremely unequal making the voice pathology detection a hard problem.

In search of a robust voice pathology detection system using acoustic (voice) signals, researchers face a variety of problems. One of the major problems in this field of science, as we pointed out before, is the limited amount of data. Nevertheless, one large database from one source would not solve all the issues. A problem is also a limited number of distinct publicly available databases, using which the model could capture the variance of the data acquired in different recording conditions and environments. Following the **Aim 3**, the article Towards Robust Voice Pathology Detection explores publicly available data sources of dysphonic voices, discusses the means of combining them into one bigger database and uncovers their limitations concerning building an automatic assessment system.

The paper concludes, these commonly used databases (AVPD, MEEI, PDA, SVD) are very hard to combine because of various distinctions such as a) the databases are labeled in different languages, b) the databases do not comprise the same set of speech tasks, c) there is a variety of voice pathologies unequally distributed across the databases, etc. For these reasons, up to now, researchers have used only

a subset of the databases for their experiments providing results related to that carefully selected subset of data. However, this approach limits the possibilities of creating a robust voice pathology detector. We have conducted experiments on recordings of sustained phonation of the vowel /a/ produced at a normal pitch from the combination of these 4 different databases, trying to eliminate mentioned limitations. To the best of our knowledge, this is the first work that uses such a "large" set of data to build mathematical models for computerized, objective voice pathology detection.

To make a broader comparison, we researched 3 distinct classifiers within supervised learning and anomaly detection paradigms. Following the **Aim 4**, we have explored the usage of raw waveforms, spectrograms, MFCC, conventional dysphonic features and their combinations as input data. We observed that XGBoost classifier achieved the best results amongst DenseNet and Isolation Forest classifiers. In the article, we also investigated and described stratification and group weighting, to equalize the uneven distribution of gender-age groups, which is important to take into account, because of the different voice and speech properties of patients with different ages and gender.

Even though combining the available databases, we have obtained a relatively large amount of data samples, it still seems not to be enough to train a successful DL model on raw waveforms, and from the observed performances, we conclude that in voice pathology detection scenarios, with this (from AVCA perspective large, but from the DL perspective small) amount of training data, it is better to use inputs with reduced dimensionality in contrary to raw waveform inputs, and/or make use of transfer learning, data augmentation or other means to alleviate the problem with the lack of data. On the other hand, reviewing the performances achieved in scenarios with only MFCC as input data, we conclude that representations, as reduced in dimensionality as MFCC alone, are not reliable enough for robust voice pathology detection, which was also concluded by Ali et al. in [1].

We anticipate, that making the combination of the databases more controlled and coherent to reduce the noise in the database and simplifying the complexity of the target space would boost the performance of the system. Thus we think that recordings of the databases commonly used for automatic voice pathology detection should be consulted with clinicians as a whole, to evaluate the severity of vocal manifestation of the present pathologies based on perceptual evaluation as opposed to plain names of present pathologies. There are standard metrics, which are used to evaluate the quality of voice that can be used for this purpose [9, 10, 14, 28]. The addition of such information to the databases could provide researchers with a unique possibility to build models capable of classification and prediction, emphasizing the severity of the exact vocal-manifestation (increased acoustic tremor, roughness,

breathiness, etc.) of these pathologies.

At the end of the article, we anticipated that deep learning will play its role in robust voice pathology detection on the assumption that more data will be available, or at least reasonable combination of available databases will be made and limitations of these databases will be partially diminished by data augmentation and other countermeasures. Besides, we presume that the use of deep learning methods for novelty detection such as deep autoencoder [40] for modeling the normophonic voice could be an interesting idea for future investigation with a prospect to identify even disordered voices that are sparsely distributed across databases.

The first two publications were focused mainly on the specifics of predictive modeling using DL in voice pathology detection. They were concerned with identifying the prospective DNN architectures and dove deep into the analysis of available data sources. The following three publications all look at the problem of insufficient data, which was repeatedly mentioned in the first two publications, from a different perspective. As defined in the **Aim 5**, their objective is to propose methods of input and target space transformation in such a way, the DNN can learn with fewer data.

In the article On Orthogonal Projections for Dimension Reduction and Applications in Augmented Target Loss Functions for Learning Problems, we introduced a general framework of AT. These loss functions integrate additional information via transformations and projections of the target data. In two supervised learning problems, clinical image segmentation and music information classification, the application of our proposed AT increased the accuracy.

Next, in the article Gabor Frames and Deep Scattering Networks in Audio Processing, we introduced Gabor scattering (GS), a scattering transform based on Gabor frames and we investigated its properties. Thereby, we have been able to mathematically express the invariances introduced by GS within the first two layers. We have experimentally shown that explicit encoding of invariances by using an adequate feature extractor is beneficial when a restricted amount of data is available. It was shown that in the case of a limited data set the application of a GS representation improves the performance in classification tasks in comparison to using Gabor transform (GT). This property can be utilized in restricted settings, e.g. in embedded systems with limited resources or in medical applications, where sufficient data sets are often too expensive or impossible to gather, while the highest possible performance is crucial.

The common choice of a time-frequency representation of audio signals in predictive modeling is mel-spectrogram; hence, as a natural step, we introduced Mel scattering (MS) in Improving Machine Hearing on Limited Data Sets, a new feature extractor which combines the properties of GS with mel-filter averaging. We also investigated the impact of additional information about the target space through

AT on the performance of the trained CNN.

From the newly proposed methods, AT is the least expensive in terms of training time, but on the other hand, yields the smallest improvement in this experimental setup. Nevertheless, it has another advantage: it steers the training towards learning the penalized characteristics. We can conclude that AT provides a more precise measure of the distance between outputs and targets. That's why it can help in scenarios where the training set is not large enough to allow the learning of all characteristics but can be penalized by AT. We suggest using/experimenting with the proposed methods for other data sets if there is not a sufficient amount of data available or/and there exist reasonable transformations in the target space relevant to the task being solved. All proposed methods might be found useful also in scenarios with limited resources for training.

Beyond State of the Art This section concluding four long years of work is not short either, thus this paragraph briefly lists the achievements compactly:

- the first-ever use of deep learning in the field of voice pathology detection
- identification of limitations of deep learning w.r.t. this field
- identification of limitations of existing voice pathology databases
- experiments on the largest combined database of dysphonic voices
- design of new time-frequency representations based on Gabor transform
- improvement in the performance of convolutional neural networks on limited audio data sets using proposed novel time-frequency representations, namely Gabor scattering and Mel scattering, and a new class of loss functions, that yield beneficial target representations

Concurrent and Future Work The timeline in Table 2.1 constitutes only the main thread of my doctoral work, even though more work has been done during this period. Most notable are two collaborations: one with the Department of The Communication Disorders of the Comenius University in Bratislava. It is concerned with consulting available voice pathology databases combined into one, with clinical speech therapists, to evaluate the severity of vocal manifestation of the present pathologies based on perceptual evaluation according to GRBAS scale [9]. And the other, with the Austrian Research Institute for Artificial Intelligence (OFAI), experimenting with novel preprocessing steps for learning algorithms. During this collaboration, an experimental software library Redistributor [23] was developed. The results of these collaborations, unfortunately, did not make it into this work and are going to be worked upon and finalized in the future.

Bibliography

[1] Z. Ali, M. Alsulaiman, G. Muhammad, I. Elamvazuthi, A. Al-nasheri, T. A. Mesallam, M. Farahat, and K. H. Malki. Intra-and inter-database study for arabic, english, and german databases: Do conventional speech features detect voice pathology? *Journal of Voice*, 31(3):386–e1, 2017.

[2] E. Alpaydin. *Introduction to machine learning*. Adaptive Computation and Machine Learning series. MIT press, third edition edition, 2014.

[3] R. Bammer, M. Dörfler, and P. Harar. Gabor frames and deep scattering networks in audio processing. *arXiv preprint*, 2017. `arXiv:1706.08818`.

[4] A. Beavers. Alan turing: Mathematical mechanist. *Cooper, S. Barry; van Leeuwen, Jan. Alan Turing: His Work and Impact. Waltham: Elsevier*, pages 481–485, 2013.

[5] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2011.

[6] L. Brabenec, J. Mekyska, Z. Galaz, and I. Rektorova. Speech disorders in parkinson's disease: early diagnostics and effects of medication and brain stimulation. *Journal of neural transmission*, 124(3):303–334, 2017. `doi:10.1007/s00702-017-1676-0`.

[7] A. Breger, J. I. Orlando, P. Harar, M. Dörfler, S. Klimscha, C. Grechenig, B. S. Gerendas, U. Schmidt-Erfurth, and M. Ehler. On orthogonal projections for dimension reduction and applications in augmented target loss functions for learning problems. *Journal of Mathematical Imaging and Vision*, in press. `arXiv:1901.07598`.

[8] K. Cukier. Ready for robots: How to think about the future of ai. *Foreign Aff.*, 98:192, 2019.

[9] M. S. De Bodt, F. L. Wuyts, P. H. Van de Heyning, and C. Croux. Test-retest study of the grbas scale: influence of experience and professional background on perceptual rating of voice quality. *Journal of Voice*, 11(1):74–80, 1997.

[10] P. H. Dejonckere, P. Bradley, P. Clemente, G. Cornut, L. Crevier-Buchman, G. Friedrich, P. Van De Heyning, M. Remacle, and V. Woisard. A basic protocol for functional assessment of voice pathology, especially for investigating the efficacy of (phonosurgical) treatments and evaluating new assessment techniques. *Eur Arch Otorhinolaryngol.*, 258(2):77–82, Feb. 2001.

[11] D. Doran, S. Schulz, and T. Besold. What does explainable ai really mean? a new conceptualization of perspectives. In *CEUR Workshop Proceedings*, volume 2071, 2018. URL: http://openaccess.city.ac.uk/id/eprint/18660/, arXiv:https://arxiv.org/abs/1710.00794.

[12] P. Enderby. Frenchay dysarthria assessment. *British Journal of Disorders of Communication*, 15(3):165–173, 1980. doi:10.3109/13682828009112541.

[13] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24, 2019. doi:10.1038/s41591-018-0316-z.

[14] B. R. Gerratt, J. Kreiman, N. Antonanzas-Barroso, and G. S. Berke. Comparing internal and external standards in voice quality judgments. *J Speech Hear. Res.*, 36(1):14–20, Feb. 1993.

[15] J. I. Godino-Llorente, P. Gómez-Vilda, F. Cruz-Roldán, M. Blanco-Velasco, and R. Fraile. Pathological likelihood index as a measurement of the degree of voice normality and perceived hoarseness. *Journal of Voice*, 24(6):667–677, 2010. doi:10.1016/j.jvoice.2009.04.003.

[16] J. Gómez-García, L. Moro-Velázquez, and J. I. Godino-Llorente. On the design of automatic voice condition analysis systems. part I: Review of concepts and an insight to the state of the art. *Biomedical Signal Processing and Control*, 51:181–199, 2019. doi:10.1016/j.bspc.2018.12.024.

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[18] K. Gröchenig. Foundations of time-frequency analysis. 2001.

[19] P. Harar. Gabor scattering. https://gitlab.com/paloha/gabor-scattering, 2019.

[20] P. Harar, J. B. Alonso-Hernandez, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice pathology detection using deep learning: a preliminary study. In *2017 international conference and workshop on bioinspired intelligence (IWOBI)*, pages 1–4. IEEE, 2017. arXiv:1907.05905, doi:10.1109/IWOBI.2017.7985525.

[21] P. Harar, R. Bammer, A. Breger, M. Dörfler, and Z. Smekal. Improving machine hearing on limited data sets. In *2019 The 11th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*. IEEE, in press. arXiv:1903.08950.

[22] P. Harar, R. Burget, and M. K. Dutta. Speech emotion recognition with deep learning. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 137–140. IEEE, 2017. doi:10.1109/SPIN.2017.8049931.

[23] P. Harar and D. Elbraechter. Redistributor. https://gitlab.com/paloha/redistributor, 2018.

[24] P. Harar, Z. Galaz, J. B. Alonso-Hernandez, J. Mekyska, R. Burget, and Z. Smekal. Towards robust voice pathology detection. *Neural Computing and Applications*, pages 1–11, 2018. arXiv:1907.06129, doi:10.1007/s00521-018-3464-7.

[25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. URL: http://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html, arXiv:https://arxiv.org/abs/1608.06993.

[27] M. Košťálová, M. Mračková, R. Mareček, D. Beránková, I. Eliášová, E. Janoušová, J. Roubíčková, J. Bednařík, and I. Rektorová. Test 3F dysartrický profil–normativní hodnoty řeči v češtině. *Česká a Slovenská Neurologie a Neurochirurgie*, 76(109):5, 2013. URL: https://is.muni.cz/th/ucvby/8_Kostalova_3F.pdf.

[28] J. Kreiman, B. R. Gerratt, G. B. Kempster, A. Erman, and G. S. Berke. Perceptual evaluation of voice quality: review, tutorial, and a framework for future research. *J Speech Hear. Res.*, 36(1):21–40, Feb. 1993.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[30] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. doi:10.1038/nature14539.

[31] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. doi:10.1016/j.media.2017.07.005.

[32] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017. doi:10.1016/j.neucom.2016.12.038.

[33] R. G. Lyons. *Understanding digital signal processing*. Pearson Education India, 2004.

[34] Massachusetts Eye and Ear Infirmary. Voice disorders database, version. 1.03 (cd-rom). *Lincoln Park, NJ: Kay Elemetrics Corporation*, 1994.

[35] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. doi:10.1007/BF02478259.

[36] Mekyska, J. Akustická analýza hypokinetické dysartrie u pacientů s Parkinsonovou nemocí: od základů až po integraci v mHealth systémech. XII. konference - neurogenní poruchy komunikace dospělých, Brno, 5 2017. FN Brno.

[37] T. A. Mesallam, M. Farahat, K. H. Malki, M. Alsulaiman, Z. Ali, A. Al-nasheri, and G. Muhammad. Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms. *Journal of healthcare engineering*, 2017. doi:10.1155/2017/8783751.

[38] G. Muhammad, M. Alsulaiman, Z. Ali, T. A. Mesallam, M. Farahat, K. H. Malki, A. Al-nasheri, and M. A. Bencherif. Voice pathology detection using interlaced derivative pattern on glottal source excitation. *Biomedical Signal Processing and Control*, 31:156–164, 2017. doi:10.1016/j.bspc.2016.08.002.

[39] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing*. Pearson Education Limited, 2014.

[40] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[41] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):92, 2018. doi:10.1145/3234150.

[42] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019. doi:10.1109/JSTSP.2019.2908700.

[43] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. doi:10.1037/h0042519.

[44] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[45] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015. doi:10.1109/ICASSP.2015.7178838.

[46] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959. doi:10.1147/rd.33.0210.

[47] J. Schmidhuber. Deep learning. *Encyclopedia of Machine Learning and Data Mining*, pages 1–11, 2016. doi:10.1007/978-1-4899-7502-7_909-1.

[48] U. Zölzer. *Digital audio signal processing*, volume 9. Wiley Online Library, 2008.

Publications

I Voice Pathology Detection Using Deep Learning: a Preliminary Study

Outline

Bibliographic Information

P. Harar, J. B. Alonso-Hernandez, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice Pathology Detection Using Deep Learning: a Preliminary Study. In *2017 international conference and workshop on bioinspired intelligence (IWOBI)*, pages 1–4. IEEE, 2017. arXiv:1907.05905, doi:10.1109/IWOBI.2017.7985525.

Author's Contribution

The author surveyed related works, designed and performed the analysis, and wrote a significant part of the manuscript. He was also working on the finalization of the whole manuscript, i.e. reviewing, copyediting, etc.

Copyright Notice

Abstract

This paper describes a preliminary investigation of Voice Pathology Detection using Deep Neural Networks (DNN). We used voice recordings of sustained vowel /a/ produced at normal pitch from German corpus Saarbruecken Voice Database (SVD). This corpus contains voice recordings and electroglottograph signals of more than 2 000 speakers. The idea behind this experiment is the use of convolutional layers in combination with recurrent Long-Short-Term-Memory (LSTM) layers on raw audio signal. Each recording was split into 64 ms Hamming windowed segments with 30 ms overlap. Our trained model achieved 71.36 % accuracy with 65.04 % sensitivity and 77.67 % specificity on 206 validation files and 68.08 % accuracy with 66.75 % sensitivity and 77.89 % specificity on 874 testing files. This is a promising result in favor of this approach because it is comparable to similar previously published experiment that used different methodology. Further investigation is needed to achieve the state-of-the-art results.

Acknowledgment

This work was supported by the grant of the Czech Ministry of Health 16-30805A (Effects of non-invasive brain stimulation on hypokinetic dysarthria, micrographia, and brain plasticity in patients with Parkinsons disease) and the following projects: SIX (CZ.1.05/2.1.00/03.0072), and LOl401. For the research, infrastructure of the SIX Center was used.

I.1 Introduction

According to [14] the automatic detection of vocal fold pathologies is a task of assigning normophonic or dysphonic labels to a given phonation produced by a specific speaker. This objective is an interest to the researchers of speech or voice community, as well as the respective medical community. This is due to its non-invasive nature, free from subjective biasness, and relatively low cost. So far, many researchers aimed to detect voice pathology by analyzing the voice with the emphasis to develop features that can effectively distinguish between normal and pathological voices [16].

On the contrary, in this paper we investigate a way to skip the phase of developing the features. Instead, we aim to create an end-to-end deep neural network model capable of voice pathology assessment using raw audio signal. To achieve this goal, we used voice recordings from Saarbruecken Voice Database (SVD) [21] that contains the samples of healthy persons and patients with one up to 71 different pathologies.

Nowadays, thanks to huge increases in computational power and data amounts, the Deep Learning (DL) models delivered the state-of-the-art results in many domains including Speech processing. Using this approach to tackle the voice pathology detection problem we are allowed to use complex multi-layer model architectures. We expect the convolutional layers [12] to learn to detect various patterns that could help us to differentiate between healthy and pathological voice. Long-Short-Term-Mermoy layers [25] should then transform the time distributed abstract feature vectors outputted from convolution stacks into understandable representation for fully connected dense layers, which should do the final classification.

The rest of this paper is organized as follows. Section I.2 introduces the related works in this area of expertise. In Section I.3, data and methodology of the experiment are be discussed. The results are presented in Section I.4. Conclusions are drawn in Section I.5.

I.2 Related Work

There is already a great number of related works in this area of expertise [16, 1, 13, 18, 19, 2, 10, 6, 3, 15, 8].

Detailed information about papers published on SVD can be found in Table I.1. In summary, the authors that used SVD extracted various features from the voice recordings prior to pathology detection. The features were usually extracted from time, frequency and cepstral domains and contained mel-frequency cepstral coefficients (MFCC), energy, entropy, short-term cepstral parameters, harmonics-to-noise ratio, normalized noise energy, glottal-to-noise excitation ratio, multidimensional

Table I.1: Overview of related works

Article	Feature set	Employed classifier	Accuracy	Notes
[8]	28 parameters extracted from time, frequency and cepstral domain	KM, RF	100.00 %	Used combination of vowels /a/, /i/, /u/ Females and Males separately
[15]	energy, entropy, contrast, homogeneity	GMM	99.98 %	Used combination of voice and EGG signals
[3]	MDVP parameters	SVM	99.68 %	Used subset containing 4 of 71 pathologies
[6]	MFCC	GMM	99.00 %	Used combination of vowels /a/, /i/, /u/
[10]	MPEG-7 low-level audio and IDP	SVM, ELM, GMM	95.00 %	Used mix of MEEI and SVD data
[16]	IDP	SVM	93.20 %	Used subset containing 3 of 71 pathologies
[2]	Maximum peak and lag	SVM	90.98 %	Used subset containing 4 of 71 pathologies
[19]	MFCC first and second derivatives	ANN	87.82 %	Used subset containing 4 of 71 pathologies
[18]	short-term cepstral parameters	SVM	86.44 %	Used subset containing 4 of 71 pathologies
[13]	MFCC, harmonics-to-noise ratio, normalized noise energy, glottal-to-noise excitation ratio	GMM	79.40 %	Used combination of vowels /a/, /i/, /u/
[1]	Peak value and lag for every frequency band	GMM, SVM	72.00 %	Used 200 samples of vowel /a/ at high pitch

voice program parameters (MDVP), etc. After the feature extraction, multiple classifiers have been used. Most authors relied on Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) but K-means clustering (KM), Random forests (RF), Extreme Learning Machines (ELM) and Artificial Neural Networks (ANN) were also utilized in several papers. To our best knowledge, this is the first paper that presents the voice pathology detection using DNN.

The results vary greatly between the published papers mainly due to differences between sets of data that were used for the experiment. Martínez et al. in [13] reported 72 % accuracy using 200 recordings of sustained vowel /a/ at high pitch, which is the most similar experiment to ours. All other authors used combination of vowels /a/, /i/ and /u/. Souissi et al. in [18, 19] reported the highest accuracy of 87.82 % using a subset containing 4 types of pathologies from the total number of 71 as well as Al-nasheri et al. in [2, 3] who pushed the accuracy of 99.68 %. The reason to use a subset containing only some of the pathologies was to conduct an experiment on data that were also present in other available databases, namely Massachusetts Eye and Ear Infirmary Database (MEEI) and Arabic Voice Pathology Database (AVPD). Muhammad et al. in [16] used subset containing 3 types of pathology and reported 93.20 % accuracy and then in [15] he used combination of voice recordings as well as electroglottograph (EGG) signals to boost the accuracy to 99.98 %. The highest possible accuracy of 100 % was achieved by Hemmerling et al. in [8] who approached the detection problem separately for female and male speakers. However, since the accuracy is so high the reported results are questionable.

I.3 Methodology

I.3.1 Data

We used Saarbruecken Voice Database, which is a collection of voice recordings and EGG signals from more than 2 000 persons. It contains recordings of 687 healthy persons (428 females and 259 males) and 1356 patients (727 females and 629 males) with one or more of the 71 different pathologies. One recording session contains the recordings of the following components:

- Vowels /i/, /a/, /u/ produced at normal, high and low pitch
- Vowels /i/, /a/, /u/ with rising-falling pitch
- Sentence "Guten Morgen, wie geht es Ihnen?" ("Good morning, how are you?")

All samples of the sustained vowels are between 1 and 3 seconds long, sampled at 50 kHz with 16-bit resolution [21]. In contrary to MEEI database, all audio samples (healthy and pathological) in SVD were recorded in the same environment. This preliminary experiment was conducted using samples of sustained vowel /a/ produced at normal pitch. Each file was split into multiple 64 ms long segments

(Hamming windowed) with 30 ms overlap. One file was therefore represented to the input of the neural network as a matrix containing total number of n (segments) · 3 200 features (0.064 s · 50 000 Hz = 3 200 features).

We divided all data into TRAIN (70 %), VALIDATION (15 %) and TESTING (15 %) sets and we assured that the number of healthy and pathological samples in training and validation sets are equal. The rest was appended to the testing set. In total, there were 960 samples (480 healthy, 480 pathological) in the training set, 206 samples (103 healthy, 103 pathological) in validation set and the rest 874 samples (104 healthy and 770 pathological) were used as testing samples.

I.3.2 DNN Architecture

While constructing the network, it is always good to have a clear "story" in mind that would reason the task of every layer or stack of layers in the proposed architecture. The "story" behind our architecture is simple. We used 2 stacks of convolutional layers to transform the input vectors into a set of more abstract repeating patterns that seem important for the network cost to decrease. Between each stack of convolutions, there is a pooling layer [9] that reduces the dimensionality of the vector. Since each file is a sequence of multiple time-steps (segments), all convolutions and pooling layers were wrapped in TimeDistributed layer (built in layer in Keras framework [5] for keeping the time axis unchanged). Afterwards we reshaped the resulting matrices from the last pooling layer so it could be connected to the recurrent LSTM layer. Before the experiment, we legitimized the presence of LSTM to ourselves as a context learning element that remembers the changes in time. As the last component of our network, there is a stack of 3 fully connected layers ended with Softmax layer with 2 neurons (one neuron for class = healthy and the other neuron for class = pathological) for the final classification.

For the first two convolutional layers we used 16 kernels of size 160 succeeded with max pooling layer of size 4. The second stack of another two convolutional layers used 13 kernels of size 320 again succeeded with the same max pooling as before. Then we connected the flattened output from the last layer to the LSTM layer with 25 units. To prevent overfitting we set the dropout probability [20] on LSTM layer to 0.1 for input gates and 0.5 for the recurrent connections. From this point on the DNN used only fully connected layers. The first two of size 32 and the last one with 2 output neurons and Softmax activation [4].

Rectified linear unit (Relu) as activation function [17] was used for all convolutional and dense layers except the Softmax output layer. LSTM used Hyperbolic tangent (Tanh) activation function. All layers were initialized using Glorot uniform initialization [7]. This whole DNN had overall 428 772 trainable parameters and its

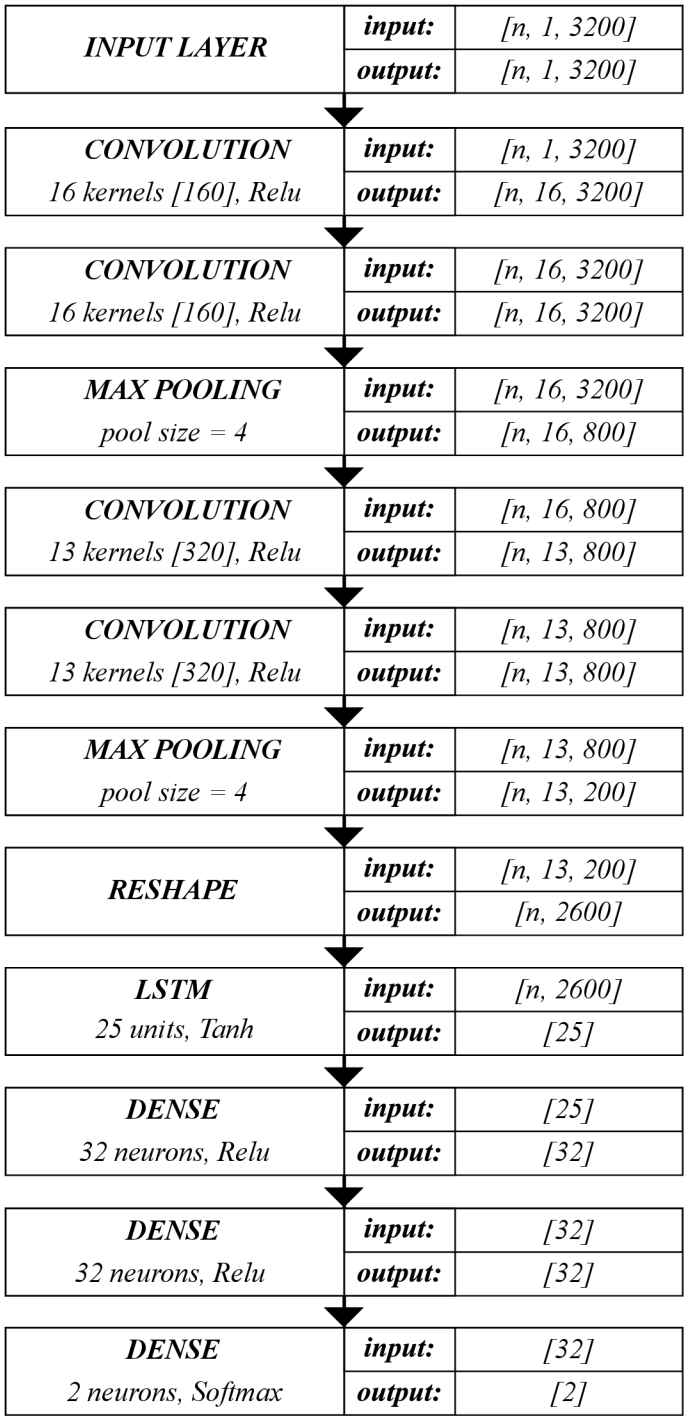


Fig. I.1: Detailed DNN architecture.

whole architecture is depicted in Fig. I.1.

DNN models consist of manifold of hyper-parameters. Sensitivity of each model on particular hyper-parameter is different due to a distinctive nature of the system that is modeled. Our strategy of its selection and fine tuning with aim of finding the best performing model was based on community standards, intuition and grid search for which we have utilized our open-source library KEX[1].

I.3.3 Experimental Setup

For gradient-based optimization of Cross-entropy loss function used during training of our proposed model we utilized Adam algorithm [11] with initial learning rate of $6 \cdot 10^{-5}$. The learning rate was not fixed and was decreased by factor 0.5 each time there was no improvement in validation accuracy for 8 consecutive training epochs (iterations). The minimum learning rate was set to $1 \cdot 10^{-7}$.

The data were presented to the DNN one file at a time (batch size = 1) in a matrix of size n (the number of segments) $\cdot$ 3 200 features for 34 epochs. We chose to use batch size equal to 1 because the length of each file is different, therefore each of the matrices had different number of segments. If we wanted to make the batches bigger, we would have to either put together files of the exact same length or cut the files to the same length.

To eliminate unnecessary training we set the patience equal to 20. That means the experiment was terminated if no progress on validation loss had been made for more than 15 epochs of training. The best results were recorded after the 25^{th} epoch. In order to train the DNN on GPU (Nvidia GeForce GTX 690) and build the models quickly, we utilized the capabilities of Keras framework. The whole 25 epochs long training took 101 minutes to finish. All hyper-parameters were tuned based on validation results.

I.4 Results

In order to perform a pathology detection using voice signal, we built a deep neural network model consisting of convolutional, pooling, LSTM and fully connected layers. We trained, validated and tested it using recordings of sustained vowel /a/ produced at normal pitch from Saarbruecken Voice Database containing 71 types of pathologies. The signal was split into 64 ms long Hamming windowed segments with 30 ms overlap and was presented to the neural network as a sequence of vectors in time. The training and validation sets contained exactly the same number of healthy and pathological samples as can be seen in Tab. I.2.

[1]KEX avaliable from http://splab.cz/en/download/software/kex-library

Table I.2: VALIDATION confusion matrix

	true: pathological	true: healthy	no. of segments
pred: pathological	67	36	103
pred: healthy	23	80	103

Table I.3: VALIDATION classification report

class	precision	f1-score	recall
pathological	0.74	0.69	0.65
healthy	0.69	0.73	0.78
overall accuracy:			71.36 %

Table I.4: TESTING confusion matrix

	true: pathological	true: healthy	no. of segments
pred: pathological	514	256	770
pred: healthy	23	81	104

Table I.5: TESTING classification report

class	precision	f1-score	recall
pathological	0.96	0.79	0.67
healthy	0.24	0.37	0.78
overall accuracy:			68.08 %

Out of 206 validation samples, the proposed trained model predicted 59 samples to belong to a wrong class as opposed to 147 correct predictions resulting in 71.36 % validation accuracy with 65.04 % sensitivity (recall of class pathological) and 77.67 % specificity (recall of class healthy). The precision, recall and f1-score of validation samples is shown in Tab. I.3.

Tab. I.4. shows the DNN predicted 279 testing samples to belong to a wrong class as opposed to 595 correct predictions resulting in 68.08 % testing accuracy with 66.75 % sensitivity and 77.89 % specificity. The precision, recall and f1-score of validation samples is shown in Tab. I.5.

I.5 Conclusions

The objective of this paper was to carry out a preliminary study which would clarify whether the use of Deep Neural Network model, especially combination of convolutional and LSTM layers would prove itself worthy of further exploration in case of Voice Pathology Detection problem using only sustained vowel. Using just recordings of vowel /a/ produced at normal pitch, the examined method achieved 71.36 % accuracy on validation data and 68.08 % accuracy on testing data. Since this result is comparable to that published in [1] we assume that further investigation is in place and could lead to much better results.

The main advantage of this approach is that one does not need to build the feature vector as opposed to the previously proposed methods, thus it saves great amount of time and expertise in the area of the problem being solved. On the other hand, the main disadvantage is the amount of data needed to train the model which is also a limitation of this experiment. The SVD database is extensive in numbers of persons recorded, but there is not enough samples of healthy persons in comparison with the number of samples of pathological patients. Also the distribution of individual pathologies is extremely unequal making the Voice Pathology Detection a hard problem, because some of the samples with certain type of pathology that occurs just once in the whole dataset can end up in testing set. Hence the network could not be trained to recognize it resulting in worse accuracy.

Our future work will build on current experiment, but we will limit the number of pathologies only to those having the most samples as in [16, 2, 18, 19] and we will train separate models for males and females as in [8]. We will investigate whether training with combination of vowels /a/, /i/ and /u/ help to improve the accuracy as in [13, 6, 8]. Also we will incorporate the data from other publicly available datasets and introduce permutation test to validate if the model learned to recognize meaningful features or just overfits on noise or remembers the samples.

Bibliography

[1] A. Al-nasheri, Z. Ali, G. Muhammad, and M. Alsulaiman. Voice pathology detection using auto-correlation of different filters bank. In *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*, pages 50–55. IEEE, 2014.

[2] A. Al-nasheri, G. Muhammad, M. Alsulaiman, and Z. Ali. Investigation of voice pathology detection and classification on different frequency regions using correlation functions. *Journal of Voice*, 31(1):3–15, 2017.

[3] A. Al-nasheri, G. Muhammad, M. Alsulaiman, Z. Ali, T. A. Mesallam, M. Farahat, K. H. Malki, and M. A. Bencherif. An investigation of multidimensional voice program parameters in three different databases for voice pathology detection and classification. *Journal of Voice*, 31(1):113–e9, 2017.

[4] J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[5] F. Chollet et al. Keras: Deep learning library for theano and tensorflow. *URL: https://keras.io/*, 2015.

[6] Ö. Eskidere and A. Gürhanlı. Voice disorder classification based on multitaper mel frequency cepstral coefficients features. *Computational and mathematical methods in medicine*, 2015, 2015.

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[8] D. Hemmerling, A. Skalski, and J. Gajda. Voice data mining for laryngeal pathology assessment. *Computers in biology and medicine*, 69:270–276, 2016.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[10] M. S. Hossain and G. Muhammad. Healthcare big data voice pathology assessment framework. *IEEE Access*, 4:7806–7815, 2016.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] D. Martínez, E. Lleida, A. Ortega, A. Miguel, and J. Villalba. Voice pathology detection on the saarbrücken voice database with calibration and fusion of scores using multifocal toolkit. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 99–109. Springer, 2012.

[14] J. Mekyska, E. Janousova, P. Gomez-Vilda, Z. Smekal, I. Rektorova, I. Eliasova, M. Kostalova, M. Mrackova, J. B. Alonso-Hernandez, M. Faundez-Zanuy, et al. Robust and complex approach of pathological speech signal analysis. *Neurocomputing*, 167:94–111, 2015.

[15] G. Muhammad, M. F. Alhamid, M. S. Hossain, A. S. Almogren, and A. V. Vasilakos. Enhanced living by assessing voice pathology using a co-occurrence matrix. *Sensors*, 17(2):267, 2017.

[16] G. Muhammad, M. Alsulaiman, Z. Ali, T. A. Mesallam, M. Farahat, K. H. Malki, A. Al-nasheri, and M. A. Bencherif. Voice pathology detection using interlaced derivative pattern on glottal source excitation. *Biomedical Signal Processing and Control*, 31:156–164, 2017.

[17] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[18] N. Souissi and A. Cherif. Dimensionality reduction for voice disorders identification system based on mel frequency cepstral coefficients and support vector machine. In *Modelling, Identification and Control (ICMIC), 2015 7th International Conference on*, pages 1–6. IEEE, 2015.

[19] N. Souissi and A. Cherif. Speech recognition system based on short-term cepstral parameters, feature reduction method and artificial neural networks. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, pages 667–671. IEEE, 2016.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] B. Woldert-Jokisz. Saarbruecken voice database. 2007.

II Towards Robust Voice Pathology Detection: Investigation of supervised deep learning, gradient boosting, and anomaly detection approaches across four databases

Outline

Bibliographic information

P. Harar, Z. Galaz, J. B. Alonso-Hernandez, J. Mekyska, R. Burget, and Z. Smekal. Towards robust voice pathology detection. *Neural Computing and Applications*, pages 1–11, 2018. arXiv:1907.06129, doi:10.1007/s00521-018-3464-7.

Author's contribution

The author significantly contributed to the survey of related works, obtained and prepared the data, designed and performed the analysis, and wrote a significant part of the manuscript. He contributed to each section of the article and organized the finalization of the whole manuscript, until its publication.

Copyright Notice

Abstract

Automatic objective non-invasive detection of pathological voice based on computerized analysis of acoustic signals can play an important role in early diagnosis, progression tracking and even effective treatment of pathological voices. In search towards such a robust voice pathology detection system we investigated 3 distinct classifiers within supervised learning and anomaly detection paradigms. We conducted a set of experiments using a variety of input data such as raw waveforms, spectrograms, mel-frequency cepstral coefficients (MFCC) and conventional acoustic (dysphonic) features (AF). In comparison with previously published works, this article is the first to utilize combination of 4 different databases comprising normophonic and pathological recordings of sustained phonation of the vowel /a/ unrestricted to a subset of vocal pathologies. Furthermore, to our best knowledge, this article is the first to explore gradient boosted trees and deep learning for this application. The following best classification performances measured by F1 score on dedicated test set were achieved: XGBoost (0.733) using AF and MFCC, DenseNet (0.621) using MFCC, and Isolation Forest (0.610) using AF. Even though these results are of exploratory character, conducted experiments do show promising potential of gradient boosting and deep learning methods to robustly detect voice pathologies.

Acknowledgement

This study was funded by the grant of the Czech Ministry of Health 16-30805A (Effects of non-invasive brain stimulation on hypokinetic dysarthria, micrographia, and brain plasticity in patients with Parkinson's disease) and the following projects: SIX (CZ.1.05/2.1.00/03.0072), and LO1401. For the research, infrastructure of the SIX Center was used. The authors (P. Harar, Z. Galaz) of this study also acknowledge the financial support of Erwin Schrödinger International Institute for Mathematics and Physics during their stay at the "Systematic approaches to deep learning methods for audio" workshop held from September 11, 2017 to September 15, 2017 in Vienna.

II.1 Introduction

Voice pathology can be caused by the presence of tissue infection, systemic changes, mechanical stress, surface irritation, tissue changes, neurological and muscular changes, and other factors [60]. Due to vocal pathology, the mobility, functionality and shape of the vocal folds are affected resulting into irregular vibrations and increased acoustic noise. Such a voice sounds strained, harsh, weak, and breathy [59, 28], which significantly contributes to the overall poor voice quality [10, 40].

Up to this day, vocal pathology detection has been approached by subjective and objective evaluations [38]. The first category (subjective evaluation) consists of so called in-hospital auditory-perceptual and visual examination of the vocal folds [47, 55]. For the visual examination laryngostroboscopy is commonly used [62]. For the auditory-perceptual examination several clinical rating scales to diagnose and rate severity of vocal pathologies have been developed [15, 19, 33, 15, 16]. Methods for subjective evaluation are, however, subject to inter-rater variability [9, 21]. Furthermore, they require patients' presence at the clinic, which can be a serious problem especially in more severe stages of a disease. This type of evaluation is also time costly, and it requires careful evaluation and scoring by clinicians.

