Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), pages 9–20
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Eidos: An Open-Source Auditory Periphery Modeling Toolkit and Evaluation of Cross-Lingual Phonemic Contrasts

Alexander Gutkin
Google Research
6 Pancras Square, London, N1C 4AG, United Kingdom
agutkin@google.com

Abstract

Many analytical models that mimic, in varying degree of detail, the basic auditory processes involved in human hearing have been developed over the past decades. While the auditory periphery mechanisms responsible for transducing the sound pressure wave into the auditory nerve discharge are relatively well understood, the models that describe them are usually very complex because they try to faithfully simulate the behavior of several functionally distinct biological units involved in hearing. Because of this, there is a relative scarcity of toolkits that support combining publicly-available auditory models from multiple sources. We address this shortcoming by presenting an open-source auditory toolkit that integrates multiple models of various stages of human auditory processing into a simple and easily configurable pipeline, which supports easy switching between ten available models. The auditory representations that the pipeline produces can serve as machine learning features and provide analytical benchmark for comparing against auditory filters learned from the data. Given a low- and high-resource language pair, we evaluate several auditory representations on a simple multilingual phonemic contrast task to determine whether contrasts that are meaningful within a language are also empirically robust across languages.

Keywords: open-source, auditory perception, features, software, phonology

1. Introduction

The science of hearing is an interdisciplinary field that studies the perception of sound (Schnupp et al., 2011), including speech (Moore, 2007; Young, 2007). It places a particular emphasis on studying the function of the auditory periphery, defined between the point where the sound pressure wave meets the ear and the auditory nerve (AN). This region is thought to be of critical importance because it converts continuous analog signal into discrete all-or-nothing nerve action potentials. A simplified description of this mechanism consists of several complex processing stages: the pressure wave causes vibration of the eardrum, which is passed to the cochlea. Inside the cochlea the basilar membrane (BM) responds with tuned vibrations that are further modified by the cochlear amplification feedback mechanism provided by the outer hair cells (OHCs) (LeMasurier and Gillespie, 2005). The BM motion is detected by inner hair cells (IHCs) that transduce it into electric receptor potentials that control the generation of action potentials in the AN fibers converging on the IHCs through the release of a neurotransmitter into the AN synaptic cleft (Meddis et al., 2010; Manley et al., 2017).

Many models that approximate the functioning of the human auditory periphery to varying degrees of detail have been developed over the decades. Comprehensive reviews of the most popular ones are provided in (Lopez-Poveda, 2005; Rudnicki et al., 2015; Saremi et al., 2016; Verhulst et al., 2018). The models range from phenomenological models that reproduce the overall auditory input-output relation by employing filterbanks (Meddis et al., 2010; Lyon, 2017), often implemented in hardware (Freedman et al., 2013), or transmission lines (Verhulst et al., 2012), to detailed biophysical models (Bell, 2012; Corey et al., 2017). Among many applications, such as cochlear implants (Tabibi et al., 2017), the phenomenological models of human auditory periphery were shown to benefit the automatic speech recognition (ASR) systems in noisy conditions (Hemmert et al., 2004; Harczos et al., 2007; Tjandra et al., 2015; Li and Príncipe, 2018; Pan et al., 2018), improve neural network-based speech enhancement (Baby and Verhulst, 2018) and provide high-quality text-to-speech vocoding (Irino et al., 2006).

The success of the end-to-end approaches to ASR (Tjandra et al., 2017; Zeyer et al., 2018; Zeghidour et al., 2018) was facilitated by the observation that traditional fixed front-ends can be replaced by the feature extractors learned from data by joint optimization with the rest of the deep network architecture (Sainath et al., 2015; Ghahremani et al., 2016). The learned representations outperform the traditional fixed features on many tasks (Zeghidour, 2019). This led some to question the relevance of traditional approaches that handcraft valuable prior knowledge (Trigeorgis et al., 2016). However, the properties of the band pass-like filters learned by deep networks roughly correspond to human audio-biological distribution (Tüske et al., 2014) and the recent study by Ondel et al. (2019) demonstrates that human auditory processing and data-driven methods are not necessarily as divergent as they would often appear. In a machine learning context, this observation leads us to believe that the powerful analytical models developed by hearing science, including the ones provided by the toolkit described in this paper, are still very useful for informing model design and explaining the structure of the representations learned by black box-like data-driven approaches. Moreover, analytical models of human hearing may provide useful insights in low and zero-resource speech and language learning scenarios (Dupoux, 2018), where the data scarcity can potentially be alleviated by employing models incorporating rich prior knowledge.

This paper describes EIDOS, an open-source toolkit[1] con-

[1] `https://github.com/google/eidos-audition`

Name	Period	Language
Auditory Toolbox	1982–1993	MATLAB
HUTear	1991–1998	MATLAB, C/MEX
DSAM	1986–2007	C
AMToolbox	1979–2017	MATLAB, C/MEX
Brian Hears	1987–2003	Python
Cochlea	2007–2014	Python, Cython
UR EAR	2004–2018	MATLAB, C/MEX

Table 1: Auditory model collections.

taining the collection of various auditory perception models developed over the years by hearing scientists. The original models were developed by diverse research groups using different programming languages and software design approaches. In this work the models have been reimplemented entirely in modern C++ and integrated into a single easily configurable and simple to use pipeline that represents all critical stages of the auditory periphery from the BM to the AN. The pipeline produces auditory representations that are easy to integrate with the popular Python machine learning toolkits (Abadi et al., 2016; Paszke et al., 2019).

We evaluate the quality of several auditory representations offered by our toolkit on a cross-lingual phonemic contrast detection task recently introduced by Johny et al. (2019). The method involves training separate binary neural classifiers for several phonological contrasts, defined in terms of phonological features, in audio spans centered on particular segments within continuous speech. To assess cross-linguistic consistency, these classifiers are evaluated on held-out languages and classification quality is reported. More often than not, phoneme inventories and their corresponding phonemic featurizations are provided by cross-linguistic typological ontologies, such as PHOIBLE (Moran et al., 2014). This approach can be used to test how accurately such phoneme inventories for low- or zero-resource languages describe the speech data at hand.

This paper is organized as follows: various auditory toolkits developed over the years are presented in Section 2. Section 3 provides an overview of the auditory models currently supported by our software. The core library design features and basic usage examples are presented in Section 4. In Section 5 we evaluate and discuss several audio representations on the phonemic contrast task. Section 6 concludes the paper.

2. Related Work

Several collections of auditory models have been developed over the years. The list of such collections that we review below is in no way complete, but covers some of the most popular and widely used pipelines for auditory processing that support combining models from various sources. This review does not cover the initiatives undertaken in specialized areas of auditory perception, such as binaural processing, for which excellent reviews exist (Dietz et al., 2018).

The list of the auditory model collections of interest is shown in Table 1 where, along with the name of each toolbox, the years when the models were published are shown along with the programming language(s) used to implement them. The toolboxes vary among several dimensions. One dimension is the year when the models were first published, with the Auditory Toolbox being the oldest among toolkits covered. An additional dimension is the number of supported models, with the AMToolbox and DSAM being the most comprehensive in the list. A further dimension reflects the design philosophy, with some of the toolboxes focusing of reproducibility (AMToolbox and Cochlea), others designed for efficiency (Brian Hears and DSAM), or both.

Auditory Toolbox One of the earliest and arguably the most widely used collections of auditory models is the Auditory Toolbox by Slaney (1998). The toolbox is a MATLAB (Pärt-Enander et al., 1996; Moore, 2017) reimplementation of the earlier package (Slaney, 1988) written in Mathematica (Wolfram, 1999) and includes public-domain implementations of several classical machine perception algorithms from the early days of the field: the cochlear model by Lyon (1982) that combines a series of filters that model traveling pressure waves with Half Wave Rectifiers (HWR) to detect the energy in the signal and several stages of Automatic Gain Control (AGC), the cochlear model by Seneff (1988) that combines a critical band filterbank with models of detection and AGC, and the original hair cell model by Meddis (1986) using the physiological AN parameters from Meddis et al. (1990). Finally, the toolbox includes the implementation of gammatone filterbank (Johannesma, 1972) – a model of psychoacoustic filtering (Moore and Glasberg, 1983; Glasberg and Moore, 1990) based on critical bands originally proposed by Roy Patterson (Patterson, 1986; Slaney, 1993). Most of these implementations have been widely used in many auditory processing scenarios, reimplemented in various programming languages and have made their way into other software.

