Separation and Count Estimation for Audio Sources Overlapping in Time and Frequency

Trennung und Schätzung der Anzahl von Audiosignalquellen mit Zeit- und Frequenzüberlappung

Dissertation

Der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg
zur
Erlangung des Doktorgrades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von
Fabian-Robert Stöter

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	19.09.2019
Vorsitzender des Promotionsorgans:	Prof. Dr.-Ing. Reinhard Lerch
Gutachter:	Prof. Dr.-Ing. Bernd Edler
	Prof. Gaël Richard (Ph.D)

For Claudia and Emil

ABSTRACT

Everyday audio recordings involve mixture signals: music contains a mixture of instruments; in a meeting or conference, there is a mixture of human voices. For these mixtures, automatically separating or estimating the number of sources is a challenging task. A common assumption when processing mixtures in the time-frequency domain is that sources are not fully overlapped. However, in this work we consider some cases where the overlap is severe — for instance, when instruments play the same note (unison) or when many people speak concurrently ("cocktail party") — highlighting the need for new representations and more powerful models.

To address the problems of source separation and count estimation, we use conventional signal processing techniques as well as deep neural networks (DNN). We first address the source separation problem for unison instrument mixtures, studying the distinct spectro-temporal modulations caused by vibrato. To exploit these modulations, we developed a method based on time warping, informed by an estimate of the fundamental frequency. For cases where such estimates are not available, we present an unsupervised model, inspired by the way humans group time-varying sources (common fate). This contribution comes with a novel representation that improves separation for overlapped and modulated sources on unison mixtures but also improves vocal and accompaniment separation when used as an input for a DNN model.

Then, we focus on estimating the number of sources in a mixture, which is important for real-world scenarios. Our work on count estimation was motivated by a study on how humans can address this task, which lead us to conduct listening experiments, confirming that humans are only able to estimate the number of up to four sources correctly. To answer the question of whether machines can perform similarly, we present a DNN architecture, trained to estimate the number of concurrent speakers. Our results show improvements compared to other methods, and the model even outperformed humans on the same task.

In both the source separation and source count estimation tasks, the key contribution of this thesis is the concept of “modulation”, which is important to computationally mimic human performance. Our proposed Common Fate Transform is an adequate representation to disentangle overlapping signals for separation, and an inspection of our DNN count estimation model revealed that it proceeds to find modulation-like intermediate features.

ZUSAMMENFASSUNG

Im Alltag sind wir von gemischten Signalen umgeben: Musik besteht aus einer Mischung von Instrumenten; in einem Meeting oder auf einer Konferenz sind wir einer Mischung menschlicher Stimmen ausgesetzt. Für diese Mischungen ist die automatische Quellentrennung oder die Bestimmung der Anzahl an Quellen eine anspruchsvolle Aufgabe. Eine häufige Annahme bei der Verarbeitung von gemischten Signalen im Zeit-Frequenzbereich ist, dass die Quellen sich nicht vollständig überlappen. In dieser Arbeit betrachten wir jedoch einige Fälle, in denen die Überlappung immens ist — zum Beispiel, wenn Instrumente den gleichen Ton spielen (unisono) oder wenn viele Menschen gleichzeitig sprechen (Cocktailparty) —, so dass neue Signal-Repräsentationen und leistungsfähigere Modelle notwendig sind.

Um die zwei genannten Probleme zu bewältigen, verwenden wir sowohl konventionelle Signalverbeitungsmethoden als auch tiefgehende neuronale Netze (DNN). Wir gehen zunächst auf das Problem der Quellentrennung für Unisono-Instrumentenmischungen ein und untersuchen die speziellen, durch Vibrato ausgelösten, zeitlich-spektralen Modulationen. Um diese Modulationen auszunutzen entwickelten wir eine Methode, die auf Zeitverzerrung basiert und eine Schätzung der Grundfrequenz als zusätzliche Information nutzt. Für Fälle, in denen diese Schätzungen nicht verfügbar sind, stellen wir ein unüberwachtes Modell vor, das inspiriert ist von der Art und Weise, wie Menschen zeitveränderliche Quellen gruppieren (Common Fate). Dieser Beitrag enthält eine neuartige Repräsentation, die die Separierbarkeit für überlappte und modulierte Quellen in Unisono-Mischungen erhöht, aber auch die Trennung in Gesang und Begleitung verbessert, wenn sie in einem DNN-Modell verwendet wird.

Im Weiteren beschäftigen wir uns mit der Schätzung der Anzahl von Quellen in einer Mischung, was für reale Szenarien wichtig ist. Unsere Arbeit an der Schätzung der Anzahl war motiviert durch eine Studie, die zeigt, wie wir Menschen diese Aufgabe angehen. Dies hat uns dazu veranlasst, eigene Hörexperimente durchzuführen, die bestätigten, dass Menschen nur in der Lage sind, die Anzahl von bis zu vier Quellen korrekt abzuschätzen. Um nun die Frage zu beantworten, ob Maschinen dies ähnlich gut können, stellen wir eine DNN-Architektur vor, die erlernt hat, die Anzahl der gleichzeitig sprechenden Sprecher zu ermitteln. Die Ergebnisse zeigen Verbesserungen im Vergleich zu anderen Methoden, aber vor allem auch im Vergleich zu menschlichen Hörern.

Sowohl bei der Quellentrennung als auch bei der Schätzung der Anzahl an Quellen ist ein Kernbeitrag dieser Arbeit das Konzept der “Modulation”, welches wichtig ist, um die Strategien von Menschen mittels Computern nachzuahmen. Unsere vorgeschlagene Common Fate Transformation ist eine adäquate Darstellung, um die Überlappung von Signalen für die Trennung zugänglich zu machen und eine Inspektion unseres DNN-Zählmodells ergab schließlich, dass sich auch hier modulationsähnliche Merkmale finden lassen.

PUBLICATIONS

Parts of chapters 4-8 of this thesis are based on contributions that I published as first author during my time as a doctoral student.

MAIN PUBLICATIONS

This thesis is based on several previously published publications. As such, they are cited repeatedly throughout the remainder of this thesis. If a section is mainly based on one of these publications, a remark is added at the side of the page, instead of citing the same publication exhaustively. The following is a list of my publications and my own contributions therein, ordered by the first appearance in the chapters of this thesis.

Chapter 4

[275] F.-R. Stöter, M. Müller, and B. Edler, "Multi-sensor cello recordings for instantaneous frequency estimation," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, Brisbane, Australia, 2015, pp. 995–998.

Besides the leading authoring of the text, my contribution in this work was the initial idea which was inspired by the gap in existing F_0 estimation datasets not providing sufficient level of annotation to derive an accurate ground truth. The work was done together with our student, Michael Müller, who much helped to design and manufacture the custom experiment hardware, organize the actual recording and assist in analyzing and converting the recorded data. Bernd Edler revised the article.

Chapter 5

[270] F.-R. Stöter, S. Bayer, and B. Edler, "Unison source separation," in *17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

This work is based on my initial idea. Besides being the principal author of the text, I created the experimental designs the software implementation and evaluation. My college Stefan Bayer contributed important insights about the theory and implementation of time warp-

ing framework and formulated the mathematical notation therein. Bernd Edler revised the article.

[277] F.-R. Stöter, N. Werner, S. Bayer, and B. Edler, "Refining fundamental frequency estimates using time warping," in *Proceedings of EUSIPCO 2015*, Nice, France, Sep. 2015.

My contribution to this work was the initial idea, the literature overview of F_0 estimation algorithm and the evaluation of the algorithms. Furthermore, I authored the main part of the text. The work was done in close collaboration with my colleague Nils Werner who contributed to the efficient implementation of the F_0 warping algorithm and the generation of appropriate warp contours to match mathematical constraints of time-warping. Bernd Edler revised this publication.

Chapter 6

[274] F.-R. Stöter, A. Liutkus, R. Badeau, B. Edler, and P. Magron, "Common fate model for unison source separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016.

My contribution to this work was the experimental design, implementation and evaluation and the writings of the main parts of the publication. The original idea was developed by Antoine Liutkus, who also helped to formulate the theory. Paul Magron provided code and results to compare with the HR-NMF method. Roland Badeau and Bernd Edler revised the article.

Chapter 7

[276] F.-R. Stöter, M. Schoeffler, B. Edler, and J. Herre, "Human ability of counting the number of instruments in polyphonic music," in *Proceedings of Meetings on Acoustics*, vol. 19, 2013.

The work is based on a collaboration with Michael Schöffler and Jürgen Herre. My contribution to this work was the initial idea, as well as the experimental prototype design, and evaluation and the leading authoring of the text. My college Michael Schöffler contributed to the development of the web-based evaluation software that later led to a follow-up work [254] which I co-authored. Jürgen Herre and Bernd Edler revised the article.

[272] F.-R. Stöter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "CountNet: Estimating the number of concurrent speakers using supervised learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 268–282, Feb. 2019.

[273] F.-R. Stöter, S. Chakrabarty, B. Edler, and E. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

The publications were a result of collaboration with Soumitro Chakrabarty and Emanuël Habets. My contribution to this work was the initial problem formulation and the core idea to address the problem using deep neural networks. Besides the leading authoring of the text, I designed the dataset and created experiments and evaluation. My college Soumitro Chakrabarty contributed to the development of the deep learning method; Emanuël A. P. Habets and Bernd Edler revised the articles.

ADDITIONAL PUBLICATIONS

The following publications that I co-authored, were not directly referred to in this thesis but are nonetheless very closely related to audio based methods presented in this thesis.

1. F. Nagel, F.-R. Stöter, N. Degara, S. Balke, and D. Worrall, "Fast and accurate guidance - response times to navigational sounds," in *2014 International Conference on Auditory Display*, 2013.
2. M. Schoeffler, F.-R. Stöter, H. Bayerlein, B. Edler, and J. Herre, "An experiment about estimating the number of instruments in polyphonic music: A comparison between internet and laboratory results," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 389–394.
3. M. Schoeffler, F.-R. Stöter, B. Edler, and J. Herre, "Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R recommendation BS.1534 (MUSHRA)," in *1st web audio conference, Paris, France*, 2015.
4. A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Feb. 2017, pp. 323–332.

5. N. Werner, S. Balke, F.-R. Stöter, M. Müller, and B. Edler, "Trackswitch.js: A versatile web-based audio player for presenting scientifc results," in *3rd web audio conference, London, UK*, 2017.

6. W. Mack, S. Chakrabarty, F.-R. Stoeter, S. Braun, B. Edler, and E. Habets, "Single-channel dereverberation using direct mmse optimization and bidirectional lstm networks," Sep. 2018.

7. Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, Aug. 2018.

8. M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webMUSHRA — a comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, 2018.

9. F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2018, pp. 293–305.

10. E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F. Stöter, "Musical source separation: An introduction," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, Jan. 2019.

OPEN DATASETS AND SOFTWARE

To foster reproducible research, the following datasets and code were contributed under open licenses:

1. Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *Musdb18 - a corpus for music separation*, Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372.
2. M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, *webMUSHRA*, Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1069840.
3. F.-R. Stöter, *WICE-DB - web instrument count experiment*, Jun. 2013. [Online]. Available: https://doi.org/10.5281/zenodo.1469076.
4. F.-R. Stöter, *Unison source separation dataset*, Sep. 2014. [Online]. Available: https://doi.org/10.5281/zenodo.1467921.
5. F.-R. Stöter, *CountIt - an auditory experiment to estimate the number of speakers*, Dec. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1467968.
6. F.-R. Stöter, S. Chakrabarty, E. Habets, and B. Edler, *LibriCount, a dataset for speaker count estimation*, Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1216072.
7. F.-R. Stöter, M. Müller, and B. Edler, *MUSERC: Multi-sensor cello recordings for instantaneous frequency estimation*, Oct. 2015. [Online]. Available: https://doi.org/10.5281/zenodo.1560651.
8. F.-R. Stöter and N. Werner, *SiSEC 2016 website*, Nov. 2016. [Online]. Available: https://doi.org/10.5281/zenodo.1490095.

ACKNOWLEDGMENTS

First of all, I want to thank my supervisor Prof. Dr.-Ing. Bernd Edler for all his help and support throughout all the years. Bernd is the nicest and kindest supervisor anyone could have: his door was always open for me to have fruitful discussions and solve tricky problems together and at the same time he was giving me enough freedom to allow me to develop my own research ideas.

Besides my advisor, I want to thank Prof. Gaël Richard for taking the time to review my thesis. I also want to thank Dr. Antoine Liutkus for inviting me to Nancy for a summer research visit that resulted in so many good ideas and a great friendship!

I want to thank all the amazing people in the AudioLabs, which made my work in Erlangen so enjoyable: First, I want to thank Elke, Tracy and Day-See for all the administrative help and beyond. Then, I want to thank Stefan Turowski for his great technical management in the AudioLabs. Next, I want to thank all of the colleagues at the AudioLabs (in alphabetical order): Alexander Adami, Stefan Bayer, Stefan Balke, Sebastian Braun, Tom Bäckström, Soumitro Chakrabarty, Youssef El Baba, Alexandra Craciun, Christian Dittmar, Sascha Disch, Jonathan Driedger, Esther Feichtner, Johannes Fischer, Yesenia Lacouture Parodi, Emanuël Habets, Jürgen Herre, Nanzhu Jiang, Patricio Lopez-Serrano, Wolfgang Mack, Goran Markovic, Vlora Arifi Müller, Meinard Müller, Thomas Prätzlich, Sebastian Rosenzweig, Konstantin Schmidt, Michael Schöffler, Armin Taghipour, Stefan Turowski, Maja Taseska, Maria Luis Valero, Elke Weiland, Christof Weiß, Nils Werner, Frank Zalkow and Julia Zalkow. Thanks to the people at Fraunhofer IIS, especially to Sascha Disch, Christopher Oates, Frederik Nagel, Christian Uhle. And I also want to thank Thomas Zeiser from the RRZE high performance cluster for his great support. In this vein, I want thank to all the great scientific open source tools out there that powered most of the experiments in this thesis. A big thank to the students and interns I supervised and I thank them for all the great work: Berkan Ercan, Erik Johnson, Aravindh Krishnamoorty, Jeremy Hunt, Bufei Liu and Qiao Wang. To Karlheinz Busch from the Bamberg Symphonic Orchestra, Johannes Huber and Michael Müller for making our datasets possible.

To Annika, Lisa, Florian, Chris and Chris for the great time in Nuremberg and to Mathieu, Cheryl and Elias for the warm welcome in Montpellier. I also want to thank the Faller family for their great support during the hard times of writing this thesis.

I deeply thank my family — Dagmar, Heinrich and Marion — for supporting me in every point of time in my life.

And last but not least, I want to thank my beloved partner and friend Claudia who gave me so much joy and hope that this journey succeeds.

...and to my son Emil for his beautiful smile.

CONTENTS

ACRONYMS

STFT	short-time Fourier transform
AM	amplitude modulation
FM	frequency modulation
NMF	non-negative matrix factorization
NTF	non-negative tensor factorization
SiSEC	Signal Separation Evaluation Campaign
SDR	Source to Distortion Ratio
SIR	Source to Interferences Ratio
SAR	Sources to Artifacts Ratio
DNN	deep neural network
FNN	fully-connected neural network
CNN	convolutional neural network
RNN	recurrent neural network
LSTM	long short-term memory network

1 INTRODUCTION

It is very likely that you know the following situation: you were at a crowded party and the next day your best friend, who was unable to join, asked you "How many people were there?". You then struggled to find an answer because you had such intense conversations that you were unable to put your focus on the other guests. This scene includes many interesting aspects that are relevant to this thesis. Notably, it reminds us that in our daily life we are exposed to situations where multiple events overlap in time.

In contrast to our vision, although we might be physically able to *hear* all sounds, we deliberately choose to *listen to* a few sources and attenuate others. In a noisy environment, we can steer our attention to one sound source, even without eye contact and using only a single ear [32]. However, this attention mechanism prevents us from observing the acoustic scene as a whole. This ability to concentrate on a single source is not limited to conversations — it also applies to music. If we imagine attending a music concert, it is likely that we focus on the lead vocalist and miss out many details of the background band, demonstrating our ability to *separate* audio mixtures, at least cognitively.

In the audio research community, the task of attenuating undesired speakers when multiple concurrent speakers are present is known as the "cocktail party problem" [110]. For the past 70 years [47], researchers have been fascinated by this idea to computationally imitate this ability of humans to separate the sources in a mixture. In the general setting, which is not restricted to the cocktail party scenario but also includes music processing, this problem is called *source separation* and is one of the key topics considered in this thesis.

Although most scientific efforts have focused on separation, our example highlights the fact that even the number of sources is a valuable information. From a more technical point of view, many separation methods rely on prior knowledge of the number of sources, requiring this information to be estimated beforehand, or provided by a user [168]. As we illustrated, humans are not very good at estimating the number of sources based on audio which sharply contrasts with our ability to focus on a single source in a crowded audio scene.

Both scenarios of music and speech mixtures have in common that estimating the number of sources and separating them becomes more challenging when the signals are more *overlapped*. And both tasks become even more challenging when sources are almost entirely overlapped such as when multiple instruments are playing the same

note (in unison). In this unique scenario where sources are overlapped in time and frequency, observable differences between sources are difficult to obtain at a short time scale. However, differences do appear when we consider longer time contexts, for which the variations of sources over time become important. For instance, speech and the sound of musical instruments can have distinct modulations such as vibrato, created by conscious physical manipulation of the sound, to make a sound more intelligible or pleasant.

In this thesis, we aim to investigate if modulations can be utilized for analyzing and processing of highly overlapped speech and music source signals. For this, we first study and develop new representations that could improve the analysis and processing of such signals, to address if modulations are automatically detected or extracted from highly overlapped signals. Second, we develop new methods built upon such representations to address source separation. These methods are designed for constrained scenarios where modulation effects can easily be exploited. Third, to investigate and develop new methods to address the task of estimating the number of sources in highly overlapped mixtures. Finally, we want to show how such research may be transferred from synthetic signals to real-world scenarios.

1.1 SUMMARY OF CONTRIBUTIONS

This thesis contains five main contributions:

1. I reviewed scenarios of time and frequency overlapped audio sources. I studied a scenario where instruments are highly overlapped (unison) but I also considered known scenarios for speech and music. In this unison scenario, I reviewed how slowly-varying tempo-spectral modulations, caused e.g. by vibrato, can be utilized for separation and source count estimation of highly overlapped signals. Furthermore, I showed how these scenarios can stimulate new research directions to *analyze* and *process* such signals.

2. I designed two novel methods to *separate* unison instrument mixtures: one is informed by an estimate of the fundamental frequency variation. The other is unsupervised, inspired by the way how humans segregate time-varying sources. Along the way, I also proposed a post-processing to improve F_0 estimates based on the same principles. Next, I studied how the observations from the unison scenario can be transferred to real-world scenarios such as lead accompaniment separation by applying the deep learning framework.

3. I conducted two detailed experimental studies to assess how humans perceive highly overlapped mixtures and how they

perform when asked to estimate the number of sources. In these studies, I focussed on scenarios of overlapped speech as well as polyphonic music recordings. Both studies confirmed previous work, indicating that humans can only correctly estimate the number of concurrent sources up to three.

4. We designed a method to automatically estimate the maximum number of concurrent speakers. This method uses deep neural networks to addresses "cocktail party" like environments. This model reached state-of-the-art performance when compared to other models and also supersedes human performance when compared with the results of my subjective listening experiments. Finally, I revealed the relation between slow modulations in speech and the ability of a model of *learning to count*.

5. As a practical contribution, I developed tools to assess the quality of the separation system using interactive web applications. I helped to create publicly available datasets for separation and F_0 estimation. Furthermore, I co-organized the Signal Separation Evaluation Campaign (SiSEC) to improve sustainability and reproducibility for the research community.

1.2 STRUCTURE OF THIS THESIS

The thesis and its relevant linked publications are organized into six main chapters.

CHAPTER 2 explains the fundamental concepts of audio signals (Section 2.1), sources and overlapped sounds (Section 2.2), relevant for the remainder of this thesis. Furthermore, the process of mixing sound sources as well as its inverse task — sound source separation — are explained. The chapter also covers basics of fundamental frequency and its variations (Section 2.1.4) as an important feature for harmonic audio signals.

CHAPTER 3 introduces relevant tasks and applications in the context of highly overlapped sounds. Furthermore, the importance of slow modulations in this context is discussed (Section 3.2).

CHAPTER 4 presents and discusses the importance of data for analysis and evaluation. In this chapter we present a dataset for unison instrument mixtures (Section 4.1 [270, 279]) and a dataset for precise fundamental frequency estimates (Section 4.2 [275, 284]). Furthermore we also present a short overview of relevant multitrack datasets for music separation [166, 283].

CHAPTER 5 presents separation methods that are developed in this thesis. This covers techniques that utilize modulation information, when available. We present a method based on time warping using F_0 estimates for unison mixtures (Section 5.1 [270]) and show how it can be extended for the scenario of vocal and accompaniment separation (Section 5.2). Furthermore, we present a method to improve the precision of F_0 estimates (Section 5.3 [277]).

CHAPTER 6 presents separation methods utilizing modulation when prior information is not available. We present the *Common Fate Model* based on tensor factorization for unison mixtures (Section 6.2 [274]) and also propose an extension based on DNN for vocal and accompaniment separation (Section 6.3).

CHAPTERS 7 presents our listening experiments to find out what the number of sources is that humans are able to identify in music (Section 7.1 [254, 276]) or concurrent speech (Section 7.2 [272, 273]).

CHAPTERS 8 presents *CountNet*, our method to address the source count estimation problem using a data-driven model in a simulated "cocktail-party" scenario [272].

CHAPTER 9 concludes this thesis and gives an outlook into future research directions.

2 FUNDAMENTALS OF OVERLAPPED SOUNDS

In this thesis, the core part is focussed on *analysis* and *processing* of sound recordings of music and speech, commonly referred to as *audio signals*. In this chapter, we introduce basic concepts of digital audio signals which are relevant to apprehend the remaining chapters.

2.1 AUDIO SIGNALS

When a sound wave travels through a medium like air, a signal can be captured using a microphone by measuring the local pressure deviation over time. Such a signal can be written as a function $x(t)$, continuous in both time $t \in \mathbb{R}$ and the amplitude $x(t_0) \in \mathbb{R}$. An *audio signal* is meant to be perceived by the human auditory system — through our ears. Therefore, we can observe specific properties, consistent with the limitations of the human hearing, for example in dynamics as well as in limited signal bandwidth. Many other signals exist with similar characteristics such as signals from finance, geophysics, meteorology or medical data. The result is that audio research is inspired by applications of other fields of signal processing and vice versa.

2.1.1 Digital Representations of Audio Signals

Today, digital representations are used to store, analyze or process audio signals conveniently. A digital audio signal can be obtained from an analog signal using analog-to-digital/digital-to-analog converters (ADC/DAC) which can be found in almost any every-day device such as laptops and smartphones. In short, this process includes two steps: first, the continuous time signal $x(t)$ is converted to a discrete time series, so that one sample[1] x_n is *sampled* with equidistant steps Γ; second, the amplitude values can be *quantized*, resulting in a vector where each element $x_i \in \mathbb{R}$, thus $\boldsymbol{x}$ represents a one dimensional time series of amplitudes. An important parameter in the process of digitization is the sample rate $F_s = 1/\Gamma$ where Γ is the sampling period.

To facilitate the full human hearing range of 20 Hz - 20 kHz [192, 329], due to the Nyquist-Shannon sampling theorem, often, sample

1 Please note, that the use of the word *sample* will have different meanings in the context of machine learning, where a sample is an instance of a full signal instead of a single time step.

rates of at least 40 kHz are chosen. However, for many applications, a lower sampling rate is sufficient, e.g., in speech communication where intelligibility often is more important than quality. For further details, we refer the reader to audio signal processing basics such as Chapter 1 in [215] or Chapter 2 in [193].

2.1.2 Time-Frequency Representation

We often analyze sounds in the frequency domain where the reduced redundancy of the signal improves the computational efficiency of signal processing methods, especially for speech and music that have periodicities. It is common to achieve this through the use of discrete Fourier transform (DFT) and its fast FFT implementation [59] (for details, the reader is referred to Chapter 4 of [215]). Spectral representations also relate to our human auditory system [192, 329], allowing us to process sounds closer to how we perceive them.

The periodicity of real-world sounds, usually only holds for short durations of several milliseconds, often referred to as *quasi-stationarity*. We analyze and process short-time spectra, computed in an overlapped fashion, resulting in a *time-frequency* (TF) representation. The short-time Fourier transform (STFT) is the most commonly used TF representation [183]. It encodes the time-varying spectra into a matrix $\boldsymbol{X}$ with frequencies f and time frames t.

STFT matrices $\boldsymbol{X} \in \mathbb{C}^{T \times F}$ are complex and include phase information. When sounds are processed in the time-frequency domain, the transformation greatly benefits from being invertible to reconstruct a time domain signal. However, analysis and processing is often focussed on the *magnitude* $|\boldsymbol{X}|$ or the *spectrogram* $|\boldsymbol{X}|^2$.

2.1.3 Fundamental Frequency and Harmonicity

Speech and music signals are characterized by its periodicity. And it is this property we perceive as *pitched*. *Pitch* is defined by Klapuri in [149] as

> "a perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from low to high."

It is important to note that *pitch* is a subjective measure. The objective equivalent is referred to as the *fundamental frequency* (F_0)[2]. All frequencies together formed by the integer multiples of the fundamental frequency are named *harmonics* [250]. F_0 can be defined as the lowest frequency/partial of a harmonic signal. An example of a harmonic signal can be seen in Figure 2.1 that depicts a single note

2 Pitch and F_0 are often used synonymously in audio research. Even though this is incorrect, we sometimes may refer to other work where pitch instead of F_0 is used.

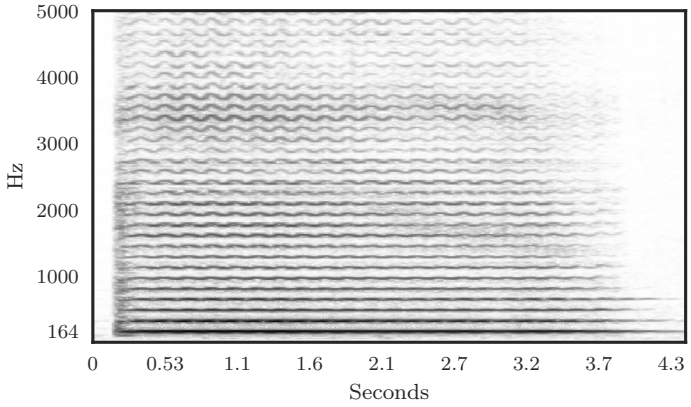


Figure 2.1: Spectrogram of single violoncello note (E3) of a fundamental frequency F_0 of about 164Hz. The vibrato is clearly visible in the upper part of the harmonic spectrum. X-axis shows time (in seconds), y-axis depicts frequency (in Hertz). The audio signal is part of the MUSERC dataset [275].

(E3) played by violoncello. When the fundamental frequency changes, the frequencies of these harmonics change accordingly. This results in the typical comb-like structure of harmonic signals when analyzed in the time-frequency domain. For a detailed overview into the research field of pitch and F_0, the reader is referred to [149].

2.1.4 Time-Variant Audio Signals

Audio signals are considered to be stationary or time-invariant when their properties such as the amplitude of the fundamental frequency of the signal do not change over time. The signal becomes time-variant when an external function changes (modulates) the parameters of a signal over time. This type of modulation was the basis for many break-through inventions such as radio transmission [260]. In the case of audio signals, often, both the modulating function (modulator or carrier) and the signal being modulated (input) are periodic. Signal modulations are often created intentionally but also occur naturally in many real-world audio signals such as speech. In the following, we will present audio modulation categories and their cause, underlining the importance of them.

Audio Signal Communication

Audio modulations play an essential role in the transmission of audio signals such as in radio broadcasting. It is based on the principle of a modulator/demodulator (modem) where a high-frequency carrier signal is modulated by a (lower frequency) audio signal to be transmitted. The modulator varies the amplitude or the frequency of the carrier signal. Let us imagine a sinusoidal carrier signal $x(t) = \sin \omega_c t$ where

ω_c is the carrier frequency. Now, applying a time varying amplitude $a(t)$ results in amplitude modulation (AM):

$$s_{AM}(t) = a(t) \sin(\omega_c t) .$$

In comparison to AM, frequency modulation (FM) varies the frequency of the carrier, so that:

$$s_{FM}(t) = A \sin\left(\omega_c t + p_0 + M_f \int f(t) dt\right)$$

where A is the amplitude, ω_c is the carrier frequency and M_f is called modulation index. When the modulation function $f(t)$ is a single sine wave, $\cos(\omega_m t + p_m)$, of frequency ω_m, the Fourier spectrum of $s_{FM}(t)$, in theory, depends on Bessel functions that do not admit a closed-form expression, being intractable in practice [1]. For audio communication such as FM radio, the modulation frequency or rate is the same as the audio signal being transmitted (audio rate modulations). In music signals, the carrier could be a single note, played by a violin and the modulation signal is the movement of the finger on the fretboard, producing a vibrato effect. In music or speech, these modulations are much slower — typically up to 10 Hz (slow) or up to 100 Hz (medium/fast).

Modulations in Music — Vibrato

Both, frequency and amplitude modulations are a recurrent phenomenon in music as well. In traditional instruments, modulations are known as *vibrato*, defined by [256] as

> "...a periodic pulsation, generally involving pitch, intensity, and timbre, which produces a pleasing flexibility, mellowness and richness of tone."

Pitch and intensity vibrato can directly be mapped to AM and FM, a timbre vibrato, however, is not easily defined and describes a joint AM/FM modulation [64]. Vibrato is an essential playing style for string instruments like a violin. For these instruments, that are usually plucked or bowed, the strings are the primary source of excitation that is modulated in frequency by the player's finger on a fretboard [174]. Modulations are also present in woodwind and brass instruments; however, instead of the excitation signal, the modulations affect the resonator. Many musicians use similar modulation rates to perform a vibrato, usually in the range of 4-8Hz. A detailed overview of the different musical instruments and their modulation characteristics is presented in [87].

Real instruments are not capable of purely amplitude modulated sounds (*tremolo*). Today, however, many instruments are electric or attached to electronic effects where pure modulations can be applied

using digital or analog signal processing. For example, popular electric pianos like the "Fender Rhodes" can include an optional tremolo effect[3]. In its most pure form, synthesizers like [34, 208] allow modulating almost any parameter of a sound using low-frequency oscillators (LFOs) or envelopes that produce sinusoidal, square or triangle functions. One of the most important sound synthesis methods — FM Synthesis — became popular in the early days of digital signal processing. Chowning found in [54] that the modulation of sinusoids using audio rate modulators provides a computationally efficient way of producing fairly complex sounds which mimic, e.g. piano sounds using just four sinusoidal modulators.

It turns out that instrumental vibrato has similar properties compared to vibrato produced in singing voice. Vocal vibrato mainly depends on frequency modulation even though amplitude fluctuations are present [288]. Vocal vibrato rates are similar to that of instrumental vibrato rates with an average of 5 Hz. However, analysis of exceptional voices such as from Freddie Mercury, shows peak rates of up to 7 Hz [113].

Modulations in Speech

Unlike singing voice, speech modulations are part of human communication and therefore part of our language. Modulations in speech include medium to fast modulations of up to a few hundred Hertz, perceived as *roughness* or *residue pitch*. However, often research is focussed on slow modulations around 4 Hz [88, 105] that correlate to the syllable rate [121, 211]. In fact, it was found in [137] that speech is the reason why our human auditory system is so sensitive to amplitude modulations and even our brain is capable of processing rhythm-like envelope fluctuations of the same rate [211, 255].

The importance of Slow Modulations

It is interesting to observe that many modulations have a rate of around 5 Hz. Zwicker found in [328] that humans are very sensitive at detecting amplitude modulations at such a low modulation frequency. This observation can be confirmed when looking at physical modulations that occur when humans suffer from vocal tremor [224] or Parkinson [31]: in both cases, muscle contractions are actuated with the same frequency, indicating that these modulations are natural for humans.

3 Even though it is labeled as *vibrato*.

	Instantaneous	Convolutive
Time-Invariant	$\mathbf{x} = \sum_{j=1}^{J} a_j \mathbf{s}_j$	$\mathbf{x} = \sum_{j=1}^{J} r_j * \mathbf{s}_j$
Time-Variant	$\mathbf{x} = \sum_{j=1}^{J} a_j(n) \mathbf{s}_j$	$\mathbf{x} = \sum_{j=1}^{J} r_j(n) * \mathbf{s}_j$

Table 2.2: Overview of linear mixing models for a mixture $\mathbf{x}$, sources $\mathbf{s}_j$ and a filter response r_j.

2.2 SOURCES AND MIXTURES

In the real world, single isolated audio signals are rare. Instead, we are faced with sets of *sound sources* that make up an *acoustical sound scene*. When multiple sources are active at the same time, the sound that reaches our ears or is recorded using a microphone is superimposed or *mixed* to a single sound. A *mixture* represents a mapping from a set of sources $\mathbf{s}$ to an output signal $\mathbf{x}$. There exist a variety of different mixing models that are utilized in literature.

Usually, these are built upon several assumptions to constrain the scenario and model specific aspects of real-world signals. The most important assumption is that the mixture is the linear sum of all sources. Another differentiation is made between instantaneous or convolutive mixtures. For instantaneous mixtures, all sources are mixed using fixed mixing parameters a_j. This is the typical scenario when sources are mixed using a mixing console. In *convolutive* mixtures, each source $\mathbf{s_j}$ is convolved by a filter response r_j before summation.

Usually, the mixing process is assumed to be time-invariant but for a variety of signals, such as live recordings with moving sources, it can also be time-variant. The mathematical notations of different mixing models are summarized in Table 2.2.

In the remainder of this thesis, we will only consider the linear (instantaneous) case of time-invariant mixing, but many of the methods could be transferred to other cases.

Specifics of Music Mixtures

In music, the process of mixing is an essential step in the process of music creation. Mixing sources is a creative task that involves recording engineers and tonmeisters, and often the artists itself. In today's digital mastering processes, professionally produced music consists of several intermediate mixing steps before the final mixture is produced:

1) **MICROPHONE RECORDING:** in this step, the analog sources are captured and analog-to-digital converted. Vocals and other acoustic instruments are recorded using one or multiple microphones.

Electric instruments such as electric guitars, keyboards or synthesizers may be amplified and then directly digitized.

2) **RAW SOURCE IMAGE:** the digital raw source signals are grouped and mixed into a *source image* (also *stem*). This grouping involves a creative process; hence it is usually done by a recording engineer. The source image is mixed to a specific number of output channels (e.g., stereo) even though the recording may have used less (e.g., vocals) or more than two microphones (e.g., drums). In this stage, a panning is added to position the source images spatially.

3) **MASTERED SOURCE IMAGE:** for each of the images, an additional mastering step is being applied. At this stage, effects such as artificial reverberation are added.

4) **RAW MIX:** the linear sum of all source images are mixed.

5: **MASTERED MIX:** Optionally, further mastering is applied. Often, this step involves non-linear processing such as dynamic range compression.

This emphasizes that the definition of a source is subjective and depends on the application and its context. In this thesis, we mainly deal with tasks where we observe 4) and want to obtain 3) which is a common restriction made in tasks that are concerned with professionally produced music [286].

2.3 PROCESSING AND ANALYSIS OF MIXTURES

While in many ways, mixtures are not different to any other audio signal, two research questions stand out prominently:

- Can we obtain the sources $\boldsymbol{s}_j$ from the mixture $\boldsymbol{x}$?
- Can we find the number of sources J from $\boldsymbol{x}$?

These two questions are addressed in the scientific fields of *sound source separation* and *source count estimation*.

2.3.1 Sound Source Separation

One of the earliest work on audio source separation started in the mid 70s [190]. Since then a large number of contributions were made in this field, both, targeted at speech and music separation. Due to this, it is hardly feasible to give an extensive overview of all existing methods in the context, and the reader is referred to [58, 218, 302].

Source separation methods have relevant applications for music and speech mixtures such as attenuation in hearing aids, karaoke or music creation due to isolated sample composition. It also indirectly helps for related tasks such as upmixing/remixing, improved music transcription or automatic speech recognition (ASR).

In the following, we present three ways to group separation scenarios:

Underdetermined vs. Overdetermined Separation

As mentioned in Section 2.2, generating sound mixtures is closely related to the process of mixing taken place during recording (speech) or with the help of professional recording engineers (music). One assumption which was not mentioned before, is the importance of the number of sensors or microphones used to create the mixture. A source separation problem is *over-determined* when the number of sources is smaller than the number of sensors; *determined* when they are equal. For these two cases, a large number of methods exist and in a closed form solution is possible. The reader is referred to [57], which gives a detailed overview of these methods.
Many real-world source separation problems, however, are *under-determined* and up to date for a large number of scenarios, the problem of separating sources is still very challenging.

In this thesis, we only focus on methods that perform separation on underdetermined mixtures.

Single Channel vs. Multichannel Separation

Today music recordings are mostly produced in stereo. In many music recordings certain assumptions can be made (and utilized) of how sources are balanced between the two channels. E.g., often in popular music, a fixed panning for the vocals is correctly assumed.

As a large number of recording nowadays is still stored in mono, in this thesis, we want to focus on single channel separation.

Blind vs. Supervised Separation

A blind source separation system does not require additional information about the source signals, the location or acoustical environment to perform separation [176]. In practice, blind source separation is ill-posed and it is not generally possible to find a single solution. This is why many proposed methods rely on additional information such as the acoustic environment, the musical score or the fundamental frequency [80, 168].

2.3.2 Vocal Accompaniment Separation

The separation of music into two parts, the foreground lead (e.g. vocals or solo guitar) and the background accompaniment (drums, bass, other), is one of the most relevant scenarios in music separation with a large number of applications such as creating karaoke or a capella tracks. Lead and accompaniment separation has specific issues and assumptions when compared to other separation scenarios like speech. Many music separation methods often rely on knowledge about the mixing process as made in Table 2.2. While there exist many source separation methods that aim to extract the actual raw audio recording, often it is sufficient to extract the source images from the raw mixture. In a live recording, this results in inverting the process of convolution as well. Separation of convoluted mixtures is a very active field in source separation described in [207]. In the context of music separation, however, this becomes less relevant as today's recording and studio mixing environment are mostly digital. Here, the last step in creating music mixtures, as described earlier, is a linear mix. While the source images can yield from a mixing process undergoing the various assumptions, for the case of a mixture of source images, we consider only linear mixing in this thesis. For a more detailed description of this scenario and applications of source image extraction, see [286].

Another characteristic of music separation is that it is typically restricted to a well-defined set of musical sources. Often these restrictions need to be made because not for all kind of music separation scenarios, datasets are available. Thus, the most popular task is to extract the vocals and the background of the music. An extensive overview of music separation methods can be found in [219]. Even though the overview is focused on vocal accompaniment separation, most approaches can be generalized to other sources.

To evaluate separation systems for this scenario, the majority of publications used the Blind Source Evaluation (BSS Eval) toolbox [36, 301] that provides "different and complementary metrics for evaluating separation that measure the amount of distortion, artifacts, and interference in the results" [219].

2.3.3 Estimating the Number of Sources

The number of sources is an important information to be used in source separation and many other related research fields. In real-world applications, information about the actual number of concurrent speakers is often not available.

The *number of sources* $k \in \mathbb{Z}_0^+$ appears to be a clearly defined property of a mixture. However, the meaning of it can differ, depending on its application. Let us assume that we have L sources and a mix-

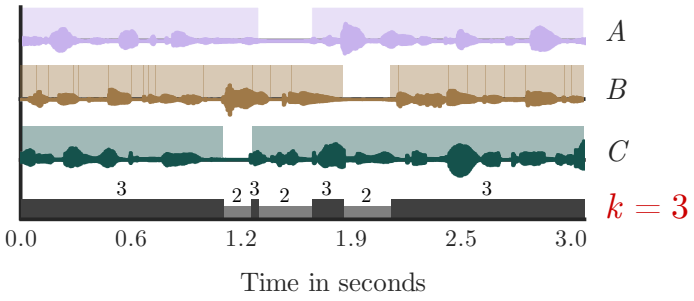


Figure 2.2: Illustration of three concurrent sources (A, B, C) and their respective activity. Bottom plot shows the mixture (input), the number of concurrently active sources and its maximum k. Figure was published in [273] ® 2018 IEEE.

ture of duration N. Further, we imagine a latent binary *source activity* variable $v_{nl} \in \{0, 1\}$ that indicates the activity of each source l and for each time instance n. Now, concerning the number of sources, two definitions of k can be imagined:

A) maximum number of sources, even if not concurrently active. It is simply the sum of all sources that are active at least once within N. This definition is more useful when the sources can be identified or detected first. This definition can also be considered as "counting by detection".