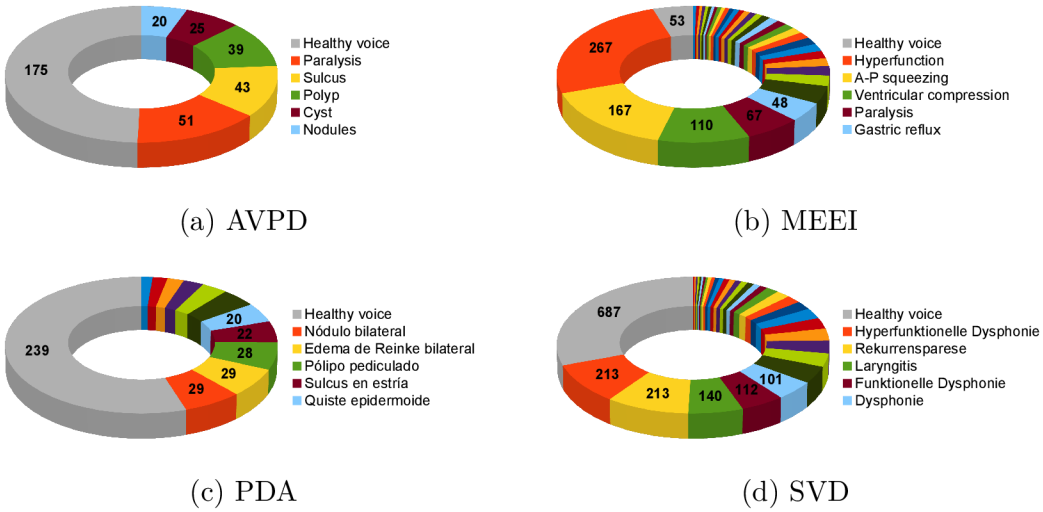
The second category (objective evaluation) is based on the automatic non-invasive computerized analysis of acoustic signals to quantify and identify the underlying vocal pathology that may not even be audible to a human being [40]. This type of evaluation is therefore inherently free from the subjective bias. Moreover, voice can be nowadays easily recorded using a variety of smart devices, and processed remotely using cloud technologies. From these reasons, works such as [17, 27, 45, 2] focused on using signal processing techniques (to quantify vocal-manifestations of the pathology under focus) and machine learning algorithms (to automate the process of voice pathology detection) to build a system capable of accurate discrimination of healthy and pathological voices. In Table II.1, we summarize recent (2015 – now) related works focused on the objective voice pathology detection.

For the purpose of the objective voice pathology evaluation, several databases have been frequently used by the researchers. Massachusetts Eye and Ear Infirmary Database (MEEI) [18, 40], Saarbruecken Voice Database (SVD) [63, 45, 2], and Arabic Voice Pathology Database (AVPD) [42, 45] are among the most commonly used ones. More specifically, most researchers have analyzed sustained phonation of the vowel /a/, e.g. [1, 44, 7, 14] due to its presence in most databases (language-independent speech task [60]). Some researchers also analyzed a combination of the vowels, e.g. [37, 17, 27], etc. From the voice pathologies point of view, most researchers restricted the dataset to a limited set of pathologies [7, 44, 14, 26, 52, 45, 5, 3, 4, 2].

Table II.1: Overview of related works focused on voice pathology detection.

First author	Year	Ref.	Database	Input vowels	Classifier	Accuracy [%]
Hemmerling	2016	[27]	SVD	/a, i, u/	KM, RF	100.00
Muhammad	2017	[44]	SVD	/a/	GMM	99.98
Al-nasheri	2017	[2]	MEEI, SVD, AVPD	/a/	SVM	99.81 (MEEI), 91.17 (AVPD), 90.98 (SVD)
Al-nasheri	2017	[3]	MEEI, SVD, AVPD	/a/	SVM	99.79 (AVPD), 99.69 (MEEI), 92.79 (SVD)
Al-nasheri	2017	[4]	MEEI, SVD, AVPD	/a/	SVM	99.68 (SVD), 88.21 (MEEI), 72.53 (AVPD)
Eskidere	2015	[17]	SVD	/a, i, u/	GMM	99.00
Amami	2017	[7]	MEEI	/a/	SVM	98.00
Sabir	2017	[52]	SVD	/a/	ANN	97.90
Hossain	2016	[29]	MEEI, SVD	/a, i, o/	SVM, ELM, GMM	95.00
Ali	2017	[5]	MEEI, SVD, AVPD	/a/	GMM	94.60 (MEEI), 83.65 (AVPD), 80.20 (SVD)
Muhammad	2017	[45]	MEEI, SVD, AVPD	/a/	SVM	99.40 (MEEI), 93.20 (SVD), 91.50 (AVPD)
Dahmani	2017	[14]	SVD	/a/	NB	90.00
Souissi	2016	[57]	SVD	/a/	ANN	87.82
Hemmerling	2017	[26]	SVD	/a/	ANN	87.50
Souissi	2015	[56]	SVD	/a/	SVM	86.44

Table notation: Ref. – reference, MEEI – Massachusetts Eye and Ear Infirmary Database [18, 40], SVD – Saarbruecken Voice Database [63, 45, 2], AVPD – Arabic Voice Pathology Database [42, 45], KM – K-means [24], RF – Random Forests [11], GMM – Gaussian Mixture Models [51], SVM – Support Vector Machines [25], NB – Naive Bayes [46], ELM – Extreme Learning Machine [31], and ANN – Artificial Neural Networks [54].



(a) AVPD (b) MEEI

(c) PDA (d) SVD

Fig. II.1: Visualization of inequality of samples per vocal pathology in the datasets used in this work (only 5 most common pathologies in each database are present in the legend), and healthy samples. Databases: a) AVPD [42, 45], b) MEEI [18, 40], PDA [20, 8, 40], and SVD [63, 45, 2].

Next, conventional and clinically interpretable [10] acoustic features were usually computed prior to pathology detection [44, 14, 52]. The acoustic features such as multidimensional voice program parameters (MDVP) [4], mel-frequency cepstral coefficients (MFCC) [53], glottal-to-noise excitation ratio (GNE) [43], etc. were usually extracted. For more information about methods for pathological speech parametrization, see [40]. After the feature extraction, multiple conventional classifiers have been used to detect the presence of voice pathology. Most authors relied on the following algorithms: Support Vector Machines (SVM), Gaussian Mixture Models (GMM), Random Forests (RF), and Artificial Neural Networks (ANN) [26, 5, 7, 14], etc.

As can be seen, the results vary greatly between the published papers mainly due to differences in selected voice pathology samples, acoustic features, and classifiers that were used for the experiment. However, we can conclude that:

1. most works analyzed a single speech task, mainly the sustained phonation of the vowel /a/ (language independent speech task)
2. most works analyzed datasets that were restricted to a subset of vocal pathologies from 1 to 3 databases (MEEI, SVD, AVPD)
3. most works extracted conventional dysphonic features to quantify major vocal-manifestations of specific vocal pathologies
4. most works employed conventional supervised learning algorithms such as the following: SVM, GMM, RF, ANN, and others

To propose results comparable with the previously published works, we analyze voice recordings of sustained phonation of the vowel /a/ as well. However, unlike the previous works, we analyze a larger dataset composed of 4 different databases, namely: MEEI [18, 40], SVD [63, 45, 2], AVPD [42, 45] (these databases are commonly used by the community), and Príncipe de Asturias Database (PDA) [20, 8, 40]. Furthermore, to propose models capable of robust voice pathology detection, we do not restrict the dataset to only a subset of common vocal pathologies. With this approach, our dataset does contain a large number of pathologies with only few recordings. To see the sparsity of distribution and inequality of the number of pathologies in the databases, see Figures II.1a (AVPD), II.1b (MEEI), II.1c (PDA), and II.1d (SVD).

By using 4 different databases, we aim to increase the size of our dataset to enable exploring possibilities of using supervised deep learning techniques that delivered state-of-the-art results in many domains including speech processing. To our best knowledge, despite our previous work [22], there are no other papers using deep learning algorithms for voice pathology detection. Next, we also employ the conventional voice pathology detection approach based on acoustic feature extraction procedure. However, unlike previous works, we use gradient boosting techniques for classification. To tackle the problem of sparse distribution of a variety of vocal pathologies with only few recordings across the databases, we also investigate usage of anomaly detection procedure.

The rest of this paper is organized as follows. Section II.2 introduces databases utilized in this article. In Section II.3, the methodology of the experiment is discussed. The results are presented in Section II.4. Conclusions are drawn in Section II.5.

II.2 Databases

As mentioned previously, we chose the following speech task: sustained phonation of the vowel /a/ as a basis for our experiments. During this particular speech task a speaker is asked to sustain phonation of a vowel, attempting to maintain steady frequency and amplitude at a comfortable level [60]. The advantage of this speech task in comparison with other common speech tasks such as reading tasks, or a running speech is that it is free of articulatory and other linguistic confounds [60]. This independence makes this task an ideal choice to construct a large dataset that is necessary for supervised deep learning algorithms. In fact, sustained /a/ vowel phonation is the only speech task that is present in all databases used in this work. The contents of the databases relevant to this work can be seen in Table II.2.

II.2.1 AVPD database

Arabic Voice Pathology Database (AVPD) [42, 45] was developed at the Communication and Swallowing Disorders Unit of King Abdul Aziz University Hospital, Riyadh, Saudi Arabia. The database contains recordings (366 samples: 188 healthy, 178 pathological) of sustained phonation of the vowels /a, e, o/, counting from 0-10, standardized Arabic passage, and reading three words. All recordings are sampled at $f_s = 48\,000\,\text{Hz}$ with a bit depth of 16 bits. The database comprises five organic voice disorders: vocal fold cysts, nodules, paralysis, polyps, and sulcus. Multiple recordings of the same vowel were taken to help model the intra-speaker variability.

II.2.2 MEEI database

Massachusetts Eye and Ear Infirmary Database (MEEI) [18, 40] is one of the most popular and most widely-used database (used for many years as a benchmark in the field of pathological speech analysis). It contains more than 1 400 recordings of sustained phonation of the vowel /a/ (recorded from 657 pathological speakers with different types of pathologies and 53 healthy speakers). This database has several disadvantages such as the fact that recordings of the normophonic voice were recorded in different conditions (e.g. different environment, recordings are sampled at: $f_s = 50\,000\,\text{Hz}$, $f_s = 25\,000\,\text{Hz}$, and $f_s = 10\,000\,\text{Hz}$) when compared to pathological recordings. The database is also gender-unbalanced, etc.

II.2.3 PDA database

Príncipe de Asturias Database (PDA) [20, 8, 40] contains recordings of 200 pathological speakers with different types of organic pathologies (e.g. nodules, polyps, oedemas, and carcinomas, etc.) and 239 healthy speakers. For each speaker, sustained phonation of the vowel /a/ is recorded. All recordings are sampled at $f_s = 25\,000\,\text{Hz}$. This database contains more speakers than a balanced version of MEEI database that according to [48] comprise only 173 recordings of pathological speakers.

II.2.4 SVD database

Saarbruecken Voice Database (SVD) [63, 45, 2] is a collection of voice recordings and electroglottography (EGG) signals from more than 2 000 speakers. It contains recordings of 687 healthy persons (428 females and 259 males) and 1356 patients (727 females and 629 males) with one or more of the 71 different pathologies. One session contains the recordings of the following components: a) vowels /i, a, u/ produced at normal, high and low pitch; vowels /i, a, u/ with rising-falling pitch; and c)

Table II.2: Contents of the databases used in this work.

	AVPD	MEEI	PDA	SVD
H samples	188	53	239	687
P samples	178	657	200	1356
vowels	/a, e, o/	/a/	/a/	/a, i, u/
running speech	yes	yes	no	yes
EGG	no	no	no	yes
GRBAS	yes	no	no	no

Table notation: PDA – Príncipe de Asturias Database (PDA) [20, 8, 40], MEEI – Massachusetts Eye and Ear Infirmary Database [18, 40], SVD – Saarbruecken Voice Database [63, 45, 2], AVPD – Arabic Voice Pathology Database [42, 45], H – healthy, P – pathological, and GRBAS – Grade, Roughness, Breathiness, Asthenia, Strain scale [15].

sentence "Guten Morgen, wie geht es Ihnen?" ("Good morning, how are you?"). All samples of the sustained vowels are between 1 to 3 seconds long, sampled at $f_s = 50\,000$ Hz with 16-bit resolution [63]. In contrary to MEEI database, all audio samples (healthy and pathological) in SVD were recorded in the same environment.

II.3 Methodology

II.3.1 Data processing

We used 720 recordings from AVPD, 709 recordings from MEEI, 422 from PDA and 2 040 from SVD. We only excluded samples that were shorter than 0.750 s in length (removed 319 recordings). We also excluded all recordings of speakers bellow the age of 19 and also above the age of 60 (it is known that the most significant changes of voice happen during adulthood until the voice matures at around age of 20 and remains relatively stable until around age of 60) [58]. After these restrictions, the final number of samples equaled to 2 707.

Using SOX library (version 14.4.2), we re-sampled each recording to $f_s = 16\,000$ Hz. Then we trimmed each sample to exactly 0.750 s in duration. If a recording was below 0.950 s in duration, we extracted only one sample from the middle of it. For longer recordings we trimmed each end by 0.100 s and extracted as many 0.750 s long chunks as possible without overlap with stride of 0.375 s. Using this approach, the total number of 8 042 chunks was obtained. Further details regarding the number of chunks used can be found in Table II.3.

Table II.3: Number of chunks used in the experiments.

Database	H (M)	P (M)	H (F)	P (F)	Total
AVPD	625	509	872	804	2810
MEEI	126	114	185	168	593
PDA	1158	331	5	605	2099
SVD	400	645	624	871	2540
Total	2309	1599	1686	2448	**8042**

Table notation: PDA – Príncipe de Asturias Database (PDA) [20, 8, 40], MEEI – Massachusetts Eye and Ear Infirmary Database [18, 40], SVD – Saarbruecken Voice Database [63, 45, 2], AVPD – Arabic Voice Pathology Database [42, 45], H – healthy, P – pathological, M – males, and F – females.

II.3.2 Feature extraction

At first, we considered raw audio samples as an input data for the voice pathology detection model. Each file (the 0.750 s chunk) was therefore inserted to the input of the neural network as a vector of 12 000 features (computed as: 0.750 s · 16 000 Hz = 12 000 features). Additionally, we normalized each sample using min-max scaling to a range $\langle 0, 1 \rangle$.

Next, we extracted a set of conventional commonly-used acoustic (dysphonic) features [10, 40] using Neurological Disorder Analysis Tool (NDAT) [41, 40] written in MATLAB and developed at the Brno University of Technology. Specifically, we computed the following acoustic features: pitch, jitter, shimmer, harmonic-to-noise ratio, detrended fluctuation analysis parameters, glottis quotients (open, closed), glottal-to-noise excitation ratio, Teager-Kaiser energy operator, modulation energy, and normalized noise energy. We further applied the following statistical properties: mean, standard deviation, coefficient of variation, quartiles (1^{st}, 2^{nd}, 3^{rd}), interquartile range, kurtosis, and skewness.

Moreover, we computed spectrograms using Matplotlib (version 2.1.2) library for Python. The computation setup: mode (power spectral density), no windowing, no overlap, and NFFT (512 samples). Following Ali et al. [6], we used data up to 1 500 Hz (1 150 features). Furthermore, we normalized the values of this matrix with min-max scaling to a range between 0 and 1 as well.

At last, we computed most commonly used perceptual [41] acoustic feature: MFCC using Python Speech Features library. The computation setup: length of a window function (0.025 s), step size (0.010 s), number of filters in the filter-bank (26), number of coefficients (13), and NFFT (512 samples). With this approach,

we obtained a matrix consisting of 962 features (13 coefficients $\times$ 74 time steps). We also computed the mean and standard deviation of the 13 coefficients along the time axis, which resulted into additional 26 features per sample. Next, we scaled the MFCC feature matrix by min-max algorithm (means and standard deviations were computed before scaling). The statistical features were scaled separately to have 0 mean and unit variance before classification.

II.3.3 Experiments

As mentioned previously, there is a wide range of pathologies present in the databases used in this work. For more information, see Figures II.1a (AVPD), II.1b (MEEI), II.1c (PDA), and II.1d (SVD). Each database was labelled in different language and with different experts by different criteria. Therefore, it is almost impossible to combine these databases to obtain one consistent database of multiple pathologies with reasonable number of samples. Only feasible way of combining the samples seems to be the exhaustive manual pairing by an expert clinician, which is also rather difficult since lots of recordings are labelled with multiple pathologies. In order to conduct inter-database experiments, authors therefore usually pick a smaller sub-sample of 2 to 5 pathology types that are relatively easier to pair.

Next, most of these pathologies are very sparsely distributed across the databases. Searching for an ideal subset of acoustic features that would yield a good classification performance for each voice pathology is therefore almost impossible. Furthermore, it is not well-known if these pathologies present in the databases have similar vocal-manifestations.

In contrast to the previous works, we aim to investigate possibilities of robust voice pathology detection using a set of 4 almost unrestricted databases comprising a large number of pathologies. From these reasons, we decided to conduct several experiments: a) supervised learning (assuming the pathologies have similar manifestations and therefore the number of samples per pathology type is irrelevant), b) anomaly detection (assuming the pathologies do not have similar manifestations and therefore the number of samples per pathology type cannot be neglected).

Regarding the supervised learning approach, we used the state-of-the-art gradient boosting algorithm: XGBoost [12] (version 0.6) for its current successes in many Kaggle competitions, fast training and model interpretability. Additionally, we explored possibilities of deep learning approach for its ability to robustly model complex relationships when optimized using enough data. However, the equation for computing the sufficient size of training dataset has not been formally described yet. Generally established rule of thumb in machine learning community is to have more training samples than trainable parameters. For this reason, we used the

DenseNet [30] architecture, which succeeded in overcoming the problem of having too many trainable parameters by densely connecting the convolutional layers. We adjusted Thibault de Boissiere's Keras implementation of the DenseNet (Keras framework [13], version 2.1.2), to process 1D signals treating raw audio as 1D vector. Spectrograms were processed as a matrix using the frequency bins not as y dimension, but rather as a stack of channels in the same way the 3 RGB channels are stacked in an image [64]. The MFCC were processed the same way as spectrograms. Since we are not able to say with 100% certainty that healthy examples are not polluted by deviant samples, we decided to use anomaly detection in favor of novelty detection in which case it is important to model the non-deviant samples. In this case, we chose Isolation Forest [35, 36] classifier implemented in scikit-learn library [49] (version 0.19.1).

For the above mentioned experiments, we decided to analyze the performance of the voice pathology detection models using multiple types of input data: a) raw audio samples to follow our previous work [23] and further explore possibilities of robust voice pathology detection without manually-selected features (DenseNet), b) conventional acoustic (dysphonic) features to follow the previously published works and quantify most common vocal pathologies (XGBoost, Isolation Forest), c) spectrograms to achieve a reasonable trade-off between dimensionality of the data and amount of information (DenseNet), and d) MFCC to follow the previous works focusing on voice and speech modelling, and voice pathology detection (all models).

II.3.4 Training and validation

To train and validate the models, we split the data to training, validation and testing sets. On top of that, we generated 10-fold validation indices using training and validation sets, so we can use exactly the same sets of data for each experiment. The test set was left for final evaluation of the models. Next, we stratified the testing and validation sets by medical state (healthy – H, pathological – P), gender, age, and gender-age group. Since the longer recordings were split into multiple chunks, we had to prevent the samples with chunks in the test or validation sets from leaking into the training set. Such chunks were carefully removed from the set. All other chunks were used in the training set.

At this point there is an unequal distribution of samples within the training set. We reacted to this fact by computing sample weights that can be used during training as a compensation measure for under-represented groups. The final sample weight is a product of 3 partial weights. Each of the partial weights quantifies the ratio between subgroups within the selected group (e.g. the ratio between the

number of normophonic and pathological samples). For this purpose, we introduced a class weight α, gender weight β, and gender-age group weight γ resulting in final sample weight ω that is computed as $\omega = \alpha \cdot \beta \cdot \gamma$. Weight for a particular sample that belongs to subgroup α_i within group α, β_i within group β, and γ_i within group γ can be computed as follows:

$$\omega_{\alpha_i,\beta_i,\gamma_i} = \frac{n_{\alpha_i}}{\max(n_\alpha)} \cdot \frac{n_{\beta_i}}{\max(n_\beta)} \cdot \frac{n_{\gamma_i}}{\max(n_\gamma)} \tag{II.1}$$

where n represents the total number of samples: n_{α_i} is the number of samples in a particular class; n_{β_i} is the number of samples in a particular gender; and n_{γ_i} is the number of samples in a particular gender-age group.

We used 30 to 100 iterations of randomized cross-validation search for hyper-parameter optimization for both XGBoost and Isolation Forest classifiers. The number of iterations did depend on the fitting time. More specifically, in the case of XGBoost, we were searching over the following hyper-parameters: number of estimators $\langle 3, 300 \rangle$, learning rate $\langle 0.006, 1 \rangle$, gamma $\langle 10, 60 \rangle$, maximum depth $\langle 0, 9 \rangle$, minimum child weight $\langle 1, 3 \rangle$, sub-sample ratio $\langle 0.3, 1 \rangle$ and colsample bytree (sub-sample ratio of columns when constructing each tree) $\langle 0.1, 1 \rangle$. Regarding Isolation Forest we were searching over the following hyper-parameters: number of estimators $\langle 6, 200 \rangle$, maximum samples $\langle 8, 64 \rangle$, contamination $\langle 0.40, 0.76 \rangle$ and maximum features $\langle 0.05, 1 \rangle$. As a performance measure, we used F1 micro score as a criteria of choosing the best hyper-parameters in the cross-validation setup. After the search for hyper-parameter, we re-fitted the models with the best hyper-parameters on the entire training set, and consequently evaluated on the testing set. The final results are presented in the form of confusion matrix (CM), and classification report (CR) tables. The formulas II.2, II.3 and II.4 describe the way of computing the precision, recall and F1 score (weighted average of the precision and recall) metrics presented in CR tables.

$$precision = \frac{tp}{tp + fp}, \tag{II.2}$$

where tp denotes the number of correct predictions (observed class), and fp determines the number of incorrect predictions (observed class). The precision is a ratio between the number of correct predictions of the observed class and the total number of predictions of the observed class.

$$recall = \frac{tp}{tp + fn}, \tag{II.3}$$

where tp denotes the number of correct predictions (observed class), and fn determines the number of incorrect predictions (opposing class). The recall is a ratio

between the number of correct predictions of the observed class and the total number of samples in the observed class.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{II.4}$$

II.4 Results

XGBoost [12] trained (10-fold validation) with all features (consisting of 96 conventional dysphonic features and 26 MFCC coefficients) yielded an average F1 score of $0.922\,(\pm\,0.004)$ on the training set, and $0.829\,(\pm\,0.028)$ on the validation set. The final F1 score on the dedicated testing set was 0.733. Performance details (classification matrix and classification report) can be found in Table II.4, and Table II.5. Based upon the performance on the development set (training and validation sets) the 50 iterations of randomized cross-validated search selected the following hyper-parameters: number of estimators (294), learning rate (0.3), gamma (10), max. depth (3), sub-sample (0.5), minimum child weight (1), colsample bytree (1). Details regarding the classification performance in relation to input data can be found in Table II.6.

Regarding deep learning approach, we used the adjusted DenseNet [30] architecture with the binary cross-entropy loss optimized using Adam optimizer [32]. The initial learning rate was set to 0.01 with decay of $1e-04$ on each epoch. Hyper-parameter optimization was done using training and validation sets, and the final parameters of the DenseNet network were set as follows: depth (4), number of dense blocks (2), growth rate (5), number of filters (10), drop-out rate (0.3), l2 weight decay ($1e-04$). The input shape of this network was (13×47) with one neuron in the last layer with sigmoid activation function, and the total of 1 629 trainable parameters. For this particular setup with MFCC as the input data, the system yielded F1 score of 0.595 on the training set, and 0.648 on the validation set. After the hyper-parameter optimization, we retrained the network on all data from the training and validation sets (the development set), and the system yielded the final F1 score on the dedicated testing set of 0.621. Performance details (classification matrix and classification report) can be found in Table II.7 and Table II.8.

DenseNet trained with spectrograms had input shape (46×25) and total of 301 trainable parameters. Even though this setup was considerably less complex, and regularized with drop-out (0.3) and l2 weight decay ($1e-04$), the network tended to over-fit after enough training epochs, which we prevented using early stopping that monitored changes in the validation accuracy. This system yielded F1 score of 0.635 and 0.531 on the training and validation sets, respectively. The performance on the testing set was 0.562 (F1 score 0.514 for class H and 0.609 for class P). After

Table II.4: Testing CM for XGBoost

	true H	true P	total predicted
predicted H	**82**	26	108
predicted P	38	**94**	132
total true	120	120	accuracy: **0.733**

Table II.5: Testing CR for XGBoost

	precision	recall	f1-score	no. samples
class H	0.759	0.683	0.719	120
class P	0.712	0.783	0.746	120
avg. / total	0.736	0.733	**0.733**	240

Table II.6: XGBoost performance related to input data

Input data	F1 CV train	F1 CV valid	F1 test
ALL	0.922 ($\pm$ 0.004)	0.829 ($\pm$ 0.028)	0.733
AF stats	0.886 ($\pm$ 0.004)	0.791 ($\pm$ 0.034)	0.686
AF	0.892 ($\pm$ 0.006)	0.798 ($\pm$ 0.025)	0.658
AF base	0.745 ($\pm$ 0.009)	0.689 ($\pm$ 0.036)	0.646
MFCC	0.680 ($\pm$ 0.010)	0.769 ($\pm$ 0.037)	0.623

Table notation and description of acoustic features used to build XGBoost model: MFCC – 26 Mel Frequency Spectral Coefficients (13 means & 13 standard deviations), AF base – 12 common acoustic (dysphonic) features, AF stats. – 84 acoustic features' statistics, AF – AF base & AF stats, ALL – AF & MFCC.

Table II.7: Testing CM for DenseNet (MFCC)

	true H	true P	total predicted
predicted H	**73**	44	117
predicted P	47	**76**	123
total true	120	120	accuracy: **0.621**

Table II.8: Testing CR for DenseNet (MFCC)

	precision	recall	f1-score	no. samples
class H	0.624	0.608	0.616	120
class P	0.618	0.633	0.626	120
avg. / total	0.621	0.621	**0.621**	240

Table II.9: Testing CM for Isolation Forest

	true H	true P	total predicted
predicted H	**58**	30	88
predicted P	62	**90**	152
total true	120	120	accuracy: **0.617**

Table II.10: Testing CR for Isolation Forest

	precision	recall	f1-score	no. samples
class H	0.659	0.483	0.558	120
class P	0.592	0.750	0.662	120
avg. / total	0.626	0.617	**0.610**	240

refitting on the whole development set, the final F1 score got worse on the dedicated testing set to 0.460 due to difficulties with classification of healthy voices (F1 score 0.239 for class H and 0.680 for class P). With raw input data, the network failed to learn any meaningful features (the size of out training dataset is still too small to provide deep learning algorithm to overcome more conventional approaches).

Hyper-parameter optimization for Isolation Forest trained (10-fold validation) with 96 speech parameters was done the same way as for XGBoost. The best parameters selected upon performance on the development set were as follows: number of estimators (200), contamination (0.4), maximum features (0.3), maximum samples was set to "auto". The system yielded F1 score of 0.576 ($\pm$ 0.005) on the training set and 0.578 ($\pm$ 0.023) on the validation set. The final F1 score on the dedicated testing set was 0.610. The performance details (classification matrix and classification report) can be found in Table II.9 and Table II.10. This system showed to be sensitive to the number of input features and the performance raised when we selected just a subset of them.

II.5 Conclusions

In search towards robust voice pathology detection system using acoustic (voice) signals, researchers face a variety of problems. One of the major problems in this field of science is the limited number of available databases. Moreover, commonly used databases [18, 45, 42, 20] are very hard to combine because of various distinctions such as: a) the databases are labeled in different languages, b) the databases do not comprise same set of speech tasks, c) there is a variety of voice pathologies unequally distributed across the databases, etc. For these reasons, researchers have used only a subset of the databases for their experiments providing results related to those carefully selected subset of data. However, this approach limits the possibilities of creating a robust voice pathology detector. Therefore, in this work, we have conducted experiments on recordings of sustained phonation of the vowel /a/ produced at normal pitch from 4 different databases trying to eliminate those limitation. To the best of our knowledge, this is the first work that uses such a large set of data to build mathematical models for computerized, objective voice pathology detection.

We researched 3 distinct classifiers within supervised learning and anomaly detection paradigms. We have explored raw waveforms, spectrograms, MFCC and conventional dysphonic features as input data. All experiments were evaluated by the same criteria on the same dedicated testing set. We observed that XGBoost classifier achieved the best results amongst DenseNet and Isolation Forest classifiers. We also observed that not only XGBoost provided the best performance, it could also handle the feature selection (input: all features) by itself in contrary

to Isolation Forest classifier, which showed to be sensitive on the feature selection (input: manually selected subset of features). Overall advantage of using speech features and MFCC with XGBoost was the computation speed that allowed us to use exhaustive randomized cross-validated search to optimize the hyper-parameters, as well as the possibility to sort features by importance. This property is useful for clinical interpretability. Nevertheless, we consider these results exploratory due to the limitations of the databases. Reviewing the performances achieved in scenarios with MFCC as input data we conclude that MFCC alone are not reliable enough for robust voice pathology detection, which was also concluded by Ali et al. in [5]. Regarding the DenseNet, we conclude that in voice pathology detection scenarios with this little training data it is better to use inputs with reduced dimensionality in contrary to raw waveform inputs, or make use of transfer learning or data augmentation.

In this article there are several limitations. Firstly, there are limitations inherited from the databases along with new limitations caused by their combination. For instance, some databases have extremely unequal distribution of healthy and pathological classes, most of the databases have alarming inequalities between the number of samples per pathology type (e.g. many pathologies are present less than 3 times in the database), see Figures II.1a (AVPD), II.1b (MEEI), II.1c (PDA), and II.1d (SVD). Most databases have no information about severity of the pathology, nor they have information about manifestation of the pathology in phonation, which means that some of the samples might sound as healthy even though they are labelled as pathological and vice versa. Not to mention that recordings are labelled with more than 1 type of pathology, and in different languages, which makes it especially hard to combine or exclude the samples. Since we used 4 available databases, we utilized only the speech task available in all of them: sustained phonation of the vowel /a/ produced at normal pitch. Secondly, even though we have taken countermeasures to balance the classes with sample weights, we did not conduct our experiments separately on subsets of data for different genders.

Up to this point, most papers focused on voice pathology detection used conventional dysphonic features to quantify the underlying voice pathology. In general, these features are conceptually simple, which on one hand is an advantage as these features are clinically interpretable (i.e. clinicians are able to associate the values of the features with known physiological phenomena inside human body) [41], but on the other hand these features are often unable to describe the exact voice pathology under focus in a more complex way, especially in advanced stages of the disease (high level of acoustic noise, irregularity of voice, etc.). In future studies, researchers may consider exploring usage of a more sophisticated set of acoustic features to complexly and robustly describe the voice and speech production deterioration.

For instance, such features have already been successfully applied in the field of non-invasive assessment of Parkinson's disease [34, 61, 39].

With the previously mentioned facts in mind, we think that recordings of the databases commonly used for automatic voice pathology detection should be consulted with clinicians to evaluate the severity of vocal manifestation of the present pathologies. There are standard metrics, which are used to evaluate the quality of voice that can be used for this purpose [15, 19, 33, 15, 16]. Addition of such information to the databases could provide researchers with a unique possibility to build models capable of classification and prediction emphasizing the severity of the exact vocal-manifestation (increased acoustic tremor, roughness, breathiness, etc.) of these pathologies.

We also anticipate that deep learning will play its role in robust voice pathology detection on the assumption that more data will be available, or at least reasonable combination of available databases will be made and limitations of these databases will be partially diminished by data augmentation and other countermeasures. In addition, we presume that use of deep learning methods for novelty detection such as deep autoencoder [50] for modelling the normophonic voice could be an interesting idea for future investigation with prospect to identify even disordered voices that are sparsely distributed across databases.

In summary, acoustic (voice) signals can nowadays be recorded using a variety of smart devices and processed remotely using modern cloud technologies. In comparison with the conventional perceptual voice quality examination, computerized acoustic analysis of voice signals can provide clinicians with fast, supportive methodology of objective voice pathology detection, assessment, and monitoring that can be used on everyday basis (see Health 4.0). However, to take advantage of such methodology, robust mathematical models capable of precise voice pathology detection must be introduced. Our work proposes the next step towards this goal using various state-of-the-art machine learning algorithms applied to the largest dataset that have been used for the purpose of automatic voice pathology detection.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Bibliography

[1] A. Al-nasheri, Z. Ali, G. Muhammad, and M. Alsulaiman. Voice pathology detection using auto-correlation of different filters bank. In *Computer Systems*

and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, pages 50–55. IEEE, 2014.

[2] A. Al-nasheri, G. Muhammad, M. Alsulaiman, and Z. Ali. Investigation of voice pathology detection and classification on different frequency regions using correlation functions. *Journal of Voice*, 31(1):3–15, 2017.

[3] A. Al-nasheri, G. Muhammad, M. Alsulaiman, Z. Ali, K. Malki, T. Mesallam, and M. Farahat. Voice pathology detection and classification using auto-correlation and entropy features in different frequency regions. *IEEE Access*, PP(99):1–1, 2017. doi:10.1109/ACCESS.2017.2696056.

[4] A. Al-nasheri, G. Muhammad, M. Alsulaiman, Z. Ali, T. A. Mesallam, M. Farahat, K. H. Malki, and M. A. Bencherif. An investigation of multidimensional voice program parameters in three different databases for voice pathology detection and classification. *Journal of Voice*, 31(1):113–e9, 2017.

[5] Z. Ali, M. Alsulaiman, G. Muhammad, I. Elamvazuthi, A. Al-nasheri, T. A. Mesallam, M. Farahat, and K. H. Malki. Intra-and inter-database study for arabic, english, and german databases: Do conventional speech features detect voice pathology? *Journal of Voice*, 31(3):386–e1, 2017.

[6] Z. Ali, G. Muhammad, and M. F. Alhamid. An automatic health monitoring system for patients suffering from voice complications in smart cities. *IEEE Access*, 5:3900–3908, 2017.

[7] R. Amami and A. Smiti. An incremental method combining density clustering and support vector machines for voice pathology detection. *Computers & Electrical Engineering*, 57:257–265, 2017.

[8] J. D. Arias-Londoño, J. I. Godino-Llorente, M. Markaki, and Y. Stylianou. On combining information from modulation spectra and mel-frequency cepstral coefficients for automatic detection of pathological voices. *Logopedics Phoniatrics Vocology*, 36(2):60–69, 2011.

[9] D. Armstrong, A. Gosling, J. Weinman, and T. Marteau. The place of inter-rater reliability in qualitative research: an empirical study. *Sociology*, 31(3):597–606, 1997.

[10] L. Brabenec, J. Mekyska, Z. Galaz, and I. Rektorova. Speech disorders in parkinson's disease: early diagnostics and effects of medication and brain stimulation. *Journal of Neural Transmission*, 124(3):303–334, 2017.

[11] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[13] F. Chollet et al. Keras: Deep learning library for theano and tensorflow. *URL: https://keras.io/*, 2015.

[14] M. Dahmani and M. Guerti. Vocal folds pathologies classification using naïve bayes networks. In *Systems and Control (ICSC), 2017 6th International Conference on*, pages 426–432. IEEE, 2017.

[15] M. S. De Bodt, F. L. Wuyts, P. H. Van de Heyning, and C. Croux. Test-retest study of the grbas scale: influence of experience and professional background on perceptual rating of voice quality. *Journal of Voice*, 11(1):74–80, 1997.

[16] P. H. Dejonckere, P. Bradley, P. Clemente, G. Cornut, L. Crevier-Buchman, G. Friedrich, P. Van De Heyning, M. Remacle, and V. Woisard. A basic protocol for functional assessment of voice pathology, especially for investigating the efficacy of (phonosurgical) treatments and evaluating new assessment techniques. *Eur Arch Otorhinolaryngol.*, 258(2):77–82, Feb. 2001.

[17] Ö. Eskidere and A. Gürhanlı. Voice disorder classification based on multitaper mel frequency cepstral coefficients features. *Computational and mathematical methods in medicine*, 2015, 2015.

[18] M. Eye and E. Infirmary. Voice disorders database, version. 1.03 (cd-rom). *Lincoln Park, NJ: Kay Elemetrics Corporation*, 1994.

[19] B. R. Gerratt, J. Kreiman, N. Antonanzas-Barroso, and G. S. Berke. Comparing internal and external standards in voice quality judgments. *J Speech Hear. Res.*, 36(1):14–20, Feb. 1993.

[20] J. I. Godino-Llorente, P. Gómez-Vilda, F. Cruz-Roldán, M. Blanco-Velasco, and R. Fraile. Pathological likelihood index as a measurement of the degree of voice normality and perceived hoarseness. *Journal of Voice*, 24(6):667–677, 2010.

[21] K. L. Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC, 2014.

[22] P. Harar, J. B. Alonso-Hernandezy, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice pathology detection using deep learning: a preliminary study.

In *Bioinspired Intelligence (IWOBI), 2017 International Conference and Workshop on*, pages 1–4. IEEE, 2017.

[23] P. Harar, J. B. Alonso-Hernandezy, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice pathology detection using deep learning: a preliminary study. In *Bioinspired Intelligence (IWOBI), 2017 International Conference and Workshop on*, pages 1–4. IEEE, 2017.

[24] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[25] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[26] D. Hemmerling. Voice pathology distinction using autoassociative neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 1844–1847. IEEE, 2017.

[27] D. Hemmerling, A. Skalski, and J. Gajda. Voice data mining for laryngeal pathology assessment. *Computers in biology and medicine*, 69:270–276, 2016.

[28] J. Hillenbrand and R. A. Houde. Acoustic correlates of breathy vocal quality: Dysphonic voices and continuous speech. *J Speech Hear Res*, 39(2):311–21, 1996.

[29] M. S. Hossain and G. Muhammad. Healthcare big data voice pathology assessment framework. *IEEE Access*, 4:7806–7815, 2016.

[30] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[31] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[32] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] J. Kreiman, B. R. Gerratt, G. B. Kempster, A. Erman, and G. S. Berke. Perceptual evaluation of voice quality: review, tutorial, and a framework for future research. *J Speech Hear. Res.*, 36(1):21–40, Feb. 1993.

[34] M. Little, P. McSharry, E. Hunter, J. Spielman, and L. Ramig. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE T Bio-Med Eng*, 56(4):1015–1022, 2009.

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008.

[36] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3, 2012.

[37] D. Martínez, E. Lleida, A. Ortega, A. Miguel, and J. Villalba. Voice pathology detection on the saarbrücken voice database with calibration and fusion of scores using multifocal toolkit. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 99–109. Springer, 2012.

[38] D. D. Mehta and R. E. Hillman. Voice assessment: updates on perceptual, acoustic, aerodynamic, and endoscopic imaging methods. *Current opinion in otolaryngology & head and neck surgery*, 16(3):211, 2008.

[39] J. Mekyska, Z. Galaz, Z. Mzourek, Z. Smekal, and I. Rektorova. Assessing progress of Parkinson's using acoustic analysis of phonation. In *2015 International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 115–122, June 2015. doi:10.1109/IWOBI.2015.7160153.

[40] J. Mekyska, E. Janousova, P. Gomez-Vilda, Z. Smekal, I. Rektorova, I. Eliasova, M. Kostalova, M. Mrackova, J. B. Alonso-Hernandez, M. Faundez-Zanuy, et al. Robust and complex approach of pathological speech signal analysis. *Neurocomputing*, 167:94–111, 2015.

[41] J. Mekyska, Z. Smekal, Z. Galaz, Z. Mzourek, I. Rektorova, M. Faundez-Zanuy, and K. López-de Ipiña. *Recent Advances in Nonlinear Speech Processing*, chapter Perceptual Features as Markers of Parkinson's Disease: The Issue of Clinical Interpretability, pages 83–91. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-28109-4_9.

[42] T. A. Mesallam, M. Farahat, K. H. Malki, M. Alsulaiman, Z. Ali, A. Al-nasheri, and G. Muhammad. Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms. *Journal of healthcare engineering*, 2017, 2017.