HUTear This toolbox was developed in the Laboratory of Acoustics and Audio Signal Processing at Helsinki University of Technology (Härmä and Palomäki, 2000). It is implemented in MATLAB with the performance critical algorithms written in C/MEX. In addition to some popular algorithms from the Auditory Toolbox, such as the Meddis IHC model, the toolbox provides the original implementations, such as the quantitative signal preprocessing and detector model by Dau et al. (1996) and an auditory model by Karajalainen (1996). This software is distributed under an attribution license.

DSAM The Development System for Auditory Modelling (DSAM) is a computational library designed specifically for producing time-sampled auditory system simulations. It was originally developed by Lowel P. O'Mard from the Centre for the Neural Basis of Hearing (CNBH) as a joint collaboration between University of Essex and University of Cambridge (O'Mard, 2010). Implemented entirely in C programming language, this library brings together many established auditory models under a flexible programming platform. The latest DSAM version includes eight BM response models, such as gammatone filterbank and the dual-resonance nonlinear (DRNL) filter by Lopez-Poveda and Meddis (2001), seven hair cell models, such as the IHC synaptic model by Carney (1993), AN spike generation and neuron firing models, such as the cochlear nucleus neuron model by Arle and Kim (1991), many utility and analysis facilities, multi-threading, as well as the

support for most sound file formats. DSAM provides implementations for most of the algorithms from the Auditory Toolbox by Slaney (1998) described above and is licensed under the version 3 of GNU General Public License (GPL).

AMToolbox The Auditory Modeling Toolbox (AMToolbox) is likely the largest model collection for representing various stages of auditory perception (Søndergaard and Majdak, 2013). Implemented in MATLAB/Octave this toolbox is intended to serve as a common ground for reproducible research in auditory modeling. Similar to DSAM, AMToolbox is not just the collection of models, all the models are implemented using strict requirements on model interfaces. Unlike DSAM, AMToolbox also provides a comprehensive testsuite and verification guidelines to ensure that the implementations produce the results that match the results reported in the literature. Furthermore, the toolbox provides published human data and model demonstrations. The AMToolbox is maintained by Acoustic Research Institute (ARI) of Austrian Academy of Sciences, with project being supported by multiple universities across Europe and the United States. The toolbox incorporates 29 algorithms which, unlike the other software described so far, also puts an emphasis on models of binaural and spatial perception, although other types of models, such as the cochlear model of the auditory periphery by Verhulst et al. (2012), are provided as well. The toolbox is distributed under the version 3 of GNU General Public License (GPL).

Brian Hears This package is an auditory toolbox (Fontaine et al., 2011) developed in the Python programming language for the spiking neural network simulator framework called "Brian" (Goodman and Brette, 2009). Integration with Brian makes it possible to model the auditory neurons higher up in the auditory perception chain. The salient feature of the design is vectorization, an algorithmic strategy that consists in grouping identical operations operating on different data. In the context of auditory modeling, vectorization happens over the frequency channels, which makes it possible to take advantage of heavily parallel architecture of auditory models that exclusively rely on filterbanks. This greatly improves the efficiency of the implementation which otherwise relies on an interpretable language. The toolbox supports several filterbank-based models, such as gammatone, DRNL, gammachirp filter by Irino and Patterson (1997) and middle ear model by Tan and Carney (2003). In addition, modular filter design allows multiple filters to be combined efficiently to form new models. The package is distributed under CeCILL (from "CEA CNRS INRIA Logiciel Libre") Free Software License.

Cochlea This Python package contains a small collection of models of auditory periphery created by Rudnicki and Hemmert (2014). The package allows researchers to run and analyze a selection of three inner ear models, such as the algorithm by Holmberg et al. (2007), which generate AN spike trains from arbitrary sound signals. The design rationale for this package is similar to AMToolbox in that it makes it easy to run different models and to analyze and compare them with the same methods (Rudnicki et al., 2015). The package is distributed under the version 3 of GNU General Public License (GPL).

Model	Auditory Stage			
	BM	HC	Synapse	Spikes
GAMMATONE-SLANEY	✓			
GAMMATONE-COOKE	✓			
MEDDIS1986			✓	
BAUMGARTE	✓	✓		
SUMNER2002			✓	
CARFAC	✓	✓	✓	
ZILANY2014		✓		
BRUCE2018			✓	✓
ZHANG2001				✓
JACKSON				✓

Table 2: Supported auditory models and their estimates.

UR EAR "University of Rochester: Envisioning Auditory Responses" (UR EAR) is a MATLAB package with a graphical user interface designed to run various AN models and the higher-level auditory pathway models, such as inferior colliculus (IC), developed over the years by researchers in the University of Rochester, McMaster University and their collaborators (Farhadi and Carney, 2019). Computation intensive models, such as the IHC and the AN models by Bruce et al. (2018), are implemented in MEX, which is an environment for interoperability between the C functions and MATLAB. Models provided by this package are replaced whenever a new research finding results in a revised version of the existing model.

3. Overview of Supported Models

The list of auditory models currently supported by the toolkit is shown in Table 2. The models come from miscellaneous sources and differ along several dimensions, the primary of which is the stage of auditory perception that each model estimates: the BM, the hair cells (HC) and the synapse between the IHC and the AN. In addition, several models provide the estimates of AN discharge (also known as spike generation). Some models, such as CARFAC, produce estimates for several auditory stages, while others are highly detailed and focused on a single stage only.

Gammatone Filterbanks Historically, the gammatone filterbanks have perhaps been the most widely used abstractions for modeling the human auditory system. They are often used as a front-end component of the cochlear models, decomposing the stimulus into multi-channel components mimicking the function of human cochlea. The output of each filter in a filterbank estimates the BM frequency response at a particular place corresponding to the center frequency of the filter.

The gammatone function is defined in time domain by its impulse response that is a product of a gamma distribution and periodic tone (Johannesma, 1972). The gammatone implementation GAMMATONE-SLANEY provided by this toolbox follows the original version developed in (Slaney, 1998), which uses the findings of Patterson et al. (1992) who showed that the fourth-order gammatone function produces an impulse response that provides a good fit to the human auditory filter shapes proposed by Patterson (1986). Gammatone filterbank is a collection of gammatone filter functions where the filters are designed in such a way that their center frequencies are distributed across frequency range in proportion to their bandwidth according to the Equivalent Rectangular Bandwidth (ERB) scale

described in (Moore and Glasberg, 1983; Glasberg and Moore, 1990). Our implementation uses the ERB approximation from Glasberg and Moore (1990). The second gammatone filterbank GAMMATONE-COOKE provided by this toolbox derives from the implementation by Ma et al. (2007) of the original idea of Cooke (1993), also mentioned in (Slaney, 1993), who used base-band impulse invariant transformations to dramatically improve the speed efficiency of the original gammatone filterbank algorithm.

MEDDIS1986 The model proposed by Meddis (1986) represents one of the most popular models of mechanical-to-neural transduction that is performed between the HCs and the AN synapse. The model MEDDIS1986 is specified in terms of the production, movement, and dissipation of a transmitter substance in the region between the HC and the AN fiber synapse (Meddis, 1988). Briefly, the HC contains a quantity of the "free transmitter" $q(t)$, which leaks through a permeable membrane into the synaptic cleft. The permeability $k(t)$ fluctuates as a function of the instantaneous amplitude of the mechanical stimulation $s(t)$, provided by the BM model, such as GAMMATONE-SLANEY. The synaptic cleft contains a fluctuating amount of transmitter substance $c(t)$, part of it being continuously being returned to the cell and part of it continuously being lost. The cleft transmitter level $c(t)$ relates to the free transmitter quantity $q(t)$ and permeability $k(t)$ via a system of differential equations. The constant model parameters (such as replenishment rate) corresponding to physiological observations are provided in (Meddis et al., 1990). In this model, there is a linear relationship between the instantaneous value of the transmitter quanta $c(t)$ and the post-synaptic excitation potential: the greater the quantity of the transmitter, the higher the probability of a spike.

