uc3m | Universidad Carlos III de Madrid

BIO-MOTIVATED FEATURES AND DEEP LEARNING FOR ROBUST SPEECH RECOGNITION

Presented by:

FERNANDO DE LA CALLE SILOS

Supervised by:

DR. ASCENSIÓN GALLARDO ANTOLÍN

DR. CARMEN PELÁEZ MORENO

DEPARTMENT OF SIGNAL THEORY AND COMMUNICATIONS

Leganés, September, 2017

Tesis doctoral:
BIO-MOTIVATED FEATURES AND DEEP LEARNING FOR ROBUST SPEECH RECOGNITION

Autor:
FERNANDO DE LA CALLE SILOS

Directores:
DRA. ASCENSIÓN GALLARDO ANTOLÍN
DRA. CARMEN PELÁEZ MORENO

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores:

Presidente:

DR. JAVIER FERREIROS LÓPEZ

Secretario:

DR. FERNANDO DÍAZ DE MARÍA

Vocal:

DR. RUBÉN SOLERA UREÑA

acuerda otorgarle la calificación de:

Leganés, a 29 de Septiembre de 2017

ABSTRACT

In spite of the enormous leap forward that the Automatic Speech Recognition (ASR) technologies has experienced over the last five years their performance under hard environmental condition is still far from that of humans preventing their adoption in several real applications. In this thesis the challenge of robustness of modern automatic speech recognition systems is addressed following two main research lines.

The first one focuses on modeling the human auditory system to improve the robustness of the feature extraction stage yielding to novel auditory motivated features. Two main contributions are produced. On the one hand, a model of the masking behaviour of the Human Auditory System (HAS) is introduced, based on the non-linear filtering of a speech spectro-temporal representation applied simultaneously to both frequency and time domains. This filtering is accomplished by using image processing techniques, in particular mathematical morphology operations with an specifically designed Structuring Element (SE) that closely resembles the masking phenomena that take place in the cochlea. On the other hand, the temporal patterns of auditory-nerve firings are modeled. Most conventional acoustic features are based on short-time energy per frequency band discarding the information contained in the temporal patterns. Our contribution is the design of several types of feature extraction schemes based on the synchrony effect of auditory-nerve activity, showing that the modeling of this effect can indeed improve speech recognition accuracy in the presence of additive noise. Both models are further integrated into the well known Power Normalized Cepstral Coefficients (PNCC).

The second research line addresses the problem of robustness in noisy environments by means of the use of Deep Neural Networks (DNNs)-based acoustic modeling and, in particular, of Convolutional Neural Networks (CNNs) architectures. A deep residual network scheme is proposed and adapted for our purposes, allowing Residual Networks (ResNets), originally intended for image processing tasks, to be used in speech recognition where the network input is small in comparison with usual image dimensions. We have observed that ResNets on their own already enhance the robustness of the whole system against noisy conditions. Moreover, our experiments demonstrate that their combination with the auditory motivated features devised in this thesis provide significant improvements in recognition accuracy in comparison to other state-of-the-art CNN-based ASR systems under mismatched conditions, while maintaining the performance in matched scenarios.

The proposed methods have been thoroughly tested and compared with other state-of-the-art proposals for a variety of datasets and conditions. The obtained results prove that our methods outperform other state-of-the-art approaches and reveal that they are suitable for practical applications, specially where the operating conditions are unknown.

RESUMEN

A pesar del enorme impulso que las tecnologías de reconocimiento del habla han experimentado durante los últimos cinco años, su aplicación en condiciones adversas como, por ejemplo, en presencia de alto ruido, dista bastante de la capacidad de reconocimiento que tenemos los humanos. Esto ocasiona a menudo que su implantación práctica no se pueda llevar a cabo. En esta Tesis se aborda el desafío del reconocimiento de habla robusto desde dos perspectivas.

La primera se centra en modelar el sistema auditivo humano para mejorar la robustez del proceso de extracción de características. Se han concebido dos contribuciones principales. Por una lado, modelamos el fenómeno de enmascaramiento del sistema auditivo humano utilizando para ello un filtrado no lineal del espectro que se aplica simultáneamente en los dominos del tiempo y la frecuencia. En concreto e inspirándonos en técnicas de procesado de imagen, utilizamos operaciones de morfología matemática con un elemento estructurante específicamente diseñado para emular los fenómenos de enmascaramiento que se producen en la cóclea. Por otra parte, hemos modelado los patrones temporales de los impulsos nerviosos que se transmiten a través del nervio auditivo. La mayoría de las características acústicas convencionales se basan en el cálculo de la energía por banda de frecuencia durante un periodo corto de tiempo, descartando la información temporal contenida en estos patrones. Nuestra contribución consiste en el diseño de diversos esquemas de extracción de características capaces de sacar partido de dichos patrones a través del efecto de sincronía que se produce en el nervio auditivo. Con ello demostramos que el modelado de este efecto puede mejorar la precisión del reconocimiento de habla en presencia de ruido aditivo. Ambas contribuciones se integraron en el conocido esquema de los llamados Power Normalized Cepstral Coefficients (PNCC) (Coeficientes Cepstrales Normalizados en Potencia).

La segunda línea de investigación abunda en el tema de la mejora de la robustez mediante técnicas de aprendizaje profundo y en particular, utilizando redes neuronales convolucionales (Convolutional Neural Networks (CNNs)). Nuestra propuesta consiste en la adaptación de las conocidas como Residual Networks (ResNets) o redes residuales, introducidas inicialmente en el ámbito de procesado de imagen, para su uso en reconocimiento del habla, donde la dimensión de entrada es menor, en comparación con las dimensiones habituales empleadas en aplicaciones de visión artificial. Hemos comprobado que las ResNets por sí solas ya aumentan la robustez de todo el sistema frente a condiciones adversas pero además, nuestros experimentos demuestran

que su combinación con las características propuestas en esta tesis proporciona mejoras significativas comparadas con otras CNNs del estado del arte. Esta ventaja aparece cuando las condiciones de los conjuntos de entrenamiento y test no coinciden (mismatch conditions o condiciones desajustadas), manteniendo al mismo tiempo el rendimiento en conjuntos con similares condiciones (matched conditions o condiciones ajustadas).

Los métodos propuestos han sido ampliamente probados y comparados con otros del estado del arte, con una amplia variedad de bases de datos y condiciones. Los resultados obtenidos demuestran que nuestros métodos mejoran otras aproximaciones y resultan especialmente indicados en aplicaciones prácticas donde se desconocen a priori las condiciones de operación.

PUBLICATIONS

The following publications are included in parts or in an extended version in this thesis:

- F. de-la-Calle-Silos and R. M. Stern, "Synchrony-Based Feature Extraction for Robust Automatic Speech Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1158–1162, Aug. 2017.

- F. de-la-Calle-Silos, F. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "Morphologically Filtered Power - Normalized Cochleograms as Robust, Biologically Inspired Features for ASR," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 2070–2080, Nov. 2015.

- F. de-la-Calle-Silos, A. Gallardo-Antolín, and C. Peláez-Moreno, "Deep Maxout Networks Applied to Noise-Robust Speech Recognition," in *Advances in Speech and Language Technologies for Iberian Languages*, ser. Lecture Notes in Computer Science, vol. 8854, Springer International Publishing, 2014, pp. 109–118.

- F. de-la-Calle-Silos, F. J. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "ASR Feature Extraction with Morphologically - Filtered Power - Normalized Cochleograms," in *Proceedings of Interspeech (Annual Conference of the International Speech Communication Association)*, 2014, pp. 2430–2434.

Furthermore, the following publications were part of my PhD research, are however not covered in this thesis. The topics of these publications are outside of the scope of the material covered here:

- F. de-la-Calle-Silos, A. Gallardo-Antolín, and C. Peláez-Moreno, "An Analysis of Deep Neural Networks in Broad Phonetic Classes for Noisy Speech Recognition," in *Advances in Speech and Language Technologies for Iberian Languages: Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016, Proceedings*. Springer International Publishing, 2016, pp. 87–96.

- F. de-la-Calle-Silos, F. J. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "Preliminary experiments on the robustness of biologically motivated features for DNN-based ASR," in *Bioinspired Intelligence (IWOBI), 2015 4th International Work Conference on*, Jun. 2015, pp. 169–176.

ACKNOWLEDGMENTS

Con este documento termina una etapa que empezó hace casi diez años, cuando llegué a Madrid. Durante este tiempo muchas personas se han cruzado en mi camino, haciéndolo más interesante y fácil de recorrer.

En primer lugar, me gustaría dar las gracias a Fernando Díaz, por la confianza depositada en mí, dándome la posibilidad de incorporarme al GPM y realizar esta tesis. También quiero agradecer a mis tutoras, Carmen y Ascen, por todas sus aportaciones y dedicación durante estos cuatro años.

Al grupo de doctorandos de Airbus y los responsables del proyecto por hacer posible SAVIER, y a todos los compañeros que forman el Departamento de Teoría de la Señal.

A toda la gente del GPM, por hacer el día a día mucho más ameno, sois todos unos grandes: Antonio, Miguel Ángel, Javier, Chelus, Tomás entre otros. Especialmente a mi gran amiga Amaya, compañera en las alegrías y frustraciones que acontecen en un doctorado, nuestras míticas conversaciones en el zulo nunca se olvidarán. A Rubén, mi único compañero de reconocimiento del habla, por todos los ánimos dados para poder terminar esta tesis. Iván por introducirme en el mundo de la investigación durante mi trabajo de fin de grado. Al gran Eduardo Martínez por todas sus enseñanzas y ánimos, nuestro viaje a Australia será inolvidable, y al señor Azpicueta por enseñarme las peculiaridades de la enseñanza.

A Richard Stern, muchas gracias por la experiencia de trabajar con uno de los mejores y por poder pasar una temporada muy especial, en una de las mejores universidades. Fue un periodo muy enriquecedor para mí. No puedo dejar de acordarme de mi familia adoptiva de Pittsburgh; Jose, Sarah y toda su familia. Sin vosotros una parte de esta tesis no hubiera sido posible, muchas gracias por acogerme y tratarme como a uno más, nunca lo olvidaré.

A Adam por ayudarme con en inglés durante estos últimos años y convertirse en un gran amigo. A David y Aida, por todo el apoyo con mi tesis y los buenos momentos que hemos pasado juntos en el proceso.

A mi familia, en especial a mis padres por todo el esfuerzo que han hecho para que llegase hasta aquí. A mi hermano y a Elena, a Paula y Carlos que tanto hacen reír a su tito. A mi abuelo Pepe, que a sus más de 90 años todavía se interesa a diario por cuando termino mi tesis. A mi abuela Pepa y al abuelo Diego, que aunque ya no estén estoy seguro de que se sentirían orgullosos. A mis tíos Blanca y Jesús por su apoyo incondicional. Al resto de mis tíos, tías, primos y primas.

A Montse, Jaime y Romina, que durante estos años me han querido y cuidado, como a un hijo y a un hermano. A toda la familia asturiana, por acogerme como uno más.

Por último a Mon por que eres la persona de mis momentos más felices, te quiero.

Gracias a todos.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACC	Accuracy
ADC	Analog to Digital Converter
AGC	Automatic Gain Control
ALSD	Average Localized Synchrony Detection
ALSR	Average Localized Synchrony Rate
AMFB	Amplitude Modulation Filter Bank
AMI	Augmented Multi-party Interaction
ANN	Artificial Neural Network
ANN-HMM	Artificial Neural Network-Hidden Markov Model
ANS	Asymmetric Noise Suppression
ASR	Automatic Speech Recognition
BN	Batch Normalization
CD-DNN-HMM	Context Dependent-Deep Neural Network-Hidden Markov Model
CE	Cross Entropy
CF	Characteristic Frequency
CFG	Context-Free Grammar
CMU	Carnegie Mellon University
CMVN	Cepstral Mean and Variance Normalization
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DARPA	Defense Advanced Research Projects Agency
DCT	Discrete Cosine Transform
DMN	Deep Maxout Network
DMT	Discriminative Mapping Transformation
DNN	Deep Neural Network

DNN-HMM	Deep Neural Network-Hidden Markov Model
DTW	Dynamic Time Warping
EIH	Ensemble Interval Histogram
EM	Expectation Maximization
ERB	Equivalent Rectangular Bandwidth
ETSI	European Telecommunications Standards Institute
FANT	Filter and Noise-adding Tool
FIR	Finite Impulse Response
fMLLR	Feature-Space Maximum Likelihood Linear Regression
GCS	Ground Control Station
GMM	Gaussian Mixture Model
GMM-HMM	Gaussian Mixture Model-Hidden Markov Model
GP-GPU	General Purpose Graphics Processing Unit
GPU	Graphics Processing Unit
GSD	Generalized Synchrony Detector
G2P	Grapheme to Phoneme
HAS	Human Auditory System
HDF	Hidden Dropout Factor
HIST	HIerarchical Spectro-Temporal
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
ICT	Information and Communication Technologies
IFFT	Inverse Fast Fourier Transform
IHC	Inner Hair Cell
IIF	Invariant-Integration Features
ISO	International Organization for Standardization
ITD	Interaural Time Difference
ITU	International Telecommunication Union
KLD	Kullback–Leibler Divergence

LDA	Linear Discriminant Analysis
LNCC	Locally Normalized Cepstral Coefficients
LNFB	Locally-Normalized Filter-banks
LP	Linear Prediction
LTU	Linear Threshold Unit
LSTM	Long Short Term Memory networks
LVCSR	Large Vocabulary Continuous Speech Recognition
MALSR	Modified Average Localized Synchrony Rate
MF	Morphological Filtering
MF-PNFB	Morphological Filtering Power Normalized Filter Banks
MelFB	Mel Filter Bank log-energies
MFCC	Mel-Frequency Cepstral Coefficients
MFSC	Mel-Frequency Spectral Coefficients
MGSD	Modified Generalized Synchrony Detector
MGSD-MF-PNCC-NR	Modified Generalized Synchrony Detector with Morphological Filtering Power-Normalized Cepstral Coefficients Noise Reduction
MGSD-PNCC-NR	Modified Generalized Synchrony Detector with Power-Normalized Cepstral Coefficients Noise Reduction
MGSD-SS-NR	Modified Generalized Synchrony Detector with Spectral Subtraction Noise Reduction
MLLT	Maximum Likelihood Linear Transform
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
MMI	Maximum Mutual Information
MMSE	Minimum Mean Square Error
MR	Mean Rate
MVN	Mean and Variance Normalization
NLL	Negative Log-Likelihood
OLA	OverLap Add
PCA	Principal Component Analysis

PDNN	Python Toolkit for Deep Neural Networks
PER	Phone Error Rate
PLP	Perceptually-based Linear Prediction
PMC	Parallel Model Combination
PMVDR	Perceptual Minimum Variance Distortionless Response
PNCC	Power Normalized Cepstral Coefficients
PNFB	Power Normalized Filter Banks
RASTA	Relative SpecTrAl processing
RBM	Restricted Boltzmann Machine
RIR	Room Impulse Response Generator
ReLU	Rectified Linear Unit
RM	Resource Management
ResNet	Residual Network
RNN	Recurrent Neural Networks
ROVER	Recognition Output Voting Error Reduction
SAT	Speaker Adaptative Training
SAVIER	Situational Awareness Virtual Environment
SDA	Stacked Denoising Autoencoder
SE	Structuring Element
SGD	Stochastic gradient descent
SNR	Signal-to-Noise Ratio
SPARK	SParse Auditory Reproducing Kernel
SPL	Sound Pressure Level
SS	Spectral Subtraction
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SYDOCC	Synchronous Damped Oscillator Cepstral Coefficients
TIMIT	Texas Instruments and Massachusetts Institute of Technology

UAV	Unmanned Aerial Vehicle
VAD	Voice Activity Detector
VGGNet	Visual Geometry Group Network
VTS	Vector Taylor Series
WER	Word Error Rate
WSJ	Wall Street Journal
WSJ0	Wall Street Journal 0
ZCPA	Zero Crossing Peak Amplitude
3D	three dimensional
2D	two dimensional

1 INTRODUCTION

1.1 MOTIVATION

Speech is the natural way of communication between humans. From the beginning of humankind, communication has evolved from signs to a complex spoken languages, due to the necessity of transmitting more elaborated messages. Nowadays, with the arrival of Information and Communication Technologiess (ICTs), a monumental change in how we interact with machines is taking place. We would like to interact with devices as if they were humans.

An important role in this process is played by Automatic Speech Recognition (ASR) systems that allow to substitute the non-natural traditional interfaces like keyboards, mouses or screens, with the most frequent form of communication used among humans: speech.

Automatic speech recognition has a long history of research, failures and successes. Its development during the years resembles the process that humans go over when they learn to talk and understand speech, from learning single syllables to figure out the meanings of thousands of words in complex sentences. Since 1952, were researchers at Bell labs designed a machine able to understand digits [7], the development of ASR has not stopped.

Two major turning points have happened during this history. The first one was the shift from template matching based approaches to statistical modeling, in particular, the use of Hidden Markov Model (HMM) in the early 80s [8, 9] increased the accuracy of recognizers and the vocabulary size. These advances made it possible to develop the first commercially available products able to perform continuous speech recognition; but its performance were far from that of humans.

The second turning point was the breakthrough in performance achieved by using deep learning techniques [10]. In 2010 ASR systems were the first mayor industrial application of deep learning [11]. Nowadays almost all the speech recognitions systems are based on deep learning algorithms.

As can be seen ASR is a fairly developed technology, and in particular this second turning point has provided the possibility that real life speech recognition applications become mainstream. All the major smartphone and computer operating systems have its own ASR engine integrated. It has become almost mandatory in car infotainment systems and other industrial applications have adopted the technology as call-centers or courier services.

Many challenges arise in real-life applications where ASR systems are exposed to adverse conditions as environmental noise or reverberation. These scenarios degrade the performance drastically diminishing the feasibility of their deployment in several environments such as Unmanned Aerial Vehicle (UAV) remote commanding where the security is critical.

As we will see in the next chapters, the research in the robustness of ASR systems is a very broad field; but the performance of ASR systems in adverse conditions, in particular when those conditions are unknown, is still an open research question, becoming one of the major challenges for real applications.

The Human Auditory System (HAS) has evolved during thousands of years to recognize speech, in such a way that it has a remarkable ability to understand speech in hard conditions. Mimicking its most remarkable features in a realistic manner is a sound approach to the aforementioned question.

1.2 SOCIO-ECONOMIC FRAMEWORK

The world's largest technology companies are pushing out voice recognition technology to customers in their devices and mobile operating systems, Apple has Siri, Amazon designed Alexa, Google created Google Assist, Microsoft developed Cortana and Facebook includes Oculus Voice as a speech assistant in the Oculus headset. The interest in bringing speech recognizers into other fields is growing as many companies include it in their products to meet the demands of their customers.

Following the line of companies interested in speech technologies, the advances reached in this thesis in the area of robust speech recognition are developed under the framework of an Airbus project called Situational Awareness Virtual Environment (SAVIER) where new human to machine technologies are developed for the future Ground Control Station (GCS). Almost all the proposed methods are tested in a research demonstrator, where an ASR system has been implemented to control and command an UAV in presence of adverse noise conditions.

1.3 OBJECTIVES

The objective of this thesis is to address the problem of the robustness of modern automatic speech recognition systems, two main research lines are followed. On the one hand, we propose novel biologically inspired feature extraction stages based on the modeling of the HAS and, specifically, of the masking phenomenon and the synchrony effect. On the other hand, novel deep learning techniques are employed together with our auditory motivated features. All our efforts are focused on enhancing the accuracy of the recognizer in unknown conditions; although the case where the conditions affecting the speech are known has also been addressed.

Specifically, our main approaches are:

- To mimic the human auditory system realistically in order to improve the performance of ASR systems in noisy and hard conditions, proposing bio-inspired features. In this direction the main approaches are:
    - To model the masking behavior of the HAS enhancing the robustness of the feature extraction stage in ASR, using image processing techniques. The design of a morphological filter that capture the masking effect is the main objective.
    - To model the auditory nerve synchrony effect to improve the speech recognition accuracy for speech that is degraded by noise, and reverberation.
    - To integrate both models into the well known Power Normalized Cepstral Coefficients (PNCC) [12] feature scheme.
- To apply novel deep learning techniques that improve the robustness, in particular Convolutional Neural Network (CNN) as Residual Networks (ResNets) [13].
- To use auditory motivated robust input features in combination with Deep Neural Network (DNN) architectures, to achieve significant improvements with respect to conventional features when the mismatch between train and test data is high.

1.4 THESIS OUTLINE

The material presented in this thesis is organized as follows:

- *Chapter* 2 introduces the fundamentals of ASR, and defines the problem of robustness and how it can be addressed.
- *Chapter* 3 describes the HAS and how robust features can be motivated by it. The most common auditory inspired features are reviewed in detail as our following contributions are laid upon them.
- *Chapter* 4 presents our contribution on the modeling of the cochlear masking phenomenon of the HAS and the design of a robust front-end based on this model using image processing techniques.
- *Chapter* 5 answers the questions of what the synchrony effect is in the HAS and how it can be integrated in the front-end of an ASR system.

- *Chapter 6* provides an overview of deep learning techniques employed in modern ASR systems.
- *Chapter 7* describes our application of novel deep learning architectures for robust speech recognition based on CNNs and its combination with the previously presented feature extraction techniques.
- *Chapter 8* draws some conclusions and further lines of research.
- *Appendix A* contains the introduction and conclusions translated into Spanish.

2 OVERVIEW OF AUTOMATIC SPEECH RECOGNITION

2.1 INTRODUCTION

The goal of an Automatic Speech Recognition (ASR) system is to accurately and efficiently convert a speech signal into a text transcription of the spoken words. This chapter focuses on how traditional systems work, the recent changes with the advent of deep learning tools and how the problem of robustness is addressed.

Automatic Speech Recognition nowadays is highly used for human to human and human to machine interaction. It has been a very active research area ever since the first systems appeared but in recent years the field has experienced a drastic change with the advent of deep learning frameworks.

The idea of using a neural network in an ASR system is not novel [14], but the progress made in the field of machine learning that allows us to train neural networks with a higher number of layers grants the ASR systems substantial improvements in the recognition rates making it practical in every day life interaction of humans and machines.

The following sections introduce the different building blocks of an ASR architecture with a special emphasis on the ones concerning this thesis, in particular the feature extraction stage, where the speech signal is represented as a sequence of feature vectors aiming at obtaining an invariant representation of the speech waveform, and the acoustic modeling stage where the properties of the features are modeled.

As stated in the previous chapter the main goal of this thesis is to address the problem of robustness in current ASR systems, in particular when the speech signal is influenced by external factors. An introduction on how this problem has traditionally been solved and what these factors are, is also presented in this chapter.

A more detailed explanation of the robust feature extraction step is available in Chapter 3 in order to lay the foundations for our proposed robust features. Also an introduction to deep learning for ASR is presented in Chapter 6 to obtain some insights to understand the contributions in the field of the acoustic modeling stage. A more detailed review of the field can be found in numerous tutorials and books available as for example [15] or [16].

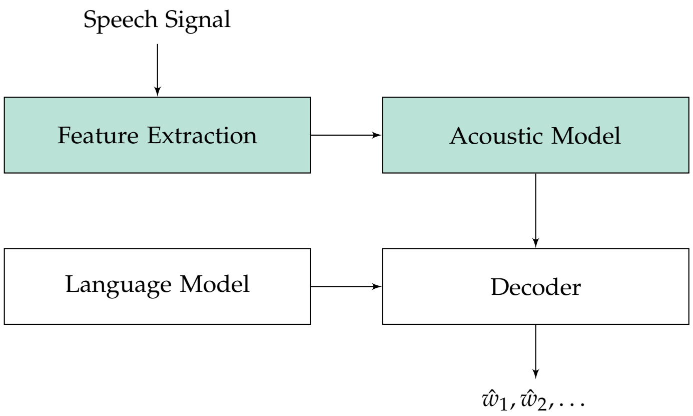


Figure 2.1: Basic architecture of an ASR system. Shaded blocks indicate where the main contributions of the thesis are circumscribed.

2.2 BASIC AUTOMATIC SPEECH RECOGNITION MODEL

The first commercially available ASR systems were based on dynamic programming and pattern matching algorithms as Dynamic Time Warping (DTW) [17] where the input is compared with predefined templates using dynamic programming for alignment which allows the comparison of inputs and templates of different lengths.

In the 80s the ASR systems started to be applied to more complex tasks, letting go the pattern matching approach in favor of statistical modeling techniques. The fundamental technique employed was the Hidden Markov Model (HMM) [18]. HMMs model speech as if it were generated by a process that goes through a series of states following a Markov chain, where each state has an emission probability customarily modeled by a Gaussian Mixture Model (GMM). The transition from one state to another is determined stochastically and is only dependent on the current state. HMMs convert the ASR problem into finding the most probable state sequence given the input signal.

The next step to improving the performance of ASR was to substitute GMMs with a neural network [14], which allowed a more robust estimation of the emission probability. This approach is known as the hybrid model.

In the recent years the introduction of Deep Neural Network (DNN) [10], first using the hybrid model and later by using an end-to-end approach [19], has improved the performance drastically.

In Figure 2.1, we present the main components of an ASR system. As can be seen it is composed of a feature extraction stage (also called front-end), an acoustic model, a language model and an hypothesis search or decoding stage (also called back-end).

The feature extraction stage tries to obtain a robust representation of the speech signal extracting vectors that are meaningful for the task. In many cases, it attempts to remove noise from the speech signal. This process takes place in the frequency domain since phonetic information is mainly contained in the spectrum and its evolution over time, although relevant information could be also present in the time domain, as we will see in Chapter 5.

In the acoustic modeling stage, the information about the sequence of acoustic units (as phonemes) is learned from the feature vectors. The particular problems addressed in this stage are the variable length of the speech utterances and the various sources of variability, which are typically addressed by HMM and robust techniques, respectively.

The language model learns the correlation between the different words of the training corpora and estimates the probability of the different output sequences, typically using Context-Free Grammars (CFGs) or statistic n-grams [20].

Finally, the hypothesis search module or decoding stage combines the output of the acoustic and language models and produces the word sequence with the highest score.

More specifically the feature extraction or parameterization stage computes a sequence of feature vectors: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T]$ of length T given the speech signal. These vectors are usually computed by windowing the speech frames using 20-30 ms analysis windows with a shift of 10-15 ms. The most usual feature extraction algorithm is the Mel-Frequency Cepstral Coefficients (MFCC) [21]. In the next chapter we will present a detailed description of the feature extraction stage as various of our main contributions are related to it.

State-of-the-art systems take a statistical approach to speech recognition, in such a way that given the feature vector sequence $\mathbf{X}$, the spoken word sequence $\mathbf{W} = \hat{w}_1, \hat{w}_2, \ldots$ can be estimated through the maximum a posteriori probability:

$$\hat{w}_1, \hat{w}_2, \cdots = \underset{w_1, w_2, \ldots}{\operatorname{argmax}} P(w_1, w_2, \ldots | \mathbf{X}). \tag{2.1}$$

This is the so-called *speech recognition problem* equation. The above equation determines the sequence that is most probable given the observations feature vector sequence $\mathbf{X}$.

To make the problem amenable to estimation, the Bayes rule can be applied, taking into account that $P(\mathbf{X})$ is independent from the $w_1, w_2, \ldots$, transforming the previous equation to:

$$\hat{w}_1, \hat{w}_2, \cdots = \underset{w_1, w_2, \ldots}{\operatorname{argmax}} P(\mathbf{X} | w_1, w_2, \ldots) P(w_1, w_2, \ldots), \tag{2.2}$$

where $P(\mathbf{X} | w_1, w_2, \ldots)$ models the acoustic properties of the speech and hence is denoted as the acoustic model, and the term $P(w_1, w_2, \ldots)$ gives the a priori distribution of the word sequence. This factor is typically modeled by the language model and is out of the scope of this thesis. Both the acoustic and language models are obtained in a previous training stage of the ASR system.

Finally the decoding stage tries to solve the equation by searching for the word sequence that best matches the input vector sequence. The challenge of ASR systems is to build the acoustic and language models that best reflect the spoken language.

The main contributions of this thesis are focused on the design of novel robust techniques of encoding the information of the speech signal into the feature vectors $\mathbf{X}$ by taking into account the Human Auditory System (HAS) and the incorporation of novel deep learning architectures into the acoustic modeling stage.

2.3 GMM-HMM BASED SPEECH RECOGNITION

This section provides an overview of how the problem of speech recognition has traditionally been modeled using HMMs.

In a Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) based ASR system, the probability distribution $P(\mathbf{X}|w_1, w_2, \dots)$ from equation 2.2 is modeled using as many HMMs as acoustic units considered. Specifically, GMM-HMMs model speech as if it were generated by a process that goes through a series of states following a Markov chain, where each state has an emission probability modeled by a GMM, and the transitions between states are given by the *transition* probabilities.

The number of words in the dictionary or *lexicon* of the system in complex tasks is usually very large making words unsuitable as acoustic units. Therefore, instead of using words, HMMs model smaller acoustic units, as for example phonemes whose inventory has a smaller cardinality. Subsequently, words can be recomposed by concatenating these units.

In current ASR systems, the acoustic units consider not only phonemes but their phonetic context, typically the so-called *triphones*, which considers every combination of previous and posterior phonemes, providing a better representation of the speech due to co-articulation. As the computational cost of using the complete set of triphones can be high, usually some of the HMM parameters to be learned are shared (or *tied*) between acoustically similar triphones.

Each acoustic unit is usually represented by a HMM with a Bakis topology [8], with non-emitting initial and final states and normally three emitting or active states. The Bakis topology does not permit the return to a state once it has transitioned out of it, which is a realistic model of how speech is produced. The first and final emitting states models the transitions between the different triphones and the central state models the stable part. A example of this architecture can be seen in Figure 2.2.

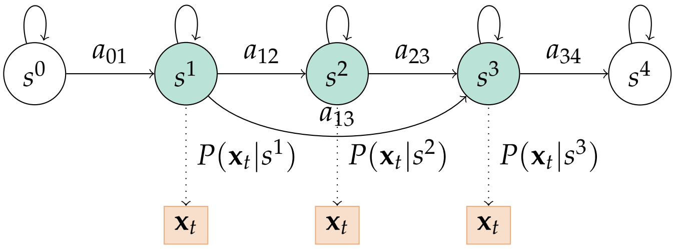


Figure 2.2: An HMM with 5 states, with non-emitting initial and final states and three emitting or active states. Only the transitions permitted in Bakis architecture are drawn. a_{ij} is the probability of transition from state s^i to state s^j, and $P(\mathbf{x}_t|s^i)$ is the emission probability of feature vector $\mathbf{x}_t$ in state s^i. This architecture is commonly employed to model triphone acoustic units.

As mentioned before, the emission probabilities are modeled using GMMs, where the probability of that a particular feature vector $\mathbf{x}_t$ of the sequence having been generated from the state s^i is given by:

$$P(\mathbf{x}_t|s^i) = \sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{x}_t; \mu_{m,i}, \mathbf{\Sigma}_{m,i}), \tag{2.3}$$

where M is the number of components in the mixture, c_m the weight of the m-th component and $\mathcal{N}(\mathbf{x}_t; \mu_{m,i}, \mathbf{\Sigma}_{m,i})$ is a multivariate Gaussian density:

$$\mathcal{N}(\mathbf{x}_t; \mu_{m,i}, \mathbf{\Sigma}_{m,i}) = \frac{1}{(2\pi)^{D/2}|\Sigma_{m,i}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_t - \mu_{m,i})^T \Sigma_{m,i}^{-1}(\mathbf{x}_t - \mu_{m,i})\right\}, \tag{2.4}$$

with mean μ_m, and covariance matrix Σ_m for the m-th component of the mixture.

In order to train the GMM-HMM model, an Expectation Maximization (EM) method is commonly considered, where the parameters of the GMMs and the transition probabilities are estimated through an iterative process using the Baum–Welch algorithm [22]. As most of the datasets are labeled in a per utterance basis, it is not possible to initially assign a feature vector to its corresponding state. Therefore, an iterative estimation process with alternate estimation and alignment procedures is repeated until convergence.

The direct evaluation of equation 2.2 is unfeasible, since we need to compute $P(\mathbf{X}|w_1, w_2, \dots)$ for each possible word sequence. In order to overcome this problem, statistical modeling using a HMM is introduced, modifying 2.2 to:

$$\hat{w}_1, \hat{w}_2, \dots = \underset{w_1, w_2, \dots}{\operatorname{argmax}} \max_{\mathbf{q}} P(\mathbf{X}, \mathbf{q}|w_1, w_2, \dots) P(w_1, w_2, \dots), \tag{2.5}$$

where $\mathbf{q}$ is a state sequence in the HMM for the spoken word sequence. This modification makes the problem feasible as only the locally most probable state sequence is considered.

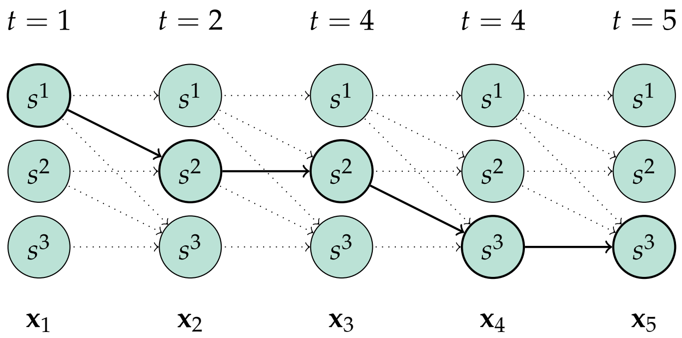


Figure 2.3: Trellis of the observation sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_5]$ for the HMM presented in Figure 2.2. Only the transitions permitted by the Bakis architecture are drawn. The Viterbi algorithm will be run over the Trellis taking into account the transition and emission probabilities finding the most probable path, in this case $\mathbf{q} = \left[s^1, s^2, s^3\right]$.

It is possible to compose an HMM for the entire language by connecting the HMMs of words (made of triphones HMMs) according to the language model.