B) maximum number of concurrent sources even if the sources belong to the same class. Here, it represents the maximum of the mixtures concurrency. This definition is more useful as a preprocessing for a separation system since such a system would only require the number of auditory streams and not the number of (non-concurrent) sources. For such approaches it becomes possible to apply separation only when its "needed". This definition can also be considered as "direct count estimation"[4].

At short time scales A) is equal to B) because the instrumentation usually does not change. In Fig. 2.2, we illustrate a setup featuring $L = 3$ unique sources. At any given time, one can see — given definition B) — that at most $k = L = 3$ sources are active at the same time but $k = 2$ could be the outcome if a smaller excerpt would be evaluated. In this thesis, we will pick definition B when concerned about developing methods to estimate the number of sources.

4 Note the subtle difference between "counting", which refers to a sequential process and "count estimation" or "denumerating", which directly relate to an integer.

3 CHALLENGES OF HIGHLY OVERLAPPED SIGNALS

In the previous chapter, I introduced signal processing fundamentals in the context of audio mixtures. In this chapter, I focus on the challenges of highly overlapped signals.

3.1 SEPARABILITY OF MIXTURES

Time-frequency representations such as the STFT have clear benefits such as the improved interpretability due to its "image-like" two-dimensional properties. More importantly, however, such a representation allows to separate mixtures of speech and musical instruments. The reason for this is that these mixtures may be fully overlapped in the time domain but are less overlapped in the frequency domain. In turn, a time-frequency representation allows to apply a filter in a way that it sufficiently extracts all targets from the mixture. Furthermore, it allows for the reconstruction of the original waveform and provides a good trade-off between computational complexity and separation quality.

Due to these reasons, many source separation methods focus on extracting individual sources by modeling their respective target in the time-frequency domain. Further, it is assumed that the STFT provides a sufficient level of separability. The actual extraction or filtering is done by synthesizing the magnitude estimate of the model and applying the originals mixture phase.

In practice, the ability to extract a source from a mixture depends on the amount of overlap between sources. Without any overlap, separation is not necessary, and a small amount of overlap can be tolerated to extract the sources still sufficiently. However, if sources are fully overlapped in both, time and frequency, a separation in the TF domain is hardly possible. A metric that is often used for evaluation is called *separability* and was found by Rickard in [231] as a useful metric for both, speech and music [94] signals.

In linear mixtures, separability is defined as *a measure that indicates the percentage of time-frequency bins of a source is disjoint from those of interfering sources* and calculated through the W-disjoint orthogonality metric *WDO* in [231].

If $\boldsymbol{M} \in \{0,1\}^{m \times n}$ is an ideal binary mask [314] for a given target $\boldsymbol{S}$ and its interfering magnitude $\boldsymbol{Y}$ of same dimensions as $\boldsymbol{M}$, the W-disjoint orthogonality metric WDO is defined as:

$$PSR_M = \frac{\|\boldsymbol{M} \otimes \boldsymbol{S}_k\|^2}{\|\boldsymbol{S}_k\|^2} \tag{3.1}$$

$$SIR_M = \frac{\|\boldsymbol{M} \otimes \boldsymbol{S}_k\|^2}{\|\boldsymbol{M} \otimes \boldsymbol{Y}_k\|^2} \tag{3.2}$$

$$WDO_M = PSR_M - \frac{PSR_M}{SIR_M} \tag{3.3}$$

Where the PSR is the reserved-signal ratio, and SIR is the signal-to-interference ratio and $\otimes$ being the element-wise product. A WDO of one means the sources are entirely disjoint, hence no overlap. A WDO zero means can be interpreted as sources being fully overlapped.

The ability to separate sources is depending on the scenario and its applications. Let us consider the following scenarios:

COCKTAIL PARTY where multiple speakers are speaking concurrently, it results in a partial overlap of speech signals in both time an frequency.

VOCALS AND ACCOMPANIMENT are often active at the same time in professionally produced music.

UNISON INSTRUMENT MIXTURES have a severe overlap in almost all active time-frequency bins.

Now, for these scenarios, the actual overlap depends on additional parameters like the number of sources, the class of source or the fundamental frequency. For instance, the overlap in a cocktail party of two speakers is smaller than ten concurrent speakers speaking. Also, the overlap between male and female or brass and string instruments is smaller than two instruments of the same class. And if two instrumental notes share the same fundamental frequency (playing in *unison*), the sources are almost entirely overlapped.

To illustrate this, we depict the different scenarios in two complementary figures. Figure 3.1 assigns each time-frequency (TF) entry to its predominant source. Figure 3.2 depicts the number of active sources (thresholded) of each TF entry. From these figures, one can see that the overlap of a typical speech mixture is comparable to a music recording where the task is to separate vocals and accompaniment. If we now compare this to the scenario where sources are fully overlapped as in the unison scenario, almost all TF bins are overlapped, and separation would hardly be possible.

While this is an extreme scenario, it still provides a useful example where common assumptions are violated, and it would facilitate the

demand to develop new methods that do not rely so much on these assumptions. By naïvely observing the time-frequency representation in Figure 3.2 closely, we see that the slow spectro-temporal modulations caused by the vibrato are one of the aspects where the two sources differ. Here, the classical STFT does not provide sufficient separability and representations as in the *modulation spectrogram*, presented in [104] may be preferable. Details about this approach are discussed in Chapter 6.

3.2 EXPLOITING SLOW MODULATIONS

Tempo-spectral modulations occur both in speech and music signals as detailed in Section 2.1.4. Exploiting modulations is natural for humans: early research from Zwicker in 1952 focused on the human ability to detect amplitude modulations [328]. Later, it was shown by Bregman, McAdams, and Fastl in [32, 182, 329] that humans use amplitude modulations to group sources; this concept was called *Common Amplitude Modulation* (CAM). CAM exploits the fact that harmonics that share the same amplitude modulation across frequency bins are perceived *integrated* as opposed to *segregated*. Further, it was shown in [15] that the ability to detect amplitude modulations can be incorporated into auditory models. It was then found by Dau in [60] that humans are especially sensitive at low-frequency modulations:

> "Slow modulations are associated with the perception of rhythm. Samples of running speech, for example, show distributions of modulation frequencies with peaks around 3 Hz to 4 Hz, approximately corresponding to the sequence rate of syllables [211]. Results from physiological studies have shown that, at least in mammals, the auditory cortex seems to be limited in its ability to follow fast temporal changes."

Dau proposed a model that mimics the ability to detect modulation patterns and pointed out applications to improve the perception for hearing-impaired listener or speech intelligibility.

Previously, research has addressed a variety of tasks of processing and analysis in the context of modulations. In the following, we give an overview of existing work focussed on analysis and separation of modulated sounds.

3.2.1 Analysis

In speech, techniques using modulation patterns improved applications such as speech discrimination [189] or extract spatial acoustic

signatures from mixtures [287]. One way of analyzing amplitude modulations is to use a modulation spectrogram [105] which is a frequency-frequency representation of a time domain input signal. In practice, the modulation spectrogram can be computed from a (magnitude) time-frequency matrix $\mathbf{X}_{f,t}$ by computing f time frequency transforms over each frequency band of $\mathbf{X}$ resulting in a tensor $\mathbf{V}_{f,b,t}$ where b represents the modulation index. The use of modulation spectra helps to identify amplitude modulations such as the one (indirectly) caused by vibrato. The modulation spectrogram has already gathered much attention in speech recognition [105, 144] and classification [145, 178].

Interestingly, Greenberg in [105] assumed that "the energy in the modulation spectrum may be derived from syllabic segmentation" and from "the preservation of the portion of the modulation spectrum of 2 Hz to 10 Hz". Following this, it was later proved that the detection of modulations improves speech intelligibility [77] or automatic speech recognition [144].

The analysis of amplitude modulations were also proposed for music tasks. Work by Scheirer in [249] proposed a method that operates by utilizing common modulation among groups of frequency sub-bands in the auto correlogram domain. In music, where modulations are predominantly caused by vibrato, frequency modulation is important. For frequency modulations, however, the modulation spectrogram is less effective, as it would only be able to track the modulation through side-lobes. Here, a common way to explicitly analyze frequency variations is first to analyze the fundamental frequency and then track the fundamental frequency over time to smoothen out the contour. An overview of techniques is summarized in [70]. The authors of this paper also proposed a novel method to directly estimate the parameters of potential frequency modulations in the time-frequency domain by matching sinusoidal templates.

Disch and Edler proposed in [66] to decompose an audio signal into bandpass signals, each of them parametrically modeled by a sinusoidal carrier and its amplitude and frequency modulation.

3.2.2 Processing

As described in the previous chapter, modulations are used by humans to group and segregate sounds. Viste et al. describes the impact of modulation in [308] as:

> "harmonic relation, the common onset, offset, AM, and FM. These are all important cues for grouping."

It is therefore not surprising that a number of methods exist, that utilize spectro-temporal modulations to separate mixtures. These methods were summarized in [218], starting with one of the first concepts introduced by [32] as the *common amplitude modulation* "which exploits

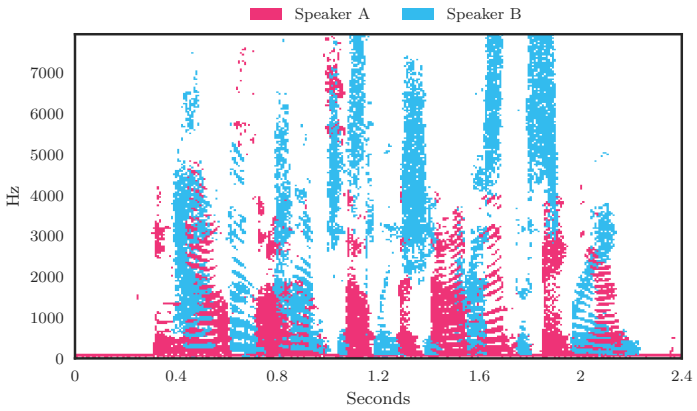
that amplitude envelopes of different harmonics of the same source tend to be similar." This was later used in models to separate mixtures such as in [162, 163].
Furthermore, common amplitude modulation characteristics was included in the separation scheme in works such as [39].
Wang proposed a technique in [310, 311] of "...instantaneous and frequency-warped techniques for signal parameterization and source separation, with application to voice separation in music."
Yen et al. proposed in [322, 323] to use spectro-temporal modulation features to decompose "a mixture using a two-stage auditory model which consists of a cochlear module [50] and cortical module [49]." (from [218]).
Virtanen made use of sinusoidal modeling [305] to model and separate sources with spectro-temporal modulation-like vibrato.

In another vein, the source-filter model was deployed to source separation in [111]. An advantage of the source-filter model, as pointed out in [218] is that "...one can dissociate the pitched content of the signal, embodied by the position of its harmonics, from its TF envelope which describes where the energy of the sound lies."

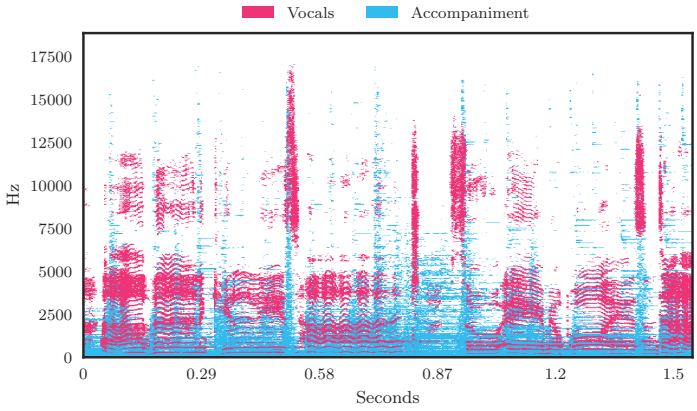
3.3 SUMMARY

Modulations play an essential role in audio signals. However, past research was mainly focused on single notes and not on overlapped sounds. It is, therefore, to be investigated if scenarios with severe overlap can utilize modulations as well: parameterization of modulation characteristics of a single source is difficult when only the mixture can be observed. It is known [241] that the extraction of the fundamental frequency in a mixture is challenging. The reason is that crossing partials are a challenging problem for sinusoidal modeling [308]. Also if tracking of them would work correctly, evaluation of robustness and accuracy is hardly possible when the reference data is annotated with human precision. Furthermore, representations like modulation spectrograms only cover amplitude modulations, whereas general modulation patterns (AM/FM, timbre modulation) cannot be covered.

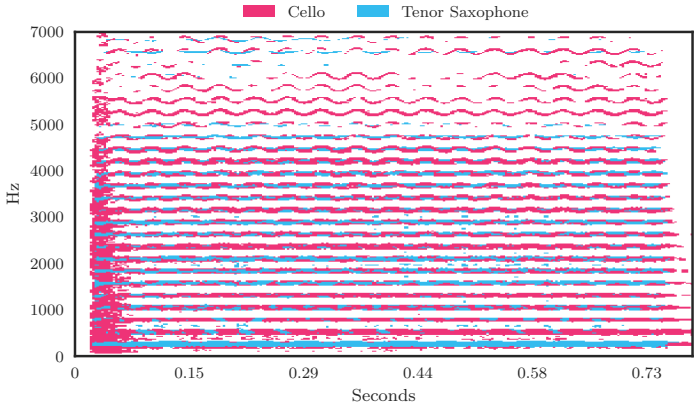
In the next two chapters, we address utilizing the modulations of sources for separation tasks; either via prior knowledge (known) or by operating blindly (unknown).



(a) Speech

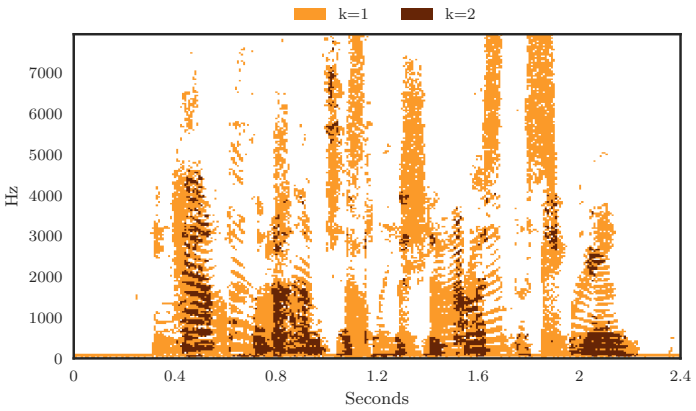


(b) Vocal/Accompaniment

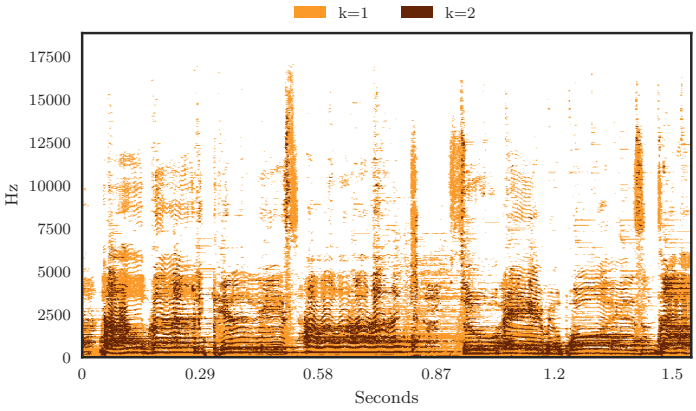


(c) Unison

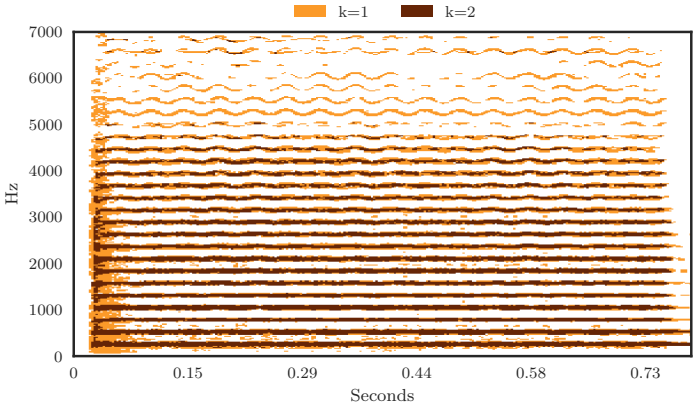
Figure 3.1: Predominant source activity, showing the predominant source for each time frequency entry. Computed using binary masks of each source entry.



(a) Speech



(b) Vocal/Accompaniment



(c) Unison Instruments

Figure 3.2: Source Count Activity showing the number of sources k for each time frequency entry. Computed using binary masks of each source entry.

4 DATASETS

In the previous chapter, we presented the fundamentals of highly overlapped audio signals. To study these signals in a reproducible manner, suitable data is of paramount importance since many methods rely on audio datasets for development and evaluation. However, suitable audio material is not publicly available, in the case of music, often because of restrictive copyright laws. Therefore, in the past, many academic endeavors were focussed on assembling such precious data. In fact, in the audio community, dataset contributions now are essential to accelerate many research directions.

In the following sections, we present three datasets which we helped creating. Each of the three datasets is released in public domain and aims to fill specific needs and is used in subsequent experiments throughout this thesis.

4.1 UNISON MIXTURES

Parts of this this section were previously published in [278].

In speech, it is common to mix clean speech and noise [297] or different clean speech signals such as [91] to generate mixtures. By contrast, the conversational aspect of human-to-human communication is lost. Compared to speech, musical content usually does share familiar orchestration, can hardly be superimposed randomly and the summing of isolated random notes from musical instrument databases does not reflect musical performances. On the positive side, there are use-cases for single note datasets such as for evaluation of fundamental frequency estimation algorithms or the detection of instruments. Furthermore, single note datasets allow using the data to synthesize musical score as long as the recordings have enough variance of expressions. It also allows to quickly generate a large number of mixtures using randomly permuted mixtures, fostering applications in machine learning.

As an exception compared to other music scenarios, when all instruments play *in unison*, single note datasets are appropriate to approximate real mixtures:

- A random summation of multiple instruments playing the same note does not necessarily differ from realistic unison mixture.
- When notes are played with vibrato, having access to the individual modulation patterns can help to study the influence of modulations systematically.

Instrument	Vibrato	MIDI #
Violin	yes	40
Viola	yes	41
Violon Cello	yes	42
Trumpet	no	56
Trombone	no	57
Horn	no	60
Bariton Sax	yes	67
Oboe	no	68
Clarinet	no	71
Flute	yes	73

Table 4.1: Selected Instruments from the *Unison Source Separation Dataset* [279] as used in [270, 274].

- Unison mixtures are part of many classical compositions, to extend the timbre of a note.

One way to assemble a dataset is to create random mixtures of single notes compiled from existing datasets such as the *Univ. of Iowa Musical Instrument Sample Database*[1]. However, to better study the influence of vibrato we require extended control over certain parameters such as note duration, vibrato duration, exact fundamental frequency, vibrato rate, vibrato extend, reproducibility, loudness or expression.

As mentioned in the previous chapter, vibrato techniques vary across instruments. Instruments such as violin and saxophone are known for their distinct frequency modulations [95]. Other instruments such as the English horn and the flute are more close to amplitude modulations.

We generated the notes using a software sampler[2] which allows us to control the parameters such as the vibrato. All our test stimuli have a duration of three seconds. Items were equalized in loudness by using an iterative calculation of the loudness algorithm of the time-varying Zwicker model [329]. We used an implementation released in [93].

We rendered 29 notes of C4, resulting in 841 unique unison instrument mixtures per pitch class. An excerpt of the instruments is listed in Table 4.1. The dataset is available from [279].

To evaluate the level of overlap, we created a small experiment where we computed the average W-disjoint orthogonality WDO metric for 1000 random combinations of mixtures for different separation scenarios. It turned out that for two sources, in speech, we observe $WDO = 0.9$ and for the lead and accompaniment scenario

1 https://web.archive.org/web/20191211134945/http://theremin.music.uiowa.edu/

2 Vienna Symphonic Library: https://web.archive.org/web/20191029200706/https://www.vsl.co.at/en

$WDO = 0.87$. These numbers are surprisingly similar even though both scenarios are so fundamentally different. In the case of two instruments playing in unison, the average WDO is 0.65, indicating that a good separation in the time-frequency domain is more challenging, thus making the dataset a useful addition compared to existing scenarios.

4.2 HIGH RESOLUTION VIBRATO RECORDINGS

Parts of this this section were previously published in [275].

Fundamental frequency F_0 estimation of a signal is a common task in audio signal processing with many applications. If the F_0 varies over time, the complexity increases, and it is also more difficult to provide ground truth data for evaluation.

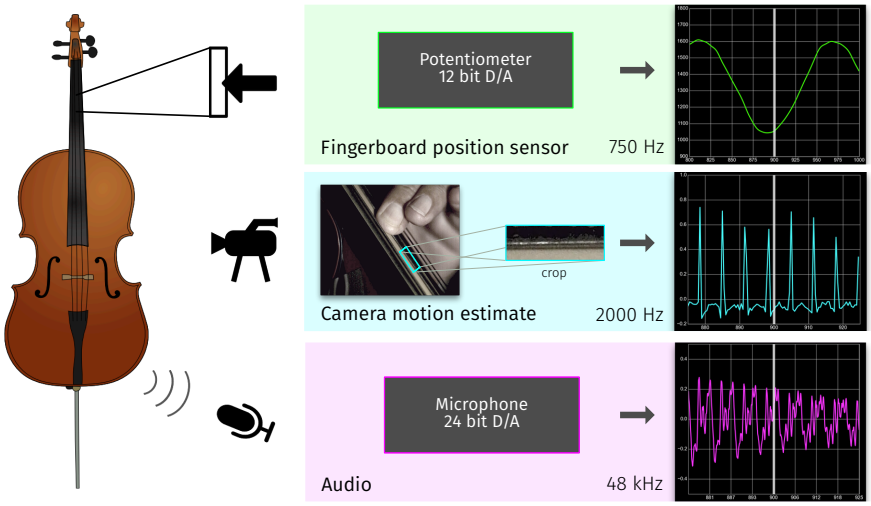


Figure 4.1: Overview of the multi-modal data recorded for the proposed dataset.

For speech signals, an EGG device (also known as laryngograph) captures the excitation of the human vocal tract. This signal is then processed by an F_0 estimator to generate the ground truth. Such a method is accepted in published research because the retrieved F_0-trajectory based on the EGG signal is easier to process and the generated annotations are considered as a good ground truth [13, 209]. Motivated by this, we proposed a new dataset for musical instruments where we recorded a violin cello with extra sensors on the fretboard in addition to audio and video. We made use of multiple sensors to capture the most relevant processes involved in creating time-varying output signals as depicted in Figure 4.1. We included sensor recordings capturing the finger position on the fingerboard which is converted into an instantaneous frequency estimate. We also included high-speed video camera data to capture excitations from the string at 2000 fps. Recording video data was inspired by the work of Davis et. al. in [61]

presenting a "visual microphone" which is able to observe sound solely with a camera, pointing to objects in the sound field.

In the proposed dataset we chose the violin cello for the following reasons: (1) vibrato is used as a common style for expression, (2) there is an observable physical relationship between frequency modulation and vibrato, and (3) the instrument is large enough to embed sensors to capture the vibrato. The properties of the cello are studied by research in acoustics [316, 317].

To capture significant aspects of the cello while being played by a musician, we focus on three main observations: (1) Excitation caused by the moving bow; (2) the vibrating string, and (3) the finger, controlling the string length by rolling it on the fingerboard. The main focus of the recordings is to analyze vibrato playing style. Since it is common that vibrato characteristics differ from musician to musician, all recordings were performed by two musicians. One is a professional cellist with 30 years of experience in a symphonic orchestra[3]. The other recording was done by the author of this thesis, who has less than 1 hour per week of practice.

Due to the width of fingerboard sensors and the attached cables we were able to equip two strings (G and A) allowing to record pitches $G2, D3, D^{\sharp}3, E3, A3, B3, C4, C^{\sharp}4$ from both musicians (see the middle part of Figure 4.1).

The dataset includes time synchronous fingerboard positions and high-speed camera recordings. The derived motion estimates show similarity to the EGG signals used in speech. The slowest feature rate of the set is 750 Hz, which enables to evaluate F_0 estimators with high temporal resolution.

In [275], we also showed how to derive high resolution F_0 contours from the data which can be used to improve F_0 estimators or help to analyze playing styles in recordings, usually relying on conventional F_0 estimators based on the audio signal [186]. By using sensor data samples from our test set, researchers get more robust and detailed data to compute features like mean vibrato frequency. Further, it can be used for synthesizers to add a natural vibrato by using the sensor data as a modulation source.

The resulting test set yields in 148 recorded notes after removal of some notes due to errors in the sensor recordings. By making this dataset public domain [284] and including the raw recordings, we believe other researchers can benefit from the data and possibly generate their own derived data.

3 https://web.archive.org/web/20170317054701/http://bambergerstreichquartett.de/de/Das_Quartett/Karlheinz_Busch

4.3 MULTITRACK MUSIC RECORDINGS

One of the core problems for the field of music processing is the lack of publicly available datasets. Many researchers aim to develop methods that could be applied to professionally produced music. However, at the same time access to professionally produced music recordings is difficult due to the complex copyright laws established by the music industry. When music is digitally stored and publicly available, such as on *youtube*, researchers can more easily use this data for academic purposes such as in [20]. Unfortunately, these platforms only host the stereo mixes of produced recordings, whereas the stems or source tracks that are used to create the master mix are a carefully guarded secret. In the case of very old recordings, the recordings were down-mixed (to tape) during the recording sessions, thus making the original stems unavailable. Now, the lack of available multitrack datasets prevents research on source separation to advance further. This is especially true for supervised methods but also affects objective evaluation where the true sources would need to be available.

The SiSEC is a publicly organized benchmark to assess the performance of source separation systems [166, 197, 199, 283]. Through this campaign, a multitrack dataset was compiled starting with the MASS dataset [304] that was used in one of the first campaigns in 2009 [299]. Up until the release of MedleyDB [25] in 2014, researchers did not have access to a large number of full-length multitrack recordings. Since then we helped to aggregate such data from multiple sources to compile the DSD100 dataset [166] which was the first dataset that could successfully be used for data-driven separation methods (See Table 4.2 for a comparison to other datasets). We compiled DSD100 to include four predefined targets: bass, drums, vocals and other. The full-length tracks enable to exploit long-term musical structures and also allow to focus on evaluation of silent parts. Many musical genres are represented: jazz, electro, metal, etc. and it is split into a training and a test set for the design of data-driven methods.

Over the years, DSD100 and its successor MUSDB18 [220] became one of the most used datasets for source separation. The dataset is still small in comparison to machine learning sets from vision such as [62] but it proved to be large enough to help DNN-based methods to reach breakthrough results in source separation [283].

Working with multitrack audio files can be cumbersome due to its hierarchical structure that needs to be parsed. For that purpose, we developed a software toolbox for Python that permits the straightforward processing of the DSD100/MUSDB18 dataset. This software is open source and was publicly broadcasted so as to allow the participants to run the evaluation themselves[4].

4 github.com/faroit/dsdtools / github.com/sigsep/sigsep-mus-db

Dataset	Year	Tracks	Track duration (s)	Full/stereo?
MASS [304]	2008	9	16 ± 7	no / yes
MIR-1K [123]	2010	1,000	8 ± 8	no / no
QUASI [167, 300]	2011	5	206 ± 21	yes / yes
ccMixter [170]	2014	50	231 ± 77	yes / yes
MedleyDB [25]	2014	63	206 ± 121	yes / yes
iKala [45]	2015	206	30	no / no
DSD100 [166]	2015	100	251 ± 60	yes / yes
MUSDB18 [283]	2017	150	236 ± 95	yes / yes

Table 4.2: Summary of datasets available music source separation datasets. Tracks without vocals were omitted in the statistics.

This package integrates with existing Python code, thus makes it easy to participate in SiSEC. The core of this package is calling a user-provided function that separates the mixtures from the dataset into several estimated target sources.

All details of this accompanying software tools may be found on its dedicated website[5].

5 https://sigsep.github.io

5 SEPARATION BY KNOWN MODULATION

For many source separation methods it is common to assume that the spectral harmonics are not fully overlapped. In turn, this assumption is exploited in methods such as non-negative matrix factorization (NMF) to approximate the mixture from a lower-rank decomposition in an unsupervised way. Still, in order to improve separation quality, researchers imposed additional constraints based on prior information about the sources in their algorithms [202]. The extent of such meta information very often depends on the availability of data. One example of an informed source separation system is described by Ewert and Müller [79]. They proposed to incorporate the note pitch and onset information of the musical score, encoded in a MIDI file, synchronized to the audio, to improve the separation result. In the case of highly overlapped signals such as unison mixtures, the score is less useful since unison mixtures share the same note pitch. Instead, in this chapter, we want to evaluate the use of fundamental frequency estimates of the source to be extracted. There has been extensive research on separation using the fundamental frequency. The first option is to use a sinusoidal model which was studied in a large number of methods. However, sinusoidal synthesis is known to suffer from "a typical *metallic* sound", according to [218]. Alternative approaches, instead, are "filtering out everything from the mixture that is not located close to the detected harmonics" in order to exploit harmonicity. In the past, many related works focused on this paradigm, a procedure as it turned out to be a common task in source separation systems. Before we present our proposed method in the next section, the following paragraphs from [218] (Section III b), give a comprehensive overview of existing methods in this field:

> "E.g. Li and Wang proposed to use a vocal/non-vocal classifier and a predominant pitch detection algorithm [161, 162]. They first detected the singing voice by using a spectral change detector [74] to partition the mixture into homogeneous portions, and GMMs on MFCCs to classify the portions as vocal or non-vocal. Then, they used the predominant pitch detection algorithm in [160] to detect the pitch contours from the vocal portions, extending the multi-pitch tracking algorithm in [318]. Finally, they extracted the singing voice by decomposing the vocal portions into TF units and labeling them as singing or accompaniment

dominant, extending the speech separation algorithm in [125].

Han and Raphael proposed an approach for de-soloing a recording of a soloist with an accompaniment given a musical score and its time alignment with the recording [109]. They derived a mask [236] to remove the solo part after using an EM algorithm to estimate its melody, that exploits the score as side information.

Hsu et al. proposed an approach which also identifies and separates the unvoiced singing voice [123, 124]. Instead of processing in the STFT domain, they use the perceptually motivated Gammatone filter-bank as in [125, 162]. They first detected accompaniment, unvoiced, and voiced segments using an HMM and identified voice-dominant TF units in the voiced frames by using the singing voice separation method in [162], using the predominant pitch detection algorithm in [68]. Unvoiced-dominant TF units were identified using a GMM classifier with MFCC features learned from training data. Finally, filtering was achieved with spectral subtraction [248].

Raphael and Han then proposed a classifier-based approach to separate a soloist from accompanying instruments using a time-aligned symbolic musical score [226]. They built a tree-structured classifier [33] learned from labeled training data to classify TF points in the STFT as belonging to solo or accompaniment. They additionally constrained their classifier to estimate masks having a connected structure.

Cano et al. proposed approaches for solo and accompaniment separation. In [40], they separated saxophone melodies from mixtures with piano or orchestra by using a melody line detection algorithm, incorporating information about typical saxophone melody lines. In [41, 67, 106], they proposed to use the pitch detection algorithm in [69]. Then, they refined the fundamental frequency and the harmonics and created a binary mask for the solo and accompaniment. They finally used a post-processing stage to refine the separation. In [42], they included a noise spectrum in the harmonic refinement stage to also capture noise-like sounds in vocals. In [39], they additionally included common amplitude modulation characteristics in the separation scheme.

Bosch et al. proposed to separate the lead instrument using a musical score [30]. After a preliminary alignment of the score to the mixture, they estimated a score confi-

dence measure to deal with local misalignments and used it to guide the predominant pitch tracking. Finally, they performed low-latency separation based on the method in [180], by combining harmonic masks derived from the estimated pitch.

Vaneph et al. proposed a framework for vocal isolation to help spectral editing [296]. They first used a voice activity detection process based on a deep learning technique [157]. Then, they used pitch tracking to detect the melodic line of the vocal and used it to separate the vocal and background, allowing a user to provide manual annotations when necessary."

5.1 F_0 INFORMED SEPARATION

Parts of this section were previously published in [270] and were revised for this thesis.

Frequency modulation caused by vibrato is a very common playing style for string instruments but also for woodwind and brass instruments. Vibrato is an effect that is well studied especially in musicology, for more information the reader is referred to the overview given in the Section 2.1.4. Performers are able to perform a vibrato in the same way when repeating a performance [87]. For example, vibrato rates vary across different instruments. In [174] the vibrato width (frequency deviation) was found to be significantly different between violinists and violists performers. This can be exploited in source separation scenarios.

However, in the case of NMF, it lacks the ability to model time-varying frequencies (See details in Chapter 6). Several extensions for NMF have been proposed to improve the decomposition quality. Hennequin et al. proposed in [112] a frequency dependent activation matrix, whereas Smaragdis et al. developed a variant of the NMF in [266] which is invariant to frequency shifts. Another approach is to model the spectral pattern changes by Markov chains [194]. All these approaches attempt to model the non-stationary effects within the decomposition model. In this work, instead, we propose a method that increases the stationarity of the signal in a preprocessing step and then use standard separation methods such as NMF for the decomposition.

5.1.1 Time Warping

The idea is to make use of *time-warping* which refers to a mapping of the linear time scale t to a warped time scale τ via a mapping function $\tau = w(t)$. To ensure a unique mapping, the mapping function needs to be strictly increasing. For the discrete time case the mapping can be achieved by a time-varying re-sampling of the linear (i.e. regularly

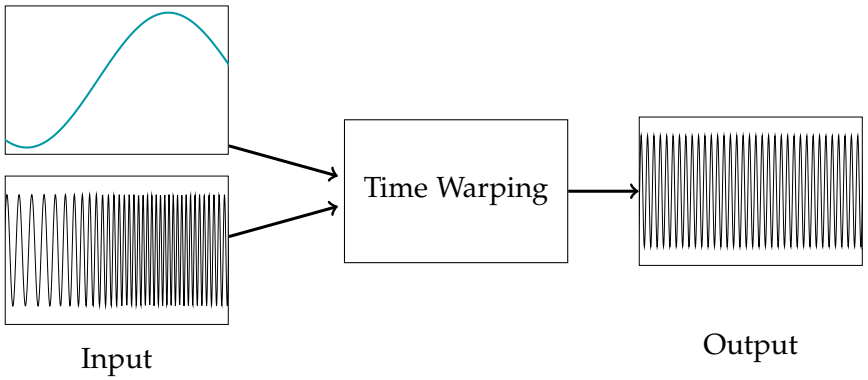


Figure 5.1: Example of applying time warping to an input signal (bottom left) by using a frequency variation contour (top left) resulting in a signal of constant fundamental frequency at the output (right).

sampled) time signal under consideration. The instantaneous sampling frequency then corresponds to the first derivative of the mapping function. Although the mapping can be done from any time-span I on the linear time scale to any time span J on the warped timescale, in the discrete time case, it is advantageous to have the same number of samples in the linear and warped time domain. This ensures that the average sampling frequency is the same in both domains. Such time-warping approaches have already been proposed for different purposes such as transform-based audio coding [76]. As in these applications, we derive the mapping function from the varying instantaneous fundamental frequency in such a manner that the variation of the frequency is reduced or removed. To be more precise the actual information needed is not the absolute instantaneous fundamental frequency but only its change over time. The discrete time warp map $w[n]$ is then simply the scaled sum of the relative frequencies (warp contour) $W[n]$:

$$w[n] = N\frac{\sum_{l=0}^{n} W[l]}{\sum_{k=0}^{N} W[k]} \qquad 0 \leq n < N, \tag{5.1}$$

where N being the number of samples of the signal under consideration. From the requirements for the mapping function it follows that the relative frequency $W[n]$ has to be positive at all instants and preferably should not exhibit large jumps. For the mapping from linear to warped time, now the linear domain sample points $s[\nu]$ for the regularly spaced samples $x[\nu]$ in the warped domain are found by inverting $w[n]$. These sample points are then used to re-sample the linear time domain samples $x[n]$ to the warped time domain samples $x[\nu]$, in our case by employing 128 times oversampled FIR low-pass filter. This processing leads to a sampling rate contour which is proportional to the F_0 contour. Or in other words, a fixed number of samples are obtained in each period of the signal with the varying fundamental

frequency. Mutatis mutandis the sample points $s[\nu]$ can be used for the re-sampling from the warped time domain to linear time domain.

In this work, the time-warping was done globally over the full lengths of the signals under consideration. The globally time-warped sample sequence was then used in the further processing steps. In Figure 5.1 we show the results of the warping process in the time domain.

The use of time variable rate sampling was first proposed by [319] which used this method to analyze FM signals.

A similar approach using frequency modulation to separate a harmonic source from a mixture was proposed in [311]. Here the individual lines are demodulated to the baseband using a combined frequency tracking/demodulation approach. The difference to our approach is that first the absolute instantaneous frequency for every harmonic line has to be known instead of a relative frequency that is common to all harmonic lines of a single source. This relative frequency might be obtained easier than its absolute value for a mixed signal. Secondly every harmonic line has to be individually frequency demodulated while in our approach the full signal is frequency demodulated in one algorithmic step.

5.1.2 Separation

With the ability to remove the frequency modulation from a signal, we included time warping in a source separation system to address the non-stationarity issues of NMF based approaches. Figure 5.2 shows how this system works on a purely harmonic FM signal mixture. Plots (a) and (b) show the two input signals which are linearly mixed (c). For each source, the warp contour needs to be calculated. The mixture is then warped with F_0 variation estimates of source 1 (d) and source 2 (e). The actual separation/filtering of the sources is then done by using NMF which is not shown here. To separate the components from the warped mixture we used NMF on a STFT computed with a long DFT (about 0.5 s). We applied NMF fully unsupervised clustering components based on tonality of $\mathbf{W}$ by using a spectral flatness measure [103]. The separated signals (f) and (g) then need to be warped back into the original time domain resulting in (h) and (i).

It is important to clarify that this approach would not be able to separate two modulating instruments playing in unison without having prior knowledge about the individual modulation functions. Although a F_0 variation estimate might be difficult to achieve in a mixture, our approach shows that such a system works if that estimate is accurate.

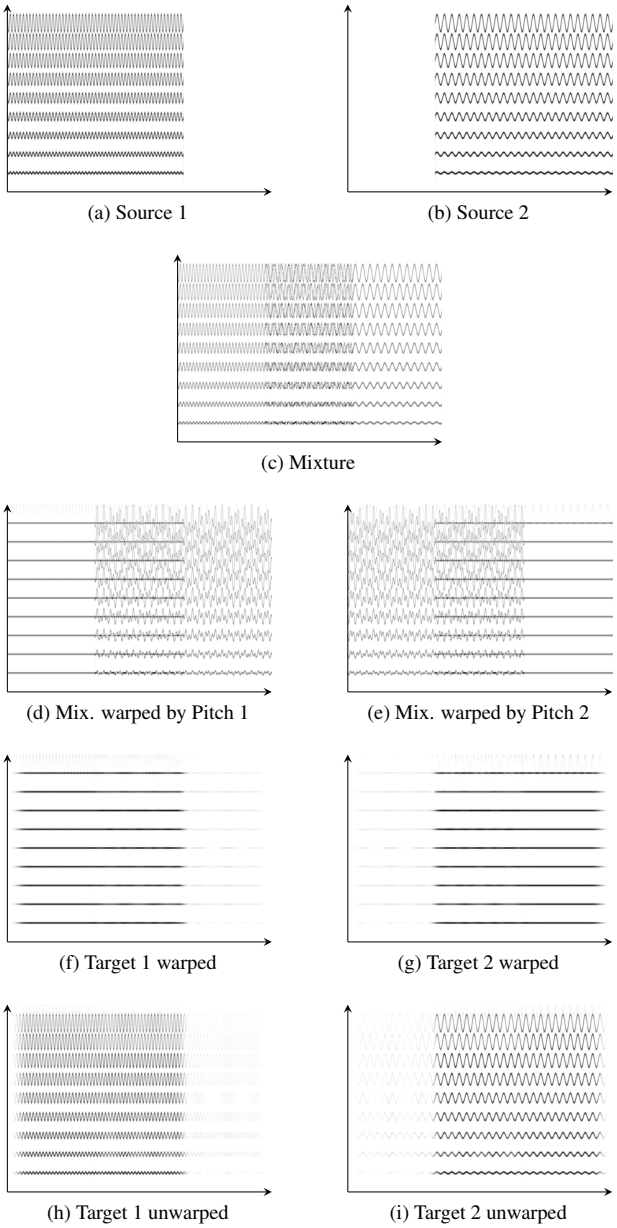


Figure 5.2: Example of F_0 variation informed NMF in the warped domain. *Time* is shown on horizontal axes. *Frequency* is shown on vertical axes.