BAUMGARTE While historically the gammatone filterbanks have provided a reasonable tradeoff between computation efficiency and physiological accuracy, alternative, more detailed models of peripheral sound processing were developed as well. The BAUMGARTE is a peripheral ear model proposed by Baumgarte (2000) which originates from the hardware analog model of Zwicker (1986) and its extensions provided by Peisl (Peisl, 1990; Zwicker and Peisl, 1990). This model includes components that model the outer, middle and inner ear structures. Both outer and middle ears are treated as reasonably simple linear filters. The inner ear model involves nonlinear mechanical filtering, which simulates the passive cochlear hydromechanics enhanced by the feedback from active OHCs providing cochlear amplification. The implementation is a one-dimensional macromechanical model, in which the length of a BM is divided into sections of equal length on a Bark scale (Zwicker, 1961) and described by a system of coupled differential equations, one equation per section. Each equation also integrates an amplifier with nonlinear feedback that models the effect of OHCs. The parallel resonance of each section is tuned to the BM resonance at the location represented by that section (Baumgarte, 1997). The system of coupled differential equations is internally represented as a collection of equivalent electrical circuits which are simulated in the time domain by a system of wave digital filters (WDF) (Fettweis, 1986). The model is capable of outputting the estimates (represented in terms of voltage) of local transverse velocity of the BM as well as the excitation of the IHCs located along its length.

SUMNER2002 This computational model of the IHC and the AN complex (Sumner et al., 2002) is a modernized version of the earlier IHC models, such as MEDDIS1986 model described above. According to the authors, the purpose of this model was to generate an accurate representation of the input–output characteristics of the HC for arbitrary stimuli. The main differences of this model with MEDDIS1986 are as follows: First, the model proposed by Sumner et al. (2002) takes into account populations of medium (MSR) and low (LSR) spontaneous rate fibers in addition to high (HSR) spontaneous rate ones traditionally considered by the earlier models.[2] Second, the model incorporates a modified version of the transduction of BM motion into receptor potentials originally developed by Shamma et al. (1986). In addition, the transmitter release rate, or permeability, $k(t)$ of the original MEDDIS1986 model is made more sophisticated by taking into account the model of calcium concentration. Finally, in SUMNER2002 model the release of transmitter into the cleft is described by a random process $N(n, \rho)$ describing probabilistic transport of transmitter quanta. Each of n possible events has an equal probability, ρdt, of occurring in a single simulation epoch (Sumner et al., 2003).

CARFAC The cascade of asymmetric resonators with fast-acting compression (CARFAC) model is based on a pole–zero filter cascade (PZFC) model of auditory filtering combined with a multi-time-scale coupled automatic gain control (AGC) network (Lyon, 2011). The model differs from other cochlear models in its application of cascaded, rather than parallel, filterbank which is well suited for modeling the traveling waves in the cochlea. In a PZFC filterbank, each filter stage models a segment of a non-uniform distributed system corresponding to a single section along the cochlear partition. The stage transfer function is a pole–zero approximation to the transfer function corresponding to the local complex wavenumber (Lyon, 2011; Lyon, 2017). The PZFC stages provide a variable peak gain via a variable pole damping. The pole damping is adjusted by slowly varying feedback control signals from the automatic gain control (AGC) smoothing network that mimics the feedback from the OHCs. The overall architecture is very efficient, with several existing hardware implementations (Thakur et al., 2014; Singh et al., 2018; Xu et al., 2018). Our implementation provides a facade over an existing open-source CARFAC library (Lyon, 2011). The model is capable of producing estimates of BM displacements, OHC control signals and neural activity patterns (NAPs). According to Lyon (2017), the NAPs can be used as an estimate of average instantaneous AN firing rates.

ZILANY2014 This particular model is derived from a model of the auditory periphery developed by Zilany et al. (2014), which is a more physiologically accurate version of the earlier phenomenological models of the synapse between the IHC and AN developed by the researchers at

[2] Our implementation only supports HSR fibers at the moment.

McMaster and Rochester Universities (Zilany et al., 2009; Ibrahim and Bruce, 2010; Zilany et al., 2013). In this toolbox, ZILANY2014 provides the component that forms the front-end of the original model corresponding to the models of the middle ear (ME) and the IHC originally described by Zilany et al. (2009). The input to the ME is an instantaneous pressure waveform of the stimulus (in pascals) sampled at 100 kHz. The ME filter is followed by three parallel filter paths: with $C1$ and $C2$ filters in the signal path and a broad-band filter in the control-path. The combined response of the two transduction functions following the $C1$ and $C2$ filters provides the input to a IHC low-pass filter the output of which can drive the IHC-AN models. The parameters of the filters used in ZILANY2014 are adjusted according to the BM tuning values described by Ibrahim and Bruce (2010) and parameters for fitting the model to the human data based on (Glasberg and Moore, 1990; Greenwood, 1990; Pascal et al., 1998; Shera et al., 2002) are provided as well.

BRUCE2018 A phenomenological model of the synapse between the IHC and the AN by Bruce et al. (2018) retains the ZILANY2014 model described above as the peripheral front-end that drives the synapse. The synapse and the spike generation model provided by the BRUCE2018 model improves upon the original approach described in (Zilany et al., 2009; Zilany et al., 2014). The first component in the model is a gently saturating nonlinearity followed by a model of power-law dynamics using two parallel paths, fast and slow. Power-law adaptation describes an adaptation process of the AN fibers to the varying stimuli that continues to adapt no matter the length of the stimulus rather than having fixed time constants (Drew and Abbott, 2006). This presynaptic adaptation portion of the model also includes the fractional Gaussian noise (fGn) model by Jackson and Carney (2005) that takes into account the observation that the spiking probability fluctuates over time and depends on the long-term history of spike times. The synaptic portion of the model includes a detailed adaptive model of neurotransmitter release and replenishment at the four synaptic vesicle docking sites, described in detail by Bruce et al. (2018), who report this model to be a more accurate fit for the available physiological data and describe the improvements in several measures of AN fiber spiking statistics.

Spike Generation In addition to the BRUCE2018 synaptic model described above that includes a spike generation component, this toolbox supports two further spike generation algorithms. The first spike generation algorithm ZHANG2001 is part of a phenomenological model for the responses of AN fibers developed by Zhang et al. (2001). In this model the discharge times are produced by a renewal process that simulates a nonhomogeneous Poisson process driven by the output from the synapse. The time-dependent arrival rate of the Poisson process $R(t)$ is defined via the synapse output $s(t)$ and the synapse discharge history $H(t)$, modeled by two exponentials, per Westerman and Smith (1988). The parameters for refractoriness and other constants were adjusted by the authors to fit the physiological data. The other spike generator JACKSON provided by this toolbox is derived from the original implementation by Jackson (2007). In terms of its function this generator is very similar to the ZHANG2001 discharge model but is significantly more computational efficient due to two main performance improvements: the use of the time-transformation method for simulating a nonhomogeneous Poisson process, as described by Jackson and Carney (2005), and avoiding the computation of the relative refractory ratio from scratch at each time bin by using running approximations to the differential equations of which the exponentials in the relative refractory equation are solutions (Jackson, 2007).

