Monaural Speech Separation with Deep Neural Networks

Diplomarbeit
von

Maximilian Warsewa

eingereicht am

Interactive Systems Lab (ISL)
Institut für Anthropomatik und Robotik
Fakultät für Informatik
Karlsruher Institut für Technologie

Betreuer: Prof. Dr. Alex Waibel

Zusammenfassung

Die vorliegende Arbeit beschreibt die Implementierung und Evaluation eines Ende-zu-Ende Systems zur Trennung von Sprache von Hintergrundgeräuschen in monauralen Sprachaufnahmen.

Die Trennung wird durch die Maskierung einer Zeit-Frequenz-Darstellung des Eingabesignals und durch einen nachgeschalteten Pfad zur Rekonstruktion des Zeitsignals realisiert.

Die Maske wird aus dem verrauschten Signal von einem tiefen künstlichen neuronalen Netzwerk (DNN) geschätzt. Dieses Netzwerk wurde auf synthetischen Daten trainiert, welche durch additives Mischen von sauberen Sprachaufnahmen und Hintergrundgeräuschen in einem fest definierten Signal-Rausch-Abstand (SNR) generiert wurden. Die Qualität des rekonstruierten Signals wird mithilfe der automatisierten Metrik zur Sprachverständlichkeit STOI bewertet.

Aus der Untersuchung geht hervor, dass die Wahl geeigneter Merkmalsvektoren essentiell ist um eine gute Generalisierbarkeit auf Signal-Rausch-Abstände zu erreichen, die während des Trainings nicht vorgekommen sind. Ein mehrschichtiges voll-vernetztes Netzwerk erreicht mithilfe der psychoakustisch motivierten Cochleagram-Merkmalsvektoren deutlich bessere Ergebnisse als beim Einsatz von Merkmalen, die auf einem gewichteten Spektrogramm basieren. Dieses Ergebnis wird noch übertroffen, wenn die Impulsantworten der verwendeten Bandpassfilter direkt optimiert werden. Hierzu wird die Merkmalsextraktion in eine Schicht des neuronalen Netzes verlagert und das Netz direkt auf der Wellenform des Signals trainiert.

Abstract

This thesis presents an implementation and performance evaluation of an end-to-end system for speech separation from background noise in monaural voice recordings.

Separation is achieved by applying a mask to a time-frequency representation of the input signal and through a subsequent reconstruction path for the clean speech signal.

The mask is estimated from the noisy input data using deep neural networks which are trained on a synthetic dataset obtained by additive mixing of separate recordings of clean speech and background noises at a tightly-controlled signal-to-noise ratio. The expected intelligibility of the reconstructed audio is compared using the automated intelligibility metric STOI.

We discover that a fully-connected feed-forward network achieves superior performance at adapting to signal-to-noise ratios not encountered during training when the psychoacoustically motivated smoothed cochleagram features instead of features obtained by weighting a spectrogram in proportion to the magnitude gain of gammatone filters are employed. This result is further improved by having the neural network adapt the finite impulse response of the bandpass filters during training when using the raw waveform as model input.

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbst und ohne fremde Hilfe implementiert und verfasst habe. Sämtliche verwendeten Quellen und Hilfsmittel sind im Literaturverzeichnis angegeben. Textstellen, die fremden Werken wörtlich oder sinngemäß entnommen wurden, sind entsprechend kenntlich gemacht.

Karlsruhe, den 25. September 2017

Contents

Acknowledgments

This thesis would not have been possible without the help and support of many people.

First of all, I want to thank Prof. Alex Waibel for taking me on the ride of my life. Bringing me to Facebook's Menlo Park campus gave me the incomparable opportunity to work alongside the world's most brilliant engineers on problems of mind-boggling scale for three years while technically still being a student. He also deserves credit for encouraging me to finish my diploma on what in hindsight looks like an almost impossible schedule and for suggesting this fascinating topic.

I also want to thank Silke Dannenmaier for the administrative support and making sure I didn't miss any important deadlines.

Raimund Warsewa provided a productive environment in his office where I could write most of this thesis without distraction.

Kay Rottmann answered countless machine-learning related questions and allowed me to take a multi-month break from our joint start-up to focus on this thesis instead.

My former colleagues Thilo Köhler and Evgeniy Shin showed me that speech processing is a fun and rewarding topic beyond building just user interfaces and service infrastructure for speech recognition.

Michael Foitzik, Felix Lübbe, Anja Sommer, Alexander Sonndag and David Weiß made my time in Karlsruhe and on campus unforgettable and remain good friends years after their graduation.

I also want to thank Isabelle Glaubitt for her friendship and for showing up with th bag of coffee which has provided me with much needed fuel during a crucial part of this thesis.

Finally, and most importantly, I would like to thank my wife Rasmie who showed me endless support during this whole ordeal. Words can't begin to express my gratitude for all the encouragement, love and understanding I have received from her.

Chapter 1

Introduction

For a healthy human it is an easy feat to tune into a single conversation in a crowded room while being completely oblivious to background noise and other conversations that happen simultaneously. This ability of the human auditory system to solve the cocktail party problem is incredible. An estimated 5% of the world's population however suffers from disabling hearing loss [20]. For the 360 million affected people the ability to perceive speech—especially in the presence of background noise—is significantly impaired. Since the perception of speech is crucial for being able to communicate verbally, this impairment has profound social and emotional consequences for the sufferers [20]. Current hearing aids offer only limited relief as they cannot distinguish between speech and background noise and will simply opt to amplify both signals which does not greatly help intelligibility.

An "intelligent" digital filter that can distinguish between the conversation you're currently having and disruptive background noise would provide an improvement to hearing aids that can hardly be overstated. It is not unconceivable that such as system would also be used by people whose hearing is fine as listening to someone in a loud environment is still exhausting.

Such a filter would not have to be perfect to be useful. Even small gains in signal-to-noise ratio (SNR) can have a profound impact on speech intelligibility. A 1 dB SNR improvement near the speech reception threshold leads to 5-10% improvement in intelligibility [28, pp. 6–7].

An estimated half of all cases of hearing loss in adults are caused by exposure to excessively loud noise [19]. There are many loud work environments where verbal communication is nonetheless necessary. Workers might therefore opt to not wear hearing protection because it hampers their ability to communicate with their coworkers. Having effective means to filter out harmful levels of background noise while still making spoken communication possible might encourage people to wear hearing protection more frequently. A filter that can remove loud background noise but allow for unhindered conversation might therefore help prevent these cases of

hearing loss.

This thesis contributes to these goals by implementing a supervised machine learning system that can learn to separate speech from background noise in monaural (single microphone) recordings given examples of how clean speech is degraded by additive noise.

Chapter 2 describes the foundational knowledge needed to understand the current state of the art that masks time frequency representations of the noisy signal. One such representation, the cochleagram, is based on psychoacoustic models of the nerve responses of the hair cells in the inner ear and explained in detail. This is followed by chapter 3 which implements a baseline system for speech separation based on the masking approach employed by Chen et. al [2]. That model is then modified to use different feature representations. It is examined if similar results can be achieved using simpler feature extraction that needs less processing power. This is especially worthwhile if one wants to implement these algorithms in battery-powered hearing aids. Following this, the system is then extended to move the feature extraction step into the deep neural network itself with the aim of better masking performance. The impulse responses of the filters themselves are optimized as part of the network training. The evaluation chapter (4) discusses the performance and peculiarities of the implemented systems.

1.1 Related work

Spectral subtraction [1] makes the assumption that the clean speech signal has been corrupted by statistically independent additive noise. The power spectrum of the noise signal can be estimated for stationary noise by taking the average of multiple frames. This estimated power spectrum is then subtracted from the mixed signal spectrum, keeping the phase information intact. The resulting signal has less noise but as a result of subtracting parts of the spectrum, it exhibits an acoustic phenomenen called *musical noise* [11, pp. 516–519]. This is caused by tones at varying frequencies that appear and disappear rapidly in the reconstructed signal and are quite disturbing to the perceptual quality of the filtered signal.

The idea to use a deep neural network (DNN) in order to filter out background noise from speech recordings goes back three decades. In 1988 Tamura and Waibel [27] showed that a four-layer feed-forward NN could successfully be used to directly separate the waveforms of Japanese speech recordings from computer lab background noise. This network is small by the standards of today, but took weeks of training on a supercomputer of its time. The authors note that the perceived quality of the filtered audio was higher, but that intelligiblity did not improve.

Artificially adding noise to existing audio recordings is routinely used in super-

vised training of automatic speech recognition systems in order to improve their robustness towards environmental noise [7].

Chen, Jitong and Wang [2] follow an approach rooted in the computational auditory scene analysis (CASA, see chapter 2). The idea is to predict an "Ideal Ratio Mask" (IRM) that describes the proportion of speech to noise energy in a frequency band at a given time. When the IRM is known, the clean speech signal can be reconstructed (separated) from the noisy signal by weighting each time frequency unit of the noisy recording according to its speech content. The IRM is computed not on a spectrogram, but on the outputs of a filterbank that is motivated by psychoacoustic research (cochleagram, see chapter 2.2). They use a 5 layer feed-forward DNN to predict 5 frames of the IRM from windows 23 frames of the noisy cochleagram. This is the architecture that serves as the baseline model of this thesis. They show that a system trained at -2 dB SNR generalizes to conditions of -5 dB, 0 dB, and +5 dB that were not seen in training. The system can also adapt to novel noises not seen in training. They evaluate their system on both hearing impaired and normal hearing subjects and demonstrate that their approach increases intelligibility on the test sentences.

Huang, Kim, Hasegawa-Johnson and Smargdis further improve this approach by presenting a framework for separating arbitrarily many sources using recurrent neural networks using a discriminative training criterion. They also move the time-frequency masking operation directly into a layer of the neural network in order to jointly optimize the network with the masking function [10].

1.2 Assumptions and limitations

There are several assumptions made in this thesis which limit the direct application of the results in a real hearing aid. Most of the limitations can however be overcome by acquiring more training data.

This thesis considers additive noise on clean speech only. While the recordings of noise naturally do exhibit complex interactions of the sound sources with the environment, the speech examples were recorded in an acoustically "dry" room. Therefore effects of room acoustics such as reverberated speech are not modeled. This however could be amended with a set of room impulse responses and an enhanced process for creating training data in section 3.3.