The final recognition is made by finding the most probable state sequence, which is usually computed using the Viterbi algorithm over a trellis diagram. An example can be seen in Figure 2.3.

2.4 HYBRID MODELS

The GMMs employed in the HMM provide the emission probabilities have low discriminative capabilities and, in addition, they are not invariant to changes in the input.

For these reasons in the early 1990s, first attempts were made to replace GMMs by Artificial Neural Networks (ANNs). Unlike GMMs, ANNs have a high discrimination capability and better generalization properties. Also GMMs assume that the emission probability distributions can be modeled using a mixture of gaussians, whereas ANN are an universal function approximator.

In the classical Artificial Neural Network-Hidden Markov Model (ANN-HMM) hybrids [14], an ANN is trained to classify the input acoustic features into classes corresponding to the states of HMMs, in such a way that the old GMM-based state emission likelihoods are replaced by the likelihoods generated by the ANN. In other words, the neural network estimates the posterior probability $P(s^i|\mathbf{x}_t)$ of each state s^i given the observation $\mathbf{x}_t$ at time t.

In a hybrid ASR system, the HMM topology is set from a previously trained GMM-HMM, and the ANN training label data come from the forced-alignment between the state-level transcripts and the corresponding speech signals obtained by using this initial GMM-HMM system.

In the recognition stage, an estimation of the emission likelihoods of each HMM state, $P(\mathbf{x}_t|s^i)$, is obtained by using the Bayes rule as follows:

$$P(\mathbf{x}_t|s^i) = \frac{P(s^i|\mathbf{x}_t)P(\mathbf{x}_t)}{P(s^i)}, \tag{2.6}$$

where $P(s^i|\mathbf{x}_t)$ is the posterior probability estimated by the ANN, $P(\mathbf{x}_t)$ is a scaling factor constant for each observation that can be ignored and $P(s^i)$ is the class prior which can be estimated by counting the occurrences of each state in the training data.

As can be seen, the ANN-HMM hybrid approach is not novel nowadays. Nevertheless, with DNN the drastic change is the addition of multiple hidden layers in the ANN and the use of other architectures like Recurrent Neural Networkss (RNNs) or Convolutional Neural Networks (CNNs) as will be explained in Chapter 6. Other techniques aside from ANN can be used to obtain the state emission likelihoods like Support Vector Machines (SVMs) [23].

Furthermore, with the introduction of the so-called *end-to-end* approach in 2015, HMMs are removed and DNNs learn all the components of the speech recognizer. This approach is used in some modern commercial ASR systems nowadays. In this thesis however only hybrid models are considered because of the high computational demands (training end-to-end models require a high number of Graphics Processing Units (GPUs)) and the large quantity of speech data needed to train these models (usually in the order of thousands of hours).

How DNNs are trained and the different architectures normally employed in speech recognition are presented in Chapter 6, whereas our proposed improvements in this stage are shown in Chapter 6 and 7.

2.5 addressing the problem of robustness in asr

Automatic speech recognition systems can be affected by external influences not related with what has been spoken. It is worth noting that ASR systems by themselves are not robust to external influences in the speech signal as humans. In this section we present the distortions that cause the ASR systems to fail and discuss how the problem can be addressed.

The problem of speech recognition can be interpreted as a Bayesian classification problem. Following [16] the error can be classified into type 1 or type 2 errors. Type 1 errors represent the error given by the use of models to represent the true data distribution, as can be the HMM or the language model, because the process that generates speech is more complex than the acoustic/language models considered. In other words, the limited capability of these models will cause recognition errors. Type 2 errors are incurred by other information presented alongside the speech signal, as can be the pitch or the spectral harmonics of a particular speaker.

In addition to type 1 and 2 errors the speech signal can be influenced by external factors leading to both types of errors.

2.5.1 *External factors affecting the speech recognition performance*

Some of these external factors can be:

- *Signal capture:* the speech is captured using a microphone which converts pressure waves into electrical signals. Ideally the microphones should present a flat response across all frequencies but real microphones can introduce distortions since their frequency response is not flat and can vary due to external factors as, for example, the distance and direction from the speaker. Also other distortions can be introduced as for example harmonic distortion, or noise can be induced due to the fact that real microphones are not ideal devices.

  After the speech signal is captured by the microphone it needs to be digitized in order to be transmitted or stored. This process can produce a variety of distortions such as the loss of spectral information if the bandwidth of the channel is restricted, clipping if the signal exceeds the dynamic range of the analog to digital converter, distortions due to the quantification step, and coding distortions if the signal is coded to be more efficiently transmitted over various communication channels.

  The effects of the signal capture are not addressed in this thesis. For us the signal is captured ideally and the effects introduced by the microphone and channels are not considered. Figure 2.4 shows an utterance of one dataset employed to test our methods.

- *Additive distortions:* in some cases other signals are captured alongside the speech signal. These other signals are considered as noises, such as the sounds produced by air conditioning machines, cars, other people speaking and many others.

  These sounds are known as additive noises as the recorded signal is mainly the sum of speech and these different signals. Typically these signals are not correlated to the speech signal allowing simpler suppression.

  In our main experiments we have employed four types of additive noises: white noise, noise recorded on urban streets, single-speaker interference and background music.

  As the level of noise affects the accuracy of the ASR system, the recognizer performance is commonly evaluated at different SNRs.

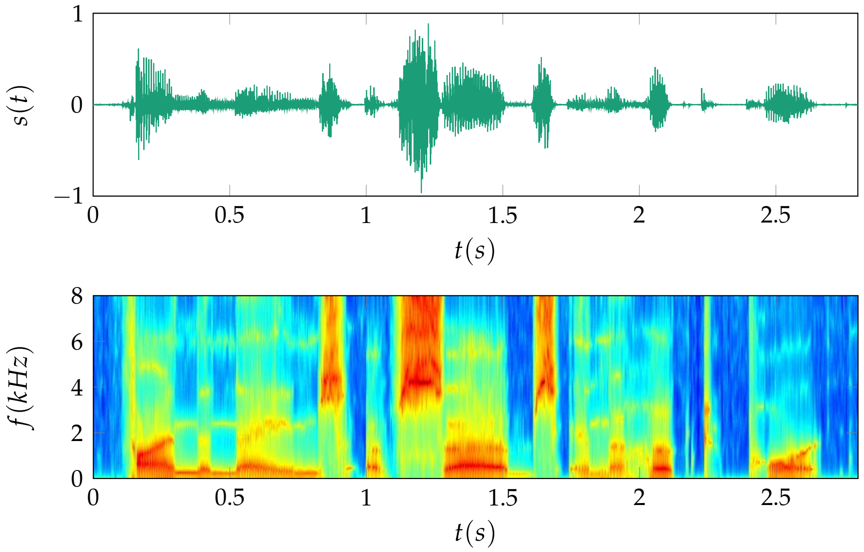


Figure 2.4: An example of a speech signal from the RM dataset [24] recorded using a professional microphone with a sampling frequency of 16 kHz. The top is the waveform and the bottom is the spectrogram of the signal. In particular the spoken sentence is: *find the names of serbs in the hooked port*.

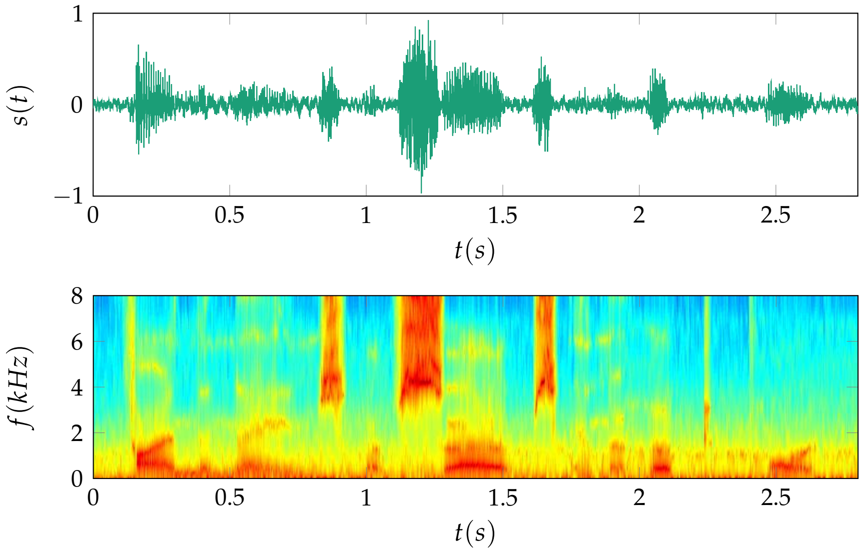


Figure 2.5: The signal from Figure 2.4 corrupted by and additive street noise at 10 dB SNR using the FANT tool [25].

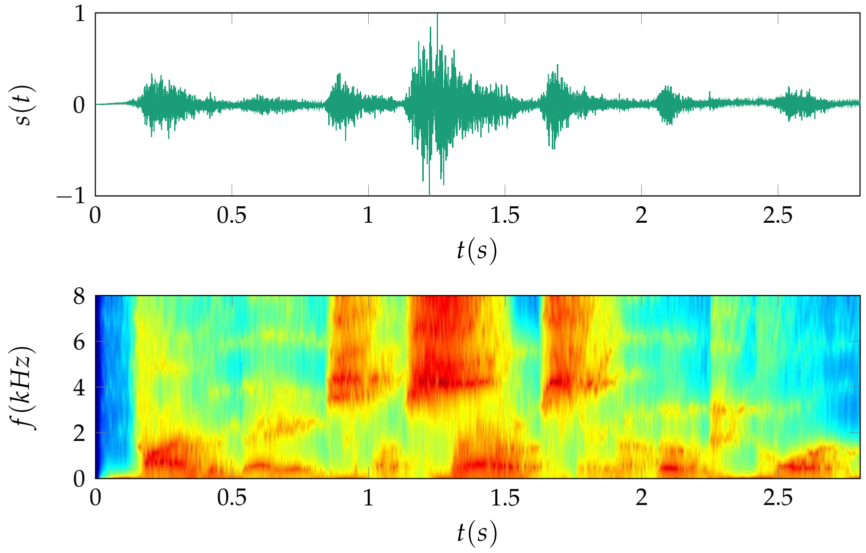


Figure 2.6: The speech signal from Figure 2.4 corrupted by passing it through a filter with impulse response derived from a room simulation algorithm using the image method [26] with $T_{60} = 0.5\,s$.

The SNR is expressed in decibels and is given by the following equation:

$$SNR(dB) = 10 \log \left(\frac{\sum_k s[k]^2}{\sum_k n[k]^2} \right), \tag{2.7}$$

where s is the clean speech signal and n is the noise. The higher the level of the noise in comparison with the speech signal, the lower the SNR. In particular, if SNR equals zero the speech and noise signals have the same power.

An example of the clean speech signal from Figure 2.4 corrupted by additive street noise at 10 dB SNR is presented in Figure 2.5. As can be observed, the waveform is highly distorted and its spectral patterns are hidden by those of the noise, specially at low frequencies, as the street noise has a strong low frequency component.

- *Reverberation:* the persistence of the sound after being produced due to reflections in the surrounding space where the speaker is talking is known as reverberation. It can be characterized by the reverberation time T_{60} defined as the time that the Sound Pressure Level (SPL) needs to decay 60 dB from its initial level.

  If the reverberation time is long, attenuated copies of what has been spoken in the past are mixed up influencing the ASR system. Even though there is a optimum reverberation time for a space where speech obtains an optimal human intelligibility, for speech recognizers this is an undesirable effect that needs to be addressed.

An example of the clean speech signal from Figure 2.4 corrupted by passing it through a filter with an impulse response derived from a room simulation algorithm using the image method [26] with $T_{60} = 0.5\,s$ is shown in Figure 2.6. As can be seen the spectrum is blurred in comparison to the original signal.

The main objective of this thesis is to mitigate the effects due to additive noise corruptions and, in some cases, reverberation distortions.

2.5.2 *Modeling the external factors*

To test our contributions we need a testbed in which the effects of additive noise, reverberation and other external influences can be introduced and measured in a controlled manner.

We can assume that the environment can be modeled as represented in Figure 2.7 where the speech signal is captured by the microphone obtaining the continuous electrical signal $s(t)$, then it is digitalized using an Analog to Digital Converter (ADC) obtaining $s[n]$. Usually, we do not need to perform the later steps as we use state-of-the-art datasets to test our methods.

Note that the effect of the noise and the environment presented in the initial signal, $s[n]$, are not taken into account, because typically these recordings are performed in a clean environment and with a close talking microphone or in an anechoic chamber, making the effect of reverberation and additive noise negligible at this stage.

The recorded signal is then fed into a linear filter, $h[n]$, and the noise, $n[n]$, is added, giving us the noisy version of $s[n]$:

$$x[n] = s[n] * h[n] + n[n], \tag{2.8}$$

where the effects of the reverberation and other convolutional distortions are modeled using $h[n]$, and different noises $n[n]$ at a variety of SNR are added in order to test different environments.

This is a reasonable model of the external influences of convolutional and additive noises. It was initially proposed by [27] and is widely used [16, 28]. Although many researchers prefer collecting recordings in real environment, for the sake of simplicity in this thesis we use the later model.

In particular for almost all the experiments noise has been added using the FANT tool [25] according to the International Telecommunication Union (ITU) recommendation P58, where a voice activity detector is used to determine an active speech level only from the speech segments of a recording ignoring pauses before and after a speech utterance and long pauses between words.

Reverberation is modeled with the Room Impulse Response Generator (RIR) [29] software where the room impulse response is obtained

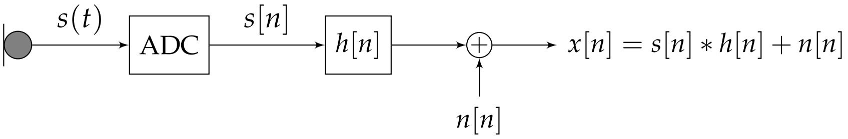


Figure 2.7: A simplified model of a noisy channel, which converts the ideally clean input signal $s[n]$ to the noisy signal $x[n]$ by passing it by a linear filter $h[n]$ and adding noise $n[n]$.

using the image method [26]. The room impulse response, in our case $h[n]$, is approximated with a Finite Impulse Response (FIR) filter that models the acoustic channel between a source and a receiver in rectangular rooms, allowing different reverberation times to be simulated.

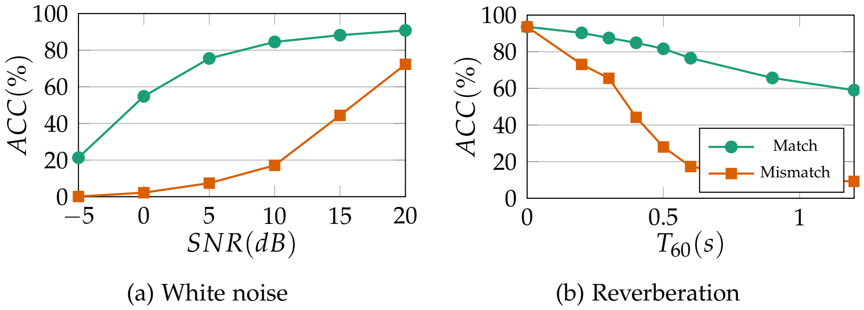
2.5.3 *How the external factors affect the speech recognizer performance*

To illustrate how the performance of the speech recognizer is affected by the external factors, an example of the accuracy achieved by a classical GMM-HMM ASR system under different external conditions is shown in Figure 2.8. The distortions are applied using the model presented in the previous section. The recognizer is trained and tested using the standard sets of the RM dataset [24] and the Kaldi toolkit for speech recognition [30], employing a traditional triphone GMM-HMM system with MFCC features.

Two external factors are considered: Figure 2.8a shows the recognition accuracy as function of the SNR of an additive white noise, and Figure 2.8b shows the recognition accuracy as function of the reverberation time in a simulated reverberation environment.

As can be seen, two different curves for each type of distortion are shown, on one hand the results when the recognizer is trained with clean speech (without the presence of noise or any other kind of distortion) and on the other, when it is trained using speech corrupted by the same condition as the test set. The first one is known as mismatched condition, and corresponds to a type 1 error, when the true data distribution of the test set is not modeled as the recognizer is trained with a clean set, compromising the performance of the recognizer. The second one is when the training set is corrupted by the same noise or reverberation conditions (in this case, at the same noise level or reverberation time) as the test set. This case is known as matched condition, and can be tied with a type 2 error. In general, in the matched case the recognizer is trained using a training set contaminated with different parameters, as the SNR of added noise, to those present in the test set.

In both cases, the performance of the recognizer worsens as the SNR decreases or the reverberation time increases, in the mismatched case



(a) White noise

(b) Reverberation

Figure 2.8: Recognition accuracy for the RM dataset under white noise corruption at different SNRs and simulated reverberant environment.

because the model moves away from the true data distribution and in the matched case due to the incorporation of other information in the speech signals that is not related with the words spoken. It is worth noting that the accuracy of the recognizer is higher when trained in the matched case but this condition is not always feasible as the test conditions can be unknown.

In this thesis our main contributions are aimed at obtaining an improvement of the performance in both matched and mismatched cases. Although nowadays with the advent of the deep-learning the matched condition has taken the lead as deep neural networks can generalize from large training sets that have been contaminated with large numbers of conditions, in our opinion the mismatched case still deserves attention.

2.5.4 *Techniques aimed at improving the robustness of the recognizer*

In this section we briefly review some of the most relevant contributions aimed at improving the accuracy of ASR systems under adverse environments. Since the literature on robust speech recognition is very large, interested readers can find an outstanding review in [28] or [16].

The main techniques can be divided into two groups following one of the possible classifications proposed in [28]:

- *Feature domain techniques:* these methods are usually based on extracting the proper features that are inherently robust to noise, or to modify the test features in order to match the training data distributions, while maintaining the parameters of the acoustic model unchanged. The methods that fall in this category can be divided into:
  - *Noise-resistant features:* this category can be further subdivided into: auditory based features and neural network approaches.

The former focus on how noise affects the speech signal and try to obtain an invariant representation of it, most of the times inspired by the HAS. There are plenty of auditory based features such us a Perceptually-based Linear Prediction (PLP) [31], Relative SpecTrAl processing (RASTA) [32] or Power Normalized Cepstral Coefficients (PNCC) [12, 33]. A more extended review of auditory features and how they are related to the HAS are presented in Chapter 3.

The later propose the employment of ANNs in the recognizer, two approaches can be found. On one hand ANN can be applied to obtain robust features using a bottleneck feature network and then those features can be combined with the GMMs estimations following Tamdem approach [34]. On the other hand it can be employed to obtain the emission probabilities of the HMM in the hybrid systems or the more recent Context Dependent-Deep Neural Network-Hidden Markov Model (CD-DNN-HMM) where DNN are used with long context window. These methods are explained in Chapter 6 as some of our contributions can be framed into this approach.

- *Feature normalization methods:* they are used to normalize the statistical moments of speech features. The most common technique is the Cepstral Mean and Variance Normalization (CMVN) or Mean and Variance Normalization (MVN) where the first and second order statistics are used to normalize the features.

  Normally in all the feature extraction procedures some degree normalization of the features is present. In this thesis all of the features are normalized in a per utterance basis employing MVN. By normalizing the features the training and testing acoustic characteristics have the same statistical moments reducing the mismatch between training and test conditions. Also, normalization can mitigate the effects of reverberation and of the frequency response of the microphone and/or transmission channel [35].

  Formally given the coefficients or the spectrum to normalize $X[m,l]$ for frame m at the l cepstral coefficient or frequency filter output for non cepstral features, MVN is computed as follows. First, the mean and standard deviation are obtained over the entire utterance as:

$$\mu[l] = \frac{1}{M}\sum_{m=1}^{M} X[m,l], \quad 1 \leq l \leq L, \tag{2.9}$$

$$\sigma[l] = \sqrt{\frac{1}{M}\sum_{m=1}^{M} (X[m,l] - \mu[l])^2}, \quad 1 \leq l \leq L, \tag{2.10}$$

where M is the total number of frames in the utterance and L the number of coefficients or filters. Second, the normalized features are obtained by:

$$X[m,l] := \frac{X[m,l] - \mu[l]}{\sigma[l]}, \quad 1 \leq m \leq M, \quad 1 \leq l \leq L. \tag{2.11}$$

- *Feature compensation:* this case is when the noise is removed given the observed features. Most of the feature compensation methods attempt to enhance the speech signal using signal processing techniques without extensive modeling of the HAS properties.

  Noise compensation techniques are pervasive in ASR. Some of them are based on the (partial) suppression of background noise from the speech signal in a preprocessing stage. Most of these methods operate on the frequency-domain, the quintessential method of this category is Spectral Subtraction (SS) [36] where the noise is eliminated in the spectral domain using the noise spectrum estimated during non-speech periods.

  The spectral subtraction technique was first proposed by [37]. In this method, a Voice Activity Detector (VAD) is used to estimate the noise spectrum $N(m,l)$ by averaging all the frames where no speech is present. Then the spectral coefficients $X(m,l)$ are modified to obtain the denoised frames as:

  $$|X[m,l]| := \max\left(|X[m,l]| - \alpha|N[m,l]|, \delta|X[m,l]|\right). \tag{2.12}$$

  where α allows overestimating the magnitude spectrum of noise and usually depends on the SNR of the utterance and δ is a small constant to avoid negative values in the spectrum.

  In some of our experiments SS is used for purposes of comparison. In particular we do not use an explicit VAD and therefore, the noise spectrum is estimated in the first frames of each utterance.

  Other techniques that can be included in this category are Wiener Filtering [38], Minimum Mean Square Error (MMSE) short-time spectral amplitude estimator [39]. A combination of these techniques are used in the well known European Telecommunications Standards Institute (ETSI) Advanced Front-End [40] that include several noise suppression algorithms.

- *Model domain:* these methods modify the acoustic model parameters in order to incorporate the effects of the noise, compensating for the mismatch between training and testing data noises.

> Methods that fall into this category are speaker adaptation techniques such as speaker independent Discriminative Mapping Transformation (DMT) [41] or the popular Maximum Likelihood Linear Regression (MLLR) [42].
>
> Methods that adapt the acoustic model parameters by explicitly addressing the nature of the distortion caused by the presence of noise such as Vector Taylor Series (VTS) [43] or Parallel Model Combination (PMC) [44] also belong to this category.

The main contributions presented in this thesis are framed into the feature domain methods, by proposing novel robust auditory features that are inspired by the human auditory system and using novel deep neural networks based robust hybrid systems. Also some of the proposed features employ some degree of feature compensation methods in order to increase the performance of the ASR systems.

A detailed introduction to auditory features is presented in the next chapter and the proposed features are explained in detail in Chapter 4 and 5. The contributions concerning the DNNs hybrid systems are explained in Chapter 6 and 7.

2.6 DATASETS

In this section we review the different datasets used in the experiments and the reasons to choose each of them. We have performed our tests in the well known, for robust tasks, Aurora 2 and 4 databases, and also in noise versions of Isolet, RM, Texas Instruments and Massachusetts Institute of Technology (TIMIT) and Wall Street Journal 0 (WSJ0).

2.6.1 *Aurora 2*

Aurora 2 database [45] consists of a set of connected digits spoken by American English speakers and recorded at a sample rate of 8 kHz. The database is contaminated with a selection of 8 different real-world noises (subway, babble, car, exhibition hall, restaurant, street, airport and train station) at different signal-to-noise ratios. In particular, SNRs from 0 dB to 20 dB with 5 dB step were considered for our experimentation.

The recognizer used for this dataset was based on Hidden Markov Model Toolkit (HTK) software package [46] with the configuration included in the standard experimental protocol of the database described in [45], where a standard GMM-HMM system with a 16-state word-based HMM and a 5-state silence model was adopted.

As our main experiments with this database were performed in mismatched conditions, acoustic models were obtained from the clean training set of the database, whereas test files correspond to the complete test sets A, B and C.

This is a small dataset that allows us fast implementation and testing of our different algorithms, in particular for our auditory motivated features.

2.6.2 *Aurora 4*

The Aurora 4 database [47] is a medium-vocabulary task based on the WSJ0 corpus. The experiments were performed using the 16 kHz clean and multi-condition training sets. Each training set consists of 7137 utterances from 83 speakers. The clean training set contains only clean data recorded with a single microphone.

On the other hand, the multi-condition training uses different microphones and artificially added noises. In particular, it is corrupted with six different noises (street traffic, train station, car, babble, restaurant, airport) at SNRs of 10-20 dB.

The evaluation set is derived from the WSJ0 5K test set corrupted with the same noises and recorded with different types of microphones, creating a total of 14 test sets with 330 utterances each. Note that the types of noise are shared across multi-condition training and test sets but the SNRs of the data are not. The results presented using this dataset are averaged across all the tests sets of the Aurora 4 dataset.

A clean and multi-condition development sets are also available, we used it for tuning parameters of the networks and as a validation set during the training.

For the experiments performed in the Aurora 4 database two different recognizers have been used. On one hand, the standard triphone Kaldi [30] recipe is used for the traditional GMM-HMM systems. Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) are applied over the features for speaker adaptation. The US English generic bigram language model and the Carnegie Mellon University (CMU) pronouncing dictionary are used.

On the other hand, Aurora 4 dataset is used for our DNN architectures experiments, using Kaldi and Tensorflow [48]. A detailed experimental setup is given in Chapter 7.

This dataset is large enough to test our methods in a more realistic environment while maintaining a reasonable size allowing fast iteration in our developments with our limited resources.

2.6.3 *Wall Street Journal 0 (WSJ0)*

The WSJ0 database [49] consists of read speech from a machine-readable corpus of Wall Street Journal (WSJ) news text [49].

The WSJ0 is a clean speech dataset. Nevertheless, for experiments performed over it, a noise contaminated version is used.

To test the robustness of the different methods we used the same four standard testing environments as [33]: (1) white noise, (2) noise recorded live on urban streets, (3) single-speaker interference and (4) background music. The street noise was recorded on streets with steady but moderate traffic. The masking signal used for single-speaker-interference experiments consisted of other utterances drawn from the same database as the target speech, and background music was selected from music segments from the original Defense Advanced Research Projects Agency (DARPA) Hub 4 Broadcast News database.

For training the acoustic models, we used the WSJ0 SI-84 training set which contains 7 308 clean recordings, making a total of 14 hours. The different experiments were performed on noisy versions of the WSJ0 5K test set, obtained by digitally adding the previously mentioned noises—white, street, speaker and music—to the corresponding clean speech at four different SNRs using the FANT tool [25] with G.712 filtering. More information about this procedure can be found in Section 2.5. All the noisy tests performed in this dataset are evaluated in mismatched conditions (that is, training on clean speech and testing on noisy speech).

Experiments using this dataset were performed using the HTK recipe described in [50], employing a tri-gram language model with 5k vocabulary size and the CMU pronunciation dictionary.

This dataset allows us a direct comparison of our auditory feature extraction algorithms with the PNCC on which they are based and is our main baseline to overtake, as the same experimental set up for which they were designed is used.

2.6.4 *Resource Management (RM)*

The DARPA RM dataset [24] is a clean continuous speech database. The complete dataset consist of 25000 utterances from more than 160 american speakers, recorded at 16 kHz. For our experiments the RM1 section is used. We use the same subsets of [33] with 1600 utterances of clean speech for training and 600 utterances of clean or degraded speech for testing.

The same experimental set up of the WSJ0 dataset is used. This dataset also allows us a direct comparison with the PNCC baseline, while being smaller than the WSJ0.

2.6.5 *Isolet*

The Isolet database [51] consists of 7 800 English alphabet spoken letters (two productions of each letter per each of the 150 speakers) at a sample rate of 16 kHz. Specifically, we used a version of this database called noisy-Isolet [52] where the original Isolet was contaminated with 8 different noise types from the Noisex database at several SNRs (clean, 0, 5, 10, 15 and 20 dB). The noise types are: speech babble, factory noises 1 and 2, car, pink, F-16 cockpit, destroyer operations room and military vehicle noise.

The experiments were performed using the Isolet testbed described in [52], where a hybrid ANN-HMM system [53] is used. This testbed employs the Quicknet Multi-Layer Perceptron (MLP) package for acoustic modeling [54].

Isolet database allowed us to perform our first experiments using a hybrid configuration with a one layer MLP when a GPU was not available.

2.6.6 *TIMIT*

The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus [55] consists of clean speech. Each utterance is recorded at 16 kHz and the corpus includes time-aligned orthographic, phonetic and word transcriptions allowing us to give results in terms of Phone Error Rate (PER).

In particular, we used the 462 speaker training set, the 50 speaker development set to tune all the parameters and finally the 24 speakers core test set.

To test the robustness of the different methods we followed the lines of the previously presented datasets, adding noise to the test set with the FANT tool [25] for four different noises (white, street, music and speaker) at different SNRs. All tests were evaluated in a mismatched condition.

This dataset is employed in our first DNN experiments using Kaldi toolkit [30] for implementing the traditional GMM-HMM ASR system and the Python Toolkit for Deep Neural Networks (PDNN) toolkit [56] for the hybrid DNN-based ASR systems.

Also the TIMIT dataset allows us to perform analysis of the errors of the different systems based in phonetic classes as we describe in [5].

3 ROBUST FEATURES MOTIVATED BY THE HUMAN AUDITORY SYSTEM

3.1 INTRODUCTION

The remarkable ability of humans in speech recognition tasks under noisy conditions is still far above that of machines. In this context, several researchers have proposed that modeling the Human Auditory System (HAS) may be an adequate strategy to reduce the gap in performance.

In this chapter we describe the basic mechanisms of human hearing. By understanding it, it is possible to design a feature extraction stage that allows Automatic Speech Recognition (ASR) systems to increase their performance in noisy and hard environments. Classic auditory models and feature extraction techniques are presented giving the reader the background to later understand our contributions.

The state of the art in feature extraction and auditory modeling is very broad. For a more extended review, we find Chapter 5 from [16] outstanding, also Chapter 4 from [35] is a good review of hearing and auditory models, and for topics related to the processing of the human hearing system [57] is a remarkable resource.

In the next sections, first, a general description of the human auditory system is presented alongside an explanation of the most important physiological observations giving insights on how humans process speech. This is necessary to understand the motivations of the auditory inspired features. Then, the general feature extraction procedures motivated by the auditory processing used by speech recognition systems are explained in detail.

3.2 HUMAN AUDITORY SYSTEM

The Human Auditory System (HAS) or human hearing system plays an important role in the everyday life of humans, mainly allowing the communication between us. The HAS has evolved during thousands of years to clearly and intelligibly understand speech in all kinds of situation. Understanding how it works and apply its fundamentals to ASR systems improves performance in noisy and hard conditions.

Many parts of the human hearing mechanisms such as the *auditory cortex* in the brain are not yet well understood while the *auditory periphery* has been studied in detail.

As shown in Figure 3.1, following [35], the human auditory system can be divided into three main stages:

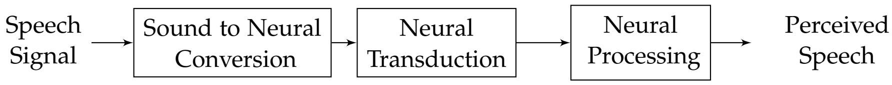


Figure 3.1: Model of the HAS: path from the speech signal to the perceived and understood speech.

1. *Sound to neural representation conversion*: first a neural representation of the acoustic wave containing the speech is obtained by the outer, middle and inner ear, also known as the auditory periphery. The speech signal travels through the air as an acoustic waveform of pressure variations, arriving to the outer ear where the external ear and the external ear canal conduct the wave to the middle ear.

   Then the tympanic membrane together with the so called hammer, incus and stapes bones act as a mechanical transducer that transforms the acoustic wave into mechanical vibrations. The middle ear also amplifies the sound leading to the remarkably low threshold of hearing that humans have in the particular range of frequencies of the speech.

   Next the mechanical vibrations are converted into neural impulses in the inner ear, which is composed of two organs: the semicircular canals and the cochlea. The former maintains the equilibrium of the human body, and the later is of vital importance as it converts the mechanical vibrations into electrical impulses (neural action potentials) that are transmitted to the brain.

2. *Neural transduction*: the electrical impulses generated by the cochlea are transmitted to the brain by the auditory nerve fibers.

3. *Neural processing*: finally the auditory nerve arrives at the auditory cortex in the brain where the nerve firings are processed creating the perceived speech.

As stated before, neural transduction and processing are not yet well understood, so we can only model them as a black-box where the acoustic signal is converted into psychological observations. These psychological observations can be obtained empirically giving us insights into how the human auditory system processes acoustic signals. In this chapter we review some physiological and psychoacoustic properties of the human auditory system that are usually employed in ASR feature extraction stages.

Although we do not know in detail the neural transduction and processing stages, the mechanisms that take place in the inner ear converting the speech waveform into neural stimuli is well documented.

The cochlea is the organ that transforms the mechanical vibrations into neural stimuli. It is composed of a spiral tube filled with fluid, where the vibrations induce a wave motion sensed by the basilar membrane which acts as a spectrum analyzer. In the basilar membrane, thirty thousand Inner Hair Cell (IHC) are moved by the fluid at different rates and when the vibration is large enough the hair cells produce a spike that is transmitted through the auditory nerve to the brain. Each hair cell is only sensitive to a particular range of frequencies depending on its location in the basilar membrane. Also the auditory nerve fibers are more sensitive to the frequencies of the hair cells which they are connected. This fact is known as the synchrony effect and is modeled in one of our contributions.

In the next subsections the psychoacoustical effects and physiological properties of the HAS as related to our proposals are presented.

3.2.1 *Frequency resolution of the human auditory system*

The basilar membrane can be thought as a bank of filters in which the bandwidth of each filter is proportional to its central frequency. Nowadays, the modeling of this effect is, to some extent, present in all feature extraction stages.

It is widely accepted that the cochlea carries out a logarithmic compression of the auditory range whereby higher frequency intervals are represented with less detail than lower frequency ranges. This realization stems from experiments to detect critical bands, i.e. the frequency bandwidth around a center frequency whose components affect the sound level and pitch perception of that center frequency.