5.1.3 Evaluation

We use the unison test set [279] as described in Chapter 4 and selected 10 stimuli as noted in Table 4.1.

We evaluated the method in terms of separation quality. Like in [21] we choose not to address the problem of clustering the components after the matrix factorization operation. Instead of processing mixtures in a $A - B - AB$ or $A - AB - B$ paradigm we went for a supervised learning phase where we had access to the original source individually. In this *oracle* supervised approach for each of the sources we then learned the spectral, temporal components and concatenated them. The learned coefficients were then used to initialize the final factorization process. This way we can achieve the upper bound separation result.

The test set was processed by two algorithms: standard NMF and the proposed F_0 variation informed NMF (PVI-NMF). The factorizations for NMF were computed by minimizing the $\beta = 1$ divergence (Kullback-Leibler divergence). We choose to calculate results with $K = 2$ and $K = 4$. The F_0 variation estimator is based on a method that was proposed by Bäckström in 2009 [14] with a subsequent post-processing to ensure the smoothness of the mapping.

Both algorithms did perform on the same filter bank output and with the same sample rate. The NMF approach did use a 2048 STFT with 512 samples hop size. All methods use soft masking/wiener filtering for the actual synthesis.

The results were evaluated by using commonly used evaluation measures provided by the PEASS Toolbox [78] and mean values are provided in Table 5.1. The used enlisted objective metrics are Source to Distortion Ratio (SDR), Source to Interferences Ratio (SIR) and Sources to Artifacts Ratio (SAR). Additionally, we also computed metrics with a strong correlation to auditory perception such as the Overall Perceptual Score (OPS), the Target-related Perceptual Score (TPS), the Interference-related Perceptual Score (IPS), and the Artifacts-related Perceptual Score (APS). It can be seen that the SDR values give a different tendency than the OPS score, showing that the differences between both measures are substantial. Since unison mixtures are even very challenging for humans to segregate, we chose to focus on the psycho-acoustically weighted performance measures only. The results show a slightly better overall performance for the PVI-NMF. The results have also been evaluated and confirmed subjectively by informal listening. Additionally, we provide selected stimuli online on an accompanying webpage [1]. In general, the PEASS scores give a good indication of quality. However, the artifacts that are introduced by the

1 https://web.archive.org/web/20191211135506/https://www.audiolabs-erlangen.de/resources/2014-DAFx-Unison/

Metric	NMF	PVI-NMF
SDR	**2.96**	2.54
SIR	**2.31**	1.80
SAR	22.87	**23.35**
OPS	15.76	**17.64**
TPS	30.17	**32.80**
IPS	26.07	**27.03**
APS	46.14	**54.74**

Table 5.1: Average results from evaluation using PEASS 2.0 Toolbox [78]. Best performing algorithm is marked bold.

standard NMF synthesis seem to be not well reflected. One possible reason is that the PEASS Toolbox has not been tested on artifacts from unison mixtures.

5.2 EXTENDING F_0 INFORMED SEPARATION

In the previous section, we showed the effectiveness of a F_0 variation informed separation system on our constrained unison source separation scenario. In the following section, we show how the method can be extended to the scenario of separating the vocals/lead from the accompaniment by using predominant melody estimation as depicted in Figure 5.3.

In this extension, we want to show how the F_0 variation informed separation system can be used in combination with a predominant melody estimation algorithm to extract singing voice from music.

In a first step, the "Melodia" algorithm [243] is used to obtain an estimate of the predominant melody from the mixture. The mixture is then time warped based on the fundamental frequency of the melody so that it's predominant solo part is nearly constant in F_0. The extraction is then carried out in the time domain using efficient comb filtering.

5.2.1 Predominant Melody Estimation

The first step to extend the F_0 variation informed separation system is to obtain a warp contour that follows the predominant melody by means of extraction from the mixture (blind) or by human annotation (informed). In the following, we want to focus on how to obtain such a warp contour using a predominant melody algorithm.

Estimating the fundamental frequency of one single source from a mixture of several sources is considered a very difficult task [148].

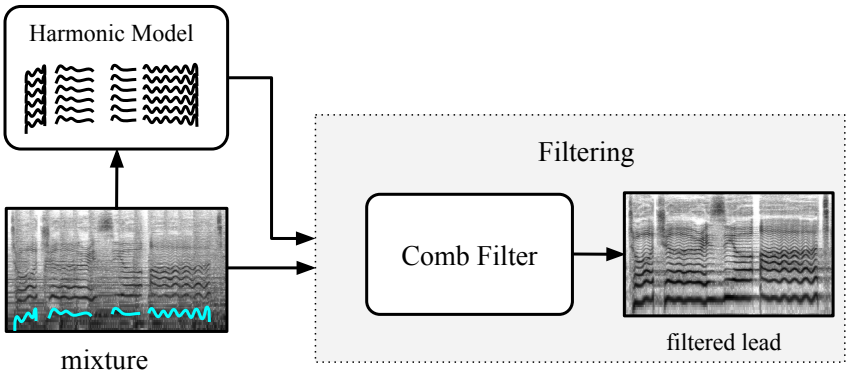


Figure 5.3: Block scheme diagram of a *harmonic assumption* for vocals. In a first analysis step, the fundamental frequency of the lead signal is extracted. From this, a separation is obtained by filtering the mixture.

However, in the case of vocal and accompaniment separation, we only consider one single source as the lead source - usually the vocals. This assumption holds true for many pieces in modern, popular music where usually the predominant voice is mixed slightly louder than accompaniment.

Now, extracting the predominant melody is an ongoing field of research named "melody estimation". However, compared to pitch or fundamental frequency, the term "melody" is only loosely defined. A widely used definition is the one from Poliner et al. in [212]:

> "...melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the 'essence' of that music when heard in comparison."

For a more comprehensive overview of melody extraction methods, the reader is referred to [242]. In turn, we used the *Melodia* algorithm, published by Salamon et al. in [243], as the basis to extract the predominant melody from the mixture.

Melodia consists of four parts: **1)**: a time-frequency transformation is applied and spectral peaks are extracted. **2)**: these form the basis of a *saliency* spectrogram that is computed using a weighted sum over all frequencies. This allows to emphasize the predominant/salient frequencies in the signal and is the core part of the *Melodia* algorithm. **3)**: from the saliency map, again, peaks are extracted and then connected to a melody line. This already is a good starting point for the melody estimate but usually contains many false positives due to the noisiness of the saliency representation. **4)**: The melody line is post-processed using a Viterbi algorithm. The purpose of it is to filter the contour by removing outliers, octave jumps and to improve the smoothness of the contour using a number of heuristics. Usually, this step is sensitive

to the overall length of the processed mixture and often, this step is computed in a semantically meaningful segment of the mixture like a full track or a refrain.

We applied *Melodia* using the implementation in *Essentia* [28] with the default parameters (sample rate 22 050 Hz, hop size 3 ms and window size 46 ms).

5.2.2 Source Extraction with Time Warping

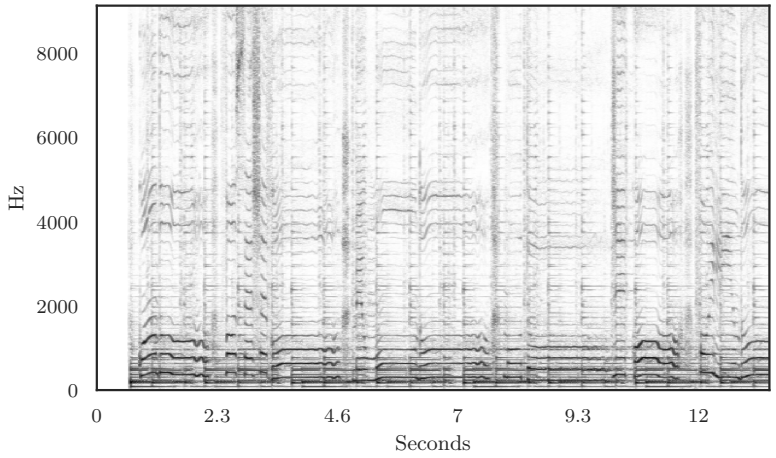
Once the predominant melody is obtained, the warp contour can be computed in the same way as described in Section 5.1.1 above. In the unison scenario, however, we were globally warping the signal using a continuous warp contour. In the case of full-length tracks, many parts are unvoiced and applying time warping on these segments would degrade the separation quality. Therefore, we only applied the warping on voiced parts and left the non-vocal parts unaltered. In order to do this, we used the built-in voice activity detection from *Melodia*. The full procedure is depicted in Figure 5.4 and Figure 5.5. For all continuously voiced segments, from the mixture (a), we compute the warp contour (c) from the melody segments (b). To reduce the complexity of the extraction, compared to the NMF mentioned in Section 5.1, we designed a comb filter that can extract the voice (f) in the warped time domain (d). Therefore, we used an IIR Filter with the frequency response

$$H(z) = \frac{1}{1 - 0.75z^{-P}}, \tag{5.2}$$

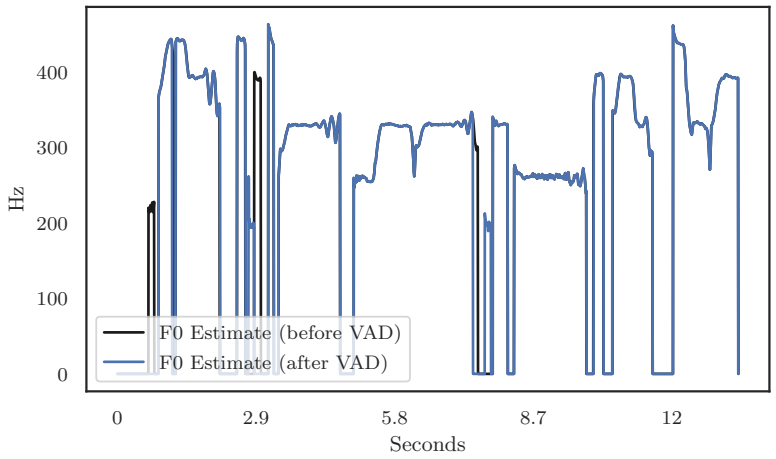
where P refers to the constant (due to the time warping) pitch period in samples. In order to then extract the vocals, zero-phase filtering is applied. The extracted vocals were inverse warped to linear time (e) and the accompaniment signal is created by subtracting the estimated vocals from the mixture signals. Each excerpt is then linearly crossfaded into the unaltered, accompaniment/mixture using a 10ms window. To further reduce the complexity of the separation system, instead one comb filter for each excerpt, we modified the warping algorithm so that a user-defined target pitch, rounded to an integer, is used. Finally, the full signal is inverse warped and resampled to the same pitch, which then only requires a single comb filter to extract the signal. A stereo signal is produced by filtering both channels individually.

5.2.3 Results in SiSEC 2015

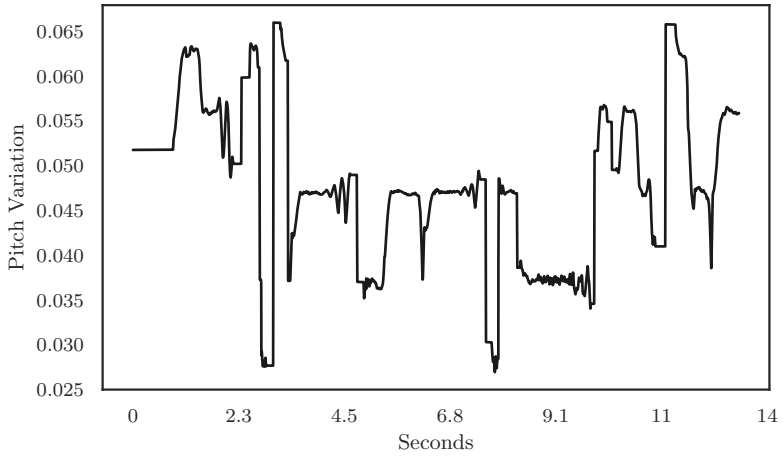
The algorithm has been applied to the Mixing Secret Dataset (MSD100) dataset, consisting of a total of 100 songs of different styles. The separation results were evaluated using BSSeval [36] and submitted



(a) Mixture Magnitude STFT

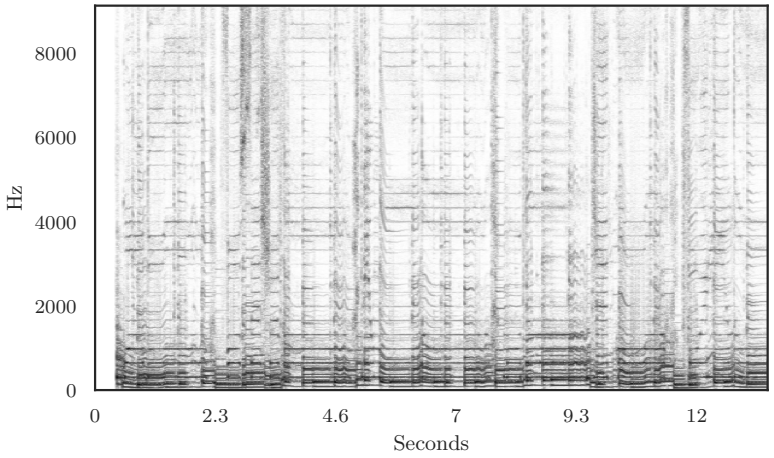


(b) Melody Estimate

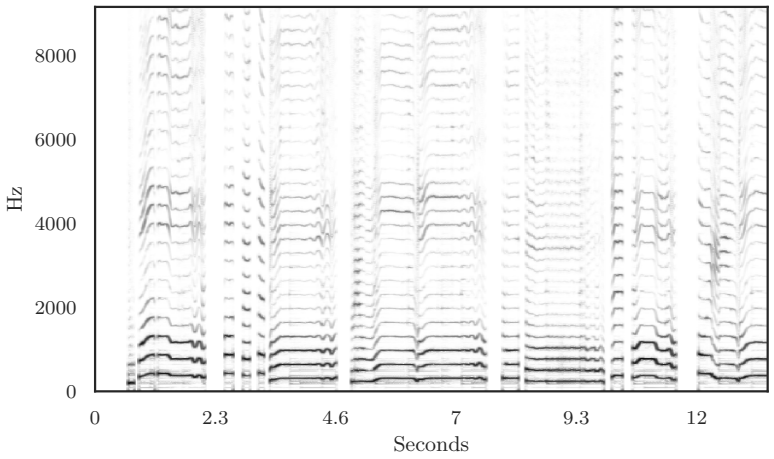


(c) Warp Contour

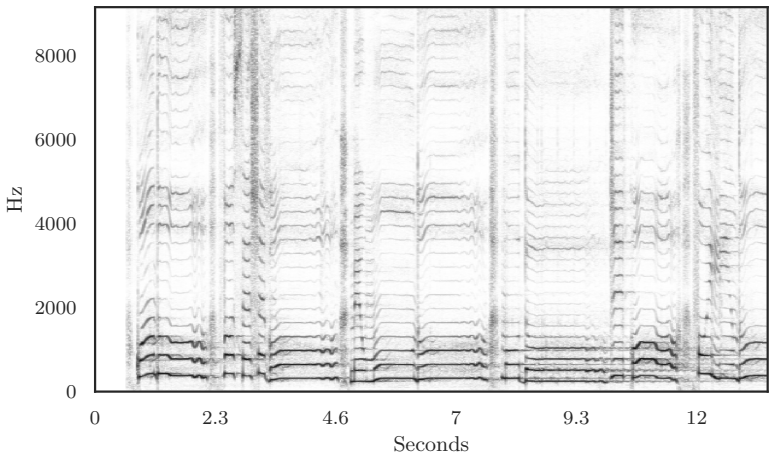
Figure 5.4: First steps to for a F_0 variation informed separation of the audio track *Tamy - Que Pena Tanto Fa* from the (MASS) dataset [304]. (a) depicts the input signal, (b) shows the estimate the MELODIA algorithm [243] and (c) the computed warp contour.



(d) Warped Mixture



(e) Vocal Estimate



(f) Ground Truth Vocals

Figure 5.5: Next steps of a F_0 variation informed separation of the audio track *Tamy - Que Pena Tanto Fa* from the (MASS) dataset [304]. (d) shows the mixtures, warped by (c) from Figure 5.4, (e) the extracted vocal signal after comb filtering and inverse warping. For comparison, (e) shows the original vocal reference.

to the SiSEC 2015 challenge [199]. The system was ranked in the last third of the participants and scored only slightly better than RPCA based methods [126]. The reason for this is that our proposed system highly depends on the melody estimation algorithm which, in turn, is based on the assumption that there exists a predominant melody in the mixture. Unfortunately, the newly created MSD100 dataset was not mixed using professional mastering, resulting in vocals that are below average in loudness. Due to their small energy, they were not detected as voiced by *Melodia*, hence the warping was not applied. Also, in some cases the estimates were one octave off, producing severe artifacts due to extreme warping.

On the positive side, the proposed method is of very low complexity.

5.2.4 Improving Voiced/Unvoiced Detection using DNNs

As mentioned in the previous section, voice activity detection is of paramount importance in the proposed music separation system. Therefore, we decided to evaluate if the performance of the system can be improved by using a more robust voice activity detection method as a separate preprocessing step. Shortly after we submitted the separation results to the SiSEC 2015 evaluation campaign, the whole audio community was shaken up by the recent success of deep learning throughout several audio related tasks that go beyond automatic speech recognition. Among them are several tasks related to music information retrieval (MIR) such as singing voice detection which received major breakthroughs in 2014 and 2015 [157–159, 253].

Therefore, we decided to integrate a state-of-the-art singing voice detection system into the separation pipeline and evaluate the end-to-end performance. We chose to reimplement the system by Leglaive [157], since it was a good compromise between complexity (its use of handcrafted features instead of large STFT frames) and performance. In fact, the system reached a state-of-the-art accuracy of 91.5% for classifying frames of singing voice for the annotated *Jamendo* singing voice detection dataset [225], which is an improvement of more than 10% compared to the best performing non-DNN system.

The input of [157] is an 80-dimensional feature vector consisting of harmonically and percussively enhanced STFT frames of the mixture as described in [198]. The output of the network is a frame-wise integer, indicating the presence of *voiced* or *unvoiced* frames. We trained the network using a fixed number of frames from the DSD100 training dataset. The vocal activity labels were obtained from the dataset by analyzing the true vocals. The network was created and trained using the Keras framework [53]. We stacked up to three layers using the parameters as mentioned in [157] but used unidirectional LSTMs instead of bi-directional to reduce computational complexity. The trained network achieves an accuracy of 85% on the test set. While

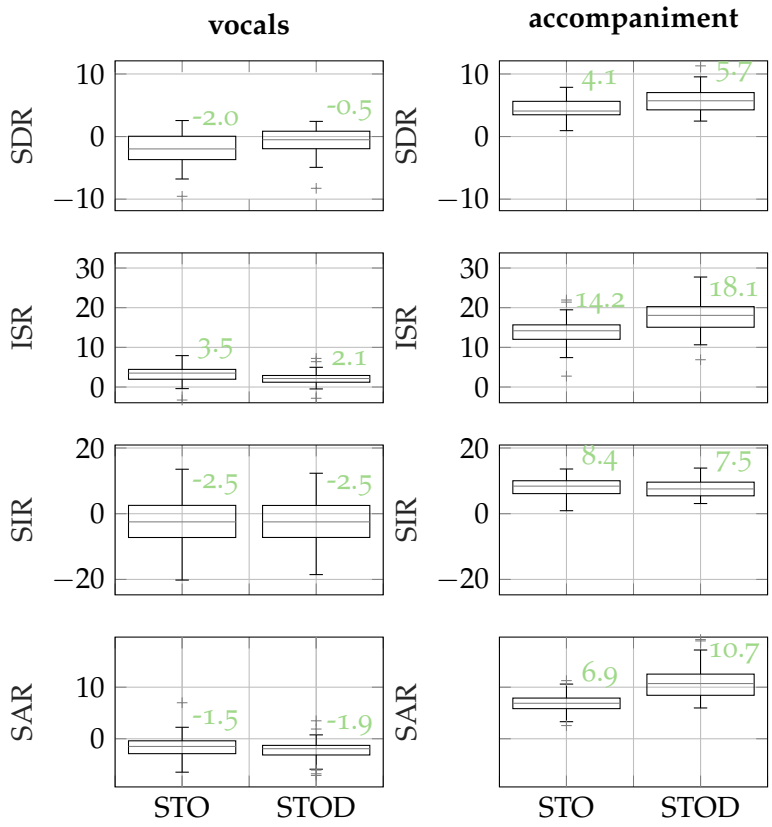


Figure 5.6: BSS Eval scores for the vocals and accompaniment estimates on the DSD100 dataset as used in [166]. Results are shown on the *test* set only. Results indicate the improvements of the DNN based vocal activity detection system (STOD) in comparison to the baseline system (STO) that was submitted to [199].

the original network proposed in [157] was framed as a classification task, we modified the sigmoid output activation and replaced it with a linear activation, then trained using the mean squared error cost function. The output is then multiplied with the mixture signal so that segments with less energy are reduced in volume instead of a boolean decision. In experiments, we found out that this helps the *Melodia* algorithm to better detect vocal activity throughout an audio track and therefore yields in melody estimates with fewer errors. This DNN-optimized version of the algorithm was then compared (but not submitted) to the new the SiSEC 2016 dataset which was released in the meantime [166]. The results, depicted in Figure 5.6 show that the DNN vocal activity detection improved the vocals SDR by 1.5 dB which is considered as a significant improvement.

5.3 IMPROVING F_0 ESTIMATION USING TIME WARPING

Parts of this section was previously published in [277] and was revised for this thesis.

An informed system like the one we described in the previous section often has limitations due to the fact that the provided fundamental frequency variation estimate might not be accurate enough and is subject to an upper limit. Further, it can be assumed that such an upper bound is especially relevant for a warping based system that relies on an instantaneous estimation of the fundamental frequency. While we developed the framework of a fundamental frequency (F_0) variation informed separation, as presented in Section 5.1 of this chapter, we found that we can utilize this to also optimize fundamental frequency estimators themselves.

An estimate of the fundamental frequency of a signal is required in applications of audio and speech signal processing. Some scenarios are targeted to extract the fundamental frequency of the predominant source [243] in a mixture of other sources. In other applications, algorithms are used to extract fundamental frequencies of multiple sources simultaneously present in a signal [147]. However, the most common scenario in many works is to extract the fundamental frequency of a monophonic and harmonic audio signal containing speech or music [27, 48, 55, 229, 290, 291].

Algorithms for estimating the F_0 of a signal vary in stability and accuracy. In turn, we proposed a method which iteratively improves the estimates of such algorithms by applying in each step, a time warp on the input signal based on the previously estimated fundamental frequency.

PROPOSED SYSTEM The development of novel methods for F_0 estimation, performing as well as earlier methods, such as the popular correlation based YIN algorithm [48], has proven challenging. In a study [13] it is stated that YIN still performs best in terms of accuracy. Nevertheless, when using YIN or other block-based algorithms, a frame length and a hop size have to be selected trading temporal resolution on one side against frequency accuracy and robustness on the other side.

Especially when the signal is polyphonic, the robustness is the most crucial aspect of a pitch estimator. In work from Mauch et al. [181], the robustness of the YIN algorithm is improved by probabilistic post-processing. However, besides robustness, there is a variety of use cases requiring high accuracy as well as high temporal resolution. Application in parametric audio coding [216] requires the parameterization of pitch bends and vibratos. Furthermore, source separation algorithms aiming at the extraction of harmonic sources from the mixture can make use of an instantaneous F_0 estimate [278, 307]. There are already contributions addressing the improvement of accuracy of F_0 estimates

such as [185] which introduced a non-integer similarity model or [55] which belongs to the group of parametric pitch estimators.

We propose to improve the output of already existing algorithms in terms of temporal resolution as well as accuracy by iterative time warping. Two other contributions already make use of time warping in the context of pitch estimation. Resch et al. [229] proposed an instantaneous pitch estimation technique which optimizes a warping function that would lead to a constant pitch signal. Their optimization framework minimizes a cost function specifically targeted for speech signals. Azarov et al. have introduced an improved version of RAPT (called iRAPT1 and iRAPT2) [12]. Our main contribution is a time warping based refinement method that is applicable to any F_0 estimate. Our method emphasizes the strengths of different estimators and thus can even help to improve their robustness.

Depending on the algorithm and application, there are several reasons why F_0 estimators deliver a less than ideal performance. When the signal tested is not tonal — like in unvoiced parts of speech — a proper estimation is impossible. If the estimator is optimized on purely harmonic signals, inharmonicity or frequency jitter of the input signal will increase the estimation error. Many of these reasons will lead to errors on the coarse level of the estimate (like octave jumps). The fine level accuracy is mostly influenced by parameters like time and/or frequency resolution of the estimator. A signal containing rapid changes of the frequency or modulations like "vibrato" is, therefore, more affected regarding fine level error. To obtain a more accurate estimate, we propose to time warp the signal by using the coarse level estimate towards a more constant pitch. The underlying assumption here is that pitch estimators generally perform better the more constant the pitch is.

Initial F_0 estimate

The first step is to calculate an initial F_0 estimate by using an existing pitch estimator. Note that we later require the estimate to be defined for every input sample, thus $F[n]$ may require interpolation. In our pipeline, we use linear interpolation for all estimators. F_0 estimators, like YIN [48], also provide a measure of confidence $c[n]$.

In our application, the warp map $w(t)$ is constructed in such a way that the instantaneous changes in frequency of the signal in the linear time domain are minimized in the warped time domain. For this, we derive the map from an estimate of the fundamental frequency F_0 using Equation 5.1 from Section 5.1.

In the scope of this work, the warping is applied globally over the full length of the signals under consideration. Here, in comparison to Section 5.1, we also consider an optional confidence measure $c[n]$ which can be incorporated for a processed version of the warping contour. This ensures that the warp contour has no discontinuities

that result in additional artifacts after re-sampling. If the estimator does not provide such a measure, a separate voiced/unvoiced detection algorithm can be used. To obtain a warp contour $f[n]$ from an F_0 estimate we propose the following steps: **(A)** initialize the warp contour with F_0 estimate $f = F$, **(B)** find contour segments with high confidence, i.e. $c[n]$ exceeds a given threshold, **(C)** linearly connect the high confidence contour segments and **(D)** set start and end of warp contour to a constant value if confidence is below threshold. That way warping according to F_0 is applied in the regions of high confidence without significantly affecting the gaps in-between.

To improve the accuracy of the F_0 estimate, time warping is applied to the input signal $x[n]$ based on W. The input signal is 128-times oversampled using sinc based interpolation filters. From $\check{x}[n]$ a new F_0 estimate $\check{F}_1[\nu]$ is being calculated as in step **(A)**[2]. The first step therefore is similar to [229]. Additionally, a warped confidence measure $\check{c}_1[\nu]$ can be used to convert $\check{F}_1[\nu]$ into a warped *warp contour* $\check{W}_1[\nu]$. It is possible to linearly add $\check{F}_1[\nu]$ to the first estimate for refinement, as it is done in [12]. However for linear sweeps, the warped estimate is shifted in time. Thus an error is introduced which is even more distinct if the first F_0 estimate is error prone. We therefore propose a method to reduce this error:

- Inverse time warping is applied to $\check{F}_1[\nu]$ based on the original warp contour W resulting in $F_1[n]$.
- In the case of a perfect F_0 estimate, the signal warped with the resulting contour would have a constant F_0 equal to the mean $\bar{W}$. Therefore, a refined F_0 estimate after one iteration is then calculated by $F_1^r[n] = F_1[n] \cdot W[n]/\bar{W}$ assuming that the warp contour is initialized as in step **(A)** above.
- The refinement can be repeated k times to obtain a better estimate. To avoid accumulating errors introduced by the re-sampling based warping, more iterations benefit from calculating a refined warp contour/warp map instead of doing a nested warping on the input signal. The map is obtained by inverse time warping of the warp contour $\check{W}_1[\nu]$ resulting in $W_1[n]$. A refined warp contour $W_1^r[n]$ is then obtained in the same way as the refined F_0 estimate is calculated. For the calculation of the kth step, time warping is based on the $W_{k-1}^r[n]$ refined warp contour.

An example of the proposed refinement is depicted in Figure 5.7. The final refined estimate is closer to the reference than the F_0 estimator without refinement. It also shows (right plot) how much "flatter" the F_0 contour becomes after each iteration. Note that compared to [229], our method does not use a complex optimization scheme but relies on the performance of the pitch estimator in successive iterations. Hence

2 Note, that ˇ indicates *warped* time instead of *linear* time.

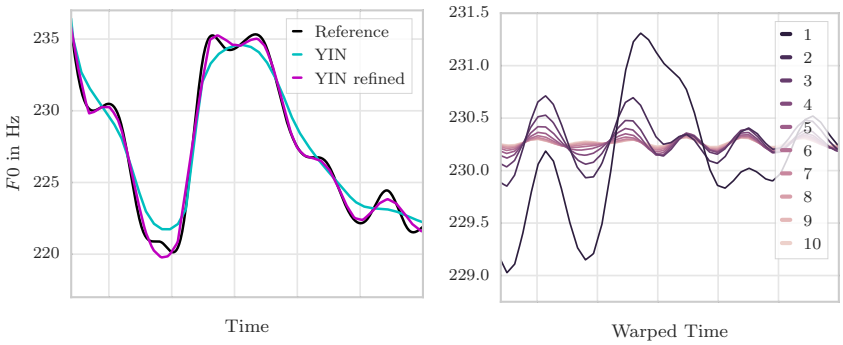


Figure 5.7: F_0 refinement for one excerpt of synthesized speech using **YIN** [48] with 10 iterations. *Left*: Estimated F_0 in linear time. *Right*: estimates after each warping iteration in warped time.

our "black box" like post processing simplifies the procedure such that it can be applied to any pitch estimator. That way the selection of a pitch estimator which best fits to the signal type can be seen as an optimization.

5.3.1 Experiments and Evaluation

For the evaluation of the proposed F_0 refinement, we test the refinement algorithm with the following F_0 estimators:
YIN [48] is used as an FFT based implementation [28]. The confidence measure is thresholded for values lower than 0.6 on the speech recordings. **iRAPT1,2** [12] are improved versions of the RAPT framework. We use the author's MATLAB implementation of the iRAPT1 and iRAPT2 algorithms. iRAPT2 is a refinement method that is comparable to our proposed method. To evaluate the results, we apply our refinement to iRAPT1 and compare it with the refinement produced by iRAPT2. $c[n] < 0.7$ is used for thresholding speech recordings. **MELODIA** [243] is not designed to be an F_0 estimator but is able to extract the *predominant* melody in a polyphonic mixture. We increase the bin resolution to 0.5 semitones, to increase the accuracy. We used the Essentia implementation. For thresholding, we use the built-in voiced/unvoiced detection. For YIN and MELODIA, we evaluate on a frame length of 64 ms and a hop size of 16 ms. For iRAPT1 and iRAPT2 we use the fixed frame length parameters of the author's implementation.

We use the established evaluation measures Gross Pitch Error (GPE) and Mean Fine Pitch Error (MFPE) [12]. We focus on MFPE in our results, measuring the absolute deviation between the reference and the estimated F_0 per sample. As mentioned in [229], evaluating the accuracy of F_0 estimates is challenging because of the lack of ground truth datasets annotated on a time scale with such a high

resolution. Most of the available audio test datasets are not suitable because the F_0 annotation is only available with low time resolutions. By using such a dataset there is a risk that the refined F_0 estimate is higher in MFPE. This is because the refined estimates show more of the fine structure deviating from the coarse annotation which then is considered as piecewise constant. To address this issue, we first present the evaluation results on synthetic data. To verify our synthetic results, we present the results of speech data annotated on 10 ms frames derived from laryngograph signals. We did only evaluate and process the voiced parts of the signals as indicated in the provided annotation labels. Also note that since we focus on the MFPE, all segments where one of the estimators results in a GPE > 0 are excluded from the results, hence the GPE for all of our results is 0. The proposed refinement has been processed with one iteration ($k = 1$). Experiments showed that more iterations only marginally improve the results.

Since the proposed refinement algorithm repeatedly applies pitch estimation, the performance of these estimators on the time-warped (nearly constant) signal is of interest. Therefore, we included the results of an oracle refinement where the first estimate is set to a ground truth pitch. Additionally, this also does reveal information about the quality of the ground truth annotation itself.

Synthetic Data

To generate synthetic test data we use pitch label annotations of the PTDB-TUG speech dataset [209]. We synthesize the melody or voice using a simple sinusoidal signal model. To get accurate ground truth data, the pitch annotations were up-sampled to audio rate by using linear interpolation. Similar to [181], we then synthesized the data using cosine based oscillators adding 10 harmonics to each signal output. The test set has been rendered at 16 kHz. The complete PTDB-TUG set results in almost 10 hours of input signal data. We present the results of the synthetic data as box plots in Figure 5.8 grouped by the estimator. It shows that all estimators benefit from the refinement in terms of MFPE. The iRAPT1 estimator shows the best improvement of 68% in MFPE. As expected, oracle refinement yields almost perfect results in terms of MFPE.

Speech Data

For the results of the algorithm on real data we first used the same PTDB-TUG items as in the synthetic data but processed the accompanying speech recordings. The MFPE values were then calculated by averaging the sample-wise F_0 estimates from our proposed method over frame lengths of 10 ms to match the annotation data. The results are shown in Figure 5.9. The mean values indicate that the MELODIA algorithm performs best overall. We can see that the refinement does not show a clear effect on the iRAPT estimator. The oracle re-

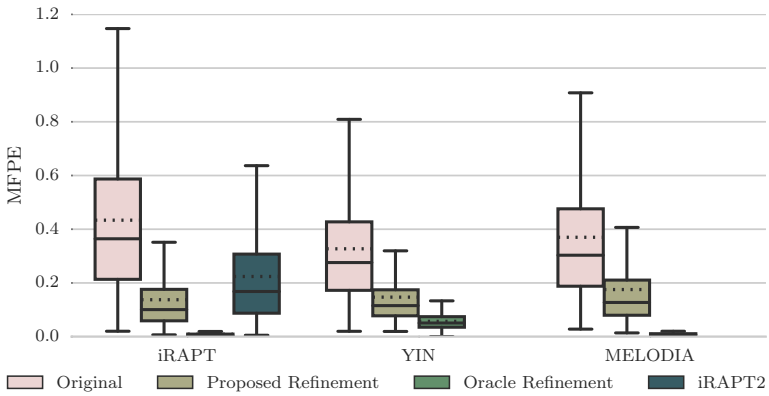


Figure 5.8: Results from the synthesized PTDB-TUG dataset. MFPE grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

finement results indicate that even if a ground truth is known, the refinement based on the warped (constant) signal cannot get much lower in MFPE. As also seen on synthetic data, iRAPT2 does not show significant improvements compared to our proposed refinements.

Polyphonic Mixtures

Pitch estimation of polyphonic mixture input signals, in general, is known to be more difficult than on monophonic signals. To show that our proposed refinement is not bound to the optimization on specific signals we processed the MedleyDB [24] which consists of 108 professionally recorded music mixes where the main melody has been annotated by humans. We only evaluate the MELODIA [242] estimator in this scenario. Frame lengths and hop sizes were increased to 92 ms and 23 ms, respectively. The set is processed at 44.1 kHz. To further back up the results of the fine pitch error in this scenario, we additionally evaluated the results of a correlation-based measure as introduced in [229] (See Equation (19)). Instead of computing the correlation coefficients on the mixture, we used the accompanying multi-tracks. The track which most predominantly contributed to the main melody has been chosen for the correlation coefficient measure. The results of the experiment are shown in Figure 5.10.

5.4 SUMMARY AND DISCUSSION

In this chapter, we highlighted the time-varying aspects of musical sources such as vibrato to be utilized for the application of source separation. To address this task, we developed a method that utilizes time warping to extract a source from the mixture. More specifically, the mixture is warped, based on the fundamental frequency estimate

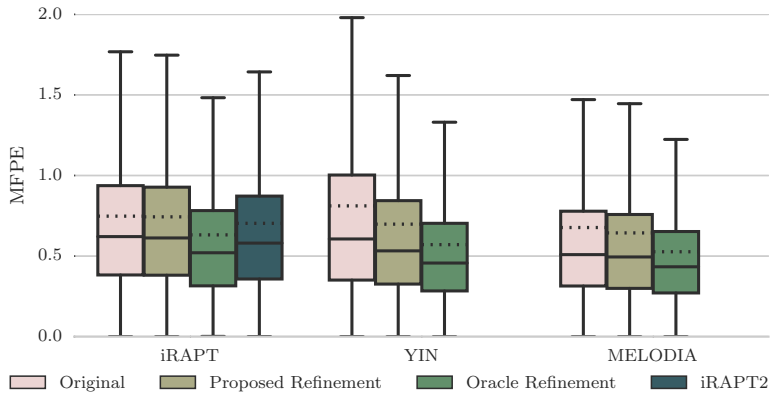


Figure 5.9: Results from the real recordings PTDB-TUG dataset. MFPE grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

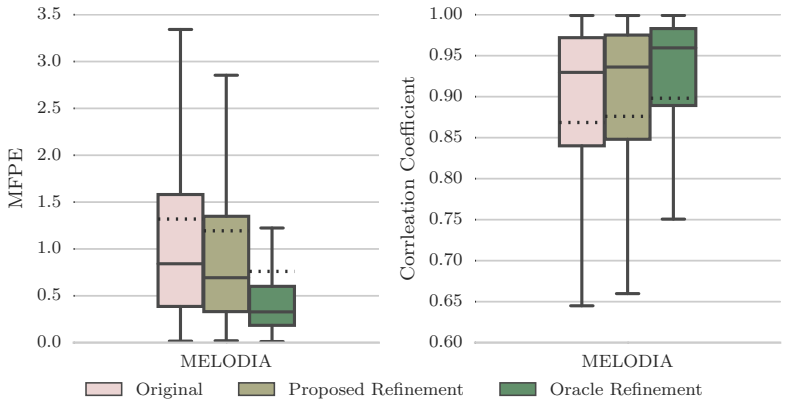


Figure 5.10: Results from the real recordings MedleyDB dataset. MFPE and Correlation Coefficient grouped by estimator. Solid/dotted lines represent medians/means. Outliers are not shown.

of the source to be extracted. In the warped time domain the frequency modulation of the desired source is removed. In our first study, we evaluated this method of separating single note from instruments playing in unison. For the actual separation, we used a "standard" NMF separation approach. The results of 45 mixtures have been evaluated by using the PEASS toolbox and the scores indicated an improvement in favor of the F_0 variation informed NMF compared to the "standard" NMF.

In order to evaluate if our method can be applied on a more realistic scenario, we proposed an extension of the method for the scenario of vocal and accompaniment separation. We used a state-of-the-art melody estimation technique to extract the F_0 variation of the vocal source to apply warping to the mixture. We performed separation in the time domain using a comb filter to further reduce artifacts. The

method relies on a robust and accurate estimate of the fundamental frequency as well as vocal activity estimate.

To address the latter, the method was extended to include a deep neural network based vocal activity detector. This helped to exclude non-vocal parts from the warping and in turn, improved the vocal separation performance by 1.5 dB SDR.

In order to improve the accuracy of F_0 estimates, we proposed a method, based on the same time warping principle, in the last section of this chapter. The proposed method applies time warping iteratively based on an initial F_0 estimate, assuming that more iterations remove more variation, thus supports the F_0 estimation process in the next iteration. This idea can be applied to any F_0 estimator as a post-processing step. Future work could include an optimization criterion to control the number of iterations, however, we have to emphasize that improvements in accuracy are difficult to evaluate on real datasets [275].

To conclude, time warping based separation can work well on some signals but requires further handcrafted tuning to yield good results. In the next chapter, we want to investigate if separation can still benefit from spectro-temporal modulations if they are not known or estimated a priori.