[43] D. Michaelis, T. Gramss, and H. W. Strube. Glottal-to-noise excitation ratio–a new measure for describing pathological voices. *Acta Acustica united with Acustica*, 83(4):700–706, 1997.

[44] G. Muhammad, M. F. Alhamid, M. S. Hossain, A. S. Almogren, and A. V. Vasilakos. Enhanced living by assessing voice pathology using a co-occurrence matrix. *Sensors*, 17(2):267, 2017.

[45] G. Muhammad, M. Alsulaiman, Z. Ali, T. A. Mesallam, M. Farahat, K. H. Malki, A. Al-nasheri, and M. A. Bencherif. Voice pathology detection using interlaced derivative pattern on glottal source excitation. *Biomedical Signal Processing and Control*, 31:156–164, 2017.

[46] K. P. Murphy. Naive bayes classifiers. *University of British Columbia*, 2006.

[47] J. Oates. Auditory-perceptual evaluation of disordered voice quality. *Folia Phoniatrica et Logopaedica*, 61(1):49–56, 2009.

[48] V. Parsa and D. G. Jamieson. Identification of pathological voices using glottal noise measures. *J Speech Lang. Hear. Res.*, 23(2):469–85, 2003.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[51] D. Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832, 2015.

[52] B. Sabir, F. Rouda, Y. Khazri, B. Touri, and M. Moussetad. Improved algorithm for pathological and normal voices identification. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(1):238–243, 2017.

[53] J. C. Saldanha, T. Ananthakrishna, and R. Pinto. Vocal fold pathology assessment using mel-frequency cepstral coefficients and linear predictive cepstral coefficients features. *Journal of medical imaging and health informatics*, 4(2):168–173, 2014.

[54] R. J. Schalkoff. *Artificial neural networks*, volume 1. McGraw-Hill New York, 1997.

[55] P. Song. Assessment of vocal cord function and voice disorders. In *Principles and Practice of Interventional Pulmonology*, pages 137–149. Springer, 2013.

[56] N. Souissi and A. Cherif. Dimensionality reduction for voice disorders identification system based on mel frequency cepstral coefficients and support vector machine. In *Modelling, Identification and Control (ICMIC), 2015 7th International Conference on*, pages 1–6. IEEE, 2015.

[57] N. Souissi and A. Cherif. Speech recognition system based on short-term cepstral parameters, feature reduction method and artificial neural networks. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, pages 667–671. IEEE, 2016.

[58] E. T. Stathopoulos, J. E. Huber, and J. E. Sussman. Changes in acoustic characteristics of the voice across the life span: measures from individuals 4–93 years of age. *Journal of Speech, Language, and Hearing Research*, 54(4):1011–1021, 2011.

[59] H. Teager. Some observations on oral air flow during phonation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(5):599–601, Oct. 1980.

[60] I. R. Titze. *Principles of voice production*. Englewood Cliffs, N.J, 1994.

[61] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig. Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average Parkinson's disease symptom severity. *J. R. Soc. Interface*, 8(59):842–855, 2010.

[62] V. Uloza, A. Vegiene, and V. Saferis. Correlation between the quantitative video laryngostroboscopic measurements and parameters of multidimensional voice assessment. *Biomedical Signal Processing and Control*, 17(Supplement C):3–10, 2015.

[63] B. Woldert-Jokisz. Saarbruecken voice database. 2007.

[64] L. Wyse. Audio spectrogram representations for processing with convolutional neural networks. *arXiv preprint arXiv:1706.09559*, 2017.

III On Orthogonal Projections for Dimension Reduction and Applications in Augmented Target Loss Functions for Learning Problems

Outline

Bibliographic information

A. Breger, J. I. Orlando, P. Harar, M. Dörfler, S. Klimscha, C. Grechenig, B. S. Gerendas, U. Schmidt-Erfurth, and M. Ehler. On orthogonal projections for dimension reduction and applications in augmented target loss functions for learning problems. *Journal of Mathematical Imaging and Vision*, in press. `arXiv:1901.07598`

Author's contribution

The author prepared the data, designed and performed the analysis described in section Application to musical data, and wrote a report based on which the section was written.

Copyright Notice

Abstract

The use of orthogonal projections on high-dimensional input and target data in learning frameworks is studied. First, we investigate the relations between two standard objectives in dimension reduction, preservation of variance and of pairwise relative distances. Investigations of their asymptotic correlation as well as numerical experiments show that a projection does usually not satisfy both objectives at once. In a standard classification problem we determine projections on the input data that balance the objectives and compare subsequent results. Next, we extend our application of orthogonal projections to deep learning tasks and introduce a general framework of augmented target loss functions. These loss functions integrate additional information via transformations and projections of the target data. In two supervised learning problems, clinical image segmentation and music information classification, the application of our proposed augmented target loss functions increase the accuracy.

Acknowledgement

This work was partially funded by the Vienna Science and Technology Fund (WWTF) through project VRG12-009, by WWTF AugUniWien/FA746A0249, by International Mobility of Researchers (CZ.02.2.69/0.0/0.0/16 027/0008371), and by project LO1401. For the research, infrastructure of the SIX Center was used.

III.1 Introduction

Linear dimension reduction is commonly used for preprocessing of high-dimensional data in complicated learning frameworks to compress and weight important data features. In contrast to nonlinear approaches, the use of orthogonal projections is computationally cheap, since it corresponds to a simple matrix multiplication. Conventional approaches apply specific projections that preserve essential information and complexity within a more compact representation. The projector is usually selected by optimizing distinct objectives, such as information preservation of the sample variance or of pairwise relative distances. Widely used orthogonal projections for dimension reduction are variants of the principal component analysis (PCA) that maximize the variance of the projected data, [39]. Preservation of relative pairwise distances asks for a near-isometric embedding, and random projections guarantee this embeddings with high probability, cf. [14, 6] and see also [1, 38, 5, 12, 32, 29]. The use of random projections is especially favorable for large, high-dimensional data ([50]), since the computational complexity is just $O(dkm)$, e.g. using the construction in [1], with $d, k \in \mathbb{N}$ being the original and lower dimensions and $m \in \mathbb{N}$ the number of samples. In contrast, PCA needs $O(d^2m) + O(d^3)$ operations ([26]). Moreover, tasks that do not have all data available at once, e.g. data streaming, ask for dimension reduction methods that are independent of the data.

In the present manuscript, we study orthogonal projections regarding the interplay between

O1) preservation of variance,

O2) preservation of pairwise relative distances,

aiming for a sufficient lower-dimensional data representation. We shall consider the Euclidean distance exclusively since it is most widely used in applications, especially for error estimation. On manifolds, the geodesic distance is locally equivalent to the Euclidean distance. The two objectives O1) and O2) are directly addressed by PCA (O1) and random projections (O2). We achieve the following goals: first we clarify mathematically and numerically that the two objectives are competing, i.e. PCA and random projections preserve different kinds of information. Depending on the objectives we discuss beneficial choices of orthogonal projections and numerically find a balancing projector for a given data set. Finally, we define a general framework of augmented target (AT) loss functions for deep neural networks, that integrate information about target characteristics via features and projections. We observe that our proposed methodology can increase the accuracy in two deep learning problems.

In contrast to conventional approaches we study the joint behavior of the two objectives with respect to the entire set of orthogonal projectors. By analyzing

the correlation between the variance and pairwise relative distances of projected data, we observe that O1) and O2) are competing and usually cannot be reached at the same time. In numerical learning experiments we investigate heuristic choices of projections applied to input features, for subsequent classification with support vector machine and shallow neural networks.

In view of learning frameworks, we utilize features and projections on target data. The class of augmented target loss functions incorporates suitable transformations and projections that provide beneficial representations of the target space. It is applied in two supervised deep learning problems dealing with real world data.

The first experiment is a clinical image segmentation problem in optical coherence tomography (OCT) data of the human retina. Related principles of dimension reduction for other clinical classification problems in OCT have already been successfully applied in [9]. In the second experiment we aim to categorize musical instruments based on their spectrogram, see [18] for related results. Our utilized augmented target loss functions can increase the accuracy in both experiments.

The outline is as follows. In Section III.2 we address the analysis of the competing objectives and Theorem III.2.5 yields the asymptotic correlation between variance and pairwise relative distances of projected data. Section III.3 prepares for the numerical investigations by recalling t-designs as considered in [10], enabling subsequent numerics. Heuristic investigations on projected input used in a straightforward classification task, are presented in Section III.4. Our framework of augmented target loss functions as modified standard loss functions for deep learning, is introduced in Section III.5. Finally, in Sections III.6 and III.7 we present classification experiments on OCT images and musical instruments using aligned augmented target loss functions.

III.2 Dimension reduction with orthogonal projections

To reduce the dimension of a high-dimensional data set $x = \{x_i\}_{i=1}^m \subset \mathbb{R}^d$, we map x into a lower-dimensional affine linear subspace $\bar{x} + V$, where $\bar{x} := \frac{1}{m}\sum_{i=1}^m x_i$ is the sample mean and V is a k-dimensional linear subspace of $\mathbb{R}^d$ with $k < d$. This mapping is performed by an orthogonal projector $p \in \mathcal{G}_{k,d}$, where

$$\mathcal{G}_{k,d} := \{p \in \mathbb{R}^{d\times d} : p^2 = p,\ p^\top = p,\ \operatorname{rank}(p) = k\}$$

denotes the Grassmannian, so that the lower-dimensional data representation is

$$\{\bar{x} + p(x_i - \bar{x})\}_{i=1}^m \subset \bar{x} + V, \tag{III.1}$$

with $\operatorname{range}(p) = V$. A suitable choice of p within $\mathcal{G}_{k,d}$ depends on further objectives, i.e. which kind of information preservation shall be favored for subsequent analysis

tasks. In the following, we consider two objectives associated to popular choices of orthogonal projectors for dimension reduction, in particular, random projectors from $\mathcal{G}_{k,d}$ and PCA. We will first observe that the two objectives are competing, especially in high dimensions, and then discuss consequences.

III.2.1 Objective O1)

The total sample variance[1] $\operatorname{tvar}(x)$ of $x = \{x_i\}_{i=1}^m \subset \mathbb{R}^d$ is the sum of the corrected variances along each dimension,

$$\operatorname{tvar}(x) := \frac{1}{m-1} \sum_{i=1}^{m} \|x_i - \bar{x}\|^2. \tag{III.2}$$

PCA aims to construct $p \in \mathcal{G}_{k,d}$, such that the total sample variance of (III.1) is maximized among all projectors in $\mathcal{G}_{k,d}$. For other equivalent optimality criteria, we refer to [51].

The total sample variance of $px = \{px_i\}_{i=1}^m \subset V$ coincides with the one of (III.1) and satisfies

$$\operatorname{tvar}(px) \leq \operatorname{tvar}(x)$$

for all $p \in \mathcal{G}_{k,d}$. Thus, PCA achieves optimal variance preservation. The total variance (III.2) can also be expressed via pairwise absolute distances

$$\operatorname{tvar}(x) = \frac{1}{m(m-1)} \sum_{i<j} \|x_i - x_j\|^2 . \tag{III.3}$$

Equally, it holds that

$$\operatorname{tvar}(px) = \frac{1}{m(m-1)} \sum_{i<j} \|p(x_i) - p(x_j)\|^2 , \tag{III.4}$$

which reveals that PCA maximizes the sample mean of the projected pairwise absolute distances.

III.2.2 Objective O2)

In contrast to pairwise absolute distances, the Johnson-Lindenstrauss Lemma targets the global property of preservation of pairwise relative distances:

Lemma III.2.1 (Johnson-Lindenstrauss, cf. [14, 38])**.** *For any* $0 < \epsilon < 1$*, any* $k \leq d, m \in \mathbb{N}$*, with*

$$\frac{4\log(m)}{\epsilon^2/2 - \epsilon^3/3} \leq k,$$

[1] We use lower case letters for samples and upper case letters for random vectors/matrices.

and any set $\{x_i\}_{i=1}^m \subset \mathbb{R}^d$*, there is a projector* $p \in \mathcal{G}_{k,d}$ *such that*

$$(1-\epsilon)\left\|x_i - x_j\right\|^2 \leq \tfrac{d}{k}\left\|p(x_i) - p(x_j)\right\|^2 \leq (1+\epsilon)\left\|x_i - x_j\right\|^2 \tag{III.5}$$

holds for all $i < j$*.*

For small $\epsilon > 0$, the projector p in Lemma III.2.1 yields that all of the $\frac{m(m-1)}{2}$ pairwise relative distances

$$\left\{\frac{d}{k}\frac{\|p(x_i) - p(x_j)\|^2}{\|x_i - x_j\|^2} : i < j\right\} \tag{III.6}$$

are close to 1, i.e. the projection p preserves all scaled pairwise relative distances well. A good choice of p in Lemma III.2.1 is based on random projectors $P \sim \lambda_{k,d}$, where $\lambda_{k,d}$ denotes the unique orthogonally invariant probability measure on $\mathcal{G}_{k,d}$. The following Theorem is essentially proved by following the lines of the proof of Lemma III.2.1 in [14] after replacing the constant 4 with $(2+\tau)2$ in the respective bound on k.

Theorem III.2.2. *For any* $0 < \epsilon < 1$*, any* $k \leq d, m \in \mathbb{N}$ *and any* $0 < \tau$ *with*

$$\frac{(2+\tau)2\log(m)}{\epsilon^2/2 - \epsilon^3/3} \leq k,$$

and any set $\{x_i\}_{i=1}^m \subset \mathbb{R}^d$*, the random projector* $P \sim \lambda_{k,d}$ *satisfies*

$$\left\{\frac{d}{k}\frac{\|P(x_i) - P(x_j)\|^2}{\|x_i - x_j\|^2} : i < j\right\} \in [1-\epsilon, 1+\epsilon] \tag{III.7}$$

with probability at least $1 - \frac{1}{m^\tau} + \frac{1}{m^{\tau+1}}$*.*

III.2.3 Competing objectives

A projector p satisfying the near-isometry property (III.5) implies

$$(1-\epsilon)\tfrac{k}{d}\operatorname{tvar}(x) \leq \operatorname{tvar}(px) \leq (1+\epsilon)\tfrac{k}{d}\operatorname{tvar}(x),$$

so that the total variance of the projected data px may not be maximized for $k < d$. In particular, with high probability a random projector $P \sim \lambda_{k,d}$ does not suit the objective of maximizing the total variance, and we even observe $\mathbb{E}\operatorname{tvar}(Px) = \frac{k}{d}\operatorname{tvar}(x)$, see (A.2) in the appendix. PCA does not guarantee any local geometric property and distances between pairs of points can be arbitrarily distorted [1], see [41] for more robust PCA. The preservation of larger distances is favored since PCA maximizes (III.4) among all $p \in \mathcal{G}_{k,d}$ and $\|p(x_i) - p(x_j)\| \leq \|x_i - x_j\|$ holds for all $i < j$. Close but distinct points, could even be projected onto a single point, which violates the preservation of pairwise relative distances, see Figure III.1.

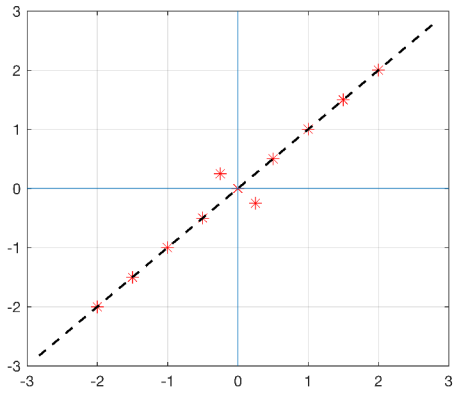


Fig. III.1: A trivial example of PCA distorting smaller distances. Choosing the first principal component, PCA projects the two dimensional data points * onto the plane of the first eigendirection (- -). The Euclidian distances of the points lying on the diagonal are preserved, whereas the two points with smaller distances are projected onto a single point (the origin).

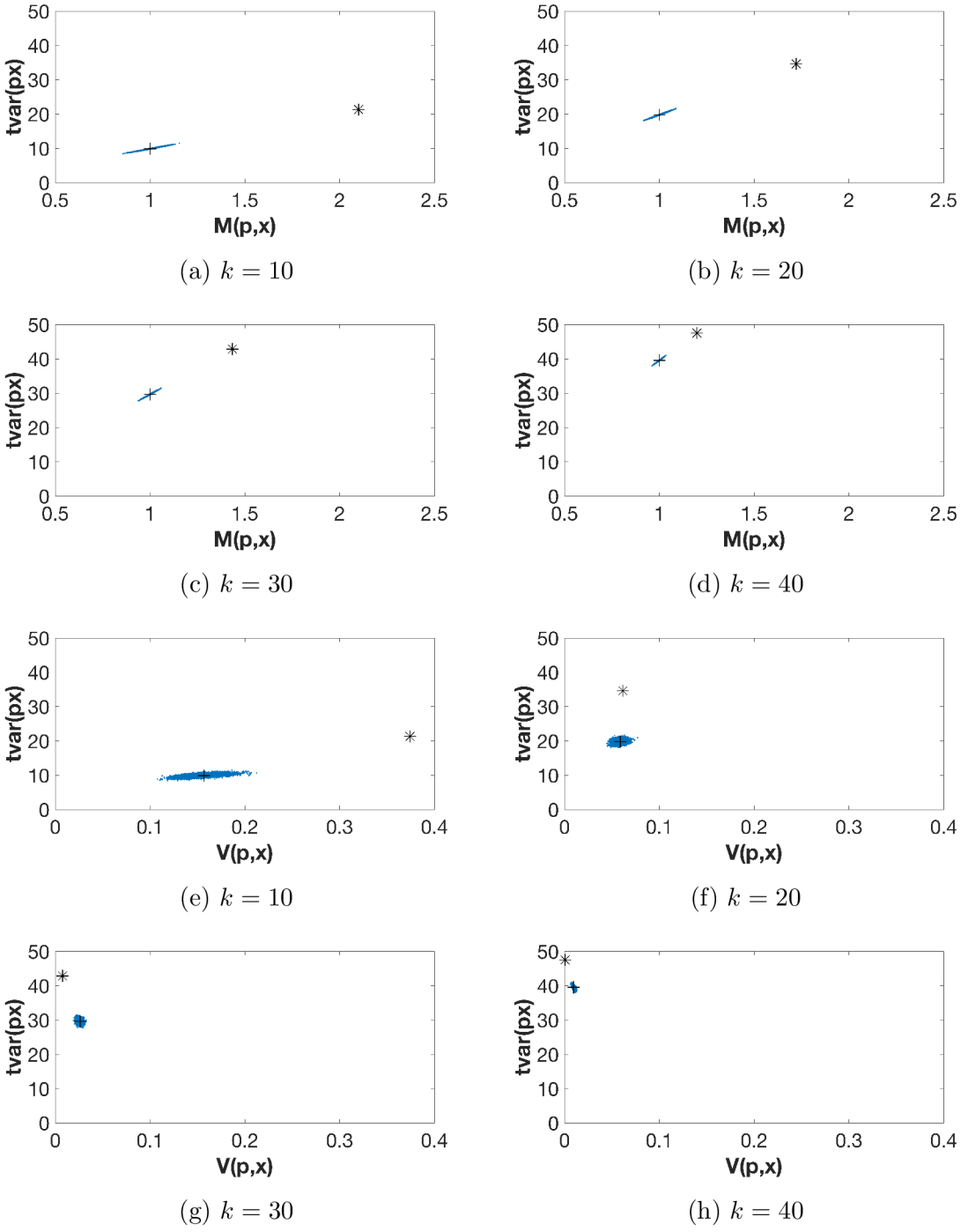
To more quantitatively understand the relation between the two competing objectives, we consider the sample mean and the uncorrected sample variance of the pairwise relative distances (III.6),

$$\mathcal{M}(p,x) := \frac{2}{m(m-1)} \sum_{i<j} \frac{d}{k} \frac{\|p(x_i - x_j)\|^2}{\|x_i - x_j\|^2}, \tag{III.8}$$

$$\mathcal{V}(p,x) := \frac{2}{m(m-1)} \sum_{i<j} \frac{d^2}{k^2} \frac{\|p(x_i - x_j)\|^4}{\|x_i - x_j\|^4} - \mathcal{M}(p,x)^2. \tag{III.9}$$

Recall that good preservation of the relative pairwise distances in (III.6) asks for $\mathcal{M}(p,x)$ being close to 1 and the variance $\mathcal{V}(p,x)$ being small. In the following, we analyze $\mathrm{tvar}(px)$, $\mathcal{M}(p,x)$, and $\mathcal{V}(p,x)$ and their expectations for random $P \in \mathcal{G}_{k,d}$.

In Figure III.2 we see a simple numerical experiment, where we first create an independent, normally distributed fixed data set $\{x_i\}_{i=1}^m$ with $x_i \in \mathbb{R}^d$ for $i = 1, \ldots, m$ and $m = 100$, $d = 50$. We then compute PCA, for $k = 10, 20, 30, 40$, as well as $n = 10000$ random projections p distributed according to $\lambda_{k,50}$. In Figure III.2 (a) - (d) we can see that the more k differs from d, the more PCA and random projections differ concerning $\mathrm{tvar}(px)$ and $\mathcal{M}(p,x)$. Those differences may lead diverse behavior in subsequent data analysis. Moreover, we compare $\mathcal{M}(p,x)$ and $\mathcal{V}(p,x)$ in Figure III.2 (e) - (h) for the different k. We can see that again when k is much smaller than d, random projections and PCA differ more concerning the variance of pairwise distances $\mathcal{V}(p,x)$. For $k = 10$ the variance for PCA is higher in comparison to random projections, see Figure III.2 (e), for $k = 40$ vice versa, see Figure III.2 (h). Note that the theoretical bounds stated in Theorem III.2.1 are



(a) $k = 10$ (b) $k = 20$

(c) $k = 30$ (d) $k = 40$

(e) $k = 10$ (f) $k = 20$

(g) $k = 30$ (h) $k = 40$

Fig. III.2: Competing properties: 10000 random projections $p \sim \lambda_{k,50}$ versus PCA ($*$), plotted concerning tvar(px), $\mathcal{M}(p, x)$ and $\mathcal{V}(p, x)$. The normal distributed fixed data set x has total variance tvar$(x) = 49.5$. Random projections cluster around their expectation values (III.10), (III.11) and (III.12), marked by $+$.

much higher than the dimensions k used in the experiments, but the projections still preserve relative pairwise distances very well. In [7] similar observations were made on empiric experiments with image and text data.

The amount of variance kept in the principal components comparing real world and random data has been experimentally studied, e.g. in [31] and [48]. Both studies determine that the difference occurs mainly in the first principal component.

Remark III.2.3. *In the numerical example we compare random projections and PCA directly, serving as the corresponding projections to the objectives O1) and O2). We observe that even for not so high dimensional ($d = 50$) data x and $k \ll d/2$, PCA severely looses information in terms of total variance, i.e. more than 50% for $k = 10$, and more importantly, looses much more information on pairwise relative distances than random projections. If both types of information are of interest, pairwise relative distances and high total variance, one should therefore favor random projections over PCA for $k \ll d/2$ to balance the two objectives O1) and O2) and vice versa. Note that with a large amount of data one might still want to favor random projectors since their construction is computationally much cheaper and independent from the data. On the other hand, if objective O2) is negligible, e.g. tasks with very noisy data, then PCA would be the favorable choice for all k.*

Information of data can be quantified and expressed in different ways. One crucial part in dimension reduction is the decision of what kind of information shall be kept, which depends on several parameters including the quality of the data and the analysis task. Variants of PCA, focusing on the preservation of variance, have been widely used in real world problems with big success, especially in denoising, when the preservation of all pairwise relative distances may be counterproductive, e.g. in dMRI imaging [53] and color filter array images [57]. Drawbacks are the necessity for all data being available from the start and the high computational costs. For very high dimensional and large data sets the computation of PCA is often not feasible. Besides the huge benefit of data independence and low computational cost when using random projections, the near-isometry property often allows to establish that the solution found in the low-dimensional space is a good approximation to the solution in the original space ([1], [37]).

Algorithms in machine learning often need or benefit from sufficient estimates of pairwise distances, e.g. approximate nearest-neighbor problems, supervised classification [29] and subspace clustering [28]. In [35] algorithmic applications of near-isometry embeddings have been introduced. In [7] random projections have been successfully applied to noisy and noiseless text and image data. The experimental studies include the comparison of preservation of pairwise distances between random projections and PCA. The results coincide with our observations, that for $k > d/2$

PCA is able to preserve the pairwise distances sufficiently, whereas for $k < d/2$ PCA distorts them. The smaller k the worse the distortion, whereas random projections preserve similarities still well for very small k, while being computationally much cheaper than PCA. One should point out again that favoring preservation of pairwise distances relies on the accuracy of the original distances.

PCA and random projections are orthogonal projections favoring two different aims. We want to study in the context of the whole set of orthogonal projections if the two objectives O1) and O2) could be reached at the same time. We will see that the objectives act competing and therefore we suggest a balancing projector for tasks that benefit from both objectives.

III.2.4 Covariances and correlation between competing objectives

For further mathematical analysis we first introduce a more general class of probability measures on $\mathcal{G}_{k,d}$ that resemble $\lambda_{k,d}$ sufficiently well:

Definition III.2.4. *A Borel probability measure λ on $\mathcal{G}_{k,d}$ is called a* cubature measure of strength t *if*

$$\int_{\mathcal{G}_{k,d}} f(p)\mathrm{d}\lambda_{k,d}(p) = \int_{\mathcal{G}_{k,d}} f(p)\mathrm{d}\lambda(p), \quad \textit{for all } f \in \mathrm{Pol}_t(\mathbb{R}^{d^2}),$$

where $\mathrm{Pol}_t(\mathbb{R}^{d^2})$ denotes the set of multivariate polynomials of total degree t in d^2 variables.

Existence of cubature measures is studied, for instance, in [16]. For random P, we now determine the expectation values for our 3 quantities of interest: $\mathrm{tvar}(Px)$, $\mathcal{M}(P,x)$, and $\mathcal{V}(P,x)$. If $P \sim \lambda$ and λ is a cubature measure of strength at least 2, the identities (A.2) and (A.3) in the appendix and a short calculation yield

$$\mathbb{E}\,\mathrm{tvar}(Px) = \tfrac{k}{d}\,\mathrm{tvar}(x), \tag{III.10}$$

$$\mathbb{E}\mathcal{M}(P,x) = 1, \tag{III.11}$$

$$\mathbb{E}\mathcal{V}(P,x) = a_{k,d}\Big(1 - \tfrac{4}{m^2(m-1)^2}\sum_{\substack{i<j\\ l<r}}\big\langle \tfrac{x_i-x_j}{\|x_i-x_j\|}, \tfrac{x_l-x_r}{\|x_l-x_r\|}\big\rangle^2\Big), \tag{III.12}$$

where $a_{k,d} = \frac{2d(d-k)}{k(d-1)(d+2)}$. The expected sample variance in (III.12) satisfies

$$\mathbb{E}\mathcal{V}(P,x) \le a_{k,d} \longrightarrow \frac{2}{k}, \quad \text{for} \quad d \to \infty.$$

This asymptotic bound relates to Theorem III.2.2 and alludes to a near-isometry property of the type (III.7) for k sufficiently large.

The following Theorem III.2.5 provides a lower bound for random P on the population correlation

$$\mathrm{Corr}(\mathcal{M}(P,x),\mathrm{tvar}(Px)) = \frac{\mathrm{Cov}(\mathcal{M}(P,x),\mathrm{tvar}(Px))}{\sqrt{\mathrm{Var}(\mathcal{M}(P,x))}\sqrt{\mathrm{Var}(\mathrm{tvar}(Px))}}. \tag{III.13}$$

It holds for arbitrary dimensions d and subsequently specifies the asymptotic behavior for $d \to \infty$:

Theorem III.2.5. *Let $x = \{x_i\}_{i=1}^m \subset \mathbb{R}^d$ be pairwise different and let $P \sim \lambda$, with λ being a cubature measure of strength at least 2. For $d \geq \frac{m(m-1)}{2}$, the correlation* (III.13) *is bounded from below by*

$$\frac{\min_{i\neq j}\|x_i-x_j\|^2}{\max_{i\neq j}\|x_i-x_j\|^2} - \frac{m(m-1)}{2d} \cdot \frac{\max_{i\neq j}\|x_i-x_j\|^2}{\min_{i\neq j}\|x_i-x_j\|^2}. \tag{III.14}$$

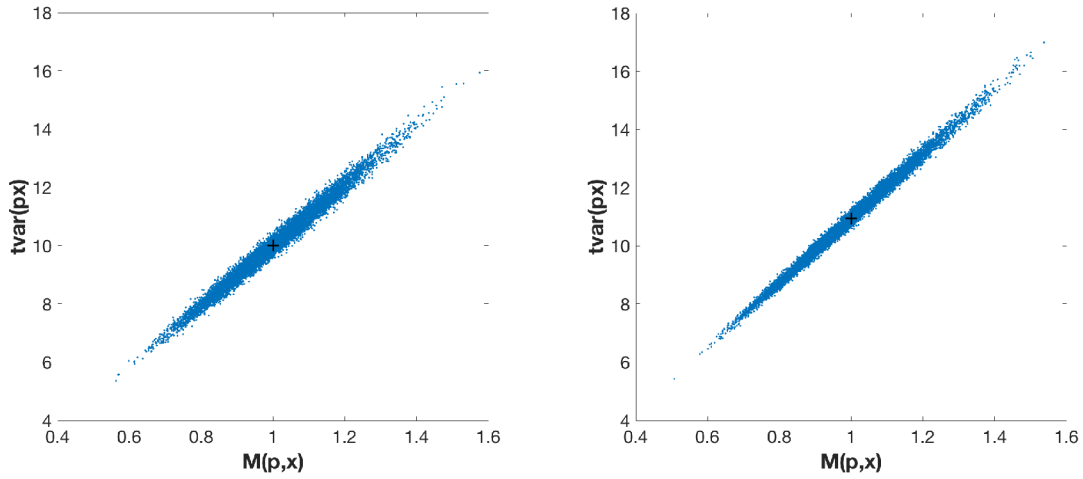
Let $\{x_i\}_{i=1}^m \subset \mathbb{R}^d$ be random points, whose entries are independent, identically distributed with finite 4-th moments, that are uniformly bounded in d. Then (III.14) *converges towards* 1 *in probability for $d \to \infty$.*

The strong correlation for large dimensions d in the second part of Theorem III.2.5 suggests that increasing $\mathrm{tvar}(Px)$ may also lead to increasing $\mathcal{M}(P,x)$, see Figure III.3 for illustration. Thus, large projected total variance $\mathrm{tvar}(Px)$ and the preservation of scaled pairwise distances, i.e. $\mathcal{M}(P,x)$ being close to 1, are competing properties. As discussed in Section III.2.3, the choice of which kind of information is favorable to preserve, depends on the data and the task; e.g. denoising (O1) and nearest neighbor classification (O2). PCA and random projections are extreme in preserving either O1) or O2). We will heuristically study the behavior of orthogonal projections balancing both objectives in the next section and will state a numerical experiment where a balancing projector yields highest classification accuracy.

Remark III.2.6. *The second part of Theorem III.2.5 relates to the well-known fact that random vectors in high dimensions are almost orthogonal, [4], and standard concentration of measure arguments may lead to more quantitative statements, cf. [54].*

III.3 Preparations for numerical experiments

For the numerical experiments we need finite sets of projectors that represent the overall space well, i.e. cover $\mathcal{G}_{k,d}$ properly.



(a) $d = 50$ (b) $d = 100$

(c) $d = 200$ (d) $d = 500$

Fig. III.3: For $x = \{x_i\}_{i=1}^{10} \subset \mathbb{R}^d$ with independent, normal distributed entries, we independently sample 10000 random projectors p from $\lambda_{10,d}$ and plot $\mathcal{M}(p, x)$ versus $\mathrm{tvar}(px)$. The expectation values with respect to $P \sim \lambda$ are marked with $+$. The correlation is already 0.9916 for $d = 50$ and grows further when d increases, namely with values $0.9961, 0.9985, 0.9996$ for $d = 100, 200, 500$.

III.3.1 Optimal covering sequences

Let the *covering radius* of a set $\{p_l\}_{l=1}^n \subset \mathcal{G}_{k,d}$ be denoted by

$$\varrho(\{p_l\}_{l=1}^n) := \sup_{p \in \mathcal{G}_{k,d}} \min_{1 \le l \le n} \|p - p_l\|_{\mathrm{F}}, \tag{III.15}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. The smaller the covering radius, the better the set $\{p_l\}_{l=1}^n$ represents the entire space $\mathcal{G}_{k,d}$. I.e., there are smaller holes and the points $\{p_l\}_{l=1}^n$ are better distributed within $\mathcal{G}_{k,d}$. Following Lemma III.2.1 we can connect finite sets of projections and their covering radius to the near-isometry property:

Lemma III.3.1. *Let $\{p_l\}_{l=1}^n \subset \mathcal{G}_{k,d}$ and denote $\varrho := \varrho(\{p_l\}_{l=1}^n)$. For any $0 < \epsilon < 1$, any $m, k, d \in \mathbb{N}$ with*

$$\frac{4\log(m)}{\epsilon^2/2 - \epsilon^3/3} \le k \le d,$$

and any $\{x_i\}_{i=1}^m \subset \mathbb{R}^d$, there is $l_0 \in \{1, \dots, n\}$ such that

$$(1-\delta)\|x_i - x_j\|^2 \le \tfrac{d}{k}\|p_{l_0}(x_i) - p_{l_0}(x_j)\|^2 \le (1+\delta)\|x_i - x_j\|^2, i < j, \tag{III.16}$$

where $\delta = \epsilon + 2\varrho\sqrt{\frac{(1+\epsilon)d}{k}} + \frac{d}{k}\varrho^2$.

Proof. Given an arbitrary projector $p \in \mathcal{G}_{k,d}$, there is an index $l_0 \in \{1, \dots, n\}$ such that

$$\|p_{l_0}x - px\| \le \|p_{l_0} - p\|_{\mathrm{F}}\|x\| \le \varrho\|x\|, \quad x \in \mathbb{R}^d.$$

From here, standard computations imply Lemma III.3.1. We omit the details. ■

The accuracy of the near-isometry property in (III.16) depends on the covering radius. Therefore, a set $\{p_l\}_{l=1}^n \in \mathcal{G}_{k,d}$ with a small covering radius ϱ is more likely to contain a projector with better preservation of pairwise relative distances. According to [11], it holds that [1] $\varrho \gtrsim n^{-\frac{1}{k(d-k)}}$, and we shall see next, how to achieve this lower bound.

A set of projectors $\{p_l\}_{l=1}^n \subset \mathcal{G}_{k,d}$ is called a *t-design* if the associated normalized atomic measure $\frac{1}{n}\sum_{l=1}^n \delta_{p_l}$ is a cubature measure of strength t (see Definition III.2.4), see [46] for general existence results. Any sequence of t_i-designs $\{p_l^i\}_{l=1}^{n_i} \subset \mathcal{G}_{k,d}$ with $t_i \to \infty$ satisfies

$$\varrho_i \asymp t_i^{-1}, \tag{III.17}$$

[1] We use the symbols $\lesssim$ and $\gtrsim$ to indicate that the corresponding inequalities hold up to a positive constant factor on the respective right-hand side. The notation $\asymp$ means that both relations $\lesssim$ and $\gtrsim$ hold.

and moreover, the bound $n_i \gtrsim t_i^{k(d-k)}$ holds, cf. [16, 11]. To relate n_i with ϱ_i via t_i, a sequence of t_i-designs $\{p_l^i\}_{l=1}^{n_i} \subset \mathcal{G}_{k,d}$ is called a *low-cardinality design sequence* if $t_i \to \infty$ and

$$n_i \asymp t_i^{k(d-k)}, \quad i = 1, 2, \dots \tag{III.18}$$

For their existence and numerical constructions, we refer to [22] and [10, 11]. According to [11], see also (III.17) and (III.18), any low-cardinality design sequence $\{p_l^i\}_{l=1}^{n_i}$ covers asymptotically optimal, i.e.,

$$\varrho_i \asymp n_i^{-\frac{1}{k(d-k)}}.$$

Benefiting from the covering property, we will use low-cardinality design sequences as a representation of the overall space of orthogonal projectors $\mathcal{G}_{k,d}$.

III.3.2 Linear least squares fit

With the linear least squares fit we can directly gain information about the relation between $\mathcal{M}(p, x)$ and $\operatorname{tvar}(px)$ for a given data set $x = \{x_i\}_{i=1}^m \subset \mathbb{R}^d$ when p varies. Given the two samples

$$\{\operatorname{tvar}(p_1 x), \dots, \operatorname{tvar}(p_n x)\}, \quad \{\mathcal{M}(p_1, x), \dots, \mathcal{M}(p_n, x)\}, \tag{III.19}$$

the linear least squares fitting provides the best fitting straight line,

$$\operatorname{tvar}(p_l x) \approx s \cdot \mathcal{M}(p_l, x) + \gamma, \quad l = 1, \dots, n,$$

where s and γ are determined by the sample variances and the sample covariance. If $\{p_l\}_{l=1}^n$ is a 2-design, then the sample (co)variances coincide with the respective population (co)variances for $P \sim \lambda_{k,d}$, see Appendix A.3 for further details. It follows that

$$s = \frac{\operatorname{Cov}(\mathcal{M}(P, x), \operatorname{tvar}(Px))}{\operatorname{Var}(\mathcal{M}(P, x))} \quad \text{with } P \sim \lambda_{k,d}, \tag{III.20}$$

$$\gamma = \tfrac{k}{d} \operatorname{tvar}(x) - s. \tag{III.21}$$

The quantities s and γ can be directly computed, where $\operatorname{tvar}(x)$ is given by (III.2) and the covariances are stated in Corollary A.1. Note that (III.20) and (III.21) are now independent of the particular choice of $\{p_l\}_{l=1}^n$.

The correlation between the two samples (III.19) yields additional information about their relation. As before, if $\{p_l\}_{l=1}^n$ is a 2-design, then the sample correlation coincides with the population correlation (III.13) for $P \sim \lambda_{k,d}$, cf. Appendix A.3. High correlation for a specific data set x suggests that random projections and PCA preserve competing properties, whose benefits need to be assessed for the specific subsequent task.

III.4 Numerical experiments in pattern recognition

We investigate the impact on classification accuracy when applying specific orthogonal projections to input data. The real world data chosen yields a straightforward classification task, serving as a toy example for comparing the accuracy of several projected input data in simple learning frameworks. Projectors are chosen from a t-design in view of $\mathrm{tvar}(px)$ and $\mathcal{M}(p,x)$. For all computations made in this Section the 'Neural Network' and 'Statistics and Machine Learning' toolboxes in MatlabR2017a are used.

We use the publicly available `iris` data set from the UCI Repository of Machine Learning Database suitable for supervised classification learning. It consists of 3 classes with 50 instances each, where each class refers to a type of iris plant. The instances are described by 4 features resulting in the input samples $\{x_i\}_{i=1}^{150} \subset \mathbb{R}^4$ and target samples $\{y_i\}_{i=1}^{150} \subset \{0,1\}^3$. For comparison we classify the diverse input data with support vector machine (SVM) and 3-layer neural networks (NN) with 5 and 10 hidden units (HU).