Another interesting real-world effect that is not taken into consideration in the implementation is the fact that humans alter their speaking style in order to allow for efficient verbal communication in noisy environments. This is called the *Lombard effect* and can best be described by the radio analogy of moving a transmission to a free band in the spectrum where there is less interference. The main acoustic differences from normal speech is an increase in the fundamental

frequency (F_0), shifting energy from lower to higher frequencies, increased vowel duration and—perhaps unsurprisingly—overall higher volume [13]. Hannun et. al had speakers wear headphones that simulated loud environments in order to capture the Lombard effect for training data in automatic speech recognition [7], however such a corpus is not known to be freely available to the author. Therefore, there is likely to be a mismatch between the data the predictor in this thesis is trained on and real-world data.

Lastly, the models detailed in section 3.4 operate on windows of ~ 300 ms for acoustical context and predict an estimate of the noise on a short segment (60 ms) centered within the analysis window. The total delay of the system (including time required for pre- and post-processing) will likely make a real-time application such as a face-to-face conversation feel like a long-distance phone call and greatly decrease the perceived naturalness and efficiency of the conversation [15]. This thesis works towards the goal of reducing end-to-end latency by showing that masking energy in frequency bands will give comparable separation results to the more complex filter used in the baseline system.

Chapter 2

Fundamentals

The baseline system uses an approach that is based on computational auditory scene analysis (CASA). The following sections give a short introduction to the relevant parts that are needed to understand the baseline system. Chapter 2.1 starts out with a quick overview of the human auditory system. This is followed by the description of the perceptionally motivated *gammatone bandpass filter* (section 2.2.1) that makes use of an idealized model of the nerve responses in the inner ear. These bandpass filters are used to calculate the *cochleagram* (section 2.2), a time-frequency (T-F) representation of the input signal. CASA suggests a *masking* approach on this representation to group and separate multiple signal sources. This is explained in section 2.3. With the help of the mask, a clean T-F representation of the signal can be recovered. For human consumption, a time domain representation has to be *reconstructed* (section 2.2.3).

The quality of the separated signal is scored using an automatic metric. This short-term objective intelligibility score is explained in detail in section 2.4.

Since calculation of the full cochleagram (as well as inversion) is rather complex and computationally expensive, an alternative that relies on weighting the output bins of a discrete Fourier transform in proportion to a gammatone filterbank's magnitude gain is explored in the evaluation part. The construction of the weighting matrix is explained in section 2.5.

2.1 The human auditory system

Sound travels through air in form of a pressure wave. This wave is transferred into nerve responses by the human ear. The ear consists of the outer, middle, and inner ear. The sound wave entering the outer ear cause the eardrum to vibrate. Three tiny bones in the middle ear transfer these vibrations into movements of the liquid that fills the inner ear. The inner ear consists of the *cochlea*, a coiled,

liquid filled tube that is divided along its length by two membranes. Along its length, the basilar membrane varies in mass and stiffness. When the liquid in the inner ear moves, this variability makes the membrane vibrate at different resonance frequencies depending on the position. This movement is transferred into neural impulses by the inner *hair cells* which are displaced when the basilar membrane moves. Because the basilar membrane's resonant frequency changes based on position, the hair cells exhibit *frequency selectivity* [28].

2.2 Cochleagram

Speech separation is a subdomain of computational auditory scene analysis (CASA). The first step of such a system is to convert a time-domain waveform into a time-frequency (T-F) representation. One such representation is the cochleagram which is computed using an array of band pass filters that each model the frequency selectivity and nerve response of a single hair cell. The difference between the cochleagram and the more common spectrogram is illustrated in figure 2.2.

2.2.1 Gammatone bandpass filter

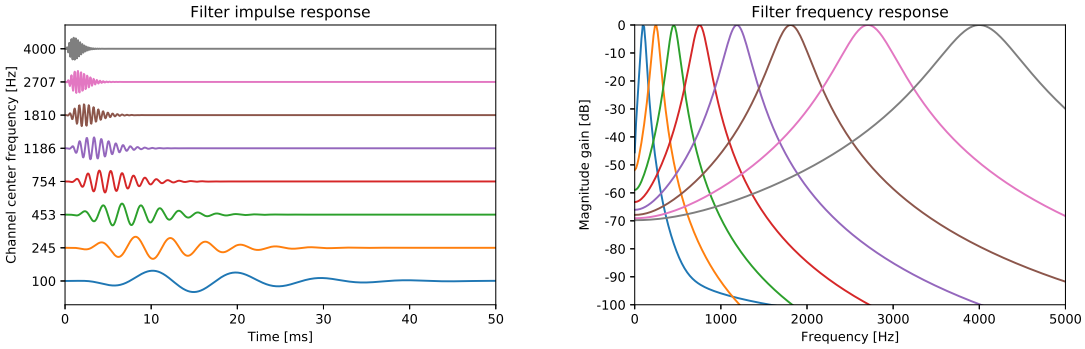
A popular choice for modelling the impulse response of a hair cell is the gammatone filter, which is a combination of the gamma function and a tone:

$$g_{f_c}(t) = t^{N-1} \exp(-2\pi t b(f_c)) \cos(2\pi f_c t + \phi) u(t) \tag{2.1}$$

In the equation above, N is the filter order, f_c is the filter center frequency in Hz, and ϕ is the phase. $u(t)$ is the unit step function which is $u(t) = 1$ for $t \geq 0$ and 0 otherwise. $b(f_c)$ is the bandwidth of the filter. Experiments have shown that $N = 4$ provides a good match to the human auditory filter shape [28, p. 16]. The bandwidth of a gammatone filter used to model the human ear is typically chosen to be $b(f_c) = 1.019 \text{ ERB}(f_c) \approx 24.7(4.37 f_c + 1)$.

In this thesis, the gammatone filter was implemented by means of convolution of the input signal and the finite impulse response of the gammatone function. However, it is worth noting that efficient implementations of the fourth-order gammatone filter in form of a second-order all-pole approximation exist [24]. The filter can also be implemented as a cascade of four filters of first order, as described by Holdsworth [9].

For calculating the cochleagram, multiple filters with differing center frequencies are arranged in a parallel filter bank. The output of the filterbank is half-wave rectified ($\hat{x} = \max(x, 0)$) to account for the fact that the hair cell nerves only fire when the hairs move in one direction. The center frequencies of the filters are



(a) Impulse response

(b) Frequency response

Figure 2.1: Impulse (2.1a) and frequency (2.1b) response of a gammatone filter bank with 8 filters, centered at equally spaced points between 100 Hz and 4 kHz on the ERB scale.

spaced at equidistant points on the logarithmic ERB-rate scale (explained in the next section). This causes the filters to be more closely spaced at low frequencies.

Figure 2.1 shows the impulse and frequency response of a gammatone filterbank with 8 filters.

2.2.2 Equivalent rectangular bandwidth

Equivalent rectangular bandwidth is a psychoacoustic measure for approximating the frequency-dependent bandwidth of the filters in human hearing. It makes the simplified assumption that these filters are implemented by rectangular bandpass filters. The bandwidth of the rectangular bandpass filter is chosen such that it has the same peak and passes the same amount of power for an input of white noise [28]. A good approximation of measurements performed on the human ear is given by

$$\mathrm{ERB}(f) = 24.7(4.37f + 1) \tag{2.2}$$

As mentioned in the last section, for building a filterbank it is convenient to space the center frequencies of the filters according to their ER-bandwidth. The ERB-scale $E(f)$, as given by

$$E(f) = 21.4 \log_{10}(0.00437f + 1) \tag{2.3}$$

can be used to find the the center frequencies (in Hz) of an n-filter cochleagram filterbank with lower cutoff frequency f_l and upper cutoff frequency f_u chosing n

c	1	2	3	4	5	6	7	8	9	10
f_c [Hz]	0.00	111.88	278.46	526.48	895.76	1445.58	2264.22	3483.10	5297.91	8000.00

Table 2.1: Center frequencies of 10 filters evenly spaced on the ERB-scale between 0 Hz and 8 kHz. Note how the filters are more tightly spaced at lower frequencies.

linearly spaced points between $E(f_l)$ and $E(f_h)$ and passing the resulting points through the inverse transform $E^{-1}(f)$. See table 2.1 for an example.

2.2.3 Reconstruction

For practical purposes outside of automatic speech recognition, the clean speech signal must be reconstructed from the masked cochleagram. That way the quality of the separation can be assessed by a human judge. There also exist automated metrics for scoring the intelligibility of time-frequency weighted noisy speech (see section 2.4) which require the input signal to be available as a waveform.

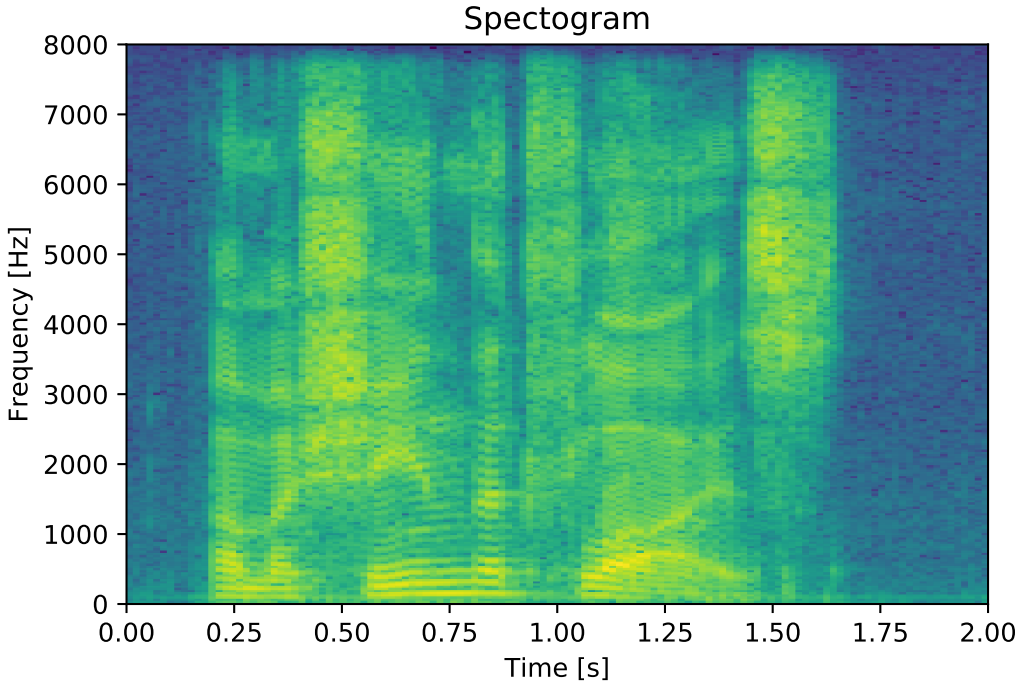
Similar to a short-time Fourier transformation, no information is lost when transforming the input signal to a cochleagram representation. Perfect reconstruction of the time signal is possible by running the steps required to obtain the cochleagram in reverse order. The most difficult part is to reconstruct the information that is removed during half-wave rectification. A good overview of this fairly involved process is given in [23]. In the implementation, we use a shortcut that skips half-wave reconstruction to achieve almost perfect results. See section 3.7.1 for more details.