In this light, the notion of an auditory filter-bank relates to three concepts:

- A discretization of a frequency range into N bands.
- A choice of the center of the bands to be related to special frequencies or frequency ranges in the inner ear, which entails the definition of a frequency scale.
- A choice of the bandwidths and shapes of the different filters that take into consideration the notion of critical bands.

The use of logarithmic frequency scales eases the conceptualization of phenomena like *masking*, *loudness* and *pitch*.

Many scales of logarithmic frequency have been proposed being the most relevant: the *critical-band rate scale*, the *Mel scale* and the Equivalent Rectangular Bandwidth (ERB) induced scale. All of them use methods to calculate the critical bandwidths at different center frequencies and at the same time define scales of equal difference in perception of pitches/levels related to those center frequencies.

Almost all of the auditory inspired feature extraction techniques start with a filter-bank where each filter has its center frequency and bandwidth defined along one of this frequency scales:

- *Critical band and critical-band rate scale:* The Bark scale was first proposed by [58]. It is based on traditional masking experiments and is defined as:

  $$F_z(f) = \frac{26.8}{1 + \frac{1960}{f}} - 0.53, \tag{3.1}$$

  where F_z is defined in bark units and f in Hz. For example the cochlear masking models described in Chapter 4, which are derived from a set of psychoacoustic experiments, are defined in terms of the Bark scale.

- *Mel scale:* The Mel scale [59] (Mel comes from *melody*) is a very well-known logarithmic transformation of the frequency scale. It comes from pitch comparisons and is approximated by:

  $$F_m(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{3.2}$$

  where F_m is in mel units and f in Hz. This frequency transformation is at the core of the most popular ASR feature extraction procedure, the Mel-Frequency Cepstral Coefficients (MFCC), where a filter-bank of triangular overlapping filters uniformly distributed in the mel scale is usually employed.

- *ERB scale:* It was defined in [60, 61] as a more adjusted measurement of the critical bands estimated using notch maskers. The bandwidth for each frequency is defined as:

  $$BW_{ERB}(f) = 6.23f^2 + 93.39f + 28.52 \;\; (f \text{ in kHz}). \tag{3.3}$$

  Based upon these bands a new logarithmic scale may be defined, the ERB-rate [62]:

  $$F_{ERB}(f) = 11.17 \log \left| \frac{f+0.312}{f+14.675} \right| + 43.0 \;\; (f \text{ in kHz}), \tag{3.4}$$

  or the ERB number:

  $$ERB_N(f) = 21.4 \log(4.37f + 1), \tag{3.5}$$

  also with f in kHz.

  Alternatively, a filter-bank can also be defined in the time domain by its impulse response, e.g. [63]:

  $$h_{f_c}(t) = kt^{n-1} exp(-2\pi Bt) cos(2\pi f_c t + \phi), \tag{3.6}$$

  where k defines the output gain, n is the order of the filter—in the range 3-5 the filter is a good approximation of the human auditory filter—, B defines the bandwidth, f_c is the filter's central frequency and ϕ is the phase. f_c is typically chosen uniformly

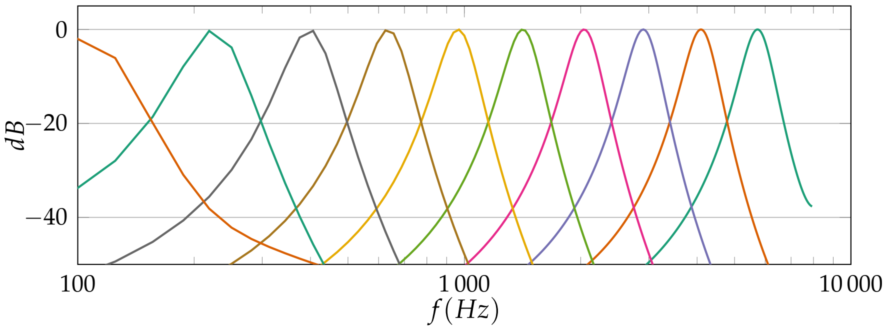


Figure 3.2: Frequency response of a gammatone filter-bank composed of 10 filters, in which central frequency are uniformly spaced according to the ERB scale.

spaced according to the ERB scale. A filter-bank composed of 10 filters is show in Figure 3.2.

This filter-bank is known as gammatone filter-bank and simulates the cochlea in a more realistic manner than the traditional triangular filters spaced in the mel scale, in such a way that it provides better recognition rates in ASR task. In addition, according to [63], the impulse response of the gammatone function provides an excellent fit to the human auditory filter shapes allowing better modeling of the masking phenomena, as we will present in the next chapter.

3.2.2 *Psychoacoustical transfer function*

The relationship between the physical intensity of the acoustic signal and the perceived loudness have been studied extensively. A physiological observation of the auditory nerve firing rate in comparison to the intensity of the acoustic wave has been measured [64], concluding that this relationship is nonlinear and compressive.

Many scales have been proposed to model this effect, the most relevant being the Fechner [65] psychophysical scale and the Stevens power law [59].

On one hand, Fechner scale is motivated by the decibel concept, proposing that the relationship between the physical intensity U and the perceived loudness V, can be approximated by:

$$V = c\log(U). \tag{3.7}$$

On the other hand, Stevens, supported by experiments where the subjective evaluation of the perceived intensity was asked to a set of listeners, proposes the following relation:

$$V = c_1 U^{c_2}, \tag{3.8}$$

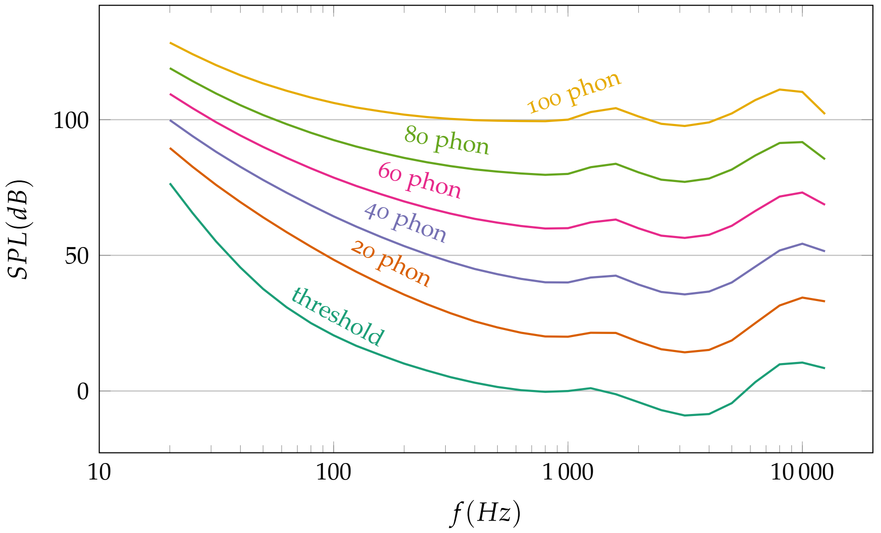


Figure 3.3: Equal loudness contours as described in ISO 226 norm.

where c_1 and c_2 are constants fitted to the experimental data, being $c_1 = 1$ and $c_2 = 0.33$ the most common values.

As will be explained later, most common feature extraction algorithms include one of these relationships. For example, in MFCCs the Fechner logarithmic nonlinearity is used and in the Perceptually-based Linear Prediction (PLP) the Stevens power law is considered. The recent Power Normalized Cepstral Coefficients (PNCC) propose to use a different exponent in the Stevens power law based on the physiological observations of [64].

3.2.3 *Auditory thresholds and equal loudness curves*

Another important psychoacoustic result present in many feature extraction and auditory models is the equal loudness contours representation [65]. Each curve defines the Sound Pressure Level (SPL) in which a constant loudness is perceived across the spectrum with pure tones. A unit of loudness, the *phon*, is also defined as the value of the curve at 1 kHz. Figure 3.3 shows the equal loudness curves from 0 to 100 phon.

In order to obtain these curves a study was performed where the subjects listened to pure tones and a reference tone at 1 kHz and the level of the reference was adjusted until the listener perceived the same loudness.

It is worth noting that the lowest equal loudness contour represents the absolute threshold of hearing and the higher the threshold of pain.

3.2.4 *Masking*

Cochlear masking is the phenomenon where the perception of some frequency at a particular time instant, known as the masked frequency, is affected by the sound level of another, the masker frequency, possibly at a different time instant, to the extent that masked frequencies may not be perceived.

The masking effect is applied in almost all lossy audio coding algorithms, where masked sounds can be eliminated without the listener perceiving the difference.

In the next chapter, a model of the masking behavior of the HAS is use to enhance the robustness of the acoustic features is introduced. A detailed description of the masking phenomenon is presented together with our proposed models and speech recognition experiments.

3.3 CLASSIC AUDITORY MODELS

In this section the most relevant auditory models are reviewed. Detailed modeling of the human auditory physical attributes and psychophysical facts started in the 1980s with the models of Seneff [66], Ghitza [67] and Lyon [68, 69].

The Seneff's auditory model has been employed widely. In this thesis we use some parts of it to model the synchrony effects as will be seen in Chapter 5.

As can be seen in Figure 3.4 the Seneff's auditory model is composed of three stages:

- *Stage I:* models the peripheral auditory frequency response using a bank of bandpass filters. 40 recursive linear filters are implemented in cascade following the nominal auditory-nerve frequency responses described in [70]. The filters are distributed following the Bark scale. This stage simulates the filtering performed by the basilar membrane in the inner ear.

- *Stage II:* models the probabilistic behavior of the inner hair cells of the cochlea, the first synapses and the nerve fibers, obtaining as an output the probabilities of firing over time of a set of auditory nerve fibers, modeling the nonlinear transduction from the motion of the basilar membrane to the mean rate of auditory-nerve spike discharges.

  Stage II is composed of four steps: (1) nonlinear half-wave rectification that uses for positive inputs the inverse tangent function and the exponential function for negative inputs and models the hair cell firing only in positive outputs, (2) short-term adaptation that models the release of the neurotransmitter in the synapse, (3) a lowpass filter with cutoff frequency of approximately 1 kHz to suppress synchronous responses at higher input frequencies, and (4) a rapid Automatic Gain Control (AGC) stage to maintain an approximately-constant response rate at higher input intensities when an auditory-nerve fiber is nominally in saturation.

- *Stage III:* has two parallel modules whose inputs are the Stage II outputs. The first module approximates the instantaneous mean rate of firing and the second measures the synchrony in response to the incoming signal.

  The first one is implemented as an envelope detector to model the instantaneous mean rate of response of a given fiber. The second operation is called a Generalized Synchrony Detector (GSD) and is motivated by the Average Localized Synchrony Rate (ALSR) measure proposed in [71]. The hair-cell output is compared to itself delayed by the reciprocal of the center frequency of the filter in each channel, and the short-time averages of the sums and differences of these two functions are divided by one another. A threshold is introduced to suppress responses to low-intensity signals and the resulting quotient is passed through a saturating half-wave rectifier to limit the magnitude of the predicted synchrony. The GSD provides a useful representation of the spectral components, including noise.

With the limited computational resources available at that time, Seneff could only compare the mean rate and the GSD response visually for some selected inputs. The Seneff's auditory model has great significance in our approach since our synchrony features, presented in Chapter 5, are computed using its first and second stages.

Other models as the Ghitza's Ensemble Interval Histogram (EIH) [67] use the peripheral auditory model proposed by Allen [72] to describe the transformation of sound pressure into the neural rate of firing and focus on the mechanism used to interpret the neural firing rates, or Lyon's model [68, 69] where nonlinear compression, lateral suppression, temporal effects and correlograms are included.

Also detailed auditory periphery models have been proposed by physiologists rather than researchers in the field of speech technology, to describe the auditory periphery in detail in contrast to the abstractions of later presented models. For example, the approach in [73] models the rate of spikes using flow dynamics of the neurotransmitter being able to describe a large number of physiological data.

An important model is the Zhang *et al.* model [74] that simulates the auditory nerve activity by using a gammatone filter-bank whose gains and bandwidths are controlled by a control path. This allows the description of other physiological phenomena as two tone suppression.

In [75] a comparison of the auditory nerve model of [74] with a simplified version applied to robust speech recognition tasks is presented. This simplified model consists of: (1) a gammatone filter-bank, (2) a

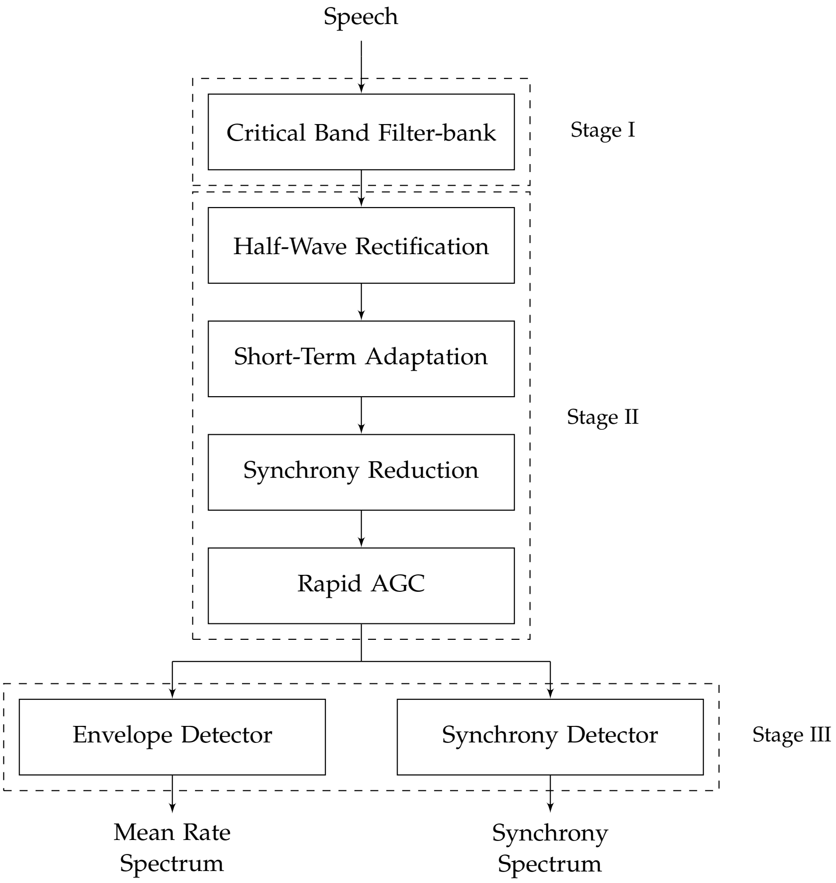


Figure 3.4: Structure of the Seneff auditory model.

non linear rectifier and (3) a low pass filter. The gammatone filter-bank models the mechanical responses to the sound, the rectifier the fact that the auditory nerve fibers only fire when the signal is positive and the low pass filter is use to suppress synchronous responses at higher input frequencies as in the Seneff's model. This model obtains a reasonable gain in performance while keeping the computational cost low, and is at the base of the PNCC [12, 33] feature extraction method explained in the next section.

Although these models do not generally provide improved performance on the recognition rate for clean speech, they obtain better results than conventional feature extraction methods when speech is degraded, for example, with added noise or reverberation. However, the usually higher computational cost and complexity (with a large number of parameters to be tuned) have prevented a more widespread adoption. In the next section we present some feature extraction methods where their resemblance of the HAS on the one hand and computational cost on the other, have been balanced to allows their use in practical speech recognition tasks.

3.4 AUDITORY BASED FEATURES

It is well established that feature extraction methods for ASR need to take into account properties of the HAS to a certain extent. The quintessential example of feature extraction process is the Mel-Frequency Cepstral Coefficients (MFCC) [21], proposed in 1980 are the most used features in automatic speech recognition systems.

Figure 3.5 shows the MFCC feature extraction procedure following these steps:

1. The input speech signal passes through a pre-emphasis filter, with response $H(z) = 1 - 0.97z^{-1}$. This filter serves the purpose of emphasizing the spectral properties of the vocal tract and reducing the effect of glottal pulses [27].

2. Then a Hamming window of 20-30 ms. length is employed. The windowing is applied each 10-15 ms.

3. The power spectrum of the windowed signal is obtained using a Short-Time Fourier Transform (STFT).

4. The power spectrum is filtered and weighted with an auditory filter-bank of triangular filters that are equally spaced in the Mel scale, obtaining a set of Mel-Frequency Spectral Coefficients (MFSC). Usually, the number of mel-scaled filters is set to 40.

5. The logarithm of each of the individual MFSC coefficients is taken.

6. The Discrete Cosine Transform (DCT) is applied obtaining a feature vector of MFCCs. Usually each 10-15 ms a feature vector containing 13 MFCCs is extracted.

7. Frequently the MFCCs are normalized using Cepstral Mean and Variance Normalization (CMVN), as explained in Section 2.5.4.

From the MFCC extraction process we can see that some considerations of the HAS have been taken into account:

1. The triangular mel-scaled filter-bank employed tries to emulate the critical bands in the cochlea.

2. The non-linear transformation (logarithm) applied over the outputs of the mel-scaled filter-bank mimics the Fletcher's psychophysical transfer function [65]. Also the non-linear perception of sound intensity is incorporated by means of a logarithmic transformation of the spectrum.

3. The DCT can be interpreted as a low pass filtering of the mel-spectrum [35], modeling the response of the auditory nerve fibers, as in the Seneff model where in the Stage II a lowpass filter is used to suppress synchronous responses at higher input frequencies.

Another common feature extraction method that takes into account the human auditory system is the PLP [76], which is a pragmatic approach to model the auditory periphery. The procedure for PLP computation, shown in Figure 3.5, it consists of the following steps:

1. Computation of the power the spectrum in the same fashion as in the MFCC using a sliding window and a standard STFT.

2. The power spectrum is integrated using a trapezoidal filters in the bark-frequency scale

3. Equal loudness pre-emphasis is applied to approximate the unequal sensitivity of the human hearing according to the threshold of hearing.

4. A non-linearity is introduced based in the power law proposed by [59].

5. Inverse Fast Fourier Transform (IFFT) and Linear Prediction (LP) are performed follow by a cepstral recursion obtaining the final features.

6. Frequently the PLP coefficients are normalized using Mean and Variance Normalization (MVN), as explained in Section 2.5.4.

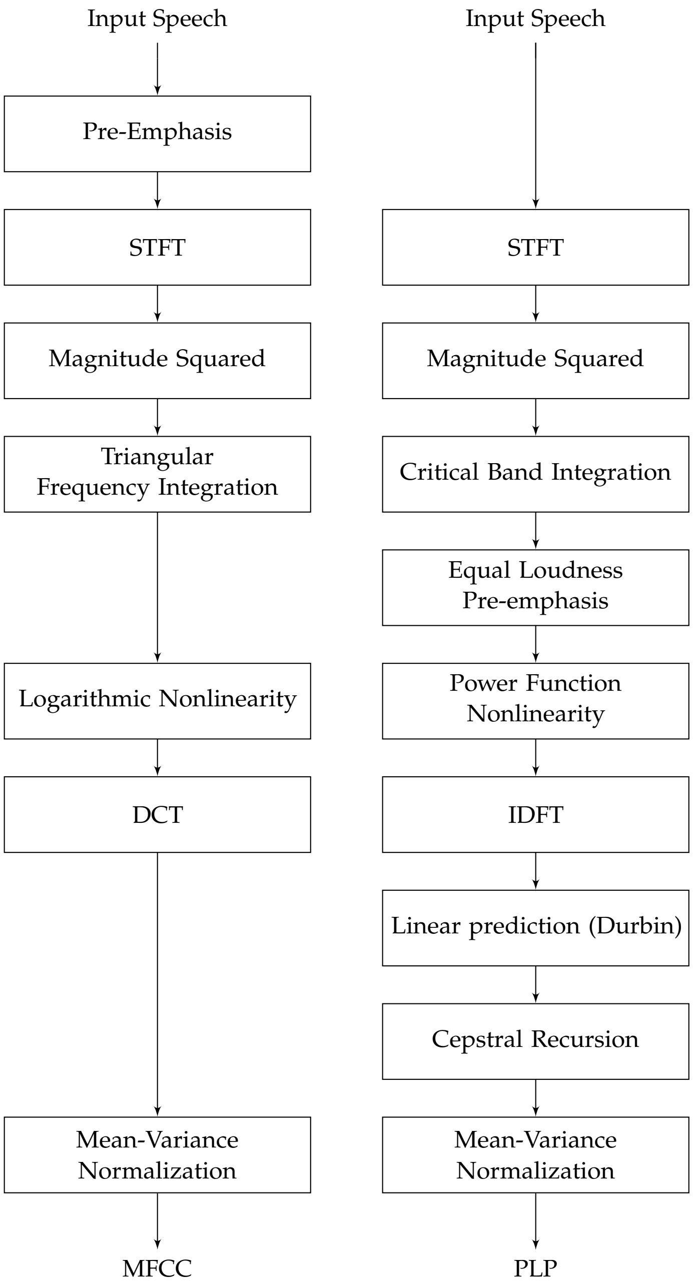


Figure 3.5: Pipeline of the MFCC and PLP feature extraction process.

It is worth noting that the computational complexity of PLP feature extraction is similar to MFCC and sometimes provides better recognition accuracy.

The PLP extraction process takes into account the following considerations of the HAS:

1. The critical bands in the cochlea are emulated with trapezoidal filters distributed according to the Bark frequency scale.
2. It approximates the unequal sensitivity of the human hearing to different components of the signal by using the equal loudness contour plots.
3. The psychophysical transfer function is applied by means of the use of the Stevens power law with cubic root on the power spectrum.

As can be seen the MFCC and PLP try to model to some extent the HAS, but many effects that take place in the human hearing system are not modeled. There are plenty of other feature extraction methods that take into account the HAS, such as Zero Crossing Peak Amplitude (ZCPA) [77], Average Localized Synchrony Detection (ALSD) [78], Perceptual Minimum Variance Distortionless Response (PMVDR) [79], Invariant-Integration Features (IIF) [80], Amplitude Modulation Filter Bank (AMFB) [81, 82], SParse Auditory Reproducing Kernel (SPARK) [83] or the well-known Relative SpecTrAl processing (RASTA) [32] that exploits the insensibility of human hearing to slowing varying stimuli by modeling the trend of the auditory periphery to emphasize the transient portions of incoming signals.

Though most of the algorithms described above include spectro-temporal notions, these are incorporated in separate stages of the processing pipeline. The idea of simultaneously performing temporal and spectral analysis to yield so-called *spectro-temporal features* has lately emerged, e.g. spectro-temporal Gabor features [84–86], HIerarchical Spectro-Temporal (HIST) [87], spectro-temporal derivative features [88] or sparse spectro-temporal features [89]. Auditory-inspired representations in these domains are reviewed in [90].

Lately PNCC were proposed as a way to model the HAS while maintaining a low computational complexity. In the next section the PNCC algorithm is explained in detail as some of our contributions are based on it.

3.4.1 *Power Normalized Cepstral Coefficients (PNCC)*

Power Normalized Cepstral Coefficients (PNCC) [12, 33] have been proposed as an alternative to capture the essentials of the HAS without the complexity of full psychoacoustical models. PNCC are based on the simplification of the previously presented physiological auditory models and the addition of a medium time non-linear processing that removes the effects of additive noise and reverberation, as will be explained in the next paragraphs.

PNCC has an important role in our contributions in the feature extraction stage as we use their spectro-temporal representation, or *cochleogram*, and noise compensation techniques in our developments.

The main innovations of the PNCC algorithm are: (1) the replacement of the traditional logarithmic non-linearity used in MFCC coefficients with a power-law non-linearity that provides a better fit to the onset portion of the rate-intensity curve developed by the model of [64], (2) the use of a noise-suppression algorithm based on asymmetric filtering that suppresses background excitation and a module that carries out a *temporal masking* by placing a peak for each frequency channel and suppressing the instantaneous power if it falls below that of the envelope, (3) the use of a medium time processing (50-120 ms.) to estimate the environment degradation, and (4) a low computational cost allowing its real time use.

The PNCC pipeline is depicted in Figure 3.6 and it is composed of the following steps:

1. *Initial Processing:* the same pre-emphasis filter as in MFCC is applied, then a STFT with a Hamming windows of 20-30 ms and 10-15 ms frame interval is used to obtain the spectrum $X[m, e^{jw_k}]$ where m represents the frame index and w_k is the discrete time frequency $\frac{2\pi k}{K}$ with K the STFT size. The outputs are weighted with a 40 normalized gammatone filter-bank equally spaced in the ERB scale to obtain the short time power $P[m, l]$:

$$P[m, l] = \sum_{k=0}^{(K/2)-1} |X[m, e^{jw_k}] H_l[e^{jw_k}]|^2, \tag{3.9}$$

   where $H_l[e^{jw_k}]$ is the response of the l-th gammatone filter at frequency w_k, m is the frame index and l is the channel index.

2. *Temporal Integration for Environmental Compensation:* A medium time power $\widetilde{Q}[m, l]$ is obtained by running a moving average over $P[m, l]$, with a temporal integration factor of five frames.

3. *Asymmetric Noise Suppression with Temporal Masking:* Motivated by the fact that the speech power changes faster that the background power, the authors develop an asymmetric filter that obtains an average noise power that is subtracted from the medium-time power, obtaining the enhanced power $\widetilde{R}[m, l]$. It is worth noting that the temporal integration and noise suppression can be seen as a noise compensation technique that suppresses the effects of additive noise and reverberation.

   The asymmetric filter has various parameters that need to be optimized if the algorithm is used in other dataset different from

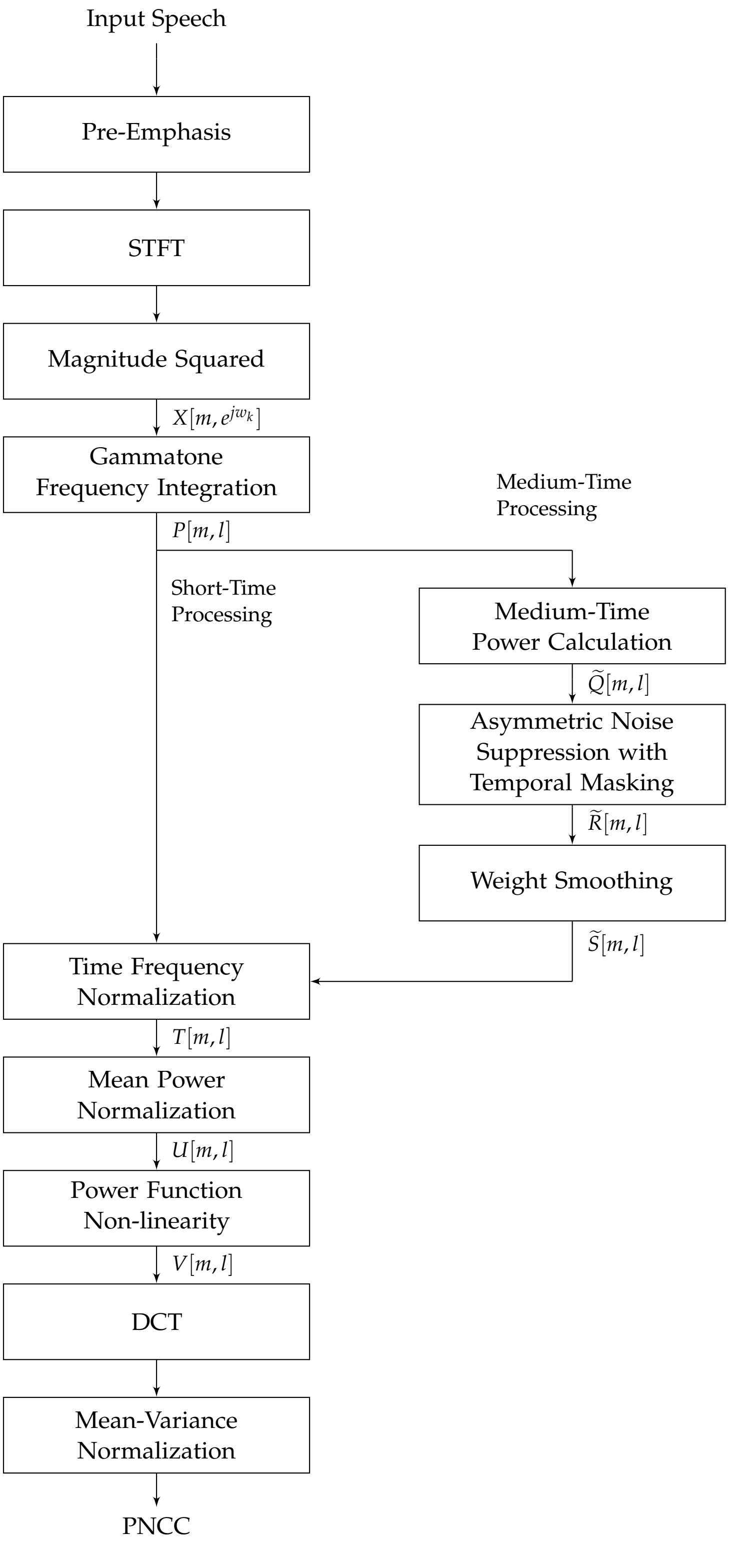


Figure 3.6: Pipeline of the PNCC feature extraction algorithm.

the one used in the original paper where these features were initially proposed.

Also a temporal masking block is included by obtaining a moving peak for each channel and suppressing the instantaneous power if it falls below the corresponding envelope.

4. *Spectral Weight Smoothing:* a smoothing of the response across channels is done by averaging the transfer function of the last steps. The frequency averaged transfer function $\widetilde{S}[m,l]$ is obtained as follows:

$$\widetilde{S}[m,l] = \frac{1}{l_2 - l_1 + 1} \sum_{l'=l_1}^{l_2} \frac{\widetilde{R}[m,l']}{\widetilde{Q}[m,l']}, \tag{3.10}$$

   where $l_1 = \max(l-N,1)$ and $l_2 = \min(l+N,L)$, where $L = 40$ is the number of channels and $N = 4$, obtaining the frequency averaged transfer function across the four upper and lower adjacent channels.

5. *Time Frequency Normalization:* The short-time power $P[m,l]$ is modulated by the later smoothed transfer function following:

$$T[m,l] = \widetilde{S}[m,l]P[m,l]. \tag{3.11}$$

6. *Mean Power Normalization:* the normalized power $U[m,l]$ is obtained as:

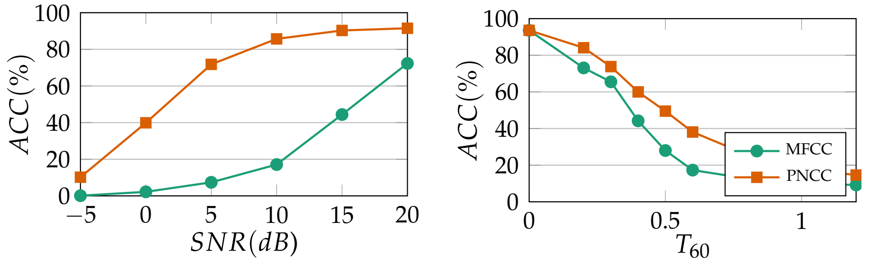
$$U[m,l] = k\frac{T[m,l]}{\mu[m]}, \tag{3.12}$$

   where $T[m,l]$ is the obtained processed power, $\mu[m]$ is a mean power estimate computed by a moving average with a time constant of 4.5 seconds and k is a constant fitted to the data. This is done to reduce the impact in the changes of the incoming signal.

7. *Rate-Level Non-linearity:* to model the psychoacoustical Transfer Function the authors use the physiological experiments of [64] to obtain the synapse output for different tones and fit the curve, obtaining a power law with an exponent of $1/15$. The final power representation $V[m,l]$ is obtained using this non-linearity:

$$V[m,l] = U[m,l]^{\frac{1}{15}}. \tag{3.13}$$

8. *Final Processing:* The coefficients are obtained in the same fashion as in the case of MFCC employing a DCT to obtain 13 uncorrelated coefficients and finally MVN is applied, as explained in Section 2.5.4.



(a) White noise

(b) Reverberation

Figure 3.7: Recognition accuracy for the RM dataset under white noise distortion at different SNRs and in a simulated reverberant environment for the traditional MFCC and PNCC feature extraction processes.

As an example of the performance of MFCC and PNCC for noisy speech recognition tasks, Figure 3.7 presents the results obtained by the PNCC in comparison to MFCC in the Resource Management (RM) dataset for white noise and reverberation in mismatched conditions. As can be seen in this figure and in the original paper [12] where more extended results are shown, PNCC features provide dramatic performance improvements over conventional MFCC and also over others as PLP, Vector Taylor Series (VTS) or the European Telecommunications Standards Institute (ETSI) advanced front end [33].

One drawback of the PNCC is that in order to use it in other datasets different from the one where it were originally developed, it may be necessary to optimize the parameters of the feature extraction process. Our proposed contributions aim at capturing auditory attributes that make an impact in the speech recognition performance minimizing the number of parameters to be adjusted. In addition PNCC processing does not consider the synchrony representation and the lateral suppression effects. Both lacks are included in the feature extraction stage in the next chapters.

We try to get inspiration from the HAS and in particular, all the steps presented in Figure 3.1 are modeled in this thesis: the inner-ear representation is modeled with a modified version of the PNCCs where the masking effects that take place in the cochlea are considered in detail (Chapter 4), the neural transduction is simulated by introducing the synchrony effect performed in the auditory nerve (Chapter 5), and the neural processing step is modeled using deep learning techniques (Chapter 6 and 7).

4 POWER-NORMALIZED COCHLEOGRAMS FEATURE EXTRACTION

4.1 INTRODUCTION

As stated in the previous chapter, mimicking the human auditory system may contribute to improve the performance of Automatic Speech Recognition (ASR) systems in noisy and hard conditions. Specifically, in this chapter we model the masking behavior of the Human Auditory System (HAS) to enhance the robustness of the feature extraction stage in ASR. Despite ingrained intuitions that masking deteriorates signal quality, we propound that it smooths away some noise and artifacts.