III.4.1 Choice of orthogonal projection

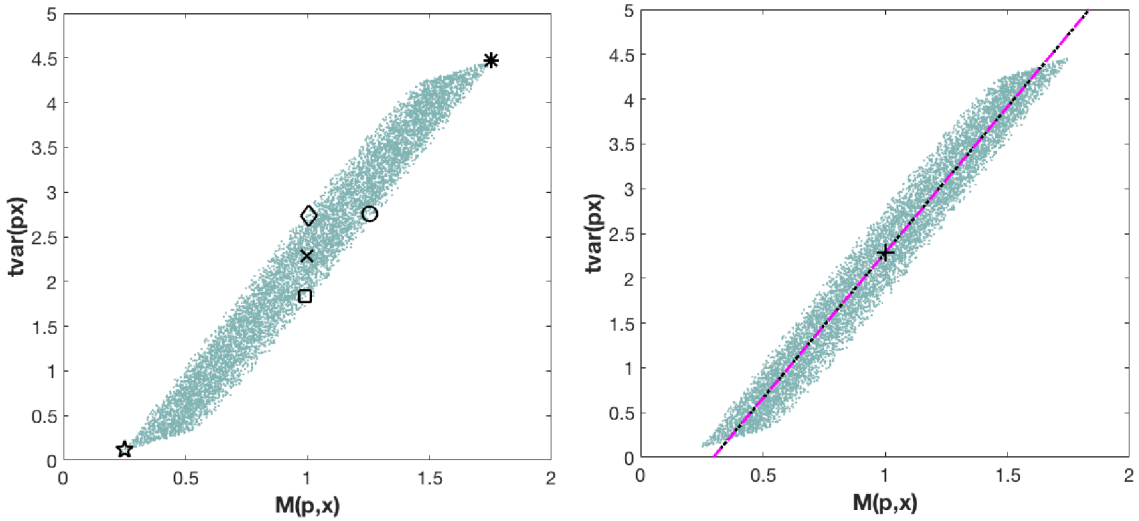
In the experiment we use projections $p \in \mathcal{G}_{2,4}$ reducing the original dimension from $d = 4$ to $k = 2$. As a finite representation of the overall space, we use a t-design of strength 14 from a low-cardinality sequence (see Section III.3.1) consisting of 8475 orthogonal projectors. Note that the dimension reduction in practice takes place by applying $q \in \mathcal{V}_{k,d}$ with $q^\top q = p \in \mathcal{G}_{k,d}$, where

$$\mathcal{V}_{k,d} := \{q \in \mathbb{R}^{k\times d} : qq^\top = I_k\}$$

denotes the Stiefel manifold. When taking norms, p and q are interchangeable, i.e., $\|q(x)\|^2 = \|p(x)\|^2$, for all $x \in \mathbb{R}^d$. Therefore we can use w.l.o.g. the theory developed for p.

The projections are chosen in a deterministic manner viewing the previously described competing properties. In Figure III.4 the three quantities $\mathrm{tvar}(px)$, $\mathcal{M}(p,x)$ and $\mathcal{V}(p,x)$ are pairwise plotted for all projectors in $\{p_l\}_{l=1}^{8475}$. For comparison we choose the following projections $p \in \{p_l\}_{l=1}^{8475} \subset \mathcal{G}_{2,4}$, see Figure III.4a for a visualization.

- $p_\times$ closest to the expected values 1 and $\frac{k}{d}\,\mathrm{tvar}(x)$ (see (III.10) and (III.11)),
- $p_\diamond$ preserving $\mathcal{M}(p,x) \approx 1$ and maximizing $\mathrm{tvar}(px)$,
- $p_\square$ preserving $\mathcal{M}(p,x) \approx 1$ and minimizing $\mathrm{tvar}(px)$,
- $p_\circ$ $\mathrm{tvar}(px) \approx \mathrm{tvar}(p_\diamond x)$ and maximizing $\mathcal{M}(p,x)$,
- $p_\star$ minimal $\mathrm{tvar}(px)$,
- p_* maximal $\mathrm{tvar}(px)$ (PCA).



(a) $\{\mathcal{M}(p_l, x), \mathrm{tvar}(p_l x)\}_{l=1}^{8475}$ (b) least squares fit

Fig. III.4: Projections $\{p_l\}_{l=1}^{8475} \subset G_{2,4}$ from a t-design of strength 14 evaluated on the iris data set $x \subset \mathbb{R}^{4\times 150}$.

III.4.2 Results

In Figure 4.III.4b we see the linear least squares fitting line, computed directly and via the slope and intercept as stated in (III.20) and (III.21). The correlation coefficient (III.13) is 0.98, which suggests that preserving the two properties is highly competing and needs to be balanced.

In Table III.1 the classification results of the iris data are presented. We can see that in this comparison the projector $p_\diamondsuit$, which corresponds to preserving $\mathcal{M}(p, x) \approx 1$ and maximizing $\mathrm{tvar}(px)$, yields the highest and most robust results. It even yields better results than working with the original input data. The projections that preserve $\mathcal{M}(p, x) \approx 1$ but do not take care of the magnitude of the total variance yield much worse results. On the other hand, the projections that just focus on high total variance still do not yield as high results as the projection $p_\diamondsuit$ that balances both properties.

Remark III.4.1. *Given a data set x, the projector $p_\diamondsuit$ is a good choice to balance both objectives O1) and O2). It can be computed by directly analyzing $\{\mathrm{tvar}(p_1 x), \ldots, \mathrm{tvar}(p_n x)\}$ and $\{\mathcal{M}(p_1, x), \ldots, \mathcal{M}(p_n, x)\}$ of a finite covering $\{p_l\}_{l=1}^{n}$ of $\mathcal{G}_{k,d}$. For higher dimensions an accurate representation of $\mathcal{G}_{k,d}$, in order to heuristically select $p_\diamondsuit$, requires large computational costs. The least squares regression line for a 2-design, as stated in III.21, can be directly computed with low computational cost. This offers helpful information about the interplay between O1) and O2).*

Input/Method	**NN (10 HU)**	**NN (5 HU)**	**SVM**
$\mathbf{x}$	(97.6, 1.25)	(97.5, 2.29)	(96.7, 0.15)
$\mathbf{p}_{\diamond}\mathbf{x}$	(98.4, 0.42)	(98.3, 2.15)	(97.3, 0.06)
$\mathbf{p}_{\times}\mathbf{x}$	(88, 1.73)	(87.9, 1.92)	(87.6, 0.63)
$\mathbf{p}_{\square}\mathbf{x}$	(87.3, 9.56)	(86.9, 10.81)	(87.7, 0.42)
$\mathbf{p}_{\circ}\mathbf{x}$	(96.8, 2.74)	(96.7, 1.77)	(96, 0.17)
$\mathbf{p}_{*}\mathbf{x}$ (PCA)	(96.9, 1.36)	(96.5, 4.07)	(96, 0.37)
$\mathbf{p}_{\star}\mathbf{x}$	(62.1, 44.30)	(58.9, 70.78)	(56, 0.61)

Table III.1: Classification results of iris data, when using projected input data in support vector machine (SVM) and shallow neural networks (NN). Mean and Variance ($\times 10^{-4}$) of 1000 independent NN runs, 100 independent runs with 10-fold cross-validation in SVM.

III.5 Augmented target loss functions

In the previous section projectors were applied to input features of shallow neural networks. In more complex architectures, such as deep neural networks, the adaption of weights can be viewed as optimization of input features, e.g. arising features can be used for transfer learning [56]. Whereas the input data is processed and optimized in each iteration, the target data stays usually unchanged during the whole learning process, serving as measure of accuracy. The representation of the target data is one key property for successful approximation with neural networks. Here, we will introduce a general class of loss functions, i.e. augmented target (AT) loss functions, that use projections and features to yield beneficial representations of the target space, emphasizing important characteristics.

In optimization problems additional penalty terms are used for regularization or to enforce other beneficial constraints. In deep learning, weight decay (i.e. Tikhonov regularization) is a standard adaption of the loss function to that effect. Incorporating additional underlying information via features of the output/target data has been studied in diverse settings tailored to particular imaging applications. Perceptual loss functions have been used in [34] for image super-resolution, incorporating the comparison of high-level image features that arise from pretrained convolutional neural networks, i.e. the VGG-network [47]. Deep perceptual similarity metrics have been proposed in [19] for generating images, comparing image features instead of the original images. In [30] a similar approach was successfully used for style transfer and super-resolution, adding a network that defines loss functions. Anatomically constrained neural networks (ACNN) have been introduced in [42] and applied to cardiac image enhancement and segmentation. Their loss functions incorporate

structural information by using autoencoders to gain features about lower dimensional parametrization of the segmentation. Brain segmentation was studied in [24], where information about the desired structure has been added in the loss function via an adjacency matrix. It was used for fine-tuning the supervised learned network with unlabeled data, reducing the number of abnormalities in the segmentation.

The information of certain target characteristics can be very powerful and even replace the need of annotations in some tasks. In [49] label-free learning is approached by using just structural information of the desired output in the loss function instead of annotated target values.

In the following, we will define a general framework of loss functions that add information of target characteristics via features and projections in supervised learning tasks.

III.5.1 General framework

Let the training data be input vectors $\{x_i\}_{i=1}^m \subset \mathbb{R}^r$ with associated target values $\{y_i\}_{i=1}^m \subset \mathbb{R}^s$. We consider training a neural network

$$f_\theta : \mathbb{R}^r \to \mathbb{R}^s,$$

where $\theta \in \mathbb{R}^N$ corresponds to the vector of all free parameters of a fixed architecture. In each optimization step for θ, the network's output $\{\hat{y}_i = f_\theta(x_i)\}_{i=1}^m \subset \mathbb{R}^s$ is compared with the targets $\{y_i\}_{i=1}^m$ via an underlying loss function L.

In contrast to ordinary learning problems with highly accurate target data, complicated learning tasks arising in many real world problems do not yield sufficient results when optimizing neural networks with standard loss functions L, such as the widely used mean least squares error

$$L_{\mathrm{MSE}}(\{y_i\}_{i=1}^m, \{\hat{y}_i\}_{i=1}^m) := \frac{1}{m} \sum_{i=1}^m \|y_i - \hat{y}_i\|^2 . \tag{III.22}$$

The training data may include important information that is obvious for humans, but poorly represented within the original target data and therefore lacks consideration in the learning process. To overcome this issue, we propose to add information tailored to the particular learning problem represented by additional features of the outputs and targets.

First, we select transformations

$$T_j : \mathbb{R}^s \to \mathbb{R}^t, \quad j = 1, \dots, d,$$

to enable error estimation in transformed output/target spaces. Note that the transformations T_j are not required to be linear. However, they should be piecewise

differentiable to enable subsequent optimization of the loss function with gradient methods. We shall allow for additional weighting of the transformations $T_1, \dots, T_d$ to facilitate the selection of features for a specific learning problem. The previous sections suggest that orthogonal projections can provide favorable feature combinations, which essentially turns into a weighting procedure.

To enable suitable projections, we stack the d output/target features

$$T(y_i) := \begin{pmatrix} T_1(y_i)^\top \\ \vdots \\ T_d(y_i)^\top \end{pmatrix} \in \mathbb{R}^{d \times t},$$

so that applying a projector $p \in \mathcal{G}_{k,d}$ to each column of $T(y_i)$ yields $p(T(y_i)) \in \mathbb{R}^{d \times t}$. We now define the augmented target loss function with projections by

$$L_p\Big(\{y_i\}, \{\hat{y}_i\}\Big) := L(\{y_i\}, \{\hat{y}_i\}) + \alpha \cdot \tilde{L}\Big(\{p(T(y_i))\}, \{p(T(\hat{y}_i))\}\Big), \tag{III.23}$$

where $\alpha > 0$ and L, $\tilde{L}$ correspond to conventional loss functions. Apparently, L_p depends on the choice of $p \in \mathcal{G}_{k,d}$. The projection $p(T(y_i))$ weighs the previously chosen feature transformations $T(y_i)$. Standard choices of L and $\tilde{L}$ are L_{MSE}, in which case L_p becomes

$$L_p\Big(\{y_i\}, \{\hat{y}_i\}\Big) = \frac{1}{m} \sum_{i=1}^{m} \|y_i - \hat{y}_i\|^2 + \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \|p(T(y_i)) - p(T(\hat{y}_i))\|_{\mathrm{F}}^2. \tag{III.24}$$

Remark III.5.1. *For $k = d$ the projector p is the identity. In this case the transformations can map into different spaces, i.e.*

$$T_j : \mathbb{R}^s \to \mathbb{R}^{t_j}, \quad j = 1, \dots, d,$$

and we can now write the standard augmented target loss function by

$$L_{AT}\Big(\{y_i\}, \{\hat{y}_i\}\Big) = \sum_{j=1}^{d} \alpha_j \cdot L^j\Big(\{T_j(y_i)\}, \{T_j(\hat{y}_i)\}\Big), \tag{III.25}$$

where T_1 corresponds to the identity function, $L^1, \dots, L^d$ are common loss functions and $\alpha_1, \dots, \alpha_d > 0$ are weighting parameters.

It should be mentioned that α resembles a regularization parameter. The actual minimization of (III.22) among θ is usually performed through Tikhonov type regularization in many standard deep neural network implementations. The formulation (III.23) adds one further variational step for beneficial output data representation.

Remark III.5.2. *Our proposed structure with target feature maps $T_1, \dots, T_d$ as in* (III.25) *relates to multi-task learning, which has been successfully used in deep neural networks [13]. It handles multiple learning problems with different outputs at the same time. In contrast to multi-task learning, we aim to solve a single problem but also penalize the error in transformed spaces enhancing certain target characteristics.*

For the projected feature transformations in the augmented target loss function it is not possible to identify a balancing projection p heuristically (such as $p_\diamond$ in Section III.4), because the output y changes in each iteration when the loss function is called. In the following clinical numerical experiment we overcome this issue by using random projections and PCA in each optimization step and compare it to prior deterministic choices of projections.

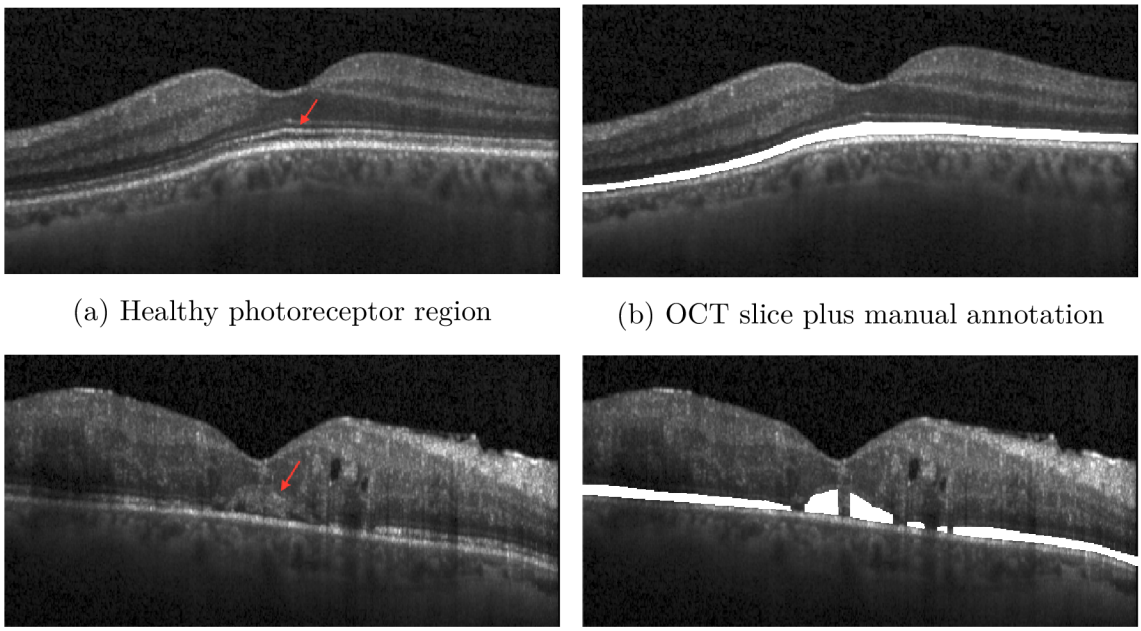
III.6 Application to clinical image data

The first experiment is a clinical problem in retinal image analysis of the human eye, where the disruptions of the so-called photoreceptor layers need to be quantified in optical coherence tomography images (OCT). The photoreceptors have been identified as the most important retinal biomarker for prediction of vision from OCT in various clinical publications, see e.g. [25]. As OCT technology advances, clinicians are not able to look at each slice of OCT themselves (in mean they get 250 slices per patient and have 3-5 minutes/patients including their clinical examination). Therefore, automated classification of e.g. photoreceptor status is necessary for clinical guidance.

III.6.1 Data and objective

In this application, OCT images of different retinal diseases (diabetic macular edema and retinal vein occlusion) were provided by the Vienna Reading Center recorded with the Spectralis OCT device (Heidelberg Engineering, Heidelberg, Germany). Each patient's OCT volume consists of 49 cross-sections/slices (496×512 pixels) recorded in an area of 6×6 mm in the center of the human retina, which is the part of the retina responsible for vision. Each of the slices was manually annotated by a trained grader of the reading center. This is a challenging and time-consuming procedure that is not feasible in clinical routine but only in a research setting. The binary pixelwise annotations serve as target values, enabling a supervised learning framework.

The objective is to accurately detect the photoreceptor layers and their disruptions pixelwise in each OCT slice by training a deep convolutional neural network with a suitable loss function. The learning problem is complicated by potentially inaccurate target annotations, as studies have shown that inconsistencies between trained graders are common, cf. [52]. Moreover, the learning task is unbalanced in the sense that there are many more slices showing none or very little disruptions. We shall observe that optimization with respect to standard loss functions performs



(a) Healthy photoreceptor region

(b) OCT slice plus manual annotation

(c) Disrupted photoreceptor region

(d) OCT slice plus manual annotation

Fig. III.5: OCT provides cross-sectional visualization of the human retina.

poorly in regards to detecting disruptions. The augmented target loss function proposed in the previous section can enhance the detection.

III.6.2 Convolutional neural network learning

We implemented our experiments using Python 3.6 with Pytorch 0.4.0. A deep convolutional neural network f_θ is trained by applying the U-Net architecture reported in [45] with a softmax activation function and Tikhonov regularization. A set of 20 OCT volumes (980 slices) from different patients with corresponding annotations are used for training, where 4 volumes were used for calibration (validation set). Another 2 independent test volumes were identified for evaluating the results, one without any disruptions in the photoreceptor layers, whereas the other one includes a high number of disruptions.

Each OCT slice is represented by a vector $x_i \in \mathbb{R}^r$ with $r = 496 \cdot 512$. The collection $\{x_i\}_{i=1}^m$ corresponds to all slices from the training volumes, i.e. $m = 20 \cdot 49$. Further matching the notation of the previous section, we have $r = s$ and $f_\theta : \mathbb{R}^r \to \mathbb{R}^r$ with binary target vectors $y_i \in \{0,1\}^r$. We observe that disruptions are not identified reliably when using the least squared loss function (III.22). To overcome this issues, we use the proposed augmented target loss function with least squared losses as stated in (III.24).

To enhance disruptions within the output/target space, we heuristically choose

$d = 4$ local features of the original representation. They are derived from Shearlet coefficients ([33]) T_1 and convolution with a gradient filter (Prewitt) T_2, a Gaussian highpass filter T_3, and a Frangi Filter ([23]) T_4, see Figure III.6. These feature transformations keep the same size, i.e. $T_j : \mathbb{R}^r \to \mathbb{R}^r$ for $j = 1, \ldots, d$.

We can derive different augmented target loss functions L_p by choosing different $p \in \mathcal{G}_{k,d}$ for (III.23). In this experiment we use the following projections:

- $p = I_4$,
- $\{p_l\}_{l=1}^{15}$, all projections from a t-design of strength $2 \subset \mathcal{G}_{2,4}$ (see [10]),
- $p_{\text{PCA}} \in \mathcal{G}_{2,4}$, projection determined by PCA for each mini-batch,
- $p_{\lambda_{2,4}}$, random projection chosen according to $\lambda_{2,4}$ in each mini-batch.

III.6.3 Results

Since the detection problem is highly unbalanced we use precision/recall curves [15] for evaluating the overall performance of each loss function model. The area under the curve (AUC) was used as a numerical indicator of the success rate, [43]. The higher the AUC the better the classification.

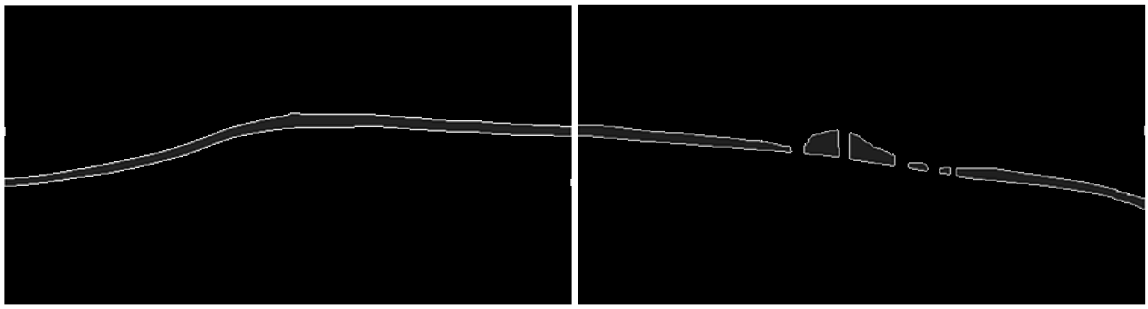
The results of the different loss functions on the independent test set are stated in Table III.2. Due to the imbalance within the data, the photoreceptor region is identified well, but disruptions are not identified reliably when using the least squared loss function (III.22). For $\alpha = 0.5$ all proposed augmented target loss functions L_p immensely increase the success rate of the disruption quantification. The highest result was achieved by using the fixed projection p_{12} from the t-design sequence $\{p_l\}_{l=1}^{15}$ on the output/target features. This corresponds to the results of the previous sections, stating that depending on the particular data there are projections in the overall space acting beneficially. Since this projection generally cannot be found beforehand, using random projections or PCA in each loss function evaluation step is easier possible in practice. Random projections yield the highest overall accuracy and also beat PCA concerning the detection of disruptions.

The choice of random projections in L_p seems beneficial. Random projections can be computed very efficiently and randomization can generalize and robusten the information, cf. [37]. In the following we will view a second classification problem based on spectrograms, where augmented target loss functions with random projections can improve the accuracy.

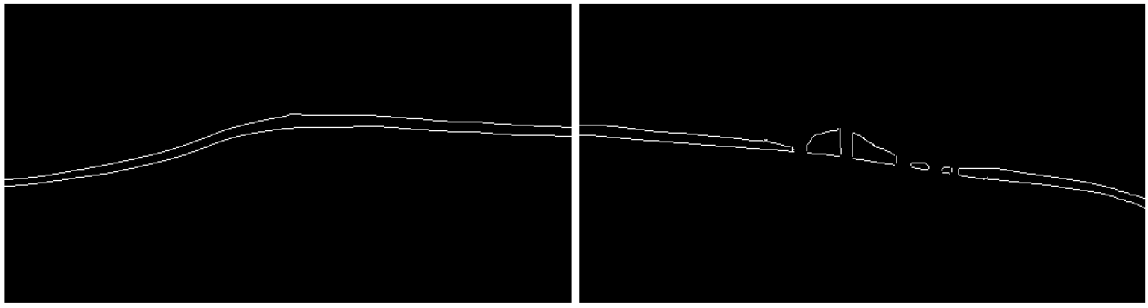
III.7 Application to musical data

Here, the learning task is a prototypical problem in Music Information Retrieval, namely multi-class classification of musical instruments. In analogy to the MNIST

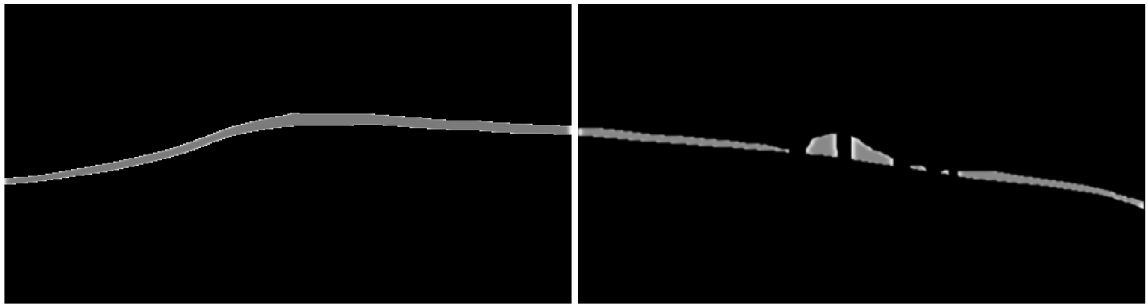
(a) Shearlet coefficients



(b) Prewitt



(c) Gaussian highpass



(d) Frangi filter

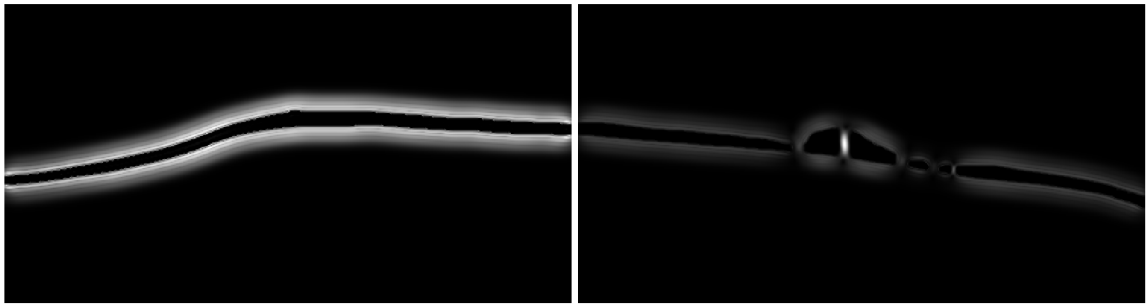


Fig. III.6: Features on output and targets that enhance edges in different ways. It is not obvious which transformations are of most importance, weighting by projections can overcome this issue.

Table III.2: Comparison of AUC values for photoreceptors segmentation and disruption detection.

Loss function	Photoreceptors	Disruptions
L_{MSE}	0.9724	0.3954
L_p		
$p = I_4$	0.9766	0.6045
$p_{\lambda_{2,4}}$	0.9796	0.5690
p_{PCA}	0.9734	0.5196
p_{12}	0.9755	0.6490

problem in image recognition, this classification problem is commonly used as a basis of comparison for innovative methods, since the ground truth is unambiguous and sufficient annotated data are available. The input to the neural network are spectrograms of audio signals, which is the standard choice in audio machine learning. Spectrograms are calculated from the time signal using a short-time Fourier transform and taking the absolute value squared of the resulting spectra, thus yielding a vector for each time-step and a two-dimensional array, like an image, cf. [17].

Reproducible code and more detailed information of our computational experiments can be found in the online repository [27].

III.7.1 Data and objective

The publicly available GoodSounds dataset [44] contains recordings of single notes and scales played by several single instruments. To gain equally balanced input classes we restrict the classification problem to 6 instruments: clarinet, flute, trumpet, violin, alto saxophone and cello. Note that the recordings are monophonic, so that each recording yields one spectrogram that we aim to correctly assign to one of the 6 instruments.

After removing the silence [3, 40], segments from the raw audio files are transformed into log mel spectrograms [20], so that we obtain images of time-frequency representations with size 100×100. One example spectrogram for each class of instruments is depicted in Figure III.7.

III.7.2 Convolutional neural network learning

We implemented a fully convolutional neural network $f_\theta : \mathbb{R}^r \to [0,1]^s$, cf. [36], where $r = 100 \times 100$ and $s = 6$, in Python 3.6 using Keras 2.2.4 framework [21]

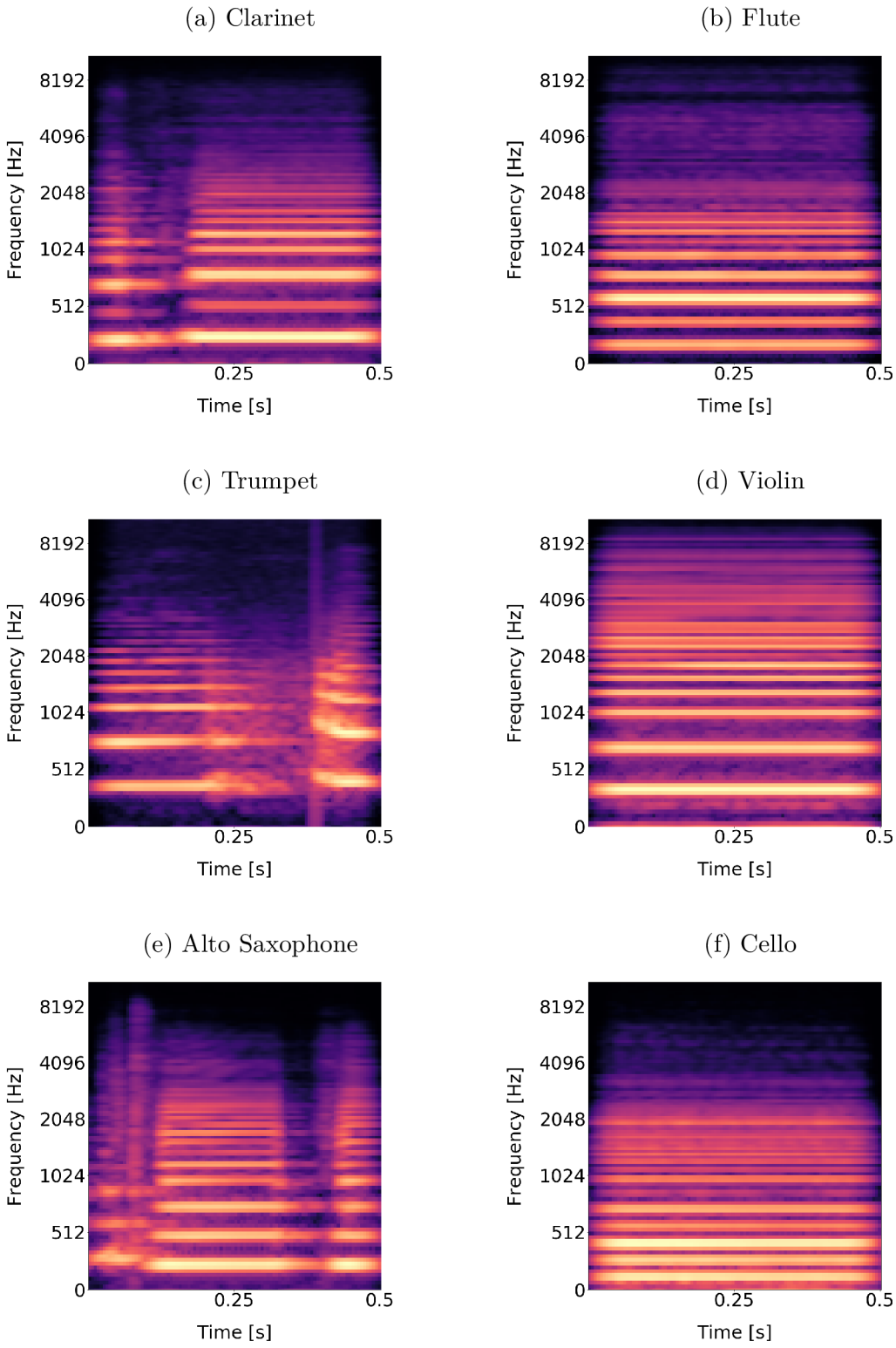


Fig. III.7: Log mel spectrograms of the 6 different instruments. Intensities range from zero (black) to 1 (yellow).

α	β	p	training	test data
0	0	-	0.5541	0.5716
0.01	0	I_{16}	0.5650	0.5683
0.01	0	$p_{\lambda_{6,16}}$	0.7722	0.7657
0	0.05	-	0.9771	0.9729
0.01	0.05	I_{16}	0.9849	0.9802
0.01	0.05	$p_{\lambda_{6,16}}$	0.9857	0.9833

Table III.3: Classification results with different parameter choices. The standard inbuilt Tikhonov regularization (ℓ_2-norm of θ) is weighted by β. For $\alpha > 0$ the feature transformations $\{T_j\}_{j=1}^{16}$ are used in the loss function, either directly or weighted by a random projection $p_{\lambda_{6,16}}$. The accuracy of the model is measured by the number of correctly classified samples divided by the number of all samples.

and trained it on the Nvidia GTX 1080 Ti GPU. The data is split into 140 722 training, 36 000 validation and 36 000 independent test samples. We heuristically choose $d = 16$ output features arising directly from the particular output class. The transformations $T_1, \ldots, T_{16}$, with $T_j : \mathbb{R}^6 \to \mathbb{R}$ for $j = 1, \ldots, 16$, are then given by the inner product of the output/target and the feature vectors. Amongst others the features are chosen from the enhanced scheme of taxonomy [55] and from the table of frequencies, harmonics and under tones [59]. We use the proposed augmented target loss function L_p (III.23), where L_1 corresponds to the categorical-cross-entropy loss [58] and L_2 to the mean squared error as in (III.24). We consider here two choices of p: the identity I_{16} and random projectors $p \sim \lambda_{6,16}$ in $\mathcal{G}_{6,16}$.

The deep learning model is sensitive to various hyper-parameters, including α and p, in addition to conventional parameters, such as the number of convolutional kernels, learning rate and the parameter β for Tikhonov regularization. To find the best choices in a fair trial we utilize a random hyper-parameter search approach, where we train 60 models and select the 3 best ones for a more precise search over different α in the augmented target loss function and β for Tikhonov regularization. This results in 212 models that are evaluated on the training and validation set. Finally, we select the best model based on the accuracy of the validation set and evaluate it on the independent test set. For comparison we also evaluate this model with no Tikhonov regularization, i.e. $\beta = 0$, see Table III.3.

III.7.3 Results

Table III.3 shows that no regularization and no features provide the poorest results. It seems that adding features with random projections have a regularizing effect and improve the results significantly. As expected, it is important to include Tikhonov regularization on θ. Further enhancement happens by adding features via the modified augmented target loss function with or without additional weighting from projections. All results are very stable and are generalizing very well from training to the independent test set, see [27] for further details.

Appendix

A Proof of Theorem III.2.5

A.1 Proof of (III.14) in Theorem III.2.5

For $\{y_i\}_{i=1}^M \subset \mathbb{R}^d$ and $p \in \mathcal{G}_{k,d}$, we define

$$f(p, \{y_i\}_{i=1}^M) := \tfrac{1}{M} \sum_{i=1}^{M} \tfrac{d}{k} \|p(y_i)\|^2. \tag{A.1}$$

Given two sets, $\{y_i\}_{i=1}^{M_1}, \{z_j\}_{j=1}^{M_2} \subset \mathbb{R}^d$, suppose that $P \in \mathcal{G}_{k,d}$ is a random matrix, distributed according to a cubature measure of strength at least 2. The covariance is given by

$$\begin{aligned}\operatorname{Cov}(f(P, \{y_i\}_{i=1}^{M_1}), f(P, \{z_j\}_{j=1}^{M_2})) = \\ \mathbb{E}[(f(P, \{y_i\}) - \mathbb{E}[f(P, \{y_i\})])(f(P, \{z_i\}) - \mathbb{E}[f(P, \{z_i\})]\end{aligned}$$

Using the identity, cf. [2],

$$\tfrac{d}{k} \mathbb{E}\Big[\|Py\|^2 \Big] = \|y\|^2, \tag{A.2}$$

directly yields

$$\begin{aligned}\operatorname{Cov}(f(P, \{y_i\}_{i=1}^{M_1}), f(P, \{z_j\}_{j=1}^{M_2})) = \\ \mathbb{E}[(\tfrac{1}{M_1} \sum_{i=1}^{M_1} \tfrac{d}{k} \|P(y_i)\|^2 - \tfrac{1}{M_1} \sum_{i=1}^{M_1} \|y_i\|^2)(\tfrac{1}{M_2} \sum_{i=1}^{M_2} \tfrac{d}{k} \|P(z_i)\|^2 - \tfrac{1}{M_2} \sum_{i=1}^{M_2} \|z_i\|^2)]\end{aligned}$$

Following [8, Theorem 2.4, Section 3.1] we use that

$$\mathbb{E}\Big[\|Py\|^2 \|Pz\|^2 \Big] = \frac{1}{q}(\alpha_1 \|y\|^2 \|z\|^2 + \alpha_2 \langle y, z \rangle^2), \quad y, z \in \mathbb{R}^d, \tag{A.3}$$

holds, where $q = (d-1)d(d+2)$, $\alpha_1 = (d+1)k^2 - 2k$ and $\alpha_2 = 2k(d-k)$. This leads to the explicit formula of the population covariance

$$\begin{aligned}\mathrm{Cov}(f(P,\{y_i\}_{i=1}^{M_1}), f(P,\{z_j\}_{j=1}^{M_2})) = \\ \frac{a_{k,d}}{M_1 M_2}\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\langle y_i, z_j\rangle^2 - \frac{a_{k,d}}{d}\Big(\frac{1}{M_1}\sum_{i=1}^{M_1}\|y_i\|^2\Big)\Big(\frac{1}{M_2}\sum_{j=1}^{M_2}\|z_j\|^2\Big),\end{aligned} \tag{A.4}$$

with $a_{k,d} = \frac{2d(d-k)}{k(d-1)(d+2)}$.
For $y := \{y_i\}_{i=1}^M \subset \mathbb{R}^d \setminus \{0\}$ we set $\hat{y}_i := \frac{y_i}{\|y_i\|}$, for $i = 1, \ldots, M$. The identity (A.4) enables us to compute the population correlation

$$\mathrm{Corr}(f(P,y), f(P,\hat{y})) = \frac{\mathrm{Cov}(f(P,y), f(P,\hat{y}))}{\sqrt{\mathrm{Var}(f(P,y))}\sqrt{\mathrm{Var}(f(P,\hat{y}))}} \tag{A.5}$$

by the explicit formulas

$$\begin{aligned}\mathrm{Cov}\Big[f(P,y), f(P,\hat{y})\Big] &= \frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle y_i, \hat{y}_j\rangle^2 - \frac{a_{k,d}}{d}\cdot\frac{1}{M}\sum_{i=1}^{M}\|y_i\|^2 \\ \mathrm{Cov}\Big[f(P,y), f(P,y)\Big] &= \mathrm{Var}[f(P,y)] = \frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2 - \frac{a_{k,d}}{d}\Big(\frac{1}{M}\sum_{i=1}^{M}\|y_i\|^2\Big)^2 \\ \mathrm{Cov}\Big[f(P,\hat{y}), f(P,\hat{y})\Big] &= \mathrm{Var}[f(P,\hat{y})] = \frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle \hat{y}_i, \hat{y}_j\rangle^2 - \frac{a_{k,d}}{d}.\end{aligned}$$

Since the variance is always nonnegative and $\frac{a_{k,d}}{d} > 0$, the denumerator of $\mathrm{Corr}(f(P,y), f(P,\hat{y}))$ in (A.5) satisfies

$$\begin{aligned}\sqrt{\mathrm{Var}(f(P,y))}\sqrt{\mathrm{Var}(f(P,\hat{y}))} &\le \sqrt{\Big(\frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2\Big)\Big(\frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle \hat{y}_i, \hat{y}_j\rangle^2\Big)} \\ &\le \frac{a_{k,d}}{M^2}\sqrt{\Big(\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2\Big)\Big(\frac{1}{\min_i(\|y_i\|)^4}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2\Big)} \\ &\le \frac{1}{\min_i(\|y_i\|)^2}\frac{a_{k,d}}{M^2}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2.\end{aligned}$$

The enumerator of $\mathrm{Corr}(f(P,y), f(P,\hat{y}))$ in (A.5) is estimated by

$$\mathrm{Cov}(f(P,y), f(P,\hat{y})) \ge \frac{a_{k,d}}{\max_i(\|y_i\|)^2}\frac{1}{M^2}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2 - \frac{a_{k,d}}{d}\max_i(\|y_i\|)^2$$

For $d \ge M$, a short calculation yields $\mathrm{Cov}(f(P,y), f(P,\hat{y})) \ge 0$, so that we obtain

$$\mathrm{Corr}(f(P,y), f(P,\hat{y})) \ge \frac{\min_i(\|y_i\|)^2}{\max_i(\|y_i\|)^2} - \frac{\min_i(\|y_i\|)^2\max_i(\|y_i\|)^2}{\frac{d}{M^2}\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2}.$$

The lower bound $\sum_{i,j=1}^{M}\langle y_i, y_j\rangle^2 \geq M \min_i(\|y_i\|)^4$ yields

$$\mathrm{Corr}(f(P,y), f(P,\hat{y})) \geq \frac{\min_i(\|y_i\|)^2}{\max_i(\|y_i\|)^2} - \frac{M}{d}\cdot\frac{\max_i(\|y_i\|)^2}{\min_i(\|y_i\|)^2}.$$

Since the correlation is scaling invariant the choice $y = \{x_i - x_j : 1 \leq i < j \leq m\}$ with $M = \frac{m(m-1)}{2}$ implies (III.14) in Theorem III.2.5. Incorporating the correct scaling yields the following corollary:

Corollary A.1. *For a given data set $x = \{x_i\}_{i=1}^m$ and for random $P \in \mathcal{G}_{k,d}$ the (co)variances of* $\mathrm{tvar}(Px)$ (III.4) *and* $\mathcal{M}(P,x)$ (III.8) *are given by*

$$\mathrm{Cov}(\mathcal{M}(P,x), \mathrm{tvar}(Px)) = \frac{k}{2d}\Big(\frac{a_{k,d}}{M^2}\sum_{i<j}\sum_{l<r}\langle x_i - x_j, \frac{x_l - x_r}{\|x_l - x_r\|}\rangle^2 - \frac{a_{k,d}}{d}\cdot\frac{1}{M}\sum_{i<j}\|x_i - x_j\|^2\Big),$$

$$\mathrm{Var}(\mathrm{tvar}(Px)) = \frac{k^2}{4d^2}\Big(\frac{a_{k,d}}{M^2}\sum_{i<j}\sum_{l<r}\langle x_i - x_j, x_l - x_r\rangle^2 - \frac{a_{k,d}}{d}\big(\frac{1}{M}\sum_{i<j}\|x_i - x_j\|^2\big)^2\Big),$$

$$\mathrm{Var}(\mathcal{M}(P,x)) = \frac{a_{k,d}}{M^2}\sum_{i<j}\sum_{l<r}\langle\frac{x_i - x_j}{\|x_i - x_j\|}, \frac{x_l - x_r}{\|x_l - x_r\|}\rangle^2 - \frac{a_{k,d}}{d},$$

where $M = \frac{m(m-1)}{2}$ *and* $a_{k,d} = \frac{2d(d-k)}{k(d-1)(d+2)}$.