6 SEPARATION BY UNKNOWN MODULATION

In the previous chapter, methods were proposed to separate highly overlapped signals, informed by an estimate of fundamental frequency (F_0) variation of the source to be separated. Even though we showed that F_0 variation could be estimated from the mixture, often, it is not easy to obtain in practice. In contrast, in this chapter, we will introduce separation methods which do not incorporate prior knowledge about the modulation. In fact, some of the methods operate blindly, meaning they do not require any further information about the sources except for the number of sources to be separated.

In blind source separation research, many contributions were based on NMF [153, 154]. NMF quickly became one of the main scientific frameworks in the field of audio source separation with a large number of contributions. The popularity of NMF algorithms can be explained by the intuitive way in which they work on (non-negative) time-frequency representations of the mixture signal. Let us consider the magnitude STFT $\boldsymbol{X} \in \mathbb{R}_+^{F \times T}$ with F being the number of frequency bins and T the number of time frames. Now, the NMF incorporates non-negative constraints to perform the separation into the sum of K latent components which are all factored into two matrices (referred to as frequency *basis* $\boldsymbol{W} \in \mathbb{R}_+^{F \times K}$ and temporal *activations* $\boldsymbol{H} \in \mathbb{R}_+^{T \times K}$):

$$\mathbf{X} \approx \boldsymbol{W}\boldsymbol{H}^T = \sum_{k=1}^{K} \boldsymbol{w}_k \circ \boldsymbol{h}_k \tag{6.1}$$

As it can also be seen in Figure 6.1, the factorization can also be written as the sum of K outer products between two rank-one matrices $\boldsymbol{w}_k \in \mathbb{R}^F$ and $\boldsymbol{h}_k \in \mathbb{R}^T$.

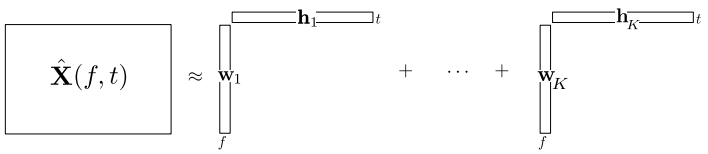


Figure 6.1: Visualization of the product of two rank-one matrices as being used in NMF.

The NMF provides a rank reduction which allows decomposing mixtures into K source components. At the same time, the factorization inherently follows specifics of music, observable in time-frequency

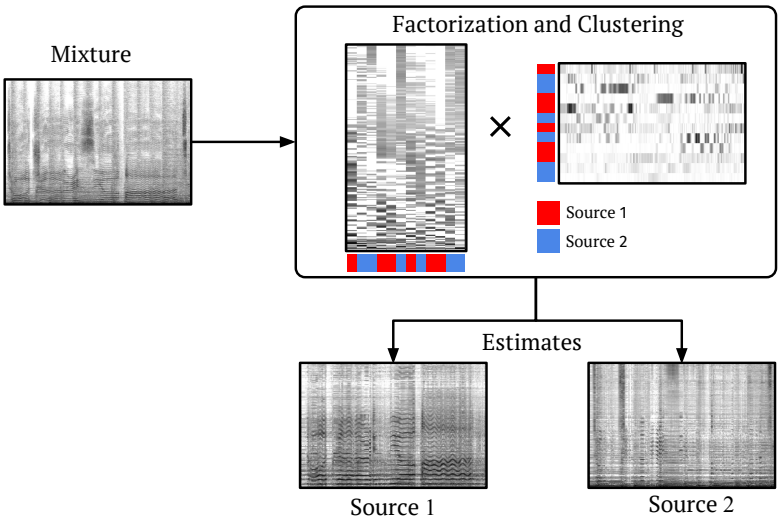


Figure 6.2: Example of spectrogram based source separation using NMF.

representations: the fact that harmonic sources can be described using a pitch/tone (represented in $\boldsymbol{W}$) and its duration (represented in $\boldsymbol{H}$). It is this property that also allowed to use NMF for the purpose of transcriptions [267].

To obtain the factorization, an optimization problem needs to be solved, resulting in a non-unique solution. Each factorization is calculated by minimizing the error between $\boldsymbol{X}$ and $\boldsymbol{W}\boldsymbol{H}^T$ with respect to some cost function

$$\min_{\boldsymbol{W},\boldsymbol{H}^T} \mathrm{D}(\boldsymbol{X}|\boldsymbol{W}\boldsymbol{H}) \text{ subject to } \boldsymbol{W} \geq 0, \mathbf{H} \geq 0. \tag{6.2}$$

In most source separation methods the beta-divergence cost function

$$\mathrm{D}_\beta(x|y) = \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right) \quad \beta \in \mathbb{R}\backslash\{0,1\} \tag{6.3}$$

is being used [85]. For the special cases of $\beta = 0$ and $\beta = 1$, D correspond to the Itakura-Saito (IS) and Kullback-Leibler (KL) divergence and the euclidean distance equals to $\beta = 2$. To efficiently compute the optimization, an algorithm was proposed in [154] which makes use of simple to use multiplicative update rules, derived from the cost function. For further details, we refer to [56].

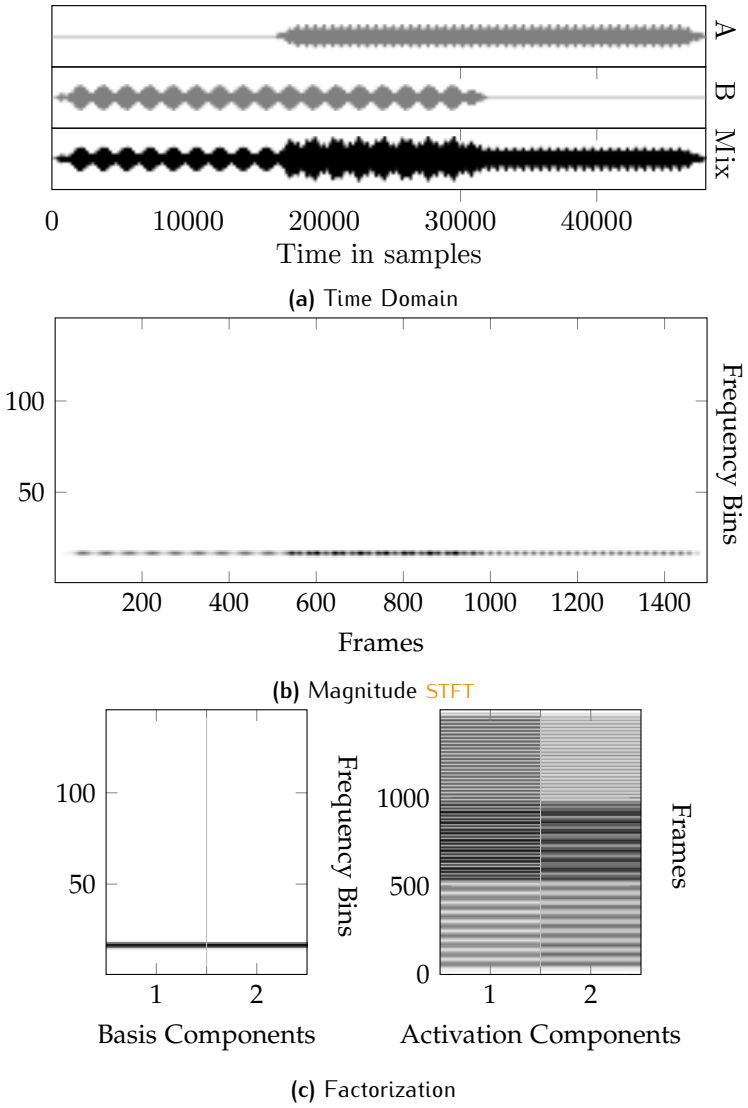
After factorizing $\boldsymbol{X}$ into K components, one can obtain K magnitude spectra. However, K is usually selected to be larger or equal to the total number of sources. When it is larger than the number of sources, the components can be clustered into the number of desired sources. Often this can be achieved by some similarity metric that allows to calculate pairwise distances between the K components and the desired sources

and apply k-means clustering algorithm [268]. A separation example is depicted in Figure 6.2. Here the number of components K is six per source, resulting in a total number of $K = 12$. Each component is clustered into one of two sources to generate two estimates.

It is this step that transforms the NMF into a supervised algorithm. As for other separation methods, performed in the time-frequency domain, separation is achieved using Wiener filters or ratio masks to extract the sources from the mixture [169].

Since NMF first appeared to the source separation community in [267] and in [298], a large variety of NMF "flavors" were introduced to improve certain aspects of the NMF in the application of music. In [302, Chapter 16], the authors describe one of the main problems of NMF which is that "standard NMF is shown to be effective when the notes of the analyzed music signal are nearly stationary". As we described in Chapter 3, this is especially problematic for pitches that incorporate vibrato. Here, NMF-based processing suffers from its simplified model and its magnitude STFT representation makes it harder to model these time-varying sources.

To underpin this issue, we depict this problem in Figure 6.3 which shows the factorization of a simple amplitude modulated input signal. The signal consists of two sinusoids which are linearly mixed. Both share the same carrier frequency but have different amplitude modulation rates. When we apply a factorization with $K = 2$, one can see that NMF has difficulties to separate the two signals sufficiently and instead activates both sources in an alternating pattern. One way towards better separation is to increase the number of components per source, however, this introduces difficulties in the clustering. Another method is proposed in [86, 133, 233, 264] which use convolutions to model shifts in components. This leads to factorizations that are able to also model vibrato events. However as stated in [112], it does "not permit any variation between different occurrences of the same event (atom), its duration and spectral content evolution being fixed". Instead, they proposed a frequency-dependent activation matrices by using a source/filter-based model. The model is based on an Auto-Regressive Moving Average (ARMA) time-varying model that allows single spectral components to be modeled along with their spectral variations. The model, as reported by the authors, however, does only allow for small frequency variations and fitting the ARMA model is a time-consuming process.



(a) Time Domain

(b) Magnitude STFT

(c) Factorization

Figure 6.3: Example of separating a mixture of two amplitude modulated sinusoids using NMF
(a) Mixture of two sinusoids at 440 Hz with AM of 4.7 Hz and 12.6 Hz, **(b)** STFT ($FFTlength = 256$), **(c)** Non-negative matrix factorization results in $\boldsymbol{W}$ and $\boldsymbol{H}$, after 100 iterations ($\beta = 1$).

6.1 TENSOR FACTORIZATIONS FOR MODULATION SPECTROGRAMS

Parts of this subsection is also based on the work published in [270].

Another way to improve separation of modulated sources is the use of higher-dimensional tensor representations as a signal representation as introduced in Section 3.2.1. A variety of models exist to factorize a tensor into three components and apply a similar rank reduction as in the NMF case. Tensor factorizations are useful for applications of data with more than two dimensions. In audio separation, tensor factorization was originally proposed to address multichannel sepa-

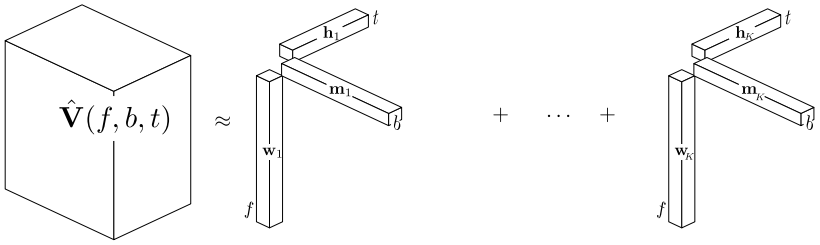


Figure 6.4: PARAFAC decomposition in an example for a three-dimensional tensor $\boldsymbol{V}$ into the sum of K outer products of three rank-one matrices.

ration as in [83, 84, 201]. Barker and Virtanen [21] were the first to propose modulation tensor representations for single-channel source separation.

As proposed by [21], the non-negative tensor factorization approximates a modulation tensor $\boldsymbol{V}_{f,b,t}$ by a product of three matrices containing the frequency/basis $\boldsymbol{W}$, time/activation $\boldsymbol{H}$ signals, and the modulation gain for each component $\boldsymbol{M}$. The product of this factorization is generally referred to as *PARAFAC* product[1].

In the vein of the NMF factorization mentioned in Equation 6.1, a 3-way non-negative tensor factorization (NTF) can simply be extended to:

$$\mathbf{V} \approx \sum_{k=1}^{K} \mathbf{w}_k(f) \circ \mathbf{m}_k(b) \circ \mathbf{h}_k(t). \tag{6.4}$$

This notation is commonly used in many tensor factorization applications [151]. However, we found that it is easier to follow when the individual tensor elements are used, which is also the recommended notation proposed in [142], for $f = 1, \ldots, F; b = 1, \ldots, B; t = 1, \ldots, T$:

$$v_{fbt} \approx \sum_{k=1}^{K} w_{fk} m_{bk} h_{tk}. \tag{6.5}$$

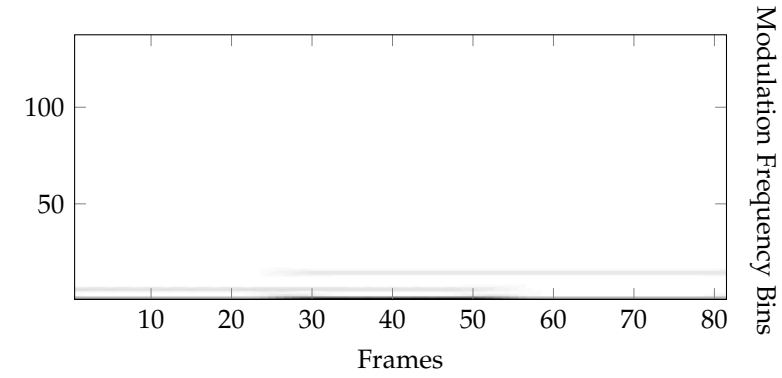
A visualization of the three-way PARAFAC product is depicted in Figure 6.4.

Compared to Barker and Virtanen in [21], we chose to generate the modulation tensor in a way that is simpler and easier to invert. They used a Gammatone filter bank and rectification to model the characteristics of the human auditory system. We simplified the processing and used a two-stage DFT filter bank where the modulation domain is based on magnitude STFT. Although this can give perceptually less optimal results, each step can be directly inverted by using the complex representation. Barker already showed that the NTF based approach gives good results on speech signals compared to the "standard" NMF.

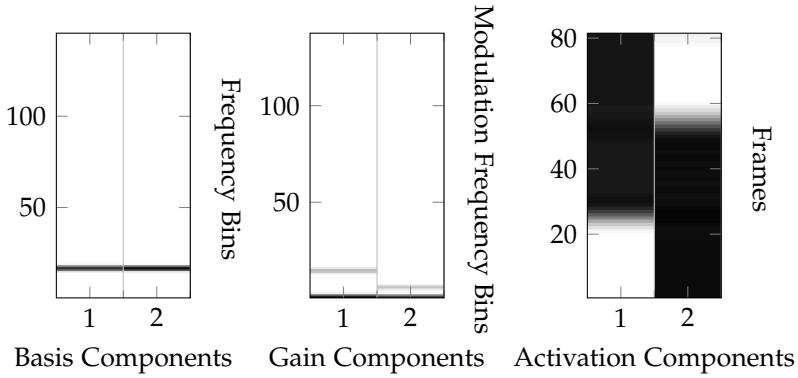
1 Also known as "Polyadic form of a tensor", PARAFAC (parallel factors), CANDECOMP or CAND (canonical decomposition) or CP (CANDECOMP/PARAFAC) [151].

Motivated by these results, we wondered if the modulation NTF can be used to separate two instrument mixtures by their amplitude modulation characteristics, as it is the case in the unison scenario.

Thus, let us return to the example from Figure 6.3 of two harmonic signals having the same fundamental frequency of 440 Hz, with a stationary amplitude of 4 Hz and 10 Hz respectively. These differences now turn out to be latent in a non-negative modulation tensor representation. In contrast to NMF, Figure 6.5 shows valid factorizations of a unison signal using NTF. It gives a smoother activation matrix and is able to generate the output with the separated amplitude modulations on each sinusoid. The modulation frequency gain matrix shows the two modulation frequency templates and the DC-component.



(a) Slice of Modulation Tensor



(b) Tensor Factorization

Figure 6.5: (a) Modulation tensor slice of a mixture of two sinusoids at 440 Hz with AM of 4.7 Hz and 12.6 Hz ($FFTlength = 256$), **(b)** $\boldsymbol{V} \approx \boldsymbol{WMH}$ Result of Non-Negative Tensor Factorization ($\beta = 1$) after 100 iterations.

A comparison of the modulation tensor approach compared to the F_0 variation informed method on the unison separation scenario has been carried out in our work published in [270]. Results indicated that the modulation tensor factorization generally performs worse than informed methods. This is because it only considers amplitude modulations even though frequency modulations are the actual source

of the modulation. In the next section, we investigated how more complex modulation patterns can be utilized for separation.

6.2 COMMON FATE MODEL FOR UNISON MIXTURES

This section was previously published in [274] and was revised for this thesis with permission (®2016 IEEE).

In this work, a novel tensor signal representation is introduced which additionally exploits similarities in the frequency direction. We, therefore, make use of dependencies between modulations of neighboring bins. This is similar to the proposed high-resolution non-negative Matrix Factorization model that accounts for dependencies in the time-frequency plane (HR-NMF [16]). In short, HR-NMF models each complex entry of a time-frequency transform of an audio signal as a linear combination of its neighbors, enabling the modeling of damped sinusoids, along with an independent innovation. This model was generalized to multichannel mixtures in [18, 19] and was shown to provide considerably better oracle performance for source separation than alternative models in [175]. Indeed, even though some variational approximations were introduced in [17] to strongly reduce their complexity, those algorithms are often demanding for practical applications. In this work, we proposed to relax some assumptions of HR-NMF in the interest of simplifying the estimation procedure. The core idea is to divide the complex spectrogram into modulation patches in order to group common modulation in time and frequency direction. We call this the *Common Fate Model* (CFM), borrowing from the Gestalt theory, which describes how human perception merges objects that move together over time (from [32]):

> "the Gestalt psychologists discovered that when different parts of the perceptual field were changing in the same way at the same time, they tended to be grouped together and seen to be changing as a group because of their common fate."

Bregman introduced the Common Fate theory for auditory scene analysis as the ability to group sound objects based on their common motion over time, as occurs with frequency modulations of harmonic partials. As outlined by Bregman, the human ability to detect and group sound sources by small differences in FM and AM is outstanding. Also, it turns out, as mentioned in Section 3.2, that humans are especially sensitive to modulation frequencies around 5 Hz, which is the typical vibrato frequency that many musicians produce naturally.

6.2.1 The Common Fate Transform

Let $\tilde{x}$ denote a single channel audio signal. Its STFT is computed by splitting it into overlapping frames and then taking the discrete Fourier

transform (DFT) of each one. Since the waveform $\tilde{x}$ is real, the Fourier transform of each frame is Hermitian. In the following, we assume that the redundant information has been discarded to yield the STFT. The resulting information is gathered into an $N_\omega \times N_\tau$ matrix written $\boldsymbol{X}$, where N_ω is the number of frequency bands and N_τ the total number of frames. In this study, we will consider the properties of another object, built from $\boldsymbol{X}$, which we call the Common Fate Transform (CFT).

It is constructed as illustrated in Figure 6.6. We split the STFT $\boldsymbol{X}$ into overlapping rectangular $N_a \times N_b$ patches, regularly spaced over both time and frequency. For reference, later in this thesis, we will call this representation the *Grid STFT* (GFT). Then, the 2D-DFT of each patch is computed[2]. This yields an $N_a \times N_b \times N_f \times N_t$ tensor where N_f and N_t are the vertical and horizontal positions for the patches, respectively.

As can be seen, the CFT is basically a further short-term 2D-DFT taken over the "standard" STFT $\boldsymbol{X}$. One of the main differences compared to modulation spectrograms is that the CFT is computed using the complex STFT $\boldsymbol{X}$, and not a magnitude representation such as $|\boldsymbol{X}|$. As we will show, this simple difference has many interesting consequences, notably that the CFT is invertible: the original waveform $\tilde{x}$ can be exactly recovered by cascading two classical overlap-add procedures. Another difference is that the patches span several frequency bins, *i.e.* we may have $N_a > 1$. This contrasts with the conventional modulation spectrogram, that is defined using one frequency band only.

A Probabilistic Model for the CFT

When processing an audio signal $\tilde{x}$ for source separation, it is very common to assume that all time-frequency (TF) bins of its STFT are independent [72, 82, 165, 202]. This is often the consequence of two different assumptions. The first one is to consider that all frames are independent, thus leading to the independence of all entries of the STFT that do not belong to the same column. The second one is related to the notion of stationarity: roughly speaking, the Fourier transform is known to decompose stationary signals into independent components. As a consequence, when the signals are assumed to be *locally stationary*, it is theoretically sound to assume that all the entries of their STFT are independent.

Still, both assumptions can only be considered as approximations. First, adjacent frames are obviously not independent, notably because of the overlap between them. Second, the stationarity assumption is only approximate in practice, especially when percussive elements are found in the audio, leading to strong dependencies among the different frequency bins. Let $\{\boldsymbol{X}_{ft}\}_{f,t}$ denote all the $N_a \times N_b$ patches taken on the STFT to compute the CFT, as depicted in Figure 6.6. The

2 Note that since each patch is complex, its 2D-DFT is not Hermitian, thus all its entries are kept.

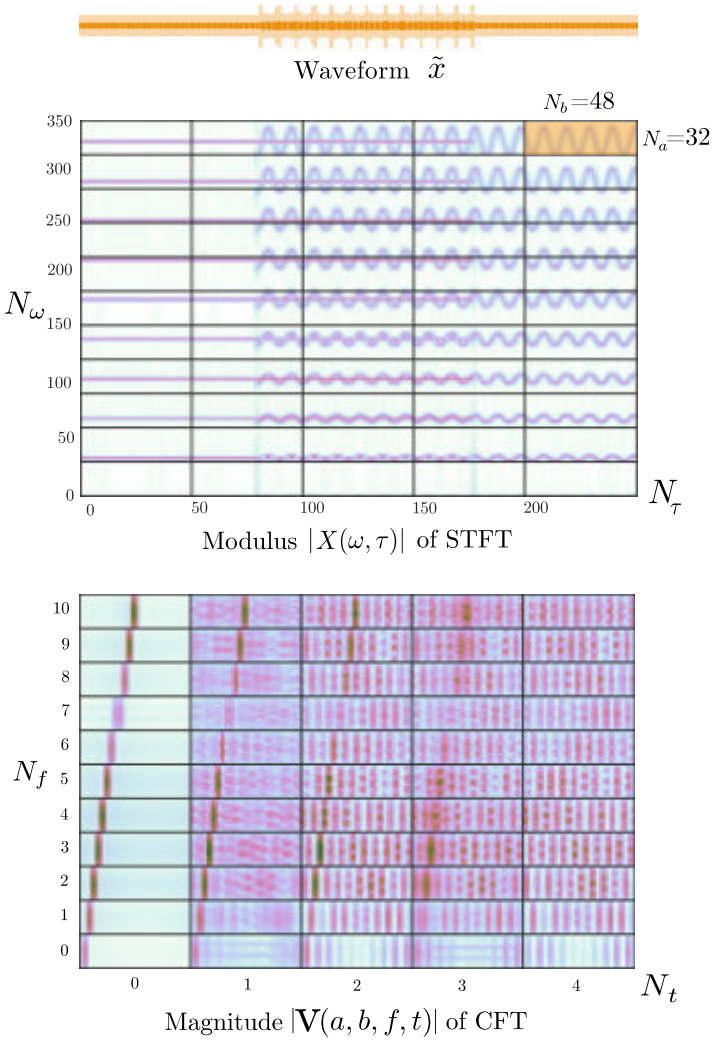


Figure 6.6: Common Fate Transform. For convenience, the splitting of the STFT into patches has been depicted without overlap, but overlapping patches are used in practice. ® 2016 IEEE.

probabilistic model we choose is the combination of *three* different assumptions made on the distribution of these patches[3].

1. All patches are independent. Just as the classical locally stationary model [165] assumes independence of overlapping frames, we assume here independence of overlapping patches. Due to the overlap between them, this assumption is an approximation, and one may wonder what the advantage is of dropping independent frames for independent patches. The answer lies in the fact that the latter permits us to model phase dependencies between neighboring STFT entries, and also to model much longer-term dependencies, as required for instance by deterministic damped or frequency-modulated sinusoidal signals.

3 A forth assumption made in [274] refers to the joint distribution of all entries of each patch which are α-stable [246].

2. Each patch is *stationary*: its distribution is assumed invariant under translations in the TF plane. This is where we do not assume independence, but on the contrary, expect dependencies among neighboring STFT entries. Our approach assumes this happens in a way that only depends on the relative positions in the TF plane. It can easily be shown that mixtures of damped sinusoids have this property. Assuming stationarity not only over time but over both time and frequency also permits us to naturally account for mixtures of frequency-modulated sounds. In short, we assume that throughout each patch, we observe one coherent STFT "texture". The difference with the HR-NMF model is that we have independent and identically distributed (i.i.d.) innovations for one given patch, whereas HR-NMF model has more variability. However, taking overlapping patches somehow compensates for this limitation.

3. All entries of the Fourier transform of each patch are assumed to be asymptotically independent, as the size of the patch gets larger. This rather technical condition, often tacitly made in signal processing studies, permits efficient processing in the frequency domain.

Under those assumptions, all entries of the CFT are independent (assumptions 1 and 2)[4] where $P(a, b, f, t)$ is a non-negative Tensor with dimensions $N_a \times N_b \times N_f \times N_t$ that we call the *modulation density*. In the general case, it can basically be understood as the energy found at (a, b) for patch (f, t), just like more classical power spectral densities describe the spectro-temporal energy content of the STFT of a locally stationary signal.

Interpretation of the CFT as Filterbank

An alternative interpretation of the CFT can be obtained by regarding the 2D-DFT as two subsequent 1D-DFTs. If the transform in frequency direction (DFT-F) is applied first, it is equivalent to a partial inverse DFT plus time reversal. If the time reversal would be undone and an overlap-add would be applied, the output would correspond to a subband representation with a frequency resolution of N_ω / N_a. Each of the N_a final transformations (DFT-T) in one patch takes output values from N_b DFT-Ts with equal indices. This corresponds to a splitting into poly-phase components with downsampling factor N_a of the time signal obtained by placing the output frames from the DFT-Ts in a row. Thus, the outputs of the DFT-Fs have a very high frequency resolution of $N_\omega N_b$ but contain aliasing components from the downsampling.

This interpretation of the CFT gives some indications for its benefits in the separation of modulated sources. Due to the poly-phase representation, it has a relatively high temporal resolution. The periodicities

4 This result is the direct generalization of [246, th. 6.5.1] to multi-dimensional stationary processes.

in the spectra caused by downsampling make the CFT relatively independent of frequency shifts, so that, for example, the output patch of a single sinusoidal sweep is mainly influenced by the sweep rate.

6.2.2 Signal Separation

Now, let us assume that the observed waveform is actually the sum of K underlying components $\{\tilde{s}_k\}_{k=1,\dots,K}$. Due to the linearity of the CFT, this can be expressed in the CFT domain as:

$$\forall\,(a,b,f,t)\,,x\,(a,b,f,t)=\sum\nolimits_k s_k\,(a,b,f,t)\,.$$

If we adopt the model presented above for each source and use the stability property, we have:

$$\boldsymbol{V}\,(a,b,f,t)\sim\sum\nolimits_k P_k\,(a,b,f,t)\,,$$

where P_k is the modulation density for component k. The resulting waveforms are readily obtained by inverting the CFT. As can be seen, we now need to estimate the modulation densities $\{P_k\}_k$ based on the observation of the mixture CFT x, similarly to the estimation of the sources' Power Spectral Densities (PSD) in source separation studies.

Factorization Model and Parameter Estimation

In order to estimate the sources' modulation densities, we first impose a factorization model over them, so as to reduce the number of parameters to be estimated. In this study, we set:

$$\boldsymbol{P}_{abft}\approx\sum_{k=1}^{K}a_{abfk}h_{tk}, \tag{6.6}$$

for $a=1,\dots,N_a;b=1,\dots,N_b;f=1,\dots,N_f;t=1,\dots,N_t;k=1,\dots,K$ non-negative tensors, respectively. We call this a *Common Fate Model*. Intuitively, $\boldsymbol{A}\triangleq\{a_{abfk}\}_{a,b,f,k}$ is a modulation density template that is different for each frequency band f, and that captures the long term modulation profile of each source around that frequency. Then, $\boldsymbol{h}\triangleq\{h_{tk}\}_{t,k}$ is an activation vector that indicates the strength of source on the patches located at temporal position t. The factorization model is depicted in Figure 6.7. We also experimented with other two and three-factor combinations but never got any promising results, suggesting that our proposed NMF-like model is a good choice.

Learning those parameters can be achieved using the non-negative tensor factorization (see e.g. [56, 202, 265] for an overview), except that it is applied to the CFT instead of the STFT, and that the particular factorization to be used is equation 6.6. In essence, it amounts to estimating the parameters $\{A_k,H_k\}$ so that the modulus of the CFT is as close as possible to $\sum_k P_k$, with some particular cost function as a

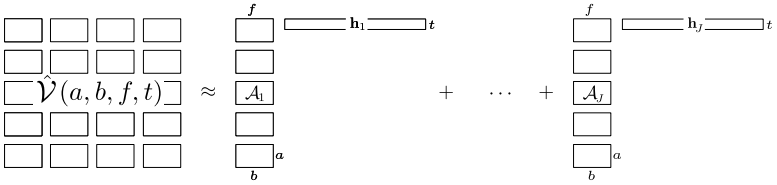


Figure 6.7: Visualization of the Common Fate factorization model (CFM).

Algorithm 1 Fitting parameters of the non-negative CFM equation 6.6.

With $v = |x| \, \forall a,b,f,t v(a,b,f,t) = |x(a,b,f,t)|$ and always using the latest parameters available for computing $\hat{P}(a,b,f,t) = \sum_{k=1}^{K} A_k(a,b,f) H_k(t)$, iterate:

$$A_k(a,b,f) \leftarrow A_k(a,b,f) \frac{\sum_t v(a,b,f,t) \hat{P}(a,b,f,t)^{\cdot(\beta-2)} H_k(t)}{\sum_t \hat{P}(a,b,f,t)^{\cdot(\beta-1)} H_k(t)}$$

$$H_k(t) \leftarrow H_k(t) \frac{\sum_{a,b,f} v(a,b,f,t) \hat{P}(a,b,f,t)^{\cdot(\beta-2)} A_k(a,b,f)}{\sum_{a,b,f} \hat{P}(a,b,f,t)^{\cdot(\beta-1)} A_k(a,b,f)}.$$

data-fit criterion called a β-divergence and which includes Euclidean, Kullback-Leibler and Itakura-Saito as special cases [85]. As usual in non-negative models, each parameter is updated in turn, while the others are kept fixed. We provide the multiplicative updates in Algorithm 1. After a few iterations, the parameters can be used to separate the sources using the Wiener filter as described in [164].

6.2.3 Experiments

In this section, we present separation experiments utilizing CFM and compare it with other methods.

Method	Signal Representation	Factorization Model
CFM [282]	STFT $\rightarrow$ Grid Slicing $\rightarrow$ 2D-DFT	$V(a,b,f,t) = P(a,b,f) \times H(t)$
NMF [306]	STFT	$V(f,t) = W(f) \times H(t)$
HR-NMF [17]	Output of any filterbank (STFT, MDCT, ...)	AR filtering of NMF excitation
MOD [21]	STFT $\rightarrow |\ldots| \rightarrow$ STFT along each bin	$V(f,m,t) = W(f) \times A(m) \times H(t)$
CFMM	STFT $\rightarrow |\ldots| \rightarrow$ Grid Slicing $\rightarrow$ 2D-DFT	$V(a,b,f,t) = P(a,b,f) \cdot H(t)$
CFMMOD	STFT $\rightarrow |\ldots| \rightarrow$ Grid Slicing $\rightarrow$ 2D-DFT	$V(a,b,f,t) = P(a,b,f) \cdot H(t)$

Table 6.1: Overview of the evaluated algorithms.

Synthetic Example

To illustrate the CFT representation, we processed a mixture consisting of two sinusoidal sources. One source is a pure sine wave of fundamental frequency 440 Hz whereas the other is frequency modulated by a sinusoid of 6.3 Hz. In the first step, a STFT with a DFT-length of 1024 samples and a hop-size of 256 samples was processed at a sample rate of 22.05 kHz. Patches of size $(N_a, N_b) = (32, 48)$ (not respecting overlaps) were then taken from the STFT output. Figure 6.6 in Section 6.2.1 then shows the Common Fate Transform for the mixture. One can see that the CFT representation shows distinct patterns across time, suggesting that the factorization is able to separate the sources. Furthermore, if we now look at a smaller excerpt of the same synthetic example, depicted in Figure 6.8, we can also observe the additivity property of the common fate representations.

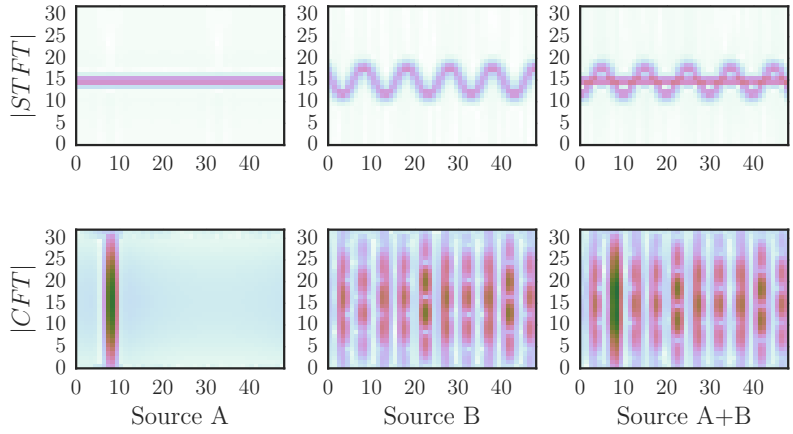


Figure 6.8: Examples of patches of size $(N_a, N_b) = (32, 48)$. The upper row shows the STFT output, the lower row the corresponding Common Fate Transform (CFT).

Objective Evaluation on Unison Instrument Mixtures

To evaluate the proposed method, five musical instrument samples were selected from the Unison Separation Dataset [279] — all of them feature vibrato: violin, cello, tenor sax, English horn, and flute. It is important to note that vibrato techniques differ between these instruments: whereas the English horn and the flute only produce a very subtle modulation, the violin and tenor sax have powerful frequency modulations with a higher modulation frequency as well as a higher modulation index. All samples last about three seconds. We then generated a combination of ten mixtures of two instruments, each one generated with a simple SourceA — SourceB — (SourceA + SourceB) scheme. Data were encoded in 44.1 kHz / 16 bit. We compared the separation performance of six different methods, summarized in Table 6.1:

CFM For the CFM model, we took an STFT with frames of 1024 samples and a hop-size of 512 samples. The resulting complex STFT was then split into a grid of patches of size $(N_a, N_b) = (4, 64)$, each having a half-window overlap in both dimensions.

MOD We implemented a modified version of [21] where for the sake of comparability, we used a STFT instead of a gammatone filterbank. A DFT length of 1024 and a hop-size of 512 samples were chosen. After taking the magnitude value, a second STFT of size 32 and hop-size 16 samples was computed for each frequency.

CFMMOD We selected patch sizes of $(N_a, N_b) = (1, 64)$ and modified the representation so that the magnitude was applied before computing the 2D-DFT. This permits to compare the advantage of our proposed factorization model (6.6) over MOD, when using the same kind of energy-modulation representation in both cases.

CFMM For comparing the influence of computing modulations over complex STFT or magnitude STFT, we tried our factorization model when the magnitude of the STFT is taken before 2D-DFT, with patches of the same size as for the CFM method.

NMF We took a "standard" NMF based method [306]. We highlight that taking a STFT with frames of length 1024 would not make a fair comparison, because the CFM model actually results in a larger frequency resolution. Therefore a comparable NMF is based on an STFT of DFT-length 32768.

HR-NMF See description in [175].

All factorizations ran for 100 iterations and were repeated five times. We chose $k = (2, \ldots, 6)$ components for each factorization. For $k > 2$ we used oracle clustering to show the upper limit of SDR which can be achieved.

We ran the performance evaluation by using BSSeval [301]. The results of SDR, SIR, and SAR are depicted in Figure 6.9. Results indicate that the CFM model performs well in all measures. However, in terms of SIR, the results of HR-NMF are better than CFM method. The results for CFMMOD and CFMM indicate the positive influence of the CFM factorization compared to [21]. The results of CFMM indicate that the complex CFT lead to better results. NMF did perform surprisingly well, which may only hold for our test set, where each source is active for a long period. This results in a cyclic stationary vibrato, revealing spectral side lobes at such a high resolution. With more than one component per source, the results of CFM do improve, but it can be seen that more than two components ($k = 4$) will not increase the SDR values as indicated in Figure 6.10. The separation results and a full

Python implementation of the CFM algorithm can be found on the companion website [5].

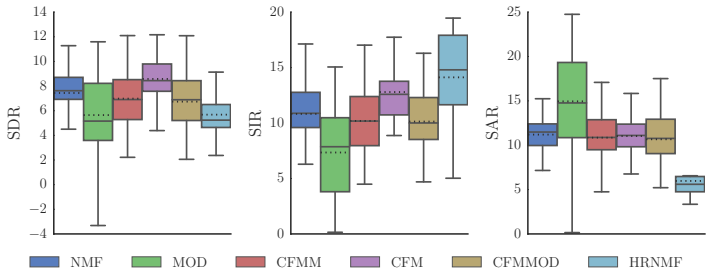


Figure 6.9: Boxplots of BSS-Eval results of the unison dataset. Solid/dotted lines represent medians/means. ® 2016 IEEE.

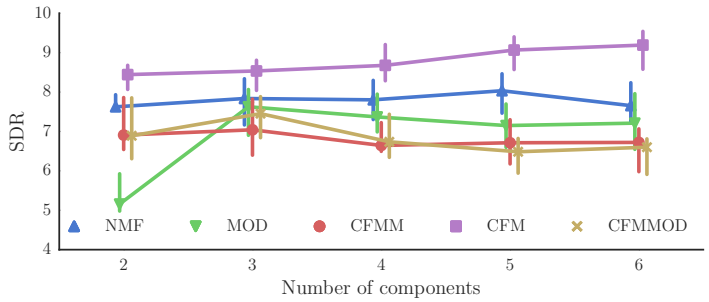


Figure 6.10: Boxplots of SDR values of the unison dataset over the number of components k. For $k > 2$ oracle clustering was applied. ® 2016 IEEE.

6.3 COMMON FATE TRANSFORM FOR MUSIC SEPARATION

In the previous section, we showed that the Common Fate Model is suitable to separate highly overlapped signals based on their spatial-temporal modulation texture. In this section, we want to show how this method can be extended for the application of vocal and accompaniment separation [218]. This scenario is significantly more complex than the separation of instrument mixtures, hence the separation model needs to be flexible enough to handle many of the critical edge cases which make music separation challenging. With the recent success of machine learning models [117], it became likely that an unsupervised model such as NTF or CFM may not be flexible enough to enforce the significant amount of domain knowledge that is present in this scenario to improve performance. Rafii et. al describe the current machine learning situation in [218]:

5 github.com/aliutkus/commonfate

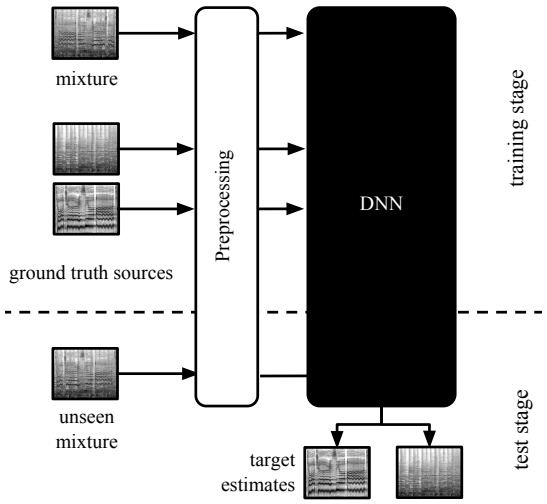


Figure 6.11: General architecture for methods exploiting deep learning. The network inputs the mixture and outputs either the sources magnitude STFT or a TF mask. Methods usually differ in their network architecture and the way use the training data for learning (a modified version of this Figure was published in [218]). ® 2018 IEEE.