2.2.4 Smoothed cochleagram

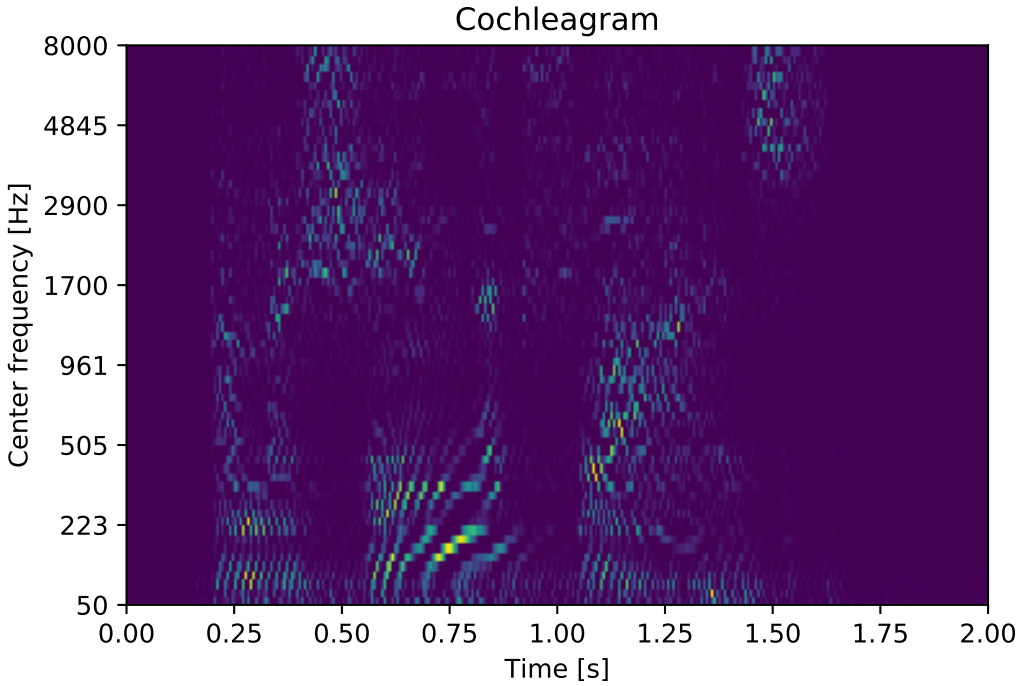
The smoothed cochleagram is computed from the cochleagram by calculating the average on overlapping sliding windows. For this thesis, a window size of 20 ms and 50% overlap were chosen. The result looks similar to a spectrogram. See figure 2.3 for an illustration of the information that is lost during smoothing and for the resulting smoothed cochleagram.

2.3 Masking

In CASA systems separation of the speech signal from background noise is achieved by masking the T-F representation of the mixture with a suitable mask. Two types of masks are commonly found in literature: The Ideal Binary Mask (IBM) and the Ideal Ratio Mask (IRM).

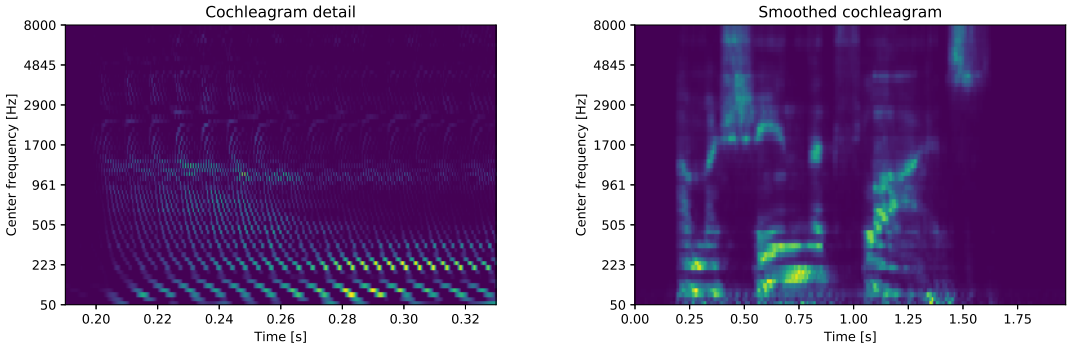


(a) Spectrogram



(b) Cochleagram

Figure 2.2: Spectrogram (2.2a) vs. cochleagram (2.2b) of a clean recording of the utterance of "a machine that flies."



(a) Detail view of cochleagram 2.2b

(b) Smoothed cochleagram with 20 ms window size and 10 ms window shift

Figure 2.3: Cochleagram detail and smoothed cochleagram.

The IRM is given by

$$\mathrm{IRM}(t,f) = \sqrt{\frac{s(t,f)}{s(t,f)+n(t,f)}} \tag{2.4}$$

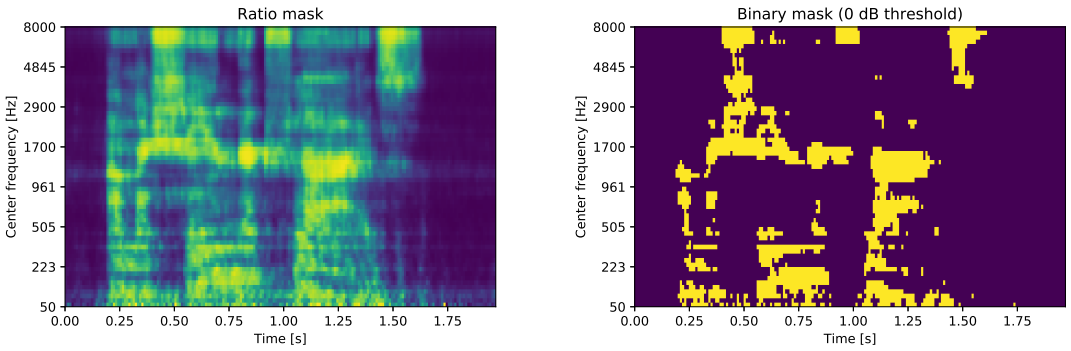
where $s(t,f)$ denotes speech energy at time t in frequency band f and $n(t,f)$ denotes noise energy. It is therefore a measure of the proportion of speech to total energy in the signal. At evaluation time, only the combined term $s(t,f)+n(t,f)$—the combined energy of the speech signal masked by the noise signal—is known.

The IBM only considers a T-F unit to belong to the speech signal if the difference of the signal and noise energy is higher than a given threshold. It is given by

$$\mathrm{IBM}(t,f) = \begin{cases} 1, & \text{if } s(t,f) - n(t,f) > \theta \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

The binary mask needs a carefully chosen threshold (θ in equation 2.5). For practical purposes, the parameter θ is often chosen to be 0 dB, i.e. the mask is 1 when a T-F unit carries more signal than noise energy. However, for best speech intelligibility a threshold of $\theta = -6$ dB is most effective [28, p. 23].

In order to obtain the separated signal, the mask can be applied to the T-F representation of the noisy signal by element-wise multiplication.



(a) Ideal Ratio Mask

(b) Ideal Binary Mask

Figure 2.4: Ideal ratio mask (2.4a) and ideal binary mask with $\theta = 0$ dB (2.4b) for the utterance "a machine that flies" overlaid with siren background noise.

2.4 Short Term Objective Intelligibility

In order to assess how different approaches of noise-reduction affects intelligiblity and how different algorithms stack up in respect to each other, a metric is needed. The use of human judges who rate algorithm output of course is slow and cost-prohibitive. A good metric should be automatic and allow for quick turn-around times. This metric should also be closely correlated with how human judges would rate intelligiblity of the separated audio.

The short term objective intelligiblity (STOI) metric lives up to these demands. The following paragraphs are a summary of the relevant parts of [26].

STOI is a function of the clean and degraded speech. It compares the short-term temporal envelope of a clean and degraded speech signal in critical frequency bands. In the following equations, x denotes the clean speech while y denotes the degraded speech. STOI can therefore only be calculated if the original, uncorrupted signal is known.

The original description of STOI assumes that x and y are recorded with a sample rate of 10 kHz as it captures all relevant frequency ranges of speech. However, the authors note that the metric can be adapted to other sample rates when the length of the analysis window (in ms) is kept constant.

Both signals are first segmented into frames with a length of 256 samples (25 ms) with 50% overlap. A Hann-window is applied to each frame before it is decomposed into the frequency domain via the discrete Fourier transform (DFT) that is padded to 512 samples. Since silent frames do not contribute to speech intelligiblity, these are discarded. Silent frames in this context are all frames which have an energy content less than 40 dB (the dynamic range of speech) of that of the frame with the maximum energy content.

STOI is calculated on 15 critical frequency bands. These bands are one-third octave bands with center frequencies evenly distributed between 150 Hz and 4.3 kHz. This leads to the definition of the *TF-unit*, which is obtained by grouping the energy in DFT-bins by band:

$$X_j(m) = \sqrt{\sum_{k=k_1(j)}^{k_2(j)-1} |\hat{x}(k,m)|^2} \tag{2.6}$$

$k_1(j)$ and $k_2(j)$ denote the indices of the lower and upper third-octave band edge (rounded to the nearest DFT bin).

These TF-units are then used to formulate the short-time temporal envelope of x for frame m in band j.

$$x_{j,m} = \left[X_j(m-N+1), \quad X_j(m-N+2), \quad \ldots, \quad X_j m\right]^T \tag{2.7}$$

$N = 30$ is equivalent to an analysis window of 384 ms.

The temporal envelope of y is calculated in the same fashion.

Before the envelopes are compared, the envelope of the degraded speech, y_j is first normalized to account for global differences in volume which do not impact speech intelligibility. The envelope of y is also clipped to a lower bound, so that the model is robust against TF-units that are severely degraded.

$$\bar{y}_{j,m}(n) = \min(\frac{||x_{j,m}||}{||y_{j,m}||} y_{j,m}(n), (1 + 10^{\frac{-\beta}{20}}) x_{j,m}(n)) \tag{2.8}$$

β is the lower SNR bound ($\beta = -15$ dB) before clipping sets in.