A.2 Proof of the second part of Theorem III.2.5

For fixed parameters $\mu > 0, \sigma^2 > 0$, that do not depend on d, let $Y_1 \in \mathbb{R}^d$ be a random vector, whose squared entries are independent, identically distributed with mean $\mathbb{E}Y_{1,l}^2 = \mu$ and variance $\mathrm{Var}(Y_{1,l}^2) = \sigma^2$, for $l = 1, \ldots, d$. We immediately observe

$$\mathbb{E}\Big(\frac{\|Y_1\|^2}{\sqrt{d}}\Big) = \sqrt{d}\mu, \qquad \mathrm{Var}\left(\frac{\|Y_1\|^2}{\sqrt{d}}\right) = \sigma^2.$$

For any $c > 0$, Chebychev's inequality yields

$$\mathbb{P}\Big(\Big|\frac{\|Y_1\|^2}{\sqrt{d}} - \sqrt{d}\mu\Big| \geq c\sigma\Big) \leq \frac{1}{c^2}.$$

Suppose that $Y_2, \ldots, Y_M$ are copies of Y_1, not necessarily independent. Then the union bound

$$\mathbb{P}\Big(\Big|\frac{\|Y_i\|^2}{\sqrt{d}} - \sqrt{d}\mu\Big| \geq c\sigma, \text{ for some } i = 1, \ldots, M\Big) \leq \frac{M}{c^2}$$

implies that

$$\sqrt{d}\mu - c\sigma \leq \frac{\min_i(\|Y_i\|)^2}{\sqrt{d}} \leq \frac{\max_i(\|Y_i\|)^2}{\sqrt{d}} \leq \sqrt{d}\mu + c\sigma$$

holds with probability at least $1 - \frac{M}{c^2}$. Provided that $\sqrt{d}\mu \neq c\sigma$ and $0 < \sqrt{d}\mu - c\sigma$, we deduce

$$\frac{\sqrt{d}\mu - c\sigma}{\sqrt{d}\mu + c\sigma} \leq \frac{\min_i(\|Y_i\|)^2}{\max_i(\|Y_i\|)^2} \leq \frac{\sqrt{d}\mu + c\sigma}{\sqrt{d}\mu - c\sigma}.$$

We can choose $c = \frac{\mu}{\sigma}\sqrt[4]{d}$, since $0 < c \leq \frac{\sqrt[4]{d}\mu}{\sigma} \leq \frac{\sqrt{d}\mu}{\sigma}$. That directly yields

$$\frac{1 - \frac{1}{\sqrt[4]{d}}}{1 + \frac{1}{\sqrt[4]{d}}} \leq \frac{\min_i(\|Y_i\|)^2}{\max_i(\|Y_i\|)^2} \leq \frac{1 + \frac{1}{\sqrt[4]{d}}}{1 - \frac{1}{\sqrt[4]{d}}}$$

holds with probability at least $1 - \frac{\mu^2 M}{\sigma^2\sqrt{d}}$.
It follows directly that $\frac{\min_i(\|Y_i\|)^2}{\max_i(\|Y_i\|)^2}$ converges towards 1 in probability for $d \to \infty$,
The choice $\{Y_1, \ldots, Y_M\} = \{X_i - X_j : 1 \leq i < j \leq m\}$ implies the second part of Theorem III.2.5.

A.3 Calculations for population covariances

We notice that $\|p(x_i - x_j)\|^2 = \operatorname{trace}(px_ix_i^\top - px_jx_j^\top)$ is a polynomial of degree 1 in p. Hence, $\operatorname{tvar}(px)$ in (III.4) is also a polynomial of degree 1 in p. If $\{p_l\}_{l=1}^n$ is a 1-design, then the sample mean of $\{\operatorname{tvar}(p_1x), \ldots, \operatorname{tvar}(p_nx)\}$ satisfies

$$\frac{1}{n}\sum_{l=1}^{n} \operatorname{tvar}(p_lx) = \mathbb{E}\operatorname{tvar}(Px),$$

which is the population mean of $\operatorname{tvar}(Px)$, with $P \sim \lambda_{k,d}$. Similarly, the term $\|p(x_i - x_j)\|^4$ is a polynomial of degree 2 in p, so that $(\mathcal{M}(p, x))^2$ in (III.8) is a polynomial of degree 2 in p. If $\{p_l\}_{l=1}^n$ is a 2-design, then we derive

$$\sum_{l=1}^{n}(\mathcal{M}(p_l, x))^2 - \Big(\sum_{j=1}^{n}\mathcal{M}(p_l, x)\Big)^2 = \mathbb{E}(\mathcal{M}(P, x))^2 - \mathbb{E}\Big(\sum_{j=1}^{n}\mathcal{M}(P, x)\Big)^2,$$

with $P \sim \lambda_{k,d}$. In other words, the sample variance of $\{\mathcal{M}(p_1, x), \ldots, \mathcal{M}(p_n, x)\}$ coincides with the population variance $\operatorname{Var}(\mathcal{M}(P, x))$. Analogously, we deduce that the sample covariance of (III.19) coincides with the population covariance $\operatorname{Cov}(\mathcal{M}(P, x), \operatorname{tvar}(Px))$ with $P \sim \lambda_{k,d}$.

Bibliography

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] C. Bachoc and M. Ehler. Tight p-fusion frames. *Appl. Comput. Harmon. Anal.*, 35(1):1–15, 2013.

[3] C. Bagwell. SoX - Sound Exchange the swiss army knife of sound processing. https://launchpad.net/ubuntu/+source/sox/14.4.1-5. Accessed: 2018-10-31.

[4] K. Ball. An elemntary introduction to modern convex geometry. *Flavors in Geometry*, 31:1–58, 1997.

[5] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx.*, 28(3):253–263, 2008.

[6] R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. In *Foundations of Computational Mathematics*, volume 9, pages 941–944, 2006.

[7] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 245–250, New York, NY, USA, 2001. ACM. URL: `http://doi.acm.org/10.1145/502512.502546`, `doi:10.1145/502512.502546`.

[8] B. Bodman, M. Ehler, and M. Gräf. From low to high-dimensional moments without magic. *J. Theor. Probab.*, 2017.

[9] A. Breger, M. Ehler, H. Bogunovic, S. M. Waldstein, A. Philip, U. Schmidt-Erfurth, and B. S. Gerendas. Supervised learning and dimension reduction techniques for quantification of retinal fluid in optical coherence tomography images. *Eye, Springer Nature*, 2017.

[10] A. Breger, M. Ehler, and M. Gräf. Quasi Monte Carlo integration and kernel-based function approximation on Grassmannians. *Frames and Other Bases in Abstract and Function Spaces, Applied and Numerical Harmonic Analysis series (ANHA), Birkhauser/Springer*, 2017.

[11] A. Breger, M. Ehler, and M. Gräf. Points on manifolds with asymptotically optimal covering radius. *Journal of Complexity*, 48:1–14, 2018.

[12] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[13] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997. URL: `https://doi.org/10.1023/A:1007379606734`, `doi:10.1023/A:1007379606734`.

[14] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[15] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. URL: http://doi.acm.org/10.1145/1143844.1143874, doi:10.1145/1143844.1143874.

[16] P. de la Harpe and C. Pache. Cubature formulas, geometrical designs, reproducing kernels, and Markov operators. In *Infinite groups: geometric, combinatorial and dynamical aspects*, volume 248, pages 219–267, Basel, 2005. Birkhäuser.

[17] M. Dörfler, R. Bammer, and T. Grill. Inside the spectrogram: Convolutional neural networks in audio processing. *IEEE International Conference on Sampling Theory and Applications (SampTA)*, pages 152–155, 2017.

[18] M. Dörfler, T. Grill, R. Bammer, and A. Flexer. Basic filters for convolutional neural networks applied to music: Training or design. *Neural Comput. & Applications*, pages 1–14, 2018.

[19] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 658–666, USA, 2016. Curran Associates Inc. URL: http://dl.acm.org/citation.cfm?id=3157096.3157170.

[20] B. M. et al. Librosa: 0.6.2. https://doi.org/10.5281/zenodo.1342708, 2018. doi:10.5281/zenodo.1342708.

[21] F. C. et al. Keras. https://keras.io, 2015.

[22] U. Etayo, J. Marzo, and J. Ortega-Cerdà. Asymptotically optimal designs on compact algebraic manifolds. *J. Monatsh. Math.*, 186(2):235–248, 2018.

[23] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496, 1998.

[24] P.-A. Ganaye, M. Sdika, and H. Benoit-Cattin. *Semi-supervised Learning for Segmentation Under Semantic Constraint: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III*, pages 595–602. 09 2018. doi:10.1007/978-3-030-00931-1_68.

[25] B. Gerendas, X. Hu, A. Kaider, A. Montuoro, A. Sadeghipour, S. Waldstein, and U. Schmidt-Erfurth. Oct biomarkers predictive for visual acuity in patients with diabetic macular edema. *Investigative Ophthalmology & Visual Science*, 58(8):2026–2026, 06 2017.

[26] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, 1996.

[27] P. Harar. Orthovar. https://gitlab.com/hararticles/orthovar, 2018.

[28] R. Heckel, M. Tschannen, and H. Bölcskei. Dimensionality-reduced subspace clustering. *Information and Inference: A Journal of the IMA*, 6, 03 2017. doi:10.1093/imaiai/iaw021.

[29] C. Hedge, A. C. Sankaranarayanan, W. Yin, and R. G. Baraniuk. Numax: A convex approach for learning near-isometric linear embeddings. *IEEE Transactions on Signal Processing*, 83, 2015.

[30] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[31] J. R. Karr and T. E. Martin. Random numbers and principal components: further searches for the unicorn. Technical report, United States Forest Service General Technical Report, 1981.

[32] F. Krahmer and R. Ward. New and improved Johnson Lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.

[33] G. Kutyniok, W.-Q. Lim, and R. Reisenhofer. Shearlab 3d: Faithful digital shearlet transforms based on compactly supported shearlets. *ACM Trans. Math. Softw.*, 42, 2016, www.shearlab.org.

[34] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017.

[35] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, Jun 1995. URL: https://doi.org/10.1007/BF01200757, doi:10.1007/BF01200757.

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[37] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011. doi:10.1561/2200000035.

[38] J. Matousek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

[39] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26, 1981.

[40] J. Navarrete. The sox of silence. https://digitalcardboard.com/blog/2009/08/25/the-sox-of-silence, 2009.

[41] S. Neumayer, M. Nimmer, S. Setzer, and G. Steidl. On the robust PCA and Weiszfeld's algorithm. *Appl. Math. Optim.*, 2019.

[42] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, R. Guerrero, S. A Cook, A. de Marvao, T. Dawes, D. O'Regan, B. Kainz, B. Glocker, and D. Rueckert. Anatomically constrained neural networks (acnn): Application to cardiac image enhancement and segmentation. *IEEE Transactions on Medical Imaging*, PP, 05 2017. doi:10.1109/TMI.2017.2743464.

[43] G. Pabst. *Parameters for Compartment-free Pharmacokinetics - Standardisation of Study Design, Data Analysis and Reporting*, chapter 5. Area under the concentration-time curve, pages 65–80. Shaker Verlag, 1999.

[44] O. R. Picas, H. P. Rodriguez, D.Dabiri, H.Tokuda, W. Hariya, K. Oishi, and X. Serra. A real-time system for measuring sound goodness in instrumental sounds. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[45] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. arXiv:1505.04597.

[46] P. Seymour and T. Zaslavsky. Averaging sets: a generalization of mean values and spherical designs. *Advances in Math.*, 52:213–240, 1984.

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. arXiv:1409.1556.

[48] D. F. Stauffer, E. O. Garton, and R. K. Steinhorst. A comparison of principal components from real and random data. *Ecology*, 66(6):1693–1698, 1985. URL: http://www.jstor.org/stable/2937364.

[49] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, 2017.

[50] G.-A. Thanei, C. Heinze, and N. Meinshausen. *Random Projections for Large-Scale Regression*, pages 51–68. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-41573-4_3.

[51] M. Udell. *Generalized Low Rank Models*. PhD thesis, Stanford University, 2015.

[52] E. S. Varnousfaderani, J. Wu, W.-D. Vogl, A.-M. Philip, A. Montuoro, R. Leitner, C. Simader, S. M. Waldstein, B. S. Gerendas, and U. Schmidt-Erfurth. A novel benchmark model for intelligent annotation of spectral-domain optical coherence tomography scans using the example of cyst annotation. *Computer Methods and Programs in Biomedicine*, 130:93–105, 2016.

[53] J. Veraart, D. S. Novikov, D. Christiaens, B. Ades-aron, J. Sijbers, and E. Fieremans. Denoising of diffusion mri using random matrix theory. *NeuroImage*, 142:394 – 406, 2016. doi:https://doi.org/10.1016/j.neuroimage.2016.08.016.

[54] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012.

[55] E. von Hornbostel and C. Sachs. Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann. *The Galpin Society Journal*, pages 3–29, 1961.

[56] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press. URL: http://dl.acm.org/citation.cfm?id=2969033.2969197.

[57] L. Zhang, R. Lukac, X. Wu, and D. Zhang. Pca-based spatially adaptive denoising of cfa images for single-sensor digital cameras. *IEEE Transactions on Image Processing*, 18(4):797–812, April 2009. doi:10.1109/TIP.2008.2011384.

[58] Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems (NIPS)*, 31, 2018.

[59] I. ZyTrax. Frequency ranges. http://www.zytrax.com/tech/audio/audio.html, 2018.

IV Gabor Frames and Deep Scattering Networks in Audio Processing

Outline

Bibliographic information

R. Bammer, M. Dörfler, and P. Harar. Gabor frames and deep scattering networks in audio processing. *arXiv preprint*, 2017. `arXiv:1706.08818`, submitted.

Author's contribution

The author contributed to section Introduction, designed the implementation of the proposed algorithm and created most of the visualizations. Furthermore, he designed the synthetic data generator, preprocessed the data and performed the numerical experiments. Wrote a significant part of section Discussion and Future Work. He was helping with the finalization of the manuscript.

Abstract

This paper introduces Gabor scattering, a feature extractor based on Gabor frames and Mallat's scattering transform. By using a simple signal model for audio signals specific properties of Gabor scattering are studied. It is shown that for each layer, specific invariances to certain signal characteristics occur. Furthermore, deformation stability of the coefficient vector generated by the feature extractor is derived by using a decoupling technique which exploits the contractivity of general scattering networks. Deformations are introduced as changes in spectral shape and frequency

modulation. The theoretical results are illustrated by numerical examples and experiments. Numerical evidence is given by evaluation on a synthetic and a "real" data set, that the invariances encoded by the Gabor scattering transform lead to higher performance in comparison with just using Gabor transform, especially when few training samples are available.

Acknowledgment

This work was supported by the Uni:docs Fellowship Programme for Doctoral Candidates in Vienna, by the Vienna Science and Technology Fund (WWTF) project SALSA (MA14-018), by the International Mobility of Researchers (CZ.02.2.69/0.0/0.0/16027/0008371), and by the project LO1401. Infrastructure of the SIX Center was used for computation.

IV.1 Introduction

During the last two decades, enormous amounts of digitally encoded and stored audio have become available. For various purposes, the audio data, be it music or speech, need to be structured and understood. Recent machine learning techniques known as (deep) convolutional neural networks (CNN) have led to state of the art results for several tasks such as classification, segmentation or voice detection, cf. [21, 10]. CNNs were originally proposed for images [21], which may be directly fed into a network. Audio signals, on the other hand, commonly undergo some pre-processing in order to extract features which are then used as input to a trainable machine. Very often, these features consist of one or several two-dimensional arrays, such that the image processing situation is mimicked in a certain sense. However, the question about the impact of this very first processing step is important and it is not entirely clear whether a short-time Fourier transform (STFT), here based on *Gabor frames*, the most common representation system used in the analysis of audio signals, leads to optimal feature extraction. The convolutional layers of the CNNs can themselves be seen as feature extractors, often followed by a classification stage, either in the form of one or several dense network layers or classification tools such as support vector machine (SVM). Stéphane Mallat gave a first mathematical analysis of CNN as feature extractor, thereby introducing the so called *scattering transform*, based on wavelet transforms and modulus non-linearity in each layer [22]. The basic structure thus parallels the one of CNNs, since these networks are equally composed of multiple layers of local convolutions, followed by a non-linearity and, optionally, a pooling operator, cp. Section IV.1.1.

In the present contribution, we consider an approach inspired by Mallats scattering transform, but based on Gabor frames, respectively Gabor transform (GT). The resulting feature extractor is called *Gabor scattering* (GS). Our approach is a special case of the extension of Mallat's scattering transform proposed by Wiatowski and Bölcskei [28, 27], which introduces the possibility to use different semi-discrete frames, Lipschitz-continuous non-linearities and pooling operators in each layer. In [22, 3, 2], invariance and deformation stability properties of the scattering transform w.r.t. operators defined via some group action were studied. In the more general setting of [28, 27], vertical translation invariance, depending on the network depth, and deformation stability for band-limited functions have been proved. In this contribution, we study the same properties of the GS and a particular class of signals, which model simple musical tones (Section IV.2.2).

Due to this concrete setting, we obtain quantitative invariance statements and deformation stability to specific, musically meaningful, signal deformations. Invariances are studied considering the first two layers, where the feature extractor extracts cer-

tain signal features of the signal model (i.e. frequency and envelope information), cp. Section IV.3.1.1. By using a low-pass filter and pooling in each layer, temporal fine structure of the signal is averaged out. This results in invariance w.r.t. the envelope in the first and frequency invariance in the second layer output. In order to compute deformation bounds for the GS feature extractor, we assume more specific restrictions than band-limitation and use the decoupling technique, first presented in [28] and [12]. Deformation stability is proven by only computing the robustness of the signal class w.r.t spectral shape and frequency modulation, see Section IV.3.1.2. The robustness result together with contractivity of the feature extractor, which is given by the networks architecture, yields deformation stability.
To empirically demonstrate the benefits of GS time-frequency representation for classification, we have conducted a set of experiments. In a supervised learning setting, where the main aim is the multi-class classification of generated sounds, we have utilized a CNN as a classifier. In these numerical experiments, we compare the GS to a STFT-based representation. We demonstrate the benefits of GS in a quasi-ideal setting on a self implemented synthetic data set and we also investigate, if it benefits the performance on a real data set, namely *GoodSounds* [26]. Moreover we focus on comparing these two time-frequency representations in terms of performance on limited sizes of training data, see Section IV.4.

IV.1.1 Convolutional Neural Networks (CNNs) and Invariance

CNNs are a specific class of neural network architectures which have shown extremely convincing results on various machine learning tasks in the past decade. Most of the problems addressed by using CNNs are based on, often big amounts of, annotated data, in which case one speaks about supervised learning. When learning from data, the intrinsic task of the learning architecture is to gradually extract useful information and suppress redundancies, which always abound in natural data. More formally, the learning problem of interest may be invariant to various changes of the original data and the machine or network must learn these invariances in order to avoid over-fitting. Since, given a sufficiently rich architecture, a deep neural network can practically fit arbitrary data, cp. [30, 18], good generalization properties depend on the systematic incorporation of the intrinsic invariances of the data. Generalization properties hence suffer if the architecture is too rich given the amount of available data. This problem is often addressed by using data augmentation. Here, *we raise the hypothesis that using prior representations which encode some potentially useful invariances will increase the generalization quality, in particular when using a restricted size of data set.* The evaluation of the performance on validation data in comparison to the results on test data strengthens our hypothesis for the

experimental problem presented in Section IV.4.

In order to understand the mathematical construction used within this paper, we briefly introduce the principal idea and structure of a CNN. We shall see, that the scattering transforms in general, and the GS in particular, follow a similar concept of concatenating various processing steps which ultimately lead to rather flexible grades of invariances in dependence on the chosen parameters. Usually a CNN consists of several layers, namely an input, several hidden (since we consider the case of **deep** CNN the number of hidden layers is supposed to be ≥ 2) and one output layer. A hidden layer consists of the following steps: first the convolution of the data with a small weighting matrix, often referred to as a kernel [1], which can be interpreted as localization of certain properties of the input data. The main advantage in this setup is that only the size and number of these (convolutional) kernels is fixed, but their coefficients are learned during training. So they reflect the structure of the training data in the best way w.r.t the task being solved. The next building block of the hidden layer is the application of a non-linearity function, also called activation function, which signals if information of this neuron is relevant to be transmitted. Finally, in order to reduce redundancy and increase invariance, pooling is applied. Due to these building blocks, invariances to specific deformations and variations in the data set, are generated in dependence on the specific filters used, whether they are learned, as in the classical CNN case or designed, as in the case of scattering transforms [23]. In this work, we will derive concrete qualitative statements about invariances for a class of music signals and will show by numerical experiments that these invariances indeed lead to a better generalization of the CNNs used to classify data.

Note that in a neural network, in particular in CNNs, the output, e.g. classification labels, is obtained after several concatenated hidden layers. In the case of scattering network the outputs of each layer are stacked together into a feature vector and further processing is necessary to obtain the desired result. Usually, after some kind of dimensionality reduction, cf. [29], this vector can be fed into a SVM or a dense NN, which performs the classification task.

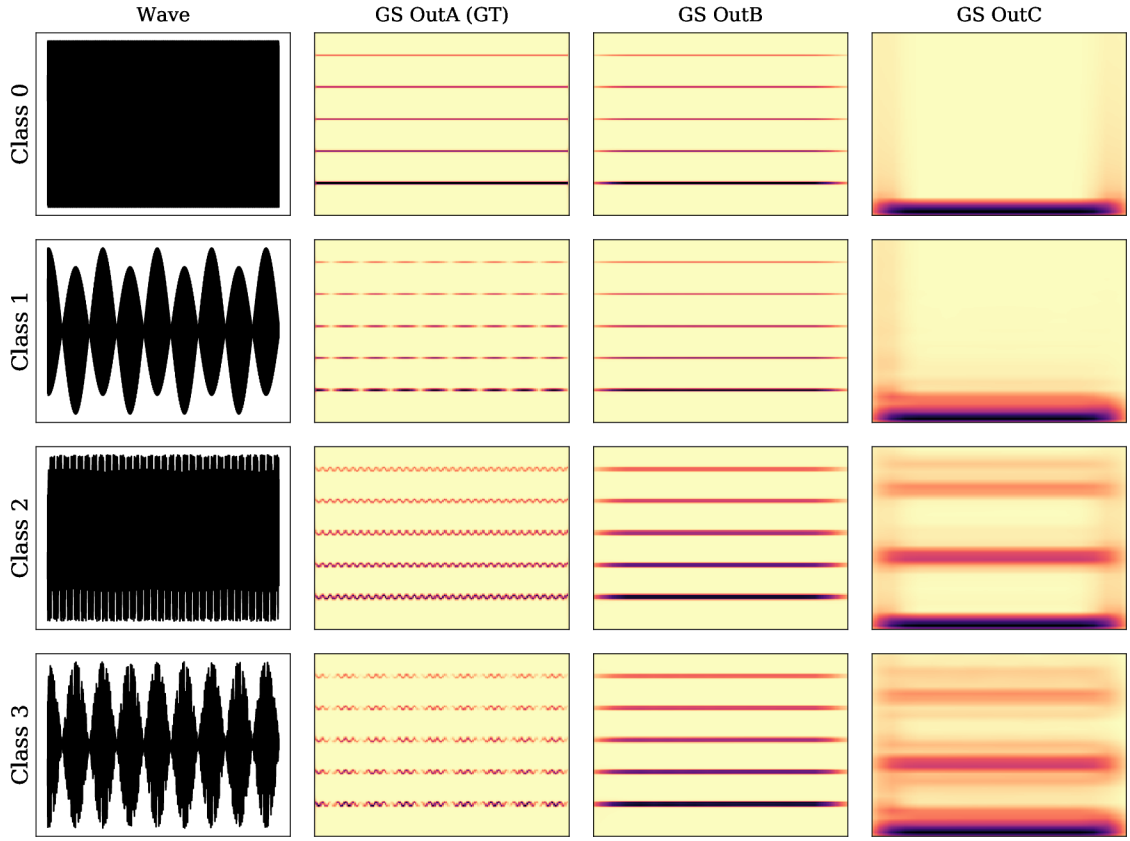


Fig. IV.1: Waves, Outputs A, i.e. GT, Outputs B and Outputs C of GS for all four classes of generated sound.

IV.1.2 Invarince induced by Gabor Scattering

In this section we give a motivation for the theory and underlying aim of this paper. In Fig. IV.1 we see sound examples from different classes, where Class 0 is a pure tone with 5 harmonics, Class 1 is an amplitude modulated version thereof, Class 2 is the frequency modulated version and Class 3 contains the amplitude and frequency modulated signal. So we have got classes with different amplitudes, as clearly visible in the wave forms shown in the left-most plots. In this paper we introduce GS, as a new feature extractor that introduces certain invariances. GS has several layers, denoted by OutA, OutB and OutC and each layer is invariant with respect to some features. The first layer, here OutA, is the spectrogram of the wave form. So we see the time-frequency content of the four classes. OutB can be seen to be invariant

[1]In order to prevent any confusion with the kernels used in classical machine learning methods based on reproducing kernel Hilbert spaces, e.g. the famous support vector machine, c.f.[16], we point out that the term kernel as used in this work *always* means convolutional kernels in the sense of filterbanks. Both the fixed kernels used in the scattering transform and the kernels used in the CNNs, whose size is fixed, but whose coefficients are learned, should be interpreted as convolutional kernels in a filterbank.

w.r.t. amplitude changes, while the last layer, OutC is invariant w.r.t. to frequency content while encoding the amplitude information. With GS it is therefore possible to separate different qualities of information contained in a spectrogram.
We introduce GS mathematically in the Section IV.2.1, and elaborate on the resulting invariances in different layers in Section IV.3.1.1. Numerical experiments, showing the benefit of GS, are discussed in Section IV.4.

IV.2 Materials and Methods

IV.2.1 Gabor Scattering

Since Wiatowski and Bölcskei used general semi-discrete frames to obtain a wider class of window functions for the scattering transform (cp. [28, 27]), it seems natural to consider specific frames used for audio data analysis. Hence we use Gabor frames for the scattering transform and study corresponding properties. We next introduce the basics of Gabor frames and refer to [11] for more details. A sequence $(g_k)_{k=1}^{\infty}$ of elements in a Hilbert space $\mathcal{H}$ is called frame if there exist positive frame bounds $A, B > 0$ such that for all $f \in \mathcal{H}$

$$A\|f\|^2 \leq \sum_{k=1}^{\infty} |\langle f, g_k \rangle|^2 \leq B\|f\|^2. \tag{IV.1}$$

If $A = B$, then we call $(g_k)_{k=1}^{\infty}$ a tight frame.

Remark 1. *In our context the Hilbert space $\mathcal{H}$ is either $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$.*

In order to define Gabor frames we need to introduce two operators, i.e. the translation and modulation operator.

- The translation (time shift) operator:
  - for a function $f \in L^2(\mathbb{R})$ and $x \in \mathbb{R}$ is defined as $T_x f(t) := f(t - x)$ for all $t \in \mathbb{R}$.
  - for a function $f \in \ell^2(\mathbb{Z})$ and $k \in \mathbb{Z}$ is defined as $T_k f(j) := (f(j - k))_{j \in \mathbb{Z}}$.
- The modulation (frequency shift) operator:
  - for a function $f \in L^2(\mathbb{R})$ and $\omega \in \mathbb{R}$ is defined as $M_\omega f(t) := e^{2\pi i \omega t} f(t)$ for all $t \in \mathbb{R}$.
  - for a function $f \in \ell^2(\mathbb{Z})$ and $\omega \in [-\frac{1}{2}, \frac{1}{2}]$ is defined as $M_\omega f(j) := (e^{2\pi i \omega j} f(j))_{j \in \mathbb{Z}}$.

We use these operators to express the STFT of a function $f \in \mathcal{H}$ with respect to a given window function $g \in \mathcal{H}$ as $V_g f(x, \omega) = \langle f, M_\omega T_x g \rangle$. In order to reduce redundancy, we sample $V_g f$ on a separable lattice $\Lambda = \alpha\mathbb{Z} \times \mathcal{I}$, where $\mathcal{I} = \beta\mathbb{Z}$ in case of $\mathcal{H} = L^2(\mathbb{R})$ and $\mathcal{I} = \{0, ..., \frac{(M-1)}{M}\}$ with $\beta = \frac{1}{M}$ in case $\mathcal{H} = \ell^2(\mathbb{Z})$. The

sampling is done in time by $\alpha > 0$ and in frequency by $\beta > 0$. The resulting samples correspond to the coefficients of f w.r.t. a *Gabor system*.

Definition 1. (Gabor System)
Given a window function $0 \neq g \in \mathcal{H}$ and lattice parameters $\alpha, \beta > 0$, the set of time-frequency shifted versions of g

$$G(g, \alpha, \beta) = \{M_{\beta j} T_{\alpha k} g : (\alpha k, \beta j) \in \Lambda\}$$

is called a Gabor system.

This Gabor system is called Gabor frame if it is a frame, see (IV.1). We proceed to introduce a scattering transform based on Gabor frames. We base our considerations on [28] by using a triplet-sequence $\Omega = \Big((\Psi_\ell, \sigma_\ell, S_\ell)\Big)_{\ell \in \mathbb{N}}$, where ℓ is associated to the ℓ-th layer of the network. Note that in this contribution we will deal with Hilbert spaces $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$; more precisely in the input layer, i.e. the $0-$th layer, we have $\mathcal{H}_0 = L^2(\mathbb{R})$ and due to the discretization inherent in the GT, $\mathcal{H}_\ell = \ell^2(\mathbb{Z})\ \forall \ell > 0$.
We recall the elements of the triplet:

- $\Psi_\ell := \{g_{\lambda_\ell}\}_{\lambda_\ell \in \Lambda_\ell}$ with $g_{\lambda_\ell} = M_{\beta_\ell j} T_{\alpha_\ell k} g_\ell$, $\lambda_\ell = (\alpha_\ell k, \beta_\ell j)$, is a Gabor frame indexed by a lattice Λ_ℓ.
- A non-linearity function (e.g. rectified linear units, modulus function, see [28]) $\sigma_\ell : \mathbb{C} \to \mathbb{C}$, is applied pointwise and is chosen to be Lipschitz-continuous, i.e. $\|\sigma_\ell f - \sigma_\ell h\|_2 \leq L_\ell \|f - h\|_2$ for all $f, h \in \mathcal{H}$. In this paper we only use the modulus function with Lipschitz constant $L_\ell = 1$ for all $\ell \in \mathbb{N}$.
- Pooling depends on a pooling factor $S_\ell > 0$, which leads to dimensionality reduction. Mostly used are max- or average-pooling, some more examples can be found in [28]. In our context, pooling is covered by choosing specific lattices Λ_ℓ in each layer.

In order to explain the interpretation of GS as CNN, we write $\mathcal{I}(g)(t) = g(-t)$ and have

$$|\langle f, M_{\beta j} T_{\alpha k} g \rangle| = \Big| f * \Big(\mathcal{I}\Big(M_{\beta j}(g)\Big)\Big)\Big| (\alpha k). \tag{IV.2}$$

Thus the Gabor coefficients can be interpreted as the samples of a convolution.
We start by defining *paths* on index sets $q := (q_1, ..., q_\ell) = (\beta_1 j_1, ..., \beta_\ell j_\ell) \in \beta_1 \mathbb{Z} \times ... \times \beta_\ell \mathbb{Z} =: \mathcal{B}^\ell, \ell \in \mathbb{N}$.

Definition 2. (Gabor Scattering)
Let $\Omega = \Big((\Psi_\ell, \sigma_\ell, \Lambda_\ell)\Big)_{\ell \in \mathbb{N}}$ be a given triplet-sequence. Then components of the ℓ-th

layer of the GS transform are defined to be the output of the operator $U_\ell[q_\ell] : \mathcal{H}_{\ell-1} \to \mathcal{H}_\ell$, $q_\ell \in \beta_\ell \mathbb{Z}$*:*

$$f_\ell^{(q_1,...,q_\ell)}(k) = U_\ell[\beta_\ell j_\ell] f_{\ell-1}^{(q_1,...,q_{\ell-1})}(k) := \sigma_\ell \left(\langle f_{\ell-1}^{(q_1,...,q_{\ell-1})}, M_{\beta_\ell j_\ell} T_{\alpha_\ell k} g_\ell \rangle_{\mathcal{H}_{\ell-1}} \right) \quad j_\ell, k \in \mathbb{Z}, \tag{IV.3}$$

where $f_{\ell-1}$ *is some output-vector of the previous layer and* $f_\ell \in \mathcal{H}_\ell \ \forall \ell \in \mathbb{N}$*. The GS operator is defined as*

$$U[q]f = U[(q_1, ..., q_\ell)]f := U_\ell[q_\ell] \cdots U_1[q_1]f.$$

Similar to [28], for each layer, we use one atom of the Gabor frame in the subsequent layer as output-generating atom, i.e. $\phi_{\ell-1} := g_\ell$. Note that convolution with this element corresponds to low-pass filtering.[1] We next introduce a countable set $\mathcal{Q} := \bigcup_{\ell=0}^{\infty} \mathcal{B}^\ell$, which is the union of all possible paths of the net and the space $(\ell^2(\mathbb{Z}))^{\mathcal{Q}}$ of sets of $\mathcal{Q}$ elements from $\ell^2(\mathbb{Z})$. Now we define the feature extractor $\Phi_\Omega(f)$ of a signal $f \in L^2(\mathbb{R})$ as in [28, Def. 3] based on chosen (not learned) Gabor windows.

Definition 3. (Feature Extractor)
Let $\Omega = \left((\Psi_\ell, \sigma_\ell, \Lambda_\ell) \right)_{\ell \in \mathbb{N}}$ *be a triplet-sequence and* ϕ_ℓ *the output-generating atom for layer* ℓ*. Then the feature extractor* $\Phi_\Omega : L^2(\mathbb{R}) \to (\ell^2(\mathbb{Z}))^{\mathcal{Q}}$ *is defined as*

$$\Phi_\Omega(f) := \bigcup_{\ell=0}^{\infty} \left\{ (U[q]f) * \phi_\ell \right\}_{q \in \mathcal{B}^\ell}. \tag{IV.4}$$

In the following section we are going to introduce the signal model which we consider in this paper.

IV.2.2 Musical Signal Model

Tones are one of the smallest units and simple models of an audio signal, consisting of one fundamental frequency ξ_0, corresponding harmonics $n\xi_0$ and a shaping envelope A_n for each harmonic, providing specific timbre. Further, since our ears are limited to frequencies below $20kHz$, we develop our model over finitely many harmonics, i.e. $\{1, ..., N\} \subset \mathbb{N}$.
The general model has the following form

$$f(t) = \sum_{n=1}^{N} A_n(t) e^{2\pi i \eta_n(t)}, \tag{IV.5}$$

where $A_n(t) \geq 0 \ \forall n \in \{1, ..., N\}$ and $\forall t$. For one single tone we choose $\eta_n(t) = n\xi_0 t$. Moreover we create a space of tones $\mathcal{T} = \left\{ \sum_{n=1}^{N} A_n(t) e^{2\pi i n \xi_0 t} | A_n \in \mathcal{C}_c^\infty(\mathbb{R}) \right\}$ and assume $\|A_n\|_\infty \leq \frac{1}{n}$.

[1] In general one could take $\phi_{\ell-1} := g_{\lambda_\ell^*}, \lambda_\ell^* \in \Lambda_\ell$. Since this element is the ℓ-th convolution, it is an element of the ℓ-th frame, but because it belongs to the $(\ell-1)$-th layer, its index is $(\ell-1)$.

IV.3 Theoretical Results

IV.3.1 Gabor Scattering of Music Signals

IV.3.1.1 Invariance

In [3] it was already stated that due to the structure of the scattering transform the energy of the signal is pushed towards low frequencies, where it is then captured by a low-pass filter as output-generating atom. The current section explains how GS separates relevant structures of signals modeled by the signal space $\mathcal{T}$. Due to the smoothing action of the output-generating atom, each layer expresses certain invariances, which will be illustrated by numerical examples in Section IV.3.2. In Proposition 1, inspired by [3], we add some assumptions on the analysis window in the first layer $g_1 : |\hat{g}_1(\omega)| \leq C_{\hat{g}_1}(1+|\omega|^s)^{-1}$ for some $s > 1$ and $\|tg_1(t)\|_1 = C_{g_1} < \infty$.

Proposition 1 (Layer 1)**.** *Let $f \in \mathcal{T}$ with $\|A_n'\|_\infty \leq C_n < \infty \ \forall n \in \{1, ..., N\}$. For fixed j, for which $n_0 = \underset{n\in\{1,...,N\}}{argmin} |\beta_1 j - \xi_0 n|$ such that $|\beta j - \xi_0 n_0| \leq \frac{\xi_0}{2}$, can be found, we obtain:*

$$U[\beta_1 j](f)(k) = |\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1\rangle| = A_{n_0}(\alpha_1 k)|\hat{g}_1(\beta_1 j - n_0\xi_0)| + E_1(k) \tag{IV.6}$$

$$E_1(k) \leq C_{g_1} \sum_{n=1}^{N} \|A_n' \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty + C_{\hat{g}_1} \sum_{n=2-n_0}^{N-n_0} \frac{1}{n_0 + n - 1} \left(1 + \left|\xi_0\right|^s \left|n - \frac{1}{2}\right|^s\right)^{-1}, \tag{IV.7}$$

where χ is the indicator function.