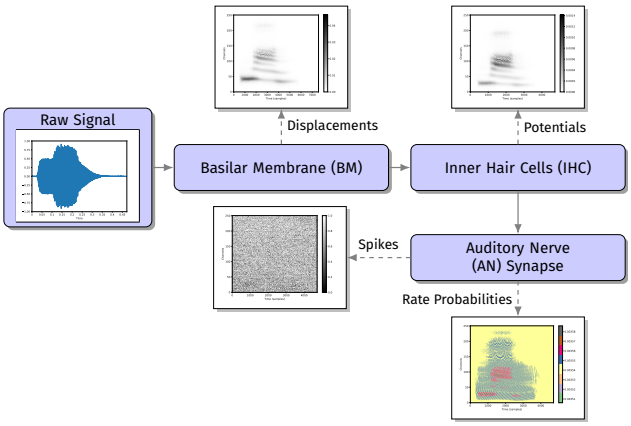

Figure 1: Schematic depiction of an auditory pipeline.

4. Design Features and Usage

Our toolkit consists of an engine and the corresponding tools for running the auditory pipeline over the supplied stimulus, storing the response (we refer to this process as *feature extraction*) and visualizing it. The engine and feature extraction are written in C++, while the visualization component is implemented in Python.

The engine defines the necessary interfaces for implementing the auditory pipeline which can be thought of as a sequence of auditory stages, or models, shown in Table 2. Each model is capable of outputting responses of one or more types and the implementation logic ensures that each model can receive the inputs of the valid type from the previous stage in the pipeline. For example, the BM displacements can be used to estimate the transmembrane potentials across the IHC, but not the other way around. A simplified depiction of the auditory pipeline, with the outputs of each stage visualized, is shown in Figure 1. The models used in this pipeline include BAUMGARTE model for estimating the BM displacements and IHC response, the SUMNER2002 model for the AN synapse rate probabilities and the spike discharge estimates from 2000 HSR fibers from the ZHANG2001 model.

The toolbox is structured along simple lines depicted in Figure 2. There are three main directories. The directory `build` contains the necessary scaffolding for building and testing the dependencies. The directory `audition` contains the pipeline and configuration parser implementations, as well as main tools and model tests. In addition, this directory contains the CARFAC model (denoted by a blue oval shape in Figure 2). Finally, the `third_party` directory houses most of the auditory model implementations.

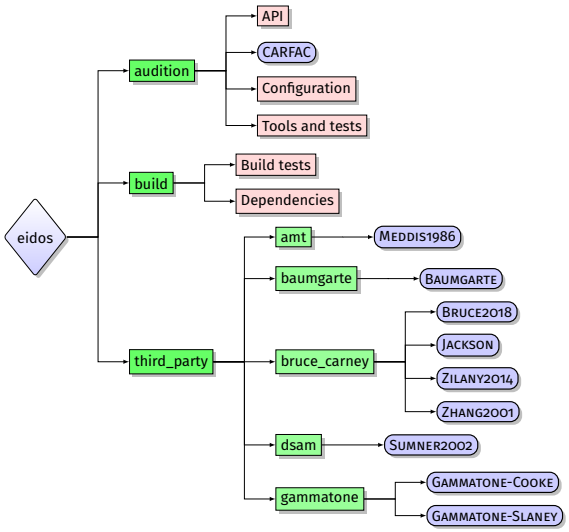

Figure 2: Toolbox directory layout.

```
# Download the toolbox.
git clone https://github.com/google/eidos-audition.git
cd eidos-audition
# Build the libraries, tools and tests.
bazel build -c opt ...
# Run the tests.
bazel test -c opt ...
```

Table 3: Setting up the toolbox.

It contains five subdirectories, whose names correspond to either the original name of an author, the university lab or name of the toolkit where the original implementation is found. For example, the MEDDIS1986 model is located under the `amt` directory which stands for Auditory Modeling Toolbox, while the subdirectory `bruce_carney` houses several models developed by Ian Bruce, Laurel Carney and their colleagues in their respective university labs. In addition to the code, each subdirectory contains the `LICENSE` file that contains the license for the original implementation from which our code derives. Further information about the models can be found in `README` files accompanying the code.

The toolbox is hosted on GitHub open-source repository that uses Git version control system (Blischak et al., 2016). The toolbox uses the flexible Bazel framework (Google, 2019) for building its dependencies, libraries, and tools in both C++ and Python. Bazel is also used to build and invoke the unit and integration tests which are implemented using Google Testing and Mocking Framework (Google, 2010). The sequence of steps for fetching the toolbox, building it and running the provided unit and integration tests is shown in Table 3.

As was mentioned above, the toolbox consists of three functional components: the pipeline, the feature extractor and the visualizer. The auditory pipeline is implemented using Google Protocol Buffers, which is a platform and language-neutral framework for serializing structured data (Google, 2008). The added benefit of using the protocol buffers is a simple and flexible configuration language and the built-in parser for instantiating the pipeline from its textual specification. Given the auditory stimulus (in PCM RIFF format) and a pipeline configuration provided

```
# Compute BM response.
DIR=eidos/audition
cd eidos-audition
bazel-bin/${DIR}/auditory_feature_extractor --helpshort
bazel-bin/${DIR}/auditory_feature_extractor \
  --waveform_file test.wav \
  --config_proto_file ${DIR}/configs/bm_carfac.textproto \
  --output_file bm.npy --num_channels 251
# Visualize the response.
bazel-bin/${DIR}/visualize_auditory_signals \
  --input_signal_file bm.npy --color_map gray_r
```

Table 4: Example: Computing the BM response.

as a simple string list of model names or in a protocol buffer format, the `auditory_feature_extractor` utility can be used to instantiate the pipeline and process the provided stimulus, storing the outputs in NumPy numeric format (Van der Walt et al., 2011) suitable for processing by various machine learning toolkits in Python (Pedregosa et al., 2011; Abadi et al., 2016; Paszke et al., 2019). An output of the last stage only can be saved by providing the `npy` extension for the output file name. The outputs from all auditory stages in the pipeline can be saved by providing the `npz` extension. Finally, the outputs can be visualized using the `visualize_auditory_signals` utility. This process is demonstrated in Table 4, where a stimulus provided in `test.wav` is processed using the pipeline in `bm_carfac.textproto` consisting of a single CARFAC model configured to provide the BM response only.

Since this toolkit provides different front-end models for processing the audio stimuli for converting them into the estimates of BM displacement, these implementations impose different requirements on the input signal. The CARFAC and GAMMATONE models can process signals at 16 kHz sampling rate and above, while the BAUMGARTE model requires a signal with a sampling rate of 100 kHz. The ZHANG2001 model can produce accurate middle-ear estimates for sampling rates between 100 kHz and 500 kHz. Providing an input at the sampling rate that a model cannot process will result in an error message, as the current version of the toolkit does not support automatic resampling.

Licensing Because the toolkit contains work derived from the original implementations available under different licenses, we chose to distribute the software under open-source version 3 of GNU General Public License (GPL), which is the most restrictive (in terms of commercial use) license among the original algorithms.

5. Experiments

Previously, we introduced the methodology for evaluating the cross-lingual consistency of phonological features in a multilingual setting (Johny et al., 2019). Whether grounded in acoustic, articulatory or phonological process properties, phonological features are the recurrent elementary components that form the sound systems of world's languages and describe the individual phonemes in a succinct way (Clements, 2009). We hypothesized that in order to consider a phonemic contrast to be consistent or robust across languages, it needs to be easily predicted on held-out languages. We performed classification experiments on a wide range of phonemic contrasts in multiple languages. Here, we focus on similar experiments on a smaller number

of tasks on different data, but using a much wider range of acoustic features provided by our toolkit.

Problem Formulation A particular phonemic contrast is presented as a binary classification problem. An instance of this problem consists of a span of a speech signal (e.g. a vowel in surrounding context) and a positive or negative label (e.g. front vowel vs. back vowel). We train a classifier on a (possibly multi-speaker) dataset for one language and hold out another language. We then evaluate the trained classifier on the held-out data and report its quality in terms of Area Under (resp. Over) the receiver operating characteristic Curve (AUC, resp. AOC). If the binary contrast in question is cross-linguistically consistent, we expect it to be readily predictable on a held-out language (Johny et al., 2019).