"Taking advantage of the recent availability of sufficiently large databases of isolated vocals along with their accompaniment, several researchers investigated the use of machine learning methods to directly estimate a mapping between the mixture and the sources [128, 293]. However, most systems today still use classical time-frequency representations.

The common structure of deep learning methods for lead and accompaniment separation usually corresponds to the one depicted in Figure 6.11. Most methods mainly differ in the architecture picked for the network, its input, and output representation as well as in the way the network is trained.

For the understanding of this section, it is sufficient to mention that DNNs consist of a cascade of several, possibly non-linear transformations of the input, which are learned during a training stage. They were shown to effectively learn representations and mappings, provided, enough data is available for estimating their parameters [63, 96, 152]. Different architectures for neural networks may be combined/cascaded together, and many architectures were proposed in the past, such as fully-connected neural network (FNN), (convolutional neural network (CNN)), or recurrent neural network (RNN) and variants thereof such as the long short-term memory network (LSTM) and

> the gated-recurrent units (GRU). Training of such functions is achieved by stochastic gradient descent [232] and associated algorithms, such as backpropagation [238] or backpropagation through time [237] (BTT) for the case of RNNs.
>
> Huang et al. were the first to propose RNNs [114, 204] for singing voice separation in [128, 129]. They adapted their framework from [127] to model all sources simultaneously through masking. Input and target functions were the mixture magnitude and a joint representation of the individual sources. The objective was to estimate jointly either singing voice and accompaniment music, or speech and background noise from the corresponding mixtures.
>
> Modeling the temporal structures of both the lead and the accompaniment is a considerable challenge. As an alternative to the RNN approach proposed by Huang et al. in [128], Uhlich et al. proposed the usage of simpler FNNs [293] whose input consists of *supervectors* stacked of few consecutive frames from the mixture."

We decided to reimplement Uhlich's model [293] to evaluate the separation quality. The aim of this work was not to exactly reproduce the results, but instead, to evaluate one main research questions: does a DNN-based model benefit from the common fate representation being able to better capture the modulation texture? For the implementation of the model, we used the Keras [53] deep learning framework to systematically assess different combinations of input-and-output representation of the system.

The network, as proposed in [293] consists of three fully connected layers, where each of the hidden layers has the same number of hidden nodes as the targeted output representation. This method can be described as a variant of a stacked denoising autoencoder [303], where the noisy input is mapped to a clean output of the same dimensionality. The architecture is depicted in Figure 6.12.

6.3.1 Inputs and Outputs

The purpose of the model is to create a non-linear mapping function from the magnitude of the input mixture $\boldsymbol{X}$ to the magnitude of the target source $\boldsymbol{Y}_j$. The optimal parameters (weights) θ_j of such a mapping function $\boldsymbol{Y}_j = f_{\theta_j}(\boldsymbol{X})$ are learned via supervised training. FNN networks, such as the one used here, can only deal with temporal structure by reshaping the time-frequency input to a super vector to be processed by the FNN. However, this drawback is compensated by a large number of parameters in an FNN layer.

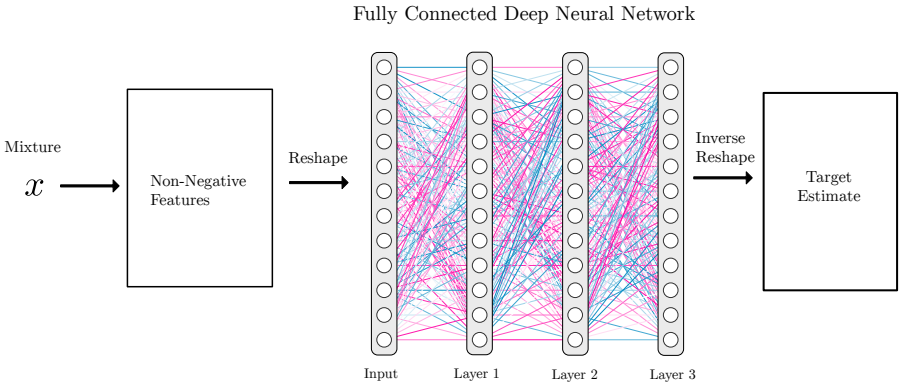


Figure 6.12: Simplified block diagram of fully connected denoising autoencoder network as proposed by [293] for time-frequency based separation.

Since the STFT, GFT and CFT are lapped transforms, different scenarios for the input and output representation of the FNN can be envisioned:

STFT-STFT : for the *input*, we computed the STFT ($N = 1024, Hop = 512$) of each the audio track and selected excerpts of size, $\mathbf{X} \in \mathcal{R}_+^{2C+1}$, where C is the number of preceding and succeeding frames around the central frame $\mathbf{X}_{i+C}$. For the *output*, only a single central frame $\mathbf{Y}_{i+C}$ is selected. We used $C = 2$, reflecting the setting in [293]. This results in an input sample size of $\mathbf{X} \in \mathcal{R}_+^{5\times513}$ and $\mathbf{Y} \in \mathcal{R}_+^{1\times513}$.

GFT-GFT/GFT-STFT : instead of taking excerpts from the STFT, like in *STFT-STFT*, we computed overlapping patches, as described in Section 6.2.1. Each patch is of size $(5, 8)$, which means that the same number of time frames are used compared to *STFT-STFT* but additional redundancy has been added because of the overlap between neighboring patches. For the *output*, we chose the GFT of Y. This results in an input sample size of $\mathbf{X} \in \mathcal{R}_+^{128\times5\times8}$ and $\mathbf{Y} \in \mathcal{R}_+^{128\times5\times8}$. Furthermore, to reduce the number of parameters, we also evaluated a setting where just the *output* is the central frame of the STFT.

CFT-CFT/CFT-STFT : in the first step, a processing as in *GFT-GFT* was applied and then the 2D-DFT transform was applied (see Section 6.2.1). This results in identical shapes as in the *GFT-GFT* but with added benefits of this representation that can model neighboring phase dependencies.

We used the DSD100 dataset [199] for training and test. For each sample fed into the network, we randomly selected mixtures (without replacement) from the DSD100 dataset.

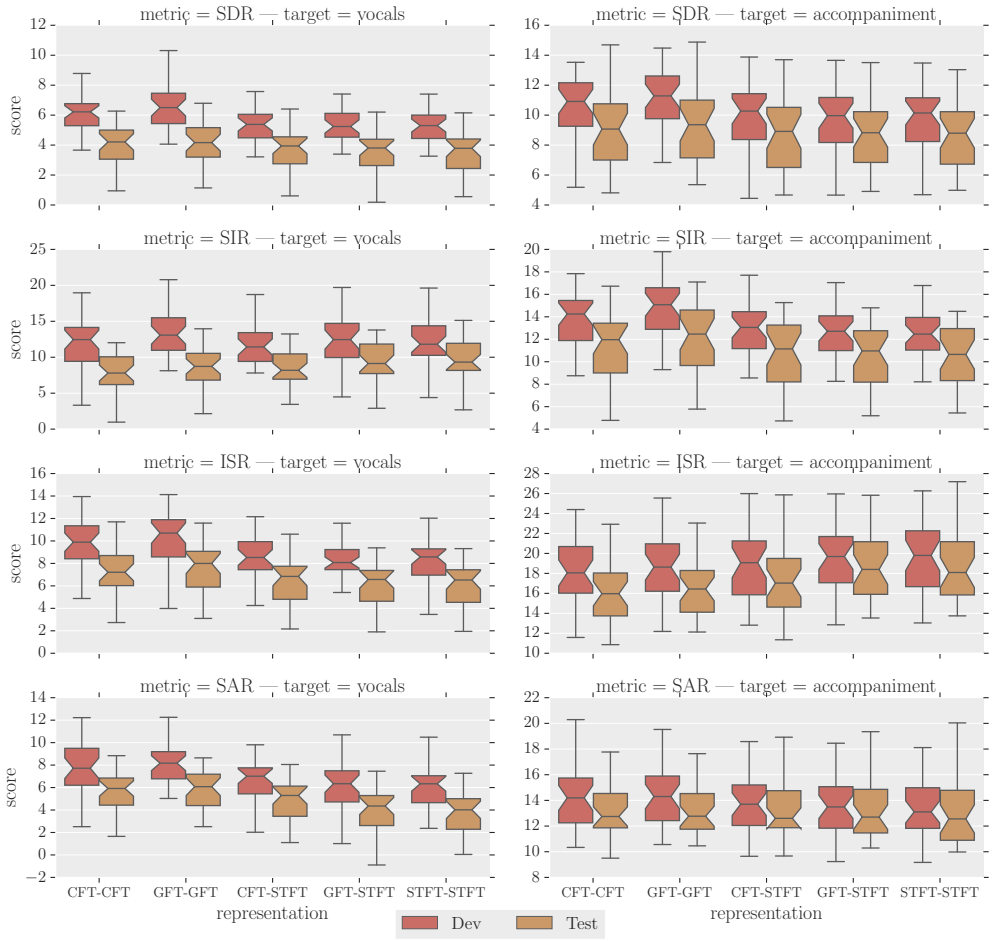


Figure 6.13: BSSEval separation results of the DSD100 dataset results for vocals and accompaniment sources. Several combinations of 'input-output' were tested, as indicated by the x-axis.

6.3.2 Training

We then sample from the DSD100 tracks and form single samples as input for the FNN. Therefore, we first compute the input representation of an audio track from the DSD100 set and then randomly sampling without replacement from these tracks. The actual training has been done using mini-batches of size 32. Each architecture is trained using the ADAM optimizer [143] (learning rate: $1 \cdot 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \cdot 10^{-8}$). The model was trained for a fixed number of 30 epochs and, in contrast to [293], greedy layerwise pre-training was not applied.

6.3.3 Results

For evaluation, the BSSEval metrics of all representations for the DSD100 test set were computed. The results are depicted in Figure 6.13. They indicate that the common fate representation is indeed improving the baseline *STFT-STFT* results. Overall, we can report a mean

difference of 0.4 dB for CFT-CFT and 0.5dB for GFT-GFT, compared to the *STFT-STFT* representation. However, it is worth mentioning that both, the GFT and the CFT representation lead to a significant increase in redundancy in the representation, thus increasing the number of trainable parameters per network layer (26 million for CFT vs 0.26 million for STFT). Since we could not observe a large increase in difference of training (Dev) vs. test (Test) performance, we assume that the increasing number of parameters does not lead to large overfitting.

6.3.4 Submission to SiSEC 2016

The results have been submitted to the 2016 SiSEC [172]. This enables to relate the work compared to 22 other source separation methods, all evaluated on the same test data, as part of the task for separating professionally-produced music recordings at SiSEC 2016. Table 6.2 lists the participating systems.

Acronym	Ref.	Summary
STO1	Proposed	FNN on GFT representation
STO2	Proposed	FNN on CFT representation
HUA	[126]	RPCA standard version
RAF1	[222]	REPET standard version
RAF2	[171]	REPET with time-varying period
RAF3	[221]	REPET with similarity matrix
KAM1-2	[164]	KAM with different configurations
CHA	[45]	RPCA with vocal activation information
JEO1-2	[134]	l_1-RPCA with vocal activation information
DUR	[73]	Source-filter NMF
OZE	[244]	Structured NMF with learned dictionaries
KON	[129]	RNN
GRA2-3	[100]	DNN ensemble
UHL1	[293]	FNN with context
NUG1-4	[196]	FNN with multichannel information
UHL2-3	[294]	LSTM with multichannel information
IBM		ideal binary mask

Table 6.2: Methods evaluated in SiSEC 2016.

The objective scores for our proposed methods were obtained using BSSEval and are given in Figure 6.14.

An obvious observation in Figure 6.14 is the difference in performance between data-driven methods and "classical" unsupervised methods. Further, it shows that exploiting learning data does help separation compared to only relying on *a priori* assumptions such as the harmonicity or redundancy. Additionally, dynamic models such as LSTM from UHL2-3 appear more adapted to music than FNN. These good performances in audio source separation go in line with the success of DNNs in fields as varied as computer vision, speech recognition, and natural language processing [152].

Figure 6.14: BSSEval scores for the vocals and accompaniment estimates for SiSEC 2016 on the DSD100 test dataset. Scores are sorted according the vocal median SDR and grouped by supervised and unsupervised methods. The proposed systems are *STO1*, based on GFT-GFT and *STO2*, based on CFT-CFT. Dashed lines indicate respective group medians.

Relating our results of STO1 and STO2 to the other methods, we observe that the performance was only slightly below the two state-of-the-art performances of UHL and NUG. Furthermore, the difference between test and validation dataset indicates, that our CFT DNN model even has better generalization as the one in NUG. While our STO model shares the same network architecture with UHL1, we were unable to reproduce the results, as we are approximately 1 dB below UHL1. One reason is in the difference in initialization of the model as well as the fact that NUG and UHL models exploit multichannel information.

6.3.5 Evaluation Website

For more details and the ability for playback of the estimates, we refer to the dedicated interactive website that we built as part of my organizational help for SiSEC[6]. In fact, for the first time, interested researchers are now able to listen to over 10000 stimuli from all participating systems. This was made possible through modern JavaScript technologies like the Web Audio API to interactively assess source separation results in the browser. For each track, separation results are provided as well as the objective BSSeval scores, all of the data is reproducible and was made available in [285]. Figure 6.15 depicts a screenshot of the website. The objective scores are depicted as an interactive matrix where users are able to sort the results for each track and each system interactively by the source of interest. Clicking on one rectangle in the heatmap opens the interactive player that allows to simultaneously playback the separated sources including changing the volume of each source.

6.4 SUMMARY AND DISCUSSION

In this chapter, we presented methods that exploit modulations for source separation without knowing them a priori. In the first part of this chapter, we presented a study where we demonstrated the use of the modulation spectrogram tensors for separating unison instrument mixtures, comparing the results with those presented in the previous chapter. In a next step, we proposed a complex tensor representation, the Common Fate Transform (CFT), computed from rectangular patches of the complex STFT using two-dimensional DFTs. This novel representation exposes joint modulation characteristics of amplitude and frequency modulated signals while being fully invertible. We demonstrated the usefulness of this representation using our Common Fate Model that factorizes patches from the CFT into two components, a modulation pattern and its activation. The model

6 http://www.sisec17.audiolabs-erlangen.de

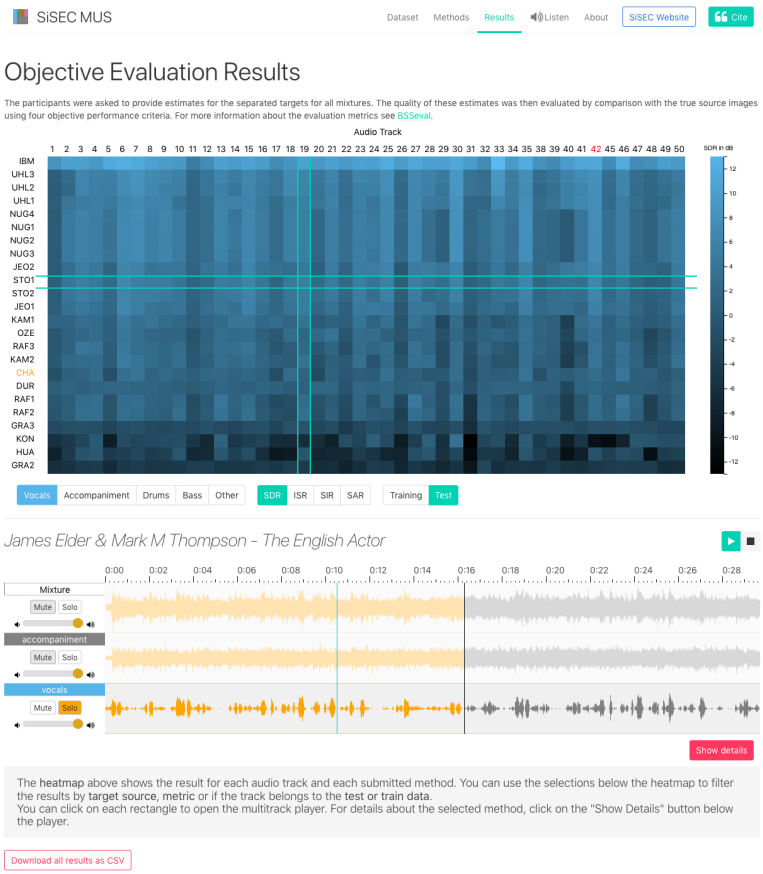


Figure 6.15: Screenshot of the SiSEC 2016 Website `http://sisec17.audiolabs-erlangen.de`

is inspired by the human's ability to group common modulations into single sources. We presented results on unison musical instrument mixtures, indicating that it outperforms existing methods.

In the second part of this chapter, we combined the CFT with an existing deep learning based separation model [293]. We showed that the CFT improved separation quality compared to STFT in a fully connected deep neural network. Even though the CFT significantly increases the input size (due to its redundancy) and the number of trainable network parameters (due to its stacked auto-encoder architecture), generalization performance did not suffer.

The results of the best performing model were submitted to SiSEC 2016, where we scored among the top three participating research teams. Finally, we presented interactive evaluation tools, developed for SiSEC, allowing to interactively assess the performance of separation systems both objectively and subjectively.

7 ONE, TWO, THREE, MANY: PERCEPTUAL EXPERIMENTS ON ESTIMATING THE NUMBER OF SOURCES

The separation of mixtures into its original audio sources, as presented in the previous chapters, is a challenging task. Furthermore, source separation is difficult to evaluate, and researchers compare to a known reference even though this does not reflect how humans separate mixtures. It has long been a research topic to answer the question *if* humans can separate by fully extracting the desired source or if we can focus our attention on one source — segregate them [32]. Moreover, despite recent progress in auditory science [43, 150, 217], research still is investigating how separation takes place in the auditory cortex or other parts of the human brain. One thing, however, we can assess directly is if humans can reliably detect the number of sources in a mixture of several sources.

In order to address this question, it helps to understand how humans infer counts and if our strategy is depending on the count. When we look at vision, these are questions that have already been discussed over a hundred years ago in scientific research; an early study in the field of psychology of vision was published by Jevons in 1871 [135]. Jenvons presented an experiment to quickly infer the number of objects (beans) and came up with the hypothesis that humans can instantly estimate the number of objects without actually counting and therefore identifying them. Jenvons mentioned in [135]:

> "It is well known that the mind is unable through the eye to estimate any large number of objects without counting them successively. A small number, for instance, three or four, it can certainly comprehend and count by an instantaneous and apparently single act of mental attention."

This "one-two-three-many" hypothesis was a fundamental observation. The fact that we can directly infer the numerosity of small numbers of objects, up to about four, is also known as *subitizing* [35, 139]. We refer to this strategy as "direct count estimation". Concerning our hearing, there are indications that the auditory system is capable of subitizing audio sources [120]. And surprisingly, as shown in [138, 140], humans share the same limitations of correctly estimate up to three simultaneously active speakers.

In this chapter, we present two experiments contributing to this interesting field of research. The first experiment (Section 7.1) addresses

the question if the number of instruments in polyphonic music is subject to the same limitations. In the second contribution (Section 7.2) the aim is to verify the findings of an earlier experiment focussed on speech and increase the number of stimuli to be used for comparison of machine learning based count estimations methods

7.1 INSTRUMENT COUNT ESTIMATION

This chapter was previously published in [276] and has been revised for this thesis.

While source separation methods can be objectively evaluated given a true reference, a human versus machine comparison is cumbersome because measuring the human's ability to perform separation is difficult. However, one can easily evaluate if humans can detect the number of sources in a mixture of several sources. In this Section, we take a first step towards designing an experiment where we focus on polyphonic music of inhomogeneous timbre, where the question is: What is the number of instruments humans can estimate correctly? Such knowledge can be used in auditory modeling or as a pre-processing step for source separation algorithms.

In previous work, the perception of concurrent sound sources has been analyzed on different scales so far. Bregman's and McAdams' auditory stream theory [182] can be seen as an analytical way of describing how sound events are perceived by the human auditory system. Unfortunately, it is difficult to model professionally produced music by auditory stream models because of its high complexity. Also, none of these models is motivated to predict the perceived number of musical sources. There are indications that for this task, humans tend to fail if more than three sources are present at the same time [131]. Kashino et. al [138] addresses the questions for concurrent speakers in a "cocktail party" like environment and found an upper limit of three voices humans can perceive. When the focus shifted to musical instruments as sources, research took concepts from musicology into account. Huron [131] was the first who addressed this question in 1989 at a musically meaningful level. Huron asked for the number of voices within a piece of music, whereby voices in musicology one can define it as a line of sound or note events (See [38] for further definitions). Huron determined by experimental results that the number of correctly identified voices is up to three.

Several results are addressed in this section, including a possible upper limit of the number of perceived instruments but also if one can see significant differences in the performance of musicians compared to non-musicians.

7.1.1 Experiment

For the purpose of gaining more knowledge in understanding the human perception of multiple present instruments, an experiment was conducted. Huron selected voices from organ pieces only. We wanted to address the more general case where voices are played by different instruments. As we set our focus on comparison between musicians and non-musicians, our experiment was designed so that it respects the fact that the latter have only limited musical background.

Although it might be interesting to have direct comparison with Huron's experiment, we agree that expanding the methods to an inhomogeneous timbre case is error prone. One reason is that there is reasonable doubt about the non-musicians understandings in terms of how a voice is defined. This is why we choose a trade-off with a more simplified experiment where we asked for the number of instruments instead of voices. Also whereas Huron [131] excluded subjects from his experiment because of their lower performance, we compared the results of both groups.

7.1.2 Stimuli

The selection of music items is crucial for our experimental setup. Usually music recordings have no ground truth metadata available to determine the actual number of instruments. Using annotated music like that from the RWC database [98] fulfills this requirement but lacks the possibility to remix, attenuate or suppress specific sources. This is important so that the experiment consists of equally grouped stimuli. Instead of the original RWC recordings, the annotated MIDI data itself was used as prototypes for the stimuli. To make the count estimation task less ambiguous for the subjects, the instrumentation was chosen to be mostly constant during the music piece. Therefore we calculated an "instrumental stationarity" metric. The annotated MIDI files from [98] were converted into piano roll representations for each instrument channel. This representation was then converted into a binary *instrumentation activity matrix* $\mathbf{I}_{AM}[\underline{\mathbf{k}}_1|\underline{\mathbf{k}}_2|...|\underline{\mathbf{k}}_N] \in \{0,1\}$, where at each discrete time instance i a vector $\underline{\mathbf{k}}_i$ indicates which instruments are active. The aim is then to select frames of length N which are stationary by means of changes in instrumentation and activity. To get many items with a high instruments count, the maximum number of instruments within a frame was stored in a binary mask $\underline{\mathbf{k}}_{max}$ which was compared with all $\underline{\mathbf{k}}_{i=1...N}$ so that $(|\underline{\mathbf{k}}_i \oplus \underline{\mathbf{k}}_{max}| \leq 1) \vee (\underline{\mathbf{k}}_i = \underline{0})$. The resulting binary vector was smoothed with an averaging kernel of size N. By peak picking we got a list of possible candidates which contained a high stationarity in instrumentation. Further the RWC files were filtered a priori to exclude items dominated by electronic instruments or singing voice. Table 7.1 presents the selected 12 items

representing pairs of one to six simultaneously present instruments. Each item is around seven seconds long. By cutting at note offsets we varied the lengths of the items to make it semantically more meaningful. Six items (notated as RM-C***) belong to the classical western music genre whereas the other items are of mixed genre.

The MIDI files were humanized randomly and rendered in a professional sequencer software utilizing state-of-the-art commercial sampling products. The process is similar to the dataset creation mentioned in Section 4.1. The rendered files were processed with convolutive reverb to match the original recordings. Additionally a loudness normalization was applied according to EBU-R128[75]. To avoid spatial cues every item was rendered to mono at 16 bit/44.1 kHz.

RWC ID	Start [s]	Dur. [s]	Instrumentation	k
J021	46.5	6.6	Piano, Contrabass (pizz.) and Trumpet	3
C001	0.0	9.0	Bassoon	1
G047	35.3	8.3	Violoncello	1
C016	0.9	7.6	Viola and Violoncello	2
G068	132.4	6.6	Violin and Flute	2
C018	240.4	5.4	French Horn, Piano and Violin	3
G046	0.3	7.9	Contrabass, Piano and Violoncello	3
C013	5.6	6.0	Flute, Viola, Violin and Violoncello	4
G036	0.0	6.5	Acoustic Guitar, Electric Bass, Piano and Violin	4
C012	112.0	6.0	Contrabass, Flute, Viola, Violin and Violoncello	5
G037	67.1	7.0	Acoustic Guitar, Contrabass (pizz.), Flute, Piano and Tenor Sax	5
C001	147.8	6.0	Bassoon, Clarinet, Contrabass, French Horn, Oboe and Violin	6
G028	17.5	6.5	Electric Bass, Electric Guitar, Flute, Piano, Trombone and Trumpet	6

Table 7.1: Selected items from the RWC Music Database [98]. Item *J021* is used as training item.

7.1.3 Methods and Participants

The experiment was attended by 62 participants, where half of them regularly play a musical instrument. They were asked to count how many different instruments they can hear. 12 items from the test set (Table 7.1) were played back in random order. The experiment was presented by a user interface depicted in Figure 7.1. Except for the training item, every subject could play back each stimulus up to three times. Additionally they were asked to estimate how certain they were in their decision (ranged from *uncertain* to *very certain*). Instead of a slider UI-element, the interface only features plus and minus buttons so that the subjects were not biased about the maximum number of instruments. Item *J021*** had been selected as a training item and was presented to the subjects during the introduction phase to make them

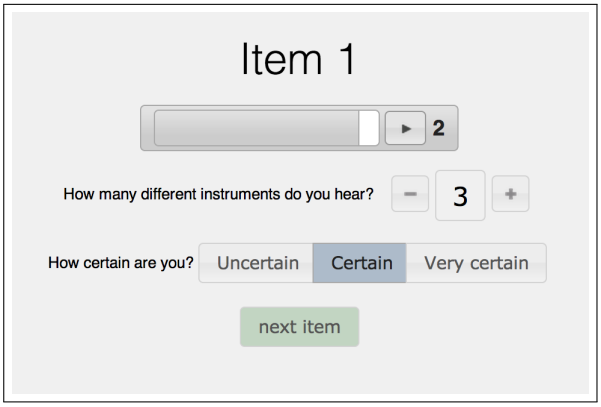


Figure 7.1: Experiment User Interface

familiar with the user interface. This trial also unveiled the number and name of the instruments within that piece. After they had read the introduction page, the subjects were asked to adjust the volume during the training example to their preference and leave the volume at that level for the duration of the experiment. The stimuli were presented on BEYERDYNAMICS DT770 headphones connected to a RME BABYFACE. The complete test took about 20 minutes on average for every participant.

7.1.4 Results: A gap of one instrument

The independent variable $I(i)$ is the number of instruments of one music item i where in this case $I(i) \in \{1, 2, ..., 6\}$. $R(i, s)$ is defined as the number of instruments that are perceived and counted by subject s for music item i. The dependent variable is then derived from the main subject response as $\Delta(i, s) = I(i) - R(i, s)$ transformed into a binary scale:

$$E(i, s) = \begin{cases} 0 & \text{if } |\Delta| = 0 \\ 1 & \text{if } |\Delta| > 0 \,. \end{cases} \tag{7.1}$$

The primary statistical null hypothesis (H_1) is stated in that the means of Δ and E[1], grouped by the number of instruments, do not differ significantly. As we also want to test the between-groups performance of musicians versus non-musicians, we introduce another dependent variable $M(s) \in \{0, 1\}$ of binary scale. This is stated in a secondary null hypothesis (H_2) where the means of Δ and E are not significantly different between musicians and non-musicians. No subjects were screened from the results, although there are two cases

1 The fact that E is dichotomous will lead to a mean value that equals to a probability of a binary distribution.

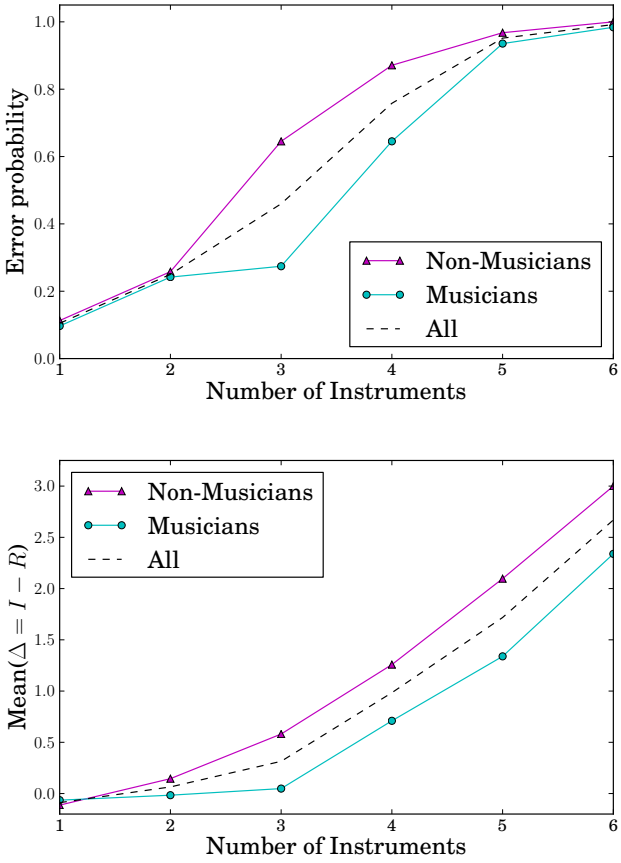


Figure 7.2: Error probability (top) and Mean of $\Delta = I - R$ (right) categorized by the number of instruments.

where no valid response had been made. Results are grouped by items of I instruments.

In general, participants tended to perform worse for items with more than two instruments. The probability of correctly estimating one instrument was 90.0% whereas only one person out of 62 gave a correct response for an item with six instruments. In some cases, the number of instruments does not correspond to the number of voices for every item. Items where an instrument plays more than one voice and voices which are played by more than one instrument. However, most of the chosen instruments are monophonic so in our case, this occurred only for items where piano or guitar is present. Also, we made sure that the number of total voices did not exceed the maximum number of instruments in that item. Voices being played by more than one instrument (unison), present in Go68, showed surprisingly good results.

Underestimation

We confirmed the results in [131] that the most common error is underestimation of one instrument, although this accounts only for 43 % of the responses in our experiment. Only in one case Δ is negative (overestimation) which is item C016, a "Clarinet Quintet in A major by Wolfgang Amadeus Mozart (K.581. 1st movement)" where we have excluded the solo clarinet part and two strings. Still, the remaining sound seems to be so similar to that of a quartet that musicians tended to hear "phantom" instruments.

Self-Evaluation

Figure 7.3 shows the results of the subjects certainty grouped by instrument count. Although the rate of "very certain" responses drops down to 11.3% for items with six instruments the rate of "certain" responses is still as high as 43.5%. When we take Δ into account we find a significant linear correlation between Δ and certainty where 0 is uncertain and 2 is very certain (Pearson's $r = -0.227$ at the $p = 0.05$ level).

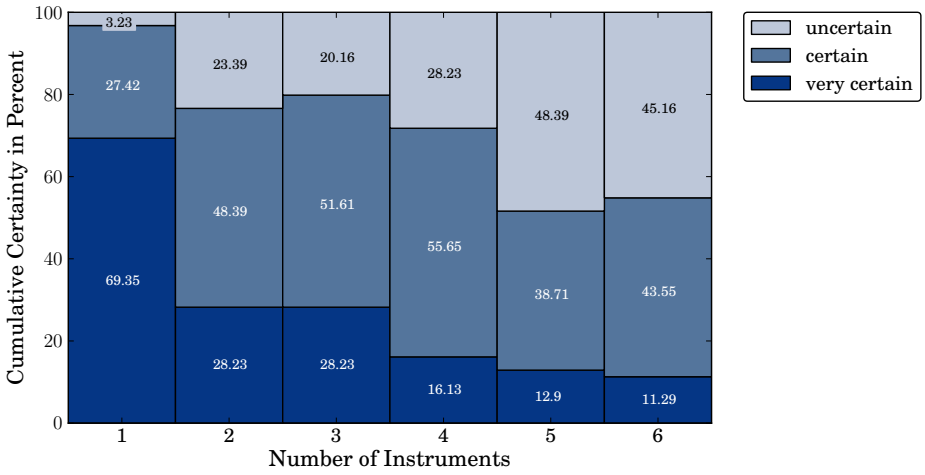


Figure 7.3: Responses for certainty of subjects by number of instruments

Main Effects

To test the null hypotheses (H_1 and H_2), statistical tools are required. The first tests focus on Δ which is an interval-scaled variable. To show differences between means of two or more groups, usually, One-Way-ANOVA tests are applied. ANOVA tests expect independent normally distributed variables and homogeneity of the variances in each group. However both the Kolmogorov–Smirnov test of normal distribution and Levene's test to determine the homogeneity of group variances fail. Although ANOVA is known to be robust enough to run the

tests against non-normal distributed cases and unequal variances, the significance levels of the results are doubtful. Therefore we choose to run a non-parametric test. The Kruskal–Wallis test can be applied even if the data is not normally distributed. However, it has to be run on a slightly modified hypothesis which compares the medians of groups instead of the means. The Kruskal–Wallis test allows to reject both modified hypotheses (asymptotic $p = 0.000$, $\chi^2 = 499636$, $df = 5$).

Concerning E which is a categorical variable, linear models such as ANOVA cannot be used. As described in [132], instead, a binary regression model that turns the mean of E into a binomial distributed probability can be used. Similar to ANOVA, the output variable will be modeled by a *binary logit regression* that models the output using *log linear* values.

By including the main factors I and M we set up a *Generalized Linear Model* (GLM)

$$logit(E) = \text{Intercept} + x_1\text{I} + x_2\text{M}. \tag{7.2}$$

A test of the main effects is statistically significant (χ^2 = 437418, $p <$ 0.000, $df = 6$) so that both null hypotheses (H_1 and H_2) can be rejected. The significance of both effects as well as parameter estimates and Wald values of the calculated model are shown in [276].

The results indicate that there is a significant difference in the error probability for groups of instrumentation counts but also for musicians versus non-musicians. A pairwise comparison test based on the mean differences reveals where these differences are located. Regarding the error probability of different instrument counts, the pairwise comparison test reveals that nearly all groups show significant mean differences between each other, which was the expected result. However, by calculation using the logit GLM model shown in equation 7.2 we found that there are two groups of items of five and six instruments (mean difference 0.04, std. error = 0.019, $df = 1$, $p = 0.055$) that did not show any significant difference. For both groups, the error probability is close to 100%.

To investigate the difference in performance between musicians and non-musicians a pairwise comparison between those two groups was run. Overall musicians perform about 20% better throughout the test (mean difference = 0.18, std. error = 0.0044, $df = 1$, $p = 0.000$). We do not know what caused these differences as the level of professionalism had not been surveyed. Also, 37 % of the musicians additionally had experience in audio engineering due to their profession.

Further, to look at possible interaction effects between the number of instruments and the groups of musicians and non-musicians we adapted our logit equation to

$$logit(E) = \text{Intercept} + x_1\text{I} + x_2\text{M} + x_3\text{M} \times \text{I}. \tag{7.3}$$

We then reran the GLM analysis selecting only items of three and four instruments. This avoids quasi-complete separation in the logit regression model which is caused by low variances in the error probability for items of $I \in \{1, 2, 5, 6\}$. The model effects of the subset can be found in [276].

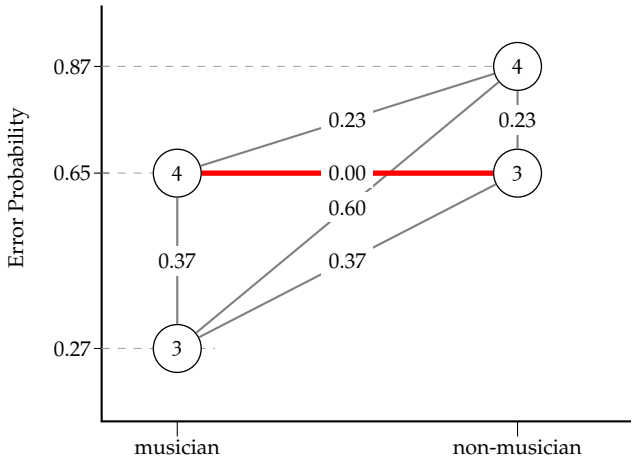


Figure 7.4: Pairwise comparison between the interaction of Musician/Non-Musician and the number of instruments (labeled in nodes). The costs between nodes indicate the mean differences between groups. The red/bold line indicates that is there is no significant difference at the $p = 0.005$ level.

The results indicate that the interaction of Musician×Instruments is not significant on a $p = 0.05$ level in general and a pairwise comparison test reveals two groups of equal probability. The pairwise comparisons are depicted in Figure 7.4. The red vertex indicates there is no significant difference in the error probability for the group of musicians in items with four instruments compared to non-musicians in items of three instruments. Therefore a gap in the error probability of one instrument between those two groups becomes apparent.

This experiment shows that instrument count estimation tasks in music is a difficult task for humans. Our experiment with 62 participants was conducted to address the question of how many instruments one can estimate correctly. The focus was set on stimuli of inhomogeneous timbre and also mixed genre. By comparing musicians to non-musicians, we revealed that there is a significant difference in performance. Particularly this gap is most prominent for items of three and four instruments. Furthermore, for all stimuli (ranging from one to six instruments) we see that musicians performed about 20% better than non-musicians. The experiment shows an assumed upper limit for items with more than three instruments.

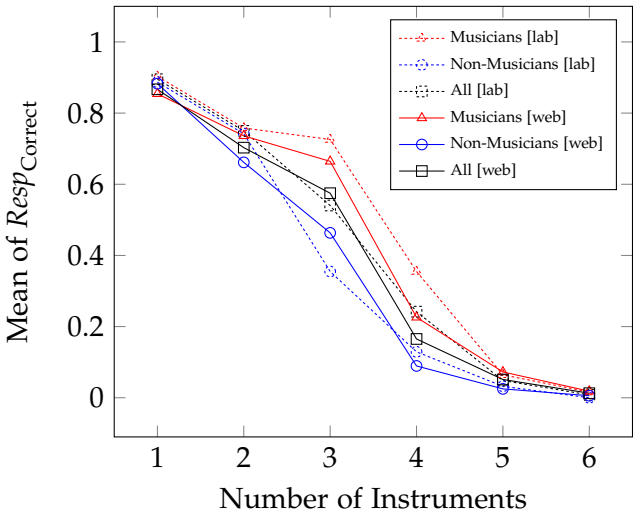


Figure 7.5: Probability of $Resp_{\text{Correct}}$ grouped by Internet experiment (web) and laboratory experiment (lab). Solid lines represent the results of the Internet experiment and dashed lines represents the results of the laboratory experiment. Figure from [254].

7.1.5 Experiment at Larger Scale

Many tasks in auditory experiments such as quality assessment [227], are not suitable for untrained participants, limited resources (low quality headphones, noisy environments) or time constraints. We found that an experimental design such as count estimation, where only a single number is asked to the participant, is an ideal experimental environment to evaluate scale in an uncontrolled environment such as on the web. We therefore designed a follow up study, published in [254] that compares a large scale web experiment to our laboratory results, similar to previous comparisons for other auditory tasks [155, 228, 245, 315].

We used the same stimuli as in the laboratory experiment, however, the training phase was slightly shortened. The experiment took place between February 2013 and April 2013 where participants visited the experimental website[2]. After a screening procedure, described in [254], a total of 1168 valid participants remained.

The main results of the web based experiment in comparison to the previous (lab-bases) experiment is depicted in Figure 7.5. The figure shows the mean probability of correct responses for both environments. The result indicate that there a only very small differences between both experiments. In fact, a detailed analysis in [254] revealed that there are no statistically significant differences between the results of the two experiments.

Further analysis in [254] revealed that "the participants in the laboratory experiment were about 4.6% better in average for all stimuli than

2 http://www.audiolabs-erlangen.com/experiments/wice/

the participants of the Internet experiment. When looking into the differences between musicians and non-musicians, the outcome for the Internet experiment and laboratory experiment differ slightly. In the laboratory experiment musicians performed about 31.6% better than non-musicians and in the Internet experiment musicians performed about 20.85% better." Given a hypothesis that musicians are generally better at performing certain musical related tasks, it indicates that in the (non-anonymous) laboratory experiment, participants that answered to be a musician were, on average, more professional than in the web based experiment.