Remark 2. *Equation* (IV.6) *shows that for slowly varying amplitude functions A_n, the first layer mainly captures the contributions near the frequencies of the tone's harmonics. Obviously, for time-sections during which the envelopes A_n undergo faster changes, such as during a tone's onset, energy will also be found outside a small interval around the harmonics' frequencies and thus the error estimate* (IV.7) *becomes less stringent. The second term of the error in* (IV.7) *depends only on the window g_1 and its behaviour is governed by the frequency decay of g_1. Note that the error bound increases for lower frequencies, since the separation of the fundamental frequency and corresponding harmonics by the analysis window deteriorates.*

Proof. Step1 – Using the signal model for tones as input, interchanging the finite

sum with the integral and performing a substitution $u = t - \alpha_1 k$, we obtain

$$\begin{aligned}\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle &= \langle \sum_{n=1}^{N} M_{n\xi_0} A_n, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle \\ &= \sum_{n=1}^{N} \langle A_n, M_{\beta_1 j - n\xi_0} T_{\alpha_1 k} g_1 \rangle \\ &= \sum_{n=1}^{N} \int_{\mathbb{R}} A_n(u + \alpha_1 k) g_1(u) e^{-2\pi i (\beta_1 j - n\xi_0)(u+\alpha_1 k)} du.\end{aligned}$$

After performing a Taylor series expansion locally around $\alpha_1 k$:
$A_n(u + \alpha_1 k) = A_n(\alpha_1 k) + u R_n(\alpha_1 k, u)$, where the remainder can be estimated by $|R_n(\alpha_1 k, u)| \leq \|A_n' \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty$, we have

$$\begin{aligned}\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle = \sum_{n=1}^{N} \Bigg[& e^{-2\pi i (\beta_1 j - n\xi_0)\alpha_1 k} A_n(\alpha_1 k) \int_{\mathbb{R}} g_1(u) e^{-2\pi i (\beta_1 j - n\xi_0) u} du \\ & + \int_{\mathbb{R}} u R_n(\alpha_1 k, u) g_1(u) e^{-2\pi i (\beta_1 j - n\xi_0)(u+\alpha_1 k)} du \Bigg].\end{aligned}$$

Hence we choose $n_0 = \underset{n}{\operatorname{argmin}} |\beta_1 j - \xi_0 n|$, set

$$\mathcal{E}_n(k) = \int_{\mathbb{R}} u R_n(\alpha_1 k, u) g_1(u) e^{-2\pi i (\beta_1 j - n\xi_0)(u+\alpha_1 k)} du \tag{IV.8}$$

$$\widetilde{E}(k) = \sum_{\substack{n=1 \\ n \neq n_o}}^{N} e^{-2\pi i (\beta_1 j - n\xi_0)\alpha_1 k} A_n(\alpha_1 k) \hat{g}_1(\beta_1 j - n\xi_0) \tag{IV.9}$$

and split the sum to obtain

$$\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle = A_{n_0}(\alpha_1 k) e^{-2\pi i (\beta_1 j - n_0 \xi_0)\alpha_1 k} \hat{g}_1(\beta_1 j - n_0 \xi_0) + \widetilde{E}(k) + \sum_{n=1}^{N} \mathcal{E}_n(k).$$

Step 2 – We bound the error terms, starting with (IV.8):

$$\left| \sum_{n=1}^{N} \mathcal{E}_n(k) \right| = \left| \sum_{n=1}^{N} \int_{\mathbb{R}} u R_n(\alpha_1 k, u) g_1(u) e^{-2\pi i (\beta_1 j - n_0 \xi_0)(u+\alpha_1 k)} du \right|.$$

Using triangle inequality and the estimate for the Taylor remainder, we obtain together with the assumption on the analysis window

$$\begin{aligned}\left| \sum_{n=1}^{N} \mathcal{E}_n(k) \right| &\leq \sum_{n=1}^{N} \|A_n' \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty \int_{\mathbb{R}} |u g_1(u)| du \\ &\leq C_{g_1} \sum_{n=1}^{N} \|A_n' \cdot T_k \chi[-\alpha_1; \alpha_1]\|_\infty.\end{aligned}$$

For the second bound, i.e. the bound of Equation (IV.9), we use the decay condition on $\hat{g}_1$, thus

$$|\widetilde{E}(k)| \leq C_{\hat{g}_1} \sum_{\substack{n=1 \\ n \neq n_o}}^{N} |A_n(\alpha_1 k)| \left(1 + |\beta_1 j - \xi_0 n|^s\right)^{-1}.$$

Next we split the sum into $n > n_0$ and $n < n_0$. We estimate the error term for $n > n_0$:

$$\sum_{n=n_0+1}^{N} |A_n(\alpha_1 k)|\Big(1+|\beta_1 j-\xi_0 n|^s\Big)^{-1} = \sum_{n=1}^{N-n_0} |A_{n_0+n}(\alpha_1 k)|\Big(1+|\beta_1 j-\xi_0 n_0-\xi_0 n|^s\Big)^{-1}. \tag{IV.10}$$

Since $n_0 = \underset{n}{\operatorname{argmin}}|\beta_1 j - \xi_0 n|$, we have $|\beta_1 j - \xi_0 n_0| \leq \frac{\xi_0}{2}$ and also using $\|A_n\|_\infty \leq \frac{1}{n}$, we obtain

$$\sum_{n=1}^{N-n_0} |A_{n_0+n}(\alpha_1 k)| \left(1+\left|\frac{\xi_0}{2}-\xi_0 n\right|^s\right)^{-1} \leq \sum_{n=1}^{N-n_0} \frac{1}{n_0+n}\left(1+\left|\xi_0\right|^s\left|n-\frac{1}{2}\right|^s\right)^{-1}. \tag{IV.11}$$

Further we estimate the error for $n < n_0$:

$$\sum_{n=1}^{n_0-1} |A_n(\alpha_1 k)|(1+|\beta_1 j-\xi_0 n|^s)^{-1} \leq \sum_{n=1}^{n_0-1} |A_n(\alpha_1 k)|(1+|\beta_1 j-\xi_0 n_0+\xi_0 n_0-\xi_0 n|^s)^{-1},$$

where we added and subtracted the term $\xi_0 n_0$. Due to the reverse triangle inequality and $|\beta_1 j - \xi_0 n_0| \leq \frac{\xi_0}{2}$ we obtain:

$$\Big|\beta_1 j-\xi_0 n_0-\xi_0(n-n_0)\Big| \geq \Big|\xi_0(n_0-n)-\frac{\xi_0}{2}\Big|.$$

For convenience we call $m = n - n_0$ and perform a little trick by adding and subtracting $\frac{1}{2}$, so $\Big|\xi_0(n_0-n)-\frac{\xi_0}{2}\Big| = \Big|\xi_0\Big| - (m+1) + \frac{1}{2}\Big|$. The reason for this steps will become more clear when putting the two sums back together. Now we have

$$\sum_{n=1}^{n_0-1} |A_n(\alpha_1 k)|(1+|\beta_1 j-\xi_0 n|^s)^{-1} \leq \sum_{m=1-n_0}^{-1} |A_{n_0+m}(\alpha_1 k)|\Big(1+\Big|\xi_0\Big|^s\Big|(m+1)-\frac{1}{2}\Big|^s\Big)^{-1}.$$

Shifting the sum, i.e. taking $n = m + 1$, and using $\|A_n\|_\infty \leq \frac{1}{n}$, we get

$$\sum_{m=1-n_0}^{-1} |A_{n_0+m}(\alpha_1 k)|\Big(1+\Big|\xi_0\Big|^s\Big|(m+1)-\frac{1}{2}\Big|^s\Big)^{-1} \leq \sum_{n=2-n_0}^{0} \frac{1}{n_0+n-1}\Big(1+\Big|\xi_0\Big|^s\Big|n-\frac{1}{2}\Big|^s\Big)^{-1}. \tag{IV.12}$$

Combining the two sums (IV.11) and (IV.12) and observing that $\frac{1}{n_0+n} < \frac{1}{n_0+n-1}$, we obtain

$$|\widetilde{E}(k)| \leq C_{\hat{g}_1} \sum_{n=2-n_0}^{N-n_0} \frac{1}{n_0+n-1}\left(1+\left|\xi_0\right|^s\left|n-\frac{1}{2}\right|^s\right)^{-1}. \tag{IV.13}$$

Summing up the error terms, we obtain (IV.7).

□

To obtain the GS coefficients, we need to apply the output-generating atom as in (IV.4).

Corollary 1 (Output of Layer 1)**.** *Let $\phi_1 \in \Psi_2$ be the output-generating atom, then the output of the first layer is*

$$\Big(U_1[\beta_1 j]f * \phi_1\Big)(k) = |\hat{g}_1(\beta_1 j - n_0\xi)|(A_{n_0} * \phi_1)(k) + \epsilon_1(k),$$

where

$$\epsilon_1(k) \leq \|E_1\|_\infty^2 \|\phi_1\|_1^2.$$

Here E_1 is the error term of Proposition 1.

Remark 3. *Note that we focus here on an unmodulated Gabor frame element ϕ_1 and the convolution may be interpreted as a low-pass filter. Hence, in dependence on the pooling factor α_1, the temporal fine-structure of A_{n_0} corresponding to higher frequency content is averaged out.*

Proof. For this proof we use the result of Proposition 1. We show the calculations for the first layer, for the second layer it is similar:

$$\begin{aligned}&\Big|\sum_k \Big(|\langle f, M_{\beta_1 j}T_{\alpha_1 k}g_1\rangle| - |\hat{g}_1(\beta_1 j - \xi_0 n_0)|A_{n_0}(k)\Big)\cdot \phi_1(l-k)\Big|^2 \\ &= \Big|\sum_k E_1(k)\phi_1(l-k)\Big|^2 \leq \|E_1\|_\infty^2\|\phi_1\|_1^2\end{aligned} \tag{IV.14}$$

where $E_1(k) \leq C_{g_1}\sum_{n=1}^{N}\|A_n'\cdot T_k\chi[-\alpha_1;\alpha_1]\|_\infty + C_{\hat{g}_1}\sum_{n=2-n_0}^{N-n_0}\frac{1}{n_0+n-1}\Big(1+|\xi_0|^s|n-\frac{1}{2}|^s\Big)^{-1}$.

□

We introduce two more operators, first the sampling operator $S_\alpha\Big(f(x)\Big) = f(\alpha x)$ $\forall x \in \mathbb{R}$ and second the periodization operator $P_{\frac{1}{\alpha}}\Big(\hat{f}(\omega)\Big) = \sum_{k\in\mathbb{Z}}\hat{f}(\omega - \frac{k}{\alpha})$ $\forall\omega \in \mathbb{R}$. These operators have the following relation $\mathcal{F}\Big(S_\alpha(f)\Big)(\omega) = P_{\frac{1}{\alpha}}(\hat{f}(\omega))$. In order to see how the second layer captures relevant signal structures, depending on the first layer, we propose the following Proposition 2. Recall that $g_\ell \in \mathcal{H}_\ell$ $\forall \ell \in \mathbb{N}$.

Proposition 2 (Layer 2)**.** *Let $f \in \mathcal{T}$, $\sum_{k\neq 0}|\hat{A}_{n_0}(.-\frac{k}{\alpha_1})| \leq \varepsilon_{\alpha_1}$ and $|\hat{g}_2(h)| \leq C_{\hat{g}_2}(1+|h|^s)^{-1}$. Then the elements of the second layer can be expressed as*

$$U_2[\beta_2 h]U_1[\beta_1 j]f(m) = \Big|\hat{g}_1(\beta_1 j - \xi_0 n_0)\Big|\Big|\Big\langle M_{-\beta_2 h}A_{n_0}, T_{\alpha_2 m}g_2\Big\rangle\Big| + E_2(m), \tag{IV.15}$$

where

$$E_2(m) \leq \varepsilon_{\alpha_1}C_{\hat{g}_2}|\hat{g}_1(\beta_1 j - \xi_0 n_0)|\sum_r\Big(1+|\beta_2 h - r|^s\Big)^{-1} + \|E_1\|_\infty\cdot\|g\|_1.$$

Remark 4. *Note that, since the envelopes A_n are expected to change slowly except around transients, their Fourier transforms concentrate their energy in the low frequency range. Moreover the modulation term $M_{-\beta_2 h}$ pushes the frequencies of A_{n_0} down by $-\beta_2 h$ and therefore they can be captured by the output-generating atom ϕ_2 in Corollary 2. In Section IV.3.2 it will be shown by means of the analysis of example signals, how the second layer output distinguishes tones which have a smooth onset (transient) from those which have a sharp attack, which leads to broadband characteristics of A_n around this attack. Similarly, if A_n undergoes an amplitude modulation, the frequency of this modulation can be clearly discerned, cf. Figure IV.5 and the corresponding example. This observation is clearly reflected in expression* (IV.15).

Proof. Using the outcome of Proposition 1 we obtain

$$\begin{aligned}
&U_2[\beta_2 h]U_1[\beta_1 j]f(m) = \\
&|\langle S_{\alpha_1}(A_{n_0})|\hat{g}_1(\beta_1 j - \xi_0 n_0)| + E_1, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})}| \leq \\
&|\langle S_{\alpha_1}(A_{n_0})|\hat{g}_1(\beta_1 j - \xi_0 n_0)|, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})}| + |\langle E_1, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})}|.
\end{aligned}$$

For the error $E_1(k)$ we use the global estimate $|\langle E_1, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})}| \leq \|E_1\|_\infty \cdot \|g\|_1$. Moreover using the notation above and ignoring the constant term $|\hat{g}_1(\beta_1 j - \xi_0 n_0)|$ we proceed as follows:

$$\begin{aligned}
&\langle S_{\alpha_1}(A_{n_0}), M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})} = \sum_{k\in\mathbb{Z}} S_{\alpha_1}(A_{n_0}(k))T_{\alpha_2 m}g_2(k)e^{-2\pi i\beta_2 hk} = \\
&\mathcal{F}\Big(S_{\alpha_1}(A_{n_0})\cdot T_{\alpha_2 m}g_2\Big)(\beta_2 h) = \mathcal{F}\Big(S_{\alpha_1}(A_{n_0})\Big) * \mathcal{F}\Big(T_{\alpha_2 m}g_2\Big)(\beta_2 h) = \\
&P_{\frac{1}{\alpha_1}}\Big(\hat{A}_{n_0}\Big) * \Big(M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h) = \left(\sum_{k\in\mathbb{Z}} \hat{A}_{n_0}\Big(.-\frac{k}{\alpha_1}\Big)\right) * \Big(M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h). \quad \text{(IV.16)}
\end{aligned}$$

Since $\hat{g}$ is concentrated around 0, the right-hand term in (IV.16) can only contain significant values, if A_{n_0} has frequency-components concentrated around $\beta_2 h$, hence we consider the case $k = 0$ separately and obtain

$$\begin{aligned}
\langle S_{\alpha_1}(A_{n_0}), M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle_{\ell^2(\mathbb{Z})} &= \Big(\hat{A}_{n_0} * M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h) \\
&+ \left(\sum_{k\in\mathbb{Z}\setminus\{0\}} \hat{A}_{n_0}\Big(.-\frac{k}{\alpha_1}\Big)\right) * \Big(M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h). \quad \text{(IV.17)}
\end{aligned}$$

It remains to bound the sum of aliases, i.e. the second term of Equation (IV.17):

$$\begin{aligned}
&\left|\left(\sum_{k\in\mathbb{Z}\setminus\{0\}} \hat{A}_{n_0}\Big(.-\frac{k}{\alpha_1}\Big)\right) * \Big(M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h)\right| = \\
&\left|\sum_r \left(\sum_{k\in\mathbb{Z}\setminus\{0\}} \hat{A}_{n_0}\Big(r-\frac{k}{\alpha_1}\Big)\right)\cdot \Big(M_{-\alpha_2 m}\hat{g}_2\Big)(\beta_2 h - r)\right| \leq \\
&\sum_r \sum_{k\in\mathbb{Z}\setminus\{0\}} \left|\hat{A}_{n_0}\Big(r-\frac{k}{\alpha_1}\Big)\right| \cdot \Big|\hat{g}_2(\beta_2 h - r)\Big| \quad \text{(IV.18)}
\end{aligned}$$

Using the assumption $\sum_{k\in\mathbb{Z}\setminus\{0\}} |\hat{A}_{n_0}(. - \frac{k}{\alpha_1})| \leq \varepsilon_{\alpha_1}$ and also the assumption on the analysis window g_2, namely the fast decay of $\hat{g}_2$ we obtain:

$$\sum_r \sum_{k\in\mathbb{Z}\setminus\{0\}} \left|\hat{A}_{n_0}\left(r - \frac{k}{\alpha_1}\right)\right| \cdot \left|\hat{g}_2(\beta_2 h - r)\right| \leq \varepsilon_{\alpha_1} \sum_r \left|\hat{g}_2(\beta_2 h - r)\right|$$
$$\leq \varepsilon_{\alpha_1} C_{\hat{g}_2} \sum_r \left(1 + |\beta_2 h - r|^s\right)^{-1}. \quad \text{(IV.19)}$$

We rewrite the first term in (IV.17) and make use of the operator $\mathcal{I}$ introduced in (IV.2):

$$\left(\hat{A}_{n_0} * M_{-\alpha_2 m}\hat{g}_2\right)(\beta_2 h) = \sum_r \hat{A}_{n_0}(r)\left(M_{-\alpha_2 m}\hat{g}_2\right)(\beta_2 h - r) =$$
$$\langle \hat{A}_{n_0}, T_{\beta_2 h}\mathcal{I}M_{-\alpha_2 m}\hat{g}_2\rangle = -\langle A_{n_0}, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle. \quad \text{(IV.20)}$$

The last Equation (IV.20) uses Plancherl's theorem. Rewriting the last term we obtain

$$-\langle A_{n_0}, M_{\beta_2 h}T_{\alpha_2 m}g_2\rangle = -\langle M_{-\beta_2 h}A_{n_0}, T_{\alpha_2 m}g_2\rangle.$$

□

Remark 5. *For sufficiently big s the sum $\sum_r \left(1 + |\beta_2 h - r|^s\right)^{-1}$ decreases fast, e.g. taking $s = 5$ the sum is approximately 2.*

The second layer output is obtained by applying the output-generating atom as in (IV.4).

Corollary 2 (Output of Layer 2)**.** *Let $\phi_2 \in \Psi_3$, then the output of the second layer is*

$$\left(U_2[\beta_2 h]U_1[\beta_1 j]f * \phi_2\right)(m) = \left(|\hat{g}_1(\beta_1 j - \xi_0 n_0)||\left\langle M_{-\beta_2 h}A_{n_0}, T_{\alpha_2 m}g_2\right\rangle| * \phi_2\right)(m) + \epsilon_2(m)$$

where

$$\epsilon_2(m) \leq \|E_2\|_\infty^2 \|\phi_2\|_1^2.$$

Here E_2 is the error of Proposition 2.

Remark 6. *Note that in the second layer, applying the output-generating atom $\phi_2 \in \Psi_3$ removes the fine temporal structure and thus, the second layer output reveals information contained in the envelopes A_n.*

Proof. Proof is similar to the first layer output, see Corollary 1.

□

IV.3.1.2 Deformation Stability

In this section we study to which extent GS is stable with respect to certain, small deformations. This question is interesting, since we may often intuitively assume, that the classification of natural signals, be it sound or images, is preserved under mild and possibly local deformations. For the signal class $\mathcal{T}$, we consider musically meaningful deformations and show stability of GS with respect to these deformations. We consider changes in spectral shape as well as frequency modulations. Note that, as opposed to the invariance properties derived in Section IV.3.1.1 for the output of specific layers, the derived stability results pertain to the entire feature vector obtained from the GS along all included layers, cp. the definition and derivation of deformation stability in [22]. The method we apply is inspired by [12] and uses the decoupling technique, i.e. in order to prove stability of the feature extractor we first take the structural properties of the signal class into account and search for an error bound of deformations of the signals in $\mathcal{T}$. In combination with the contractivity property $\|\Phi_\Omega(f) - \Phi_\Omega(h)\|_2 \leq \|f - h\|_2$ of Φ_Ω, see [28, Prop. 4], which follows from $B_\ell \leq 1 \ \ \forall \ell \in \mathbb{N}$, where B_ℓ is the upper frame bound of the Gabor frame $G(g_\ell, \alpha_\ell, \beta_\ell)$, this yields deformation stability of the feature extractor.
Simply deforming a tone would correspond to deformations of the envelope $A_n, \ n = 1, ..., N$. This corresponds to a change in timbre, for example by playing a note on a different instrument. Mathematically this can be expressed as: $\mathfrak{D}_{A_\tau}(f)(t) = \sum_{n=1}^{N} A_n\Big(t + \tau(t)\Big)e^{2\pi i n \xi_0 t}$.

Lemma 1 (Envelope Changes)**.** *Let $f \in \mathcal{T}$ and $|A_n'(t)| \leq C_n(1 + |t|^s)^{-1}$, for constants $C_n > 0, \ n = 1, ..., N$ and $s > 1$. Moreover let $\|\tau\|_\infty < \frac{1}{2}$. Then*

$$\|f - \mathfrak{D}_{A_\tau}(f)\|_2 \leq D\|\tau\|_\infty \sum_{n=1}^{N} C_n,$$

for $D > 0$ depending only on $\|\tau\|_\infty$.

Proof. Setting $h_n(t) = A_n(t) - \mathfrak{D}_{A_\tau}(A_n(t))$ we obtain

$$\|f - \mathfrak{D}_{A_\tau}(f)\|_2 \leq \sum_{n=1}^{N} \|h_n(t)\|_2.$$

We apply the mean value theorem for a continuous function $A_n(t)$ and get

$$|h_n(t)| \leq \|\tau\|_\infty \sup_{y \in B_{\|\tau\|_\infty}(t)} |A_n'(y)|.$$

Applying the $2-$norm on $h_n(t)$ and the assumption on $A_n'(t)$, we obtain:

$$\int_{\mathbb{R}} |h_n(t)|^2 dt \leq \int_{\mathbb{R}} \|\tau\|_\infty^2 \Big(\sup_{y \in B_{\|\tau\|_\infty}(t)} |A_n'(y)| \Big)^2 dt$$
$$\leq C_n^2 \|\tau\|_\infty^2 \int_{\mathbb{R}} \sup_{y \in B_{\|\tau\|_\infty}(t)} (1 + |y|^s)^{-2} dt.$$

Splitting the integral into $B_1(0)$ and $\mathbb{R} \backslash B_1(0)$ we obtain

$$\|h_n(t)\|_2^2 \leq C_n^2 \|\tau\|_\infty^2 \Big(\int_{B_1(0)} 1 dt + \int_{\mathbb{R}\backslash B_1(0)} \sup_{y \in B_{\|\tau\|_\infty}(t)} (1 + |y|^s)^{-2} dt \Big).$$

Using the monotonicity of $(1 + |y|^s)^{-1}$ and in order to remove the supremum, by shifting $\|\tau\|_\infty$, we have

$$\|h_n(t)\|_2^2 \leq C_n^2 \|\tau\|_\infty^2 \Big(\int_{B_1(0)} 1 dt + \int_{\mathbb{R}\backslash B_1(0)} (1 + ||t| - \|\tau\|_\infty|^s)^{-2} dt \Big).$$

Moreover for $t \notin B_1(0)$ we have $|(1 - \|\tau\|_\infty)t|^s \leq |(1 - \frac{\|\tau\|_\infty}{|t|})t|^s$. This leads to

$$\|h_n(t)\|_2^2 \leq C_n^2 \|\tau\|_\infty^2 \Big(2 + \int_{\mathbb{R}\backslash B_1(0)} (1 + |(1 - \|\tau\|_\infty)t|^s)^{-2} dt \Big).$$

Performing a change of variables, i.e. $x = (1 - \|\tau\|_\infty)t$ with $\frac{dx}{dt} = 1 - \|\tau\|_\infty > \frac{1}{2}$ we obtain

$$\|h_n(t)\|_2^2 \leq C_n^2 \|\tau\|_\infty^2 \Big(2 + 2 \int_{\mathbb{R}} (1 + |x|^s)^{-2} dx \Big)$$
$$= C_n^2 \|\tau\|_\infty^2 \Big(2 + 2 \Big\| \frac{1}{1 + |x|^s} \Big\|_2^2 \Big).$$

Setting $D^2 := 2\Big(1 + \|\frac{1}{1+|x|^s}\|_2^2\Big)$ and summing up we obtain

$$\|f - \mathfrak{D}_{A_\tau}(f)\|_2 \leq D \|\tau\|_\infty \sum_{n=1}^{N} C_n.$$

□

Remark 7. *Harmonics' energy decreases with increasing frequency, hence $C_n \ll C_{n-1}$, hence the sum $\sum_{n=1}^{N} C_n$ can be expected to be small.*

Another kind of sound deformation results from frequency modulation of $f \in \mathcal{T}$. This corresponds to, for example, playing higher or lower pitch, or producing a vibrato. This can be formulated as:

$$\mathfrak{D}_\tau : f(t) \mapsto \sum_{n=1}^{N} A_n(t) e^{2\pi i \left(n\xi_0 t + \tau_n(t) \right)}.$$

Lemma 2 (Frequency Modulation)**.** *Let $f \in \mathcal{T}$. Moreover let $\|\tau_n\|_\infty < \frac{\arccos(1-\frac{\varepsilon^2}{2})}{2\pi}$. Then*

$$\|f - \mathfrak{D}_\tau(f)\|_2 \leq \varepsilon \sum_{n=1}^{N} \frac{1}{n}.$$

Proof. We have

$$\|f - \mathfrak{D}_\tau f\|_2 \leq \sum_{n=1}^{N} \|h_n(t)\|_2,$$

with $h_n(t) = A_n(t)(1 - e^{2\pi i \tau_n(t)})$. Computing the $2-$norm of $h_n(t)$ we obtain:

$$\int_{\mathbb{R}} |h_n(t)|^2 dt = \int_{\mathbb{R}} |A_n(t)(1 - e^{2\pi i \tau_n(t)})|^2 dt \leq \|1 - e^{2\pi i \tau_n(t)}\|_\infty^2 \|A_n(t)\|_\infty^2.$$

We rewrite

$$|1 - e^{2\pi i \tau_n(t)}|^2 = \left|1 - \left(\cos\left(2\pi\tau_n(t)\right) + i \sin\left(2\pi\tau_n(t)\right)\right)\right|^2 = 2\left(1 - \cos\left(2\pi\tau_n(t)\right)\right).$$

Setting $\|1 - e^{2\pi i \tau_n(t)}\|_\infty^2 \leq \varepsilon^2$, this term gets small if $\|\tau_n(t)\|_\infty \leq \frac{\arccos(1-\frac{\varepsilon^2}{2})}{2\pi}$. Using the assumptions of our signal model on the envelopes, i.e. $\|A_n\|_\infty < \frac{1}{n}$, we obtain

$$\|f - \mathfrak{D}_\tau(f)\|_2 \leq \varepsilon \sum_{n=1}^{N} \frac{1}{n}.$$

□

Proposition 3 (Deformation Stability)**.** *Let $\Phi_\Omega : L^2(\mathbb{R}) \to (\ell^2(\mathbb{Z}))^{\mathcal{Q}}$, $f \in \mathcal{T}$ and $|A_n'(t)| \leq C_n(1 + |t|^s)^{-1}$, for constants $C_n > 0$, $n = 1, ..., N$ and $s > 1$. Moreover let $\|\tau\|_\infty < \frac{1}{2}$ and $\|\tau_n\|_\infty < \frac{\arccos(1-\frac{\varepsilon^2}{2})}{2\pi}$. Then the feature extractor Φ is deformation stable w.r.t.*

- *envelope changes $\mathfrak{D}_{A_\tau}\Big(f\Big)(t) = \sum_{n=1}^{N} A_n\Big(t + \tau(t)\Big)e^{2\pi i n \xi_0 t}$:*

$$\Big\|\Phi_\Omega\Big(f\Big) - \Phi_\Omega\Big(\mathfrak{D}_{A_\tau}(f)\Big)\Big\|_2 \leq D\|\tau\|_\infty \sum_{n=1}^{N} C_n,$$

  for $D > 0$ depending only on $\|\tau\|_\infty$.

- *frequency modulation $\mathfrak{D}_\tau\Big(f\Big)(t) = \sum_{n=1}^{N} A_n(t)e^{2\pi i\Big(n\xi_0 t + \tau_n(t)\Big)}$:*

$$\Big\|\Phi_\Omega\Big(f\Big) - \Phi_\Omega\Big(\mathfrak{D}_\tau(f)\Big)\Big\|_2 \leq \varepsilon \sum_{n=1}^{N} \frac{1}{n}.$$

Proof. The Proof follows directly from a result of [28, Prop. 4], called contractivity property $\|\Phi_\Omega(f) - \Phi_\Omega(h)\|_2 \leq \|f - h\|_2$ of Φ_Ω, which follows from $B_\ell \leq 1 \ \forall \ell \in \mathbb{N}$, where B_ℓ is the upper frame bound of the Gabor frame $G(g_\ell, \alpha_\ell, \beta_\ell)$ and deformation stability of the signal class in Lemma 1 and 2.

□

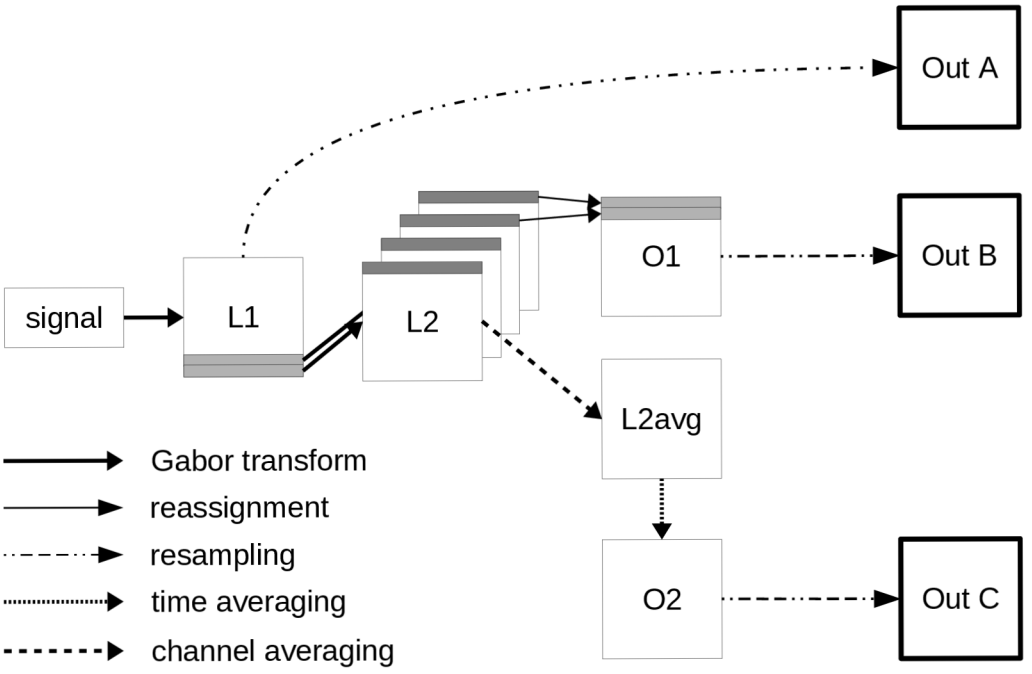


Fig. IV.2: Diagram explaining the naming of the GS building blocks of the Python implementation in the following sections.

IV.3.2 Visualization Example

In this section we present some visualizations based on two implementations, one in MATLAB, which we call the GS implementation and the other one in Python, which is the channel averaged GS implementation. The main difference between these implementations is an averaging step of Layer 2 in the case of the Python implementation; averaging over channels is introduced in order to obtain a 2D representation in each layer. Furthermore, the averaging step significantly accelerates the computation of the second layer output.

Referring to Figure IV.2 the following nomenclature will be used: Layer 1 (L1) is the GT, which, after resampling to the desired size, becomes Out A. Output 1 (O1) is the output of L1, i.e. after applying the output-generating atom. Recall that this is done by a low-pass filtering step. Again, Out B is obtained by resampling to the desired matrix size.

Layer 2 (L2) is obtained by applying another GT for each frequency channel. In the MATLAB code Output 2 (O2) is then obtained by low-pass filtering the separate channels of each resulting spectrogram. In the case of Python implementation (see Fig. IV.2), we average all the GT of L2 to one spectrogram (for the sake of speed) and then apply a time averaging step in order to obtain O2. Resampling to the desired size yields Out C.

As input signal for this section we generate single tones following the signal model from Section IV.2.2.

IV.3.2.1 Visualization of different frequency channels within the GS implementation

Figure IV.3 and IV.4, show two tones, both having a smooth envelope, but different fundamental frequencies and number of harmonics. The first tone has fundamental frequency $\xi_0 = 800$Hz and 15 harmonics and the second tone has fundamental frequency $\xi_0 = 1060$Hz and 10 harmonics.

Content of Figures IV.3 and IV.4:

- *Layer 1:* The first spectrogram of Figure IV.3 shows the GT. Observe the difference in the fundamental frequencies and that these two tones have a different number of harmonics, i.e. tone one has more than tone two.
- *Output 1:* The second spectrogram of Figure IV.3 shows Output 1, which is is time averaged version of Layer 1.
- *Output 2:* For the second layer output, (see Fig.IV.4) we take a fixed frequency channel from Layer 1 and compute another GT to obtain a Layer 2 element. By applying an output-generating atom, i.e. a low-pass filter, we obtain Output 2. Here we show, how different frequency channels of Layer 1 can affect Output 2.

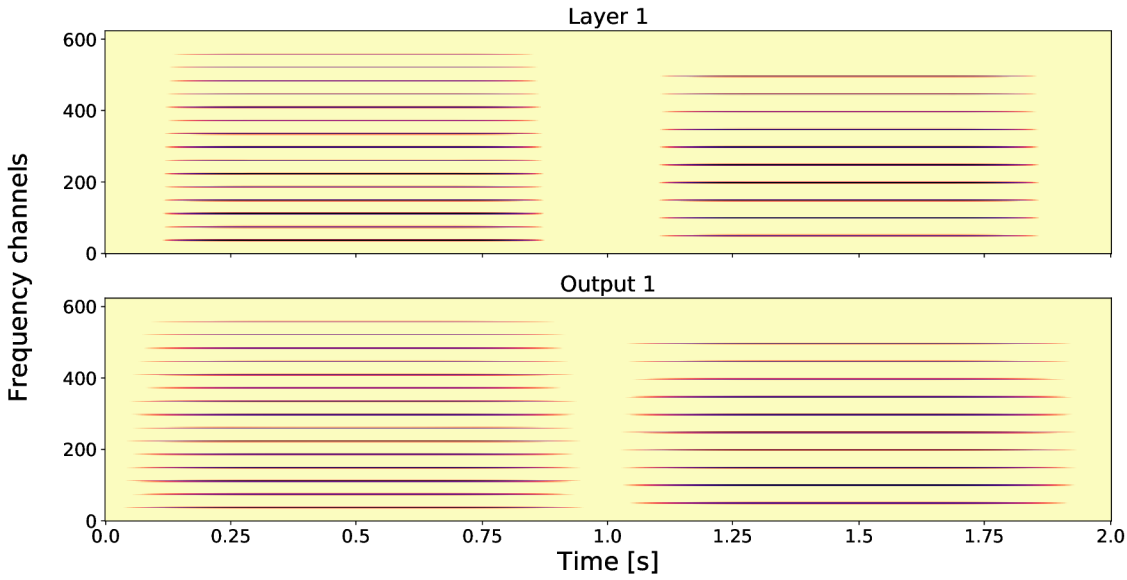


Fig. IV.3: First layer (i.e. GT) and Output 1 of two tones with different fundamental frequencies.

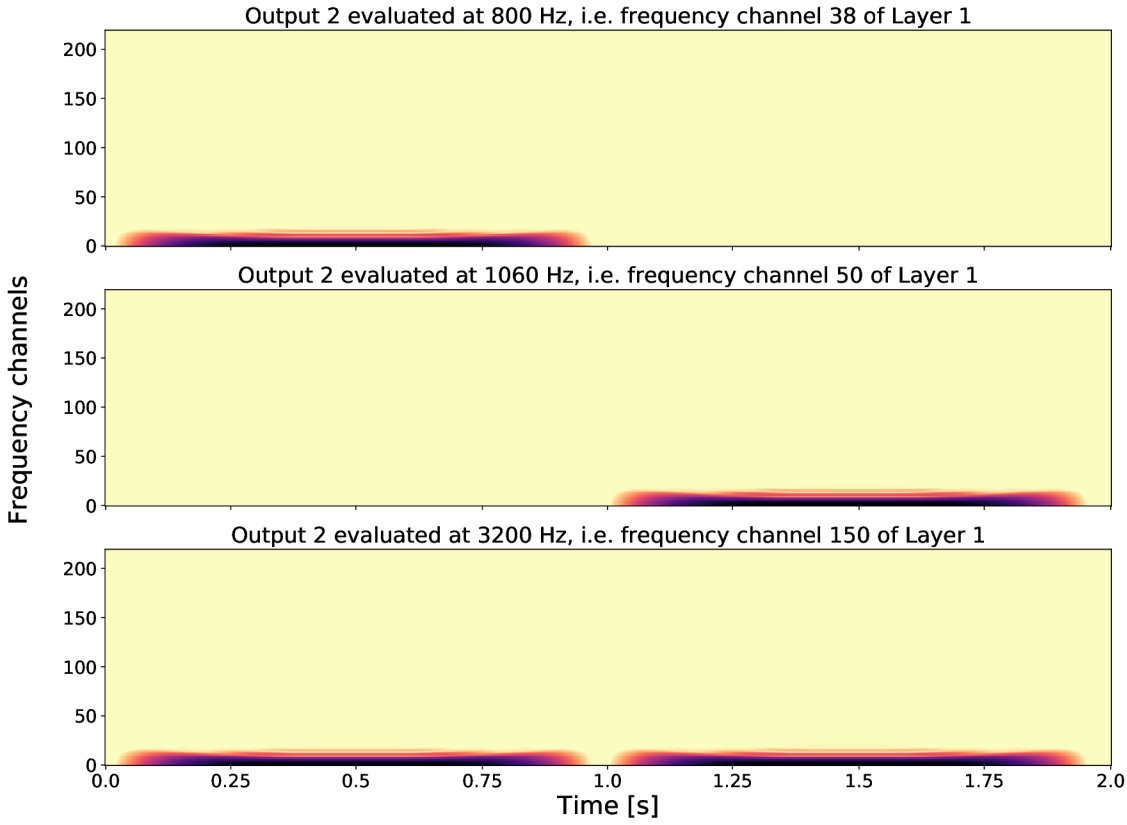


Fig. IV.4: Output 2 of two tones with different fundamental frequencies, at different fixed frequency channels of Layer 1.

The first spectrogram shows Output 2 w.r.t. the fundamental frequency of tone one, i.e. $\xi_0 = 800$Hz. Hence no second tone is visible in this output. On the other hand, in the second spectrogram, if we take as fixed frequency channel in Layer 1 the fundamental frequency of the second tone, i.e. $\xi_0 = 1060$Hz, in Output 2 the first tone is not visible. If we consider a frequency that both share, i.e. $\xi = 3200$Hz, we see that for Output 2 in the third spectrogram both tones are present. Since GS focuses on one frequency channel in each layer element, the frequency information in this layer is lost, in other words Layer 2 is invariant w.r.t. frequency.