We focus our experiments on the Bengali and Spanish language pair that demonstrates really well the subtle confounding factor, which is due to well-known mismatches in how different languages group allophones under different phonemes. The aspiration is contrastive in Bengali, but not in Spanish. In Bengali, the phoneme /p/ (unaspirated) contrasts with an aspirated phoneme, which has [p^h] and [f] as allophones (our Bengali corpus uses /f/ as the phoneme label). In Spanish, the phoneme /p/ is unmarked for aspiration and could be realized as [p^h], which contrasts with the phoneme /f/. That means in a given multilingual dataset we may find [f] and [p^h] sounds labeled differently depending on language, because we are working with phonemic rather than phonetic transcriptions.

The first experiment, denoted P-F, deals with classifying the phonemic contrast between the labial phonemes /p/ (positive class) and /f/ (negative class) across Bengali and Spanish. This experiment is interesting because it validates the robustness of phonemic labels /p/ and /f/ in the presence of conflicting allophone [p^h] mentioned above. The second experiment, denoted VOICED, deals with classifying the voicing contrast between the labial sets { /p/, /f/ } (positive class) and their voiced counterparts { /b/, /b^h/ } (negative class). Each experiment has four possible configurations: training and testing on disjoint sets of the same language (`bn–bn` and `es–es`), and training on one language while testing on a heldout language (`bn–es` and `es–bn`).

Corpora Details For our experiments, we used a proprietary high-quality corpus of Castilian Spanish from a single female speaker that consists of around 20,000 utterances and a crowd-sourced multi-speaker corpus of Bengali (as spoken in India) that includes around 8,000 utterances from 23 female volunteer speakers. The original audio for both languages was recorded at 48 kHz. The speech data was downsampled to 16 kHz and then parameterized into HTK-style Mel Frequency Cepstral Coefficients (MFCCs) (Ganchev et al., 2005) using a 10 ms frame shift. The dimension of the MFCC parameters is 39 (13 static + Δ + $\Delta\Delta$ coefficients). To determine the phoneme time boundaries, the MFCCs were force-aligned with the corresponding transcriptions independently for each language (Young et al., 2006).

Acoustic Representations We chose four auditory representations provided by our toolkit for the experiments: the two BM displacement measurements provided by CARFAC and GAMMATONE-SLANEY models, and the measurements of the IHC transmembrane potentials provided by BAUMGARTE and ZILANY2014 models that were introduced in Section 3. We compare the performance of acoustic features derived from these models against two baselines: the MFCC parameters, described above, and the mel-frequency filterbank features, denoted MEL-FBANK, that are often preferred to MFCCs, which are strongly decorrelated because their computation includes an additional discrete cosine transform (DCT) (Ahmed et al., 1974). The dimension of MEL-FBANKs is 120 (40 static + Δ + $\Delta\Delta$ coefficients).

The four auditory representations provide frequency-selective features at the full sampling rate of the stimulus, which is computationally expensive. Similar to the approach taken by Hemmert et al. (2004), we temporally integrated the root-mean-square energy of each channel using using a Hann window (25 ms width) advanced in 10 ms steps in order to obtain the same number of frames as for the baselines. For CARFAC and GAMMATONE-SLANEY models, the analysis is performed at 16 kHz. For BAUMGARTE and ZILANY2014, the analysis was performed at 112 kHz, hence downsampling to 16 kHz was required prior to temporal integration. No spectral integration across channels was performed, instead a simple decimation was applied to reduce the frequency resolution, when required.

Experiment Setup A single training example consists of 40 frames. It is constructed by stacking the frames corresponding to the particular phoneme plus its right and left context frames, possibly padding with zeros if the context is too short. Phonemes longer than 40 frames are ignored.

The training and evaluation sets in our experiments always consist of disjoint sets of languages and speakers. For each dataset we limit the number of training examples to 50,000 and evaluation examples to 10,000. In order to keep the overall set of training labels balanced, with equal number of positive and negative examples, we employ a simple under-sampling approach (Japkowicz and Stephen, 2002; Krawczyk, 2016). If enough examples are available, we sample equal number of them from every language in the training set. Conversely, an imbalance in a language is preferred over the lack of training examples. It is important to note that we do not guarantee that the number of training examples is the same across speakers of a language. We use mean and standard deviation computed over the training set input features to scale the training as well as evaluation sets.

Model Architectures We employ vanilla feed-forward Deep Neural Network (DNN) binary classifier from TensorFlow (Abadi et al., 2016), further tuning the model hyperparameters for maximizing the AUC. A simple two-layer architecture with 200 Softplus (Zheng et al., 2015) units in each layer, dropout probability of 0.2 (Srivastava et al., 2014), Adadelta optimizer (Zeiler, 2012) and the decaying learning rate of 0.6 with a large batch size of 6000 (Smith et al., 2017) were found to perform well across our experiments.

We also used a Convolutional Neural Network (CNN) (Abdel-Hamid et al., 2014) architecture. The network has two CNN layers, where each layer consists of

Features	Channels	Model	P - F					VOICED				
			bn–bn	bn–es	es–bn	es–es	avg	bn–bn	bn–es	es–bn	es–es	avg
MFCC	39	CNN	1.17	0.16	6.02	0.00	1.84	0.85	1.31	2.38	0.11	1.16
		DNN	2.11	**0.07**	**3.70**	0.00	**1.47**	0.71	1.90	2.55	0.04	1.30
MEL-FBANK	120	CNN	**0.05**	0.52	7.70	0.00	2.07	0.48	1.11	2.92	0.03	1.14
		DNN	1.82	0.04	7.08	0.00	2.24	0.63	1.82	3.02	0.06	1.38
GAMMATONE-SLANEY	64	CNN	1.14	0.40	9.04	0.06	2.66	0.88	0.85	2.89	0.09	1.18
		DNN	3.22	1.71	12.64	0.11	4.42	0.54	3.58	4.85	0.05	2.26
	32	CNN	2.34	0.31	9.65	0.00	3.08	0.44	**0.51**	2.23	0.11	0.82
		DNN	2.87	0.47	10.05	0.06	3.36	0.68	1.67	1.68	0.27	1.08
BAUMGARTE	83	CNN	0.81	0.19	8.11	0.00	2.28	**0.37**	0.67	2.70	0.09	0.96
		DNN	0.59	0.26	8.41	0.00	2.32	0.52	2.80	1.80	0.02	1.29
	50	CNN	1.98	0.40	7.98	0.00	2.59	0.59	0.74	1.33	0.06	**0.68**
		DNN	2.30	0.22	8.21	0.00	2.68	0.70	2.47	**1.32**	0.03	1.13
CARFAC	65	CNN	2.54	0.51	7.83	0.00	2.72	0.69	0.99	2.64	**0.02**	1.09
		DNN	4.39	0.80	9.11	0.00	3.57	0.57	1.96	2.14	0.14	1.20
ZILANY2014	64	CNN	2.44	0.15	13.29	0.00	3.97	0.50	0.77	3.88	0.08	1.31
		DNN	7.72	0.75	16.63	0.06	6.29	0.44	2.10	4.03	0.04	1.65
	32	CNN	1.11	0.17	13.81	0.00	3.77	0.65	1.15	2.56	0.13	1.12
		DNN	6.69	0.77	14.73	0.01	5.55	0.57	2.42	1.86	0.02	1.22

Table 5: Bengali–Spanish phoneme asymmetry experiments evaluated using AOC metric.

two-dimensional convolution layer (Abdel-Hamid et al., 2013) with 32 filters with receptive field of 3×3, followed by a max-pooling layer with a pooling region of 2×2 and a stride of 2. The CNN layers are followed by a dense layer with 200 ReLU (Zeiler et al., 2013) units. Batch normalization was applied after each layer in the network (Ioffe and Szegedy, 2015). Similar hyperparameters to DNN were used, with a smaller batch size of 400 and a decaying learning rate of 0.4.