Methods that comprise procedures that emulate HAS masking can be found in the literature: as seen in the previous chapter, Power Normalized Cepstral Coefficients (PNCC) that includes temporal masking, simultaneous or frequency masking is considered in [91] where a frequency-dependent masking threshold is computed, or [92] that performs an estimation of the clean signal taking into account masking effects. In [93] both temporal and simultaneous masking are incorporated performing a time-frequency noise spectral subtraction.

The three cornerstones of our procedure are first, the use of *mathematical morphology* operations to emulate the masking processing of the cochlea, second, the design of *a single auditory-inspired three-dimensional mask* independent of frequency and intensity and third, the use of an adequate *underlying spectro-temporal representation* of speech such that the non-linearities in frequency and intensity observed in the auditory masking phenomena are significantly equalized licensing a biologically meaningful application of the two previously mentioned elements.

In particular, our model filters a cochleogram—a spectro-temporal representation of speech— as if it were an image, allowing for the simultaneous processing of both dimensions, time and frequency. For that purpose, the *mask*—or in mathematical morphology terminology, the Structuring Element (SE) reproduces the spectro-temporal masking behavior as induced from well-known empirical measurements. Thus, the design of the SE is the crux of our approach.

Note that these empirical measurements were either carried out in the spectral or the temporal domains separately, but we need to extrapolate this to both dimensions. In this chapter, we present various structuring element designs that aim at closely resembling the auditory masking phenomena that take place in the cochlea and we also refine our hypothesis that morphological filtering produces a smoothing of the spectro-temporal envelope that better models the masking behavior of the cochlea.

In previous works by our group [94, 95], some evidence of the utility of the morphological filtering of speech spectrograms with a roughly-approximated SE was presented. Such rough modeling already yielded an enhancement of the filtered speech both in terms of objective quality measures and ASR performance. Note that, although some work has been carried out in the field of morphological processing of speech spectrograms using dilation across spectral lines to reduce spectral fluctuations [96], such efforts did not take into account the properties of the HAS.

Finally and for the sake of simplicity, we employ a single mask across all frequencies and intensities despite the fact that the masking properties are frequency- and sound intensity-dependent [57], relying on the underlying spectro-temporal representation to accommodate these effects. The proper choice of this representation is essential in our feature extraction method. We have selected the one proposed as part of the PNCC [12, 33] in combination with conventional spectral subtraction.

In summary, our contribution in this chapter models the HAS masking phenomena by using Morphological Filtering (MF) operations while maintaining a low computational cost with very few tuning parameters. A key aspect is the design of a single bio-inspired three-dimensional SE that is used across the board unlike other spectro-temporal techniques that need a large number of different bases as in [84–86], for example, where a reduced set of temporal, spectral and spectro-temporal filters need to be chosen to make it feasible. For this single SE to remain invariant in frequency and intensity we rely on an underlying spectro-temporal representation that already accounts for that variability. In particular, we have borrowed that of PNCC—even improving the temporal masking there included—while maintaining a low computational complexity with respect to the PNCC baseline.

The rest of this chapter is organized as follows: Section 4.2 introduces the underlying spectro-temporal representation, Section 4.3 explains the theoretical and empirical basis of cochlear masking, Section 4.4 describes our three-dimensional model of this phenomenon introducing the basic terminology of mathematical morphology and the design of our biologically inspired SE. Finally Section 4.5 presents the results obtained in various datasets followed by some conclusions and further lines of research in Section 4.6.

4.2 SPECTRO-TEMPORAL REPRESENTATION

As highlighted before, the underlying spectro-temporal representation—the cochleogram—where the morphological filtering will be applied needs to adopt the necessary frequency scaling and intensity normalization to allow for a single SE to be valid across the full spectrum and intensity range.

Our masking models have been tested in two different auditory-motivated frequency-scaled cochleograms, known as *Mel-frequency and Power Normalized based spectro-temporal representations*. The cochleograms are denoted by $V[m, l]$ where m is the frame index and l the frequency index or filter output. A detailed description of both spectro-temporal representations was presented in the previous chapter.

Dashed boxes in Figure 4.1 contain the block diagrams of the two spectro-temporal representations considered in this work: Mel-frequency (left) and Power Normalized (right). The outputs of both submodules are the corresponding cochleograms $V[m, l]$ on which further processing with morphological filters is applied as explained in subsection 4.4.3. Note that spectral subtraction, show as a shadow block after Short-Time Fourier Transform (STFT), is not part of the original mel-frequency and power-normalized representations computations, but it is included here as a basic denoising technique (see subsection 4.4.3).

4.3 COCHLEAR MASKING EMPIRICAL RESULTS AND MODELS

As mentioned in the previous chapter, the cochlea is the organ that converts the mechanical vibrations in the middle ear to neural impulses. The basilar membrane—the sensing structure that runs the length of the cochlea—has a particular frequency and time response [97].

Cochlear masking is the phenomenon whereby the perception of some frequency at a particular time instant, the *masked frequency*, is affected by the sound level of another, the *masker frequency*—possibly at a different time instant—, to the extent that masked frequencies may not be perceived.

A masking tone will be defined as

$$s(t, F) = L_m \delta(t - T_m, F - F_m), \tag{4.1}$$

where F is expressed in any of the transformed frequency scales introduced in Section 3.2.1, L_m is the sound pressure level of the tone, F_m and T_m are the masker frequency and time instant and δ represents the Dirac delta function.

Cochlear masking has been studied as the effect of a masker on simultaneously masked frequencies, *simultaneous masking*, and as the phenomenon whereby a masker affects non-simultaneous frequencies, *temporal masking*. Classical masking experiments concentrated in determining the amount of masking in either of these directions—frequency or time—in isolation. But it is important to notice that a given (masked) frequency is *always* being masked by maskers at

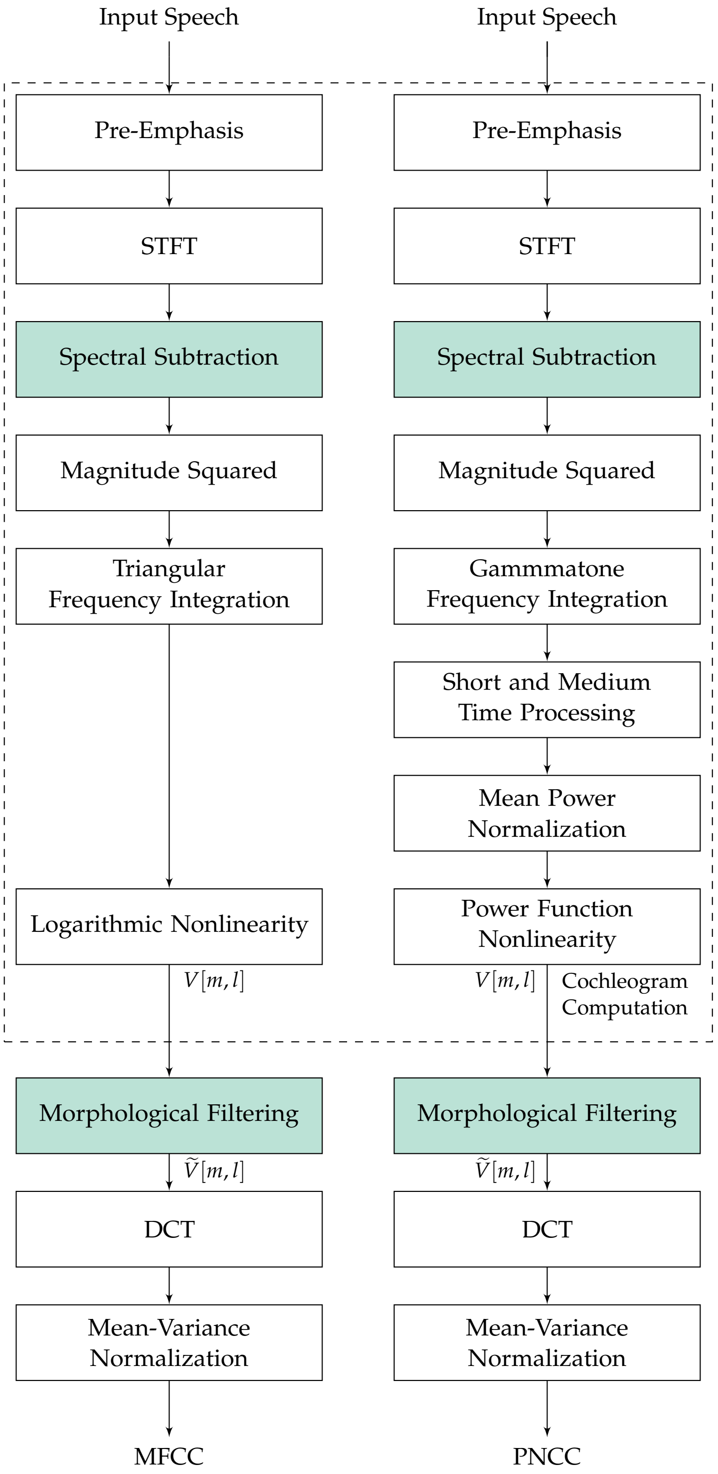


Figure 4.1: Structure of the proposed front-ends for the two spectro-temporal representations; the dashed boxes contain the submodules corresponding to the mel-frequency (left) and power-normalized (right) representations. The shaded blocks (Spectral Subtraction (SS) and Morphological Filtering (MF)) indicate the differences regarding conventional MFCC-based and PNCC-based feature extraction.

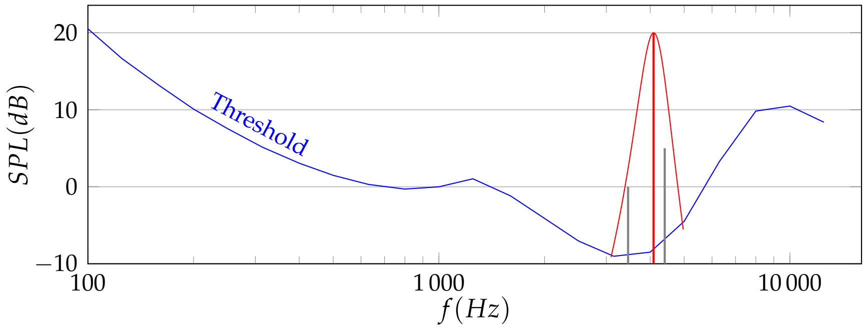


Figure 4.2: An example of simultaneous masking, where a masker tone, represented in red, makes the gray tones, known as masked tones, disappear from perception.

different time instants—both from earlier and later maskers—and frequencies—both from lower and higher frequency maskers.

4.3.1 *Simultaneous masking*

Simultaneous masking is defined as the minimum sound pressure level of a *test sound, probe or signal*—normally a pure tone—that is audible in the presence of a *masker*. By varying the frequency of the probe throughout the spectrum, a *masking pattern* may be obtained.

Figure 4.2 shows an example of simultaneous masking, where the masker tone (drawn in red), shifts the hearing threshold (blue curve) making the gray masked tones disappear from perception. All the signals below the red curve are not perceived by the subject.

An experimental fact is that the shape and sound pressure level L_m of the masker is quite determinant of the masking pattern. Regarding the change of masking with masker parameters, [98] noticed that simultaneous masking is better represented in logarithmic scales where the spacing and the masker frequency slopes extend more regularly to either side of the spectrum.

A simultaneous masking model can be extracted from the psychoacoustic experiments data presented in [57] (Figure 6.14 in the text), by fitting linear slopes for $L_m = 60\,\mathrm{dB}$ in the Bark scale. We assume a constant L_m across all frequencies and intensities, relying on the underlying spectro-temporal representation to accommodate the frequency-intensity dependency of the masking properties.

4.3.2 *Temporal masking*

Temporal masking has methodologically been treated as two separate processes: *premasking* occurs before the appearance of the masker while *postmasking* manifests itself after the masker is no longer present.

It is well agreed-upon that premasking is noticeable about 20 ms prior to the masker, while the duration of postmasking extends well beyond 200 ms, perhaps as far as 500 ms [94].

Thus, premasking can be modeled as a constant slope of +25 dB/ms, starting 20 ms before the masker. Postmasking can be modeled with the fitted model for single masker-induced postmasking presented in [99], defined as:

$$M\left(t - T_m, L_m\right) = a\big(b - \log\left(t - T_m\right)\big)\big(L_m - c\big), \tag{4.2}$$

where M is the amount of masking, t is measured in ms, L_m is the masker Sound Pressure Level (SPL) in dB, and a, b and c are parameters obtained by fitting the curve to the data, in particular:

- a is related to the slope of the time course of masking.
- b is the logarithmic of the probe-masker delay intercept.
- c is the intercept when masker level is expressed in dBL.

4.3.3 *Smoothed masking responses*

As suggested in the previous sections, an idealized masking model for a masker at (T_m, F_m) could be a cone with the appropriate decays in the logarithmically scaled frequency and time coordinates. But findings consistently suggest a masking model that is smooth around (T_m, F_m), with sublinear decays or concave downwards ($f(x+y) <= f(x) + f(y)$) close to this point and superlinear decays or concave upward ($f(x+y) >= f(x) + f(y)$) further away [57]: a sort of apex-smoothed cone.

At this point, it is worth mentioning that it seems that the masking capabilities of the cochlea co-evolved in the presence of a noise that has the peculiarity of raising masking thresholds uniformly, that is, giving a flat frequency response [57]. We hypothesize that at the level of granularity at which the cochlear response is being observed, this phenomenon is also present, and the masking response of a particular tone (T_m, F_m) must be the non-linear aggregation of many masking responses of other neighboring masking tones $(T_m + \Delta T, F_m + \Delta F)$ with $\Delta T << T_m$, $\Delta F << F_m$ which accounts for the smooth sublinear decay. This would manifest as a smoothness constraint for the model of the masking response in the neighborhood of (T_m, F_m). These observations will be used in Section 4.4.2 to constraint the SE.

4.4 A THREE-DIMENSIONAL MODEL OF COCHLEAR MASKING

4.4.1 *An overview of morphological processing*

Mathematical morphology is a theory for the analysis of spatial structures [100] whose main application domain is in image processing as a tool for thinning, pruning, structure enhancement, object marking, segmentation and noise filtering [101]. It may be used on both binary and grey-scale images.

To perform MF operations, we first convolve the image with a structuring element and then select the output value depending on the thresholded result of the convolution. The MF operation is applied on *cochleograms*, our underlying spectro-temporal representation, that will be processed as if they were images. This spectro-temporal representation is explained on Section 4.2.

With a proper choice of the SE, morphological operations on the cochleogram reproduce the phenomenon of auditory masking where the most prominent or salient elements of the cochleogram mask their surroundings in the temporal and frequency domains.

Erosion and *dilation* are the basic morphological operations. Erosion is used to reduce objects, while dilation produces an enlargement and fills in small holes. Let V be the underlying spectro-temporal representation and M the three-dimensional structuring element, erosion is denoted as: $V \ominus M$ and dilation: $V \oplus M$.

Erosion and dilation with a general structuring element require relatively simple algorithms and there are fast implementations that allow us to perform such operations efficiently. For gray-scale images, erosion is the minimum over the structuring element and dilation the maximum, respectively. Specifically, for a pixel at index $[m, l]$ where l is the frequency bin and m the frame index these operations can be defined as follows:

$$\begin{aligned}(V \ominus M)(m,l) &= \min_{(\phi,\tau)\in\mathbb{Z}^2} \{V[m,l] - M[m-\phi, l-\tau]\}, && (4.3)\\ (V \oplus M)(m,l) &= \max_{(\phi,\tau)\in\mathbb{Z}^2} \{V[m,l] + M[m-\phi, l-\tau]\}, && (4.4)\end{aligned}$$

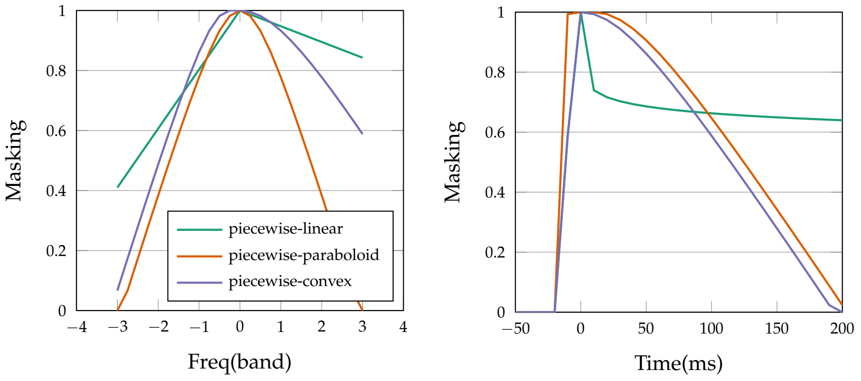
where ϕ and τ range over the domain of definition of the structuring element M.

There are two possible operators generated by the combination of erosion and dilation using the same structuring element for both operations: opening ($V \circ M$) and closing ($V \bullet M$). The first one is an erosion followed by a dilation and the second, a dilation followed by an erosion. Mathematically they can be expressed as:

$$\begin{aligned}V \circ M &= (V \ominus M) \oplus M, && (4.5)\\ V \bullet M &= (V \oplus M) \ominus M. && (4.6)\end{aligned}$$

The opening operator tends to remove the outer tiny leaks and round shapes, whereas the closing operator preserves the regions that have a similar shape as the structuring element. Previous experiments carried out by our group [94] show that closing performs better for ASR than opening.

For producing the final *masked* cochleogram $\widetilde{V}$, first the closing operator is applied on the original (possibly de-noised) spectro-temporal



(a) Simultaneous masking

(b) Temporal masking

Figure 4.3: Comparison between the piecewise-linear, piecewise-paraboloid and piecewise-convex models in both frequency and time domains.

representation V using the structuring element M and the result is subsequently added to V:

$$\widetilde{V} = \lambda V + (1 - \lambda)V \bullet M, \tag{4.7}$$

where λ is a configuration parameter that weights both contributions, $\lambda = 1$ indicates no morphological filtering and corresponds with our baseline system.

The tuned parameter λ sets a trade off between clean and noisy performance, we typically select a value in the range 0.45-0.50, based on the recognition rate in a development set, although the performance is not affected significantly.

From this enhanced cochleogram $\widetilde{V}$, mel-frequency or power normalized based coefficients are computed following the procedure explained in subsection 4.4.3 and following the pipeline represented in Figure 4.1.

4.4.2 *Structuring element*

In this section we describe the auditory motivated structuring elements that try to emulate the complex phenomenon of cochlear masking when used in combination with MF. The SEs act as the cochlea's response to tone maskers, and the morphological filtering mechanism reproduces the masking itself. Three different structuring elements are presented, the *piecewise-linear*, *piecewise-paraboloid* and *piecewise-convex* models.

The basic piecewise-linear model for masking can be observed in Figure 4.3 (green line). This SE is built with linear slopes for the simultaneous masking model and the logarithmic model of Equation 4.2 for the temporal masking. In this model, referred to as the *ideal model*

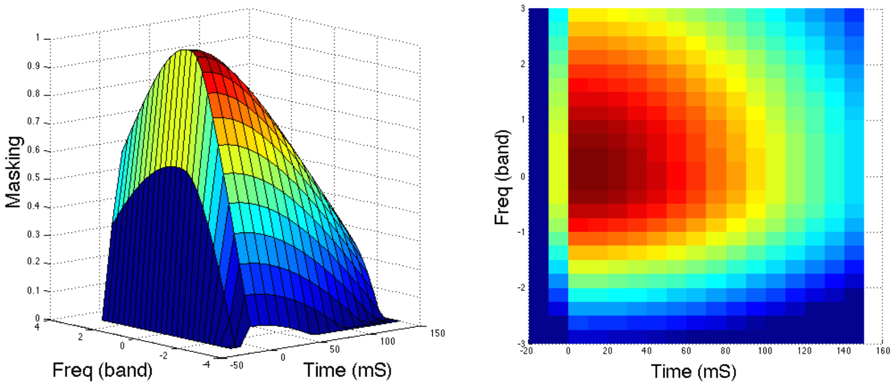


Figure 4.4: Three-dimensional representation of the piecewise-convex SE. Color represents the weight of each pixel in the morphological operations.

of masking in Section 4.3.3, the SE for a single frequency-time point at $[m,l]$ is not smooth.

To be consistent with the smoothness constraint we created two new SE based in three dimensional (3D) quadrics, built by aggregating 4 asymmetric quadric quadrants with different parameters centered at $[m,l]$ fitted to the empirical models in Sections 4.3.1 and 4.3.2.

The piecewise-paraboloid model is built by aggregating paraboloid quadrants and the piecewise-convex model using hyperboloid quadrants. A comparison of the masking response of these models with the piecewise-linear model projected onto the time and frequency coordinates can be observed in Figure 4.3.

As confirmed by the results in Section 4.5, filtering with the piecewise-convex obtains the best performance. Different sizes in both frequency and time domains were tested, and the best performance was obtained by taking 10 ms of premasking, 150 ms of postmasking, and 6 bands (in bark scale) in frequency. The 3D shape of this structuring element can be seen in Figure 4.4. Note how temporal and simultaneous masking are interpolated by the quadrics over the parameters suggested by the pure temporal and frequency models mentioned in Section 4.3. The asymmetry in the slopes towards higher and lower frequencies reflects the choice of different parameters to define the hyperboloids in each quadrant. This effect is more evident in the post-masking than in the pre-masking part of the SEs skirt.

Since the cochlear masking model is defined in terms of the Bark scale but the spectro-temporal representations considered in this work are related to the Mel (MFCC) or Equivalent Rectangular Bandwidth (ERB) (PNCC) scales, the appropriate transformations between scales are applied before the morphological processing (see Section 3.2.1).

Finally, a normalization between zero and one was applied on the intensity dimension and the SE was padded with zeros in the negative time region to center it in the mask around the pixel in which the morphological closing operation is to be performed.

The SE finally chosen can be seen at the upper left of Figure 4.5(a) at scale, along with examples of the output of some of the processing steps leading to the final cochleogram.

4.4.3 *Morphological filtering-based front-ends*

In this subsection, we describe how the morphological filtering is embedded in the whole feature extraction process for automatic speech recognition.

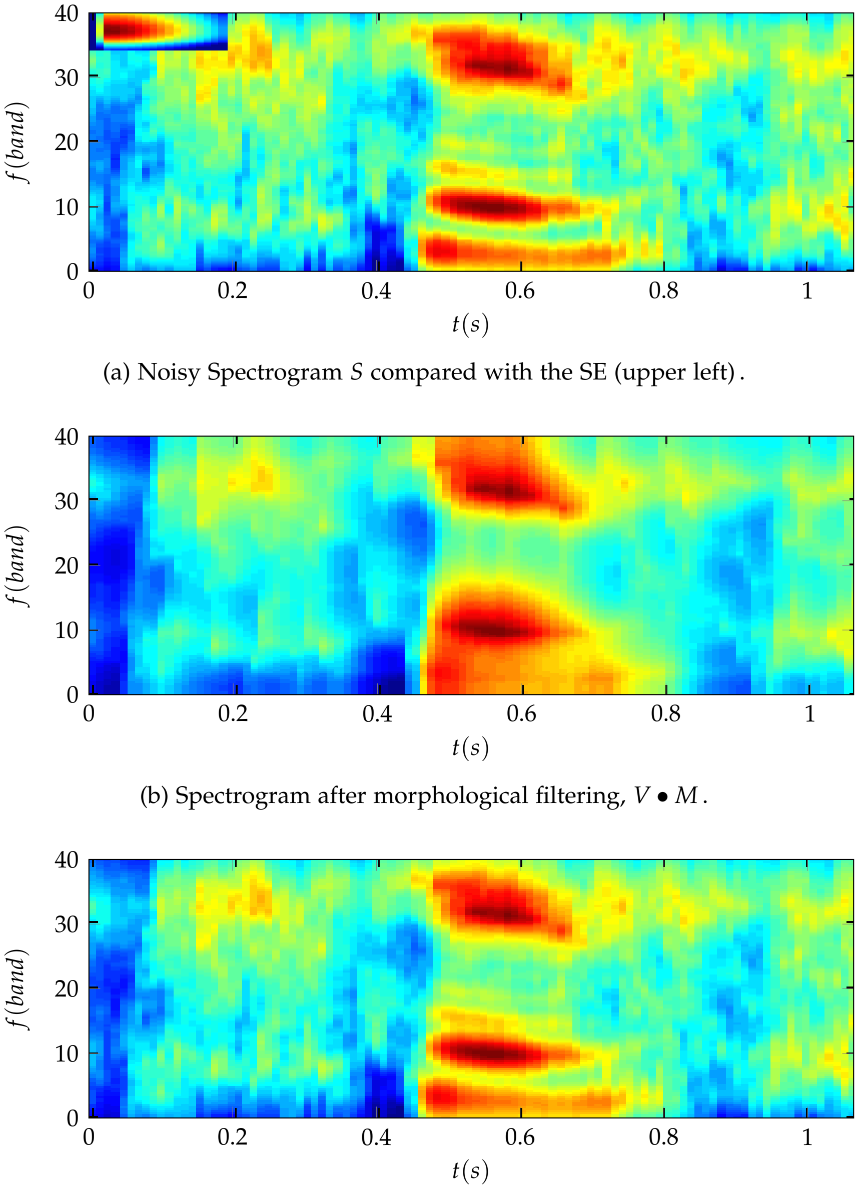
Figure 4.1 represents the block diagram of the two complete proposed front-ends based on Mel-frequency (left) and Power Normalized (right) spectro-temporal representations where the shadow blocks are our additions to, respectively, conventional MFCC and PNCC feature extraction: MF and SS. What we call a *masked cochleogram*, $\widetilde{V}[m,l]$, is obtained by performing morphological filtering on $V[m,l]$ using one of the single structuring elements described in Subsection 4.4.2. As for the spectral subtraction block, we found synergies with MF under the MFCC framework [94, 95, 102] that we also confirmed for PNCC (see Section 4.5). The last two blocks in both schemes carry out the usual procedure, to de-correlate the resulting filter-bank energies by means of the Discrete Cosine Transform (DCT), followed by a Cepstral Mean and Variance Normalization (CMVN).

4.5 EXPERIMENTAL RESULTS

In this section we present the experiments carried out on three different datasets: Aurora 2, Isolet and a noisy version of the Wall Street Journal 0 (WSJ0) dataset. All of then were presented in Section 2.6.

4.5.1 *Feature extraction*

As mentioned before, two different spectro-temporal representations were considered: mel-frequency and power-normalized cochleograms (see Section 4.2). For either type, speech was analysed using a frame length of 25 ms and a frame shift of 10 ms. After preemphasis and Hamming windowing an auditory filter bank analysis was applied over the spectrogram computed by using the STFT. In particular, in the case of the mel-frequency representation, a set of triangular mel-scaled filters was used, whereas, for power-normalized cochleograms a bank of 40 gammatone-shaped filters whose center frequencies are linearly spaced in the ERB scale between 200 Hz and 4000 Hz was applied, followed by the PNCC [12] medium-duration power bias subtraction and power function nonlinearity. In both cases, in order to decorrelate the filterbank log-energies obtained in the previous stage, a DCT was



(a) Noisy Spectrogram S compared with the SE (upper left).

(b) Spectrogram after morphological filtering, $V \bullet M$.

(c) Final cochleogram S' with $\lambda = 0.5$.

Figure 4.5: Selected spectrograms output by each step of the architecture.

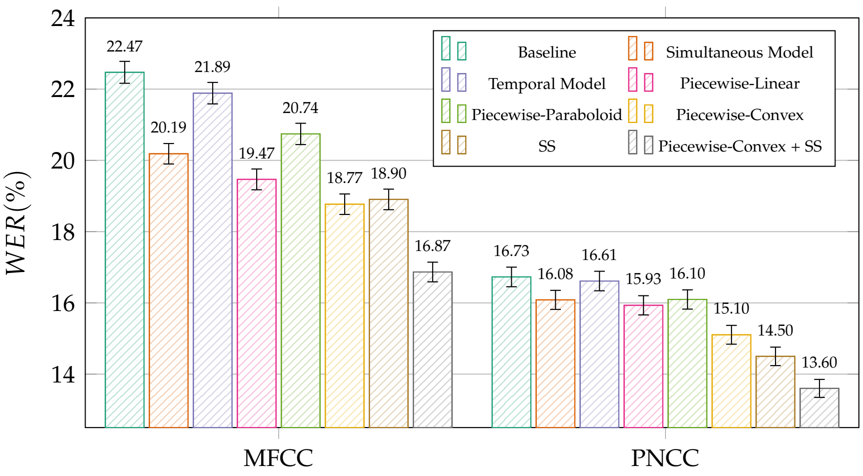


Figure 4.6: Recognition results in terms of WER(%) and 95% confidence intervals using the Aurora 2 dataset (averaged over all test sets).

computed over them. Cepstral coefficients C_0 to C_{12} were retained together with their corresponding first and second order derivatives to yield feature vectors of 39 components. CMVN were applied on each of the components.

When indicated, a conventional SS was employed over the noisy signal in order to emphasise speech over noise and MF was apply over the corresponding enhanced cochleograms. Samples of the feature files for the different datasets, and the scripts to replicate the results on the Aurora 2 dataset are available at [103].

4.5.2 *Aurora 2 dataset*

We used the Aurora 2 dataset, to test our model, and to select the best structuring element. As explained in Section 2.6.1 where details about the dataset were presented, the standard experimental protocol of the database described in [45] using Hidden Markov Model Toolkit (HTK) is used. The proposed front-ends were tested in mismatched conditions.

Recognition results in terms of WER and their 95% confidence intervals are shown in Figure 4.6.

The confidence intervals are computed following [104] according to the next expresion:

$$\frac{\Delta}{2} = 1.96\sqrt{\frac{p(100-p)}{n}}, \qquad (4.8)$$

where p is the WER, and n is the number of words in the dataset. Obtaining the 95% confidence intervals show in the experiments as: $[p - \frac{\Delta}{2}, p + \frac{\Delta}{2}]$.

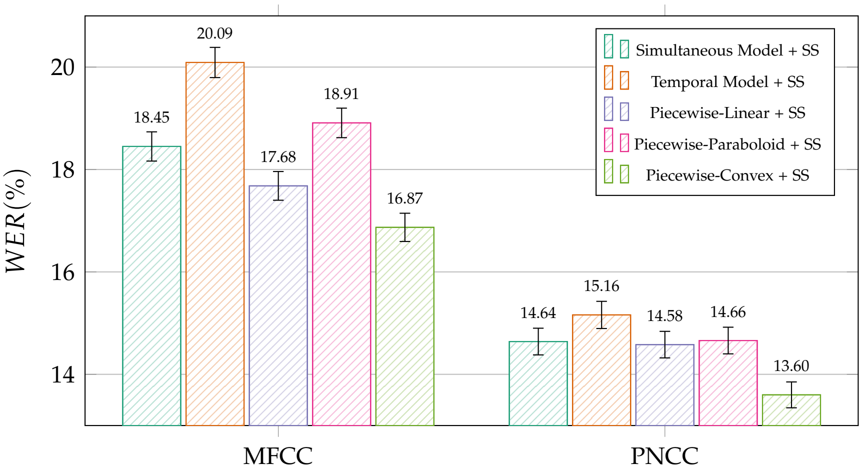


Figure 4.7: Recognition results in terms of WER(%) and 95% confidence intervals using the Aurora 2 dataset (averaged over all test sets) for the different structuring elements in combination with SS.

These results correspond to several experiments carried out to study the impact of MF with the SE described in Section 4.4 applied in isolation or in combination with SS and employing mel-frequency or power-normalized based spectro-temporal representations (labeled respectively, as MFCC and PNCC).

We consider first the influence of MF in the ASR system performance with different SEs. From Figure 4.6, applying MF only in the frequency domain to simulate simultaneous masking (results labeled as *Simultaneous Model*) produces better results than applying MF only in the temporal domain (results labeled as *Temporal Model*). The comparison between the three three-dimensional SE considered (*piecewise-linear*, *piecewise-paraboloid* and *piecewise-convex*) indicates that the last one outperforms the other 3D models as well as the baseline and the simultaneous and the temporal models for both spectro-temporal representations and therefore was chosen for the subsequent experiments. In particular, the application of MF with the *piecewise-convex* SE over noisy spectrograms produces relative error reductions of 16.5% for MFCC and 9.7% for PNCC with respect to the corresponding baselines, both statistically significant. This suggests that the proposed model is suitable for representing the robust behavior of the HAS in the presence of noise.

Furthermore, Figure 4.7 presents the results obtained employing the different proposed SE in combination with SS: the piecewise-convex SE obtained the best performance using either MFCC or PNCC. For these reasons, from now on the rest of the thesis, MF will refer to morphology filtering with the *piecewise-convex* SE.

Secondly, we also investigated combinations of SS and MF. As expected, for both spectro-temporal representations, SS with no MF clearly outperforms the corresponding baselines. For both, MFCC and PNCC, the joint use of SS and MF improves the recognition rates obtained with SS in a statistically significant manner. In particular, for MFCC the relative error reduction achieved by MF+SS with respect to SS is 10.7% and 24.9% with respect to the baseline. The relative error reduction obtained with PNCC is 6.2% and 18.7% related to using only SS and the baseline, respectively. These results show that a positive synergy exists between the SS and MF techniques. Other spectral suppression methods like Minimum Mean Square Error (MMSE) [39] and Wiener [38] filtering were also initially tested but yielded worse results than SS in conjunction with PNCC.

Third, the comparison of both spectro-temporal representations shows that the different versions of features based on PNCC (baseline, SS, MF, SS+MF) achieve in all cases better recognition rates than the corresponding features based on MFCC. The best combination of PNCC (MF+SS) produces a relative error reduction of 19.4% with respect to the best combination of MFCC (MF+SS) and of 39.5% with respect to the MFCC baseline. Also, it is worth noting that PNCC in isolation obtains similar results than the best combination of MFCC-based features (MF+SS).