IV.3.2.2 Visualization of different envelopes within the GS implementation

Here, Figure IV.5, shows two tones, played sequentially, having the same fundamental frequency $\xi_0 = 800$Hz and 15 harmonics, but different envelopes. The first tone has a sharp attack, maintains and goes softly to zero, the second starts with a soft attack and has some amplitude modulation. An amplitude modulated signal would for example correspond to $f(t) = \sum_{n=1}^{N} \sin(2\pi 20t)e^{2\pi i n \xi_0 t}$, here the signal is modulated by $20Hz$. The GS output of these signals are shown in Figure IV.5:

- *Layer 1:* In the spectrogram showing the GT, we see the difference between the envelopes and we see that the signals have the same pitch and the same harmonics.
- *Output 1:* The output of the first layer is invariant w.r.t. the envelope of the signals. This is due to the output-generating atom and the subsampling, which removes temporal information of the envelope. In this output no information about the envelope (neither the sharp attack nor the amplitude modulation) is visible, hence the spectrogram of the different signals look almost the same.
- *Output 2:* For the second layer output we took as input a time vector at fixed frequency of 800Hz (i.e. frequency channel 38) of the first layer. Output 2 is invariant w.r.t. the pitch, but differences on larger scales are captured. Within this layer we are able to distinguish the different envelopes of the signals. We first see the sharp attack of the first tone and then the modulation with a second frequency is visible.

The source code of the MATLAB implementation and further examples can be found in [14].

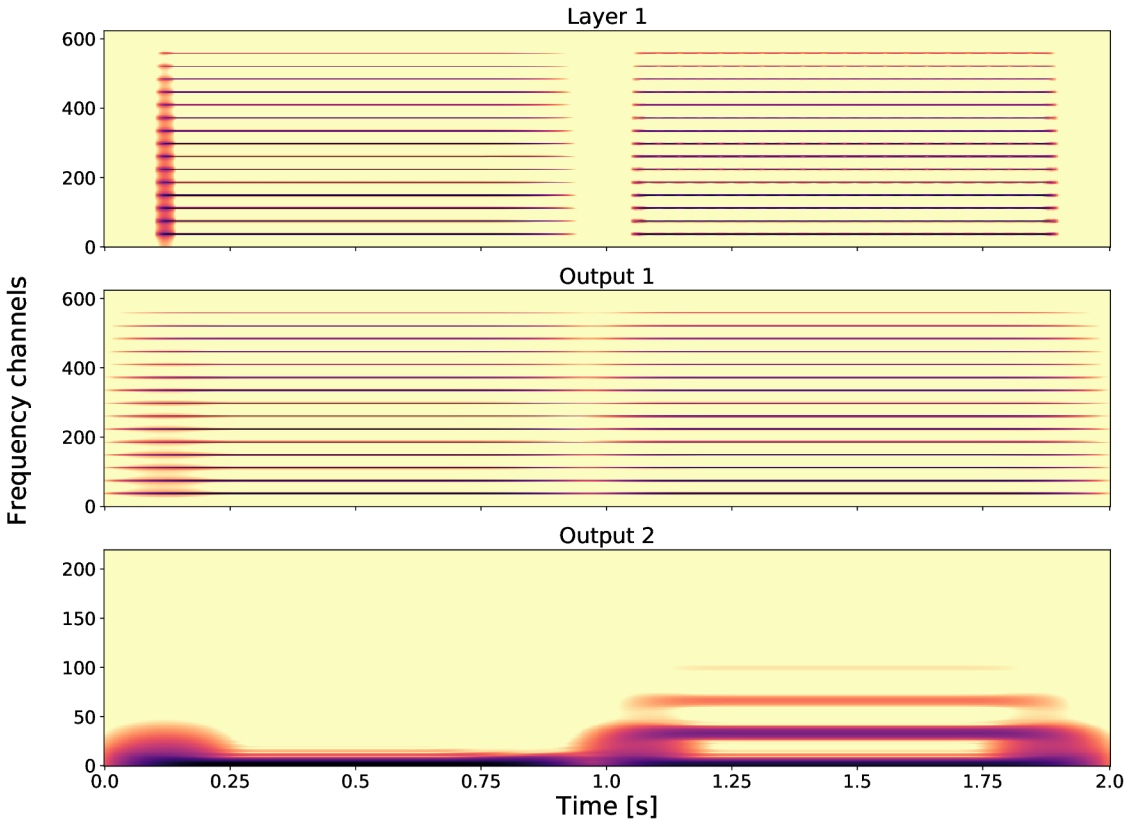


Fig. IV.5: Layer 1 (i.e. GT), Output 1 and Output 2 of the signal having a sharp attack and afterwards some modulation.

IV.3.2.3 Visualization of how frequency and amplitude modulations influence the outputs using the channel averaged implementation

In order to visualize the resampled transformation in a more structured way, we created an interactive plot (see Fig.IV.6), which shows 25 different synthetic audio signals side by side, transformed into Out A, Out B and Out C with chosen GS parameters. Each signal consists of one or more sine waves modulated in amplitude and frequency with 5 Hz steps.

The parameters can be adjusted by sliders and the plot is changed accordingly. The chosen parameters to be adjusted were number of frequency channels in Layer 1, number of frequency channels in Layer 2, sampling rate and number of harmonics of the signal. The code for the interactive plot is available as a part of the repository [14].

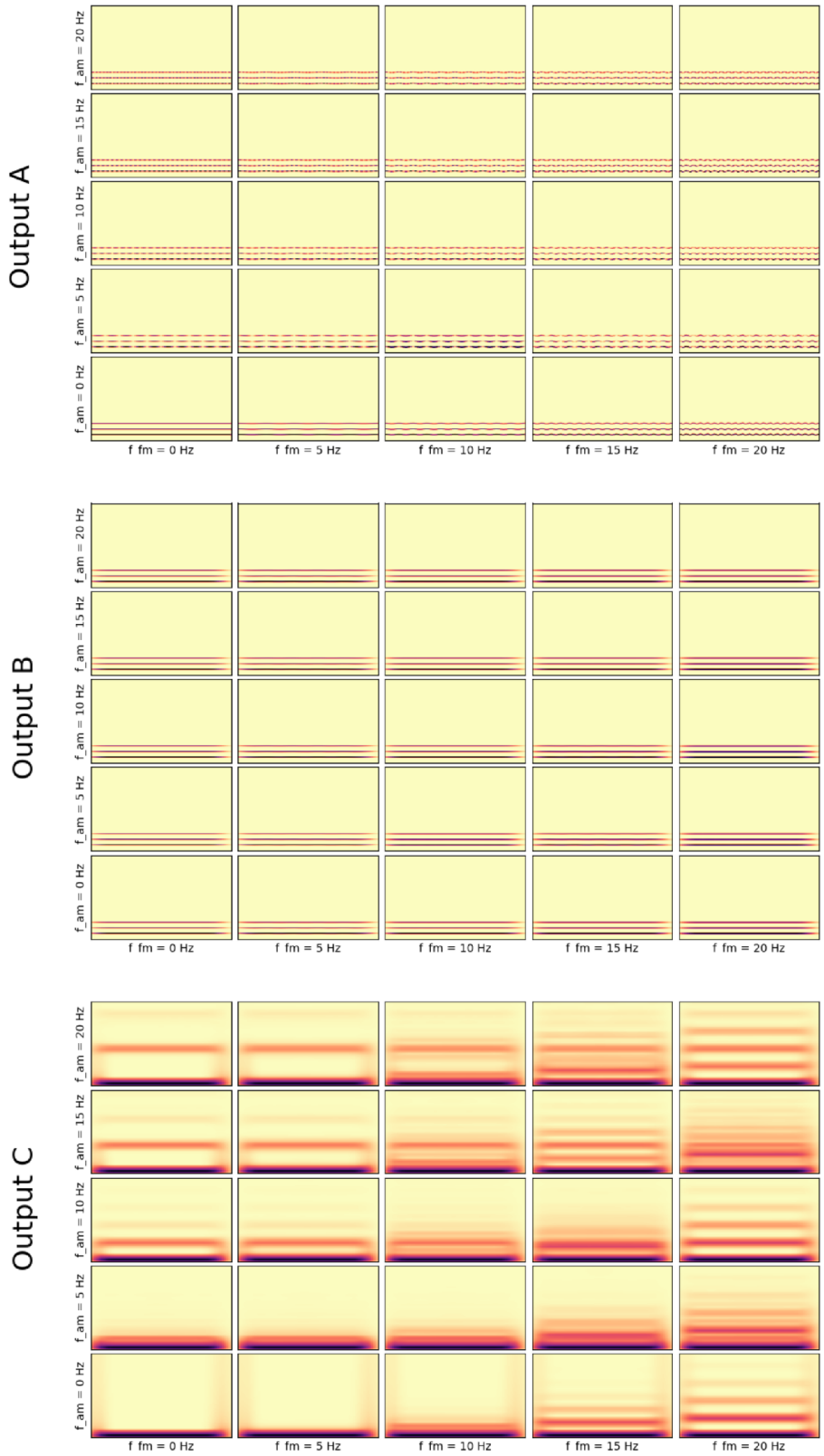


Fig. IV.6: Example output of the interactive plot.

IV.4 Experimental Results

In the numerical experiments, we compare the GS to a GT representation which is one of the standard time-frequency representations used in a preprocessing phase for training neural networks applied to audio data. We compare these two time-frequency representations w.r.t. the performance on limited size of the training data set.

To convert the raw waveform into the desired representations (GS and GT), we have used the Gabor-scattering v0.0.4 library [13], which is our Python implementation of the GS transform based on the Scipy v1.2.1 [17, 25, 9] implementation of STFT.

To demonstrate the beneficial properties of GS, we first create synthetic data in which we have the data generation under a full control. In this case we generate four classes of data that reflect the discriminating properties of GS. Secondly we investigate whether the GS representation is beneficial when using a "real" data set for training. For this purpose we have utilized the GoodSounds data set [26].

IV.4.1 Experiments with Synthetic Data

In the synthetic data set we created four classes, containing 1 second long signals, sampled at 44.1 kHz with 16 bit precision. All signals consist of a fundamental sine wave and four harmonics. The whole process of generating sounds is controlled by fixed random seeds for reproducibility.

IV.4.1.1 Data

We describe the sound generator model for one component of the final signal by the following equation:

$$f(t) = A \cdot \sin\Big(2\pi\Big(\xi t + cw_{fm}(t, A_{fm}, \xi_{fm}, \varphi_{fm})\Big) + \varphi\Big) \cdot cw_{am}(t, A_{am}, \xi_{am}, \varphi_{am}), \tag{IV.21}$$

where

$$cw_{fm}(t, A_{fm}, \xi_{fm}, \varphi_{fm}) \;=\; A_{fm} \cdot \sin(2\pi\xi_{fm} t + \varphi_{fm})$$

is the frequency modulation and

$$cw_{am}(t, A_{am}, \xi_{am}, \varphi_{am}) = \left\{ \begin{array}{ll} A_{am} \cdot \sin(2\pi\xi_{am} t + \varphi_{am}) & \text{if } A_{am} > 0 \text{ and } \Big(\varphi_{am} > 0 \text{ or } \xi_{am} > 0\Big) \\ 1 & \text{else} \end{array} \right\}$$

is the amplitude modulation. Here A is the amplitude, ξ denotes the frequency and φ denotes the phase. Furthermore, the amplitude, frequency and phase of the

frequency modulation carrier wave is denoted by A_{fm}, ξ_{fm} and φ_{fm} respectively and for the case of amplitude modulation carrier wave we have A_{am}, ξ_{am} and φ_{am}.

To generate five component waves using the sound generator described above, we needed to decide upon the parameters of each component wave. We started by randomly generating the frequencies and phases of the signal and the carrier waves for frequency and amplitude modulation from given intervals. These parameters describe the fundamental sine wave of the signal. Next we create harmonics by taking multiples (from 2 to 5) of the fundamental frequency ξ, where A of each next harmonic is divided by a factor. Afterwards, by permuting the two parameters, namely by turning the amplitude modulation and frequency modulation on and off, we defined four classes of sound. These classes are indexed starting from zero. The 0^{th} class has neither amplitude nor frequency modulation. Class 1 is just amplitude modulated, Class 2 is just modulated in frequency and Class 3 is modulated in both, amplitude and frequency, as seen in Table IV.1. At the end, we used those parameters to generate each harmonic separately and then summed them together to obtain the final audio file.

Table IV.1: Overview of classes.

	$A_{am} = 0$	$A_{am} = 1$
$A_{fm} = 0$	class 0	class 1
$A_{fm} = 1$	class 2	class 3

The following parameters were used to obtain GS: n_fft = 500 - number of frequency channels, n_perseg = 500 - window length, n_overlap = 250 - window overlap were taken for Layer 1, i.e. GT, n_fft = 50, n_perseg = 50, n_overlap = 40 for Layer 2, window_length of the time averaging window for Output 2 was set to 5 with mode set to 'same'. All the shapes for Output A, Output B and Output C were 240×160. Bilinear resampling [20] was used to adjust the shape if necessary. The same shape of all of the outputs allows the stacking of matrices into shape $3 \times 240 \times 160$, which is convenient for CNN, because it can be treated as a 3-channel image. Illustration of the generated sounds from all four classes transformed into GT and GS can be seen in Section IV.1.2 in Figure IV.1.
With the aforementioned parameters, the mean time necessary to compute the GS was 17.4890 ms, while the mean time necessary to compute the GT was 5.2245 ms, which is approximately 3 times less. It is important to say, that such comparison is only indicative, because the time is highly dependent on chosen parameters, hence the final time depends on the specific settings.

IV.4.1.2 Training

In order to compare the discriminating power of both GS and GT, we have generated 10 000 training samples (2 500 from each class) and 20 000 (5 000 from each class) validation samples. Since the task at hand is not as challenging as some real world data sets, we assume these sizes to be sufficient for both time-frequency representations to converge to very good performances. To compare the performance of GS and GT on a limited set of training data, we have altogether created four scenarios in which the training set was limited to $400, 1\,000, 4\,000$ and 10 000 samples. In all of these scenarios, the size of the validation set remained at its original size of 20 000 samples and we have split the training set into smaller batches each containing 100 samples with the same number of samples from each class. Batches were used to calculate the model error based on which the model weights were updated.

The CNN consisted of the batch normalization layer, that acted upon the input data separately for each channel of the image (we have 3 channels, namely Out A, Out B and Out C), followed by four stacks of 2D convolution with average pooling. The first three convolutional layers were identical in the number of kernels which was set to 16, of the size 3×3 with stride 1×1. The last convolutional layer was also identical apart from using just 8 kernels. Each convolutional layer was initialized by a Glorot uniform initialization [7] and followed by a ReLu nonlinearity [15] and an average pooling layer with a 2×2 pool size. After the last average pooling the feature maps were flattened and fully connected to an output layer with 4 neurons and a softmax activation function [8]. For more details about the networks architecture the reader should consult the repository [14]. There they also find the exact code in order to reproduce the experiment.

The network's categorical cross-entropy loss function was optimized using the Adam optimizer [19] with lr=0.001, β_1=0.9 and β_2=0.999. In order to have fair comparison, we limit each of the experiments in the terms of computational effort as measured by a number of weight updates during the training phase. One weight update is made after each batch. Each experiment with synthetic data was limited to 2 000 weight updates. To create the network, we have used Python 3.6 programming language with Keras framework v2.2.4 [6] on Tensorflow backend v1.12.0 [1]. To train the models, we have used two GPUs, namely NVIDIA Titan XP and NVIDIA GeForce GTX 1080 Ti on the OS Ubuntu 18.04 based system. Experiments are fully reproducible and can be obtained by running the code in the repository [14].

Table IV.2: Performance of the CNN trained using GS and GT data.

TF	N train	N valid	BWU	train	valid
GS	400	20 000	280	1.0000	0.9874
GT	400	20 000	292	1.0000	0.9751
GS	1 000	20 000	650	0.9990	0.9933
GT	1 000	20 000	1 640	1.0000	0.9942
GS	4 000	20 000	1 640	0.9995	0.9987
GT	4 000	20 000	1 720	0.9980	0.9943
GS	10 000	20 000	1 800	0.9981	0.9968
GT	10 000	20 000	1 800	0.9994	0.9985

Table notation:
TF – Time-frequency representation. N train and N valid – Number of samples in training and validation sets. BWU – Weight update after which the highest performance was achieved on the validation set. Train and valid – accuracy on training and validation sets.

IV.4.1.3 Results

The results are shown in Table IV.2, listing the accuracies of the model's best weight update on training and validation sets. The best weight update was chosen based on the performance on the validation set. More detailed tables of the results can be found in the aforementioned repository. In this experiment we did not use any testing set, because of the synthetic nature of the data. Accuracy is computed as a fraction of correct predictions to all predictions.

The most important observation is visible in Fig.IV.7 where it is shown that in the earlier phases of the training GS reaches higher accuracies after less weight updates than GT. This effect diminishes with bigger training sets and vanishes completely in case of 100 training batches. In case of very limited data i.e. with only 400 training samples, the results show that GS even outperformed GT. With more training samples, i.e. 1 000 and 4 000, the best performances of GT and GS are nearly the same. In this case we could hypothesize that the prior knowledge of the intrinsic properties of a time series signal shown by GS (in the invariances of Layer 1 and Layer 2) is not needed anymore and the network is able to learn the necessary transformation itself.

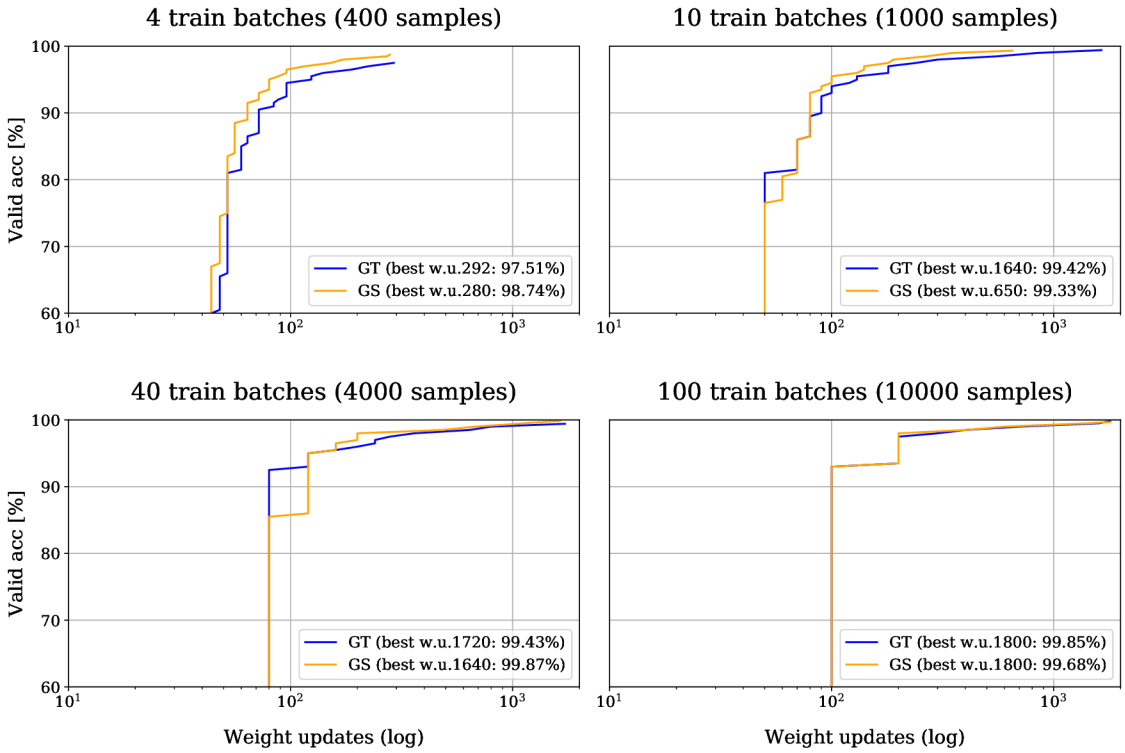


Fig. IV.7: CNN performance milestone reached over number of weight updates - Synthetic data.

Figure notation: Valid acc – Accuracy performance metric measured on the validation set. Best w.u. – Weight update after which the highest performance was reached.

IV.4.2 Experiments with GoodSounds Data

In the second set of experiments, we have used the GoodSounds data set [26]. It contains monophonic audio recordings of single tones or scales played by 12 different musical instruments. The main purpose of this second set of experiments is to investigate, whether GS shows superior performance to GT in a classification task using real-life data.

IV.4.2.1 Data

To transform the data into desired form for training, we removed the silent parts using the SoX v14.4.2 library [4, 24], next we have split all files into 1 s long segments sampled at a rate of 44.1 kHz with 16 bit precision. A Tukey window was applied to all segments to smooth the onset and the offset of each with the aim to prevent undesired artifacts after applying the STFT.

The data set contains 28.55 hours of recordings, which is a reasonable amount of audio data to be used in training of Deep Neural Networks considering the nature

of this task. Unfortunately, the data are distributed into classes unevenly, half of the classes are extremely underrepresented, i.e. half of the classes together contain only 12.6% of all the data. In order to alleviate this problem, we decided upon an equalization strategy by variable stride.
To avoid extensive equalization techniques, we have discarded all classes that spanned less than 10% of the data. In the end we have used 6 classes, namely clarinet, flute, trumpet, violin, sax alto and cello. To equalize the number of segments between these classes, we introduced aforementioned variable stride, when creating the segments. The less data a particular class contains, the bigger is the overlap between segments, thus more segments are generated and vice versa. The whole process of generating sounds is controlled by fixed random seeds for reproducibility. Detailed information about the available and used data, stride settings for each class, obtained number of segments and their split can be seen in Table IV.3.

Table IV.3: Overview of available and used data.

		All available data			Obtained segments			
	Class	**Files**	**Dur**	**Ratio**	**Stride**	**Train**	**Valid**	**Test**
Used	Clarinet	3 358	369.70	21.58%	37 988	12 134	4 000	4 000
	Flute	2 308	299.00	17.45%	27 412	11 796	4 000	4 000
	Trumpet	1 883	228.76	13.35%	22 826	11 786	4 000	4 000
	Violin	1 852	204.34	11.93%	19 836	11 707	4 000	4 000
	Sax alto	1 436	201.20	11.74%	19 464	11 689	4 000	4 000
	Cello	2 118	194.38	11.35%	15 983	11 551	4 000	4 000
Not used	Sax tenor	680	63.00	3.68%				
	Sax soprano	668	50.56	2.95%				
	Sax baritone	576	41.70	2.43%				
	Piccolo	776	35.02	2.04%				
	Oboe	494	19.06	1.11%				
	Bass	159	6.53	0.38%				
	Total	16 308	1 713.23	100.00%		70 663	24 000	24 000

Table notation: Files – Number of available audio files. Dur – Duration of all recordings within one class in minutes. Ratio – Ratio of the duration to total duration of all recordings in the data set. Stride – Step size (in samples) used to obtain segments of the same length. Train, Valid, Test – Number of segments used to train (excluding the leaking segments), validate and test the model.

As can be seen in the table, the testing and validation sets were of the same size comprising the same number of samples from each class. The remaining samples were used for training. To prevent leaking of information from validation and testing sets into training set, we have excluded all the training segments originating from the audio excerpts, which were already used in validation or testing set. More information can be found in the repository [14].
The following parameters were used to obtain GS: n_fft = 2000 - number of frequency channels, n_perseg = 2000 - window length, n_overlap = 1750 - window overlap were taken for Layer 1, i.e. GT, n_fft = 25, n_perseg = 25, n_overlap = 20 for Layer 2, window_length of the time averaging window for Output 2 was set to 5 with mode set to 'same'. All the shapes for Output A, Output B and Output C were 480×160. Bilinear resampling [20] was used to adjust the shape if necessary. The same shape of all the outputs allows the stacking of matrices into shape $3 \times 480 \times 160$. Illustration of the sounds from all six classes of musical instruments transformed into GT and GS can be found in the repository [14].

IV.4.2.2 Training

In order to make the experiments on synthetic data and the experiments on GoodSounds data comparable, we have again used the CNN as a classifier trained in a similar way as described in Section IV.4.1.2. We have also pre-processed the data, so the audio segments are of the same duration and sampling frequency. However, musical signals have different distribution of frequency components than the synthetic data, therefore we had to adjust the parameters of the time-frequency representations. This led to a change in the input dimension to $3 \times 480 \times 160$. These changes and the more challenging nature of the task led to slight changes in the architecture:
The number of kernels in the first three convolutional layers was raised to 64. The number of kernels in the last convolutional layer was raised to 16. The output dimension of this architecture was set to 6, since this was the number of classes. The batch size changed to 128 samples per batch. Number of weight updates was set to 11 000. To prevent unnecessary training, this set of experiments was set to terminate after 50 consecutive epochs without an improvement in validation loss as measured by categorical crossentropy. The loss function and optimization algorithm remained the same as well as the used programming language, framework and hardware. Experiments are fully reproducible and can be obtained by running the code in the repository [14]. Consider this repository also for more details about the networks architecture.
In this set of experiments, we have trained 10 models in total, 5 scenarios with

limited training set (5, 11, 55, 110 and 550 batches each containing 128 samples) for each time-frequency representation. In all of these scenarios, the sizes of the validation and testing sets remained at their full sizes each consisting of 188 batches containing 24 000 samples.

IV.4.2.3 Results

Table IV.4 shows the accuracies of the model's best weight update on training, validation and testing sets. The best weight update was chosen based on the performance on the validation set. As before, more details can be found in the aforementioned repository. In this experiment using GoodSounds data, a similar trend as for the synthetic data is visible. GS performs better than GT if we are limited in training set size, i.e. having 640 training samples, the GS overperformed GT.

Table IV.4: Performance of CNN - GoodSounds data.

TF	N train	N valid	N test	BWU	train	valid	test
GS	640	24000	24000	485	0.9781	0.8685	0.8748
GT	640	24000	24000	485	0.9766	0.8595	0.8653
GS	1408	24000	24000	1001	0.9773	0.9166	0.9177
GT	1408	24000	24000	1727	0.9943	0.9194	0.9238
GS	7040	24000	24000	9735	0.9996	0.9846	0.9853
GT	7040	24000	24000	8525	0.9999	0.9840	0.9829
GS	14080	24000	24000	10780	0.9985	0.9900	0.9900
GT	14080	24000	24000	9790	0.9981	0.9881	0.9883
GS	70400	24000	24000	11000	0.9963	0.9912	0.9932
GT	70400	24000	24000	8800	0.9934	0.9895	0.9908

Table notation:
TF – Time-frequency representation. N train, N valid and N test – Number of samples in training, validation and testing sets. BWU – Weight update after which the highest performance was achieved on the validation set. Train, valid and test – accuracy on training, validation and testing sets.

In Fig.IV.8 we again see that in earlier phases of the training, GS reaches higher accuracies after less weight updates than GT. This effect diminishes with bigger training sets and vanishes in case of 550 training batches.

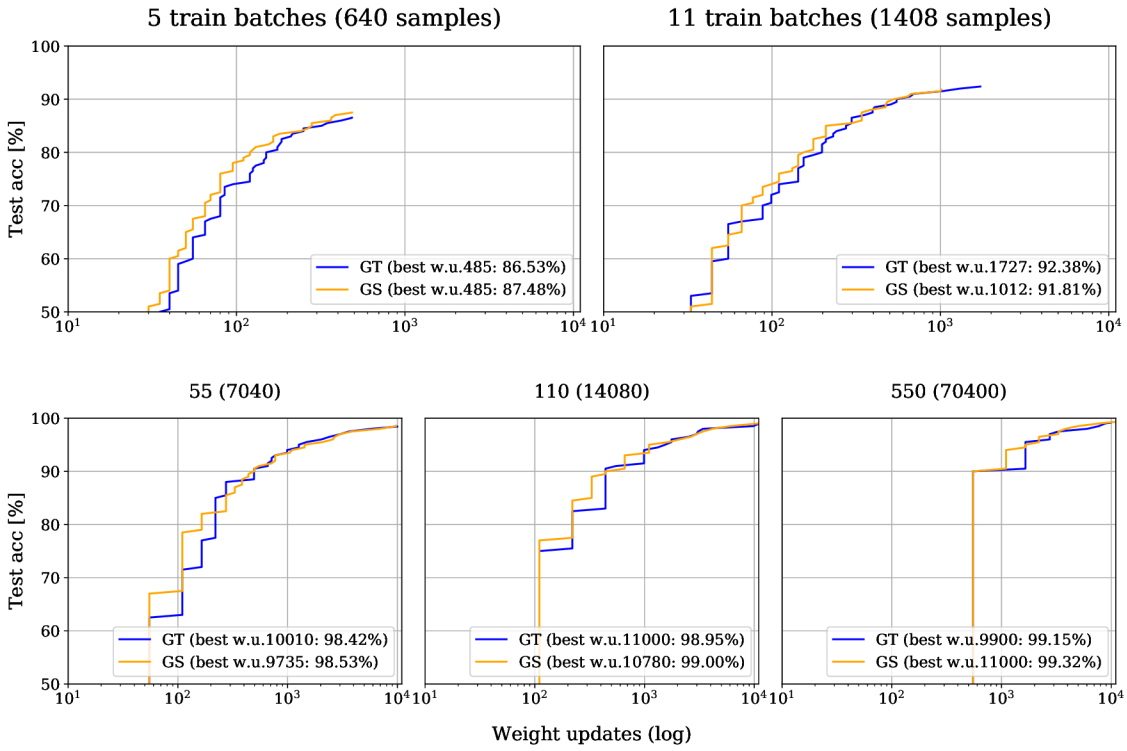


Fig. IV.8: CNN performance milestone reached over number of weight updates - GoodSounds data.

Figure notation: Test acc – Accuracy performance metric measured on the testing set. Best w.u. – Weight update after which the highest performance was reached.

IV.5 Discussion and Future Work

In the current contribution, a scattering transform based on Gabor frames has been introduced and its properties investigated by relying on a simple signal model. Thereby, we have been able to mathematically express the invariances introduced by GS within the first two layers.

The hypothesis raised in Section IV.1.1, that explicit encoding of invariances by using an adequate feature extractor is beneficial when a restricted amount of data is available was substantiated in the experiments presented in the previous section. It was shown that in the case of a limited data set the application of a GS representation improves the performance in classification tasks in comparison to using GT. In the current implementation and with parameters described in Section IV.4.1.1, the GS is approximately 3 times more expensive to compute than GT. However, this transformation needs to be done only once - in the preprocessing phase. Hence, the majority of computational effort is still spent during training. E.g. in case of GoodSounds experiment, the training with GS is about 2.5 times longer than with

GT. We need to point out that this is highly dependent on the used data handling pipeline, network architecture, software framework and hardware, which all can be optimized to alleviate this limitation. While GS is more computationally expensive, the obtained improvement justifies its use in certain scenarios, in particular for classification tasks which can be expected to benefit from the invariances introduced by GS. In these cases, the numerical experiments have shown that by using GS instead of GT a negative effect of a limited data set can be compensated.

Hypothetically, with enough training samples, both GS and GT should perform equally assuming sufficient training, i.e. performing enough weight updates. This is shown in the results of both numerical experiments presented in this article (see Tables IV.2 and IV.4). This is justified by the fact that GS comprises exclusively the information contained within GT, only separated into 3 different channels. We assume it is easier for the network to learn from such a separated representation. The evidence to support this assumption is visible in the earlier phases of the training, where GS reaches higher accuracies after less weight updates than GT (see Fig.IV.7 and Fig.IV.8). This effect increases with smaller data sets while with very limited data GS even surpasses GT in performance. This property can be utilized in restricted settings, e.g. in embedded systems with limited resources or in medical applications, where sufficient data sets are often too expensive or impossible to gather, while the highest possible performance is crucial.

We believe that GT would eventually reach the same performance as GS even on the smallest feasible data sets, but the network would need more trainable parameters, i.e. more complex architecture to do the additional work of finding the features that GS already provides. Unfortunately in such a case it remains problematic to battle the overfitting problem. This opens a new question - whether the performance boost of GS would amplify on lowering the number of trainable parameters of the CNN. This is out of the scope of this article and will be addressed in the future work.

In another paper [5], we extended GS to mel-scattering (MS), where we used GS in combination with a mel-filterbank. This MS representation reduces the dimensionality and hence it is computationally less expensive compared to GS.

It remains to be said, that the parameters in computing GS coefficients have to be carefully chosen in order to exploit the beneficial properties of GS by systematically capturing data-intrinsic invariances.

Future work will consist in implementing GS on the GPU, to allow for fast parallel computation. At the same time, more involved signal models, in particular concerning long-term correlations, will be studied analytically to the end of achieving results in the spirit of the theoretical results presented in this paper.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, ..., and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: `https://www.tensorflow.org/`.

[2] J. Andén, V. Lostanlen, and S. Mallat. Joint time-frequency scattering for audio classification. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2015.

[3] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

[4] C. Bagwell. SoX - Sound Exchange the swiss army knife of sound processing. `https://launchpad.net/ubuntu/+source/sox/14.4.1-5`. Accessed: 2018-10-31.

[5] R. Bammer, A. Breger, M. Dörfler, P. Harar, and Z. Smékal. Machines listening to music: the role of signal representations in learning from music. *CoRR*, abs/1903.08950, 2019. URL: `http://arxiv.org/abs/1903.08950`, `arXiv:1903.08950`.

[6] F. Chollet et al. Keras. `https://keras.io`, 2015.

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[9] D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[10] T. Grill and J. Schlüter. Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[11] K. Gröchenig. *Foundations of time-frequency analysis*. Applied and numerical harmonic analysis. Birkhäuser, Boston, Basel, Berlin, 2001.

[12] P. Grohs, T. Wiatowski, and H. Bölcskei. Deep convolutional neural networks on cartoon functions. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1163–1167, July 2016. doi:10.1109/ISIT.2016.7541482.

[13] P. Harar. Gabor scattering v0.0.4. https://gitlab.com/paloha/gabor-scattering, 2019.

[14] P. Harar and R. Bammer. gs-gt. https://gitlab.com/hararticles/gs-gt, 2019.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[16] T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, June 2008.

[17] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2019-02-01]. URL: http://www.scipy.org/.

[18] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] E. J. Kirkland. Bilinear interpolation. In *Advanced Computing in Electron Microscopy*, pages 261–263. Springer, 2010.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[22] S. Mallat. Group Invariant Scattering. *Comm. Pure Appl. Math.*, 65(10):1331–1398, 2012.

[23] S. Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 374(2065), 2016. URL:

http://rsta.royalsocietypublishing.org/content/374/2065/20150203, arXiv:http://rsta.royalsocietypublishing.org/content/374/2065/20150203.full.pdf, doi:10.1098/rsta.2015.0203.

[24] J. Navarrete. The SoX of Silence tutorial. https://digitalcardboard.com/blog/2009/08/25/the-sox-of-silence. Accessed: 2018-10-31.

[25] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[26] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, H. Tokuda, W. Hariya, K. Oishi, and X. Serra. A real-time system for measuring sound goodness in instrumental sounds. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[27] T. Wiatowski and H. Bölcskei. Deep Convolutional Neural Networks Based on Semi-Discrete Frames. In *Proc. of IEEE International Symposium on Information Theory (ISIT)*, pages 1212–1216, june 2015.

[28] T. Wiatowski and H. Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *CoRR*, abs/1512.06293, 2015. URL: http://arxiv.org/abs/1512.06293.

[29] T. Wiatowski, M. Tschannen, A. Stanic, P. Grohs, and H. Bölcskei. Discrete deep feature extraction: A theory and new architectures. In *Proc. of International Conference on Machine Learning (ICML)*, june 2016.

[30] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. 2017. URL: https://arxiv.org/abs/1611.03530.

V Improving Machine Hearing on Limited Data Sets

Outline

Bibliographic information

P. Harar, R. Bammer, A. Breger, M. Dörfler, and Z. Smekal. Improving machine hearing on limited data sets. In *2019 The 11th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*. IEEE, in press. `arXiv:1903.08950`.

Author's contribution

The author preprocessed the data, designed and conducted the numerical experiments and prepared the visualizations. He wrote sections Numerical Experiments and Discussion and Conclusions and contributed to Introduction. Helped reviewing each section of the article and organized the finalization of the paper.

Copyright Notice

Abstract

Convolutional neural network (CNN) architectures have originated and revolutionized machine learning for images. In order to take advantage of CNNs in predictive modeling with audio data, standard FFT-based signal processing methods are often applied to convert the raw audio waveforms into an image-like representations (e.g. spectrograms). Even though conventional images and spectrograms differ in their feature properties, this kind of pre-processing reduces the amount of training data necessary for successful training. In this contribution we investigate how input and target representations interplay with the amount of available training data in a music information retrieval setting. We compare the standard mel-spectrogram inputs with a newly proposed representation, called Mel scattering. Furthermore, we investigate the impact of additional target data representations by using an augmented target loss function which incorporates unused available information. We observe that all proposed methods outperform the standard mel-transform representation when using a limited data set and discuss their strengths and limitations. The source code for reproducibility of our experiments as well as intermediate results and model checkpoints are available in an online repository.

Acknowledgment

This work was supported by the Uni:docs Fellowship Programme for Doctoral Candidates in Vienna, by the Vienna Science and Technology Fund (WWTF) projects SALSA (MA14-018) and CHARMED (VRG12-009), by the International Mobility of Researchers (CZ.02.2.69/0.0/0.0/16 027/0008371), and by the project LO1401. Infrastructure of the SIX Center was used for computation.

V.1 Introduction

Convolutional neural networks (CNNs) [11], a class of deep neural networks (DNNs) architectures, originated in image processing and have revolutionized computer vision. The idea of CNNs is the introduction of locality and weight-sharing in the first layers of a DNN, i.e. using convolutional layers. This leads to the extraction of local patterns, which are searched for over the entire image using the same filter kernels. By intermediate pooling operators, the extension of the local search increases across the layers and additionally introduces stability to local deformations, [13].

Using the principles of CNNs in computer vision to solve problems in machine hearing, including music information retrieval (MIR), has equally led to surprising successes in various applications. However, the data processing pipeline needs to be altered: the actual signal of interest, the raw audio signal, is not directly used as input to the network. Usually, it is first pre-processed into an image, allowing for a time-frequency interpretation. Typical representations include the spectrogram or modifications thereof. This step leads to a reduction of data needed for training [16].

In this paper we improve the performance of CNNs, which are trained with the standard mel-spectrogram (MT) [1] input representation and limited amount of training data. To do so, we propose an alternative input representation called *Mel scattering* (MS), which uses the main concept of *Gabor scattering* (GS), introduced in [2], in combination with a mel-filter bank. Moreover, we improve the learning results by transforming the target space within an *augmented target loss function* (AT), introduced in [3].

The paper is organized as follows: In Section V.2 we introduce the learning setup and the data used in the numerical experiments. In Section V.3 we present the MT, and proceed to the definitions of GS and MS. AT is explained in Section V.4. In Section V.5 we compare the results of the proposed representations by evaluating the classification results of an instrumental sounds data set, serving as a toy data set for experiments with different amount of training data.

V.2 Learning from Data

Let $\mathcal{D} \subset \mathcal{X}$ be a data set in an input space $\mathcal{X}$, together with some information about the data, often called "annotation", which is given in the target space and denoted by $\mathcal{T} \subset \mathcal{Y}$. Learning the relationship between $\mathcal{D}$ and their annotations in $\mathcal{Y}$ can then be understood as looking for a function $\psi : \mathcal{X} \mapsto \mathcal{Y}$, which describes with sufficient accuracy the desired mapping. The accuracy is usually measured by a loss function, which is optimized in each iteration step of the training process to update

[1] We abbreviate with MT, i.e. "mel-transform", in order not to collide with further abbreviations.

the weights. Once the learning process is finished, e.g. via a stopping criterion, this results in a parameter vector θ determining a particular model within the previous determined architecture.