The sample correlation coefficient of $x_{j,m}$ and $\bar{y}_{j,m}$ is now computed.

$$d_{j,m} = \frac{(x_{j,m} - \mu_{x_{j,m}})^T (\bar{y}_{j,m} - \mu_{\bar{y}_{j,m}})}{||x_{j,m} - \mu_{x_{j,m}})^T|| \; ||\bar{y}_{j,m} - \mu_{\bar{y}_{j,m}}||} \tag{2.9}$$

Where $\mu_{(\cdot)}$ denotes the mean value of the components of the vector.

With these prerequisites, the STOI score can now be formulated as the average over all correlation coefficients across all J bands and M frames.

$$d = \frac{1}{JM} \sum_{j,m} d_{j,m} \tag{2.10}$$

Listening tests conducted by the authors of STOI have shown that there is a strong monotonic relationship between intelligibility as rated by human judges and an increase in STOI.

It is important to keep in mind that STOI only compares frequency content in the band between 150 Hz and 4.3 kHz. Noises outside of this frequency range

are not taken into consideration as they generally do not affect intelligibility but might still be perceived as annoying.

2.5 Gammatone weighted spectrogram

The approach of calculating the full cochleagram as described in the previous sections is computationally expensive and costly to implement. An approach that only relies on a discrete Fourier transform (DFT) [3] and matrix multiplication is described in the this section. The main idea is to weight the DFT bins in proportion to the magnitude gain of a gammatone bandpass filter [5]. This approach is similar to the computation of log mel features [18] often used in DNN-based automatic speech recognition systems.

First, the input signal is Hann-windowed with n point window size and 50% window overlap. Then the short term n-point DFT is calculated on each window a^t:

$$A_k^t = \sum_{m=0}^{n-1} a_m^t \exp\left(-2\pi i \frac{mk}{n}\right) \quad k = 0, \ldots, n-1 \tag{2.11}$$

Since the input windows are real-valued, the output of the DFT is Hermetian-symmetric. The negative-frequency terms are therefore redundant and can be discarded. The first FFT bin A_0^t contains the zero-frequency term ("DC content") of the signal and is also discarded. This leaves us with $\frac{n}{2}$ points $\hat{A}^t = [A_1^t, A_2^t, \cdots, A_{\frac{n}{2}}^t]^T$ for each window.

The square of the absolute value of each window component is then calculated. The resulting vectors are stacked to establish the *power spectrogram* A:

$$A = \left[|\hat{A}^1|^2, \quad |\hat{A}^2|^2 \quad \cdots \quad |\hat{A}^t|^2\right] \tag{2.12}$$

Each bin $|\hat{A}_i|^2$ of the power spectrogram is now weighted according to what the magnitude gain of a gammatone filter of the same center frequency would have been for the frequency corresponding to the DFT bin. This can be expressed by the matrix multiplication $G = WA$. W is straight-forwardly calculated by transforming the impulse response of m gammatone filters evenly spaced on the ERB scale using an n-point DFT.

$$W = \begin{bmatrix} |DFT\{g_{f_1}(t)\}|^2 \\ |DFT\{g_{f_2}(t)\}|^2 \\ \vdots \\ |DFT\{g_{f_m}(t)\}|^2 \end{bmatrix} \tag{2.13}$$

Chapter 3

Implementation

The experiments were conducted using the Python programming language. Feature extraction (cochleagram and gammatone-weighted spectrogram), IRM estimation, audio reconstruction and the STOI metric (adapted to work with audio with a sampling rate of 16 kHz) were implemented in terms of the NumPy and SciPy libraries for scientific computing [12].

The neural networks were implemented using the PyTorch [21] framework. The PyTorch package offers GPU accelerated Tensor math paired with automatic tape-based gradient calculation. Building on this foundation, PyTorch ships with many standard components (layers, non-linearities, loss functions, optimizers) for quickly experimenting with deep neural networks.

3.1 System Overview

3.2 Dataset

The predictors are trained from data which is synthesized from the MUSAN (music, speech and noise) corpus [25] which is freely available. This dataset consists of music, speech recordings in twelve different languages, and a large set of naturally occuring and technical noises. Its primary intention is that of being a training corpus for voice activity detection.

For the following experiments, only the speech portion, which consists of read speech (Librivox recordings) and US government recordings (hearings, committees, and debates), and the set of noises, which ranges from beeps emitted from technical equipment, to ambient sounds such as rain and road noise). The speech portion of the corpus amounts to 60 hours of recordings, about two thirds of them are English. The duration of available noise examples is about one-tenth (6 hours) of that of speech.

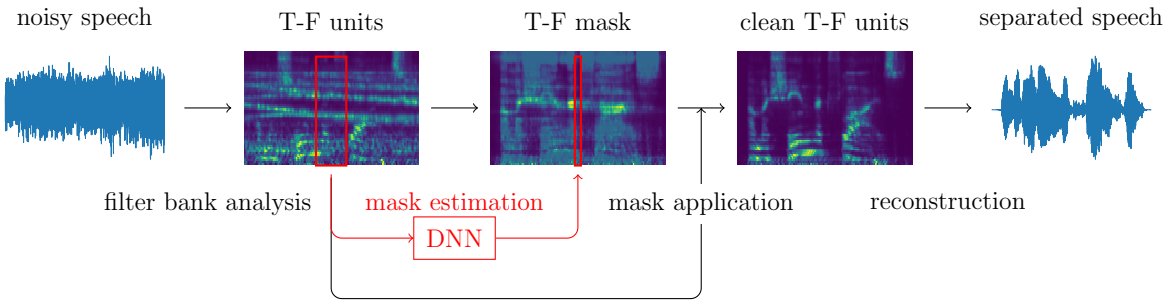


Figure 3.1: Schematic overview of the architecture used for speech separation. The input waveform is transformed into a time-frequency (T-F) representation. A mask is then estimated by repeatedly applying a DNN on sliding windows along the time axis. The resulting T-F mask is used to weight the T-F representation of the noisy signal. In a final step, audio is reconstructed from the weighted T-F units.

All recordings are available in the WAV format. They offer 16 bit of resolution. The sample format is signed little endian PCM. The sampling rate is 16 kHz which is more than sufficient to capture speech signals, whose energy is concentrated below 4 kHz [4].

Before any of the recordings were used for further calculations, they were converted to IEEE 754 single-precision floating point numbers, normalized to the range $[-1, 1]$ via the following simple (sample-wise) transformation:

$$\hat{x}_n = \frac{x_n}{\max_j |x_j|} \tag{3.1}$$

This makes the data easier to work with as it avoids problems (numerical overflows, clipping, wrapping around) commonly encountered with arithmetic on data types that have a limited range.

From these examples of clean speech and noise, a dataset for training and evaluating is synthesized. However, some examples of both categories are held out to serve as unseen data for testing the predictor's ability to generalize to new unseen data.

3.3 Synthesizing Noisy Data

The goal of this thesis is to train a predictor that can estimate a time-frequency mask (see section 2.3) for a given short window of noisy speech.

A virtually unlimited amount of training examples of speech corrupted by noise can be generated by additive mixing of the data described in the previous section.

The clean speech signal needs to be known in order to calculate the proper mask for the corrupted signal.

The training corpus for the predictor is constructed as follows: The Cartesian product of the set of speech and noise files is taken. This results in the set of all possible combinations of speech and noise recordings. For these tuples of noise and speech recordings, the average signal energy of each recording of length N (in samples) is computed by calculating the mean of the squares of each sample $x[n]$.

$$P(x) = \frac{\sum_{n=0}^{N-1} x^2[n]}{N} \tag{3.2}$$

With this information, the speech and noise examples can be mixed at a predefined ratio of signal energy to noise energy. This measure is commonly known as signal-to-noise ratio (SNR) and usually expressed on the logarithmic decibel scale (dB).

$$\mathrm{SNR}(s, n) = \frac{P(s)}{P(n)} = 10 \log_{10} \frac{P(s)}{P(n)} \mathrm{dB} \tag{3.3}$$

$P(s)$ and $P(n)$ are the power of the speech and noise signal respectively.

Chen et. al train their models to predicts masks for mixtures of -2 dB SNR [2]. In order to obtain mixtures at a predetermined target SNR, the energy of the noise signal can be scaled by a factor computed from speech and noise energy, as well as the target SNR:

$$g = 10^{\frac{-\text{target SNR}}{10 \text{ dB}}} \frac{P(s)}{P(n)} \tag{3.4}$$

In order to maximize the number of examples of speech corrupted by different noise recordings, both recordings are time shifted against each other. This is easily achieved by cutting both speech and noise into shorter segments using a sliding window, followed by taking the Cartesian product of the resulting windows. We used windows of ˜310 ms with 50% overlap.

On all possible combinations of windowed speech and noise, the mixture is then obtained by calculating the weighted sum of the clean speech window s and the noise window n. The amplitudes of n are scaled by the factor $\sqrt{g}$ (see equation 3.4).

$$m = \frac{s + \sqrt{g}n}{1 + \sqrt{g}} \tag{3.5}$$

From m and n, the smoothed cochleagram (section 2.2) and gammatone weighted spectrogram (section 2.5) are computed. From these, the masks (section 2.3) are calculated. The mixture m is saved to disk in three representations: The smoothed cochleagram and gammatone weighted spectrogram T-F units are required for

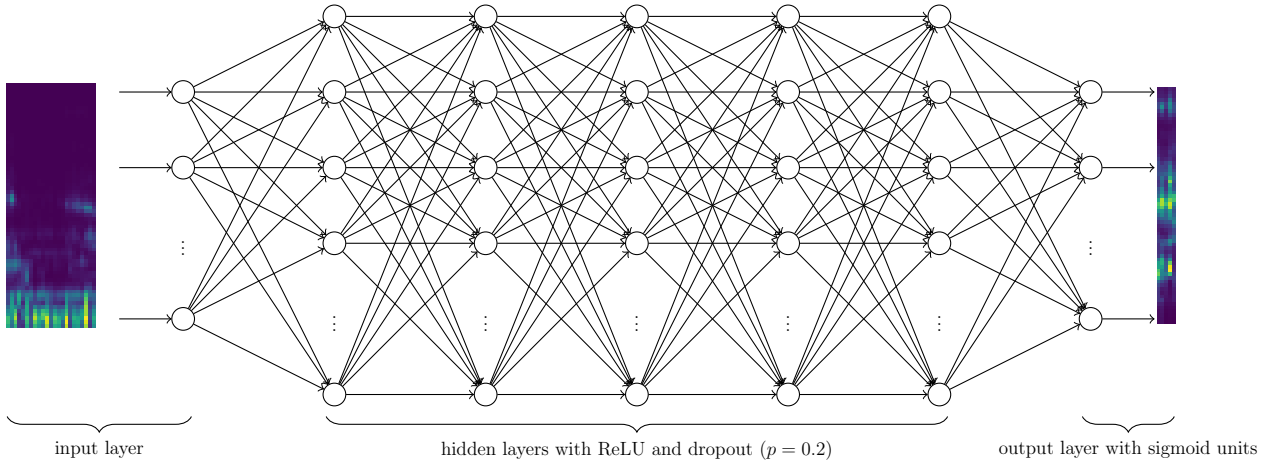


Figure 3.2: Fully-connected feed forward architecture of the baseline model

the baseline architecture (section 3.4.1). The raw waveform of the mixture is needed for the experiments in section 3.4.3. The IRM mask is calculated from both cochleagram and gammatone weighted spectrogram T-F units and saved for both representations.