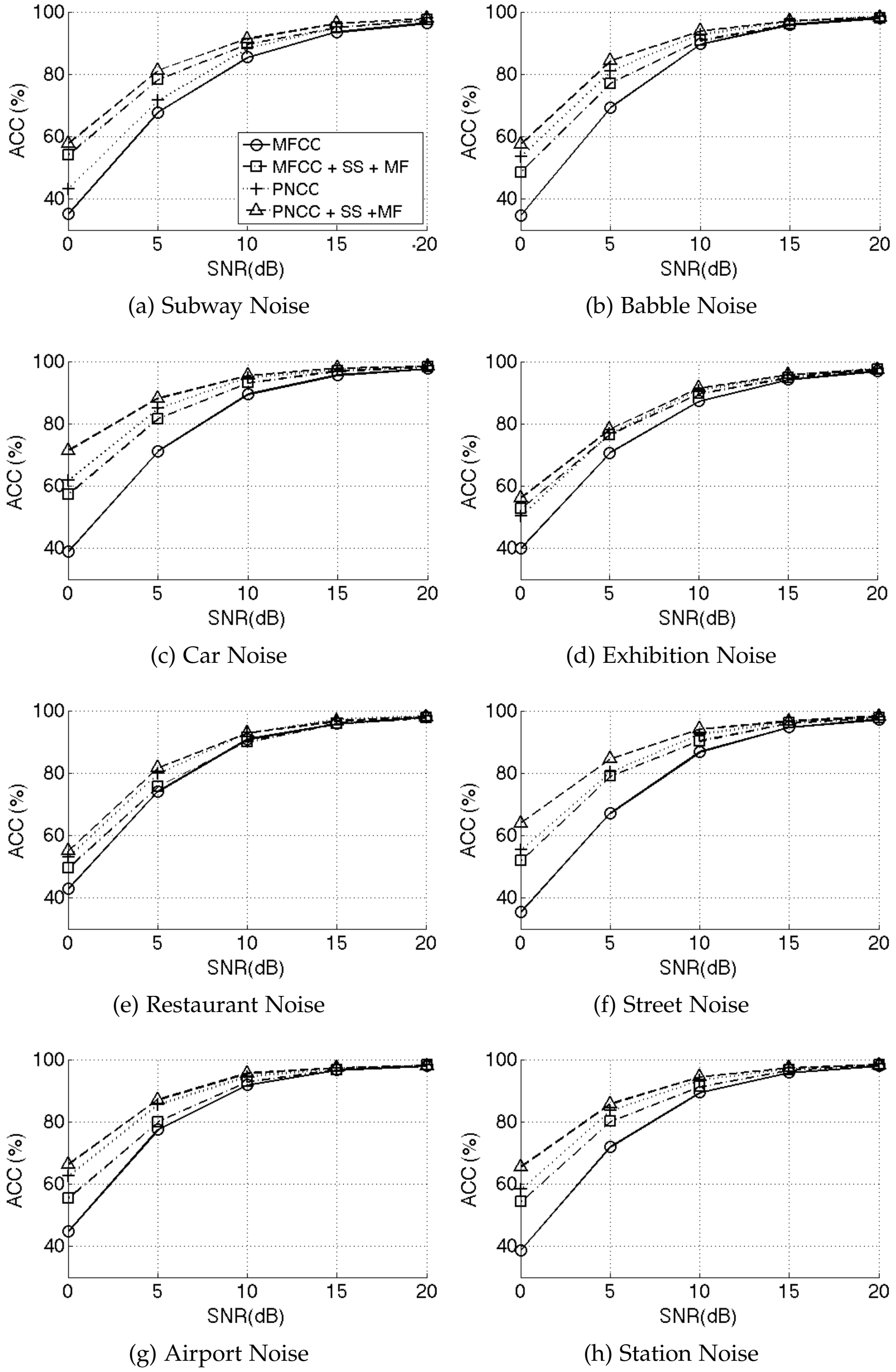
Figure 4.8 and Figure 4.9 show the recognition Accuracy (ACC) ($ACC(\%) = 100 - WER(\%)$) for each type of noise and Signal-to-Noise Ratio (SNR). For brevity, only the results obtained by the baselines and MF+SS are shown in these figures. It can be observed that the PNCC (MF+SS) method achieves the best performance in almost every noise and SNR conditions. In some cases the MFCC method (MF+SS) achieves similar results to PNCC as can be gleaned from Figures 4.8d, 4.8f, 4.8h, Figure 4.9a and 4.9b. Results in the presence of convolutional noise presented in Figure 4.9 show no degradation compared to the results obtained in the presence of additive noise only.

To conclude, we have achieve a better relative error reduction in the Aurora 2 database than some other state-of-the-art techniques; for instance, two dimensional (2D)-Gabor features based on power-normalized spectrograms achieve a relative error reduction of only 7.04% compared to PNCC using a Hidden Markov Model (HMM) backend [85].

4.5.3 *Isolet dataset*

In this section, we present the experiments carried out on the Isolet database presented in Section 2.6.5

The experiments were performed using the Isolet testbed described in [52], where an Artificial Neural Network-Hidden Markov Model (ANN-HMM) hybrid system is employed, using a context of 5 frames to yield an input dimension of 195 and only one hidden layer is



(a) Subway Noise

(b) Babble Noise

(c) Car Noise

(d) Exhibition Noise

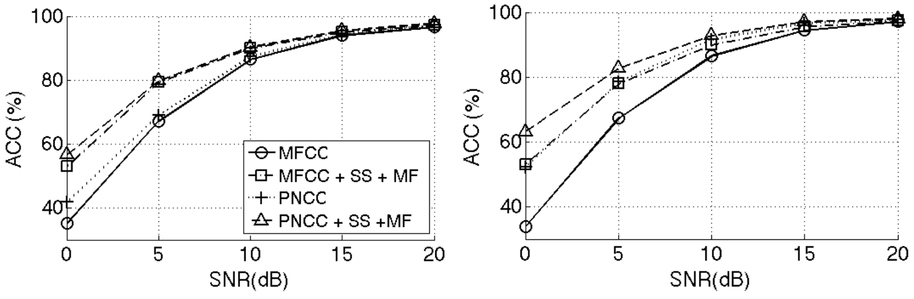
(e) Restaurant Noise

(f) Street Noise

(g) Airport Noise

(h) Station Noise

Figure 4.8: Recognition results obtained under different additive noise conditions in terms of ACC(%) using the Aurora 2 dataset.



(a) Subway (MIRS) Noise

(b) Street (MIRS) Noise

Figure 4.9: Recognition results obtained under different convolutional noise conditions in terms of ACC(%) using the Aurora 2 dataset.

employed. This experiments were carried out to test our features in different back-ends.

This system was tested under *mismatched* conditions where the system is trained using clean speech and the test set consists of speech contaminated with a balanced combination of the previously mentioned noises at several SNRs. A 5-fold cross-validation procedure was used to improve the statistical significance of the results. The corresponding recognition results in terms of WER and their 95% confidence intervals are shown in Figure 4.10.

As can be depicted form Figure 4.10, similar conclusions to those using the Aurora 2 dataset can be drawn. First, SS alone (without MF) clearly outperforms the corresponding baselines for both types of spectro-temporal representation (MFCC and PNCC-based). Second, the application of MF increases the recognition rates with respect to the corresponding baselines for both representations. Third, the joint use of SS and MF improves the recognition rates obtained with SS in a statistically significant manner. Finally, the PNCC features (PNCC baseline, SS, MF, SS+MF) achieve in all cases better recognition rates than the corresponding features based on MFCC.

With this set of experiments we have shown that the proposed front-ends achieve also good results in hybrid ASR systems. Besides, in comparison with our previous work over the Isolet database [94], it can be observed that we have successfully improved the design of the three-dimensional SE by means of the incorporation of perceptual facts, yielding better results.

4.5.4 *WSJ0 dataset*

In this section, we present the experiments carried out on the WSJ0 database presented in Section 2.6.3

The experiments were performed using the HTK recipe described in [50], employing a tri-gram language model with 5k vocabulary size

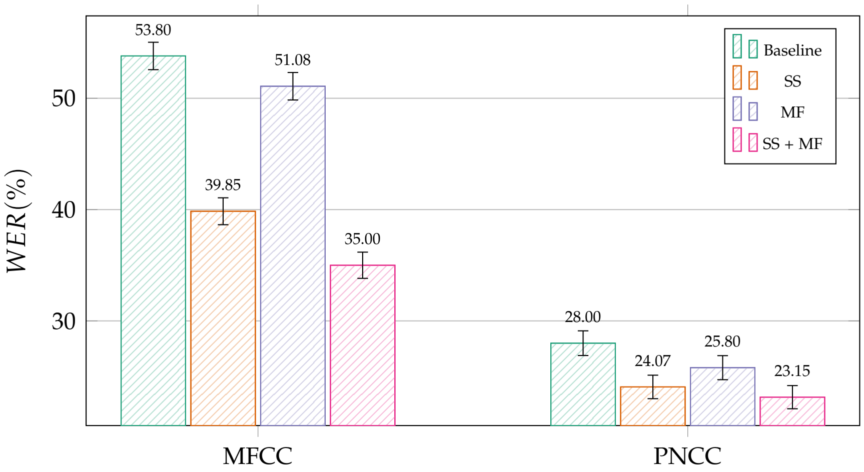


Figure 4.10: Recognition results in terms of WER(%) and 95% confidence intervals using the Isolet dataset (averaged over all the noises and SNR, tested in mismatched conditions).

and the Carnegie Mellon University (CMU) pronunciation dictionary. We use the SI-84 training set and the 5K test set contaminated with the same standard testing environments as [33]. The various front ends were tested on versions of the test set to which the noises were added to the corresponding clean speech at different SNRs using the Filter and Noise-adding Tool (FANT) [25] with G.712 filtering. All the noise tests are evaluated in mismatched conditions.

Recognition results in terms of WER and their 95% confidence intervals are shown in Figure 4.11. These results correspond to the average over all the noises and SNR conditions outlined above. The performances of our systems on clean speech employing the WSJ0 5K test set are: 5.36% WER for MFCC and 6.67% WER for PNCC.

Figure 4.12 shows the recognition ACC for each type of noise and SNR. For the sake of brevity only the results obtained by the baselines and MF+SS are shown in these figures.

Figure 4.11 shows that: (1) The PNCC spectral representation baseline clearly outperforms the corresponding MFCC baseline; (2) the application of MF improves the baseline recognition rates but not in a significant way for the PNCC case; (3) the joint use of SS and MF improves the recognition rates obtained with SS and with the baseline in a statistically significant manner for both representations; (4) the PNCC (baseline, SS, MF, SS+MF) achieve in all cases better recognition rates than the corresponding features based on MFCC, and (5) the improvements in the WSJ0 dataset are lower than the Aurora 2 and Isolet datasets. We suggest that this reduction is due to the larger size of the database and the influence of the language model in the acoustic decoding process.

Also note that, from Figure 4.12, the PNCC (MF+SS) method achieves the best performance in every noise and SNR conditions. The improve-

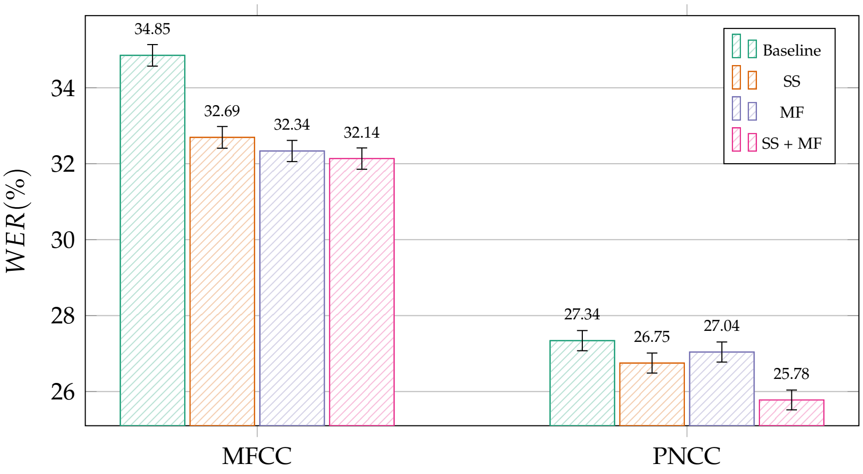
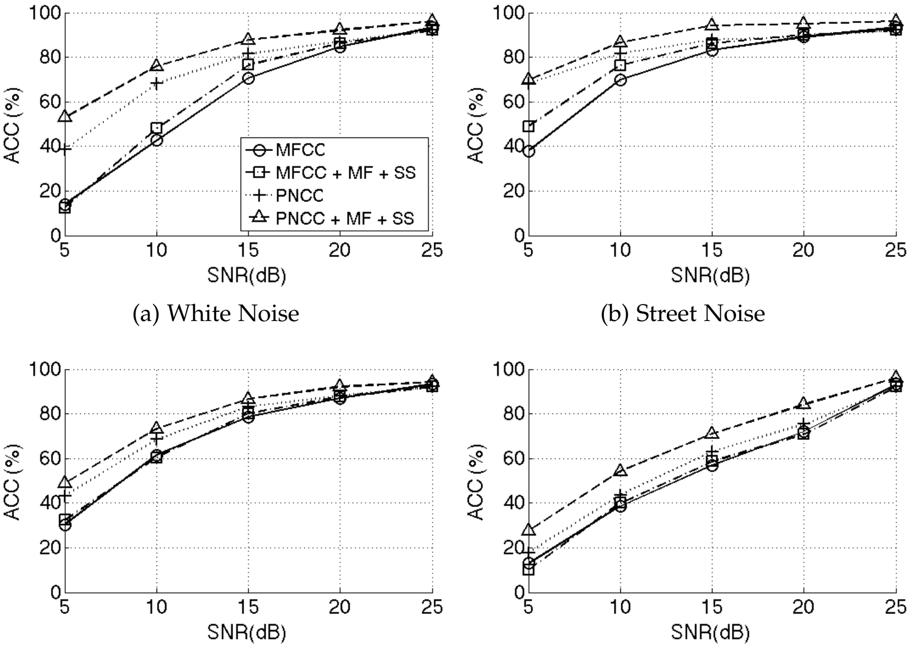


Figure 4.11: Recognition results in terms of WER(%) and 95% confidence intervals using a noisy version of the WSJ0 dataset (averaged over all the noises and SNR).



(a) White Noise

(b) Street Noise

(c) Music Noise

(d) Speaker Noise

Figure 4.12: Recognition results obtained under different additive noise conditions in terms of ACC(%) using the WSJ0 dataset.

Method	Time (ms)	% from Baseline
MFCC	19.66	–
MFCC + SS	26.98	37.23 %
MFCC + MF	21.84	11.80 %
MFCC + MF + SS	28.82	46.59 %
PNCC	67.93	–
PNCC + SS	85.69	26.14 %
PNCC + MF	69.45	2.23 %
PNCC + MF + SS	87.06	28.16 %

Table 4.1: Average runtime per utterance for the different methods over all test sets on the Aurora 2 dataset.

ment in white noise in Figure 4.12a, and speaker noise in Figure 4.12d conditions are particularly worth noticing, since the proposed method clearly outperforms the PNCC baseline.

4.5.5 *Computational complexity*

Table 4.1 shows a comparison of the runtimes for the different methods under several conditions (clean and noisy speech), using a workstation with 3.4 GHz Intel Core i7 processor. The running times were obtained by averaging each of the utterances over all testing sets on the Aurora 2 dataset. The extra time added by MF is relatively low for either MFCC or PNCC. It is worth noting that the time spent by MFCC + MF + SS is below the PNCC baseline, despite obtaining similar results in almost every noisy condition.

4.6 conclusions

In this chapter we present an enhanced, perceptually-motivated SE for morphological filtering of speech that models the complexity of HAS masking properties. Well-known empirical data in either temporal or frequency domains were interpolated to produce a three-dimensional SE for morphological filtering. We imposed a smoothness constraint we found more suited for our hypothesis that the morphological closing operation produces a convexification of the spectro-temporal envelope of speech that models the masking properties of the HAS.

Despite ingrained intuitions that this imitation of auditory masking degrades the quality of the extracted features producing a *blurring* effect, the results that we have obtained indicate that it could be in fact a sophisticated mechanism for selecting the most important parts of the spectrum from an intelligibility point of view, taking away irrelevant information and emphasizing the most robust parts of the spectrum.

The application of morphological processing with this SE in conjunction with the Power-Normalized spectro-temporal representation produces a significant increase in recognition rates in Aurora 2, Isolet and a noisy version of the WSJ0 dataset. Also the results show that our method improves the recognition rates in both hybrid and traditional Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) based back-ends. To reach these results we have tested the combination of PNCC, spectral subtraction and morphological processing.

5 SYNCHRONY-BASED FEATURE EXTRACTION

5.1 INTRODUCTION

In this chapter we discuss various ways of exploiting the temporal patterns of auditory-nerve activity to improve Automatic Speech Recognition (ASR) accuracy. Physiological studies have demonstrated that the response of an auditory-nerve fiber with a low Characteristic Frequency (CF) roughly follows the shape of the input signal at least when the signal amplitude is positive [105].

This *phase-locking* behavior enables the auditory system to compare arrival times of signals to the two ears at low frequencies, which is the basis for the spatial localization of a sound source at these frequencies. While this sort of temporal coding is clearly important for binaural sound localization, it may also play a role in the robust interpretation of signals from individual ears as well.

Much of our own work in this area is motivated by physiological findings by Sachs and Young [71] which showed that the Average Localized Synchrony Rate (ALSR) that is derived from the nerve firing times is much more robust to changes in intensity of vowel-like sounds than the corresponding Mean Rate (MR) of responses as a function of CF.

These results suggest that the timing information associated with the response to low-frequency components of a signal can be substantially more robust to variations in intensity (and potentially various other types of signal variability) than the mean rate of the neural response. Most conventional feature extraction schemes (such as Mel-Frequency Cepstral Coefficients (MFCC) and Perceptually-based Linear Prediction (PLP) coefficients) are based on the short-time energy in each frequency band, which is associated with mean rate rather than synchrony.

The remainder of this chapter is organized as follows: Section 5.2 explains the synchrony effect, Section 5.3 briefly reviews the state of the art that has motivated our formulation, Section 5.4 describes our synchrony measurements and feature extraction procedures in some detail, Section 5.5 describes our experimental results, and finally Section 5.6 summarizes our findings.

5.2 SYNCHRONY RESPONSE

The auditory nerve, as remarked in Chapter 3, transmits the electrical impulses generated by the hair cells inside cochlea to the brain. It is known that the auditory nerve fibers are more sensitive to the

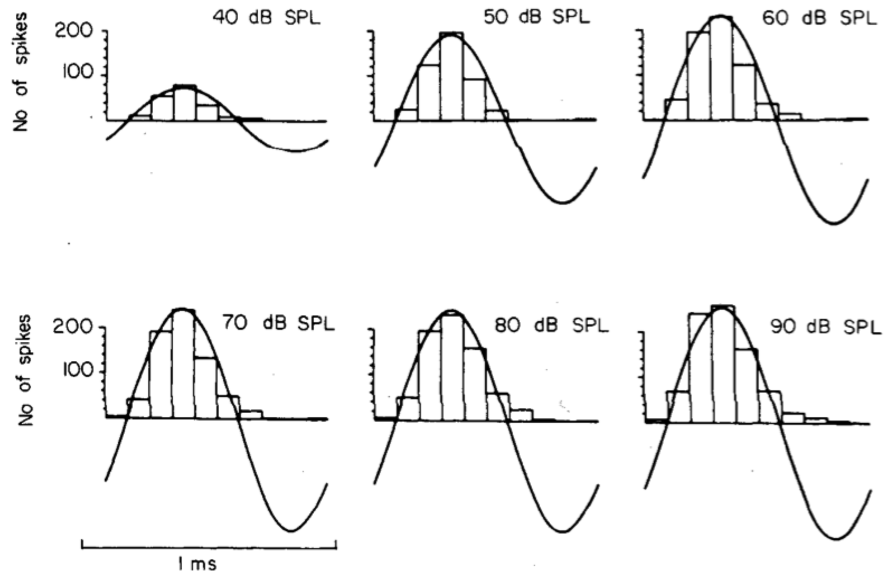


Figure 5.1: Period histograms of the auditory nerve fiber activity for a squirrel monkey in presence of a 1.1kHz tone at different intensities. Reproduced with permissions from [106].

tuned frequencies of the hair cells where they are connected. This frequency is known as the Characteristic Frequency (CF) of the fiber. This mechanism is accepted to be involved in how humans perceive the sound location and may be part of the Human Auditory System (HAS) robustness mechanisms.

The auditory nerve activity has been widely studied showing two relevant behaviors: first, that when the intensity level is below the threshold level, the auditory nerve fibers are activated randomly and second, that as the intensity of the excitation is increased the activity of the auditory nerve fibers follows the shape of the signal in the positive part until saturation. This effect is more notable in the fibers whose characteristic frequency is low. An illustration of this behavior can be seen in Figure 5.1 where the auditory nerve fiber activity of a squirrel monkey in presence of a pure tone is represented as a period histogram [106]. The bars represent the number of spikes generated during the interval given by the bar width.

This effect is known as *phase-locking* or *synchrony-effect* and allows the human auditory system to obtain the spacial locations for low frequency signals by comparing the arrival times of both ears, known as Interaural Time Difference (ITD), lying the foundations for binaural sound location. It is worth noting that when the characteristic frequency of auditory nerve fibers increases, the capacity of tracking the signals disappears, but [107] shows that the activations follow the shape of the envelope of the signal, making the listeners sensitive to some extent to ITDs in high-frequency sounds.

Some authors have hypothesized that this affects how the signals are interpreted by obtaining a representation with increased robustness mainly for low frequencies. The most relevant work in this regard is presented in [71] where the authors measure the auditory nerve activity in cats using different vowels as stimulus.

By knowing the presented vowel and the CF of the measured fibers, the ALSR and the MR response can be computed. First, the period histogram is computed for each fiber, estimating the the instantaneous discharge rate as a function of time through one fundamental period of the vowel, averaged over the analysis period, computed using 64 or 128 bins per cycle.

Second, he discrete time Fourier transform is applied over the period histogram, $r_l[n]$ for the fiber l. Then the period histogram of each fiber can then represented as:

$$r_l[n] = R_0 + 2 \sum_{k=1}^{N/2-1} R_{k,l} \cos\left(\frac{2\pi k}{N} n + \theta_{k,l}\right), \tag{5.1}$$

where N is the length of the histogram and $R_{k,l}$ indicates the magnitude of the k-th Fourier coefficient for the fiber l. It Is worth noting that every other Fourier magnitude correspond to each of the vowel harmonics as the Fourier transforms is computed from histograms containing two cycles of the vowel. An example of period histogram and its Fourier transform magnitude can be seen in Figure 5.2.

Finally, the MR is simply the mean activity for each fiber over time while the ALSR is defined as:

$$ALSR_k = \frac{1}{M_k} \sum_{l \in C_k} R_{k,l}, \tag{5.2}$$

where $R_{k,l}$ is the magnitude of the k-th Fourier component of the period histograms of the nerve fiber l, C_k is the set of fibers with characteristic frequency between $\sqrt{2}/2kf_0$ and $\sqrt{2}kf_0$ being f_0 the fundamental frequency of the vowel and M_k the number of fibers satisfying this condition.

As can be interpreted by this formula, the ALSR describes how the auditory nerve activity of a fiber with a known characteristic frequency is synchronized with the nearest harmonic of the fundamental frequency of the vowel.

Figure 5.3 shows a comparison of the auditory nerve activity measured in terms of MR and ALSR. In Figure 5.3a the original spectrum of the stimulus vowel /e/ is presented, Figure 5.3b shows the mean rate response at different sound levels, while in Figure 5.3c the ALSR is shown for the same vowel and intensities.

It can be seen that while the MR varies heavily when the intensity increases, the ALSR shape remains constant, showing that the ALSR is more robust to changes in intensity than the mean rate of the firing activity for vowel sounds.

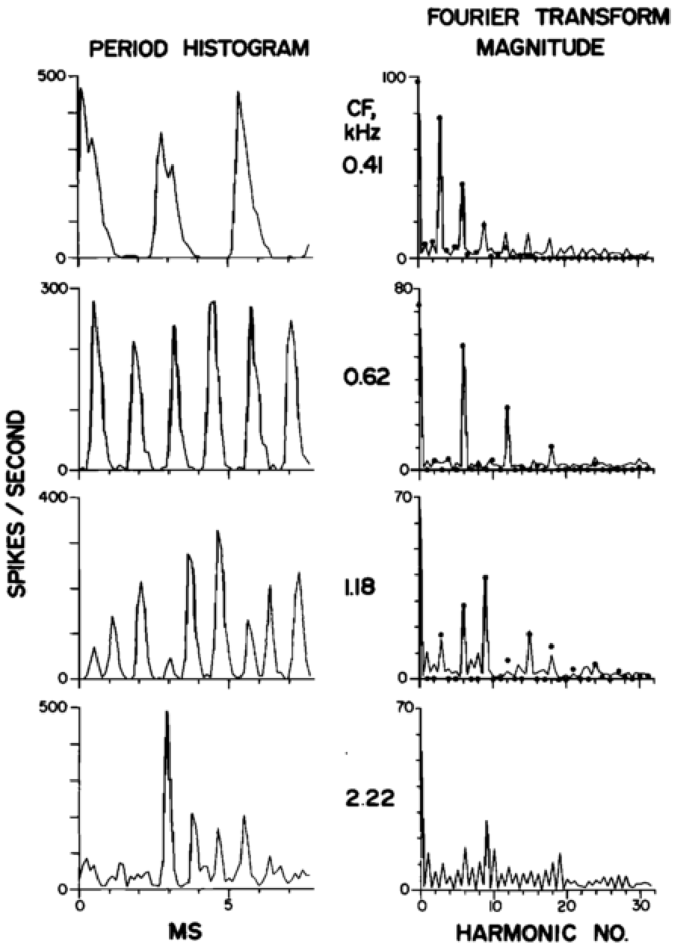
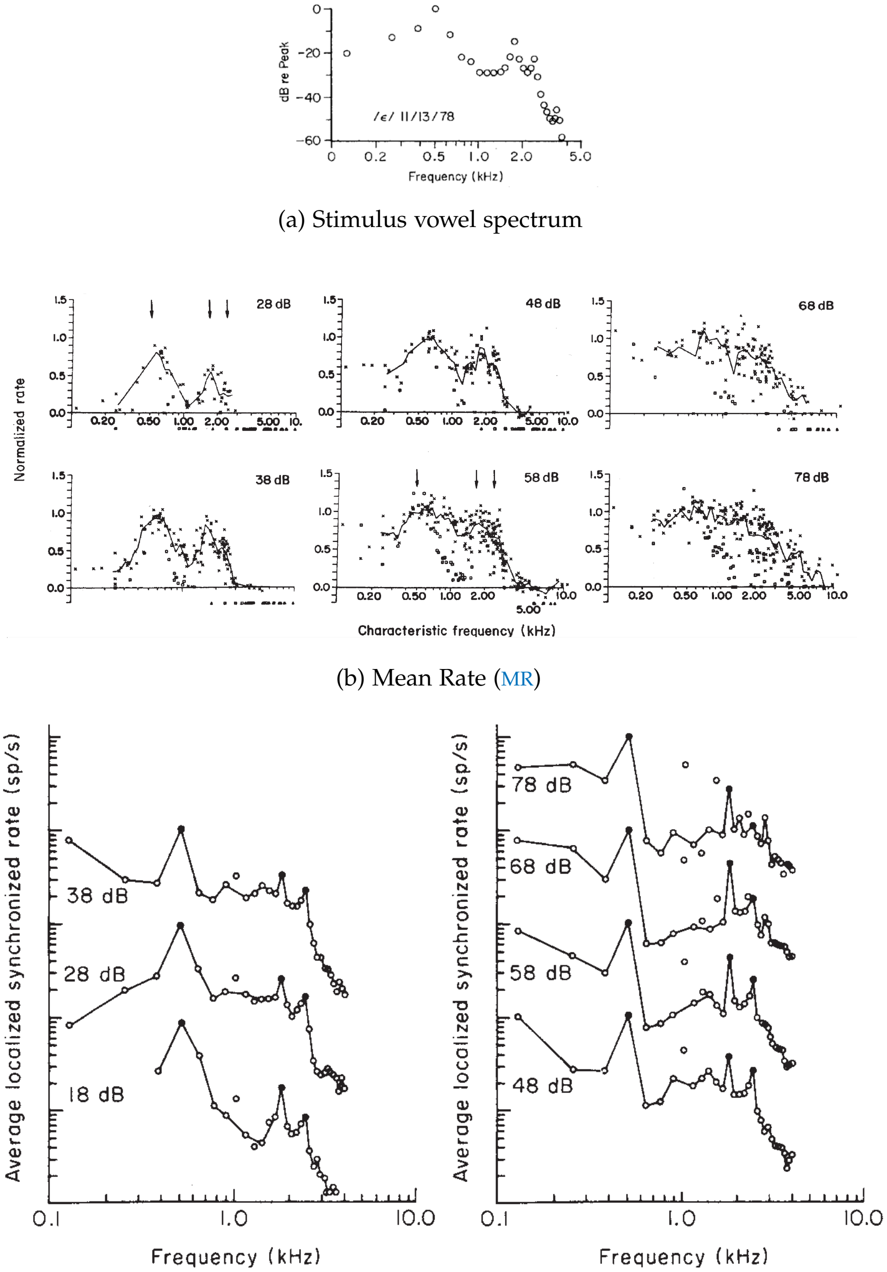


Figure 5.2: Period histogram and Fourier decomposition for four different fibers using the vowel /a/ as a stimulus. The CF of each fiber is shown in the center. Reproduced with permission from [71].



(a) Stimulus vowel spectrum

(b) Mean Rate (MR)

(c) Average Localized Synchrony Rate (ALSR)

Figure 5.3: Original spectrum, Mean Rate (MR) response and Average Localized Synchrony Rate (ALSR) for a set of fibers in presence of a synthetic vowel /e/ for different sound levels. Reproduced with permission from [71].

These results can be interpreted as the synchrony effect or the timing information in low-frequencies can have an impact in the robustness of the human auditory systems to variations of intensity and maybe to others types of degradation.

It is worth noting that traditional feature extraction algorithms, as MFCC, PLP or Power Normalized Cepstral Coefficients (PNCC), are based on the short-time energy computation of each of the band filters rather than on the synchrony or temporal information of the physiological responses of the auditory nerve. The short-time energy can be interpreted as the mean response of the auditory nerve, and therefore, trying to incorporate the temporal patterns of the signal to the feature extraction process could make the ASR systems more robust to changes in intensity and other variations for low frequencies.

5.3 RELATED WORK ON FEATURES BASED ON SYNCHRONY

In this section we very briefly review selected prior studies that describe techniques that have been proposed to develop a *synchrony spectrum* that reflects the temporal patterns of the auditory-nerve response to signals as a function of frequency.

One of the first was the Seneff's auditory model [66]. As presented in Section 3.3, it has two parallel outputs, one that approximated the instantaneous mean rate of firing of the auditory nerve and a second that measured the synchrony in response to the incoming signals. The second operation was called a Generalized Synchrony Detector (GSD), motivated by the ALSR measure of [71]. The GSD simply compares the simulated Inner Hair Cell (IHC) outputs with themselves delayed by the reciprocal of the center frequency of the cell.

A second early formulation was Ghitza's Ensemble Interval Histogram (EIH) model [67], which develops synchrony information by recording level crossings of previous stage over a set of seven logarithmically-spaced thresholds over the dynamic range of each channel.

In subsequent years the approaches of Seneff and Ghitza have been elaborated upon, and other techniques have been introduced as well. For example [78] proposed a simple but useful extension of the Seneff GSD model that develops a synchrony spectrum by simply averaging the responses of several GSDs tuned to the same frequency using inputs from bandpass filters with CFs in a small neighborhood about a central frequency. In [77] a type of processing called Zero Crossing Peak Amplitude (ZCPA) is proposed, which develops histograms of time spans between consecutive zero crossings weighted by the amplitude of the peak between them.

A synchrony-based extraction of spectral contours is presented in [75] by computing the Fourier transform of the envelope of the auditory-nerve response in each channel (similar to ALSR computation) after bandpass filtering to reduce the impact of spurious frequencies in the response produced by cochlear nonlinearities. This produces a high-resolution spectral representation at low frequencies for which the auditory nerve is synchronized to the input up to about 2.2 kHz, and that includes the effects of the nonlinearities in peripheral auditory processing. Using the auditory model of Zhang *et al.* [74], [75] show that the use of the synchrony processing at low frequencies provided a modest improvement compared to the auditory model with mean-rate processing, and a large improvement compared to baseline MFCC processing.

Other recent features motivated by synchrony include Synchronous Damped Oscillator Cepstral Coefficients (SYDOCC) features [108] and Locally Normalized Cepstral Coefficients (LNCC) features [109].

5.4 synchrony feature extraction

In this section we describe the various approaches to synchrony extraction propose in this thesis, focusing on the generalized synchrony detector proposed by Seneff [66], along with a second approach that combines the ALSR proposed by Young and Sachs [71] with MR information. We also discuss the benefit that is obtained when synchrony extraction is preceded by a noise cancellation mechanism.

5.4.1 *Application of the Seneff auditory model and Generalized Synchrony Detector (GSD)*

The Seneff auditory model [66] is well known and has received a great deal of attention in the literature. It compress a model for the auditory-nerve response with two outputs, one representing mean rate and another representing synchrony.

Synchrony is estimated via the GSD, it is based on a nonlinear statistic related to the autocorrelation of the Stage II output at a delay equal to the period associated with the CF of each filter. The GSD compares the putative instantaneous output of the hair cells in each channel with itself delayed by the reciprocal of the center frequency in each channel; the short-time averages of the sums and differences of these two functions are divided one by another. A threshold is introduced to suppress the response to low-intensity signals and the resulting quotient is passed through a saturating half-wave rectifier to limit the magnitude of the predicted synchrony.