Evaluation Results and Discussion Each classification experiment is repeated three times and the results are averaged. For each classification, we measure the area under the ROC curve (AUC) numbers for every pair of training and evaluation languages, including a language against itself. Since AUC values are generally high, we instead report Area Over the Curve (AOC) values for better readability. Classification results for cross-linguistic consistency of the two contrasts P-F and VOICED are shown in Table 5 for each of the six acoustic feature types. The averages over all four language combinations for each contrast are shown in avg columns. For some of the acoustic representations we produced the acoustic features at two frequency resolutions (shown as the number of channels in the second column). The third column shows the type of the binary classifier that we trained. Best classification results are shown in bold.

The P-F contrast only distinguishes between the phonemic labels /p/ and /f/. Both languages have phonemes that are labeled /f/ and /p/, but as discussed earlier [p^h] is an allophone of /f/ in Bengali and an allophone of /p/ in Spanish. As can be seen from Table 5, this contrast is only truly robust between Bengali and Spanish (despite the conflicting status of the allophone [p^h]) with the DNN model trained on MFCC acoustic features. This confirms the previous findings by Johny et al. (2019), who only used this type of features in their experiments. For all other acoustic configurations, the AOC values are relatively too high when predicting Bengali from the Spanish data (es–bn). It is worth noting that for this experiment, none of the sophisticated auditory configurations outperform the baseline features, although the CNN models trained on the 50-channel BAUMGARTE features and the 65-channel CARFAC features perform slightly worse than the MEL-FBANK baseline.

The VOICED contrast distinguishes between voiced and unvoiced labial sets ({ /b/, /b^h/ } and { /p/, /f/ }). As can be seen from Table 5, this contrast is generally robust and is predicted consistently by all the configurations. Furthermore, in this experiment there is at least one configuration corresponding to each of the four auditory representations that outperforms the MFCC and MEL-FBANK baselines, although not by a big margin. It is interesting to note that there is no clear "winning" representation, although the CNN architecture trained on the 50-channel BAUMGARTE features performs the best according to the average of the four corresponding AOC metrics. Moreover, the four ZILANY2014 configurations, which have the worst performance in resolving the P-F contrast, can detect the VOICED contrast reliably.

6. Conclusion and Future Work

We presented an auditory modeling toolkit designed for easy combination of various models of human auditory periphery in a flexible processing pipeline. Ten models of auditory periphery are currently supported. These range from the popular GAMMATONE filterbanks, also provided by software similar to ours, to the less frequently used peripheral BAUMGARTE model. Some models are highly specialized to model one particular biological mechanism, such as SUMNER2002, while others, such as CARFAC provide simulations for most of the critical mechanisms active in the auditory periphery. The toolkit supports some interesting, and to the best of our knowledge not explored in the literature, combinations of models in a single pipeline, such as combining the BAUMGARTE estimates of IHC transmembrane potentials with BRUCE2018 synaptic model. We demonstrated the effectiveness of the resulting auditory representations on a simple phonemic contrast detection task, where they often outperform the baselines.

Future work will focus on supporting more auditory models. In addition, no special effort was undertaken to fine-tune various model combinations, which can be problematic because different models sometimes require their inputs to be scaled appropriately. Finally, we plan to broaden the scope of experiments to evaluate more phonemic contrasts on languages less-resourced than Bengali.

7. Bibliographical References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, pages 265–283.

Abdel-Hamid, O., Deng, L., and Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Proc. of Interspeech*, volume 2013, pages 1173–1175, Lyon, France. ISCA.

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., and Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545.

Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete Cosine Transform. *IEEE transactions on Computers*, 100(1):90–93.

Arle, J. and Kim, D. O. (1991). Neural modeling of intrinsic and spike-discharge properties of cochlear nucleus neurons. *Biological Cybernetics*, 64(4):273–283.

Baby, D. and Verhulst, S. (2018). Biophysically-inspired features improve the generalizability of neural network-based speech enhancement systems. In *Proc. of Interspeech*, pages 3264–3268, Hyderabad, India. ISCA.

Baumgarte, F. (1997). A physiological ear model for specific loudness and masking. In *Proc. Workshop on Applications of Signal Processing to Audio and Acoustics*, page 4, New York. IEEE.

Baumgarte, F. (2000). *Ein psychophysiologisches Gehörmodell zur Nachbildung von Wahrnehmungsschwellen für die Audiocodierung*. Ph.D. thesis, Universität Hannover, Germany, October. In German.

Bell, A. (2012). A Resonance Approach to Cochlear Mechanics. *PloS One*, 7(11):e47918.

Blischak, J. D., Davenport, E. R., and Wilson, G. (2016). A quick introduction to version control with Git and GitHub. *PLoS Computational Biology*, 12(1):e1004668.

Bruce, I. C., Erfani, Y., and Zilany, M. S. A. (2018). A phenomenological model of the synapse between the inner hair cell and auditory nerve: Implications of limited neurotransmitter release sites. *Hearing Research*, 360:40–54.

Carney, L. H. (1993). A model for the responses of low-frequency auditory-nerve fibers in cat. *The Journal of the Acoustical Society of America*, 93(1):401–417.

Clements, G. N. (2009). *Contemporary Views on Architecture and Representations in Phonology*, volume 48 of *Current Studies in Linguistics*. MIT Press.

Cooke, M. (1993). *Modelling Auditory Processing and Organisation*. Distinguished Dissertations in Computer Science. Cambridge University Press.

Corey, D. P., Maoiléidigh, D. Ó., and Ashmore, J. F. (2017). Mechanical Transduction Processes in the Hair Cell. In Geoffrey A Manley, et al., editors, *Understanding the Cochlea*, pages 75–111. Springer.

Dau, T., Püschel, D., and Kohlrausch, A. (1996). A quantitative model of the "effective" signal processing in the auditory system. I. Model structure. *The Journal of the Acoustical Society of America*, 99(6):3615–3622.

Dietz, M., Lestang, J.-H., Majdak, P., Stern, R. M., Marquardt, T., Ewert, S. D., Hartmann, W. M., and Goodman, D. F. M. (2018). A framework for testing and comparing binaural models. *Hearing Research*, 360:92–106.

Drew, P. J. and Abbott, L. F. (2006). Models and Properties of Power-Law Adaptation in Neural Systems. *Journal of Neurophysiology*, 96(2):826–833.

Dupoux, E. (2018). Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

Farhadi, A. and Carney, L. H. (2019). UR EAR – University of Rochester Envisioning Auditory Responses (Version 2.1). `https://www.urmc.rochester.edu/MediaLibraries/URMCMedia/labs/carney-lab/codes/UR_EAR_v2_1.zip`.

Fettweis, A. (1986). Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74(2):270–327.

Fontaine, B., Goodman, D. F. M., Benichoux, V., and Brette, R. (2011). Brian hears: online auditory processing using vectorization over channels. *Frontiers in Neuroinformatics*, 5:9, July. Available from: `https://github.com/brian-team/brian2hears`.

Freedman, D. S., Cohen, H. I., Deligeorges, S., Karl, C., and Hubbard, A. E. (2013). An analog VLSI implementation of the inner hair cell and auditory nerve using a dual AGC model. *IEEE Transactions on Biomedical Circuits and Systems*, 8(2):240–256.

Ganchev, T., Fakotakis, N., and Kokkinakis, G. (2005). Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task. In *Proc. of SPECOM*, volume 1, pages 191–194, Patras, Greece.

Ghahremani, P., Manohar, V., Povey, D., and Khudanpur, S. (2016). Acoustic Modelling from the Signal Domain Using CNNs. In *Proc. of Interspeech*, pages 3434–3438, San Francisco. ISCA.

Glasberg, B. R. and Moore, B. C. J. (1990). Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47(1-2):103–138.

Goodman, D. F. M. and Brette, R. (2009). The Brian simulator. *Frontiers in Neuroscience*, 3:26. Available from: `http://briansimulator.org/`.