In order to monitor training (section 3.5) and for comparing different trained predictors, the resulting dataset is split into three parts. A standard 60%/20%/20% split into training (train), development (dev) and test sets is employed.

3.4 Network Architectures

3.4.1 Baseline Model

The baseline model from [2] is a fully connected feed forward network with 5 hidden layers (see figure 3.2). The network has 64×23 input units and 64×5 output units. It therefore can predict a T-F mask for 5 frames based on an input context of 23 frames. The input layer feeds into 5 hidden layers with 2048 ReLU units each. The output unit nonlinearity is the logistic (sigmoid) function.

During training, 20% dropout is used for the network's hidden layers. For every minibatch there is a one in five chance that the hidden unit will not produce an output. This has proven to be an effective technique for preventing co-adaptation of hidden units and therefore counteracts "overfitting" on the training data [8].

The objective function that is being optimized is the mean-square-error loss function between the target T-F mask in the training set and the T-F mask that is predicted by the network.

The baseline model uses T-F units based on the smoothed cochleagram as

described in section 2.2.4.

3.4.2 Baseline Model with Gammatone Spectrogram T-F units

In order to check the assumption that the temporal information in the cochleagram is vital for the ability to separate speech (section 2.2.1), the network from the previous section is reused with input and output features calculated from the gammatone weighted spectrogram (section 2.5).

3.4.3 Cochleagram Filter Bank as CNN

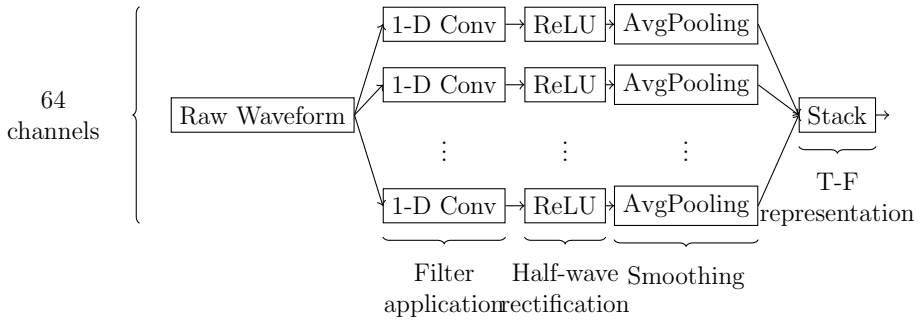


Figure 3.3: Schematic of the convolutional neural network used for obtaining the cochleagram from the raw waveform. The stacked output is fed into the baseline model described in section 3.4.1.

As mentioned in section 2.2, the cochleagram can be computed by convolution of the audio signal with the impulse responses of the gammatone bandpass filters. This approach lends itself to an implementation as a convolutional neural network (CNN) that predicts the T-F mask directly from raw (unprocessed except for range normalization) audio data.

The smoothed cochleagram can be computed by a 3 layer neural network as illustrated in figure 3.3. The input data is passed into a convolutional layer for calculating the filter response of the hair-cell model. This layer feeds into a ReLU non-linearity for half-wave rectification. The smoothed cochleagram is then obtained by running the half-wave rectified output through an average-pooling layer. This architecture has the interesting benefit that the impulse responses of the filters (the *kernel* of the convolutional layer) can be learned using backpropagation just like the other parameters of the network. The automatic learning of filters instead of engineering features by hand has lead to breakthrough results in image recognition [16] and has also recently been considered for automatic speech

recognition as a successor to the traditional log mel and mel-frequency cepstral coefficient features computed from the power spectrogram of the waveform [22].

Thinking of the learned weights of the convolutional layer as the impulse response of a filter that can be transformed to the frequency domain is a very natural way to better understand what frequencies in the input signal the neural network is learning to pay attention to.

There are three hyper parameters that need to be set for the architecture above: 1) the number of filters, 2) the size of the filter kernels (length of the impulse response of the filter in points) and 3) the window of the average pooling layer (in points).

For this implementation, the hyper parameters were chosen to match those of the baseline model in section 3.4.1: There are 64 filter output channels with a kernel size of 800 points (50 ms at 16 kHz sample rate) each. This large filter is governed by the long impulse response of a 50 Hz gammatone filter. The average pooling layer operates on windows of 320 points (20 ms), with a stride (shift) of 160 points (10 ms).

When this network is fed with 3039 points (190 ms) of raw audio at 16 kHz, it calculates the smoothed cochleagram of 23 frames with 64 channels. This output is then fed as input features into the same fully-connected network that has been described in the previous section.

The filter weights can either be randomly initialized, or bootstrapped with the impulse response of 64 gammatone filters evenly spaced along the ERB scale (see section 2.2). Experiments with both types of initialization are conducted in section 4.

3.5 Training

During training (local) optimal weights for the network are calculated via backpropagation that minimize the objective function on the training set. The gradients for backpropagation are automatically calculated by the PyTorch framework.

Adam, a variant of stochastic gradient, was chosen for optimizing the objective functions of the networks. It automatically adapts the learning rate and therefore leads to faster convergence without the need for manual adjustments [14].

The networks are very sensitive to the initial weight distribution. With poorly chosen weights, the training gets stuck early on in a local maximum. When the weights are too small, the input signal diminishes when passing through the layers and the gradient disappears. The other extreme is that the gradient "explodes": It becomes too large to be useful for optimizing the weights. Good results were achieved when using Glorot (Xavier) initialized weights for the fully-connected portion of the network. Glorot initialization is a random initialization with zero

mean and a variance for each unit that is dependent on the number of units it feeds it output into [6].

Another trick was employed to help the network's learning performance. The input and output vectors were normalized per dimension. First, the mean μ and standard deviation σ of the data set is calculated. This can efficiently done in a single pass over the data set using Welford's algorithm [30]. Then the standard score of each training vector is then calculated on each dimension:

$$\hat{x} = \frac{x - \mu}{\sigma} \tag{3.6}$$

Next the data is scaled to the range $[0, 1]$ using the following equation ($x_{\min}$ and $x_{\max}$ are calculated on the standard scores of the vector):

$$\bar{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.7}$$

This serves the purpose of the output being within the output range of the sigmoid function which is used for the network's output layer. It also makes the backpropagation converge faster [17, section 4.3]. Of course, this normalization has to be inverted when evaluating the network to predict a mask on real data.

All networks were trained on a single GeForce GTX 970 graphics processing unit (GPU) on a machine with 16 gigabytes of main memory.

3.6 Predicting longer masks

The predictor works on frames of 23 input frames (either already in a T-F representation or raw audio) and makes a prediction for 5 frames of the T-F mask, centered within that window. In order to be able to predict masks for longer audio recordings, the model is repeatedly evaluated on a sliding window across the full length of the input sequence.

Two modes of operation are possible. The first is to advance with a window shift of 1 which causes the output of the network to overlap. Each frame of the mask is predicted multiple times which likely makes the average of the overlapping frames a higher quality mask. If computational complexity is a concern, the frame shift can be set to the output width (5 frames), causing each unit of the mask to be predicted only once.

Since the predicted mask is centered within the input frame, the output of the predictor "lags behind" the input by 9 frames. To account for this offset the input has to be padded by 9 frames in the beginning. Conversely the predictor will run out of data before the full mask can be predicted. Padding at the end of the input sequence by the same amount is therefore also necessary.

3.7 Audio reconstruction

The quality of the predicted mask is assessed on the separated audio. Therefore, the mask has to be applied to the noisy mixture. Depending on whether cochleagram or gammatone weighted T-F units are used, two separate strategies for applying the mask to the mixture are employed. The strategies for transforming the masked signal to the time domain are detailed in the two subsections below.

3.7.1 Cochleagram T-F units

For cochleagram T-F units, the high-resolution cochleagram of the mixture has been calculated as part of the feature extraction process before feeding the windows of 23 frames to the smoothed cochleagram to the predictor. The predictor outputs an estimated IRM for 5 smoothed frames. These frames are then stretched out in time by means of linear interpolation. The resulting mask has the same shape as the high-resolution cochleagram. The mask can then be applied by component-wise multiplication. This "clean" cochleagram representation of the separated signal is then transformed back into the time domain by a simplified version of cochleagram inversion (see 2.2.3). In a first step, the gain added by AGC is divided out. Next, the filters are run "backwards in time": The impulse response of each filter is inverted in the time domain and then convolved with the clean cochleagram. The resulting system responses are summed to reconstruct the time-domain signal. Let $A \in \mathbb{R}^{m \times n}$ be the matrix of n impulse response coefficients of the m analysis filters used to compute the cochleagram $C(t, f)$. Further let $\hat{A}$ represent the same matrix with inverted column order (impulse responses running backwards). The separated signal $x(t)$ is thus being reconstructed as

$$x(t) = \sum_{i=1}^{m} (C(\cdot, i) * \hat{A}(\cdot, i))(t) \tag{3.8}$$

3.7.2 Gammatone spectrogram T-F units

As part of calculating the gamma-tone weighted power spectrogram, the complex n-point short time spectrogram of the input signal is calculated. This representation can be transformed back into the time domain without loss of information.

The T-F mask however is estimated on filter bank energies that group several FFT bins into a single coefficient by means of a weighted sum (see 2.5). This transformation makes the spectrogram lose its fine structure. The process can be lossily reversed by "smearing" each filter's energy across the bands it captured. The mathematical formulation is straight-forward. Given the weighting matrix

$W \in \mathbb{R}^{m \times n}$ and the gammatone weighted T-F mask $G \in \mathbb{R}^{m \times t}$, an "inverted" mask $G' \in \mathbb{R}^{n \times t}$ can be calculated by the equation below.

$$G' = W^T G \tag{3.9}$$

G' can then be used to weight the components of the complex spectrogram of the mixture prior to transforming back to a time-domain signal using an inverse DFT. The effect is that the fine structure of the mixture signal is preserved, while the frequency bands containing noise are reliably blocked during reconstruction.