Further, given a hypothesis space parametrized by θ, and a set of annotated data $\mathcal{Z}_m = \mathcal{D} \times \mathcal{T} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, we learn a model ψ_θ. Let the estimated targets be denoted by $\hat{y}_i = \psi_\theta(x_i)$ and define the empirical loss function $E_{\mathcal{Z}_m}$ as

$$E_{\mathcal{Z}_m}(\psi_\theta) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, \hat{y}_i).$$

Common, important examples of loss functions include the quadratic loss $L(y_i, \hat{y}_i) = (\hat{y}_i - y_i)^2$, and the categorical cross-entropy loss (CE). The latter is the concatenation of the softmax function on the output vector $\hat{\mathbf{y}} = (\psi_\theta(x_1), \ldots, \psi_\theta(x_m))$ and the cross-entropy loss; in other words, in the case of categorical cross-entropy, we have

$$L(y_i, \hat{y}_i) = -y_i \log \frac{e^{\hat{y}_i}}{\sum_{j=1}^{m} e^{\hat{y}_j}}.$$

V.2.1 Data Set used for Experiments

For the classification experiments presented in Section V.5, the GoodSounds data set [17] is used. It contains monophonic recordings of single notes or scales played by different instruments. From each file, we have removed the silence with SoX v14.4.2 library[2]. The output rate was set to 44.1 kHz with 16 bit precision. We have split each file into segments of the same duration (1 s = 44 100 samples) and applied a Tukey window in order to smooth the onset and offset of the segment, thus preventing the undesired artifacts after applying the short-time Fourier transform (STFT). Since the classes were not equally represented in the data set, we needed to introduce an equalization strategy. To avoid extensive equalization techniques, we have used only classes which spanned at least 10% of the whole data set, namely clarinet, flute, trumpet, violin, alto saxophone and cello. More precisely, during the process of cutting the audio samples into 1 s segments, we introduce increased overlap for instrument recordings with fewer samples, thus utilizing a variable stride. This resulted in oversampling in underrepresented classes by overlapping the segments.

V.3 Time-Frequency Representations of Audio

Classical audio pre-processing tools such as the mel-spectrogram rely on some localized, FFT-based analysis. The idea of the resulting time-frequency representation

[2] https://launchpad.net/ubuntu/+source/sox/14.4.1-5

is to separate the variability in the signal with respect to time and frequency, respectively. However, for audio signals which are relevant to human perception, such as music or speech, significant variability happens on very different time-levels: the frequency content itself can be determined within a few milliseconds. Variations in the amplitude of certain signal components, e.g. formants or harmonics, have a much slower frequency and should be measured on the scale of up to few seconds. Longer-term musical developments, which allow, for example, to determine musical style or genre, happen on time-scales of more than several seconds. The basic idea of Gabor Scattering, as introduced in [2], see Section V.3.2, is to capture the relevant variability at different time-scales and separate them in various layers of the representation.

We first recall (mel-)spectrograms and turn to the definition of the scattering transforms in Section V.3.2.

V.3.1 Spectrograms and Mel-Spectrograms

Standard time-frequency representations used in audio-processing are based on STFT. Since we are interested in obtaining several layers of time-frequency representations, we define STFT as *frame-coefficients* with respect to time-frequency-shifted versions of a basic window. To this end, we introduce the following operators in some Hilbert space $\mathcal{H}$.

- The translation (time shift) operator for a function $f \in \mathcal{H}$ and $t \in \mathbb{R}$ is defined as $T_x f(t) := f(t-x)$ for all $x \in \mathbb{R}$.
- The modulation (frequency shift) operator for a function $f \in \mathcal{H}$ and $t \in \mathbb{R}$ is defined as $M_\omega f(t) := e^{2\pi i t\omega} f(t)$ for all $\omega \in \mathbb{R}$.

Now the STFT $V_g f$ of a function $f \in \mathcal{H}$ with respect to a window function $g \in \mathcal{H}$ can be easily seen to be $V_g f(x,\omega) = \langle f, M_\omega T_x g\rangle$ with the corresponding spectrogram $|V_g f(x,\omega)|^2$. The set of functions

$$G(g,\alpha,\beta) = \{M_{\beta j} T_{\alpha k} g : (\alpha k, \beta j) \in \Lambda\}$$

is a the Gabor system and is called Gabor frame [6], if there exist positive frame bounds $A, B > 0$ such that for all $f \in \mathcal{H}$

$$A\|f\|^2 \leq \sum_k \sum_j |\langle f, M_{\beta j} T_{\alpha k} g\rangle|^2 \leq B\|f\|^2. \tag{V.1}$$

Subsampling $V_g f$ on a separable lattice $\Lambda = \alpha\mathbb{Z}\times\beta\mathbb{Z}$ we obtain the frame-coefficients of f w.r.t $G(g,\alpha,\beta)$. Choosing Λ thus corresponds to picking a particular hop size in time and a finite number of frequency channels.

The mel-spectrogram $MS_g(f)$ is defined as the result of weighted averaging $|V_g f(\alpha k, \beta j)|^2$:

$$MS_g(f)(\alpha k, \nu) = \sum_j |V_g f(\alpha k, \beta j)|^2 \cdot \Upsilon_\nu(j),$$

where Υ_ν are the mel-filters for $\nu = 1, ..., K$ with K filters.

V.3.2 Gabor Scattering and Mel Scattering

We next introduce a new feature extractor called Gabor scattering, inspired by Mallat's scattering transform [12] and first introduced in [2]. In this contribution, we further extend the idea of Gabor-based scattering by adding a mel-filtering step in the first layer. The resulting transform is called Mel scattering. Since the number of frequency channels is significantly reduced by applying the filter bank, the computation of MS is considerably faster. GS is a feature extractor for audio signals, obtained by an iterative application of Gabor transforms (GT), a non-linearity in the form of a modulus function and pooling by sub-sampling in each layer. Since most of the energy and information of an input signal is known to be captured in the first two layers, cp. [1], we only introduce and use the output of those first layers, while in principle scattering transforms allow for arbitrarily many layers. In [2], it was shown that the output of specific layers of GS lead to invariances w.r.t. certain signal properties.

Coarsely speaking, the output of the first layer is invariant w.r.t. envelope changes and mainly captures the frequency content of the signal, while the second layer is invariant w.r.t. frequency and contains information about the envelope. For more details on GS and a mathematical description of its invariances see [2].

In the following, since we deal with discrete, finite signals f, we let $\mathcal{H} = \mathbb{C}^{\mathcal{L}}$, where $\mathcal{L}$ is the signal length, and $f_\ell \in \mathbb{C}^{\mathcal{L}_\ell}$ for $\ell = 1, 2$. The lattice parameters of the GT, i.e. $\Lambda_\ell = \alpha_\ell \mathbb{Z} \times \beta_\ell \mathbb{Z}$, can be chosen differently for each layer.

The first layer, which is basically a GT, corresponds to

$$f_1[\beta_1 j](k) = |\langle f, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle|, \tag{V.2}$$

and the second layer can be written as

$$f_2[\beta_1 j, \beta_2 h](m) = |\langle f_1[\beta_1 j], M_{\beta_2 h} T_{\alpha_2 m} g_2 \rangle|. \tag{V.3}$$

Note that the input function of the second layer is f_1, where the next GT is applied separately to each frequency channel $\beta_1 j$. To obtain the *output* of one layer, one needs to apply an output generating atom ϕ_ℓ, cp. [2]:

$$f_\ell[\beta_1 s, ..., \beta_\ell j] * \phi_\ell(k) = |\langle f_{\ell-1}, M_{\beta_\ell j} T_{\alpha_\ell k} g_1 \rangle| * \phi_\ell, \tag{V.4}$$

for $\ell \in \mathbb{N}$ in general and in our case $\ell = 1, 2$.

The output of the feature extractor is the collection of these coefficients (V.4) in one vector, which is used as input to a machine learning task. Based on the GS we want to introduce an additional mel-filtering step. The idea is to reduce the redundancy in spectrogram by frequency-averaging. The expression in (V.2) is then replaced by

$$f_1[\nu](k) = \sum_j |\langle f_0, M_{\beta_1 j} T_{\alpha_1 k} g_1 \rangle| \cdot \Upsilon_\nu(j), \tag{V.5}$$

where Υ_ν corresponds to the mel-filters, as introduced in Section V.3.1. The other steps of the scattering procedure remain the same as for GS, i.e. performing another GT to obtain layer 2 and afterwards applying an output generating atom in order to obtain the MS coefficients. The output of GS and MS can be visually explained by Figure V.1. The naming Output A displays either the output of Equation (V.2) in case of GS or Equation (V.5) in the MS case. The Output B shows the spectrogram after applying the output generating atom and Output C illustrates the output of the second layer.

V.4 Augmented Target Loss Function

In the previous sections we introduced different input data representations for subsequent classification via deep learning. In the following we want to investigate possible enhancement with alternative output/target data representations. To do so, we use an augmented target loss function, a general framework is introduced in [3]. It allows to integrate known characteristics of the target space via informed transformations on the output and target data. We now recall a general formulation of AT from [3] and describe subsequently in detail, how it can be applied on the studied audio data.

Our training data is given by the MT of the sounds as inputs together with instrument classes as targets, introduced in Section V.2.1. The inputs to the network are thus arrays $\{x_i\}_{i=1}^m \subset \mathbb{R}^{120\times160}$ and have associated target values $\{y_i\}_{i=1}^m \subset \{0,1\}^6$, corresponding to the 6 instrument classes. As described in Section V.2, in each optimization step for the parameters of the neural network, the network's output $\{\hat{y}_i\}_{i=1}^m \subset \mathbb{R}^6$ is compared with the targets $\{y_i\}_{i=1}^m$ via an underlying loss function L. However, training data often naturally contains additional important target information that is not used in the original representation. We aim to incorporate such

information tailored to the particular learning problem, enhancing the information content from the original target representation. Following the definition in [3], the augmented target loss function is given by

$$L_{AT}\big(y_i, \hat{y}_i\big) = \sum_{j=1}^{n} \lambda_j L_j\Big(\mathfrak{T}_j(y_i), \mathfrak{T}_j(\hat{y}_i)\Big). \tag{V.6}$$

Here, for all $j = 1, \ldots, n$, we let $\lambda_j > 0$ be an adjustable weight of L_j, which is some standard loss function and $\mathfrak{T}_j : \{0,1\}^6 \to \mathbb{R}^{d_j}$ is a transformation which encodes the additional information on the target space.

Here, $\mathfrak{T}_1$ corresponds to the identity on $\mathbb{R}^6$, i.e. no transformation is applied in the first component, where L_1 is the categorical cross-entropy loss [20]. For $j = 2, \ldots, n$, we choose the dimension $d_j = 1$ and L_j to be the mean squared error. The incorporation of additional information on the GoodSounds data set is described in detail in the following section.

V.4.1 Design of Transformations

We heuristically choose $d = 16$ transformations $\mathfrak{T}_2, \ldots, \mathfrak{T}_{17}$ that use target characteristics (features) arising directly from the particular target class, with $\mathfrak{T}_j : \{0,1\}^6 \to \mathbb{R}$, for $j = 2, \ldots, 17$. Amongst others the features are chosen from the enhanced scheme of taxonomy [18] and from the table of frequencies, harmonics and under tones [21]. We choose transformations that provide information that is naturally contained in the underlying instrument classes. The additional terms in the loss function (V.6) shall enable to penalize common classification errors. In this experiment, the transformations are given by the inner product of the output/target and the feature vector. E.g. we directly know to which instrument family an instrument belongs and distinguish between woodwind, brass and bowed instruments, and moreover between chordophone and aerophone instruments. Let's assume a target vector $y_i(j) = \delta_{ij}$, corresponds, respectively, to the instruments clarinet, flute, trumpet, violin, saxophone and cello, and the output of the network is $\hat{y}_i = (a_1, a_2, a_3, a_4, a_5, a_6) \in \mathbb{R}^6$. The feature vector $v_1 = (1, 1, 0, 0, 1, 0)$ then captures the information "target instrument is from family woodwind". The transformation may be defined by $\mathfrak{T}_1(y_i) = \langle y_i, v_1 \rangle$ in order to incorporate this information. Additionally, by choosing λ_j, we can weight the amount of penalization for wrong assignments in $(\mathfrak{T}_1(y_i) - \mathfrak{T}_1(\hat{y}_i))^2$. Amongst others we also use minimum and maximum frequencies of the instrument as features. E.g. the feature corresponding to minimum frequency $v_2 = (b_1, b_2, b_3, b_4, b_5, b_6) \in \mathbb{R}^6$. Again the transformation is given by $\mathfrak{T}_2(y_i) = \langle y_i, v_2 \rangle$. Choosing the right penalty for this feature could prohibit that instruments belonging to the same instrument family are classified wrong, e.g.

a cello that would be classified as a violin. One can think about AT as a method to more precisely define the measure of distance between the predicted and target classes.

V.5 Numerical Experiments

In the numerical experiments, we compare the performances of CNNs trained using the CC loss and time-frequency representations mentioned in Section V.3. As a baseline, we use the results of MT. Furthermore we compare the baseline with the results of MT with AT loss as introduced in Section V.4. The overall task is a multi-class classification of musical instruments based on the audio signals introduced in Section V.2.1.

V.5.1 Computation of Signal Representations

The raw audio signals were transformed into MT, MS and GS time-frequency representations, using the Gabor-scattering v0.0.4 library [7]. The library contains our Python implementation of all previously mentioned signal representations, with the aim to provide the community with an easy access to all of the transformations. The library's core algorithms are based on Scipy v1.2.1 [9, 15, 5] implementation of STFT and mel-filter banks from Librosa v0.6.2 library [14].

All the representations are derived from GT. In order to have a good resolution in time and frequency for our classification task, we have chosen the parameters heuristically. The final shapes of the representations are shown in Table V.1. The three dimensional output of GS contains the GT and outputs of layer 1 and 2 of the GS cf.[2], the same applies to MS. The visualizations of the time-frequency transformations of an arbitrary training sample are shown in Figure V.1.

V.5.2 Deep Convolutional Neural Network

We implemented our experiment in Python 3.6. A CNN was created and trained from scratch on Nvidia GTX 1080 Ti GPU in Keras 2.2.4 framework [4] using the described training set split into batches of size 128. We used an architecture consisting of four convolutional stacks. Each of them consists of a convolutional layer, rectified-linear unit activation function and average pooling. These stacks were followed by a fully connected layer with softmax activation function. Each network had to be adjusted slightly, because the input shapes changed according to the time-frequency representation used (GS has 3 channels, MT has less frequency channels etc.). We have tried to make the results as comparable as possible, therefore

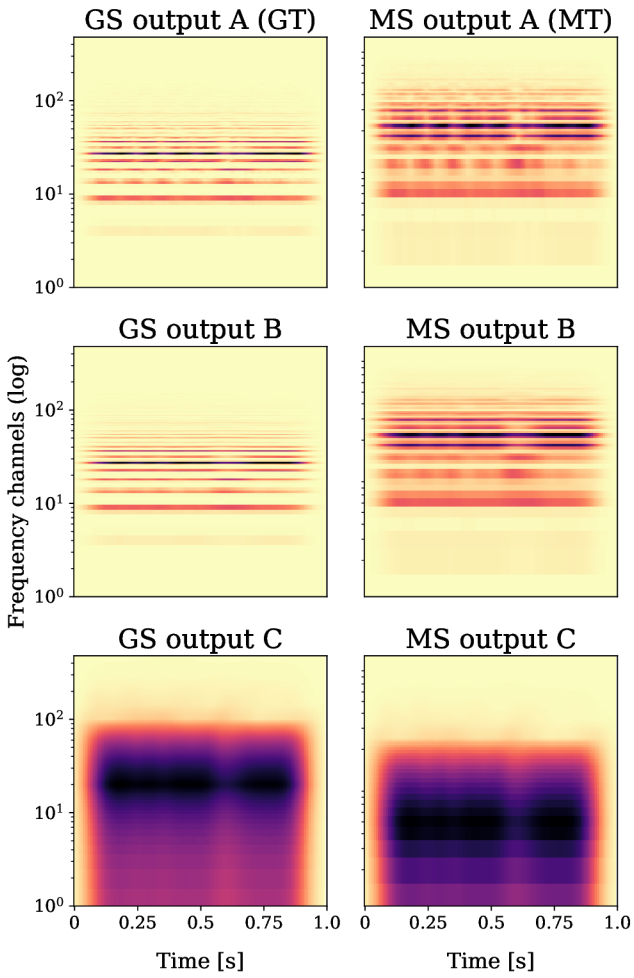


Fig. V.1: Visualization of time-frequency transformations.

the networks differ only in the number of channels of the input layer, the rest of the network is only affected by the number of frequency channels, which thanks to pooling did not cause significant difference in the number of trainable parameters. All networks have comparable number of trainable parameters within the range from 81 042 to 83 882. The weights were optimized using Adam optimizer [10]. Reproducible open source code can be found in the repository [8].

V.5.3 Training and Results

All the samples were split into training, validation and testing sets in such a way that validation and testing sets have exactly the same number of samples from each class, while this holds for training set only approximately. Segments from audio files that were used in validation or testing were not used in training to prevent leaking of information. Detailed information about the used data, stride settings for each

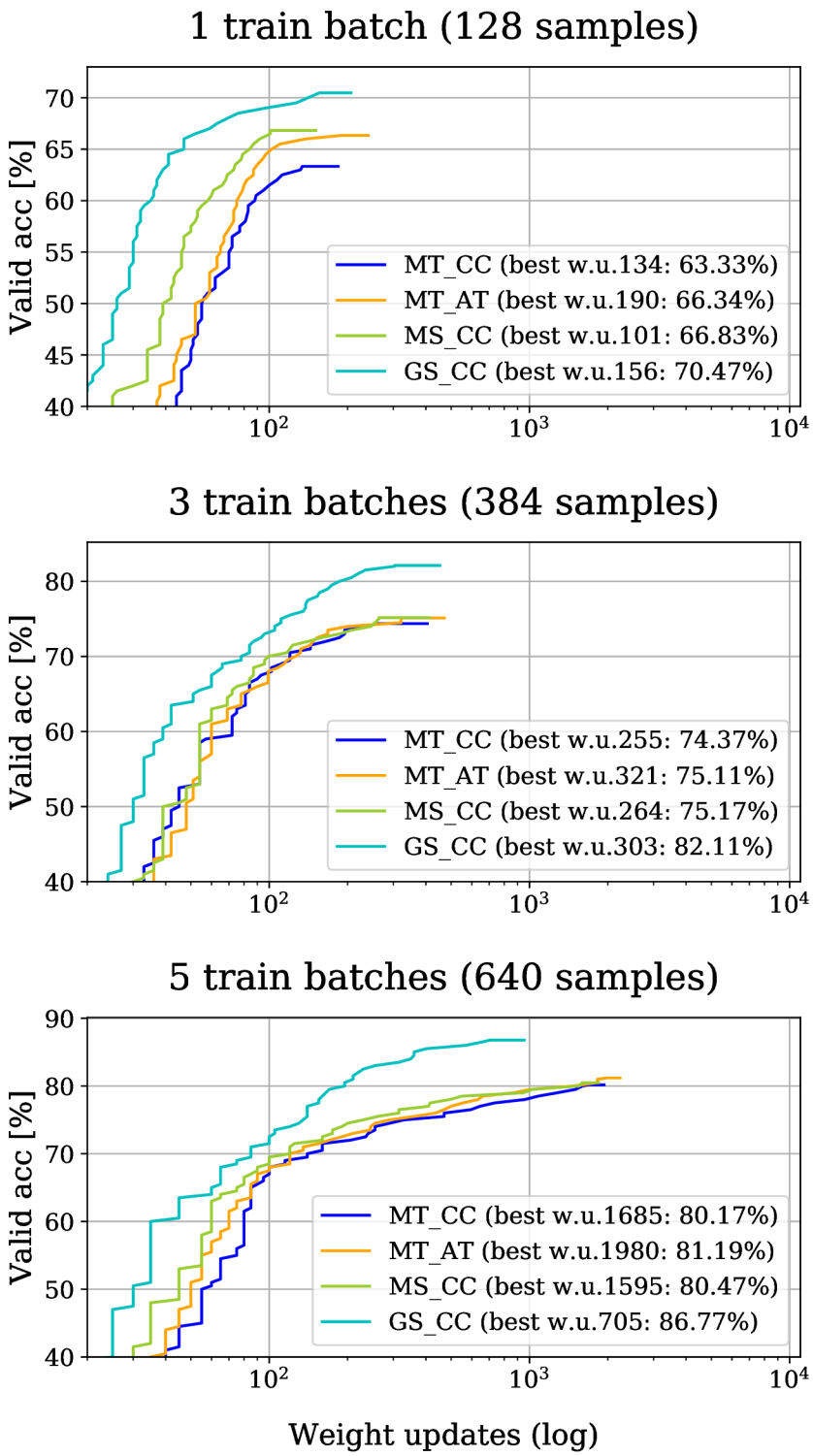


Fig. V.2: CNN performance milestone reached over number of weight updates. The computational effort in all experiments was limited to 11 000 weight updates. Figure notation: Valid acc – Accuracy performance metric measured on the validation set, Best w.u. – Weight update after which the highest performance was reached.

Table V.1: Shapes and execution time

TF	shape	CC	AT
GS	$3 \times 480 \times 160$	950 ms	-
MT	$1 \times 120 \times 160$	250 ms	320 ms
MS	$3 \times 120 \times 160$	450 ms	-

Table notation: TF – Time-frequency representation. CC/AT – The execution time of one weight update during training with CC/AT loss function.

class, obtained number of segments and their split can be found in the repository [8].

In total we have trained 36 different models (MT, MS, GS with CC and MT with AT trained on 9 training set sizes), with the following hyper-parameters: number of convolutional kernels in the first 3 convolutional layers is 64 each, learning rate is 0.001, λ of AT is 10 and λ of L_2 weight regularization is 0.001. As a baseline we have used MT with a standard CC loss function as implemented in the Keras framework and described in detail in Section V.2. The computational effort was limited to 11 000 weight updates. Time necessary for one weight update of each model is shown in Table V.1.

Table V.2 shows the highest achieved accuracies of the CNN models trained with MT for different training set sizes along with the improvements of this baseline by proposed methods. Accuracy is computed as a fraction of correct predictions to all predictions. In Figure V.2 we compare the number of weight updates necessary to surpass a certain accuracy threshold for all proposed methods. Occlusion maps [19] for a random MS sample are visualized per 3 frequency bins in Figure V.3.

V.6 Discussion and Conclusions

Our previous work on Gabor scattering showed that signal variability w.r.t. different time scales is separated by this transform, cf. [2], which is a beneficial property for learning. The common choice of a time-frequency representation of audio signals in predictive modeling is mel-spectrogram; hence, as a natural step, we introduced MS in this paper, a new feature extractor which combines the properties of GS with mel-filter averaging. We also investigated the impact of additional information about the target space through AT on the performance of the trained CNN.

From the results on GoodSounds dataset shown in Table V.2, we see that all proposed methods outperform the baseline (mel-spectrogram with categorical cross-entropy loss) on the first three most limited training sets, i.e. the data sets with

Table V.2: Improvements of the MT Baseline

Highest validation set accuracies				
NB	MT	MT_{AT}	MS	GS
1	**63.33 %**	**+3.01 %**	**+3.50 %**	**+7.15 %**
3	**74.37 %**	**+0.74 %**	**+0.80 %**	**+7.74 %**
5	**80.17 %**	**+1.02 %**	**+0.31 %**	**+6.60 %**
7	82.93 %	-1.12 %	-0.09 %	+5.63 %
9	85.40 %	+0.95 %	-0.43 %	+5.28 %
11	86.53 %	+0.33 %	+1.26 %	+5.57 %
55	96.06 %	-0.27 %	-0.27 %	+2.52 %
110	96.31 %	-0.04 %	+0.06 %	+2.53 %
550	96.00 %	+0.74 %	+0.48 %	+3.12 %

Corresponding testing set accuracies				
NB	MT	MT_{AT}	MS	GS
1	**64.28 %**	**+2.73 %**	**+3.36 %**	**+6.93 %**
3	**75.61 %**	**+0.58 %**	**+0.32 %**	**+7.26 %**
5	**80.69 %**	**+0.79 %**	**+0.07 %**	**+6.93 %**
7	83.48 %	-1.13 %	+0.37 %	+6.30 %
9	86.30 %	+0.54 %	-0.43 %	+5.23 %
11	87.41 %	-0.43 %	+1.30 %	+4.85 %
55	96.27 %	-0.20 %	-0.31 %	+2.26 %
110	96.80 %	-0.55 %	-0.12 %	+2.21 %
550	96.72 %	+0.27 %	+0.07 %	+2.29 %

Table notation: NB – Number of training batches with 128 samples each. MT, MS and GS – mel-spectrogram, Mel scattering and Gabor scattering as input representations with CC. MT here servers as a baseline for comparison with other methods. MT_{AT} – mel-spectrogram as input representation with AT. Testing set accuracies were evaluated after the epoch where the validation accuracy was the highest.

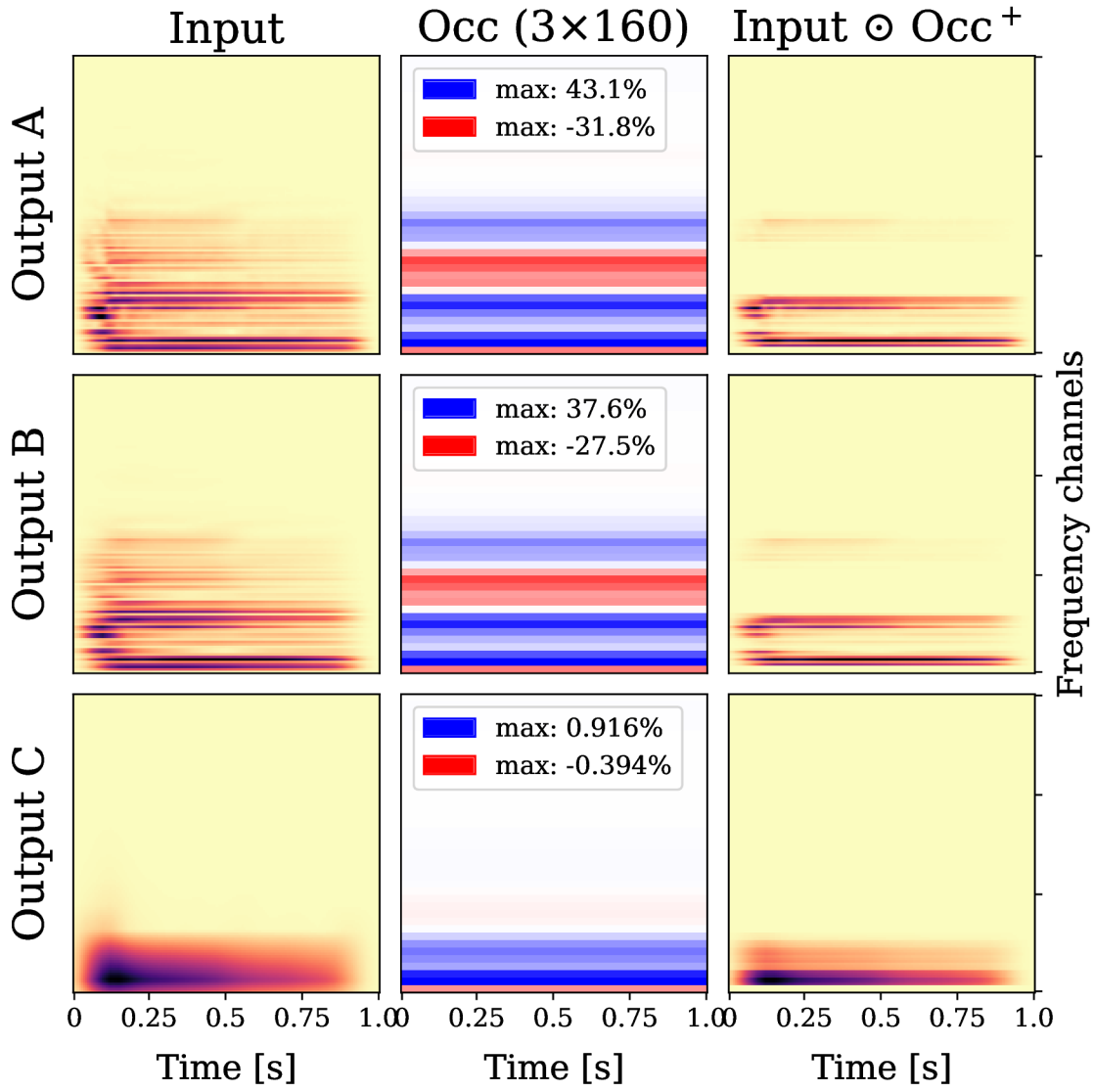


Fig. V.3: Visualization of occlusion maps and frequency channel importance based on the best performing model trained on 1 batch of MS. Signal shown is randomly selected alto sax sample. Figure notation: Input – input representation for CNN. Occ – occlusion map created by sliding occlusion window. Input $\odot$ Occ$^+$ – Elementwise multiplication of input with positive semidefinite occ (negative elements were changed to zeros before multiplication). Blue and red colors – Positive and negative influence of particular frequency channel bin on the model performance.

the least amount of data. All proposed methods also show a trend to achieve better results earlier in the training, as visible in Figure V.2. This trend seems to diminish with bigger training set sizes. Improvements on the last, biggest training set can be justified by the fact that this experiment was interrupted before it had the time to converge, therefore highlighting earlier successes of the proposed methods. From the newly proposed methods, AT is the least expensive in terms of training time, but on the other hand yields the smallest improvement in this experimental setup. Nevertheless, it has another advantage: it steers the training towards learning the penalized characteristics, e,g. to learn the characteristic of an instrument being or not being a wood instrument if the information about this grouping is provided through AT. We believe that the positive effect of AT in this setup becomes obsolete with higher number of training batches because after training above a certain accuracy threshold, the network already predicts the correct groups of classes and therefore can not gain from AT anymore.

MS performed better than both MT and $\mathrm{MT}_{\mathrm{AT}}$ for slightly higher cost of computation and also achieved the same performances earlier. GS outperformed all of the tested methods and showed an improvement over all training set sizes, however this might also suggest that GT (without mel-filtering) would be a better input data representation for this task in the first place. As in GS, MS comprises exclusively the information of its MT origin. The separation of the embedded information into three distinct channels might be the reason for its success. The evidence is visible in Figure V.2, which shows MS reaching higher accuracies after less weight updates than MT, suggesting that the network did not have to learn similar separation during training. Also, the visualization in Figure V.3 supports this by showing a positive influence of Outputs A and B on the model's performance.

It remains to be said, that improvements which can be gained by using AT, MS or GS highly depend on the task being solved, on the choice of transformations based on the amount of additional available information for AT and on the correctly chosen parameters of the time-frequency representations.

From what was stated above, we can conclude that AT provides a more precise measure of distance between outputs and targets. That's why it can help in scenarios where the training set is not large enough to allow the learning of all characteristics, but can be penalized by AT. We suggest to use/experiment with the proposed methods for other data sets if there is not a sufficient amount of data available or/and there exist reasonable transformations in the target space relevant to the task being solved. All proposed methods might be found useful also in scenarios with limited resources for training.

In order to obtain reliable statistical results on the various methods, it would be necessary to run all experiments several hundred times with different seeds. For

the current contribution, such a procedure was not included due to the restriction of computational resources and is thus left for future work.

Bibliography

[1] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

[2] R. Bammer, M. Dörfler, and P. Harar. Gabor frames and deep scattering networks in audio processing. *arXiv preprint arXiv:1706.08818*, 2017.

[3] A. Breger, J. I. Orlando, P. Harar, M. Dörfler, S. Klimscha, C. Grechenig, B. S. Gerendas, U. Schmidt-Erfurth, and M. Ehler. On orthogonal projections for dimension reduction and applications in augmented target loss functions for learning problems. *arXiv preprint arXiv:1901.07598*, 2019.

[4] F. Chollet et al. Keras. `https://keras.io`, 2015.

[5] D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[6] K. Gröchenig. *Foundations of time-frequency analysis*. Applied and numerical harmonic analysis. Birkhäuser, Boston, Basel, Berlin, 2001.

[7] P. Harar. gabor-scattering. `https://gitlab.com/paloha/gabor-scattering`, 2019.

[8] P. Harar. gs-ms-mt. `https://gitlab.com/hararticles/gs-ms-mt`, 2019.

[9] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2019-02-01]. URL: `http://www.scipy.org/`.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[12] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[13] S. Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.

[14] B. McFee, M. McVicar, S. Balke, C. Thomé, V. Lostanlen, C. Raffel, ..., and A. Holovaty. librosa/librosa: 0.6.2, Aug. 2018. URL: `https://doi.org/10.5281/zenodo.1342708`, `doi:10.5281/zenodo.1342708`.

[15] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[16] J. Pons Puig, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra. End-to-end learning for music audio tagging at scale. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018; 2018 Sep 23-27; Paris, France. p. 637-44*. International Society for Music Information Retrieval (ISMIR), 2018.

[17] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, H. Tokuda, W. Hariya, K. Oishi, and X. Serra. A real-time system for measuring sound goodness in instrumental sounds. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[18] E. M. Von Hornbostel and C. Sachs. Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann. *The Galpin Society Journal*, pages 3–29, 1961.

[19] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[20] Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788, 2018.

[21] ZyTrax. Frequency ranges. zytrax.com/tech/audio/audio.html, 2018.

Appendix

Curriculum Vitæ

Pavol Harár

PhD student @ Brno University of Technology
Researcher @ NuHAG, University of Vienna
Co-Founder & Researcher @ ACAI.AI

Main research interests

Analysis of pathological voices with deep learning
Exploration of novel time-frequency representations
Investigation of new preprocessing steps for learning algorithms

---

Personal

Residence	Vienna, Austria
Languages	Native in Slovak & Czech, English C1, French & German A1
Contact	pavol.harar@vut.cz, pavol.harar.eu

Education

2015 - *	PhD in Machine Learning, FEEC, BUT (expected in 09/2019)
2012 - 2015	Master of System Management and Informatics, FBM, BUT

Work experience

2019 - *	Researcher at NuHAG, University of Vienna
2019 - *	Co-Founder & Researcher at ACAI.AI
2015 - *	Researcher at Brain Disease Analysis Laboratory, BUT

Skills

Python, Numpy, Pandas, Keras, TensorFlow, Scikit-learn, Scipy, Matplotlib, Flask, Linux, Docker, Git, LaTeX, Java, Parallel & GPU computing

Teaching

2016 - 2017	Assistant lecturer: Basics of OOP in Java
2017	Coach: PyLadies Brno (public craschcourse of Python programming language initiated by PyLadies mentorship group)
2017	Bachelor thesis advisor: Vojtěch Hájek, Creating a database of audio recordings with artificial noise in an anechoic chamber
2016	Diploma thesis advisor: Martin Majtán, Trainable image segmentation usind deep neural networks

Participation in projects

2018 - 2019	Czech Ministry of Education, Youth And Sports (CZ.02.2.69/0.0/0.0/16_027/0008371): International mobility of researchers
2017 - 2019	Brno University of Technology (FEKT-S-17-4476) : Multimodal processing of unstructured data using machine learning and sophisticated methods of signal and image analysis
2016 - 2019	Ministry of Health of Czech Republic (NV16-30805A): Effects of non-invasive brain stimulation on hypokinetic dysarthria, micrographia, and brain plasticity in patients with Parkinson's disease
2015 - 2019	Vienna Science and Technology Fund (WWTF) (MA14-018): Semantic Annotation by Learned Structured and Adaptive Signal Representations (SALSA)
2015 - 2019	Czech Ministry of Education, Youth And Sports of Czech Republic (LO1401): Interdisciplinary research of wireless technologies (INWITE)

Internships

2018 - 2019	Numerical Harmonic Analysis Group (NuHAG), University of Vienna, Austria
2017	Technological Centre for Innovation in Communications (IDeTIC), University of Las Palmas de Gran Canaria, Spain

Awards

2017	The best lecturer at the Department of Telecommunications and 8^{th} best lecturer at the Faculty of Electrical Engineering and Telecommunications at Brno University of Technology

Invited lectures

06/06/2019	Hands on introduction to Attention mechanism at Deep Learning Seminar, University of Vienna, Austria
21/11/2018	Automatic Transformation of Empirical Data Distribution as an Experimental Pre-Processing Step for Neural Networks at Acoustic Research Institute (ARI), Vienna, Austria
25/06/2018	Basics of Neural Networks (Regression & Classification) at NuHAG, Vienna, Austria
05/06/2018	Towards Robust Voice Pathology Detection in a poster session of Harmonic Analysis and Applications Conference in Strobl, Austria
02/02/2018	Battle-proven machine learning workflow from venv to dockerization at BUT, Brno, Czech Republic
12/12/2017	How to train fox's brain to predict the future at Mergado DevTalks, Brno, Czech Republic
13/09/2019	Voice Pathology Detection Using Deep Learning: a Preliminary Study at Systematic approaches to deep learning methods for audio workshop, ESI, Vienna, Austria

Publications in journals

2019	Anna Breger, Jose Ignacio Orlando, Pavol Harar, Monika Dörfler, Sophie Klimscha, Christoph Grechenig, Bianca S. Gerendas, Ursula Schmidt-Erfurth, and Martin Ehler. On orthogonal projections for dimension reduction and applications in augmented target loss functions for learning problems. *Journal of Mathematical Imaging and Vision*, in press. arXiv:1901.07598.
2018	P. Harar, Z. Galaz, J. B. Alonso-Hernandez, J. Mekyska, R. Burget, and Z. Smekal. Towards robust voice pathology detection. *Neural Computing and Applications*, pages 1–11, 2018. arXiv:1907.06129, doi:10.1007/s00521-018-3464-7.
2018	Vojtěch Hájek, Pavol Hárar, Jiří Schimmel, and Radim Burget. But-czas: Korpus kvalitních nahrávek české řeči pořízených v bezodrazové komoře. *Elektrorevue*, 20(2):48–52, 2018.
2017	Roswitha Bammer, Monika Dörfler, and Pavol Harar. Gabor frames and deep scattering networks in audio processing. *arXiv preprint*, 2017. arXiv:1706.08818, submitted.

Publications in conference proceedings

2019 Pavol Harar, Roswitha Bammer, Anna Breger, Monika Dörfler, and Zdenek Smekal. Improving machine hearing on limited data sets. In *2019 The 11th Int. Cong. on Ultra Modern Telecom. and Control Systems (ICUMT)*. IEEE, in press. `arXiv:1903.08950`.

2018 Zoltan Galaz, Jiri Mekyska, Tomas Kiska, Vojtech Zvoncak, Jan Mucha, Pavol Harar, Zdenek Smekal, Ilona Eliasova, et al. Monitoring progress of parkinson's disease based on changes in phonation: a pilot study. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–5. IEEE, 2018. doi:10.1109/TSP.2018.8441307.

2017 P. Harar, J. B. Alonso-Hernandez, J. Mekyska, Z. Galaz, R. Burget, and Z. Smekal. Voice Pathology Detection Using Deep Learning: a Preliminary Study. In *2017 international conference and workshop on bioinspired intelligence (IWOBI)*, pages 1–4. IEEE, 2017. arXiv:1907.05905, doi:10.1109/IWOBI.2017.7985525.

2017 Pavol Harar, Radim Burget, and Malay Kishore Dutta. Speech emotion recognition with deep learning. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 137–140. IEEE, 2017. doi:10.1109/SPIN.2017.8049931.

2016 Garima Vyas, Malay Kishore Dutta, Jiri Prinosil, and Pavol Harar. An automatic diagnosis and assessment of dysarthric speech using speech disorder specific prosodic features. In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, pages 515–518. IEEE, 2016. doi:10.1109/TSP.2016.7760933.

Software

2019 Pavol Harar. Gabor scattering. https://gitlab.com/paloha/gabor-scattering, 2019.

2018 Pavol Harar and Dennis Elbraechter. Redistributor. https://gitlab.com/paloha/redistributor, 2018.

Scientific activity

H-index	3 (according to Scopus)
Citation count	20 (according to Scopus, excluding self-citations)