Google. (2008). Protocol Buffers. Google's Data Interchange Format. `https://developers.google.com/protocol-buffers/`. [Online], Accessed: 2019-12-2.

Google. (2010). Google Test – Google Testing and Mocking Framework. `https://github.com/google/googletest`. [Online], Accessed: 2019-12-2.

Google. (2019). Bazel. `http://bazel.build`. [Online], Accessed: 2019-12-2.

Greenwood, D. D. (1990). A cochlear frequency-position function for several species – 29 years later. *The Journal of the Acoustical Society of America*, 87(6):2592–2605.

Harczos, T., Szepannek, G., and Klefenz, F. (2007). Towards automatic speech recognition based on cochlear traveling wave delay trajectories. In *Proc. Interna-*

tional Symposium on Auditory and Audiological Research (ISAAR), volume 1, pages 83–92.

Härmä, A. and Palomäki, K. (2000). HUTear – a free Matlab toolbox for modeling of human auditory system. In *Proc. Matlab Digital Signal Processing (DSP) Conference*, pages 96–99, Espoo, Finland, November. Available from: `http://legacy.spa.aalto.fi/software/HUTear/`.

Hemmert, W., Holmberg, M., and Gelbart, D. (2004). Auditory-based Automatic Speech Recognition. In *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*, Jeju, Korea.

Holmberg, M., Gelbart, D., and Hemmert, W. (2007). Speech encoding in a model of peripheral auditory processing: Quantitative assessment by means of automatic speech recognition. *Speech Communication*, 49(12):917–932.

Ibrahim, R. A. and Bruce, I. C. (2010). Effects of Peripheral Tuning on the Auditory Nerve's Representation of Speech Envelope and Temporal Fine Structure Cues. In Enrique Lopez-Poveda, et al., editors, *The neurophysiological bases of auditory perception*, pages 429–438. Springer.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Irino, T. and Patterson, R. D. (1997). A time-domain, level-dependent auditory filter: The gammachirp. *The Journal of the Acoustical Society of America*, 101(1):412–419.

Irino, T., Patterson, R. D., and Kawahara, H. (2006). Speech segregation using an auditory vocoder with event-synchronous enhancements. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2212–2221.

Jackson, B. S. and Carney, L. H. (2005). The spontaneous-rate histogram of the auditory nerve can be explained by only two or three spontaneous rates and long-range dependence. *Journal of the Association for Research in Otolaryngology*, 6(2):148–159.

Jackson, B. S. (2007). The SGfast Mex Function. `https://www.urmc.rochester.edu/MediaLibraries/URMCMedia/labs/carney-lab/documents/articles/Jackson-SGfast-2003.pdf`. Department of Neurobiology and Behavior, Cornell University.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

Johannesma, P. I. M. (1972). The Pre-response Stimulus Ensemble of Neurons in the Cochlear Nucleus. In *Proc. IPO Symposium on Hearing Theory*, pages 58–69, Eindhoven, Netherlands.

Johny, C., Gutkin, A., and Jansche, M. (2019). Cross-Lingual Consistency of Phonological Features: An Empirical Study. In *Proc. of Interspeech*, pages 1741–1745, Graz, Austria. ISCA.

Karajalainen, M. (1996). A binaural auditory model for sound quality measurements and spatial hearing studies. In *Proc. International Conference on Acoustics, Speech, and Signal Processing Conference (ICASSP)*, volume 2, pages 985–988, Atlanta, USA. IEEE.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

LeMasurier, M. and Gillespie, P. G. (2005). Hair-Cell Mechanotransduction and Cochlear Amplification. *Neuron*, 48(3):403–415.

Li, K. and Príncipe, J. C. (2018). Biologically-Inspired Spike-Based Automatic Speech Recognition of Isolated Digits Over a Reproducing Kernel Hilbert Space. *Frontiers in Neuroscience*, 12:194.

Lopez-Poveda, E. A. and Meddis, R. (2001). A human nonlinear cochlear filterbank. *The Journal of the Acoustical Society of America*, 110(6):3107–3118.

Lopez-Poveda, E. A. (2005). Spectral processing by the peripheral auditory system: facts and models. *International Review of Neurobiology*, 70:7–48.

Lyon, R. (1982). A computational model of filtering, detection, and compression in the cochlea. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 7, pages 1282–1285, Paris, France. IEEE.

Lyon, R. F. (2011). Using a Cascade of Asymmetric Resonators with Fast-Acting Compression as a Cochlear Model for Machine-Hearing Applications. In *Proc. Autumn Meeting of the Acoustical Society of Japan*, pages 509–512. Available from: `https://github.com/google/carfac`.

Lyon, R. F. (2017). *Human and Machine Hearing: Extracting Meaning from Sound*. Cambridge University Press, United Kingdom.

Ma, N., Green, P., Barker, J., and Coy, A. (2007). Exploiting correlogram structure for robust speech recognition with multiple speech sources. *Speech Communication*, 49(12):874–891. Available from: `https://staffwww.dcs.shef.ac.uk/people/N.Ma/resources/gammatone/#Pat1986`.

Manley, G. A., Gummer, A. W., Popper, A. N., and Fay, R. R. (2017). *Understanding the Cochlea*, volume 62 of *Springer Handbook of Auditory Research*. Springer.

Meddis, R., Hewitt, M. J., and Shackleton, T. M. (1990). Implementation details of a computation model of the inner hair-cell auditory-nerve synapse. *The Journal of the Acoustical Society of America*, 87(4):1813–1816.

Meddis, R., Lopez-Poveda, E. A., Fay, R. R., and Popper, A. N. (2010). *Computational Models of the Auditory System*, volume 35 of *Springer Handbook of Auditory Research*. Springer.

Meddis, R. (1986). Simulation of mechanical to neural transduction in the auditory receptor. *The Journal of the Acoustical Society of America*, 79(3):702–711.

Meddis, R. (1988). Simulation of auditory–neural transduction: Further studies. *The Journal of the Acoustical Society of America*, 83(3):1056–1063.

Moore, B. C. J. and Glasberg, B. R. (1983). Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The Journal of the Acoustical Society of America*, 74(3):750–753.

Moore, B. C. J. (2007). Basic auditory processes involved in the analysis of speech sounds. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):947–963.

Moore, H. (2017). *MATLAB for Engineers*. Pearson.

Moran, S., McCloy, D., and Wright, R. (2014). *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig. Available from: `http://phoible.org/`.

O'Mard, L. P. (2010). Development System for Auditory Modelling (DSAM). `http://dsam.org.uk`. Centre for the Neural Basis of Hearing (CNBH), Version 2.8.44.

Ondel, L., Li, R., Sell, G., and Hermansky, H. (2019). Deriving Spectro-Temporal Properties of Hearing from Speech Data. In *Proc. IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 411–415, Brighton, UK. IEEE.

Pan, Z., Li, H., Wu, J., and Chua, Y. (2018). An Event-Based Cochlear Filter Temporal Encoding Scheme for Speech Signals. In *Proc. Int. Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Rio de Janeiro, Brazil. IEEE.

Pärt-Enander, E., Sjöberg, A., Melin, B., and Isaksson, P. (1996). *The MATLAB Handbook*. Addison-Wesley Harlow.

Pascal, J., Bourgeade, A., Lagier, M., and Legros, C. (1998). Linear and nonlinear model of the human middle ear. *The Journal of the Acoustical Society of America*, 104(3):1509–1516.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Patterson, R. D., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., and Allerhand, M. (1992). Complex Sounds and Auditory Images. In Y Cazals, et al., editors, *Auditory Physiology and Perception: Proc. 9th International Symposium on Hearing*, pages 429–446. Elsevier.

Patterson, R. D. (1986). Auditory filters and excitation patterns as representations of frequency resolution. *Frequency Selectivity in Hearing*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Peisl, W. (1990). *Beschreibung aktiver nichtlinearer Effekte der peripheren Schallverarbeitung des Gehörs durch ein Rechnermodell*. Ph.D. thesis, Technische Universität München, Germany. In German.