Chapter 4

Evaluation

4.1 Oracle Experiments

Oracle experiments were performed to determine the upper bound of achievable performance in terms of the STOI metric (see section 2.4) for the two feature representations. For this purpose, the T-F mask (see section 2.3) is directly calculated from the mixture and the clean speech signal. This mask is then multiplied with the T-F representation of the mixture and passed through the reconstruction path to obtain the filtered audio signal. In other words, the oracle experiment measures the loss incurred by transforming the input signal into T-F units and subsequent reconstruction.

As can be seen in table 4.1, the cochleagram features offer a slight advantage over gammatone spectrogram features in terms of measured intelligibility when the T-F mask can be calculated directly from the mixture and the clean speech signal. In a subjective listening test however, it seems like the gammatone spectrogram mask better blocks out background noise. Faint traces of background noise can still be heard when applying the smoothed cochleagram mask. This effect is not perceivable when calculating the mask directly on the full-resolution unsmoothed cochleagram T-F representation. This bleeding through of background noise can therefore be attributed to the linear interpolation step that is performed to upsample the smoothed mask prior to application (see 2.2.3). It is likely that the result can be further improved by using more sophisticated interpolation such as Lanczos filtering.

For both cochleagram and gammatone spectrogram features the intelligibility of the reconstructed signal degrades proportionally with increasingly lower SNR. At the same time however, the relative gain over baseline intelligibility increases.

Feature	Noise	SNR [dB]	Filtered	Unprocessed	Delta [%]
Cochleagram	**Siren**	**-10.0**	0.92	0.70	32.36
		-2.0	0.96	0.83	16.13
		0.0	0.97	0.86	13.14
	Speech Babble	**-10.0**	0.91	0.58	58.23
		-2.0	0.95	0.75	26.01
		0.0	0.95	0.79	20.54
Gammatone-weighted	**Siren**	**-10.0**	0.93	0.70	33.41
		-2.0	0.96	0.83	15.77
		0.0	0.96	0.86	12.68
	Speech Babble	**-10.0**	0.90	0.58	56.03
		-2.0	0.94	0.75	25.65
		0.0	0.95	0.79	20.32

Table 4.1: STOI scores obtained by oracle masks using cochleagram vs. gammatone-weighted features and reconstruction paths

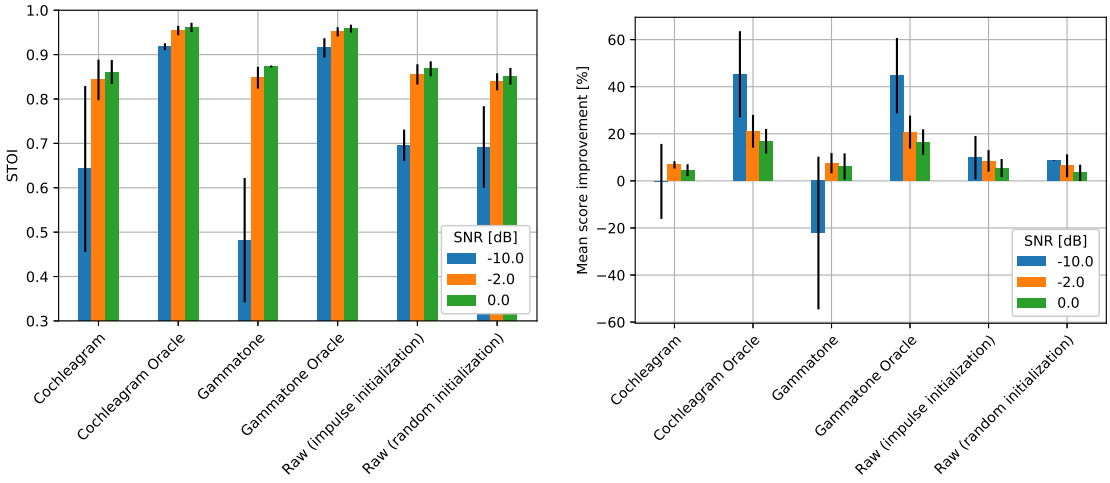
4.2 SNR conditions not encountered in training

The training data was mixed at a fixed signal-to-noise ratio of -2 dB. The signal-to-noise ratios that will be encountered in a real use-case will change based on the environment. The system will likely constantly encounter signals mismatched to the training data. Therefore it is important that the models are robust to variations in the ratio of speech to interference energies. In order to test the ability to generalize to SNRs not encountered in training, the models were additionally evaluated at -10 dB and 0 dB SNR.

As can be seen in figure 4.1, the choice of input features is an important factor for model performance. At -10 dB SNR the fully-connected model trained on cochleagram features significantly outperforms the same architecture trained on the simpler gammatone weighted spectrogram. When looking at the relative "improvements" in STOI, it becomes clear that the gammatone weighted model actually decreases intelligibility even further in low SNR conditions. The model trained on cochleagram features will also not improve intelligibility, but at least not degrade it further when taking the mean over all tested examples.

Surprisingly, both CNNs trained on raw waveforms manage to predict masks that improve STOI by almost 10% on average. The model that had its filters initialized with the impulse responses of the gammatone filters exhibits lower variance and therefore works more predictably at -10 dB.

The best performing model uses raw waveforms as the input feature representation. The output is an IRM with gammatone weighted spectrogram T-F units.



(a) Mean STOI score

(b) Mean change in STOI

Figure 4.1: Mean STOI score (4.1a) and relative change (4.1b) attained by the models on the test set at three different signal-to-noise ratios

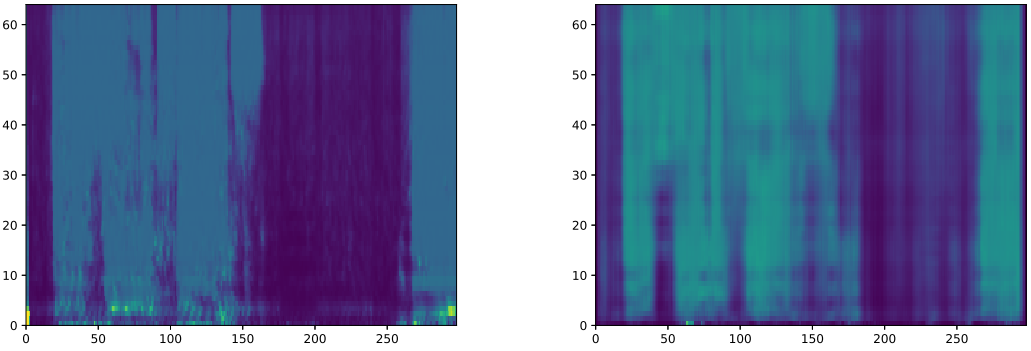
This is a strong indicator that the input feature representation matters more than the representation of the outputs, even tough the the oracle experiments (see section 4.1) show that slightly better separation would be possible when using cochleagram masks.

4.3 Degradation of mixtures with high STOI

There is evidence that the T-F masks predicted by the models are too fuzzy. Parts of the mixture that belong to the speech signal will be misclassified as noise. This shortcoming is unproblematic when the mixture has low intelligibility, as even an imperfect mask will still improve intelligibility by removing more noise than speech. When the speech signal however is overlaid by a "weak masker," i.e. interference that barely affects intelligibility, filtering out parts of the speech signal manifests in overall worse intelligibility. Examples for this phenomenon are the "Car" and "Interference" noise examples in table A.1. Figure 4.2 shows the predicted and oracle cochleagram mask for the latter.

4.4 Poor generalization

During evaluation, many examples were found for which no useful mask could be predicted by any of the systems. The resulting reconstructed audio therefore



(a) Oracle mask

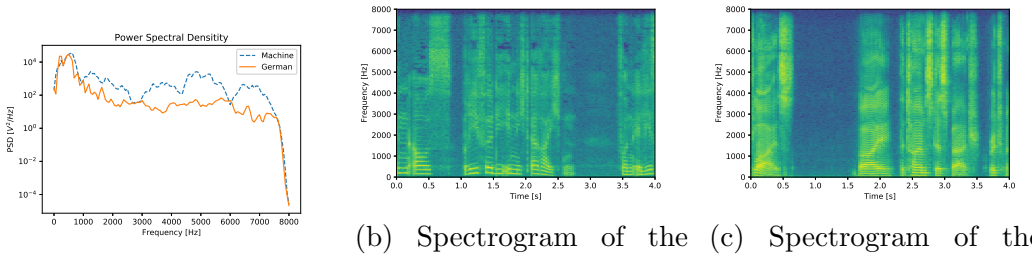
(b) Predicted mask

Figure 4.2: Comparison of a cochleagram oracle mask (4.2a) to a predicted "fuzzy" (4.2b) mask that only loosely tracks the T-F envelope of the speech signal. Note how the oracle mask has finer resolution of energy ratios. The abscissa represents time in multiples of an analysis window, 64 different filter channels along the ordinate.

had poorer intelligibility than the unprocessed audio. It is unclear what causes this problem. One attempt at explanation is, that for the affected examples, the clean signal has low dynamic range and less frequency content over 1.5 kHz (see figure 4.3). As a consequence, the input signal might lack redundancy in the form of harmonic content that helps the network identify speech when the lower frequency bands are distorted. This explanation would be in line with the findings of Chen et. al where they show that the network will learn coefficients for harmonics detectors [2].

4.5 Filters learned by CNN

In section 3.4.3 it was shown that the cochleagram can be calculated by means of a 3-layer CNN when the weights of the convolutional layer are tied to the coefficients of adequately chosen gammatone functions. It is however also of interest to learn these weights from scratch using backpropagation during model training. The aim is for the network to find features in the input data that are better suited for speech separation than cochleagram features. For that purpose, the task was to predict a gammatone-weighted T-F mask directly from the raw waveform. The weights of the convolutional layer were randomly initialized. Figure 4.4 shows both the impulse and frequency response for 16 of the 64 channels that were learned during training. It is illuminating to compare these to the responses of a traditionally



(a) Power spectral density (b) Spectrogram of the "German" example (c) Spectrogram of the "Machine" example

Figure 4.3: Power spectral density estimated using Welch's method [29] (4.3a) and spectrograms of the "German" (4.3b) and "Machine" (4.3c) test cases. The "German" example has most of its frequency content concentrated below 1 kHz and can not be separated by the system.

designed cochleagram filterbank as seen in figure 2.1 on page 11. It is immediately obvious that the learned filters aren't band-pass filters with a single passband but rather a combination of multiple band-pass filters. The attenuation outside of the passbands is also much less pronounced (note the non-logarithmic scale in figure 4.4) than in the gammatone filters. The frequency response is obviously less smooth than the ideal gammatone filter. The convolutional layer learned to put more filters into the lower frequency ranges than at higher frequencies. This is likely a result of the output layer of the network predicting a gammatone-weighted mask that has its coefficients spaced linearly on the ERB scale. Another interesting feature of the learned filters is that they appear to be receptive to harmonics: the peaks of the magnitude gain are evenly spaced on the frequency axis. This means that a filter will still detect the presence of an acoustic feature even when it is masked by noise as long as the sound's harmonics are still discernible.