The GSD seeks a clean spectral representation that preserves the prominent peaks at the formant frequencies while reducing glottal excitation components. Following the definition of the synchrony when a peak at a particular frequency is present, it will be shown as a periodicity in the output of the Sennef Stage II, and the GSD tuned to the closer CF will detect it.

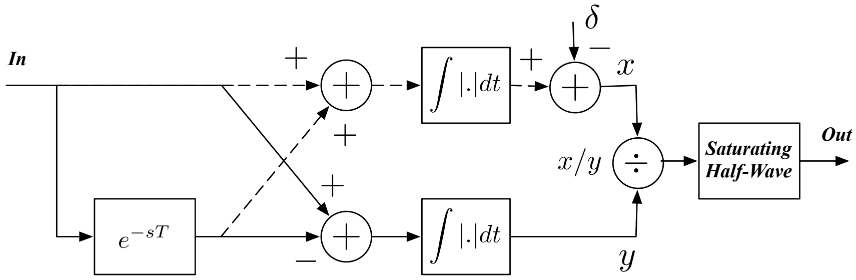


Figure 5.4: Comparison of the elements of the original GSD by Seneff and the MGSD used in this thesis. The connections denoted by the broken lines are eliminated from the original GSD in the MGSD. T denotes the reciprocal of the center frequency in each channel.

The GSD has mainly three advantages:

1. It avoids detecting the frequency of the nearest harmonic of the fundamental frequency of the vowel, as in the case of the ALSR.

2. The ratio of the numerator and denominator can be considered as an energy normalization technique. The autor of [66] considers that the ratio attenuate the fluctuations in the response due to the glottal excitation components which can be considered noise.

3. The computational complexity of the GSD is very low.

We have empirically established that performance can be improved by modifying the GSD detector by computing only the inverse of the difference between the original input signal and the original signal delayed by the period of the frequency to which the GSD is tuned. This modification is similar to the short time autocorrelation function [35] that display peaks at multiples of the period of a periodic signal. Figure 5.4 compares the structure of the original and modified GSD calculation; the modified GSD algorithm eliminates the connections denoted by the broken lines. This modification contradicts that the original GSD attenuate the fluctuations in the response, this can be due to the fact that Mean and Variance Normalization (MVN) was applied in the subsequent steps or because the glottal fluctuations contribute to increase the recognition rates.

To obtain the features the mean energy is computed over each of the outputs with a frame period of 10 ms and 25 ms window length, follow by a Discrete Cosine Transform (DCT), obtaining 13 components. From now, the features obtained using the modified GSD are denoted as Modified Generalized Synchrony Detector (MGSD).

5.4.2 *Synchrony estimation based on Average Localized Synchrony Rate (ALSR)*

We also estimated the synchronized response using a variation of the ALSR measurement of the putative auditory-nerve activity, as proposed by Young and Sachs [71].

The ALSR is defined at a specified frequency to be the ratio of the first Fourier component of the response at that frequency divided by the mean firing rate, as described in [71] and in Section 5.2. These responses are averaged over a range of one octave, again as described in [71].

In our implementation of the ALSR we take the fundamental frequency of the vowel as the CF of the fiber for which we are computing the ALSR since the fundamental frequency of the vowel uttered is normally unknown. The rest of the steps are performed as in the original ALSR: the short-time Fourier transform of the outputs of the auditory filters are averaged across channels, producing a high resolution representation at low frequencies.

This is one of the problems we face when developing a synchrony related feature: the need to choose a frequency to sync. Other approaches like using an estimation of the pitch for syncing may be possible but normally pitch estimations are not robust enough. How to choose a proper frequency to sync is still an open question.

Because synchrony in realistic neural responses disappears (at least for fine structure) at frequencies above 1-2 kHz, we used synchrony information below 1000 Hz, and mean rate of firing above, with a linear transition between the two over a range of 300 to 1200 Hz. Finally, the synchrony and mean rate information are combined and decorrelated by using a DCT, that produces a set of coefficients in the cepstral domain and also removes the horizontal striations presented in high resolution spectrograms. From now on, these features will be denoted as Modified Average Localized Synchrony Rate (MALSR).

5.4.3 *Auditory model employed to extract the synchrony information*

In order to obtain the later proposed synchrony features: MGSD and MALSR, a model of the auditory-nerve activity is required to obtain the response of the IHCs. For the computation of these features we considered two distinct models of auditory-nerve activity: the functional model proposed by Seneff [66] and that of Zhang *et al.* [74]. Both models are described in Section 3.3.

The model of Zhang *et al.* is much more detailed than the Seneff auditory model and much more computationally complex as well. However, we did not achieve much greater recognition accuracy using the complex auditory model as we expected and therefore, the results described in the next section were all obtained using the auditory model described by Seneff [66].

To compute the MGSD and MALSR features we used a simplified implementation of the Stages I and II (see Section 3.3) of the Seneff auditory model based on the Slaney auditory toolbox [110], where the amplitude of the input signals are adjusted per utterance to maintain a constant power as the auditory model is highly nonlinear.

5.4.4 *Noise removal before MGSD processing*

We noted in our original experiments that recognition accuracy using the MGSD or MALSR features could be improved through the use of a noise cancellation mechanism prior to the extraction of synchrony. In our experiments the performance improvements obtained by the MALSR features when used with noise removal techniques are much lower than the MGSD. Therefore, in the following only the MGSD features will be tested in conjunction with the noise cancellation mechanisms.

We considered two types of noise-cancellation approaches in this work. The first, and simpler, approach was to use a form of conventional Spectral Subtraction (SS) [36] (presented in Section 2.5.4), but on a band-by-band basis after the initial gammatone filtering [63, 111], as summarized in the left portion of Figure 5.5. The features obtained by using the MGSD with SS are denoted from now on as Modified Generalized Synchrony Detector with Spectral Subtraction Noise Reduction (MGSD-SS-NR).

A second approach is to integrate the synchrony effects into the PNCC. In order to perform the integration we develop a noise removal technique that incorporates the nonlinear Asymmetric Noise Suppression (ANS) components of PNCC coefficients [12, 33]. In brief, the speech signal is passed through most of the steps of PNCC processing in order to remove the noise components, and then the audio signal is recovered using spectral reshaping. The enhanced audio signal is then passed through the auditory model front end with the MGSD, as summarized in the right portion of Figure 5.5.

More specifically, this process is accomplished as follows: First, we compute the PNCC until the mean power normalization step (see Figure 3.6), retaining the original phase and modifying only the magnitude spectrum.

Then for each time-frequency bin, following the notation of the PNCC (Section 3.4.1), we obtain the weighting coefficient $w[m,l]$ for the m-th frame and l-th frequency band as a ratio of the processed power $T[m,l]$ (the output of the medium-time and short-time PNCC processing) to the original power $P[m,l]$:

$$w[m,l] = \frac{T[m,l]}{P[m,l]}. \tag{5.3}$$

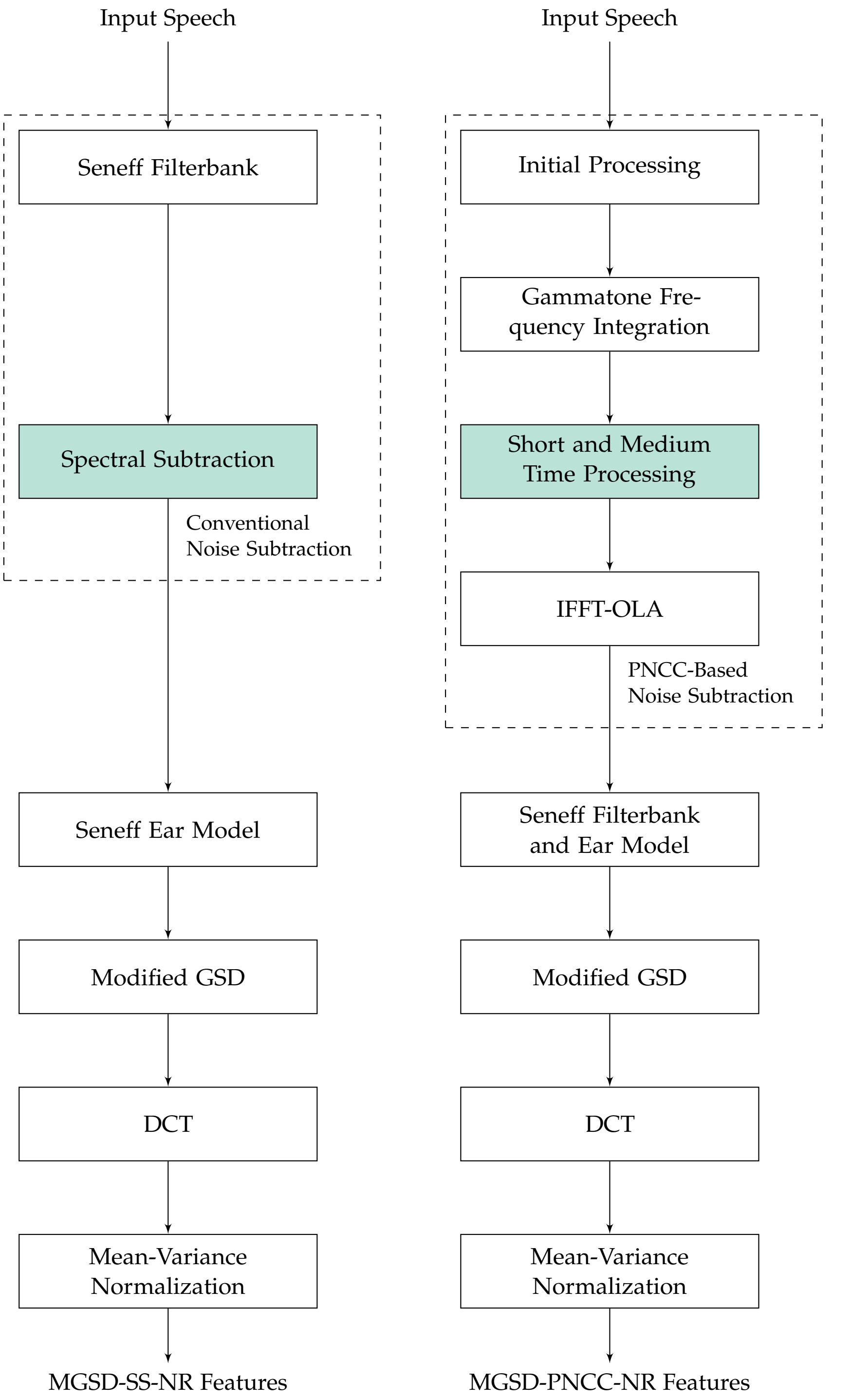

Input Speech
Seneff Filterbank
Spectral Subtraction
Conventional Noise Subtraction
Seneff Ear Model
Modified GSD
DCT
Mean-Variance Normalization
MGSD-SS-NR Features
Input Speech
Initial Processing
Gammatone Frequency Integration
Short and Medium Time Processing
IFFT-OLA
PNCC-Based Noise Subtraction
Seneff Filterbank and Ear Model
Modified GSD
DCT
Mean-Variance Normalization
MGSD-PNCC-NR Features

Figure 5.5: Block diagram comparing two ways of realizing noise reduction prior to the GSD algorithm, using subband spectral subtraction and PNCC-based noise subtraction. The shaded blocks indicates the major differences between the two approaches.

Each of the channels is associated with $H_l(e^{j\omega_k})$, the frequency response of one of a set of gammatone filters. To obtain the final spectral weights $\mu[m,k]$ we apply spectral reshaping [112] using the above weights $w[m,l]$ according to the expression:

$$\mu[m,k] = \frac{\sum_{l=0}^{L-1} w[m,l] |H_l\left(e^{jw_k}\right)|}{\sum_{l=0}^{L-1} |H_l\left(e^{jw_k}\right)|}, \quad 0 \le k \le \frac{N}{2}, 0 \le l \le L-1, \tag{5.4}$$

where L is the number of filters, in our case $L = 40$, k is the discrete frequency index and N is the number of points of the Short-Time Fourier Transform (STFT). Given the $\mu[m,k]$ for $0 \le k \le \frac{N}{2}$, we can obtain the remaining coefficients using Hermitian symmetry.

The reconstructed spectrum $\widetilde{X}\left[m, e^{jw_k}\right]$ is obtained using the original spectrum $X\left[m, e^{jw_k}\right]$ and $\mu[m,k]$ following:

$$\widetilde{X}\left[m, e^{jw_k}\right] = \mu[m,k] X\left[m, e^{jw_k}\right]. \tag{5.5}$$

Finally the enhanced speech $\hat{x}[n]$ is re-synthesized from the reconstructed spectrum $\widetilde{X}\left[m, e^{jw_k}\right]$ by applying an Inverse Fast Fourier Transform (IFFT) and using OverLap Add (OLA) [113] obtaining an undistorted reconstruction due to using 25 ms hamming window and 6.25 ms between frames.

The resulting enhanced speech is processed by the auditory model and the MGSD, as described above. The obtained synchrony spectrum is transformed using a DCT producing a set of coefficients in the cepstral domain. From now on the features obtained using the PNCC noise reduction technique and MGSD will be denoted as Modified Generalized Synchrony Detector with Power-Normalized Cepstral Coefficients Noise Reduction (MGSD-PNCC-NR).

5.5 EXPERIMENTAL RESULTS

Three standard speech corpora were used for our evaluations: RM, WSJ0, and Aurora 4 databases, all of then presented in Section 2.6.

For all the experiments using the RM and WSJ0 datasets, we use Carnegie Mellon University (CMU) Sphinx [114], with the standard recipes where a traditional Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) triphone system is employed. An US English generic bigram language model and CMU pronouncing dictionary are employed. For the Aurora 4 the standard triphone GMM-HMM Kaldi [30] recipe is used with Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) over the features for speaker adaptation.

Since we are concerned primarily with the relative performance of the various signal processing schemes considered, no attempt was made to fine tune the parameters of the CMU Sphinx training and decoder to minimize the absolute error rate.

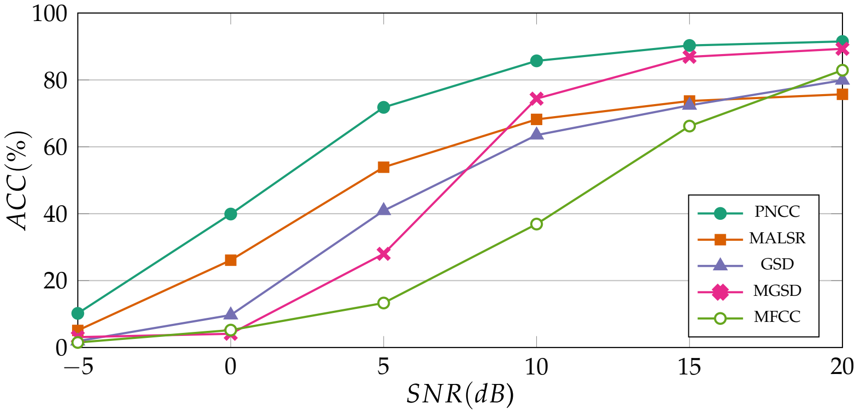


Figure 5.6: Comparison of recognition accuracies for speech corrupted with white noise in RM dataset for each of several proposed synchrony measurements: original GSD, MGSD, and MGSD. A comparison with baseline MFCC and PNCC features is also included.

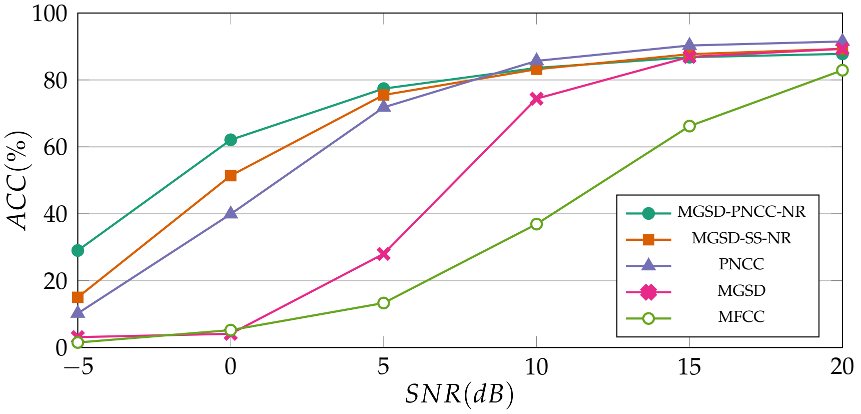


Figure 5.7: Same as Figure 5.6, but comparing the effectiveness of two types of noise subtraction preceding the MGSD processing.

The coefficients obtained with the proposed features, C_0 to C_{12}, were retained together with their corresponding delta (Δ) and acceleration ($\Delta\Delta$) coefficients to yield feature vectors of 39 components. Per utterance mean and variance normalizations were applied to each of the components.

To test the impact of the different methods of synchrony extraction on the robustness of the recognition accuracy for RM and WSJ0 databases, we used the same approach as in the previous chapter, using the four standard testing environments of [33], where the various front-ends were tested on versions of the test set distorted with the four testing environments at different Signal-to-Noise Ratios (SNRs). For Aurora 4 the noise test sets are already defined as can be seen in Section 2.6.2. Most evaluations are performed under mismatched

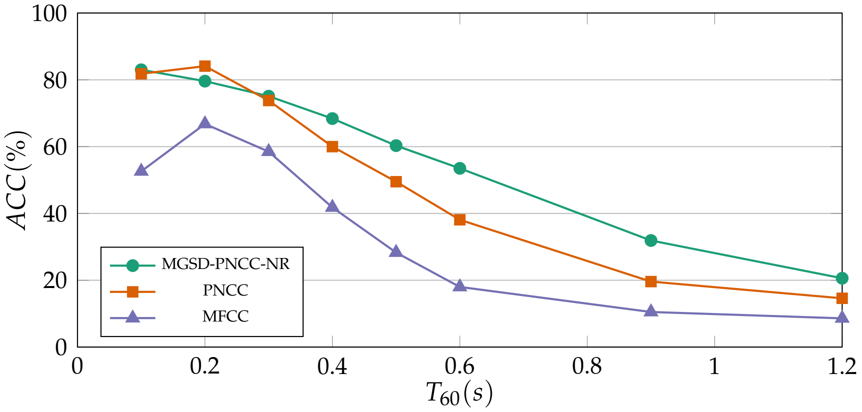


Figure 5.8: Comparison of recognition accuracies for different simulated reverberation times using the RM dataset. The MGSD-PNCC-NR is compared with baseline MFCC and PNCC processing.

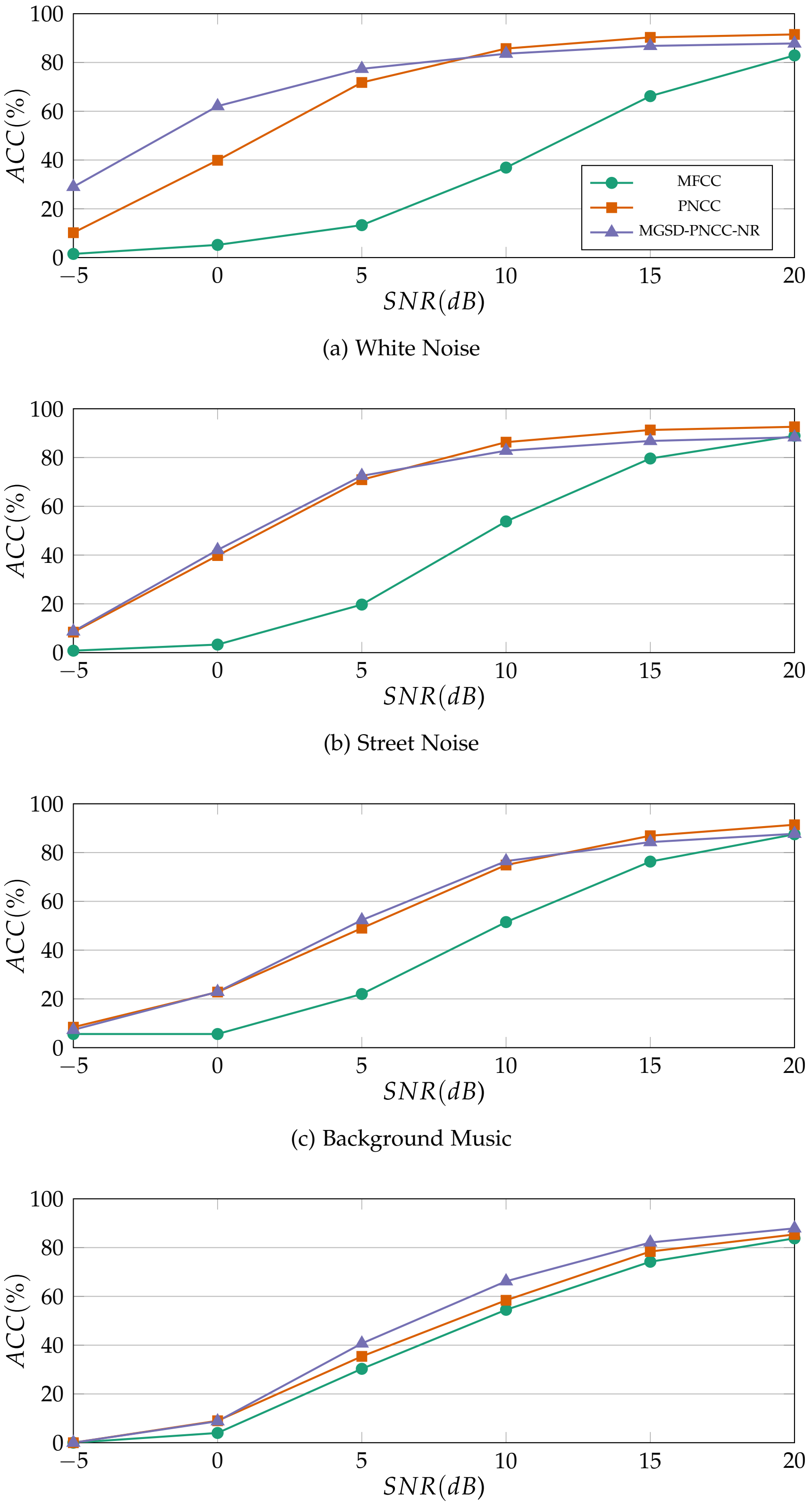
conditions (*i.e.* training on clean speech and testing on degraded speech).

Figure 5.6 compares the results for speech corrupted in white noise at various SNRs for several of the proposed methods using the RM dataset. Specifically, we compare recognition accuracy in percent using the MALSR, GSD, and MGSD methods as described in Sections 5.4.1 and 5.4.2 above, along with baseline MFCC and PNCC features. As can be seen, all synchrony-based measurements outperform the baseline MFCC features, the MGSD outperforms the original GSD at higher SNRs, and the MALSR measure outperforms both GSD measurements at the lower SNRs. Nevertheless, baseline PNCC features outperform all other methods, including the synchrony-based features. Similar results are obtained using the WSJ0 data, although the improvements over MFCC are less dramatic.

Figure 5.7 shows the impact of preprocessing for noise reduction on the effectiveness of the GSD-based features. We note that the combination of GSD features using either noise removal approach now outperforms baseline PNCC features, but that PNCC-based noise subtraction provides substantially greater accuracy than conventional noise subtraction when used in conjunction with the MGSD processing.

Figure 5.8 compares MFCC, PNCC, and MGSD-PNCC-NR in conditions of simulated reverberation, again using the RM database, as a function of reverberation time. Here the RM test set is corrupted by passing the speech signal through a filter with impulse response derived from a room simulation algorithm using the image method [26] at different reverberation times.

For reverberation times greater than 0.3 seconds, the MGSD-PNCC-NR features provides a significant improvement in recognition accuracy, demonstrating that synchrony-based processing is useful in reverberant as well as noisy environments.



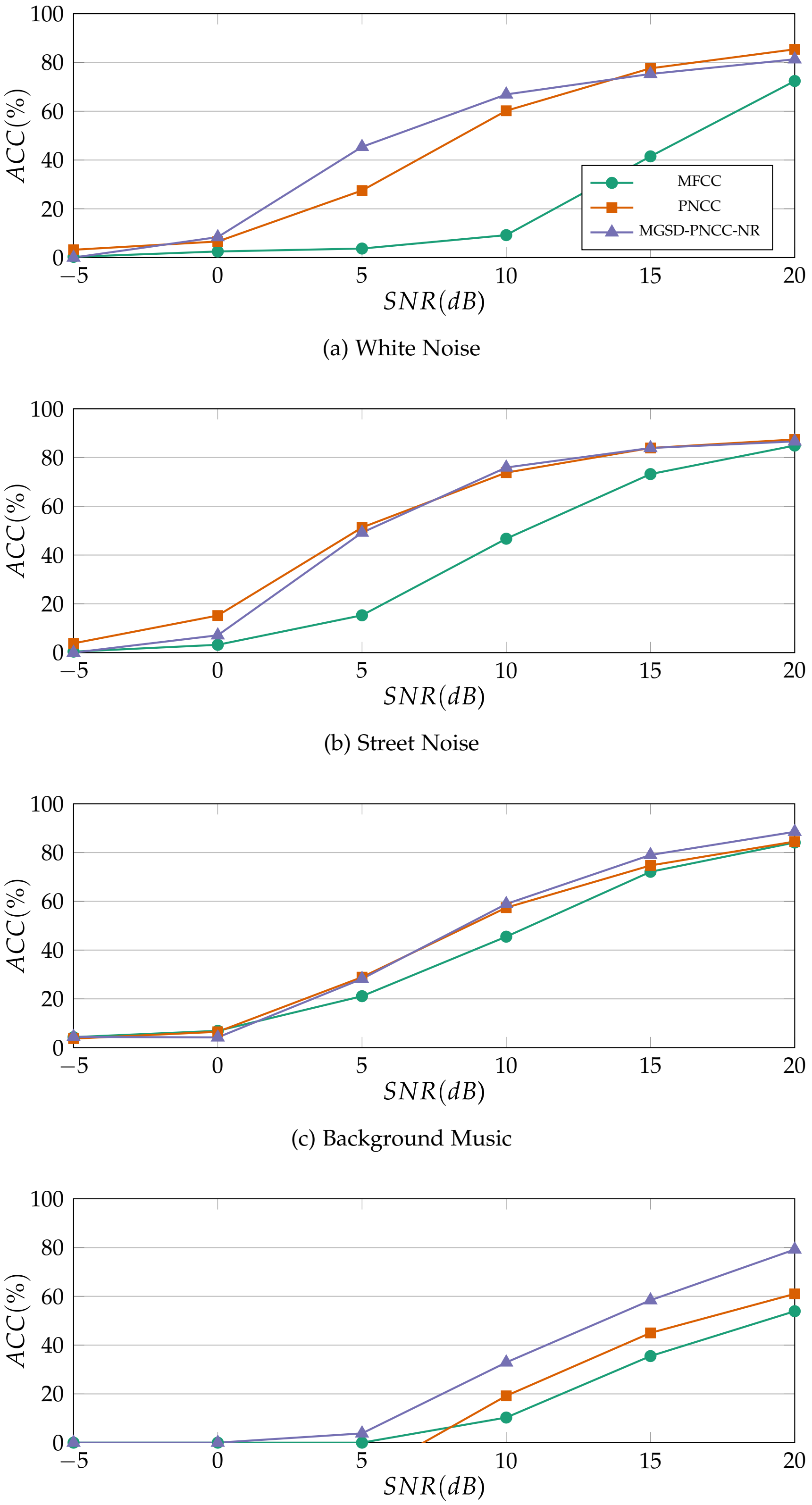
(a) White Noise

(b) Street Noise

(c) Background Music

(d) Interfering Speaker

Figure 5.9: Recognition results in terms of ACC(%) for four different noise conditions in the RM dataset.



(a) White Noise

(b) Street Noise

(c) Background Music

(d) Interfering Speaker

Figure 5.10: Recognition results in terms of ACC(%) for four different noise conditions in the WSJ0 dataset.

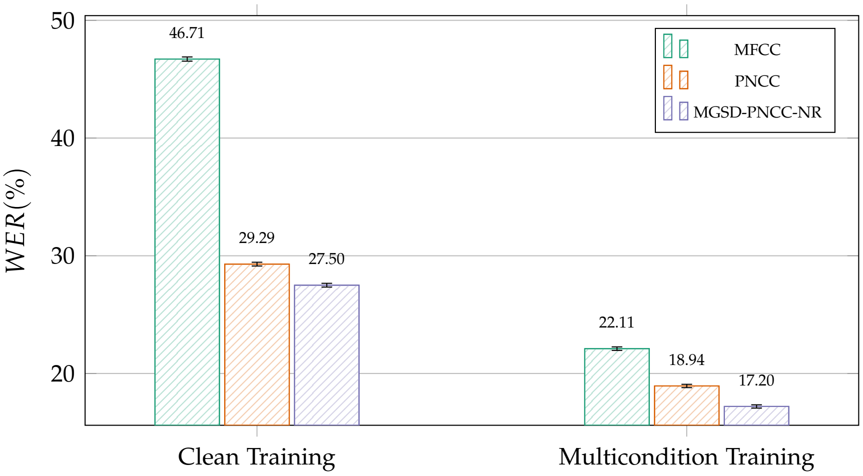


Figure 5.11: Recognition results in terms of WER(%) and 95% confidence intervals for matched and mismatched training using Aurora 4 data, average over all noise conditions.

Figures 5.9 and 5.10 compare results obtained using the MGSD-PNCC-NR processing with the MFCC and PNCC baselines for data from the RM and WSJ0 databases respectively, using the four standard noise types.

The MGSD-PNCC-NR processing provides better accuracy than MFCC features with all the tested noises, although improvements are small in some cases. MGSD-PNCC-NR processing never obtains substantially worse results than MFCC. Also MGSD-PNCC-NR processing provides better accuracy than PNCC features with additive white noise and background speakers, although improvements are small in some cases. MGSD-PNCC-NR processing never obtains substantially worse results than PNCC features except when SNRs are high.

The lack of improvement observed for clean speech and high SNRs is a common observation about the performance of auditory models as well as most other approaches to robust speech recognition.

Finally, Figure 5.11 shows selected results obtained using the Aurora 4 database under the conditions described above, reporting averages over all the test sets and their 95% confidence intervals. We note that the use of MGSD-PNCC-NR processing provides relative improvements of 6.5% in WER compared to PNCC in mismatched conditions and 6.1% for matched conditions. Improvements compared to MFCC features significantly greater as expected.

5.6 CONCLUSIONS

In this chapter we compared the improvements in speech recognition accuracy that can be obtained through the use of several types of features that are based on the extent of the synchronization of the auditory-nerve representation and its response. The most effective synchrony-based feature is a modified version of Seneff's GSD preceded by noise removal based on the PNCC algorithm. This feature provides substantially better recognition accuracy than the baseline PNCC features for speech that is degraded by white noise, interfering speakers, and reverberation. Improvements on speech distorted with street noise and background music are more modest.

6 DEEP LEARNING BACKGROUND FOR AUTOMATIC SPEECH RECOGNITION

6.1 INTRODUCTION

In the last years, the application of deep learning models to Automatic Speech Recognition (ASR) has allowed to increase the recognition rates drastically. As a consequence, nowadays Deep Neural Networks (DNNs) have become the state of the art in ASR systems, but there use in robust speech recognition is still an open problem. In particular, when the training data drastically differs from the testing data (known as mismatched case) it is necessary to propose novel methods to increase the generalization capabilities of the DNNs. This chapter describes the fundamental DNN techniques and algorithms, in order to understand the proposed methods and applications.

Following the distinction made in Section 3.2, the methods presented in this chapter aim to model how the human auditory cortex processes the speech in order to obtain the perceived sound. It is worth noting that artificial neural networks are only inspired in how real neurons work and that the objective is not to realistically model the brain.

Part of the topics treated in this chapter can be found in numerous reviews and books, in particular [115] is a remarkable introduction to deep learning and [116] presents a broad review of deep learning techniques applied to speech recognition. A general introduction to machine learning can be found in [117].

The remainder of the chapter is organized as follows. First, an introduction to machine learning is given, followed by an introduction of Artificial Neural Network (ANN) and deep feed forward networks. Convolutional networks are also included here since one of our contributions are based on them. Finally, the procedures to integrate DNNs in the ASR pipeline followed by a brief state of the art on the application of DNNs for robust speech recognition are explained.

6.2 MACHINE LEARNING

Machine learning gives humans the expertise to tackle problems that can not be solved by explicit programming, as speech recognition or computer vision. Many definitions of machine learning have been given, being that by Arthur Samuel's in 1959 the most cited: *"Machine learning is the field that gives computers the ability to learn without being explicitly programmed"* [118], or the more general definition by Tom Mitchell [119]: *"A computer program is said to learn form experience E with* *respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."*

The way by concrete samples of the experience are processed to learn constitutes the task.[1] For example, in a classification task, where a sample is defined by the vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$, and each component of the vector x_i corresponds to one of the n features of the sample, the classification algorithm has to choose which of C categories the sample belongs to. The classification task can be formalized as a function $y = f(\mathbf{x})$ where f maps from the feature space to the categories space. More specifically, y is usually a probability distribution over the different classes, estimating the posterior probability $P(y_i|\mathbf{x})$ of each of the corresponding classes y_i given the input vector $\mathbf{x}$.

Another important task is regression when the algorithms produce a continuous rather than a discrete output. Many other tasks can be solved using machine learning as clustering where a set of inputs are divided into groups or dimensionality reduction.

For example, in this thesis all the methods used fall into the task of classification because as we explained in Section 2.4, in a hybrid speech recognition task the objective is to obtain a probability distribution of the acoustic features over all the possible states of the Hidden Markov Model (HMM).

The performance measure is heavily dependent of the task. Examples of performance measures are the accuracy, i.e. the proportion of samples that the algorithm correctly classifies, or the mean square error in the case of a regression task.