Finally, the experiment also showed that humans are able to correctly estimate a count even in a very challenging scenario such as unison mixtures and when asked in a not optimal environment. In fact, the results showed that "76% of the participants correctly identified two instruments. Only 18% of the participants underestimated by one instrument, 6% overestimated by one instrument." This surprising outcome then triggered the idea to deepen research on unison instrumental recordings as presented in Chapter 5.

7.2 SPEAKER COUNT ESTIMATION

This section was previously published in [272, 273] and is reprinted, with minor modifications, with permission.

Humans are excellent in segregating one source from a mixture [32] and tend to use this skill to perceptually segregate speakers before they can estimate a count, as highlighted, e.g. in [140]. As shown in [138, 140] with extensive experiments using Japanese speech samples, humans are able to correctly estimate up to three simultaneously active speakers without using any spatial cues. In this experiment, we reproduced the experiments conducted in [138, 140] using stimuli of English speakers. Designed to address source separation research, we also modified the question to ask participants for "the maximum number of concurrent speakers" in a short excerpt of speech.

7.2.1 Stimuli

To date, many available speech datasets contain recordings where only a single speaker is active. Datasets that include overlapped speech segments either lack accurate annotations because the annotation of speech onsets and offsets in mixtures is cumbersome for humans or lack a controlled auditory environment such as in TV/broadcasting scenarios [102]. Since a realistic dataset of fully overlapped speakers is not available, we chose to generate synthetic mixtures. We recognize that in a simulated "cocktail-party" environment, mixtures lack the conversational aspect of human communication but provide a controlled environment which helps to understand how a DNN solves the count estimation problem. As we aim for a speaker independent solu-

tion, we selected a speech corpus with preference to a high number of different speakers instead of the number of utterances, thus increasing the number of unique mixtures. We selected *LibriSpeech clean-360* [203] which includes 363 hours of clean speech of English utterances from 921 speakers (439 female and 482 male speakers) sampled at 16 kHz.

To compute the maximum number of concurrent speakers k, annotation of the activity of each individual speaker is required. Even though many corpora come with word and phonemes annotation, they often are not consistent across different corpora. We, therefore, generated annotations based on a voice activity detection algorithm (VAD). As we rely on a robust VAD estimate, we found the implementation from the *Chromium Web Browser* as part of the WebRTC Standard[3] to yield good results.

To generate a single example, a tuple of a speech mixture and its ground truth speaker count k, we draw a unique set of k speakers from the corpus. For each of the speakers, we then select a random utterance, resampled to 16 kHz sampling rate and apply VAD. The VAD method was configured using default parameters using a hop size of 10 ms. Further, the VAD estimate was used to remove silence from the beginning and the end of an utterance recording. In the next step, more utterances from the same speaker are drawn from the corpus until the desired duration is reached. We removed silence in the beginning and end of each utterance to increase the overlap within one segment. Both, the audio recording and the VAD annotation of each utterance is concatenated. The procedure is repeated for all speakers such that k time domain signals are created. Signals are linearly mixed and peak normalized to avoid clipping. The ground truth output k for each sample is then computed from the VAD matrix using the maximum of the sum over all speakers.

In fact, our method to generate synthetic samples results in an average overlap of 85% for $k = 2$ and of 55% for $k = 10$ (based on 5s segments). This procedure is similar to [188] used to label the data. The dataset is available for download [281].

7.2.2 Experimental Setup

We conducted a study using the simulated data from the *LIBRI Count Dataset* as mentioned in the previous subsection. In turn, we randomly selected 10 samples for each $k \in 1, \ldots, 10$, resulting in 100 mixtures of 5 seconds duration each. The stimuli were presented in random order using a custom web-based interface (depicted in Figure 7.6) connected to a database API that saved the anonymized count responses and additional information about the participant session such as the response time. The experiment was done using *between-group design*,

3 WebRTC 1.0: Real-time Communication Between Browsers W3C Editor's Draft 05 June 2017

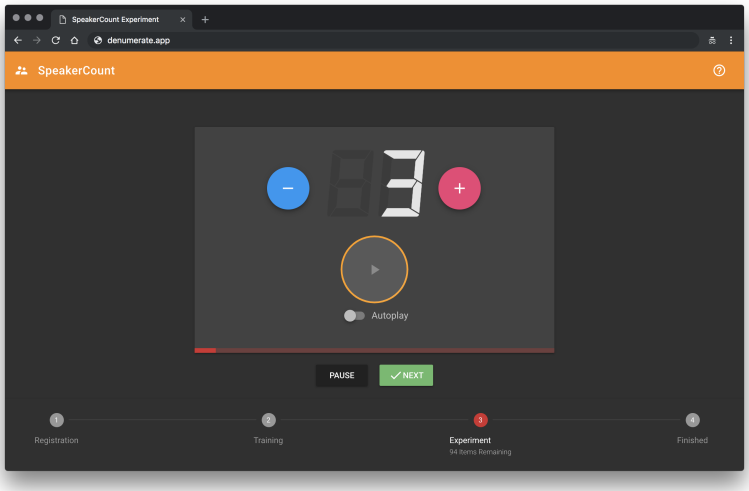


Figure 7.6: Speaker count estimation experiment user interface.

where one group (blind experiment) did not get any prior information about the maximum number of speakers in the test set (similar to [140]). However, the maximum number of speakers was revealed to the other group (informed experiment). Further, none of the participants received any feedback about the error made during the trials. The participants were able to pause and resume the experiment at any time to reduce fatigue. Similarly to [140], lab-based experiments were conducted with ten participants for each group ($n = 20$) using a custom designed web-based software. None of the participants were native English speakers. The experiment and its results from all participants is made available through [280]. A simplified version of the experiment is made available through a web application[4].

7.2.3 Experiment Results

To reveal over- and underestimation errors, we decided to report the average response for each k. As a reference, we also included the average results from [140, Experiment 1, 5 seconds duration] which shows similar (with slightly higher error probability) results compared to our blind experiment. Also, in [140] the maximum number of speakers to test was six whereas we evaluated stimuli with up to ten speakers. The results of our lab-based experiments are shown in Figure 7.7 and Figure 7.8. Results indicate that underestimation becomes apparent for $k > 3$. First and foremost, we can confirm the "one-two-three-many" paradigm on our experiment with English utterances. When we asked participants about the strategy they pursued, many reported that with more than three speakers it is not possible to identify (and count) the speakers but rather compare the *density* of the speech to that of 1-3

4 https://denumerate.app

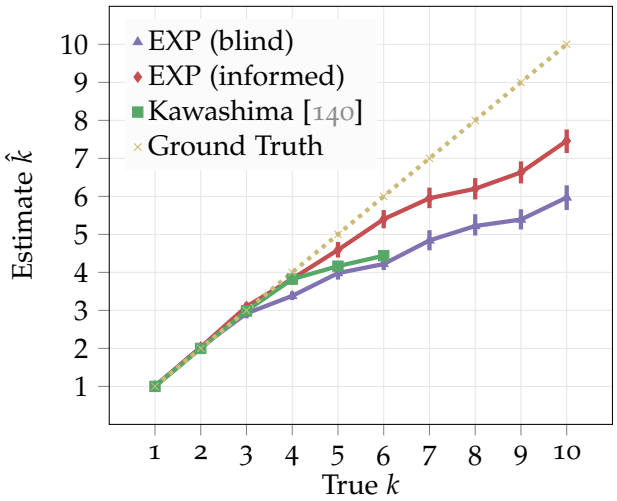


Figure 7.7: Average responses in comparison to the results of *Kawashima* [140]. Error bars show 95% confidence intervals (not available for [140]). ® 2019 IEEE.

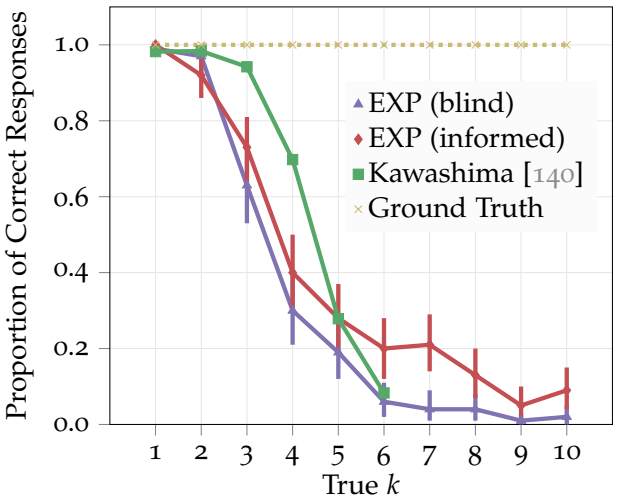


Figure 7.8: Mean probability of correct responses in comparison to the results of *Kawashima* [140]. Error bars show 95% confidence intervals (not available for [140]). ® 2019 IEEE.

speakers. For higher speaker counts, participants reported that the phoneme density was a relevant cue that allowed them to extrapolate a source count estimate. Interestingly, our results of the informed experiment reveal that they performed significantly better than those that participated blindly. This is especially obvious for six and more speakers where the informed group performed better by more than one speaker in mean absolute deviation.

7.3 SUMMARY AND DISCUSSION

In this chapter, we presented two experiments that we conducted to get a better understanding of the human ability to estimate the number of sources in overlapped mixtures.

First, we showed that estimating the number of instruments in music mixtures is a challenging task for humans. We presented the results of a controlled experiment with 62 participants. Our experiment indicated that the upper limit is reached with more than three instruments, related to an earlier experiment [131] focused on voices instead of instruments. In Chapter 2, two different questions were introduced how to frame the count estimation problem. In an experiment, we explicitly used an open question of "How many instruments can you hear" to gather more knowledge into the strategies being applied by the participants. Another reason was that the stimuli durations were too long to ask for the maximum number of concurrent stimuli, thus promoting "counting by detection" as a good strategy to approach the problem.

By comparing musicians to non-musicians, we showed that there is a significant difference in performance for count estimation, confirming similar findings in other auditory tasks [146]. Particularly, we found out that this gap is most prominent for stimuli of three and four instruments. Furthermore, for all stimuli (ranging from one to six instruments) we revealed that musicians performed about 20% better than non-musicians, hence revealing a "gap of one instrument" in mean absolute error.

We then repeated this experiment in an open, uncontrolled environment with more than 1000 participants. To our knowledge, this was the first larger crowdsourced auditory experiment within the Signal Processing or MIR community. In comparison to audio quality experiments like MUSHRA [227], we can, therefore, conclude that count estimation tasks are suitable for highly scalable crowd sourced listening experiments.

In our second experiment, we reproduced an earlier study presented in [140] to estimate the maximum number of concurrent speakers in short audio mixtures, simulating a "cocktail-party" environment. Our experiment went a step further with respect to the maximum amount of speakers to be estimated (up to $k = 10$). The results indicated almost no person was able to correctly estimate up to ten speakers. However, we also observed that even for more than seven speakers the mean response did further (but not linearly) increase. This indicates that humans are possibly interpolating some sort of a speech density to make up their decision.

We conclude that all of our count experiments share a common outcome: A) humans are unable to correctly estimate more than four sources and B), underestimation is the main cause of error.

8 COUNTNET: DATA-DRIVEN SPEAKER COUNT ESTIMATION

This chapter was previously published in [272, 273] and is reprinted, with minor modifications, with permission (®2019 IEEE).

In a "cocktail-party" scenario, one or more microphones capture the signal from many concurrent speakers. In this setting, different applications may be envisioned such as localization, crowd monitoring, surveillance, speech recognition, speaker separation, etc. When devising a system for such a task, it is typically assumed that the actual number of concurrent speakers is known. This assumption turns out to be of paramount importance for the effectiveness of subsequent processing. Notably, for separation algorithms [57], real-world systems do not straightforwardly provide information about the actual number of concurrent speakers. It, therefore, is desirable to close the gap between theory and practice by devising reliable methods to estimate the number of sound sources in realistic environments. Surprisingly, very few methods exist for this purpose in an audio context, in particular from a single microphone recording.

From a theoretical perspective, estimating the number of concurrent speakers is closely related to the more difficult problem of *identifying* them, which is the topic of speaker diarization [7, 223, 234, 235]. Intuitively, if a system is able to tell who speaks when, it is naturally also able to tell how many speakers are actually active in a mixture. We call this strategy "counting by detection". A good working diarization system would be able to sufficiently address the speaker count estimation problem using this strategy. However, it appears to be a very complex problem to tackle when one is only interested in the number of concurrent speakers. Furthermore, as current diarization systems only work when a clear segmentation is possible, the first step of such a system often is to find homogeneous segments in the audio where only one speaker is active. The segment borders can be found by speaker change detection [324]. These homogeneous segments are used to discriminate and temporally locate the speakers within a given recording. When sources are simultaneously active, as in real cocktail party environments, existing segmentation strategies fail. In fact, overlapping speech segments typically are a major source of error in speaker diarization [7].

To improve the robustness of these detection-based methods, a number of approaches attempt to detect and possibly reject the overlapping speech segments to improve performance [26, 130]. Overlap detection has since evolved into its own line of research with many recent publications such as [5, 92, 261]. Overlap detection can be seen as a binarized version of the count estimation problem where the number of speakers equals to one (*no overlap*) or more than one (*overlap*).

It is, therefore, possible to apply a count estimation system for the overlap detection problem but not vice versa. Also, an overlap detection system cannot be easily utilized in a source separation system. In fact, it should be noted that before the arrival of deep learning based separation systems, models required long context and in such a case, for methods like NMF, the number of concurrent speakers could be introduced as a regularization term [156]. In recent years, however, large improvements were achieved by deep learning based methods [116, 325] at shorter segment duration (often 1-5 seconds). In such approaches, it becomes possible to apply separation only when its "needed". In this scenario, a method of estimating the maximum number of concurrent speakers becomes useful and in some cases essential.

When speaker overlap is as prevalent as in a "cocktail-party" scenario, developing an algorithm to detect the number of speakers is challenging. Since there are indications that the auditory system is capable of subitizing sources [120], we transfer this fact to the audio domain and directly attempt in this study to estimate the number of audio sources through "direct count estimation" (see also Chapter 7). The question if machines could outperform humans, or if they are subject to similar limitations, remains to be answered and is also part of this work.

Directly estimating the number of sources in audio mixtures has many applications and appears as a reasonable objective that mimics the process of human perception. Since humans do have two ears that provide spatial diversity, a first natural idea to imitate human performance is to exploit *binaural* information to proceed to source count estimation. In terms of signal processing, this is achieved by estimating directions of arrival (DoA) and clustering them [9, 10, 71, 173, 191, 205, 206, 309]. However, many audio devices are equipped with only a single microphone, and being able to also count sources, in that case, is desirable. Consequently, the single-channel scenario has been considered in many studies.

One of the first single-channel methods was proposed in 2003 by Arai [8]. It is based on the assumption that speech mixed from more than one speaker has a more complex amplitude modulation pattern than a single speaker. The modulation pattern is aggregated and used as a decision function to distinguish between different numbers of speakers. In [247], the authors propose an energy feature based on temporally averaged mel filter outputs. The number of concurrent speakers was determined by manually determining thresholds that best match individual speaker counts. In a more recent work, Xu et.al. [320] estimate the number of speakers by applying hierarchical clustering to fixed-length audio segments using mel frequency cepstral coefficients (MFCCs) and additional pitch features. The method assumes the presence of at least some non-overlapped speech and

was evaluated on real-world data of 20 hours duration. An average count estimation error of one speaker is reported using excerpts of eight-minutes duration and featuring up to eight speakers. In another vein, Andrei et.al. [4, 6] proposed an algorithm which correlates single frames of multi-speaker mixtures with a set of single-speaker utterances. The resulting score was then used to estimate the number of speakers using thresholds.

In all the aforementioned methods, the speaker count estimation problem was devised. The different strategies undertaken there rely on classical and grounded signal processing strategies and exhibit fair performance in a controlled setup. However, our experience shows (see Section 8.4) that they leave much room for improvement when applied to more diverse and challenging signals than those corresponding to their targeted applications, notably in the case of many different and constantly overlapping speakers. This is due to their main common weakness, which is to rely on the assumption that there are segments where only one speaker is active, in a way that is similar to the classical speaker diarization studies mentioned before. In [271] we presented a first data-driven approach based on a recurrent network, motivated by the recent and impressive successes of deep learning approaches in audio tasks such as speech separation [99, 116, 325] and speaker diarization [89, 122, 321]. The methods proposed in [271] to address speaker count estimation were built upon recent methods to count objects in images, which is a popular application with many contributions from the deep learning community [11, 29, 46, 141, 179, 258, 312, 326, 327]. In [271] two main paradigms were evaluated: a) count estimation as regression problem, where the systems are directly trained to output the number of objects as a point estimate, and b) classification, where every possible number of objects is encoded as a different class and the output of a predicting system corresponds to a probability distribution over these classes. The results of the proposed method indicated that a classification based neural network performed better than one based on regression. One drawback, however, is that the maximum number of speakers (the number of classes) is known in advance.

In this study, we build upon [271] and focus on the network architecture design, as well as on finding limitations for different test scenarios. This work makes the following contributions: i) we generalize the problem formulation by fusing classification and regression, which allows estimating discrete outputs while controlling the error term. This is done by picking a point estimate from a full posterior distribution provided by the deep architectures; ii) in addition to the recurrent network introduced in [271], we propose alternative speaker-independent neural network architectures based on the convolution operation to improve count estimation. Each of the proposed networks is adjusted to estimate the number of speakers from audio segments

of 5 seconds; iii) we test the performance of these networks in multiple experiments and compare them to several baseline methods, pointing out possible limitations. Furthermore, we present a statistical analysis of the results to determine whether classification outperforms regression for all architectures; iv) we conducted a listening experiment to relate the best-performing machine to human performance. We describe one of the strategies taken by the data-driven approach that might explain its superior performance. Finally, for the sake of reproducibility, the trained networks (models), as well as the test dataset, are made available on the accompanying website[1].

8.1 PROBLEM FORMULATION

We consider the task of estimating the maximum number of concurrent speakers $k \in \mathbb{Z}_0^+$ in a single-channel audio mixture $\mathbf{x}$. This is achieved by applying a mapping from $\mathbf{x}$ to k. We now provide details on the notations, the general structure of the method, and ways to exploit the deep learning framework to estimate k.

8.1.1 Signal Model

Let $\mathbf{x}$ be a time domain vector with N samples, representing a linear mixture of L single speaker speech signal vectors $\mathbf{s}_l$. The value observed at time instant n for the mixture is given by x_n and for the individual speech segments by s_{nl}. The mixture then results in

$$x_n = \sum_{l=1}^{L} s_{nl} \; n \in \{1, \ldots, N\}. \tag{8.1}$$

Naturally, each speaker $l = 1, \ldots, L$ is not active at every time instant. On the contrary, we assume there is a latent binary *speech activity* variable $v_{nl} \in \{0, 1\}$ that is either provided by a ground truth annotation or computed using a voice activity detection method.

Our objective of estimating the maximum number of concurrent speakers can now be formulated as

$$k = \max_n \left(\sum_{l=1}^{L} v_{nl} \right) \; n \in \{1, \ldots, N\}. \tag{8.2}$$

As can be seen, our proposed task of estimating $k \leq L$, is more closely related to source separation whereas the estimation of L is more useful for tasks where speakers do not overlap (see Section 2.3.3). For instance, three non-overlapping speakers would result in $L = 3$ and $k = 1$. It should be noted that at short time scales both task

1 https://www.audiolabs-erlangen.de/resources/2017-CountNet.

definitions provide the same outcome because on such a time scale the speaker configuration usually does not change. The problem arises for long-term recordings (e.g. larger than ten seconds) which are not considered in this work. In any case, we want to emphasize that in all experiments presented in this Chapter, we made sure that for all audio segments $k = L$.

In the remainder of this chapter, we assume that no additional prior information about the speakers is given to the system except possibly the maximum number of concurrent speakers $k_{\max}$, that is application-dependent and represents an upper bound for the estimation.

While speaker diarization would mean estimating the whole speech activity matrix v_{nl}, our problem of estimating only k in (8.2) is more abstract as it requires a direct estimation of the count.

By processing such excerpts in a sliding-window fashion, our proposed solution can be applied straightforwardly to context sizes commonly used in source separation. Furthermore, our proposed system can be used also to detect overlap ($k > 1$), which can be useful as a pre-processing step for diarization.

Now, the system we propose is actually not inputting the signal vector $\mathbf{x}$, but rather a Time-Frequency (TF) representation as the absolute value of the short-time Fourier transform of $\mathbf{x}$ that is denoted by $\mathbf{X}$. In the following, $\mathbf{X}$ is the non-negative input for the system.

8.1.2 Probabilistic Formulation

In a supervised scenario, let $\{\mathbf{X}_t, k_t\}_t$ be all of our learning examples, where $t \in 1, \ldots, T$ denotes the t-th training item from the training database. For the purpose of learning a mapping between $\mathbf{X}$ and k, we adopt a probabilistic viewpoint and introduce a flexible generative model that explains how a particular source count k corresponds to some given input $\mathbf{X}$.

First, we consider that all training samples $\{\mathbf{X}_t, k_t\}_t$ are independent. For each sample, we consider that k_t is drawn from a probability distribution of a known parametric family, parameterized by some latent and unobserved parameters $\mathbf{y}_t$

$$\mathbb{P}\left(k_t \mid \mathbf{X}_t\right) = \mathcal{L}\left(k_t \mid \mathbf{y}_t\right), \tag{8.3}$$

the distribution $\mathcal{L}\left(\cdot \mid \mathbf{y}_t\right)$ is called the *output distribution* in the following. We further assume that there is some deterministic mapping between $\mathbf{X}_t$ and $\mathbf{y}_t$, embodied as

$$\mathbf{y}_t = f_{\boldsymbol{\theta}}\left(\mathbf{X}_t\right), \tag{8.4}$$

where $\boldsymbol{\theta}$ are the parameters for this deterministic mapping, that is independent of the training item t. This results in an output distribution given by

$$\mathbb{P}\left(k_t \mid \mathbf{X}_t\right) = \mathcal{L}\left(k_t \mid f_{\boldsymbol{\theta}}\left(\mathbf{X}_t\right)\right). \tag{8.5}$$

Assume for the rest of this section that these parameters $\boldsymbol{\theta}$ are known. Given a previously unseen input $\mathbf{X}$, expression (8.5) means we can compute the distribution of the source count k. The objective of our count estimation system is to produce a point estimate $\hat{k}$ rather than a whole output distribution $\mathbb{P}\left(k \mid \mathbf{X}\right)$. A first option is to pick as an estimate the most likely outcome for the output distribution, thus resorting to Maximum A Posteriori (MAP) estimation:

$$\hat{k} = \underset{k}{\operatorname{argmax}}\, \mathcal{L}\left(k \mid f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right). \tag{8.6}$$

However, MAP is not the only option and a broad range of point estimation techniques may be obtained when resorting to decision theory [22]. We may for example also choose $\hat{k}$ as the value that minimizes the marginal average cost of choosing an estimate $\hat{k}$ instead of the true value k, when k is distributed with respect to the output distribution

$$\hat{k} = \underset{u}{\operatorname{argmin}} \int_k d\left(k, u\right) \mathcal{L}\left(k \mid f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right) \mathrm{d}k, \tag{8.7}$$

where $d\left(k, u\right)$ is the cost of picking u as an estimate when the true value is k. It may be any function that seems appropriate, and does not necessarily need to be differentiable. However, we retain the more general formulation (8.7) because other choices will sometimes prove more effective, as we show later. For notational convenience, we write (8.7) as

$$\hat{k} = q\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{X}\right)\right), \tag{8.8}$$

and $q\left(\cdot\right)$ is called the *decision function*. Using this strategy, we have everything to produce a single source count estimate $\hat{k}$ from input features $\mathbf{X}$, provided the parametric family $\mathcal{L}$ and the mapping $f_{\boldsymbol{\theta}}$ as well as its parameters $\boldsymbol{\theta}$ are known. In this study, we choose a deep neural network for the mapping $f_{\boldsymbol{\theta}}$, whose weights $\boldsymbol{\theta}$ are trained in a supervised manner (compare Figure 8.1). Once a particular network architecture has been chosen, learning its parameters is achieved through classical stochastic gradient descent. If we assume that the particular family $\mathcal{L}$ of output distributions has been chosen, it appears natural to learn the parameters $\boldsymbol{\theta}$ that maximize the likelihood of the learning data. More specifically, the total cost to be minimized becomes

$$C = \sum_{t=1}^{T} -\log \mathcal{L}\left(k_t \mid f_{\theta}\left(\boldsymbol{X}_t\right)\right). \tag{8.9}$$

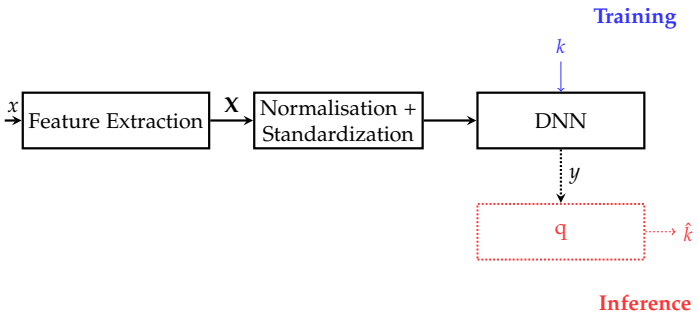


Figure 8.1: Block diagram of the proposes supervised learning model. Training is realised using tuples of spectro-temporal inputs $\mathbf{X}$ and the true number of concurrent speakers k. For inference the output y is post-processed using a decision function q to generate estimates $\hat{k}$. ® 2019 IEEE.

The derivative of this cost (8.9) with respect to the parameters can be used to learn the network parameters.

Three different choices for the family of output distributions (classification, Gaussian regression and Poisson regression) are summarized below.

Classification

In a classification setting, the output distribution is directly taken as *discrete*, discarding any meaning concerning the ordering of the different possible values. Given some particular input $\mathbf{X}$, the network generates the posterior output probability for $(k_{\max} + 1)$ classes (including $k = 0$) and a maximum a posteriori (MAP) decision function is chosen that simply picks the most likely class $q = \arg\max(\cdot)$. Classification based approaches have successfully been applied in deep neural networks for estimating counts in objects [141, 258, 327] in images.

Gaussian Regression

In regression, k is derived from an output distribution defined on the real line. However, this comes with the additional difficulty of handling the fact that k is integer.

The output distribution in this setting is assumed to be Gaussian and the associated cost function is the classical squared error. During inference and given the output $f_{\boldsymbol{\theta}}(\mathbf{X})$ of the network, the best discrete value that is consistent with the model is simply the rounding operator $q = [\cdot]$.

Gaussian regression has achieved state-of-the-art count estimation performance in computer vision using deep learning frameworks [29, 179, 326].

Discrete Poisson modeling

When it comes to modeling count data, it is often shown effective to adopt the Poisson distribution [81]. First, this strategy retains the advantage of the classification approach to directly pick a probabilistic model over the actual discrete observations, avoiding the somewhat artificial trick of introducing a latent variable that would be rounded to yield the observation. Second, the model avoids the inconvenience of the classification approach to completely drop dependencies between classes.

Due to these advantages, the Poisson distribution has been used in studies devising deep architectures for count estimation systems [230]. For instance in [44, 81, 230], it is shown that the number of objects in images can be well modeled by the Poisson distribution. Inspired by these previous works, we also consider the Poisson output distribution $\mathcal{P}\left(k \mid f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right)$ where $\mathcal{P}\left(\cdot \mid \lambda\right)$ denotes the Poisson distribution with scale parameter λ.

In that setup, the cost function at learning time is the Poisson negative log-likelihood and the deep architecture at test time provides the predicted scale parameter $f_{\boldsymbol{\theta}}\left(\mathbf{X}\right) \in \mathbb{R}_{+}$, which summarizes the whole output distribution.

As a decision function q in this setting, we considered several alternatives. A first option is to again resort to MAP estimation and pick the mode $\left[f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right]$ of the distribution as a point estimate. However, experiments showed that the posterior median yields better estimates, and is given by

$$q\left(f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right) = \operatorname*{argmin}_{\hat{k}} \sum_{k=0}^{\infty} \left|\hat{k} - k\right| \mathcal{P}\left(k \mid f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right) \tag{8.10a}$$

$$= \operatorname{median}\left(k \sim \mathcal{P}\left(f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)\right)\right) \tag{8.10b}$$

$$\approx \left\lfloor f_{\boldsymbol{\theta}}\left(\mathbf{X}\right) + \frac{1}{3} - \frac{0.02}{f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)} \right\rfloor, \tag{8.10c}$$

where the last expression is an approximation of the median of a Poisson distributed random variable of scale parameter $f_{\boldsymbol{\theta}}\left(\mathbf{X}\right)$ [51].

8.2 DNNS FOR COUNT ESTIMATION

Applying deep learning to an existing task often is a matter of choosing a suitable network architecture. Typically an architecture describes the overall structure of the network including (but not limited to) the type and number of layers in the network and how these layers are connected to each other. In turn, designing such an architecture requires deep knowledge about input and output representations and their required level of abstraction. Many audio-related applications like speech recognition [117] or speaker diarization share similar

architectural structures, often found by incorporating domain knowledge and through extensive hyper-parameter searches. For our task of source count estimation, however, domain knowledge is difficult to incorporate, as our studies aim at revealing the best strategy to address the problem. This is why we chose architectures that already have shown a good level of generalizability for audio applications.

8.2.1 Network Architectures

The input of all networks is a batch of samples, represented as time-frequency representations $\mathbf{X} \in \mathbb{R}^{D \times F \times C}$, where D refers to the time dimension, F to the frequency dimension and C to the channel dimension (in the single-channel case, $C = 1$). In the following, we discuss several commonly used DNN architectures and their benefits in using them for the task of estimating the number of speakers. All architectures under investigation are summarized in Fig. 8.2.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a variant of standard fully-connected neural networks, where the architecture generally consists of one or more "convolution layers" followed by fully-connected layers leading to the output.

A convolutional layer consists of a convolution operation, followed by feature pooling. The convolution operation applies a set of filters to local regions of the input, and the application of each such filter outputs a *feature map*. It should be noted that the convolution operation, generally, also constitutes the application of a point-wise non-linear activation function on each feature map. This is followed by feature pooling, that aims to reduce the feature space dimensions by combining the filter activations over a specified region. Since the individual elements of the filters (weights) are learned during the training stage, convolutional layers can also be interpreted as feature extractors. By stacking up additional layers, CNNs can extract more abstract features in higher level layers [263].

The sizes of the filter kernels are crucial, and it was shown in [214] that many audio applications can benefit if domain knowledge is put into the design of the filter kernel size. The use of small filter kernels, as often used in image classification tasks, does not necessarily decrease performance, when combined with many layers. Also larger kernels increase the number of parameters and therefore the computational complexity. It was shown in [252] that 3×3 kernels resulted in state-of-the-art results in singing voice detection tasks. Due to its hierarchical architecture, CNNs with small filters have the benefit that they can model time and frequency invariances regardless of the scaling of the frequency axis.

Our proposed architecture is similar to the ones proposed by [251] used for singing voice activity detection. In our proposed CNN, we consider local filters of size 3×3. In the first layer, 2D convolution is performed by moving the filter across both dimensions of the input in steps of 1 element (striding $s = 1$ to generate $C = 64$ feature maps/channels resulting in an output volume of $64 \times (D - 3 + 1) \times (F - 3 + 1)$. In the subsequent convolution layers, a similar operation is applied but for each convolutional layer, we consider a different number of feature maps. Note, that the convolution operation is performed independently for every input channel, and then summed up along the dimension C for each output element. In preliminary experiments we found that by using max-pooling we received significantly better performance when used after CNN layers.

Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) layer is very similar to a fully connected network, except that RNN applies the same set of weights $\mathbf{A}$ recursively over an input sequence. While convolutional layers excel in capturing local structures, RNNs can detect structure in sequential data of arbitrary length. This makes it ideal to model time series, however, in practice, the temporal context learnt is limited to only a few time instances, because of the vanishing gradient problem [118].

To alleviate this problem, forgetting factors (also called gating) were proposed. One of the most popular gated recurrent cells is the Long Short-Term Memory (LSTM) [119] cell. Its effectiveness has been proven in applications and LSTMs are the state-of-the-art approach for speech recognition [101] and singing voice detection [157] [2]. For a given input of dimensions $D \times F \times C$, the output of a recurrent layer is either only the last step of dimension $1 \times A$ or the full sequence $D \times A$. The latter is useful to stack multiple LSTMs or to apply temporal max pooling of the sequence. In [271] such an architecture based on three bi-directional LSTM cells, was proposed. The architecture is similar to the one employed in [157].

Convolutional Recurrent Neural Networks (CRNN)

Recently, a combination of convolutional and recurrent layers were proposed for audio-related tasks [3, 37, 52, 240].

The main motivation to stack these layers is to combine the benefits of convolutional layers with those of recurrent architectures, namely the benefit of convolutional layers in aggregating local features with the ability of recurrent layers to model long-term temporal data.

There are different ways to stack CNNs and RNNs to form a CRNN architecture. In our application the motivation is to aggregate local

2 For a deeper mathematical background of LSTMs, due to space constraints, the reader is referred to the aforementioned papers.

Layer	Parameters	Value Range
CNN 1	Feature Maps	$\{16, \mathbf{32}, 64\}$
CNN 1	Filter Length	$\{\mathbf{3}, 5, 7\}$
Pooling 1	Pooling Length	$\{1, \mathbf{2}, 4\}$
CNN2	Feature Maps	$\{16, \mathbf{32}, 64\}$
CNN2	Filter Length	$\{\mathbf{3}, 5, 7\}$
Pooling 2	Pooling Length	$\{1, 2, 4\}$
CNN 3	Presence of Layer	$\{\mathbf{Yes}, No\}$
CNN 3	Feature Maps	$\{16, 32, \mathbf{64}, 128\}$
CNN 3	Filter Length	$\{\mathbf{3}, 5, 7\}$
Pooling 3	Pooling Length	$\{1, \mathbf{2}, 4\}$
Fully Connected 1	Hidden Unit	$\{64, \mathbf{128}\}$
Dropout 1	Dropout Percentage	$[0.1, \mathbf{0.2}, 0.5]$
Fully Connected 2	Hidden Unit	$\{32, \mathbf{48}\}$
Dropout 2	Dropout Percentage	$[0.1, \mathbf{0.2}, 0.5]$

Table 8.1: Parameter optimization of F-CNN model through hyper-parameter search. Bold hyper-parameters were found optimal.

time-frequency features coming from the output convolutional neural network and use the LSTM layer to model long temporal structures. As the output of a CNN layer is a 3D volume $D \times F \times C$ and the input of a recurrent layer only takes a 2D sequence, the dimension would need to be reduced. Naturally, the time dimension would need to be kept, therefore the channel dimension C is stacked with the frequency dimension F resulting in a $D \times F \cdot C$ output.

Full-band Convolutional Neural Networks (F-CNN)

Architectures where filters span the full frequency range and therefore apply convolution in temporal direction only, have already been successfully deployed in speech [3] and music application [52, 65, 213]). Our motivation here is that the activity of speakers happen over wide frequency ranges and a count (unlike in counting objects in images) cannot be split into sub counts. The full-band kernel configuration only affects the first hidden layer, as in consecutive outputs all frequency bands are squashed down to one single frequency band using "valid" convolutions. This is computationally very efficient, because it reduces the middle layer's dimensionality of the network significantly due to this aggregation. To further optimize the performance of the network, we applied a hyper-parameter optimization technique using Tree-structured Parzen Estimator (TPE) [23]. We used a search space of several hyper-parameters as shown in Table 8.1 and set the maximum number of evaluations to 200.

The results are in agreement with the findings in [251] where small filter kernels of size 3 outperformed larger kernels. Also, it can be seen

from the results, that increasing the number of feature maps of the convolutional layers does not necessarily increase the performance.

Full-Band Convolutional Recurrent Neural Networks (F-CRNN)

Similarly to *CRNN* and to the Deep Speech 2 implementation [3], we added an LSTM recurrent layer to the output of the last convolutional layer. Since each filter output is only of dimension one, an additional flattening as in *CRNN* is not required.

8.2.2 Output Activation Functions for Count Estimation

For each of the decision functions a suitable output activation and loss is used.

Classification

For *classification*, the output is required to be one-hot-encoded so that the output is of dimension $y \in \mathbb{B}^{L+1}$, where L is the maximum number of concurrent speakers to be expected. In the final layer of the network, a softmax activation function is used with the cross-entropy function as the loss.

Gaussian Regression

For the Gaussian regression model, the final output layer is of dimension $y \in \mathbb{R}^1$. The output layer nodes have linear activation, and mean squared error is used as the loss function.

Poisson Regression

For the Poisson regression, the likelihood of parameter λ given the true count k is computed by the negative log-likelihood loss $E = \sum \lambda - k * \log(\lambda + eps)$. The output layer activation is the exponential function.

8.2.3 Speech Corpora and Annotations

To date, many available speech datasets contain recordings where only a single speaker is active. Datasets that include overlapped speech segments, either lack accurate annotations because the annotation of speech onsets and offsets in mixtures is cumbersome for humans or lack a controlled auditory environment such as in TV/broadcasting scenarios [102]. Since a realistic dataset of fully overlapped speakers is not available, we chose to generate synthetic mixtures. We recognize that in a simulated "cocktail-party" environment, mixtures lack the conversational aspect of human communication but provide a controlled environment which helps to understand how a DNN solves the

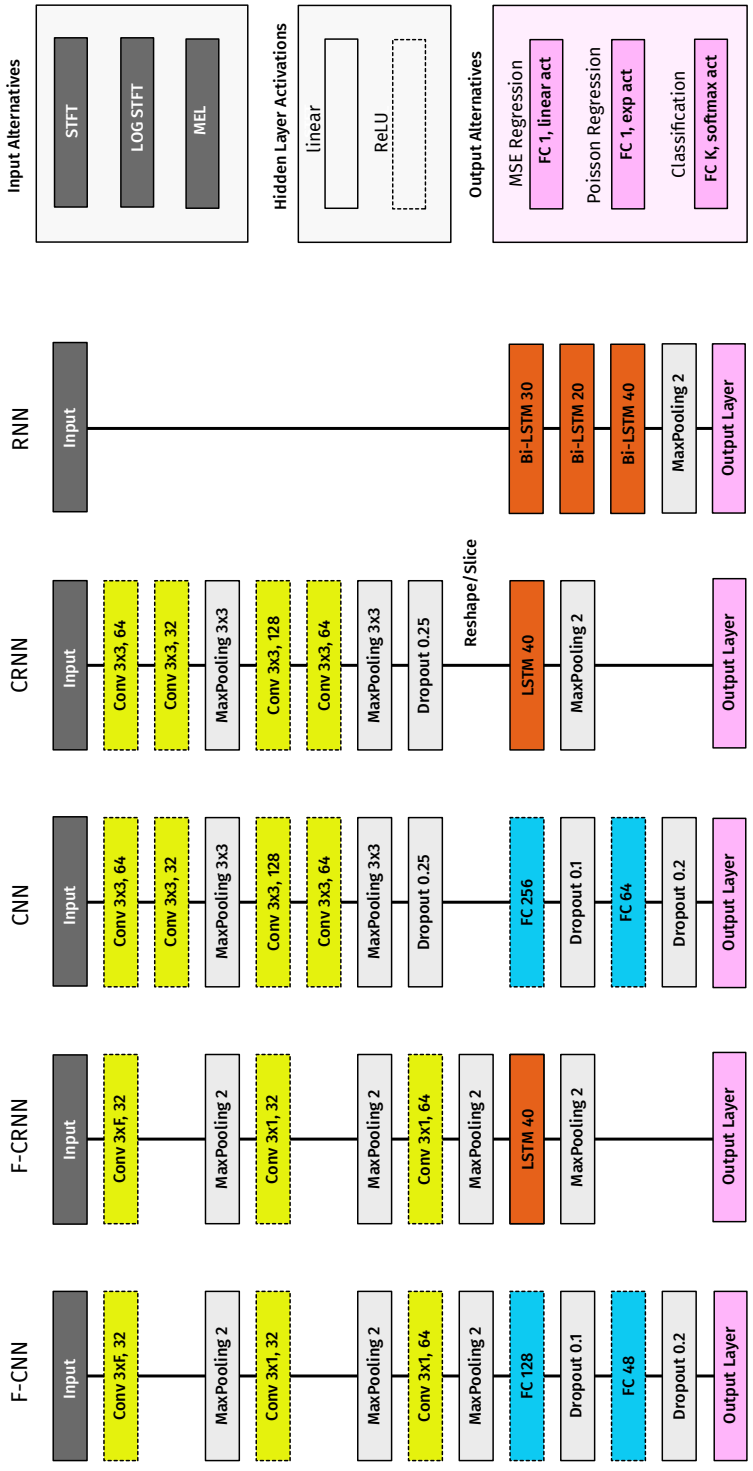


Figure 8.2: Overview of the proposed Architectures. ® 2019 IEEE.