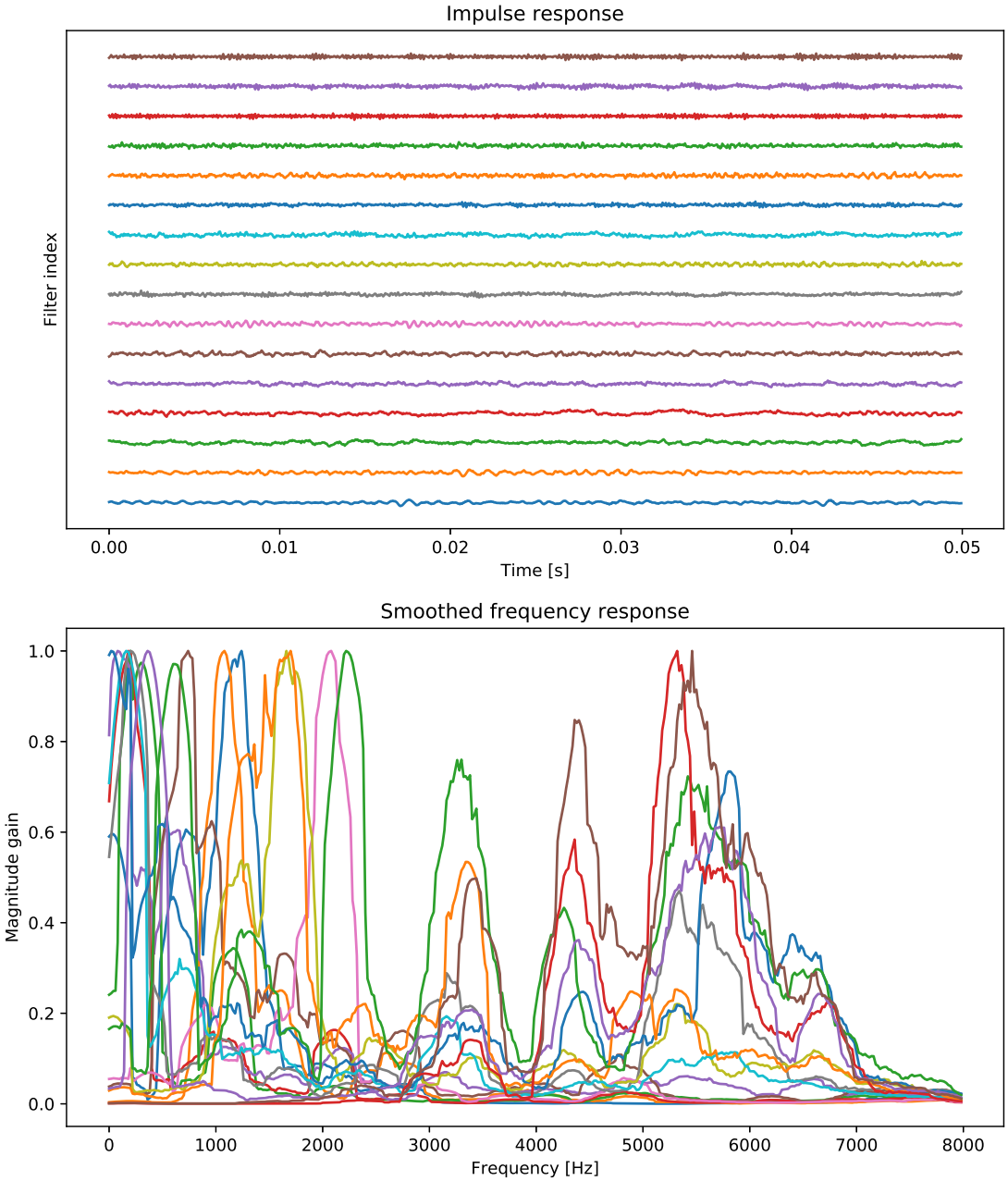


Figure 4.4: Impulse and corresponding Gaussian-smoothed frequency response of every 4^{th} filter that was learned by the network with random initialization. The plot was created by sorting the filters by the centroids of their frequency response. The frequency responses were smoothed for better visual clarity.

Chapter 5

Summary and Future Work

5.1 Summary

In the previous chapters, a full end-to-end system for separating speech from background noise based on the computational auditory scene analysis approach of masking a time-frequency (TF) representation of the mixture signal has been presented. The mask was automatically estimated by two different neural network architectures based on three different combinations of input and output feature representations. The examples for supervised training of the predictors were obtained by additive mixing of clean speech recordings with background noise at a defined signal-to-noise ratio (SNR).

Full reconstruction paths from T-F representation to waveform were implemented for both the cochleagram and gammatone-weighted spectrogram features. The reconstructed separated audio from all approaches was graded against one another using the automated short term objective intelligibility (STOI) metric which has been shown to have good monotonic relationship to intelligibility scores given by human judges.

In oracle experiments masks based on the cochleagram and on gammatone-weighted spectrogram features show virtually identical separation performance when measured in STOI. The linear interpolation used in application of the smoothed cochleagram mask however is audible as a faint bleeding-through of the background noise. In this subjective listening test, the reconstruction path of the gammatone-weighted spectrogram better attenuated the background noise.

It was found however that for the baseline model with 5 fully-connected hidden layers, the features based on the smoothed cochleagram offer superior performance over the gammatone-weighted spectrogram features when predicting masks on mixtures with SNR lower than those encountered in training.

As a logical consequence of calculating the cochleagram by means of convolu-

tion of the finite impulse response of an array of gammatone filters and the input signal, an architecture based on a 1-d convolutional layer with ReLU nonlinearities followed by averaging pooling was introduced for moving the calculation of input features from the raw waveform into the neural network itself. With weights of the convolutional layer tied to the impulse responses of a gammatone filter bank, this architecture will calculate the smoothed cochleagram. The impulse response (weights of the kernel) of the filters can however also be optimized during model training. It was found that with randomly initialized weights, the learned filters will exhibit filters which are more tightly spaced at the lower end of the spectrum. These filters however are not being receptive to a single frequency band only but also to its harmonics. This might give the network better robustness at identifying voiced speech when part of the spectrum is masked by noise. As a result, this network architecture gave better separation performance than both approaches that used separate preprocessing for feature extraction.

5.2 Future Work

The availability of virtually unlimited training data by mixing all possible combinations of speech and noise recordings from even a small corpus is both a blessing and a curse. The potentially available training data does not fit on disk, let alone into main memory. Therefore for this thesis the space of training data had to be heavily sampled for model training. An alternative approach where training data is generated on the fly should be investigated. The main challenge here is to ensure both good randomization of the training data to ensure fast convergence [17] and—depending on the complexity of the input features—the ability to generate data fast enough to ensure constant GPU utilization.

It was found that models trained on gammatone-weighted spectrogram features did not adapt well to signal-to-noise ratios not encountered in training, but otherwise delivered acceptable performance at a much lower computational cost in pre- and post-processing. Further experiments are necessary to see if performance for networks trained on these features could be improved by including examples of varying SNR in the training data.

The STOI metric is immensely useful for quickly comparing the (expected) intelligibility of different approaches for speech separation. Since it restricts its analysis to just 15 critical bands from 150 Hz to 4.3 kHz, a system's ability to remove background noise in frequencies outside of these bands is not reflected in the score but nevertheless important for the subjective quality of the separated speech. Defining a new metric that captures the subjective quality of the separated signal might therefore be a worthwhile endeavor.

While the masking approach to speech separation is intuitive and can be nicely

visualized, the prediction of a mask and its subsequent application are just an interim stage to obtaining the separated speech signal. An easy follow-up experiment would be to directly predict the T-F representation of the clean speech signal, forgoing the masking step completely. An additional minor change to the existing architecture would be to use a modified objective function for training. The mean-squared-error between prediction and target T-F units could be weighted based on the unit's impact on intelligibility. It is likely that giving more importance to units corresponding to the critical bands of the STOI metric will give better intelligibility scores (at the risk of a worse subjective hearing impression).

New network architectures should also be examined. The introduction of a convolutional layer for feature extracting from raw waveforms significantly improved the accuracy of the predicted masks. It is likely that additional convolutional layers would be beneficial as they could learn higher-level, time and pitch-independent features present in the input signal. This ultimately might lead to networks which are better at generalization to unseen data.

Appendix A

Results

A.1 Evaluation results

Speech	Noise	Model	SNR [dB]	STOI	Delta [%]
German	**Car?**	**Cochleagram**	**-10.0**	0.70	-6.55
			-2.0	0.87	-2.84
			0.0	0.90	-1.83
		Cochleagram Oracle	**-10.0**	0.92	+22.30
			-2.0	0.97	+7.83
			0.0	0.97	+5.88
		Gammatone	**-10.0**	0.74	-2.25
			-2.0	0.87	-2.74
			0.0	0.88	-4.61
		Gammatone Oracle	**-10.0**	0.91	+20.99
			-2.0	0.95	+5.97
			0.0	0.95	+3.80
		Raw (impulse initialization)	**-10.0**	0.75	-1.08
			-2.0	0.87	-3.28
			0.0	0.88	-4.29
		Raw (random initialization)	**-10.0**	0.68	-9.62
			-2.0	0.84	-6.15
			0.0	0.87	-5.67
		Unprocessed	**-10.0**	0.75	
			-2.0	0.90	
			0.0	0.92	
	Interference	**Cochleagram**	**-10.0**	0.66	-2.51
			-2.0	0.85	+1.42
			0.0	0.88	+1.53
		Cochleagram Oracle	**-10.0**	0.89	+31.39
			-2.0	0.93	+11.64