Rudnicki, M. and Hemmert, W. (2014). Cochlea: inner ear models in Python. `https://github.com/mrkrd/cochlea`.

Rudnicki, M., Schoppe, O., Isik, M., Völk, F., and Hemmert, W. (2015). Modeling auditory coding: from sound to spikes. *Cell and Tissue Research*, 361(1):159–175.

Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., and Vinyals, O. (2015). Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, pages 1–5, Dresden, Germany. ISCA.

Saremi, A., Beutelmann, R., Dietz, M., Ashida, G., Kretzberg, J., and Verhulst, S. (2016). A comparative study of seven human cochlear filter models. *The Journal of the Acoustical Society of America*, 140(3):1618–1634.

Schnupp, J., Nelken, I., and King, A. (2011). *Auditory Neuroscience: Making Sense of Sound*. MIT Press.

Seneff, S. (1988). A joint synchrony/mean-rate model of auditory speech processing. *Journal of Phonetics*, 16(1):55–76.

Shamma, S. A., Chadwick, R. S., Wilbur, W. J., Morrish, K. A., and Rinzel, J. (1986). A biophysical model of cochlear processing: Intensity dependence of pure tone responses. *The Journal of the Acoustical Society of America*, 80(1):133–145.

Shera, C. A., Guinan, J. J., and Oxenham, A. J. (2002). Revised estimates of human cochlear tuning from otoacoustic and behavioral measurements. *Proc. National Academy of Sciences*, 99(5):3318–3323.

Singh, R. K., Xu, Y., Wang, R., Hamilton, T. J., van Schaik, A., and Denham, S. L. (2018). CAR-lite: A multi-rate cochlea model on FPGA. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, Florence, Italy. IEEE.

Slaney, M. (1988). Lyon's Cochlear Model. Technical Report 13, Apple Computer, Advanced Technology Group.

Slaney, M. (1993). An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank. Tech. Rep. 35, Apple Computer, Inc. Perception Group.

Slaney, M. (1998). Auditory Toolbox: Version 2. Technical Report #1998-010, Interval Research Corporation. Available from: `https://engineering.purdue.edu/~malcolm/interval/1998-010`.

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*.

Søndergaard, P. L. and Majdak, P. (2013). The Auditory Modeling Toolbox. In Jens Blauert, editor, *The Technology of Binaural Listening*, Modern Acoustics and Signal Processing, pages 33–56. Springer. Available from: `http://amtoolbox.sourceforge.net/`.

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

Sumner, C. J., Lopez-Poveda, E. A., O'Mard, L. P., and Meddis, R. (2002). A revised model of the inner-hair cell and auditory-nerve complex. *The Journal of the Acoustical Society of America*, 111(5):2178–2188.

Sumner, C. J., Lopez-Poveda, E. A., O'Mard, L. P., and Meddis, R. (2003). Adaptation in a revised inner-hair cell model. *The Journal of the Acoustical Society of America*, 113(2):893–901.

Tabibi, S., Kegel, A., Lai, W. K., and Dillier, N. (2017). Investigating the use of a Gammatone filterbank for a cochlear implant coding strategy. *Journal of Neuroscience Methods*, 277:63–74.

Tan, Q. and Carney, L. H. (2003). A phenomenological

model for the responses of auditory-nerve fibers. II. Nonlinear tuning with a frequency glide. *The Journal of the Acoustical Society of America*, 114(4):2007–2020.

Thakur, C. S., Hamilton, T. J., Tapson, J., van Schaik, A., and Lyon, R. F. (2014). FPGA Implementation of the CAR Model of the Cochlea. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1853–1856, Melbourne, Australia. IEEE.

Tjandra, A., Sakti, S., Neubig, G., Toda, T., Adriani, M., and Nakamura, S. (2015). Combination of two-dimensional cochleogram and spectrogram features for deep learning-based ASR. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4525–4529, South Brisbane, Australia. IEEE.

Tjandra, A., Sakti, S., and Nakamura, S. (2017). Attention-based wav2text with feature transfer learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 309–315. IEEE.

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., and Zafeiriou, S. (2016). Adieu features? End-to-End Speech Emotion Recognition Using a Deep Convolutional Recurrent Network. In *Proc. IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5200–5204, Shanghai, China. IEEE.

Tüske, Z., Golik, P., Schlüter, R., and Ney, H. (2014). Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *Proc. of Interspeech*, pages 1420–1424, Singapore. ISCA.

Van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22. Online: `https://numpy.org/`.

Verhulst, S., Dau, T., and Shera, C. A. (2012). Nonlinear time-domain cochlear model for transient stimulation and human otoacoustic emission. *The Journal of the Acoustical Society of America*, 132(6):3842–3848.

Verhulst, S., Altoe, A., and Vasilkov, V. (2018). Computational modeling of the human auditory periphery: Auditory-nerve responses, evoked potentials and hearing loss. *Hearing Research*, 360:55–75.

Westerman, L. A. and Smith, R. L. (1988). A diffusion model of the transient response of the cochlear inner hair cell synapse. *The Journal of the Acoustical Society of America*, 83(6):2266–2276.

Wolfram, S. (1999). The mathematica book. *Assembly Automation*.

Xu, Y., Thakur, C. S., Singh, R. K., Hamilton, T. J., Wang, R. M., and van Schaik, A. (2018). A FPGA implementation of the CAR-FAC cochlear model. *Frontiers in Neuroscience*, 12:198.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book*. Cambridge University Engineering Department.

Young, E. D. (2007). Neural representation of spectral and temporal information in speech. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):923–945.

Zeghidour, N., Usunier, N., Synnaeve, G., Collobert, R., and Dupoux, E. (2018). End-to-End Speech Recognition From the Raw Waveform. *arXiv preprint arXiv:1806.07098*.

Zeghidour, N. (2019). *Learning Representations of Speech from the Raw Waveform*. Ph.D. thesis, Paris Sciences et Lettres (PSL), Paris, March.

Zeiler, M. D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q. V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., and Hinton, G. (2013). On rectified linear units for speech processing. In *Proc. ICASSP 2013*, pages 3517–3521, Vancouver, Canada, May. IEEE.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701.

Zeyer, A., Irie, K., Schlüter, R., and Ney, H. (2018). Improved training of end-to-end attention models for speech recognition. *arXiv preprint arXiv:1805.03294*.

Zhang, X., Heinz, M. G., Bruce, I. C., and Carney, L. H. (2001). A phenomenological model for the responses of auditory-nerve fibers: I. Nonlinear tuning with compression and suppression. *The Journal of the Acoustical Society of America*, 109(2):648–670.

Zheng, H., Yang, Z., Liu, W., Liang, J., and Li, Y. (2015). Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4, Budapest, Hungary. IEEE.

Zilany, M. S. A., Bruce, I. C., Nelson, P. C., and Carney, L. H. (2009). A phenomenological model of the synapse between the inner hair cell and auditory nerve: long-term adaptation with power-law dynamics. *The Journal of the Acoustical Society of America*, 126(5):2390–2412.

Zilany, M. S. A., Bruce, I. C., Ibrahim, R. A., and Carney, L. H. (2013). Improved parameters and expanded simulation options for a model of the auditory periphery. In *Proc. Association for Research in Otolaryngology (ARO) Midwinter Research Meeting*, pages 440–441, Baltimore, MD.

Zilany, M. S. A., Bruce, I. C., and Carney, L. H. (2014). Updated parameters and expanded simulation options for a model of the auditory periphery. *The Journal of the Acoustical Society of America*, 135(1):283–286.

Zwicker, E. and Peisl, W. (1990). Cochlear preprocessing in analog models, in digital models and in human inner ear. *Hearing Research*, 44(2-3):209–216.

Zwicker, E. (1961). Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248.

Zwicker, E. (1986). A hardware cochlear nonlinear preprocessing model with active feedback. *The Journal of the Acoustical Society of America*, 80(1):146–153.