Table 8.2: Overview of speech corpora used in this work.

		Number of Speakers		
Name	Language	Train	Valid.	Test
LibriSpeech [203]	English	921	40	40
TIMIT [90]	English	462	24	168
THCHS [313]	Mandarin	30	10	10

count estimation problem. As we aim for a speaker independent solution, we selected a speech corpus with preference to a high number of different speakers instead of the number of utterances, thus increasing the number of unique mixtures. We selected *LibriSpeech clean-360* [203] which includes 363 hours of clean speech of English utterances from 921 speakers (439 female and 482 male speakers) sampled at 16 kHz.

In the further course of this work (see Section 8.4), we also present the results from test sets of two other datasets as listed in Table 8.2. Furthermore, we included non-speech examples from the TUT Acoustic Scenes dataset [187] in our training data to avoid using zero input samples for $k = 0$ to increase the robustness against noise.

A single training tuple $\{\mathbf{X}, k\}$ is generated by a synthetic speech mixture and their ground truth speaker count k. The mixtures were generated as described in Section 7.2.1. In fact, our method to generate synthetic samples results in an average overlap for $k = 2$ of 85% and for $k = 10$ of 55% (based on 5s segments). This procedure is similar to [188] used to label the data. Signals are mixed according to (8.1), peak normalized and then transformed to a time-frequency matrix $X \in D \times F$. Based the voice activity detection algorithm (VAD), we computed the ground truth output k via (8.2). All samples are normalized to the average Euclidean norm of *duration* frames to be robust against gain variations as proposed by [293]. Furthermore, the data was scaled to zero mean and unit standard deviation across the frequency dimension F over the full training data. Scaling parameters were saved for validation and test. For a more detailed description of the dataset, the reader is referred to [271].

8.2.4 Training Procedure

For all experiments we chose a medium sized training dataset of $k \in \{0, \ldots, 10\}$ forming a total of $T_{\text{train}} = 20.020$ mixtures (1820 per k), each containing 10 seconds of audio, resulting in 55.55 hours of training material. For each sample fed into the network, we select a random excerpt of duration D from each mixture. If not stated otherwise, $D = 5$ seconds. That way, for each epoch, the network is seeing slightly different samples, reducing the number of redundant samples and thus helping to speed up the stochastic gradient based

References	Task	Representation
[92, 108]	Overlap Detection/VAD	MFCC
[101, 177, 240]	ASR	MEL
[3]	ASR	STFT
[251, 252]	Singing Voice AD	$\log(1+\mathbf{X})$

Table 8.3: Speech related input representations in related work.

training process[3]. A similar training procedure is detailed in [251, 271]. Each architecture is trained using the ADAM optimizer [143] (learning rate: $1 \cdot 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \cdot 10^{-8}$) using mini-batches of size 32. Our training procedure verifies that all samples within a batch are from a different set of speakers. In addition to the training dataset, we created a fully separated validation dataset of $T_{\text{valid}} = 5720$ samples using a different set of speakers from *LibriSpeech dev-clean*. Early stopping (*patience* $= 10$) is applied by monitoring the validation loss to reduce the effect of overfitting. Training never exceeded more than 50 epochs.

We used the Keras [53] framework and trained on multiple instances of Nvidia GTX 1080 GPUs.

8.3 MODEL SELECTION

In this section, we evaluate three configurations of our proposed architectures, introduced in Section 8.2. Besides the architecture, we investigate different input representations as well as the three proposed output distributions (see Section 8.1). The goal of this is to determine the effect of these parameters and fix them to select a final trained network (model) based on these parameters.

To allow for a controlled test environment and at the same time limit the number of training iterations, we fix certain parameters: In this experiment, the level of the speakers was adjusted before mixing such that they have equal power. Furthermore, the input duration D was fixed to five seconds. For all experimental parameters, we repeated the training three times with different random seeds for each run and report averaged results to minimize random effects caused by early stopping. We used the *LibriSpeech* dataset for both training and validation and performed evaluation of all models on $T_{\text{test}} = 5720$ unique and unseen speaker mixtures from *LibriSpeech test-clean* set with $k_{\text{max}} = 10$.

Several well-established input representations were evaluated in [271] such as (linear or logarithmically scaled) STFT, Mel filter bank outputs (MEL), Mel Frequency Cepstral Coefficients (MFCC) representations, typically chosen for speech applications (compare Table 8.3)

3 Note that for the validation and testing, excerpts are fixed.

Even though MFCCs are used in related tasks and are included in our baseline evaluations, they are known to perform poorly when used in CNNs [259]. This is why we decided to not to use the MFCCs as an input for the proposed architectures. The remaining input representations are identical to those listed in [271]:
1) STFT: magnitude of the short-time Fourier transform computed using Hann-windows. A frame length of 25 ms has been used. The resulting input is $X \in \mathbb{R}^{500\times201}$.
2) STFTLOG: logarithmically scaled magnitudes from STFT representation using $\log(1+STFT)$. The resulting input is $X \in \mathbb{R}^{500\times201}$.
3) MEL: compute mapping from the STFT output directly onto Mel basis using 40 triangular filters. The resulting input is $X \in \mathbb{R}^{500\times40}$.

Before feature transformation, all input files were re-sampled to 16 kHz sampling rate. All features are computed using a hop size of 10 ms.

8.3.1 Metric

Whereas the intermediate output y is treated as either a classification or a regression problem (see Section 8.1) we evaluate the final output k as a discrete regression problem. We, therefore, employ the mean absolute error (MAE) which is also commonly used for other count related tasks (c.f. [230, 326]). Since the MAE depends on the true count k, we also present the MAE per class as:

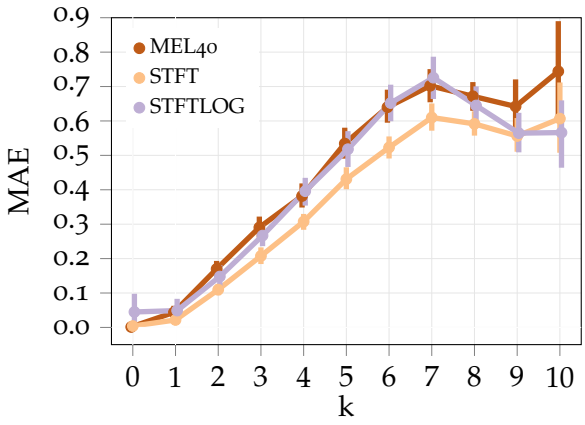
$$\mathrm{MAE}(k) = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} \left| \hat{k} - k \right| . \tag{8.11}$$

which is then averaged across the classes, i.e.,

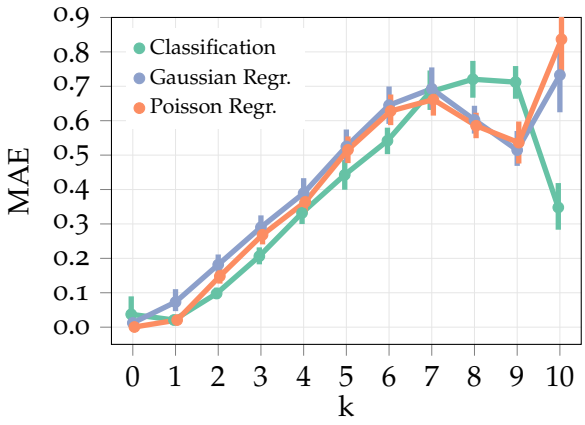
$$\mathrm{MAE} = \frac{1}{k_{\max}} \sum_{k=0}^{k_{\max}} \mathrm{MAE}(k). \tag{8.12}$$

8.3.2 Model Comparison

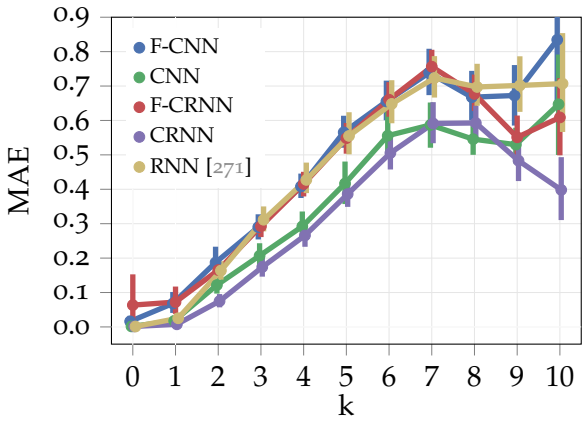
To find the best parameters we performed training and evaluation for different input representations and output distributions (c.f. [271]) as well as all proposed architectures resulting in 135 models. On average each model was trained for 25 epochs before early stopping was engaged. We present the results filtered by the three factors (Architecture, Input and Output) in Fig. 8.3. One can see that the overall trend of the count error in MAE is similar regardless of the parametrization: all models are able to reliably distinguish between $k = 0$ and $k = 1$, followed by a nearly linear increase in MAE for $k = \{1, 2, \ldots, 7\}$. For $k > 7$ it can be seen that the classification type models have learned the maximum of k across the dataset, hence the prediction error de-



(a) by feature representations.



(b) by output distribution.



(c) by feature representations.

Figure 8.3: Figure shows results of average mean absolute error (MAE) on mixtures of speakers with equal power as described in Section 8.3.2 per ground truth count $k = [0 \dots 10]$. Error bars show the 95% confidence intervals. Results in (a) are averaged over factors shown in (b) and (c) and similarly for (b) and (c).

creases when k reaches its maximum. This is because classification based models intrinsically have access to the maximum number of sources determined by the output vector dimensionality. Furthermore, one can see that all three factors have only little effect on the overall performance of the model, which is especially the case for small k. As indicated by Fig. 8.3a, choosing linear STFT as input representation generally results in a better performance compared to MEL and even STFTLOG. Concerning the output distribution, a similar observation can be made about classification which outperforms Poisson regression and Gaussian regression, as indicated by Fig. 8.3b. In Fig. 8.3c the performance of our proposed architectures are compared: while CNN and CRNN are close, both of them perform better than full frequency band F-CNN and F-CRNN models as well as the recurrent based architecture, proposed in [271]. However, it is interesting that, despite its simplicity, the F-CNN and F-CRNN, perform similarly to the Bi-LSTM architecture.

The results are supported by a statistical evaluation based on mixed effect linear model (see Table 8.4) where k is modeled as a random effect (for further details we refer to [184]). For a fair comparison (i.e. reducing the bias towards classification type network) of all models we only evaluate results for $k = \{1, 2 \dots 7\}$; however, all networks were trained on $k = \{0, \dots, 10\}$. These results indicate that CRNN performs statistically significantly better than the CNN. Concerning the input representation, we can report that using STFT representation outperforms the log-scaled STFT as well as the MEL representation. Interestingly, we did not find any significant differences between MEL and STFTLOG in MAE performance. With respect to the output distributions, we can report that Classification outperforms the other two distributions while Poisson regression performs better than Gaussian regression which confirms the findings made in [271] based on the RNN model. Therefore, we select the CRNN classification model with STFT features for subsequent experiments.

Figure 8.4 gives an indication of the efficiency of each model and the trade-off between performance and complexity in terms of parameters and floating point multiplications. It can be seen that the CRNN is not only the one that performs best but also has significantly fewer parameters than the CNN model. In contrast, the F-CRNN model does only have a fraction of the number of parameters of the other models, which makes it the most suitable model for mobile applications.

8.4 EVALUATION RESULTS

In this section, we perform several experiments on the proposed CRNN model that has been selected in the previous section. We assess the performance of this model by showing the results of three

Factor	Coef.	Std.Err.	z	$P > \|z\|$
Intercept	0.305	0.091	3.360	0.001
architecture = CRNN	-0.028	0.011	-2.419	0.016
architecture = F-CNN	0.102	0.011	8.976	0.000
architecture = F-CRNN	0.102	0.011	8.947	0.000
architecture = RNN	0.094	0.011	8.240	0.000
feature = STFT	-0.079	0.009	-8.946	0.000
feature = STFTLOG	-0.001	0.009	-0.117	0.907
objective = P-Regression	0.040	0.009	4.555	0.000
objective = G-Regression	0.067	0.009	7.651	0.000
Random Effect k	0.057	0.297		

Table 8.4: Mixed Effects Linear Model for $k = \{1, 2 \dots 7\}$. Model: $MAE \sim architecture + feature + objective + (1|k)$.

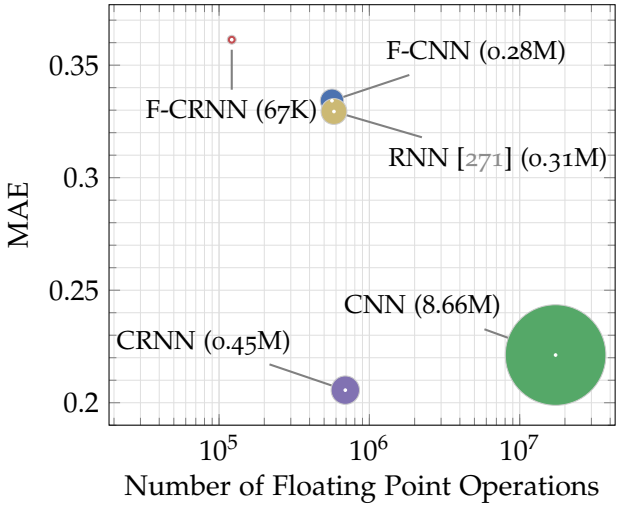


Figure 8.4: Complexity in number of floating point multiplications and number of weight parameters (in brackets) over performance in MAE of our five proposed models. ® 2019 IEEE.

experiments that augment the test data by choosing a different dataset, varying amplitude gain levels and introduce reverberation. These results also include several baseline methods. Furthermore, we present the effect of training sample duration and compare the results from the DNN to human performance gathered in a listening experiment.

8.4.1 Baselines

In order to make a meaningful comparison to the CRNN model we propose several baseline methods. Since we are dealing with a novel task description, related speaker count estimation techniques can hardly be used as baselines. Specifically, [320] would not work on fully overlapped speech, [4] does not scale to the size of our dataset, since it requires to cross-correlate the full database against

another. Finally, [247] proposes a feature but does not employ a fully automated system that can be used in a data-driven context. We, therefore, decided to propose our own baseline methods.

VQ This method uses a feature proposed by Sayoud [247] based on 7th MEL filter coefficient (MFCC_7) which was shown to encode sufficiently important speaker-related information. The temporal dimension of X is squashed down by subtracting the mean and standard deviation as $X = \overline{\text{MFCC}_7} - STD(\text{MFCC}_7) \in \mathbb{R}^1$. In [247] the mapping from $X \Rightarrow k$ is done by manually thresholding X. To translate this into a data-driven approach, we employed a vector quantizer (using k-means) to get an optimal mapping with respect to the sum of squares criterion. Further, as preprocessing, we added the same normalization as for our proposed CRNN which in turn decreases the performance of the method significantly as it is highly gain dependent.

SVM, SVR We found that the information encoded in the 7th MFCC coefficient as used in the **VQ** baseline, may not suffice to explain the high variability in our dataset. This is especially important for larger speaker counts. We therefore extended VQ by including all 20 MFCCs but using the same temporal dimensionality reduction, resulting in $X = \overline{MFCC} - STD(MFCC) \in \mathbb{R}^{20}$. To deal with significantly increased dimensionality of X, we used a support vector machine (SVM) with a radial basis function (RBF) kernel. Similarly to our proposed DNN based methods, we treat the output as either a classification problem or a regression problem through the use of support vector regression (SVR).

8.4.2 Results on Gain Variations

In our parameter optimization in Section 8.3 we evaluated mixtures with speakers having equal power. In a more realistic scenario, speakers often differ in volume between utterances. We simulate this by introducing gain factors between 0.5 and 2.0, randomly applied to the sources, hence resulting in a deviation of 6 dB compared to the reference where all speakers are mixed to have equal power. We applied this variation only to the test data to evaluate how models generalize to this updated condition. The results of this experiment are presented in Table 8.5. **MEAN** corresponds to a "dummy" estimator always predicting $k = 5$ for all test samples. Our results indicate that augmenting the mixture gains does have an impact on performance, for both, our proposed CRNN model as well as the baseline methods. For example, for the CRNN model the performance drops by 60% from 0.27 MAE to 0.43 MAE on the *LIBRI Speech* test set, which is still about 40% better than the second best-performing method *SVR* which drops from 0.58 MAE to 0.61 MAE.

Trained on	*LIBRI*							*LIBRI-Rev*
Test Set	LIBRI			THCS10		TIMIT		LIBRI-Rev
Variation	–	±6 dB	Rev	–	±6 dB	–	±6 *dB*	–
CRNN	**0.27**	**0.43**	1.63	**0.36**	**0.50**	**0.31**	**0.52**	**0.48**
RNN [271]	0.38	0.57	1.41	0.58	0.76	0.48	0.72	0.59
SVR	0.58	0.61	**0.76**	0.69	0.73	0.70	0.62	0.71
SVC	0.63	0.66	0.85	0.77	0.77	0.89	0.76	0.78
VQ [247]	2.41	2.41	2.41	2.98	2.98	2.13	2.15	2.41
MEAN	2.73	2.73	2.73	2.73	2.73	2.73	2.73	2.73

Table 8.5: Averaged MAE results of different methods on several datasets for $k = [0 \dots 10]$ with equal power and random gains (up to ±6 dB)) as well as reverberation (rev). Bold face indicates the best-performing method. Standard deviation values are listed in [272].

8.4.3 Results on Different Datasets

We also present results on two additional datasets. Again, we only changed the test data; all networks were trained on *LIBRI Speech*. Compared to *LIBRI Speech*, the *TIMIT* database has an overall lower recording quality. This is reflected by our results where the performance in MAE drops only slightly between these two datasets. Interestingly, even when we look at the results of the Mandarin language *THCS10* dataset, performance drops only slightly. More precisely, for our proposed CRNN model, test performance on *THCS10* is even better than on its own *LIBRI* dataset with gain variations. These results suggest that the trained model is speaker and language independent.

8.4.4 Effect of Reverberant Signals

Different acoustical conditions such as increased reverberation time were shown to have a large effect in speaker count estimation [205]. To analyze this effect, different acoustic conditions were simulated by generating the room impulse responses using the image method [2, 107]. For this experiment we set up an acoustical room with dimension (3.5 m × 4.5 m × 2.5 m) The microphone was positioned at (1m, 1m, 1m). For the mentioned room, 350 different reverberation times were selected uniformly sampled between 0.1 and 0.5 seconds. For each of these reverberation times, we generated unique room impulse responses that correspond to individual source positions which have minimum distance 0.1 m to the walls and are otherwise positioned randomly on the (X, Y, 1m) plane. Each speaker's signal was convolved with a randomly selected room impulse response before mixing. Results, again, are shown in Table 8.5. For the first time, we can see that the CRNN model significantly drops in performance from 0.27

MAE to 1.64 MAE, whereas the SVR and SVM baselines are only affected slightly. This is expected as these baselines are using a temporal aggregation of all frames, whereas the CRNN is based on smaller (3×3) convolutional filter operations that are able to capture the room acoustics as well. If we assume that our trained deep learning model is fully speaker independent, a mixture of two utterances from the same speaker would get the same count estimate as two different speakers. Hence, reverberation tends to result in overestimation and we observed this even for $k = 1$ where it, in turn, resulted in an increase in MAE.

To further investigate whether the overestimation can be reduced via training with reverberant samples, we created a separate set of room impulse responses for the training dataset with different room dimensions so that the model cannot learn the acoustical conditions from the training dataset. From the results shown in the last column of Table 8.5 we can see that the retrained CRNN is able to outperform the baselines again. Therefore, when retrained with reverberant samples, the proposed model is able to better discriminate between a reverberant component of the same speaker and contributions from different speakers. For robustness against different acoustic conditions, it is essential to include reverberant samples in the training dataset.

8.4.5 Effect of Duration and Overlap Detection Error

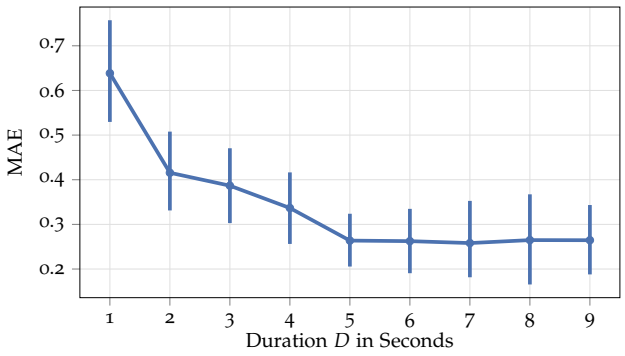


Figure 8.5: Evaluation of trained CRNN networks over different input duration length D. Error bars show 95% confidence intervals.

In our last experiment we want to address the influence of the input duration length D. In a real-world application this parameter would be chosen as small as a possible, because a longer input duration adds both algorithmic and computational delay to a real-time system. In a small experiment, we took the proposed CRNN and retrained it using a different number of input frames ranging from 100 to 900 frames (corresponding to one to nine seconds of audio). For each input duration, we trained the CRNN with three different initial seeds.

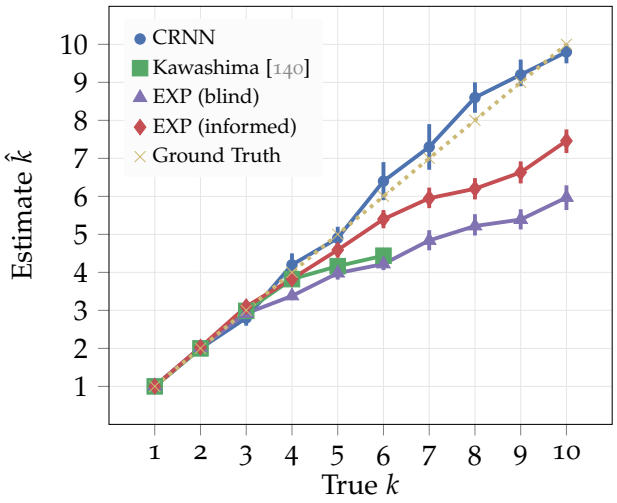


Figure 8.6: Average responses from humans (EXP and *Kawashima* [140]) compared to our proposed CRNN. Error bars show 95% confidence intervals. ® 2019 IEEE.

Results are shown in Fig. 8.5. It can be seen that five second duration is a good trade-off between performance and delay. If latency is critical, keeping D above 2 seconds is recommended for good results. For segments as short as 1 second the MAE of around 0.6 is almost twice as high as for segments of 5 seconds duration. However, if instead of the count estimation MAE we compute the accuracy to detect overlap $k > 1$ vs. non-overlap $k \in 0, 1$, we still achieve 98.7% accuracy (precision: 99.7%, recall: 98.7%). This shows that our system can be effectively used to address overlap detection.

8.4.6 Listening Experiment

We chose to compare our trained CRNN against human performance using the experiments made in [138, 140] and our own as described in Section 7.2. The results are shown in Figure 8.6. The results for up to three speakers indicate that humans perform similarly (or better in terms of variance) compared to our proposed CRNN model. For larger speaker counts, the gap between humans and algorithm is almost three speakers on average. Interestingly, the results of the informed experiment reveal that this gap closes down to an average difference of one speaker. Finally, we can report that the machine model reached superhuman performance. Unlike humans, the CRNN is subject to over-estimations for $4 < k \leq 9$. However, with extensive training, humans might be able to perform on par.

8.5 UNDERSTANDING COUNTNET

In this section, we focus on the problem of interpreting the strategy undergone by this system for successfully estimating counts.

8.5.1 Saliency Maps

We first conducted a visual analysis based on salience map representations [262]. In the deep learning context, saliency maps are visualizations that are able to show which specific input elements are important given a specific output prediction. In vision, this allows to show which pixels were most relevant to make up the decision to activate a specific output class. In the case of audio it can highlight which time-frequency bins in a spectrogram are most relevant. The common idea is to compute the gradient of the model's prediction with respect to the input, holding the weights fixed. This determines which input elements need to be changed the least to affect the prediction the most.

In this work, we used guided backpropagation, first introduced in [269] and successfully deployed in [251] to compute a saliency map for singing voice detection. For a given input of a three-speaker mixture, we depicted the saliency map in Fig. 8.7. The saliency map indicates that our proposed model does not rely much on the overlapped parts but instead utilize many of the single speaker time-frequency bins as well as many high-frequency components such as plosives and fricative phonemes.

While the saliency map confirms that the network does exploit both low and high-frequency content from the input signal, it is not sufficient to conjecture about the strategy implemented in the network.

8.5.2 Ablation Analysis

To provide further insight, we propose another layer-wise analysis, that provides information concerning the behavior of the model at different successive layers. While we cannot show all filter outputs (e.g. 64, for the first layer), instead, for each filter, we compute its loss with respect to the input of the model using gradient update and sort the filters according to their loss behavior.

Figure 8.7 depicts the nine highest loss outputs per convolutional layer. We can observe that while the first layer shows only low-level variations of the input, already the second layer seems to be more abstract and emphasizes phoneme segmentations based on mid and high frequency content. While filter outputs of layer 3 and 4 also show more low-frequency content such as the harmonic signals, the overall visual impression is that the proposed CRNN focuses on the temporal segmentation of phonemes.

The conducted analysis suggests that the network is doing count estimation based on the detection of phonemes. To assess the validity of this interpretation, we directly verified the performance of the method as a function of the phoneme activity. In the following, we verify whether count estimates are affected by the pronunciation speed.

We assume that the CRNN model learned the aggregated phoneme or syllable activity of all speakers in a fixed, given excerpt.

If that is the case, it would mean that the speaker count estimate would be affected if the speakers would speak slower or faster in relation to the fixed input window (speaking rate). We therefore want to see if very slow or very fast speakers significantly increase the error of our proposed CRNN model. In turn we define a null hypothesis that there is no association between the speaker count error probability and the value of the *speaking rate*.

To verify our hypothesis, we created another experiment based on the *TIMIT* dataset. It comes with phoneme and word level annotations, from which the speaking rate (defined as syllables per second) can be computed for each input sample [136]. To reduce the influence of the different acoustical environment in TIMIT compared to Libri Speech, we retrained the CRNN classification model on the *TIMIT* training dataset, using the same parameters as described in Section 8.2.4. At test time we randomly generated 5 seconds excerpts of $k = 6$ from the TIMIT test subset and predicted the error $E(k) = \hat{k} - k$ for each CRNN output. We grouped the estimates into three classes: $E(k) = 0$ (correct response), $E(k) > 0$ (overestimation), $E(k) < 0$ (underestimation). For $k = 6$ we ended up with two groups of results because overestimation did not take place. From the remaining two groups *underestimation* and *correct* responses we randomly selected 1000 samples each, resulting in an total sample size of $n = 2000$. For these samples we computed an average speaking rate of 3.40 syllables per second and a standard deviation of 0.2.

We chose a Generalized Linear Model (GLM) for the statistical test, as described in [132]. This allows us model the results with a binary logit regression model that turns the mean of E into a binomially distributed probability modeled by log linear values: $\text{logit}(E) \sim \text{Intercept} + \beta \cdot \text{Speaking Rate}$. The results of our test are shown in Table 8.6 and indicate the speaking rate has statistically significant influence on the error $p < 0.05, df = 1, \text{Pseudo } R^2 = 0.0111$. To better understand the effect of our predictor, we computed an odds ratio $\exp(\text{speaking rate}) = 0.28$.

This indicates that a decrease in speaking rate of 1 syllable per second will increase the likeliness of an underestimation error by 28 percent. Even though this is considered as a small effect size, it gives an interesting hint for the strategy taken of our proposed model and also suggests that for improved robustness, training would benefit

	coef	std err	z	P>|z|
speaking rate	-1.2697	0.232	-5.477	0.000
intercept	4.3213	0.790	5.468	0.000

Table 8.6: Results of a binary logit regression test for the dependent variable *correct response* over the independent variable *speaking rate*. The results are based on $n = 2000$ randomly drawn results of the CRNN model trained and evaluated on the TIMIT dataset.

from a large variety of speaking rates. Furthermore, it still remains unclear if the model would suffer from languages with a speaking rate which is naturally higher or lower than English or Chinese (see [200]).

8.6 SUMMARY AND DISCUSSION

We introduced the task of estimating the maximum number of concurrent speakers in a simulated "cocktail-party" environment using a data-driven approach, discussing how to frame this task in a deep learning context. Building upon earlier work, we investigated what method is best to output integer source count estimates and also defined suitable cost functions for optimization. In a comprehensive study, we performed experiments to evaluate different network architectures. Furthermore, we investigated and evaluated other important parameters such as input representations or the input duration. Our final proposed model uses a convolutional recurrent (CRNN) architecture, based on classification at the network's output. Compared to several baselines, our proposed model has a significantly lower error rate; it achieves error rates of less than 0.3 speakers in mean absolute error for classifying zero to ten speakers—a decrease of 28.95% compared to [271]. In further simulations, we revealed that our model is robust to unseen languages (such as Chinese), as well as varying acoustical conditions (except for reverberation, where the error increased significantly). However, including reverberated samples in the training reduces the error. Additionally, we conducted a perceptual experiment showing that these results clearly outperform humans. We hope our research stimulates future research on data-driven count estimation, a task that currently lacks real-world datasets. Future methods could also work on further reducing the duration of excerpts to improve the conceptual latency.

Lastly, in an ablation study, we found that the CRNN uses a strategy to segment phonemes/syllables to estimate the count. Hence, we hypothesize that a speaker count estimate is influenced by the average speaking rates of certain languages. To underpin this hypothesis, we

showed that the speaking rate has a significant effect on the error of our model. Interestingly, the speaking rate is an important source of modulations in speech [211] and this discovery establishes a link between speech analysis in humans and machines.

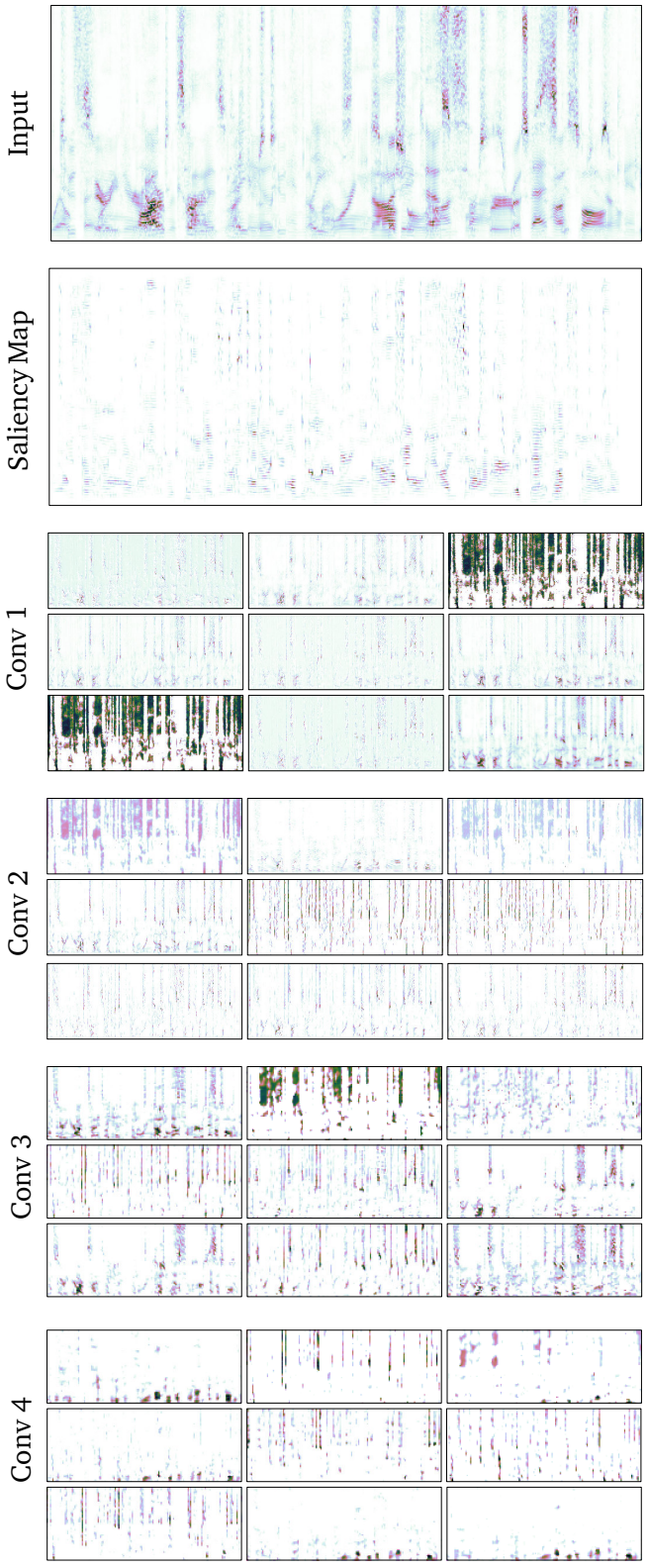


Figure 8.7: Illustration of intermediate outputs from the proposed CRNN for each convolutional layer for a given input with $k = 3$ speakers. Saliency map shows positive saliency of guided backpropagation [269]. For each convolutional layer the nine most relevant filters were selected based on their loss with respect to the input.

9 CONCLUSION

9.1 DISCUSSION

Note that additional conclusions can be found at the end of the respective chapters.

This thesis built on and contributed to work in the research field of separation and source count estimation of audio mixtures. The methods that were developed throughout this thesis share a common concept: when signals are mixed, we focussed on the overlapping part instead of the non-overlapping part. This approach allowed us to better observe the characteristics for the task of separation and source-count estimation:

- In our proposed unison source separation scenario, signals almost entirely overlap in time and frequency which reflect a natural property of many real-world audio signals.
- For the task of source count estimation, we focused on the overlap to learn a model that directly infers counts, i.e. proposed *counting without detection*. The motivation is in the tradition of [249], who proposed techniques for *understanding without separation*.

The combination of separation and count estimation is important since it allows to build a real-world separation system, where the number of sources is not known in advance. On this way, we were faced with several limitations and challenges:

When we developed time warping (Chapter 5) to separate audio signals based on their estimated F_0 trajectory, we optimized and tuned the method on a handful of music tracks, since a large test dataset was not available. When we then applied it on the (back then) new DSD100 [166] dataset, several limitations became apparent which is why the overall performance was not satisfying. One limitation of the method is that it relies too much on a precise and robust estimate of the F_0 trajectory and its voice activity. When we included a data-driven activity method in the system, the performance improved.

When it comes to source separation, at the time our research was conducted and we proposed the Common Fate Model (Section 6.2), factorization models were the state-of-the-art. However, for the scenario of music separation, it turned out that the Common Fate Model still required a significant amount of additional handcrafted engineering. By 2016, other researchers showed in [195, 293] that supervised learning methods for separation led to huge improvements. This induced

us to successfully evaluate such a system in combination with our proposed Common Fate Representation (Section 6.3). It shows that such specialized representation, designed to capture general modulation patterns, is still useful in data-driven methods. Our research marks a step towards multidimensional representations, inspired by human perception, when it comes to highly overlapped, modulated mixtures. Work by other researchers built upon our first findings and further increased the perceptual aspect of the representation by introducing a multi-resolution version of the CFT [210, 257].

In our count estimation studies on speech and music (Chapter 7), we showed that humans follow a "one-two-three-many" strategy: we can correctly infer small quantities up to three, but we have problems to extrapolate to higher numbers. We showed that our proposed CountNet model (Chapter 8) improved state-of-the-art and even reached super-human performance by shifting the performance boundary beyond four sources. In the ablations studies (Section 8.5), we revealed that modulations also played an essential role in CountNet when estimating the number of speakers. The network showed a significant dependency on variations of syllable rates. This indicates that CountNet may not have learned the actual difference between two and three speakers but instead took a "shortcut" and learned the distinct modulation patterns of our language. This achievement, however, does not mean that CountNet can generalize to examples outside of the used datasets or languages. Such generalization properties should be properly investigated further.

This work covers many different techniques ranging from advanced audio signal processing, tensor factorization, up to recent supervised machine learning methods such as deep learning. When I started this thesis, the research landscape for source separation methods was built upon the expertise from a decade of signal processing. With the success of deep learning, a paradigm shift took place that enabled many improvements with respect to the state-of-the-art in the audio domain. However, adapting deep learning techniques to work in the audio domain is far from trivial. First, it requires a significant amount of work in creating suitable datasets. Second, it still takes expertise in so-called "classical" signal processing for such models to work correctly.

9.2 PERSPECTIVES

In the following, I will present a few potential research ideas, based on the findings and limitations raised in this thesis.

Generative Modulation Models for Style Transfer

Imagine, generating contours to *add* instead of *remove* vibrato by use of the time warping (Section 5) to improve the naturalness of synthesis models. Recent progress on generative models such as GANs [97] show powerful models can extract domain properties from data. We can imagine that generative models could explore a modulation space, e.g. to generate artificial vibrato that could enable applications such as musical style transfer to apply modulations on other "flat" voices or instruments.

New Representations for Source Separation

Recall that the Common Fate Transform proposed in Section 6.2.1, led to increased redundancy, introduced by its aliasing components, which in turn, increases the complexity of the training. Interestingly, preliminary studies suggested that removal of the (redundant) components did not improve the performance. Future work could reduce the redundancy of the transform, making it more compact, while at the same time yielding similar separation results. In the context of deep learning, it remains unclear if redundancy helps or hinders supervised learning based separation systems. Taking the raw waveform as input features [65, 295] is a promising direction of research (as opposed to phase aware representations) but it requires large amounts of data since the DNN needs first to learn a filterbank representation. Such large amounts are not yet publicly available for music processing.

Deep Common Fate

The combination of more powerful DNN architectures such as convolutional neural networks (CNN) with Common Fate Transform (CFT) is a promising route for future work. This is especially interesting because it would require network architectures to be specifically designed to deal with higher dimensional data such as the four-dimensional CFT. I can imagine applying recent architecture designs such as multidimensional convolutional networks or capsule networks [239]. Furthermore, it is to be seen if learning based methods can directly utilize modulations from raw data.

Applications for Crowd Sources Count Estimation

In our experiments to study the human ability to estimate the number of sources (Section 7), we made use of recent web technologies such as the Web Audio API to enable crowdsourced listening experiments. As current web audio technologies mature we will see many more web-based experiment and evaluation tools coming. With more data

being annotated on human source count estimates, we can imagine using this data to build an auditory model that approximates the perception of estimating counts. With such a model one may further improve lossy compression for object-based music recordings such as audio coding [115] by only transmitting a "perceivable" number of sources.

Count Estimation as an Evaluation Measure for Separation

The recent results on SiSEC 2018 [283][1] indicated that for the first time since the advent of source separation, methods were proposed that perform comparably to the oracle separation methods. This success was made possible with better deep learning models such as [289] and the availability of data. The future of how to improve music source separation is unclear. Possible directions are to enhance the efficiency of the learning architectures or to develop new cost functions which better reflect human perception. Furthermore, improving the evaluation metrics is another driving force for better separation algorithms. Unfortunately for separation, humans cannot easily evaluate the audio quality without a reference, which makes annotations cumbersome and expensive. Therefore a simpler task such as count estimation could serve as an intermediate evaluation metric to quantify overlap, which is easier to annotate and does not require a reference. Then, a model such as CountNet could approximate the human results and in turn, be used (hence differentiable) inside the cost function of other separation models.

Teaching CountNet to Extrapolate

CountNet was developed to address the count estimation task in both, a classification or a regression framework. However, counts are often not bounded to a maximum number, which would require to extrapolate and not just interpolate. CountNet, as it was proposed, is unable to estimate more than ten speakers when trained using the same maximum number. This is related to learning the summation of two random numbers, known as the "adding problem" [119] which is a challenging benchmark in machine learning. Only very recently, the machine learning community proposed a solution to this problem (See [292]). With these advances, extrapolating counts seems like a natural follow-up to the work presented here.