Continued on next page

Speech	Noise	Model	SNR [dB]	STOI	Delta [%]
			0.0	0.94	+8.76
		Gammatone	**-10.0**	0.53	-21.85
			-2.0	0.82	-2.31
			0.0	0.84	-3.21
		Gammatone Oracle	**-10.0**	0.88	+30.62
			-2.0	0.93	+11.61
			0.0	0.94	+8.73
		Raw (impulse initialization)	**-10.0**	0.43	-36.71
			-2.0	0.80	-4.03
			0.0	0.85	-1.33
		Raw (random initialization)	**-10.0**	0.47	-30.83
			-2.0	0.78	-7.10
			0.0	0.82	-4.91
		Unprocessed	**-10.0**	0.68	
			-2.0	0.84	
			0.0	0.87	
	Siren	**Cochleagram**	**-10.0**	0.38	-19.13
			-2.0	0.54	-12.80
			0.0	0.57	-13.51
		Cochleagram Oracle	**-10.0**	0.82	+74.42
			-2.0	0.88	+41.94
			0.0	0.89	+35.40
		Gammatone	**-10.0**	0.29	-39.26
			-2.0	0.58	-6.50
			0.0	0.62	-6.11
		Gammatone Oracle	**-10.0**	0.84	+78.53
			-2.0	0.90	+46.42
			0.0	0.91	+39.24
		Raw (impulse initialization)	**-10.0**	0.36	-23.08
			-2.0	0.59	-3.78
			0.0	0.64	-3.07
		Raw (random initialization)	**-10.0**	0.42	-10.75
			-2.0	0.59	-5.08
			0.0	0.62	-5.81
		Unprocessed	**-10.0**	0.47	
			-2.0	0.62	
			0.0	0.66	
	Speech Babble	**Cochleagram**	**-10.0**	0.28	-16.12
			-2.0	0.47	-4.60
			0.0	0.51	-5.77
		Cochleagram Oracle	**-10.0**	0.80	+137.25
			-2.0	0.85	+72.27
			0.0	0.87	+59.77
		Gammatone	**-10.0**	0.29	-14.55

Continued on next page

Speech	Noise	Model	SNR [dB]	STOI	Delta [%]
			-2.0	0.54	+8.21
			0.0	0.57	+4.84
		Gammatone Oracle	**-10.0**	0.83	+145.59
			-2.0	0.89	+79.36
			0.0	0.90	+65.91
		Raw (impulse initialization)	**-10.0**	0.31	-9.27
			-2.0	0.53	+7.31
			0.0	0.58	+7.28
		Raw (random initialization)	**-10.0**	0.25	-27.35
			-2.0	0.44	-11.43
			0.0	0.50	-8.13
		Unprocessed	**-10.0**	0.34	
			-2.0	0.50	
			0.0	0.54	
Machine	**Car?**	**Cochleagram**	**-10.0**	0.93	+0.15
			-2.0	0.96	-0.80
			0.0	0.96	-1.25
		Cochleagram Oracle	**-10.0**	0.96	+3.75
			-2.0	0.98	+1.13
			0.0	0.98	+0.84
		Gammatone	**-10.0**	0.89	-3.74
			-2.0	0.95	-2.11
			0.0	0.94	-3.44
		Gammatone Oracle	**-10.0**	0.97	+4.55
			-2.0	0.98	+0.59
			0.0	0.98	-0.02
		Raw (impulse initialization)	**-10.0**	0.91	-2.13
			-2.0	0.94	-2.78
			0.0	0.94	-3.76
		Raw (random initialization)	**-10.0**	0.88	-5.44
			-2.0	0.95	-2.57
			0.0	0.95	-2.38
		Unprocessed	**-10.0**	0.93	
			-2.0	0.97	
			0.0	0.98	
	Interference	**Cochleagram**	**-10.0**	0.85	+5.23
			-2.0	0.94	+4.31
			0.0	0.95	+3.01
		Cochleagram Oracle	**-10.0**	0.94	+15.84
			-2.0	0.96	+6.47
			0.0	0.97	+5.13
		Gammatone	**-10.0**	0.85	+4.27
			-2.0	0.94	+3.34
			0.0	0.94	+2.38

Continued on next page

Speech	Noise	Model	SNR [dB]	STOI	Delta [%]
		Gammatone Oracle	**-10.0**	0.95	+17.30
			-2.0	0.97	+7.23
			0.0	0.97	+5.78
		Raw (impulse initialization)	**-10.0**	0.72	-11.04
			-2.0	0.94	+3.71
			0.0	0.95	+3.13
		Raw (random initialization)	**-10.0**	0.78	-3.33
			-2.0	0.93	+2.83
			0.0	0.95	+2.55
		Unprocessed	**-10.0**	0.81	
			-2.0	0.91	
			0.0	0.92	
	Siren	**Cochleagram**	**-10.0**	0.77	+11.04
			-2.0	0.88	+5.71
			0.0	0.88	+2.77
		Cochleagram Oracle	**-10.0**	0.92	+32.36
			-2.0	0.96	+16.13
			0.0	0.97	+13.14
		Gammatone	**-10.0**	0.38	-45.08
			-2.0	0.87	+4.50
			0.0	0.87	+2.18
		Gammatone Oracle	**-10.0**	0.93	+33.41
			-2.0	0.96	+15.77
			0.0	0.96	+12.68
		Raw (impulse initialization)	**-10.0**	0.72	+3.35
			-2.0	0.87	+5.25
			0.0	0.88	+2.77
		Raw (random initialization)	**-10.0**	0.76	+8.51
			-2.0	0.85	+2.93
			0.0	0.86	+0.96
		Unprocessed	**-10.0**	0.70	
			-2.0	0.83	
			0.0	0.86	
	Speech Babble	**Cochleagram**	**-10.0**	0.51	-11.47
			-2.0	0.81	+7.91
			0.0	0.84	+6.33
		Cochleagram Oracle	**-10.0**	0.91	+58.23
			-2.0	0.95	+26.01
			0.0	0.95	+20.54
		Gammatone	**-10.0**	0.58	+0.77
			-2.0	0.83	+10.56
			0.0	0.87	+10.06
		Gammatone Oracle	**-10.0**	0.90	+56.03
			-2.0	0.94	+25.65

Continued on next page

Speech	Noise	Model	SNR [dB]	STOI	Delta [%]
			0.0	0.95	+20.32
		Raw (impulse initialization)	**-10.0**	0.67	+16.38
			-2.0	0.84	+11.70
			0.0	0.86	+8.16
		Raw (random initialization)	**-10.0**	0.63	+8.72
			-2.0	0.83	+9.82
			0.0	0.84	+5.86
		Unprocessed	**-10.0**	0.58	
			-2.0	0.75	
			0.0	0.79	

Table A.1: Evalation results on two representative speech examples

List of Figures

List of Tables

Bibliography

[1] S. Boll. "Suppression of acoustic noise in speech using spectral subtraction". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.2 (Apr. 1979), pp. 113–120. ISSN: 0096-3518. DOI: 10.1109/TASSP.1979.1163209.

[2] Jitong Chen et al. "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises". In: *The Journal of the Acoustical Society of America* 139.5 (2016), pp. 2604–2612.

[3] James W Cooley and John W Tukey. "An algorithm for the machine calculation of complex Fourier series". In: *Mathematics of computation* 19.90 (1965), pp. 297–301.

[4] Irving B Crandall and Donald MacKenzie. "Analysis of the energy distribution in speech". In: *Bell Labs Technical Journal* 1.1 (1922), pp. 116–128.

[5] D.P.W. Ellis. *Gammatone-like spectrograms*. Last visited 2017/8/28. 2009. URL: http://www.ee.columbia.edu/~dpwe/resources/matlab/gammatonegram/.

[6] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[7] Awni Y. Hannun et al. "Deep Speech: Scaling up end-to-end speech recognition". In: *CoRR* abs/1412.5567 (2014). URL: http://arxiv.org/abs/1412.5567.

[8] Geoffrey E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). URL: http://arxiv.org/abs/1207.0580.

[9] John Holdsworth et al. "Implementing a gammatone filter bank". In: *Annex C of the SVOS Final Report: Part A: The Auditory Filterbank* 1 (1988), pp. 1–5.

[10] Po-Sen Huang et al. "Joint optimization of masks and deep recurrent neural networks for monaural source separation". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.12 (2015), pp. 2136–2147.

[11] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0130226165.

[12] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed 6/21/2017]. 2001–. URL: http://www.scipy.org/.

[13] Jean-Claude Junqua. "The influence of acoustics on speech production: A noise-induced stress phenomenon known as the Lombard reflex". In: *Speech Communication* 20.1 (1996). Speech under Stress, pp. 13–22. ISSN: 0167-6393. DOI: http://dx.doi.org/10.1016/S0167-6393(96)00041-6. URL: http://www.sciencedirect.com/science/article/pii/S0167639396000416.

[14] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: http://arxiv.org/abs/1412.6980.

[15] N. Kitawaki and K. Itoh. "Pure delay effects on speech quality in telecommunications". In: *IEEE Journal on Selected Areas in Communications* 9.4 (May 1991), pp. 586–593. ISSN: 0733-8716. DOI: 10.1109/49.81952.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[17] Yann A LeCun et al. "Efficient backprop". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[18] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. "Acoustic modeling using deep belief networks". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1 (2012), pp. 14–22.

[19] Deborah Imel Nelson et al. "The global burden of occupational noise-induced hearing loss". In: *American journal of industrial medicine* 48.6 (2005), pp. 446–458.

[20] World Health Organization. *Deafness and hearing loss. Fact sheet*. Online; accessed 9/5/2017. Feb. 2017. URL: http://www.who.int/mediacentre/factsheets/fs300/en/.

[21] *PyTorch*. Aug. 2017. URL: http://pytorch.org/.

[22] Tara N Sainath et al. “Learning the speech front-end with raw waveform CLDNNs”. In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[23] M. Slaney, D. Naar, and R. E. Lyon. “Auditory model inversion for sound separation”. In: *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*. Vol. ii. Apr. 1994, II/77–II/80 vol.2. DOI: 10.1109/ICASSP.1994.389714.

[24] Malcolm Slaney et al. “An efficient implementation of the Patterson-Holdsworth auditory filter bank”. In: (1993).

[25] David Snyder, Guoguo Chen, and Daniel Povey. *MUSAN: A Music, Speech, and Noise Corpus*. arXiv:1510.08484v1. 2015. eprint: 1510.08484.

[26] Cees H Taal et al. “An algorithm for intelligibility prediction of time–frequency weighted noisy speech”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), pp. 2125–2136.

[27] Shin’ichi Tamura and Alex Waibel. “Noise reduction using connectionist models”. In: *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE. 1988, pp. 553–556.

[28] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006. ISBN: 0471741094.

[29] P. Welch. “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (June 1967), pp. 70–73. ISSN: 0018-9278. DOI: 10.1109/TAU.1967.1161901.

[30] BP Welford. “Note on a method for calculating corrected sums of squares and products”. In: *Technometrics* 4.3 (1962), pp. 419–420.