The way in which the experience is defined depends of how the available dataset is used. It can be divided into unsupervised and supervised learning problems. A dataset is a collection of many different samples. If each sample of the dataset is associated with a target or label, we are facing a supervised problem. Otherwise, the problem is unsupervised and the algorithm has to learn the structure of the dataset without an explicit labeling. An example of this kind of algorithms is clustering where the goal is to group similar samples, or some dimensionality reduction algorithms as Principal Component Analysis (PCA) [120] where the objective is to reduce the number of features by retaining the relevant information without targets or labels. Besides unsupervised and supervised task, we find other algorithms as reinforcement learning, where the program interact with an environment seeking a goal.

In this thesis, the problem that we try to address using deep learning algorithms is to obtain a robust estimation of the emission probability of the HMM states, by using supervised classification algorithms, given that our dataset is defined by a set of samples and their cor-

[1] Note that this definition of *task* on a machine learning context is different from that employed in ASR throughout this document. We have decided to keep it to maintain the literality of the citation.

responding labels. These samples are obtained from speech using the previously presented features and the labels are usually given by a forced alignment between each utterance and its corresponding phonetic transcription by using a traditional Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) system, which provides an alignment of each sample to an HMM state. The reason for this procedure is the lack of a global criterion to optimize the neural network alongside the HMM. The forced alignment is needed every time a manual segmentation of the speech into different acoustic units under consideration is not available, which is usually the case.

Formally in the problem that we are facing, i.e. supervised learning classification, we try to learn the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ given a dataset of m examples $\left(\mathbf{x}^{(i)}, y^{(i)}\right) \in \mathcal{X} \times \mathcal{Y}$ sampled from de true data distribution $\left(\mathbf{x}^{(i)}, y^{(i)}\right) \sim D$, by choosing an optimal f^* belonging to the set of functions $\mathcal{F}$, that performs the mapping that is more consistent with the training set. This can be expressed as the minimizing the expected value of the lost function given samples drawn from D:

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \, \mathbf{E}_{(x,y) \sim D} L\left(f(\mathbf{x}^{(i)}), y^{(i)}\right), \tag{6.1}$$

where L is the loss function measuring the difference between the predicted label, $\hat{y}^{(i)} = f(\mathbf{x}^{(i)})$, and the true label $y^{(i)}$.

The problem expressed in Equation 6.1 is intractable due to the fact that we do not have all the possible samples of D. However it can be modified assuming that the samples are independent and identically distributed i.i.d:

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{m} L\left(f(\mathbf{x}^{(i)}), y^{(i)}\right), \tag{6.2}$$

where m is the number of samples in the dataset. In this way the loss is only optimized over the training samples expecting that those are representative of the true data distribution D.

The loss function is highly dependent of the task. For classification tasks in the field of DNN the most commonly used is the Cross Entropy (CE):

$$L_{CE}(\hat{y}^{(i)}, y^{(i)}) = -\sum_{j=1}^{C} y_j^{(i)} \log \hat{y}_j^{(i)}, \tag{6.3}$$

where C is the number of classes.

Minimizing the CE is the same as minimizing the Kullback–Leibler Divergence (KLD) between the probability distribution that the DNN estimates and the probability distribution of the training set. Frequently a hard class label is obtained by a so-called one hot encoding where $y_i = \mathbb{I}(c = i)$ where c is the hard class and $\mathbb{I}$ is the indicator function, converting Equation 6.3 into the Negative Log-Likelihood (NLL):

$$L_{NLL}(\hat{y}^{i}, y^{(i)}) = -\log \hat{y}_c^{(i)}, \tag{6.4}$$

where c is the correct class among the C possible classes.

6.2.1 *Regularization*

The main goal in machine learning in general and in DNN in particular, is to be able to perform well in new previously unseen data, rather than in the data where the model was trained.

The error in an estimator can arise from two sources. In the one hand, the *bias* is the error due to erroneous assumptions of the model, measuring the deviation from the true value. Typically, high bias indicates that the model capacity is not enough for the task, being this effect denoted as underfitting. On the other hand, *variance* is the error that comes from deviations caused by fluctuations of the particular sampling of the training data used to fit the model. High variance causes overfitting since the random fluctuations of the training data are captured instead of the underlying behavior.

Normally the dataset is divided into three sets: training, validation and test sets. The training set is used to learn the parameters of our model; the validation set allows to choose the hyperparameters of the model, as for example the number of layers in a DNN or the learning rate; finally, the test set allows the model to be tested in unseen data. If the training set is small, more complex divisions can be used like cross-validation [117].

We need to choose a model complex enough for our problem but that it does not overfit the training set and not to simple to under-fit. To solve this trade-off, regularization techniques can be used. These regularization techniques aim at reducing the generalization error but not the training error.

Regularization is introduced in machine learning models as penalties or constraints on the parameters values, such as the so called maxnorm normalization [121], in the objective function, as $l2$ regularization [115], or in the training process, as early stopping [122].

Formally regularization can be included in Equation 6.2 by introducing a regularization term R:

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{m} L(f(\mathbf{x}^{(i)}), y^{(i)}) + R(f), \tag{6.5}$$

where R is a regularization function that encodes the preference of some functions over others. Frequently it penalizes the complexity of the function f favouring the choice of simple and smooth solutions that are more likely to generalize better.

As deep learning models are very complex due to their high number of parameters, they are very prone to overfit the training set making the use of regularization almost compulsory. In our experiments some kind of regularization is always used. In Section 6.4 the applied methods are explained, a more detailed introduction can be found in Chapter 5 and 7 of [115].

6.3 ARTIFICIAL NEURAL NETWORKS

In previous chapters, we have seen how Human Auditory System (HAS) inspiration results in interesting algorithms that allow to increase the performance of speech recognition systems. The similar idea is applied to ANN with the aim of building more effective machines. As the main objective is to design a machine that allows to solve a practical problem and not to model the biological neurons in detail, ANNs have become a implausible biological model, but somehow a biological inspiration is still present.

Biological neurons are composed of a cell body that contains their principal elements. Numerous extensions called dendrites and a long extension, the axon, start from the cell body. The axon is further split into many branches called telodendria.

The simplified behavior of a neuron is as follows. The neuron receives electrical impulses from other neurons through the dendrites and when those accumulated stimuli are greater than a given threshold within a short time period, the neuron fires its own signal through the axon. Then the axon distributes the impulse across other neurons via the telodendria. The strength and timing of the spike that the neuron fires depends on its connections to other neurons. Some experiments show that the connections are reinforced depending on the frequency of the spikes received from the other neurons.

The process whereby the electrical impulses are transmitted from one neuron to another is called synapse. It is worth noting that although the behavior of a neuron can be seen simple, the high computational power comes from a broad network of neurons. For example, the human brain is composed of around 86 billion neurons, where each one is connected to thousand of others. Understanding how those bast neural networks, that is the brain, works is still an open research question.

The most common mathematical scheme [123] of the biological model, and the basics of the ANN are as follows. The signals from other neurons $(x_1, x_2, \ldots, x_n)$ enter the neuron from the dendrites and interact following a lineal combination $(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$, where the weights emulate the strength of the synapse. Those weights can be learned and control the influence of one neuron on others. Finally, when the addition of each influence is above a certain threshold, the neuron fires by sending an electrical stimulus though the axon. As can be seen the mathematical model differs from the biological model in that the timing of the spikes are not taken into account.

6.3.1 *The perceptron*

The perceptron, one of the simplest ANN types, was developed following the simplified mathematical model of the biological neuron [123]. It is composed of many Linear Threshold Unit (LTU). In an LTU the inputs are associated to a set of weights performing a linear combination of the inputs. Then a step or, in some cases, the sign function is applied to this combination. This is known as the activation function.

Specifically, given the feature vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, the weight vector $\mathbf{w} = [w_1, w_2, \dots, w_n]$ and the activation function f, the LTU unit operation can be defined as follows:

$$\hat{y} = f(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n). \tag{6.6}$$

Normally, an extra bias feature b is added as an input, resulting in the following equation:

$$\hat{y} = f(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b). \tag{6.7}$$

A single LTU can be seen in Figure 6.1a. It can be employed for binary classification by computing a linear combination of the inputs and applying a threshold just like logistic regression or Support Vector Machine (SVM). To obtain a multi-class classifier, several LTUs can be connected to all the inputs in a single layer. Figure 6.1b presents a perceptron with C output classes.

The operations performed in the perceptron in order to obtain the final output can be formally expressed in matrix notation as:

$$\mathbf{y} = f\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right), \tag{6.8}$$

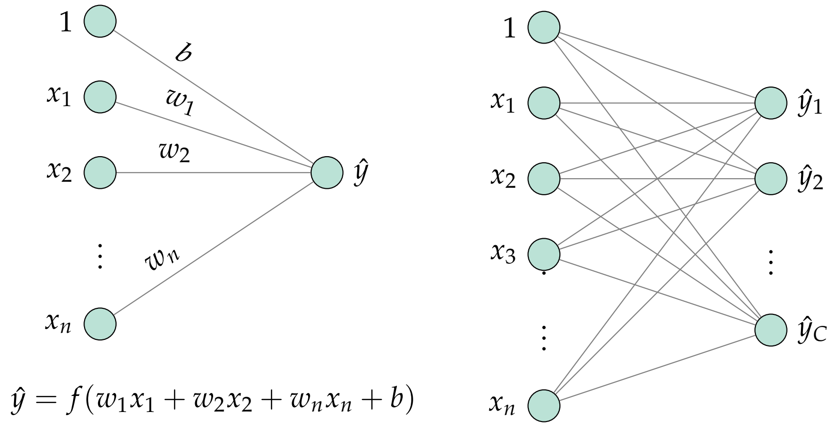
where $\mathbf{y}$ denotes the vector that contains the C outputs; $\mathbf{W} \in \mathbb{R}^{C \times n}$ is the matrix that contains the weights of the C LTUs that compose the perceptron; $\mathbf{x} \in \mathbb{R}^{n \times 1}$ is the input feature vector, $\mathbf{b} \in \mathbb{R}^{C}$ is a vector with the bias of each LTU, and f is the activation function.

The perceptron is trained by choosing the weights that minimize the chosen loss function. First, a prediction is made for a training sample and if it is incorrect, the connections that could make the correct result are reinforced. This is the basics of every training algorithm used for more complex networks.

Formally the update rule for one training sample $\mathbf{x}$ with a one hot encoding label $\mathbf{y}$ (a vector with the dimension of the number of classes, with a one in the correct class and zero in the rest) and predicted label $\hat{\mathbf{y}}$ is as follows:

$$w_{i,j} \longleftarrow w_{i,j} + \eta(\hat{y}_j - y_j)x_i, \tag{6.9}$$

where $w_{i,j}$ is the weight between the i input and the j output, and η is the learning rate that needs to be chosen for each problem.



(a) LTU (b) Perceptron

Figure 6.1: A single LTU unit and a representation of a perceptron with n input units and C output units.

It is worth noting that in order to train the perceptron, all the samples of the dataset are presented one by one until the error obtained on the development set is less than a predefined value. This kind of training where the samples are presented iteratively is called stochastic training or mini-batch training with a batch size of one.

6.3.2 *Multi-Layer Perceptron (MLP)*

Since the perceptron cannot model complex distributions such as the XOR function, a new architecture was develop by stacking multiple perceptrons, the MLP [124]. As can be seen in Figure 6.2, an MLP consists of an input layer, a fully connected hidden layer with perceptrons and an output layer also fully connected to the hidden layer.

Given the feature vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$, and the weight matrix $\mathbf{W}^{(1)} \in \mathbb{R}^{n_1\times n}$ that contains the weight of each connection, where n_1 is the number of units in the hidden layer, and the bias vector $\mathbf{b}^{(1)} \in \mathbb{R}^{n_1\times 1}$, the operation performed by the hidden layer is defined as:

$$\mathbf{h} = f(\mathbf{z}^{(1)}) = f\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right), \tag{6.10}$$

where $\mathbf{z}^{(1)} \in \mathbb{R}^{n_1\times 1}$ is the excitation vector, $\mathbf{h}^{(1)} \in \mathbb{R}^{n_1\times 1}$ is the activation vector and f is the activation function applied element-wise.

Usual activation functions are the sigmoid function:

$$f(z) = \sigma(z) = \frac{1}{1+e^{-z}}, \tag{6.11}$$

or the hyperbolic tangent function:

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{6.12}$$

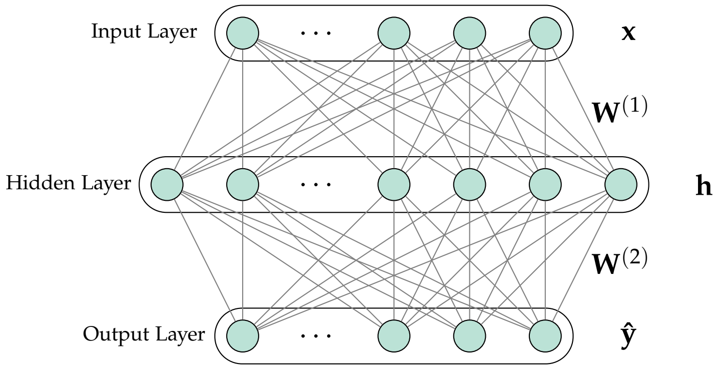


Figure 6.2: Representation of a MLP. For the sake of simplicity the bias units are not represented.

A representation of different activation functions and their derivatives are presented in Figure 6.3.

For the final output, first the excitation vector $\mathbf{z}^{(2)}$ is computed:

$$\mathbf{z}^{(2)} = \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}, \tag{6.13}$$

where $\mathbf{W}^{(2)} \in \mathbb{R}^{n_2 \times n_1}$ is the weight matrix, where n_2 is the number of units in the output layer, $\mathbf{b}^{(2)} \in \mathbb{R}^{n_2 \times 1}$ is the bias vector for the output layer and f is the activation function.

Depending on the task where the MLP is going to be used, the activation function used in the output layer can be applied or not. For example, in regression tasks no activation function is used, obtaining an output vector of n_2 dimensions.

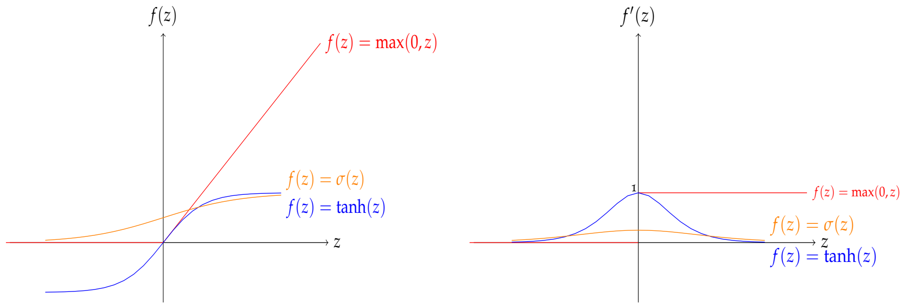
When the MLP is used for classification, each of its outputs represents the a posteriori probability of the C classes ($n_2 = C$) given the input sample. In order to obtain a valid probability the excitation vector is normalized using the softmax function. Formally the posterior probability $P(y_i|\mathbf{x})$ for each class y_i given the input vector $\mathbf{x}$ is computed as:

$$P(y_i|\mathbf{x}) = \frac{\exp\left(z_i^{(2)}\right)}{\sum_{j=1}^{C} \exp\left(z_j^{(2)}\right)}, \tag{6.14}$$

where $z_i^{(2)}$ is the i-th component of the excitation vector.

In [125] the authors show that a MLP with a softmax output layer and minimum square error as a cost function estimates the Bayesian a posteriori probabilities, $P(y_i|\mathbf{x})$. Treating the outputs as probabilities allows outputs from multiple networks to be combined, simplifies the creation of thresholds in the outputs and, particularly relevant in our case, allows the use of MLP in hybrid speech recognition systems where the emission probabilities of the states need to be estimated. Similar theoretical results are also demonstrated for other cost functions such as cross-entropy.

In order to train a MLP the back-propagation algorithm is used. It was discovered at the same time during the 1970s and 1980s by



(a) Activation functions

(b) Derivatives

Figure 6.3: Commonly used activation functions: sigmoid $\sigma(z)$, the hyperbolic tangent $tanh(z)$ and ReLU $max(0,z)$, with their corresponding derivatives

different research groups [124, 126, 127] to train MLPs. It consist of two steps: first, in the forward-pass a prediction for each training instance is obtained and then in the backward-pass its error is computed according to a cost function and the error is back-propagated through the network to measure the contribution of each connection weight to the error. Finally the weights are updated to reduce the network error using gradient descent optimization methods.

Formally, given a loss function the model parameters can be learned by following the first order error gradient direction. Stochastic gradient descent (SGD) [128] or Adam [129] algorithms are the most frequently used.

To optimize the parameters using SGD, first, a mini-batch of samples is obtained by randomly sampling the training data. This is done because the dataset that we use is too large to compute the gradient exactly, and therefore, it is estimated using a small mini-batch of samples and performing many approximated updates. Second, the gradient $\nabla_w L$ is estimated using backpropagation over the mini-batch, and the parameter update is performed:

$$w \longleftarrow w - \eta \frac{\partial L}{\partial w}, \tag{6.15}$$

where L is the loss function; w is any of the weights of the network and η is the step size or learning rate. Finally, the last steps are repeated until convergence.

The choice of η is a critical problem. Frequently cross-validation is used and also learning rate schedules are employed in order to have a higher learning rate in the beginning and smaller in the end. Usually, L is the CE loss function with some degree of regularization as explained in Section 6.2.

Nowadays, the back-propagation algorithm is described as using a gradient descent algorithm where reverse-mode auto-differentiation over a computational graph is employed to obtain the derivative

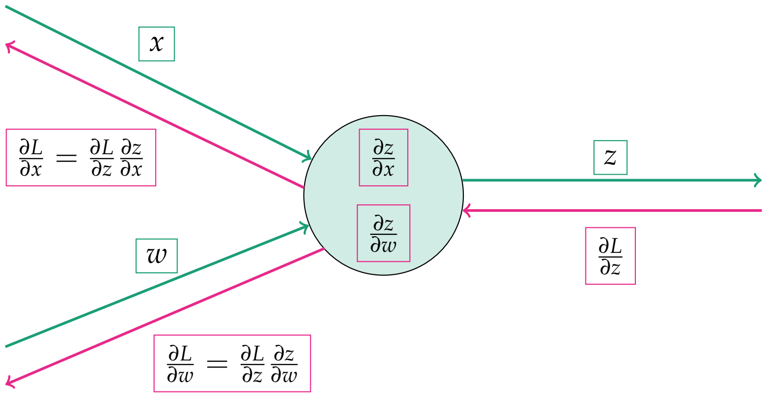

x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial w}$
z
$\frac{\partial L}{\partial z}$
w
$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial w}$

Figure 6.4: In this computational graph, $\frac{\partial L}{\partial w}$ is found by taking the previously computed gradient $\frac{\partial L}{\partial z}$ and multiplying it by the local gradient $\frac{\partial z}{\partial w}$. The gradient is recursively propagated continuing the graph in reverse direction. If z is the linear combination of the inputs $\mathbf{z} = \mathbf{W}\mathbf{x}$ then $\frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L}{\partial \mathbf{z}}\mathbf{x}$, becoming only necessary to store the value of the inputs of the node in the forward pass and the multiplication by the gradient, in the backward pass.

of a cost function respect the parameters to learn. To estimate the gradient, the value of each node and the final total loss is obtained in the forward-pass starting at the input, and then the backward-pass proceeds in reverse order applying the chain rule to find the influence of all the inputs on the final loss. An example of this process can be found in Figure 6.4.

6.4 DEEP FEED FORWARD NETWORKS

Deep feed forward networks are the quintessential deep learning model. The main difference respect to the traditional MLPs are the larger number of hidden layers between there inputs and outputs. An MLP can be considered a deep feed forward network with just one hidden layer.

As in the traditional MLP, the main characteristics of a feed-forward networks are that the information flows only in one direction (between the input and the output of the network) and that they are built as a concatenation of many functions. DNN can be seen as a structure composed of a chain of many functions, where each of the blocks of the chain computes a representation of the inputs, and the final layer acts as a classifier over the last representation.

Many authors regard deep learning as representation learning since it can be understood as a sequence of transformations of the input features in each of the layers that are finally classified in the last one. In particular, the DNN computes low, mid and high level features and acts as a classifier [130]. When the number of layers (or depth of the

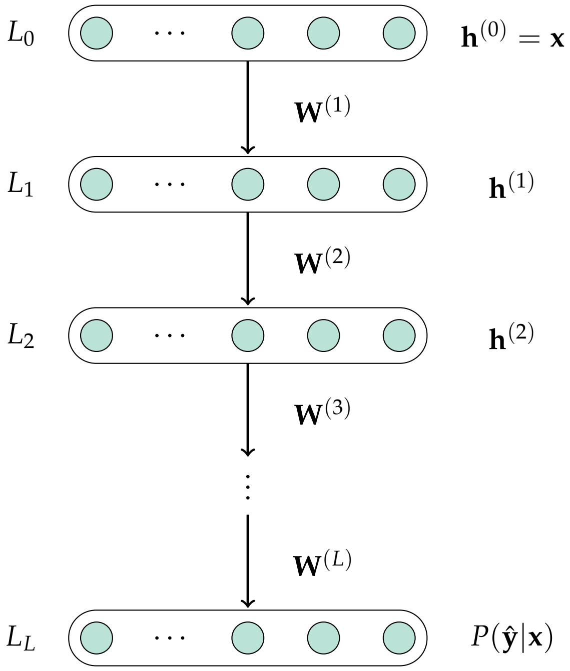


Figure 6.5: Representation of a feedforward DNN with L layers. For simplicity the bias units are not represented.

network) increases, those representations are improved. Recent experiments show that very deep networks lead to important breakthrough results [13, 131, 132].

A feed-forward neural network learns how to map a fixed-size input, as a spectrogram during a timespan, to a fixed-size output, for example the a posteriori probabilities of each state. To obtain the output of each layer, a set of units or neurons compute a weighted sum of their inputs from the previous layer and the activation function is applied to its result. Figure 6.5 shows a representation of a feed-forward network of L layers with the parameters and outputs of each layer.

Defining the input layer as layer 0 and the output layer as L the output of the l-th layer with n_l units or neurons is computed as:

$$\mathbf{h}^{(l)} = f(\mathbf{z}^{(l)}) = f\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \quad 0 < l < L, \tag{6.16}$$

where $\mathbf{h}^{(l)} \in \mathbb{R}^{n_l \times 1}$ is the activation vector of layer l; $\mathbf{z}^{(l)} \in \mathbb{R}^{n_l \times 1}$ is the excitation vector of layer l; $\mathbf{h}^{(l-1)} \in \mathbb{R}^{n_{l-1} \times 1}$ is the activation vector of the previous layer; $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the matrix that contains the weights of each connection; $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l \times 1}$ is the vector that contains the bias of each unit; L is the number of layers and f is the activation function applied elementwise.

It is worth noting that for the input layer $\mathbf{h}^{(0)} = \mathbf{x}$ and that, for classification tasks, a softmax function is used as the activation function of the output layer $\mathbf{z}^{(L)} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}$ as in MLPs, obtaining a posteriori probabilities $P(\mathbf{y}|\mathbf{x})$.

Given the input vector and all the parameters, the forward computation of the feed-forward DNN can be performed by calculating the activation functions layer by layer following the Equation 6.16.

In the 80s when MLP were presented, only one hidden layer was used, first because the computational resources were limited and second because it was believed that the use of many hidden layers will not work, as the parameters obtained using gradient descent get stuck into a local minimum of the cost function. With the appearance of General Purpose Graphics Processing Units (GP-GPUs) of a reasonable costs with computational capabilities that allow the training of bigger networks the first problem was solved. To overcome the second, an unsupervised pre-training stage was devised to initialize the search around a nearly optimum solution instead of randomly. Techniques that fall into this category are Restricted Boltzmann Machines (RBMs) [133] or Stacked Denoising Autoencoders (SDAs) [134]. Nowadays, pre-training has become unnecessary with the use of ReLU activation functions because the activations do not saturate as the sigmoids, and with appropriate initialization techniques.

When the DNN is deeper, [135, 136] show that the local minimum problem rarely occurs as the most frequent critical points presented in the cost function are saddle points, where the gradient becomes almost zero, and therefore, the training process finishes. But almost all those saddle points have similar values obtaining equivalent solutions, allowing deep DNN to be trained.

The major difficulty when training bigger DNNs is the vanishing gradient problem where the backwards gradient propagation through many layers makes it become too small and, as a consequence, the parameters are barely updated. The vanishing gradient is caused by the concatenation of products of derivatives that are close to zero every time the backward-pass is computed (see Figure 6.4).

Vanishing gradient can be mitigated with the use of the so called ReLU activation function [137], defined as:

$$f(z) = \max(0, z). \tag{6.17}$$

Figure 6.3 shows the traditional sigmoid and tanh activations functions and their corresponding derivatives in comparison to the ReLU. As can be observed, the derivative of ReLU is non-zero for all positive values with a value of one, in contrast to sigmoid and tanh where the derivate is close to zero in a large portion of the function (making the gradient null), and in the rest it is lower than one (causing vanishing gradient problems). Also the value of the derivative of ReLU is computationally efficient as no calculation other that determining the sign is needed to obtain it.

Other relevant methods, that improve the generalization error and mitigate the effects of vanishing or exploiting gradients, employed in this thesis are the following:

- *Dropout [121]:* It reduces overfitting and improves the generalization capability of the network, by randomly omitting a certain percentage of the hidden units on each training iteration.

  When dropout is employed, the activation function of Equation (6.16) can be rewritten as:

  $$\mathbf{h}^{(l)} = m^{(l)} \star f\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \quad 1 < l < L, \tag{6.18}$$

  where $\star$ denotes the element-wise product, $m^{(l)}$ is a binary vector of the same dimension of $\mathbf{h}^{(l)}$ whose elements are sampled from a Bernoulli distribution with probability p. This probability is the so called Hidden Dropout Factor (HDF) and must be determined over a validation set as it will be explained in Section 6.8. As the ReLU function has the property that $f(0) = 0$, Equation 6.18 can be rewritten as:

  $$\mathbf{h}^{(l)} = f\left(m^{(l)} \star \left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right)\right), \quad 1 < l < L, \tag{6.19}$$

  where dropout is applied directly to the inputs of the activation function improving the efficiency of training.

  Note that dropout is only applied in the training stage whereas all the hidden units become active when testing. Dropout DNN can be seen as an ensemble of DNNs, given that on each presentation of a training sample, a different sub-model is trained (the subset of selected weights) and their predictions are averaged together. This technique is similar to bagging [138], where many different models are trained using different subsets of the training data; but in dropout each model is only trained in a single iteration and all the models share some parameters.

  Following [139], the parameters of the network need to be compensated in testing by scaling the weight matrices taking into account the dropout factor as follows:

  $$\overline{\mathbf{W}}^{(l)} = (1 - HDF) \cdot \mathbf{W}^{(l)}. \tag{6.20}$$

- *Maxout [140]:* It is a modification of the feed-forward architecture (Equation (6.16)) where the maxout activation function is employed. The maxout unit simply takes the maximum over a set of inputs. In a Deep Maxout Network (DMN) each hidden unit takes the maximum value over the g units of a group. The output of the hidden node i of the layer l can be computed as follows:

  $$h_i^{(l)} = \max_{j \in 1,\ldots,g} z_{ij}^{(l)}, \quad 1 \leq l \leq L, \tag{6.21}$$

  where $z_{ij}^{(l)}$ are the linear pre-activation values or excitation vector from the l-th layer:

  $$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}. \tag{6.22}$$

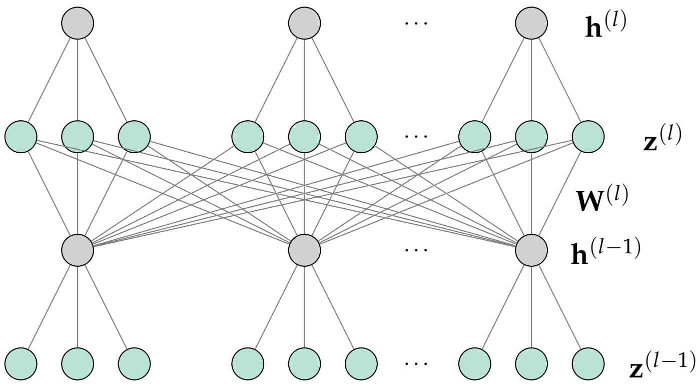


Figure 6.6: Two connected maxout layers with a group size of $g = 3$. The hidden nodes in gray perform the max operation.

As can be observed the max-pooling operation is applied over the $\mathbf{z}^{(l)}$ vector.

Note that DMNs fairly reduce the number of parameters over DNNs, as the weight matrix $\mathbf{W}^{(l)}$ of each layer in the DMN is $1/g$ of the size of its equivalent DNN weight matrix. This makes DMN more convenient for ASR tasks where the training sets and the input and output dimensions are normally very large. An illustration of two connected Maxout layers with a group size of $g = 3$ is shown in Figure 6.6.

In [140] a demonstration of the capability of maxout units to approximate any convex function by tuning the weights of the previous layers is included. For this matter, the shapes of activation functions are not fixed allowing the DMNs to model the variability of speech more smoothly. DMNs are commonly applied in conjunction with dropout magnifying its averaging effects.

- *Batch Normalization (BN) [141]*: it reduces the so-called covariance shift, i.e. the fact that the distribution of each layer's inputs changes during training, by normalizing the layers inputs. This normalization is done as a part of the model architecture for each mini-batch. BN allows a high learning rate and random parameters initialization to be used.

- *Parameter initialization strategies:* An important requisite to properly train the DNN is to use the correct distribution in order to perform the random initialization. For all networks trained in this thesis we use Xavier initialization [142].

6.5 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) [143, 144] are simply neural networks where a convolution replaces the general matrix multiplication (feed-forward) in at least one of the layers. The convolution operation can be seen as a feature map that uses a filter. A convolutional layer processes an image (spectrogram or any other time-frequency representation in the case of speech recognition) with filters whose parameters are learnt through back-propagation and SGD.

The main advantages of CNN are parameter sharing, reducing the number of parameters to train, preservation of the local correlations of the input and sparse interactions [115]. These advantages, in our case, allow to obtain better recognition rates as the convolutional networks are invariant to shifts in time and equivariant in frequency. Invariance to time shifts allows to detect the spectral patterns that represent speech anywhere in the input window and equivariants in frequency allows to produce the same representation when a shift in frequency takes place, as for example, when different speakers with different pitches produce the same phonemes.

In a more formal manner a convolutional layer can be expressed as:

$$\mathbf{h}^{(l)} = f\left(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \tag{6.23}$$

where $\mathbf{h}^{(l)}$ is the output feature map; $\mathbf{h}^{(l-1)}$ is the input feature map from the previous layer; $\mathbf{W}^{(l)}$ is the kernel or filter; $*$ denotes the convolution operation; $\mathbf{b}^{(l)}$ the bias term added after the convolution, and f denotes the activation function, typically a ReLU.

The convolution operation is normally applied to multidimensional arrays, in our case spectrograms or acoustic features composed of two dimensions or sometimes three if first and second order derivatives of the features are added as additional dimensions. For example, a spectrogram will be a two dimensional array where the rows correspond to different frequencies and the columns to different time frames.

The output in a two dimensional case of a convolution at position (i, j) using only one kernel $\mathbf{W}^{(l)}$ with dimensions $f_1 \times f_2$ and input $\mathbf{h}^{(l-1)}$ can be expressed as:

$$\left(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)}\right)[i, j] = \sum_{m=1}^{f_1} \sum_{n=1}^{f_2} \mathbf{h}^{(l-1)}[i-m, j-n]\mathbf{W}^{(l)}[m, n]. \tag{6.24}$$

This process can be interpreted as moving the kernel or filter over the feature map from the top left to bottom right; and at each step the dot product is performed between the kernel and the part of the feature map that is covered by the kernel. All the resulting dot products are assembled forming the resulting output feature map. A graphical representation of this operation can be seen in Figure 6.7, using a 3 × 3 kernel over a 5 × 5 input feature map.

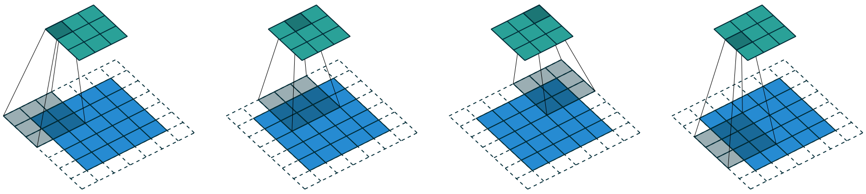


Figure 6.7: Convolving a 3 × 3 kernel over a 5 × 5 input padded with a 1 × 1 border of zeros using a stride of two (following the introduced notation: $m_i = n_i = 5$, $d_i = 1$, $f = 3$, $s = 2$ and $p = 1$). Redrawn form [145].

Some particularities in how we perform the operation involve the following terms:

- *Kernel dimensions:* the dimension of the filters need the be chosen, normally square kernels are used: $f \times f$. The third dimension of the filter is given by the number of input maps also known as input depth. Normally in the first layer the filter is larger 6×6 or 8×8, then small 3×3 kernes are common.

- *Depth:* In the previous example only one filter is used in the convolution operation, but more than one can be employed. The number of filters used in each layer is the depth. As a result the output will consist of various feature maps.

- *Stride:* It defines the pace at which the filter sweeps the input. For example with a stride of 1 the filter is slides one pixel at the time, and when the stride is 2 the kernel skips 1 pixel between computations. A stride higher than one produces a smaller output feature map.

- *Zero padding:* It is the number of zeros that are added in the border of the input feature map. This is normally used to produce a feature map that preserves the dimensions of the input.

In summary, for an input volume with dimensions $m_i \times n_i \times d_i$, where m_i and n_i are the dimensions of one input feature map (for example the spectrogram) and d_i is the number of input feature maps ($d = 3$ in the case of including the first and second order derivatives of the parameters as additional features maps in the first layer), we need to define the following parameters for each layer: the number of filters to be used k, that is the depth, the dimensions of the filters f, the stride of the convolution s, and the number of zeros padded in the border p.