1 See https://sisec18.unmix.app

BIBLIOGRAPHY

1. M. Abramowitz and I. Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical tables (applied mathematics series 55)," *National Bureau of Standards, Washington, DC*, 1964.
2. J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.
3. D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Intl. Conference on Machine Learning (ICML)*, 2016, pp. 173–182.
4. V. Andrei, H. Cucuand, and C. Burileanu, "Counting competing speakers in a time frame - human versus computer," in *Proc. Interspeech Conf.*, 2015, pp. 3399–3403.
5. V. Andrei, H. Cucuand, and C. Burilieanu, "Detecting overlapped speech on short timeframes using deep learning," 2017, pp. 1198–1202.
6. V. Andrei, H. Cucuand, A. Buzo, and C. Burileanu, "Estimating competing speaker count for blind speech source separation," in *Proc. International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2015, pp. 1–8.
7. X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 356–370, 2012.
8. T. Arai, "Estimating number of speakers by the modulation characteristics of speech," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 2003, pp. II–197.
9. S. Araki, T. Nakatani, H. Sawada, and S. Makino, "Stereo source separation and source counting with map estimation with dirichlet prior considering spatial aliasing problem," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Springer, 2009, pp. 742–750.
10. S. Arberet, R. Gribonval, and F. Bimbot, "A robust method to count and locate audio sources in a multichannel underdetermined mixture," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 121–133, 2010.

11. C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *European Conference on Computer Vision*, Springer, 2016, pp. 483–498.

12. E. Azarov, M. Vashkevich, and A. Petrovsky, "Instantaneous pitch estimation based on RAPT framework," in *20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2787–2791.

13. O. Babacan, T. Drugman, N. D'alessandro, N. Henrich, and T. Dutoit, "A comparative study of pitch extraction algorithms on a large variety of singing sounds," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7815–7819.

14. T. Bäckström, S. Bayer, and S. Disch, "Pitch variation estimation," in *Proc. Interspeech Conf.*, 2009, pp. 2595–2598.

15. S. P. Bacon and D. W. Grantham, "Modulation masking: Effects of modulation frequency, depth, and phase," *The Journal of the Acoustical Society of America*, vol. 85, no. 6, pp. 2575–2580, 1989.

16. R. Badeau, "Gaussian modeling of mixtures of non-stationary signals in the time-frequency domain (HR-NMF)," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2011, pp. 253–256.

17. R. Badeau and A. Dremeau, "Variational Bayesian EM algorithm for modeling mixtures of non-stationary signals in the time-frequency domain (HR-NMF)," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 6171–6175.

18. R. Badeau and M. D. Plumbley, "Multichannel high resolution NMF for modelling convolutive mixtures of non-stationary signals in the time-frequency domain," *IEEE Trans. on Audio, Sp. & Lang. Proc.*, vol. 22, no. 11, pp. 1670–1680, Nov. 2014.

19. R. Badeau and M. Plumbley, "Multichannel HR-NMF for modelling convolutive mixtures of non-stationary signals in the time-frequency domain," in *Proc. WASPAA*, New Paltz, NY, USA, Oct. 2013, pp. 1–4.

20. S. Balke, C. Dittmar, J. Abeßer, K. Frieler, M. Pfleiderer, and M. Müller, "Bridging the Gap: Enriching YouTube videos with jazz music annotations," *Frontiers in Digital Humanities*, vol. 5, pp. 1–11, 2018.

21. T. Barker and T. Virtanen, "Non-negative tensor factorisation of modulation spectrograms for monaural sound source separation," in *Proc. Interspeech Conf.*, 2013.

22. J. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

23. J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

24. R. M. Bittner *et al.*, "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th Int. Society for Music Information Retrieval Conference ISMIR, Taipei, Taiwan, October 27-31, 2014*, H. Wang, Y. Yang, and J. H. Lee, Eds., 2014, pp. 155–160.

25. R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive mir research," in *15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 2014.

26. K. Boakye, O. Vinyals, and G. Friedland, "Two's a crowd: Improving speaker diarization by automatically identifying and excluding overlapped speech," in *Proc. Interspeech Conf.*, 2008, pp. 32–35.

27. P. Boersma, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2001.

28. D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: An open-source library for sound and music analysis," in *Proc. ACM Intl. Conference on Multimedia (ACMMM)*, ACM, 2013, pp. 855–858.

29. L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proc. ACM Intl. Conference on Multimedia (ACMMM)*, ACM, 2016, pp. 640–644.

30. J. J. Bosch, K. Kondo, R. Marxer, and J. Janer, "Score-informed and timbre independent lead instrument separation in real-world scenarios," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012.

31. K. Bötzel, V. Tronnier, and T. Gasser, "The differential diagnosis and treatment of tremor," *Deutsches Ärzteblatt International*, vol. 111, no. 13, p. 225, 2014.

32. A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. Cambridge: Bradford Books, MIT Press, 1990.

33. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

34. D. Buchla, "A history of buchla's musical instruments.," in *NIME*, 2005, p. 1.

35. D. C. Burr, M. Turi, and G. Anobile, "Subitizing but not estimation of numerosity requires attentional resources," *Journal of Vision*, vol. 10, no. 6, p. 20, 2010.

36. C. Févotte and J.-F. Cardoso, "Maximum likelihood approach for blind audio source separation using time-frequency Gaussian models," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2005, pp. 78–81.

37. E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

38. E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.

39. E. Cano, G. S., and C. Dittmar, "Pitch-informed solo and accompaniment separation towards its use in music education applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–19, 2014.

40. E. Cano and C. Cheng, "Melody line detection and source separation in classical saxophone recordings," in *Proc. Conf. on Digital Audio Effects*, Como, Italy, Sep. 2009.

41. E. Cano, C. Dittmar, and G. Schuller, "Efficient implementation of a system for solo and accompaniment separation in polyphonic music," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012.

42. E. Cano, C. Dittmar, and G. Schuller, "Re-thinking sound separation: Prior information and additivity constraints in separation algorithms," in *Proc. Conf. on Digital Audio Effects*, Maynooth, Ireland, Sep. 2013.

43. R. P. Carlyon, "How the brain separates sounds," *Trends in cognitive sciences*, vol. 8, no. 10, pp. 465–471, 2004.

44. A. B. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *Proc. IEEE Intl. Conference on Computer Vision (ICCV)*, 2009, pp. 545–551.

45. T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the iKala dataset," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015.

46. P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 4428–4437.

47. E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

48. A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.

49. T. Chi, Y. Gao, M. C. Guyton, P. Ru, and S. Shamma, "Spectro-temporal modulation transfer functions and speech intelligibility," *Journal of the Acoustical Society of America*, vol. 106, no. 5, pp. 2719–2732, Nov. 1999.

50. T. Chi, P. Rub, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *Journal of the Acoustical Society of America*, vol. 118, no. 2, pp. 887–906, Aug. 2005.

51. K. P. Choi, "On the medians of gamma distributions and an equation of ramanujan," *Proceedings of the American Mathematical Society*, vol. 121, no. 1, pp. 245–251, 1994.

52. K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2392–2396.

53. F. Chollet *et al.*, *Keras v1.2.2*, `https://github.com/fchollet/keras/tree/1.2.2`, 2015.

54. J. M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the audio engineering society*, vol. 21, no. 7, pp. 526–534, 1973.

55. M. G. Christensen, A. Jakobsson, and S. H. Jensen, "Joint high-resolution fundamental frequency and order estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1635–1644, 2007.

56. A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing, Sep. 2009.

57. P. Common and C. Jutten, *Handbook of Blind Source Separation*. Academic Press, 2010.

58. P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation: Independent Component Analysis and Blind Deconvolution*. Academic Press, 2010.

59. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

60. T. Dau, "Modeling auditory processing of amplitude modulation," PhD thesis, 1999.

61. A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: Passive recovery of sound from video," *Proc. SIGGRAPH*, vol. 33, no. 4, 79:1–79:10, 2014.

62. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

63. L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, Jun. 2014.

64. P. Desain, H. Honing, R. Aarts, R. Timmers, *et al.*, "Rhythmic aspects of vibrato," *P. Desain and L. Windsor, Rhythm–Perception and Production*, pp. 203–216, 1999.

65. S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6964–6968.

66. S. Disch and B. Edler, "Multiband perceptual modulation analysis, processing and synthesis of audio signals," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2009, pp. 2305–2308.

67. C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch, "Music information retrieval meets music education," in *Multimodal Music Processing*, Dagstuhl Publishing, 2012, pp. 95–120.

68. K. Dressler, "Sinusoidal extraction using an efficient implementation of a multi-resolution FFT," in *Proc. Conf. on Digital Audio Effects*, Montreal, QC, Canada, Sep. 2006.

69. K. Dressler, "Pitch estimation by the pair-wise evaluation of spectral peaks," in *Proc. AES Conference on Semantic Audio*, Ilmenau, Germany, Jul. 2011.

70. J. Driedger, S. Balke, S. Ewert, and M. Müller, "Template-based vibrato analysis in music signals," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2016.

71. L. Drude, A. Chinaev, D. H. T. Vu, and R. Haeb-Umbach, "Source counting in speech mixtures using a variational EM approach for complex watson mixture models," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6834–6838.

72. N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1830–1840, Sep. 2010.

73. J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1180–1191, Oct. 2011.

74. C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "Complex domain onset detection for musical signals," in *Proc. Conf. on Digital Audio Effects*, London, UK, Sep. 2003.

75. EBU, *Loudness normalisation and permitted maximum level of audio signals (EBU Recommendation R 128)*, Geneva, 2011.

76. B. Edler, S. Disch, S. Bayer, F. Guillaume, and R. Geiger, "A time-warped MDCT approach to speech transform coding," in *126th AES Convention*, Preprint 7710, Munich, Germany, May 2009.

77. M. Elhilali, T. Chi, and S. A. Shamma, "A spectro-temporal modulation index (stmi) for assessment of speech intelligibility," *Speech Communication*, vol. 41, no. 2, pp. 331–348, 2003.

78. V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2046–2057, 2011.

79. S. Ewert and M. Müller, "Using score-informed constraints for NMF-based source separation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 129–132.

80. S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, May 2014.

81. N. Fallah, H. Gu, K. Mohammad, S. A. Seyyedsalehi, K. Nourijelyani, and M. R. Eshraghian, "Nonlinear poisson regression using neural networks: A simulation study," *Neural Computing and Applications*, vol. 18, no. 8, p. 939, 2009.

82. C. Févotte, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, Mar. 2009.

83. C. Févotte and A. Ozerov, "Notes on nonnegative tensor factorization of the spectrogram for audio source separation: Statistical insights and towards self-clustering of the spatial cues," in *proc. International Symposium on Computer Music Modeling and Retrieval*, Málaga, Spain, Jun. 2010.

84. D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, vol. 2008, 15 pages, 2008, Article ID 872425.

85. D. FitzGerald, M. Cranitch, and E. Coyle, "On the use of the beta divergence for musical source separation," in *Proc. ISSC*, Galway, Ireland, Jun. 2008.

86. D. Fitzgerald, M. Cranitch, and E. Coyle, "Shifted non-negative matrix factorisation for sound source separation," in *Proc. IEEE Workshop Statistical Signal Processing*, Jul. 2005, pp. 1132–1137.

87. N. H. Fletcher, "Vibrato in music," *Acoustics Australia*, vol. 29, no. 3, pp. 97–102, 2001.

88. C. Füllgrabe, M. A. Stone, and B. C. J. Moore, "Contribution of very low amplitude-modulation rates to intelligibility in a competing-speech task," *The Journal of the Acoustical Society of America*, vol. 125, no. 3, pp. 1277–1280, 2009.

89. D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 4930–4934.

90. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *DARPA TIMIT acoustic phonetic continuous speech corpus CDROM*, 1993.

91. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, 1993.

92. J. T. Geiger, F. Eyben, B. W. Schuller, and G. Rigoll, "Detecting overlapping speech with long short-term memory recurrent neural networks.," in *Proc. Interspeech Conf.*, 2013, pp. 1668–1672.

93. *GENESIS S.A.: Loudness toolbox (version 1.2)*, 2012.

94. D. Giannoulis, D. Barchiesi, A. Klapuri, and M. D. Plumbley, "On the disjointess of sources in music using different time-frequency representations," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2011, pp. 261–264.

95. J. Gilbert, L. Simon, and J. Terroir, "Vibrato of saxophones," *The Journal of the Acoustical Society of America*, vol. 118, no. 4, pp. 2649–2655, 2005.

96. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

97. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

98. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2002, pp. 287–288.

99. E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *Proc. GlobalSIP*, Nov. 2017, pp. 1265–1269.

100. E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Single channel audio source separation using deep neural network ensembles," in *Proc. Audio Eng. Soc. Convention*, Paris, France, May 2016.

101. A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 6645–6649.

102. G. Gravier, G. Adda, N. Paulson, M. Carr'e, A. Giraudel, and O. Galibert, "The ETAPE Corpus for the Evaluation of Speech-based TV Content Processing in the French Language," in *LREC - Eighth international conference on Language Resources and Evaluation*, Turkey, 2012.

103. A. Gray and J. Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 3, pp. 207–217, Jun. 1974.

104. S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the switchboard corpus," in *Proc. Intl. Conf. on Spoken Lang. Processing (ICSLP)*, 1996.

105. S. Greenberg and B. Kingsbury, "The modulation spectrogram: In pursuit of an invariant representation of speech," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 1997, p. 1647.

106. S. Grollmisch, E. Cano, and C. Dittmar, "Songs2See: Learn to play by playing," in *proc. AES Conference: Audio for Games*, Feb. 2011, P2–3.

107. E. A. P. Habets, *Room impulse response (RIR) generator*, `https://github.com/ehabets/RIR-Generator`, 2016.

108. G. Hagerer, V. Pandit, F. Eyben, and B. Schuller, "Enhancing LSTM RNN-based speech overlap detection by artificially mixed data," in *Proc. Audio Eng. Soc. Conference on Semantic Audio*, Jun. 2017.

109. Y. Han and C. Raphael, "Desoloing monaural audio using mixture models," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, Victoria, BC, Canada, Oct. 2007.

110. S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

111. R. Hennequin, R. Badeau, and B. David, "Time-dependent parametric and harmonic templates in non-negative matrix factorization," in *Int. Conf. on Digital Audio Effects (DAFx)*, 2010, pp. 246–253.

112. R. Hennequin, R. Badeau, and B. David, "NMF with time–frequency activations to model nonstationary audio events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 744–753, 2011.

113. C. T. Herbst, S. Hertegard, D. Zangger-Borch, and P.-Å. Lindestad, "Freddie mercury—acoustic analysis of speaking fundamental frequency, vibrato, and subharmonics," *Logopedics Phoniatrics Vocology*, vol. 42, no. 1, pp. 29–38, 2017.

114. M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Proc. Neural Information Processing Conf*, Lake Tahoe, NV, USA, Dec. 2013.

115. J. Herre *et al.*, "MPEG spatial audio object coding—the ISO/MPEG standard for efficient coding of interactive audio scenes," *J. Audio Eng. Soc*, vol. 60, no. 9, pp. 655–673, 2012.

116. J. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 31–35.

117. G. Hinton, L. Deng, D. Yu, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

118. S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

119. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

120. G. T. Hoopen and J. Vos, "Effect on numerosity judgment of grouping of tones by auditory channels," *Attention, Perception, & Psychophysics*, vol. 26, no. 5, pp. 374–380, 1979.

121. T. Houtgast and H. J. Steeneken, "A review of the mtf concept in room acoustics and its use for estimating speech intelligibility in auditoria," *The Journal of the Acoustical Society of America*, vol. 77, no. 3, pp. 1069–1077, 1985.

122. M. Hrúz and M. Kunešová, "Convolutional neural network in the task of speaker change detection," in *Proc. Speech and Computers*, Springer, 2016, pp. 191–198.

123. C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 310–319, Feb. 2010.

124. C.-L. Hsu, J.-S. R. Jang, and T.-L. Tsai, "Separation of singing voice from music accompaniment with unvoiced sounds reconstruction for monaural recordings," in *Proc. Audio Eng. Soc. Convention*, San Francisco, CA, USA, Oct. 2008.

125. G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1135–1150, Sep. 2002.

126. P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012.

127. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.

128. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 2014.

129. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, 2015.

130. M. Huijbregts, D. A. van Leeuwen, and F. Jong, "Speech overlap detection in a two-pass speaker diarization system," in *Proc. Interspeech Conf.*, Brighton, 2009.

131. D. Huron, "Voice denumerability in polyphonic music of homogeneous timbres," *Music Perception: An Interdisciplinary Journal*, vol. 6, no. 4, pp. 361–382, 1989.

132. T. Jaeger, "Categorical data analysis: Away from anovas (transformation or not) and towards logit mixed models," 4, vol. 59, Elsevier, 2008, pp. 434–446.

133. R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Towards shifted nmf for improved monaural separation," in *Signals and Systems Conference (ISSC)*, IET, 2013, pp. 1–7.

134. I.-Y. Jeong and K. Lee, "Singing voice separation using RPCA with weighted l_1-norm," in *proc. International Conference on Latent Variable Analysis and Signal Separation*, Grenoble, France, Feb. 2017.

135. W. S. Jevons, "The power of numerical discrimination," *Nature*, vol. 3, no. 67, pp. 281–282, 1871.

136. Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Online speaking rate estimation using recurrent neural networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 5245–5249.

137. P. Joris, C. Schreiner, and A. Rees, "Neural processing of amplitude-modulated sounds," *Physiological reviews*, vol. 84, no. 2, pp. 541–577, 2004.

138. M. Kashino and T. Hirahara, "One, two, many – judging the number of concurrent talkers.," *J. Acoust. Soc. Am.*, vol. 99, no. 4, pp. 2596–2603, 1996.

139. E. L. Kaufman, M. W. Lord, T. W. Reese, and J. Volkmann, "The discrimination of visual number," *The American Journal of Psychology*, vol. 62, no. 4, pp. 498–525, 1949.

140. T. Kawashima and T. Sato, "Perceptual limits in a simulated cocktail party," *Attention, Perception and Psychophysics*, vol. 77, no. 6, pp. 2108–2120, 2015.

141. A. Khan, S. Gould, and M. Salzmann, "Deep convolutional neural networks for human embryonic cell counting," in *European Conference on Computer Vision*, Springer, 2016, pp. 339–348.

142. H. A. Kiers, "Towards a standardized notation and terminology in multiway analysis," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 105–122, 2000.

143. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

144. B. E. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, vol. 25, no. 1, pp. 117–132, 1998.

145. T. Kinnunen, K. Lee, and H. Li, "Dimension reduction of the modulation spectrogram for speaker verification," in *Proc. Odyssey*, Stellenbosch, South Africa, 2008, p. 30.

146. L. Kishon-Rabin, O. Amir, Y. Vexler, and Y. Zaltz, "Pitch discrimination: Are professional musicians better than non-musicians?" *Journal of basic and clinical physiology and pharmacology*, vol. 12, no. 2, pp. 125–144, 2001.

147. A. P. Klapuri, "Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

148. A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 255–266, 2008.

149. A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. Berlin, Heidelberg: Springer-Verlag, 2006.

150. S. Koelsch and W. A. Siebel, "Towards a neural basis of music perception," *Trends in cognitive sciences*, vol. 9, no. 12, pp. 578–584, 2005.

151. T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

152. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

153. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

154. D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, MIT Press, 2001, pp. 556–562.

155. J. H. Lee, "Crowdsourcing music similarity judgments using mechanical turk.," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 183–188.

156. A. Lefevre, F. Bach, and C. Févotte, "Itakura-saito nonnegative matrix factorization with group sparsity," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 21–24.

157. S. Leglaive, R. Hennequin, and R. Badeau, "Singing voice detection with deep recurrent neural networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 121–125.

158. B. Lehner, G. Widmer, and S. Bock, "A low-latency, real-time-capable singing voice detection method with lstm recurrent neural networks," in *Proc. European Signal Processing Conf. (EUSIPCO)*, IEEE, 2015, pp. 21–25.

159. B. Lehner, G. Widmer, and R. Sonnleitner, "On the reduction of false positives in singing voice detection," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

160. Y. Li and D. Wang, "Detecting pitch of singing voice in polyphonic audio," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, PA, USA, Mar. 2005.

161. Y. Li and D. Wang, "Singing voice separation from monaural recordings," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2006.

162. Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1475–1487, May 2007.

163. Y. Li, J. Woodruff, and D. Wang, "Monaural musical sound separation based on pitch and common amplitude modulation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1361–1371, Sep. 2009.

164. A. Liutkus and R. Badeau, "Generalized Wiener filtering with fractional power spectrograms," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015.

165. A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE Trans. on Signal Processing*, vol. 59, no. 7, pp. 3155–3167, Jul. 2011.

166. A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Feb. 2017, pp. 323–332.

167. A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 59, no. 7, pp. 3155–3167, Feb. 2011.

168. A. Liutkus, J.-L. Durrieu, L. Daudet, and G. Richard, "An overview of informed audio source separation," in *proc. International Workshop on Image Analysis for Multimedia Interactive Services*, Paris, France, Jul. 2013.

169. A. Liutkus, D. FitzGerald, and Z. Rafii, "Scalable audio separation with light kernel additive modelling," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015.

170. A. Liutkus, D. FitzGerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4298–4310, Aug. 2014.

171. A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, "Adaptive filtering for music/voice separation exploiting the repeating musical structure," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012.

172. A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Springer International Publishing, 2017, pp. 323–332.

173. B. Loesch and B. Yang, "Source number estimation and clustering for underdetermined blind source separation," in *Proc. Intl. Workshop Acoust. Echo Noise Control (IWAENC)*, 2008.

174. R. B. MacLeod, "Influences of dynamic level and pitch height on the vibrato rates and widths of violin and viola players," PhD thesis, 2006.

175. P. Magron, R. Badeau, and B. David, "Phase recovery in NMF for audio source separation: an insightful benchmark," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 81–85.

176. S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation*. Springer Netherlands, 2007.

177. E. Marchi, F. Vesperini, S. Squartini, and B. Schuller, "Deep recurrent neural network-based autoencoders for acoustic novelty detection," *Computational intelligence and neuroscience*, vol. 2017, 2017.

178. M. Markaki and Y. Stylianou, "Using modulation spectra for voice pathology detection and classification," in *Proc. EMBC*, Minneapolis, USA, Sep. 2009, pp. 2514–2517.

179. M. Marsden, K. McGuiness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," in *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2017.

180. R. Marxer, J. Janer, and J. Bonada, "Low-latency instrument separation in polyphonic audio using timbre models," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Tel Aviv, Israel, Mar. 2012.

181. M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 659–663.

182. S. McAdams, "Segregation of concurrent sounds. I: Effects of frequency modulation coherence," *The Journal of the Acoustical Society of America*, vol. 86, no. 6, pp. 2148–2159, 1989.

183. R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 34, no. 4, pp. 744–754, Aug. 1986.

184. C. E. Mcculloch and J. M. Neuhaus, "Generalized linear mixed models," in *Encyclopedia of Environmetrics*. 2006.

185. Y. Medan, E. Yair, and D. Chazan, "Super Resolution Pitch Determination of Speech Signals," *IEEE Transactions on Signal Processing*, vol. 39, no. 1, pp. 40–48, 1991.

186. M. Mellody and G. H. Wakefield, "The time-frequency characteristics of violin vibrato: Modal distribution analysis and synthesis," *Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 598–611, 2000.

187. A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, 2016.

188. A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017, pp. 85–92.

189. N. Mesgarani, S. Shamma, and M. Slaney, "Speech discrimination based on multiscale spectro-temporal modulations," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, May 2004, p. 601.

190. N. J. Miller, "Removal of noise from a voice signal by synthesis," Utah University, Tech. Rep., 1973.

191. S. Mirzaei, Y. Norouzi, and H. van Hamme, "Blind audio source counting and separation of anechoic mixtures using the multichannel complex NMF framework," *Signal Processing*, vol. 115, pp. 27–37, 2015.

192. B. Moore, *An Introduction to the Psychology of Hearing*, 3rd edition. Academic Press, 1989.

193. M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, 1st. Springer-Verlag, 2015.

194. M. Nakano, J. Le Roux, H. Kameoka, Y. Kitano, N. Ono, and S. Sagayama, "Nonnegative matrix factorization with markov-chained bases for modeling time-varying patterns in music spectrograms," in *Latent Variable Analysis and Signal Separation*, Springer, 2010, pp. 149–156.

195. A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, Sep. 2016.

196. A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016.

197. N. Ono, Z. Koldovsky, S. Miyabe, and N. Ito, "The 2013 signal separation evaluation campaign," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

198. N. Ono, K. Miyamoto, J. L. Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008.

199. N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus, "The 2015 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Liberec, Czech Republic, Aug. 2015.

200. H. Osser and F. Peng, "A cross cultural study of speech rate," *Language and Speech*, vol. 7, no. 2, pp. 120–125, 1964.

201. A. Ozerov, C. Févotte, R. Blouet, and J.-L. Durrieu, "Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.

202. A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1118–1133, 2012.

203. V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

204. R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. International Conference on Learning Representations*, Banff, AB, Canada, Apr. 2014.

205. S. Pasha, J. Donley, and C. Ritz, "Blind speaker counting in highly reverberant environments by clustering coherence features," in *Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE*, Dec. 2017.

206. D. Pavlidi, A. Griffin, M. Puigt, and A. Mouchtaris, "Source counting in real-time sound source localization using a circular microphone array," in *IEEE Signal Processing Workshop on Sensor Array and Multichannel (SAM)*, 2012, pp. 521–524.

207. M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods," in *Springer Handbook of Speech Processing*, Springer Press, Nov. 2007.

208. T. J. Pinch, F. Trocco, and T. Pinch, *Analog days: The invention and impact of the Moog synthesizer*. Harvard University Press, 2009.

209. G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A Pitch Tracking Corpus with Evaluation on Multipitch Tracking Scenario," in *Proc. Interspeech Conf.*, pp. 1509–1512.

210. F. Pishdadian and B. Pardo, "Multi-resolution common fate transform," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

211. R. Plomp, "The role of modulation in hearing," in *proc. HEARING — Physiological Bases and Psychophysics*, R. Klinke and R. Hartmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 270–276.

212. G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.

213. J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Intl. Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–6.

214. J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," *Proc. European Signal Processing Conf. (EUSIPCO)*, 2017.

215. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. Prentice Hall, 1996.

216. H. Purnhagen and N. Meine, "HILN -The MPEG-4 Parametric Audio Coding Tools," in *IEEE Int. Symposium on Circuits and Systems 2000*, IEEE, vol. 3, 2000, pp. 201–204.

217. N. C. Rabinowitz, B. D. B. Willmore, A. J. King, and J. W. H. Schnupp, "Constructing noise-invariant representations of sound in the auditory pathway," *PLOS Biology*, vol. 11, no. 11, pp. 1–18, Nov. 2013.

218. Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, Aug. 2018.

219. Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 8, pp. 1307–1335, Aug. 2018.

220. Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *Musdb18 - a corpus for music separation*, Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372.

221. Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, Oct. 2012.

222. Z. Rafii and B. Pardo, "REpeating Pattern Extraction Technique (REPET): A simple method for music/voice separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 1, pp. 73–84, Jan. 2013.

223. V. S. Ramaiah and R. R. Rao, "Speaker diarization system using HXLPS and deep neural network," *Alexandria Engineering Journal*, 2017.

224. L. A. Ramig and T. Shipp, "Comparative measures of vocal tremor and vocal vibrato," *Journal of Voice*, vol. 1, no. 2, pp. 162–167, 1987.

225. M. Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2008, pp. 1885–1888.

226. C. Raphael and Y. Han, "A classifier-based approach to score-guided music audio source separation," *Computer Music Journal*, vol. 32, no. 1, pp. 51–59, 2008.

227. I. Recommendation, "Bs. 1534-1. method for the subjective assessment of intermediate sound quality (MUSHRA)," *International Telecommunications Union, Geneva*, 2001.

228. U.-D. Reips, "Using the internet to collect data," in *APA Handbook of Research Methods in Psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological*, H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, and K. J. Sher, Eds., vol. 2, Washington, US: American Psychological Association (APA), 2012, ch. 17, pp. 291–310.

229. B. Resch, M. Nilsson, A. Ekman, and B. W. Kleijn, "Estimation of the instantaneous pitch of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 813–822, 2007.

230. S. H. Rezatofighi, V. Kumar, A. Milan, E. Abbasnejad, A. Dick, and I. Reid, "DeepSetNet: Predicting sets with deep neural networks," in *Proc. IEEE Intl. Conference on Computer Vision (ICCV)*, 2017.

231. S. Rickard and O. Yilmaz, "On the approximate w-disjoint orthogonality of speech," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, Florida, USA, May 2002.

232. H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

233. F. J. Rodriguez-Serrano, S. Ewert, P. Vera-Candeas, and M. Sandler, "A score-informed shift-invariant extension of complex matrix factorization for improving the separation of overlapped partials in music recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 61–65.

234. M. Rouvier, P.-M. Bousquet, and B. Favre, "Speaker diarization through speaker embeddings," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2015, pp. 2082–2086.

235. M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Proc. Interspeech Conf.*, 2013.

236. S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems 13*, MIT Press, 2001, pp. 793–799.

237. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, MIT Press Cambridge, 1986, pp. 318–362.

238. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

239. N. F. S. Sabour and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

240. T. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580–4584.

241. J. Salamon, "Melody extraction from polyphonic music signals," PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2013.

242. J. Salamon, E. Gomez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Mag.*, vol. 31, no. 2, pp. 118–134, 2014.

243. J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 20, 2012.

244. Y. Salaün, E. Vincent, N. Bertin, N. Souviraà-Labastie, X. Jaureguiberry, D. T. Tran, and F. Bimbot, "The flexible audio source separation toolbox version 2.0," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.

245. M. J. Salganik, P. S. Dodds, and D. J. Watts, "Experimental study of inequality and unpredictability in an artificial cultural market.," *Science (New York, N.Y.)*, vol. 311, no. 5762, pp. 854–6, Feb. 2006.

246. G. Samoradnitsky and M. Taqqu, *Stable non-Gaussian random processes: stochastic models with infinite variance*. CRC Press, 1994, vol. 1.

247. H. Sayoud and S. Ouamour, "Proposal of a new confidence parameter estimating the number of speakers-an experimental investigation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 2, pp. 101–109, 2010.

248. P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, Atlanta, GA, USA, May 1996.

249. E. D. Scheirer, "Towards music understanding without separation: Segmenting music with correlogram comodulation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 1999, pp. 99–102.

250. H. Schenker, *Harmony*. University of Chicago Press, 1954.

251. J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 44–50.

252. J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks.," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 121–126.

253. J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

254. M. Schoeffler, F.-R. Stöter, H. Bayerlein, B. Edler, and J. Herre, "An experiment about estimating the number of instruments in polyphonic music: A comparison between internet and laboratory results," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 389–394.

255. C. E. Schreiner and J. V. Urbas, "Representation of amplitude modulation in the auditory cortex of the cat. ii. comparison between cortical fields," *Hearing Research*, vol. 32, no. 1, pp. 49–63, 1988.

256. C. E. Seashore, "The natural history of the vibrato," *Proceedings of the National Academy of Sciences*, vol. 17, no. 12, pp. 623–626, 1931.

257. P. Seetharaman, F. Pishdadian, and B. Pardo, "Music/voice separation using the 2d Fourier transform," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, Oct. 2017.

258. S. Segui, O. Pujol, and J. Vitria, "Learning to count with deep object features," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 90–96.

259. M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 7398–7402.

260. C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

261. N. Shokouhi and J. H. L. Hansen, "Teager–kaiser energy operators for overlapped speech detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 5, pp. 1035–1047, May 2017.

262. K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.

263. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR (workshop track)*, 2015.

264. P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," *Lecture Notes in Comp. Science*, vol. 3195, pp. 494–499, 2004.

265. P. Smaragdis, C. Févotte, G. Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 66–75, May 2014.

266. P. Smaragdis, B. Raj, and M. V. Shashanka, "Sparse and shift-invariant feature extraction from non-negative data.," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 2069–2072.

267. P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA, Oct. 2003.

268. M. Spiertz and V. Gnann, "Source-filter based clustering for monaural blind source separation," in *Proc. Conf. on Digital Audio Effects*, Como, Italy, Sep. 2009.

269. J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR (workshop track)*, 2015.

270. F.-R. Stöter, S. Bayer, and B. Edler, "Unison source separation," in *17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

271. F.-R. Stöter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

272. F.-R. Stöter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "CountNet: Estimating the number of concurrent speakers using supervised learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 268–282, Feb. 2019.

273. F.-R. Stöter, S. Chakrabarty, B. Edler, and E. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

274. F.-R. Stöter, A. Liutkus, R. Badeau, B. Edler, and P. Magron, "Common fate model for unison source separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016.

275. F.-R. Stöter, M. Müller, and B. Edler, "Multi-sensor cello recordings for instantaneous frequency estimation," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, Brisbane, Australia, 2015, pp. 995–998.

276. F.-R. Stöter, M. Schoeffler, B. Edler, and J. Herre, "Human ability of counting the number of instruments in polyphonic music," in *Proceedings of Meetings on Acoustics*, vol. 19, 2013.

277. F.-R. Stöter, N. Werner, S. Bayer, and B. Edler, "Refining fundamental frequency estimates using time warping," in *Proceedings of EUSIPCO 2015*, Nice, France, Sep. 2015.

278. F. Stöter, S. Bayer, and B. Edler, "Unison Source Separation," in *17th Int. Conference on Digital Audio Effects (DAFx)*, 2014, pp. 235–241.

279. F.-R. Stöter, *Unison source separation dataset*, Sep. 2014. [Online]. Available: https://doi.org/10.5281/zenodo.1467921.

280. F.-R. Stöter, *CountIt - an auditory experiment to estimate the number of speakers*, Dec. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1467968.

281. F.-R. Stöter, S. Chakrabarty, E. Habets, and B. Edler, *LibriCount, a dataset for speaker count estimation*, Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1216072.

282. F.-R. Stöter, A. Liutkus, R. Badeau, B. Edler, and P. Magron, "Common fate model for unison source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, China, Mar. 2016.

283. F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Intl. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2018, pp. 293–305.

284. F.-R. Stöter, M. Müller, and B. Edler, *MUSERC: Multi-sensor cello recordings for instantaneous frequency estimation*, Oct. 2015. [Online]. Available: https://doi.org/10.5281/zenodo.1560651.

285. F.-R. Stöter and N. Werner, *SiSEC 2016 website*, Nov. 2016. [Online]. Available: https://doi.org/10.5281/zenodo.1490095.

286. N. Sturmel, A. Liutkus, J. Pinel, L. Girin, S. Marchand, G. Richard, R. Badeau, and L. Daudet, "Linear mixing models for active listening of music productions in realistic studio conditions," in *Proc. Audio Eng. Soc. Convention*, Budapest, Hungary, Apr. 2012.

287. S. Sukittanon, L. E. Atlas, and S. G. Dame, "Enhanced modulation spectrum using space-time averaging for in-building acoustic signature identification," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, May 2006, pp. III–III.

288. J. Sundberg, "Acoustic and psychoacoustic aspects of vocal vibrato," *STL-QPSR*, pp. 45–67, 1994.

289. N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, Oct. 2017.

290. D. Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," in *Speech Coding & Synthesis*, W. Klejn and K. Paliwal, Eds., Elsevier, 1995, pp. 495–518.

291. D. Tidhar, M. Mauch, and S. Dixon, "High precision frequency estimation for harpsichord tuning classification," in *2010 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2010, pp. 61–64.

292. A. Trask, F. Hill, S. Reed, J. Rae, C. Dyer, and P. Blunsom, "Neural arithmetic logic units," *arXiv preprint arXiv:1808.00508*, 2018.

293. S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 2135–2139.

294. S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017.

295. A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio.," in *SSW*, 2016, p. 125.

296. A. Vaneph, E. McNeil, and F. Rigaud, "An automated source separation technology and its practical applications," in *Proc. Audio Eng. Soc. Convention*, Paris, France, May 2016.

297. A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

298. S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, London, UK, Sep. 2005.

299. E. Vincent, S. Araki, and P. Bofill, "The 2008 signal separation evaluation campaign: A community-based approach to large-scale evaluation," in *8th International Conference on Independent Component Analysis and Signal Separation*, Paraty, Brazil, Mar. 2009.

300. E. Vincent, S. Araki, F. J. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. Duong, "The signal separation evaluation campaign (2007-2010): Achievements and remaining challenges," *Signal Processing*, vol. 92, no. 8, pp. 1928–1936, Aug. 2012.

301. E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

302. E. Vincent, T. Virtanen, and S. Gannot, *Audio source separation and speech enhancement*. Wiley, Aug. 2018, p. 504.

303. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Intl. Conference on Machine Learning (ICML)*, ACM, 2008, pp. 1096–1103.

304. M. Vinyes, *MTG MASS database*, http://www.mtg.upf.edu/static/mass/resources, 2008.

305. T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, Jun. 2000, II765–II768 vol.2.

306. T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, 2007.

307. T. Virtanen, A. Mesaros, and M. Ryynänen, "Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music," in *proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition*, Brisbane, Australia, Sep. 2008.

308. H. Viste and G. Evangelista, "Separation of harmonic instruments with overlapping partials in multi-channel mixtures," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2003, pp. 25–28.

309. O. Walter, L. Drude, and R. Haeb-Umbach, "Source counting in speech mixtures by nonparametric bayesian estimation of an infinite Gaussian mixture model," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 459–463.

310. A. L. Wang, "Instantaneous and frequency-warped techniques for auditory source separation," PhD thesis, Stanford University, 1994.

311. A. L. Wang, "Instantaneous and frequency-warped techniques for source separation and signal parametrization," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA, Oct. 1995.

312. C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proc. ACM Intl. Conference on Multimedia (ACMMM)*, 2015, pp. 1299–1302.

313. D. Wang, X. Zhang, and Z. Zhang, *THCHS-30: A free chinese speech corpus*, 2015. [Online]. Available: http://arxiv.org/abs/1512.01882.

314. D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*, Springer, 2005, pp. 181–197.

315. N. Welch and J. H. Krantz, "The world-wide web as a medium for psychoacoustical demonstrations and experiments: experience and results," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 192–196, Jun. 1996.

316. J. Woodhouse and P. Galluzzo, "The bowed string as we know it today," *ACTA Acustica united with Acustica*, vol. 90, no. 4, pp. 579–589, 2004.

317. J. Woodhouse and A. Loach, "Torsional behaviour of cello strings," *Acta Acustica united with Acustica*, vol. 85, pp. 734–740, Sep. 1999.

318. M. Wu, D. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 11, no. 3, pp. 229–241, May 2003.

319. D. Wulich, E. I. Plotkin, M. N. S. Swamy, and W. Tong, "Pll synchronized time-varying constrained notch filter for retrieving a weak multiple sine signal jammed by fm interference," *IEEE Transactions on Signal Processing*, vol. 40, no. 11, pp. 2866–2870, Nov. 1992.

320. C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in *Proc. of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, ACM, 2013, pp. 43–52.

321. S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2014, pp. 402–406.

322. F. Yen, M.-C. Huang, and T.-S. Chi, "A two-stage singing voice separation algorithm using spectro-temporal modulation features," in *Proc. Interspeech Conf.*, Dresden, Germany, Sep. 2015.

323. F. Yen, Y.-J. Luo, and T.-S. Chi, "Singing voice separation using spectro-temporal modulation features," in *Proc. Intl. Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, Oct. 2014.

324. R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast TV using bidirectional long short-term memory networks," in *Proc. Interspeech Conf.*, ISCA, 2017, pp. 3827–3831.

325. D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

326. C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 833–841.

327. J. Zhang, S. Ma, M. Sameki, S. Sclaroff, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech, "Salient object subitizing," in *Proc. Intl. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4045–4054.

328. E. Zwicker, "Die Grenzen der Hörbarkeit der Amplitudenmodulation und der Frequenzmodulation eines Tones," *Acta Acustica United With Acustica*, vol. 2, no. 3, pp. 125–133, 1952.

329. E. Zwicker and H. Fastl, *Psychoacoustics: Facts and models*. Springer-Verlag Berlin Heidelberg, 2013.

Fabian-Robert Stöter: *Separation and Count Estimation for Audio Sources Overlapping in Time and Frequency*, Dissertation, © 7. April 2019
Final Version as of January 24, 2020 (commit 58f3e24).