The convolution operation will then produce an output volume of $m_o \times n_o \times d_o$, where $m_o = (m_i - f + 2p)/s + 1$, $n_o = (n_i - f + 2p)/s + 1$ and $d_o = k$. Note that the stride and the dimensions of the kernel could be different in height and width although is not considered in this notation for simplicity.

As a result the number of parameters of the network for each layer will be $f \cdot f \cdot d_i$ parameters per filter, resulting in a total $f \cdot f \cdot d_i \cdot k$ per layer. It is worth comparing the number of parameters in CNNs with respect to a fully connected layer where the parameters are not shared between inputs, as in the convolution when a small filter is slid over the input. To produce the same output dimension in a fully connected layer $m_i \cdot n_i \cdot d_i \cdot m_o \cdot n_o \cdot d_o$ parameters will be required, as each output needs to be connected to each input, resulting in a vast number of parameters when the input image or spectrogram is large.

Another important type of layer utilized in the convolutional networks are pooling layers, which are normally used to reduce the dimension of the feature maps and therefore to reduce the number of parameters employed in the following layers controlling the overfitting. The pooling layer is applied over each feature map in the same manner as the convolution, i.e. by sliding a filter over the input feature map, but instead of performing the dot product over the input and some learned weights, the pooling layer applies the maximum (in case of a max-pooling) or the average (in case of an avg-pooling) over the values of the input feature map that are covered by the filter. The most common is a max-pooling layer with a filter size of 2×2, applied with a stride of 2. This layer reduces the input volume by 2 in each dimension except the depth (number of feature maps) as the pooling operator is applied independently over each feature map.

In a conventional CNN pooling layers are included between convolutional layers, reducing the dimension of the feature maps. In the usual configuration each convolutional layer increases the number of filters as the layer is closer to the output of the network, and pooling layers are inserted to reduce the feature maps size. Finally when the number of parameters is reasonably low, fully connected layers are included followed by a softmax output layer that produces the probabilities of each class to be classified.

6.6 DNN BASED SPEECH RECOGNITION

As we explained in Chapter 2, DNN can be applied both in the so-called *tandem* [34] and *hybrid* [146] architectures. Also recently newer end-to-end deep models where the HMM is removed have been proposed [19]. All the results presented in this thesis are based in the hybrid architecture because of its reduced computational cost in comparison with other approaches.

6.6.1 *Hybrid speech recognition systems*

In Chapter 2 we presented hybrid Artificial Neural Network-Hidden Markov Models (ANN-HMMs). In essence, in a hybrid scheme the acoustic modeling performed by the Gaussian Mixture Models (GMMs) is replaced by an ANN computing the state emission likelihoods.

In a hybrid Deep Neural Network-Hidden Markov Model (DNN-HMM) system, just like in classical ANN-HMM hybrids [14], a DNN is trained to classify the input acoustic features into classes corresponding to the states of HMMs. The HMM topology is set from a previously trained GMM-HMM system, and the DNN training data come from the forced-alignment between the speech signals and the corresponding state-level transcriptions obtained by using this initial GMM-HMM system. A hybrid DNN-HMM model is depicted in Figure 6.8.

DNN-HMM hybrid systems combine several features that make them superior to previous ANN-HMM hybrid schemes [28]:

- DNNs have a larger number of hidden layers leading to systems with many more parameters than the later. As a result, these models are less influenced by the mismatch between training and testing data but can easily suffer from overfitting if the training set is not big enough.

- The network usually models senones (tied states) directly (although there might be thousands of senones).

- In the DNN-HMM long context windows (10 to 15 frames) are used.

Although conventional ANNs also take into account longer context windows than HMM or are able to model senones, the key to the success of the DNN-HMM is the combination of these components. DNN-HMM systems with these properties are often named Context Dependent-Deep Neural Network-Hidden Markov Model (CD-DNN-HMM).

As previously stated the DNN estimates the a posteriori probability $P(s^i|\mathbf{x}_t)$ of each state s^i given the observation $\mathbf{x}_t$ at time t, through a softmax final layer. In the recognition stage, the DNN estimates the emission probability of each HMM state, $P(s^i|\mathbf{x}_t)$, in order to obtain the state emission likelihoods $P(\mathbf{x}_t|s^i)$ for the HMM, the Bayes rule is used as explained in Section 2.4, employing the priors of each state $P(s^i)$ which can be estimated by counting the occurrences of each state on the training data.

Almost in every case the input vector $\mathbf{x}$ is a concatenation of the features along a context window, as illustrated in Figure 6.8.

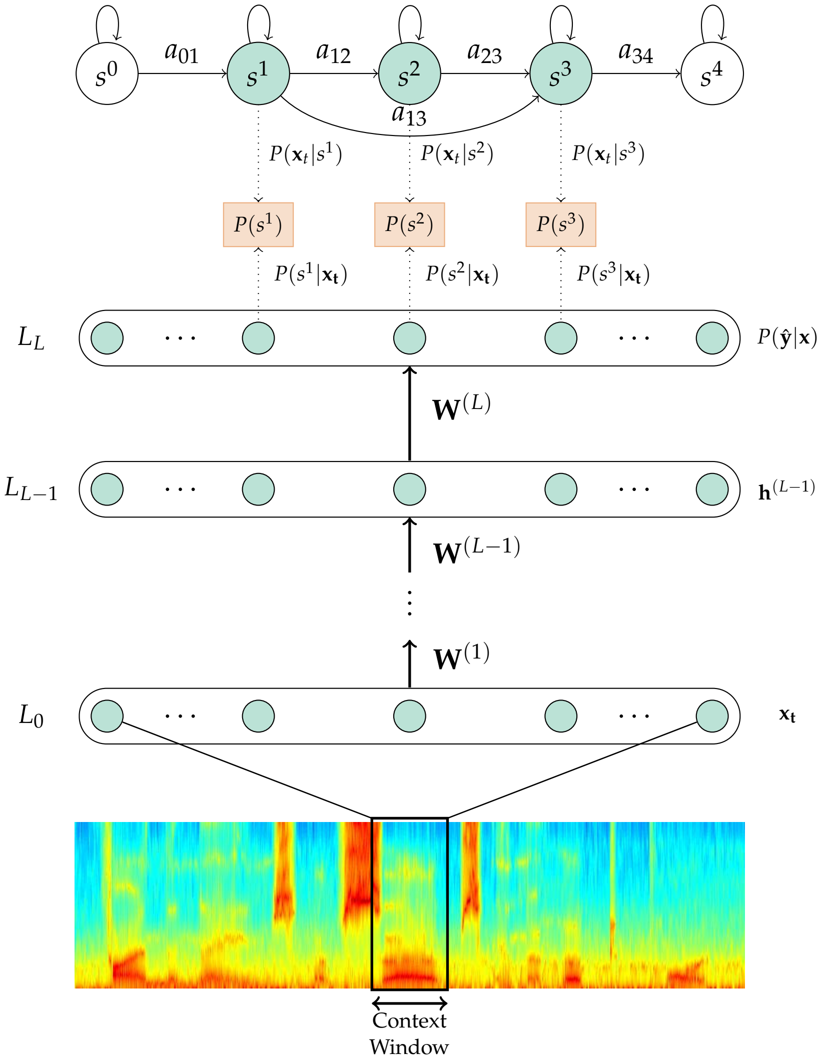


Figure 6.8: Representation of the ASR hybrid system.

6.6.2 *End-to-end deep models*

Lately advanced deep models have been proposed where an end-to-end deep learning approach is used to model the ASR problem, this models learn the alignment jointly with the acoustic model.

In [19, 147] a speech recognition system is shown where the HMM have been eliminated with the use of Recurrent Neural Networks (RNN)-Long Short Term Memory networks (LSTM) [148] trained using the Connectionist Temporal Classification (CTC) cost function [149].

The end-to-end approach has the advantage of not requiring the alignment between frames and states in order to train the network. Using the CTC the network is trained using only a sequence of data inputs, without the alignment, obtaining a better performance because the system learns the alignment together with the weights. Although the advantage of not requiring the frame alignment can also be reach in hybrids models using the Maximum Mutual Information (MMI) criterion [150].

This kind of systems are used in some ASR industrial products [19, 151], but they have various drawbacks: they require a large number of Graphics Processing Units (GPUs) and large amount of varied data for training, becoming unfeasible for cases where the computational power and data are limited. End-to-end systems have obtained good results only when trained with phoneme acoustic units not words [151, 152], making necessary to incorporate phones to words converters as Grapheme to Phoneme (G2P) [153]. Also if the systems is trained using words the adaptation to newer scenarios is harder as training data with the same vocabulary is needed.

Some controversies exits about the end-to-end systems as some experiments show that the improvements obtained for end-to-end training, on a phone, not word level, were due to the lower frame rate and not because the CTC cost function. A lower frame rate is possible in a CTC model as the blank symbol is available as an output together with the regular acoustic units. In [154] the authors show that by lowering the frame rate to 40 ms in conventional deep hybrid models a 3% relative decrease in Word Error Rate (WER) is obtained in comparison with a CTC based model on a large vocabulary Voice Search task.

6.7 DEEP NEURAL NETWORKS FOR ROBUST SPEECH RECOGNITION

The important leap in performance that ASR has experienced in the last years is mostly due to the introduction of new acoustic models based on DNNs [10, 146, 155], becoming the state of the art in acoustic modeling.

In spite of the recent achievements of ASR systems, their performance is still worse than that of humans in noisy or reverberant environments.

Also an important deterioration in noisy environments is showed in mismatched conditions [156]. A broad range of techniques have been proposed aimed at solving this problem, but the performance is still far from that of high Signal-to-Noise Ratio (SNR) scenarios, where human performance has been obtained in some cases [157].

To apply DNN in robust speech recognition different approaches are feasible:

- *Data augmentation with multi-condition data:* a straightforward solution to increase the robustness is to train the DNN with a large variety of noisy data. This is an effective approach to reduce the mismatch between the training and testing sets. As DNNs are able to be trained with a large amount of data the performance increases substantially, but in some situations where the test conditions are unknown, it can be impractical or impossible and the performance deteriorates drastically.

  As we presented in Chapter 2 in the matched case the obtained results improves drastically in comparison to conventional GMM-HMM systems. The same applies when the training data is corrupted with noise as the training data distribution is similar to the test data distribution. In [158] data augmentation is used increasing the performance in noisy conditions, in [159] artificially added reverberation in the Aurora 2, shows a moderate improvement. Similar results can be found in [160] for reverberant speech in a large dataset.

- *Incorporating a noise model:* the performance can increase when an estimation of the noise is given to the DNN. In [161] an estimation of the noise is included as an input to the network aiming at making the DNN automatically learn the relationship between noisy speech and noise.

- *Robust network architectures:* another solution to the problem is to apply robust architectures or training methods.

  Dropout has already been successfully tested on noisy speech in [161]. The benefits come from the improved generalization abilities attained by reducing the network capacity. Another interpretation of the behavior of dropout is that in the training stage it adds random noise to the training set resulting in a network that is very robust to variabilities in the inputs (in our particular case, due to the addition of noise).

  In [162], DMNs are used in low-resource speech recognition and in [3] we apply them to robust speech recognition. Those results are presented in the next section.

  Finally, the application of CNNs in robust speech recognition tasks has also produced an increase in the performance in noisy conditions [163, 164] and reverberation [165, 166], as CNNs are noise robust themselves especially when noise or distortion are located in the spectrum [167]. The advantages of using CNNs are explained in detail in next chapter.

- *Robust features:* another approach to boost the robustness is to use robust features that provide an invariant representation of the acoustic space, allowing the DNN to increase the performance in the mismatched case.

  Features presented in Chapter 3 can be used with some modifications in a DNN acoustic modeling stage. In particular the features need to be log-spectral rather than cepstral. In GMM-HMM systems uncorrelated features, normally obtained by applying the Discrete Cosine Transform (DCT) over log-spectral coefficients, are necessary as typically the GMMs used are chosen to have diagonal covariance matrix by design. In a DNN-HMM system, this is not necessary and even it has been demonstrated that DNNs obtain better results when cross-correlations are present in the input features [168]. For these reasons, the log-spectral features perform better [161], being the current trend in state of the art DNN-based ASR systems.

  Other types of inherently robust features are valid alternatives, as shown in [169] where Gabor-derived acoustic characteristics are incorporated in CNN as kernel filters.

  Finally, some transformations performed over conventional features initially proposed for speaker adaptation, such as Feature-Space Maximum Likelihood Linear Regression (fMLLR) [170] or i-vectors [171, 172], have shown to be effective when used together with DNN in robust speech recognitions tasks.

  A complete review of robust features for DNN can be found in [156].

- *System combination:* in [173] it was suggested that the errors produced by a DNN-HMM ASR system are different than those generated by a GMM-HMM, benefiting from fusing approaches such as Recognition Output Voting Error Reduction (ROVER).

  Following this idea, in [5] we analyzed the errors of traditional and deep systems in six broad phonetic classes: vowels, semivowels, nasal consonants, fricative consonants, affricate consonants, and stop closures and silence segments. This analysis was performed over a GMM-HMM system and a DNN-HMM hybrid system using feed-forward DNN using different initialization methods and DMN.

  The experiment showed that the performance is still tightly related to the particular phonetic class being stops and affricates the least resilient, but also that relative improvements of both

> DNN variants are distributed unevenly across those classes having the type of noise a significant influence on the distribution.
>
> A combination of the different DNN-based systems and classical GMM-HMM using ROVER was also proposed to validate our hypothesis that the traditional GMM-HMM systems have a different type of errors than the Deep Neural Networks hybrid models. The results showed that the combined systems achieved better accuracies than the individual ones in a robust speech recognition task. Although improvements were small in some cases, they were consistent with our analysis in which we concluded that the performance was significantly dependent on the phonetic classes.

In this thesis the challenge of applying DNN to robust speech recognition is addressed by using the previously presented features together with novel deep models.

First, in the next section we describe our preliminary experiments where DMNs were applied to robust speech recognition in a small dataset, as our computational resources were limited at that time. Next, when a GPU became available[2], we explored very deep convolutional networks for acoustic modeling together with our auditory motivated features. These results are presented in Chapter 7.

6.8 DMNS APPLIED TO ROBUST SPEECH RECOGNITION

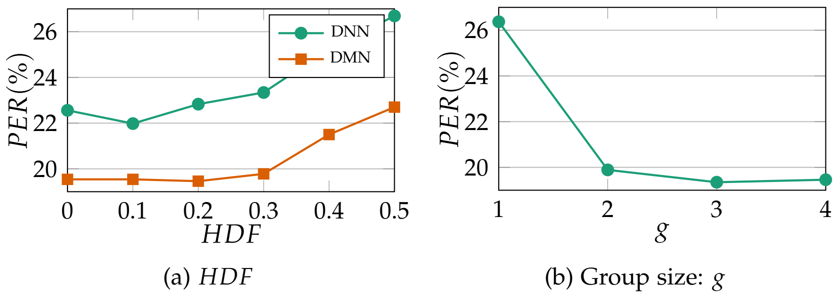
Our first contribution using DNNs in robust speech recognition is to apply DMNs in combination with dropout strategies in a noisy speech recognition task, demonstrating a substantial improvement of the recognition accuracy over common DNNs and other traditional techniques.

DMNs reduce the size of the parameter space significantly making them very suit for ASR tasks, where the training sets and input and output dimensions are normally quite large.

For this reason, DMNs have been employed in low-resources speech recognition devices [162] boosting the performance over other methods. We hypothesize that DMNs can improve the recognition rates in noisy conditions given that they are capable to obtain a robust model of the speech from limited data more effectively [162].

In this section, we present the experiments carried out for evaluating and comparing the performance of a conventional GMM-HMM and the different hybrid deep neural networks-based ASR systems: basic DNN with random initialization, basic DNN with pre-training, DNN with dropout and DMN. In all the basic DNN the ReLU activation function is applied in all hidden layers.

[2] We are grateful to Nvidia corporation for supporting our research by donating a GeForce Titan X.



(a) *HDF*

(b) Group size: *g*

Figure 6.9: Results in terms of PER(%) as a function of HDF for DNN and DMN (left) and the group size for DMN (right) on TIMIT development set. Both nets have 5 layers.

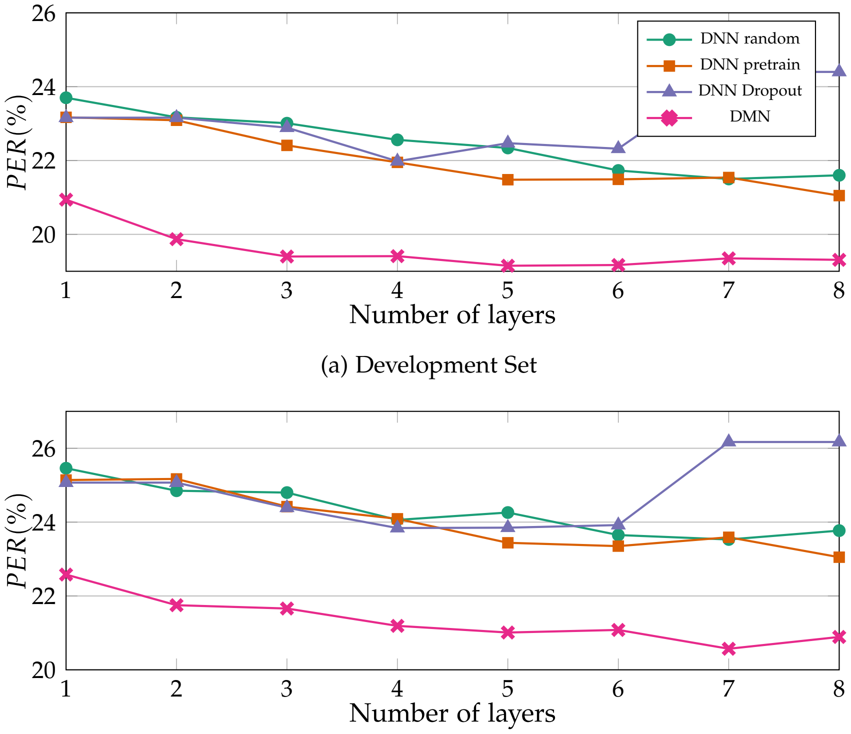
The experiments were performed on the TIMIT corpus. In particular, we used the 462 speaker training set. A development set of 50 speakers to tune all the parameters and finally the 24 speakers core test set. Each utterance is recorded at 16 kHz and the corpus includes time-aligned phonetic transcriptions allowing us to give results in terms of PER.

To test the robustness of the different methods we followed the lines of the previously presented experiments. We added noise with the Filter and Noise-adding Tool (FANT) tool [25] for four different noises (white, street, music and speaker) at different SNRs. All tests were evaluated in a mismatched condition. We employed the Kaldi toolkit [30] for implementing the traditional GMM-HMM ASR system and the Python Toolkit for Deep Neural Networks (PDNN) toolkit [56] for the hybrid DNN-based ASR systems.

For the traditional GMM-HMM the input features were 12th-order Mel-Frequency Cepstral Coefficientss (MFCCs) plus a log-energy coefficient, and their corresponding first and second order derivatives yielding a 39 component feature vector. Mean and variance normalization on each of the components were applied. For the hybrid models, 40 log filter-banks with a context of 5 frames was chosen. All the hybrid systems were trained with the labels generated from the best performance GMM-HMM system through forced alignment.

First, we tuned the configuration parameters of the networks (number of hidden layers, HDF and group size, when applicable) under clean conditions. HDF and group size were validated on the development set as can be seen in Figure 6.9, considering 5 hidden layer networks, yielding an optimal dropout factor of 0.1 for dropout DNNs, 0.2 for DMNs and a group size of $g = 3$. These values of HDF and group size were used throughout the rest of the experiments. DMNs were always employed in conjunction with dropout.

Figure 6.10 shows the PERs as a function of the number of hidden layers for the development and test sets for different types of hybrid DNN-based ASR systems: randomly initialized, with a pre-training



(a) Development Set

(b) Test Set

Figure 6.10: Comparison of the performance of the different hybrid DNN-based ASR systems in terms of PER(%) as a function of the number of hidden layers for TIMIT development and test sets.

stage, with dropout and maxout networks. The number of hidden nodes in all of the DNNs is 1024. To be fair, we chose 400 hidden maxout units for the DMN since $400 \times 3 = 1200$ yields a number of parameters in the same order as the DNNs. For the networks without dropout, the learning rate started at 0.08 for 30 epochs and was subsequently divided in half while the validation error decreased. For the dropout and DMNs, we started with a higher learning rate of 0.1. As can be seen in Figure 6.10, DMNs clearly outperform the other networks for all the number of layers considered. The best results were obtained in the development set with DMNs of 5 layers.

Second, we compared different variants of the baseline system GMM-HMM (Monophone, Triphone, Triphone with Linear Discriminant Analysis (LDA), Maximum Likelihood Linear Transform (MLLT) and Speaker Adaptative Training (SAT)) with the best configuration of the different hybrid ASR systems under clean conditions. Results for the development and test sets are shown in Table 6.1.

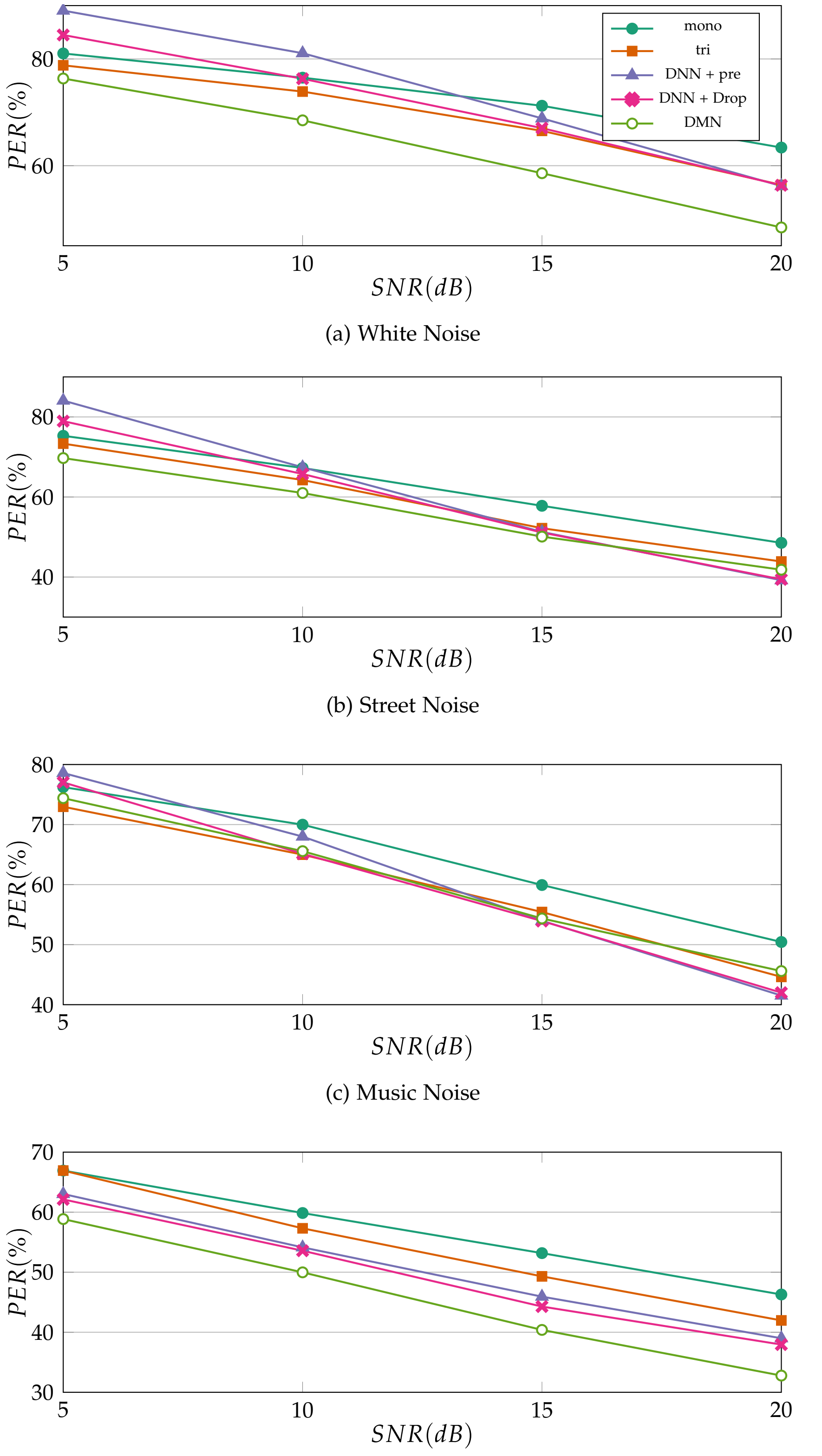
As can be observed, all the hybrid systems outperform the different versions of the baseline system, in both development and test sets. DNNs with random initialization, pre-training and dropout achieve similar results whereas the lowest PER is obtained with DMNs.

Method	Dev PER(%)	Eval PER(%)
Monophone	33.33	34.30
Triphone	28.64	30.42
Triphone + LDA + MLLT	26.44	27.62
Triphone + LDA + MLLT + SAT	23.56	25.79
DNN without pretraining (7 layers)	21.50	23.53
DNN with pretraining (8 layers)	21.05	23.05
DNN with dropout (4 layers)	21.98	23.84
DMN (5 layers)	19.15	21.01

Table 6.1: Recognition results in terms of PER(%) for the TIMIT development and core test sets in clean conditions.

Third, we tested the different systems in noisy conditions. Results achieved by the monophone baseline, the best triphone baseline (LDA+MLLT+SAT) and the best configurations for the hybrid DNN with pre-training, DNN with dropout, and DMN-based ASR systems in the noisy contaminated version of the TIMIT core test set, are shown in Figure 6.11 for the different types of noises and four different SNRs. As can be seen, DMN performs better in almost every situation for white, street and speaker noises in comparison to the other systems. The performance of DMN in white and speaker noises is specially remarkable. For street noise, results obtained with DMN are very similar to those achieved by the triphone GMM-HMM systems and both DNNs at high and medium SNRs; whereas it obtains the lowest PER at low SNRs. For music noise, the results of all the systems are very similar. As expected, dropout performs better than DNN with pre-training at low SNR in all the noises, given that dropout is very robust to the variations of the input. In addition, the gain in performance obtained by DMNs is due to the flexibility of the activation functions, that allows a better modeling of speech variability.

This experiment shows that DMN and other DNN schemes can be employed for robust speech recognition using hybrid architectures and that better performances can be achieved by these systems in comparison to those attained by traditional ones.



(a) White Noise

(b) Street Noise

(c) Music Noise

(d) Speaker Noise

Figure 6.11: Comparison of the performance of the different systems in terms of PER(%) for TIMIT test set in different noisy conditions.

7 CONVOLUTIONAL NEURAL NETWORKS AND BIO-INSPIRED FEATURES COMBINATION

7.1 INTRODUCTION

The advantages of Convolutional Neural Network (CNN) were presented in the last chapter. In particular their ability to be invariant to shift in time and equivariant in frequency together with the advantages provided by their smaller number of parameters in comparison with the traditional feedforward networks, allow to obtain better performance in computer vision and speech recognition tasks.

In this chapter, we propose a modification of the features presented in Chapter 4 and Chapter 5 to output a filter-bank like representation, that allows us to increase the recognition rates when used in conjunction to CNN and in particular with Residual Networks (ResNets) [13, 174].

The remainder of this chapter is organized as follows: Section 7.2 presents the previous work done in robust speech recognition based on the use of CNNs, Section 7.3 introduces deep residual learning and our proposed architecture that adapts the original computer vision ResNets to speech recognition. The modification of our features is presented in Section 7.4. Sections 7.5 and 7.6 contain, respectively, the experimental results achieved in comparison with other state of the art techniques and a discussion about them. Finally, we draw some conclusions in Section 7.7.

7.2 RELATED WORK

CNNs have become the state of the art in computer vision [13, 131, 132, 144] and also have been widely used for Automatic Speech Recognition (ASR) as, for example, in [175] where they were first employed, in [19] where a architecture with two convolutional layers where used in the first layers or in [176] where CNNs were utilized for Large Vocabulary Continuous Speech Recognition (LVCSR).

Normally CNNs are used in the configuration described in [177] that consists of two convolutional layers followed by four fully connected layers. This traditional set-up is used as one of our baselines in this chapter.

Another relevant application of CNN is [164] where this kind of neural networks and maxout activation functions are used together for phoneme recognition in the Texas Instruments and Massachusetts Institute of Technology (TIMIT) dataset.

Regarding the use of CNN for robust speech recognition, there exist several recent works, such as [161, 178] where simple architectures are employed.

It is worth noting [163] where the application of a very deep convolutional neural network to noisy speech recognition provides an important enhancement over the Aurora 4 dataset and the large Augmented Multi-party Interaction (AMI) corpus, over traditional feed-forward Deep Neural Network (DNN) and standard CNN architectures. Showing that very deep architectures are feasible for robust tasks. This work is our starting point and main baseline to improve on for our proposed methods. Specifically, in [163], authors present various network architectures based on the well known Visual Geometry Group Network (VGGNet) [131], consisting of small 3×3 convolutional filters and 2×2 pooling layers in a network with a high number of layers: in particular, they employ 10 convolutional layers and 4 fully connected ones, obtaining high recognition scores in the multi-condition training scenario (the mismatched training problem is not addressed).

Recently in the computer vision community, ResNets [13, 174] have been shown to improve the VGGNet baseline, by increasing the number of convolutional layers through the inclusion of shortcut connections. In this chapter we put forward that ResNets can improve speech recognition rates in noisy conditions given that they are capable to more effectively model the speech variability of data.

Also in this chapter we explore the combination of our previously presented features, the morphological filtering-based acoustic characteristics and the synchrony features, with CNN, aiming at modeling the last step, the neural processing, of the Human Auditory System (HAS) model presented in Section 3.2. Some specific modifications to our features are needed in order to be used as an input of artificial neural networks. These modifications are presented in Section 7.4.

Successful combinations of robust features and DNN back-ends to address the mismatched problem have been proposed in numerous works, as for example [156] where a review of different feature extraction strategies is presented showing that manually designed (as opposed to automatically learnt) feature extraction is still relevant or in [179] where a specific feature based on Locally-Normalized Filter-banks (LNFB) is tailored to a DNN architecture.

7.3 DEEP RESIDUAL LEARNING

Deep residual learning addresses the problem of degradation when the number of layers in a network is high. In a vanilla network (i.e. standard backpropagation trained) the stacked layers directly try to fit the underlying mapping. On the contrary, in ResNets the layers goal is to fit a *residual* function. The resulting residual mapping is more

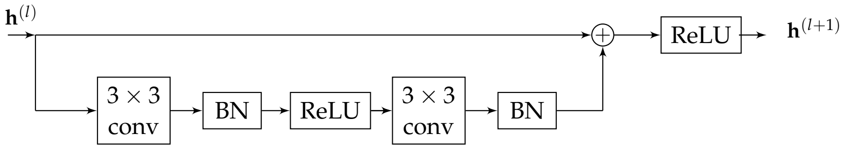


Figure 7.1: A typical residual unit. BN and ReLU activation function are applied after each convolution.

amenable for optimization since it is easier to push a residual to zero than to try to fit an underlying mapping.

Being $H(x)$ the target mapping, where x denotes the input of the first layer of the residual block, ResNets try to fit the mapping: $F(x) = H(x) - x$, and therefore the original function becomes $H(x) = F(x) + x$. This can be implemented by the addition of shortcut connections among the layers, as can be observed in Figure 7.1. The shortcut connections perform an identity mapping and the inputs are added to the output of the stacked layers. With adequate activation functions, this architecture is differentiable and therefore can be trained with traditional backpropagation.

The original ResNet [13] aimed at solving computer vision problems is composed of several residual units (Figure 7.1) stacked together, where each residual unit consists of two convolutional layers with 3×3 filter sizes, Batch Normalization (BN) [141] applied after each convolution and ReLU [180] activation functions after the first convolution and after the shortcut connection addition operation. Combining residual connections and batch normalization simplifies the training process since even when the weight matrix has small parameters (a typical cause of vanishing gradients) the addition of the input (to compute the residual) produces a more stable gradient across the network.

The original architecture is easy to implement when a layer output map has the same dimensions than the input it entails a simple addition. However if the layer output map is halved, the number of convolutional filters needs to be doubled, halving convolutional layers with a stride of 2 are applied instead of the more usual pooling layers. The dimensions of the shortcut connections and the mapping done by the convolution layers need to have the same size, to no include an extra parameter by introducing a projection of the shortcuts to mach dimensions, in the shortcuts a 1×1 convolution with a stride of 2 is applied to mach the dimensions.

The original architecture was built by stacking residual units, with a final global average pooling layer, a thousand units fully-connected layer and a final softmax output.

Our proposed architecture (Figure 7.2) adapts the original ResNet to speech recognition by taking into account the lower dimensions of the input in comparison with those of images. The input dimension in our case is 17×64 since we depart from a filter bank with 64 filters and a temporal context window of 17 frames. Thus, the ResNet is built by stacking as many residual units as to reduce the temporal dimension to one and the stride is applied every other unit. As in the original ResNet every time a stride is performed, the layer feature maps are doubled. This gives us a total of 6 residual units with 512 feature maps in the final layer. A final average pooling is performed to obtain an output size of 512 to finish with a fully connected layer of 1000 ReLU units and softmax output.

7.4 ROBUST FEATURES IN DEEP LEARNING-BASED ASR

In order to use our proposed features in a deep learning based ASR system, we need to modify the previously presented Power Normalized Cepstral Coefficients (PNCC) pipeline whose last stage consists on applying a Discrete Cosine Transform (DCT) to the log filter-bank energies to obtain the features in the cepstral domain. Since we would like to remain in the frequency domain this last operation is removed obtaining a filter-bank like representation.

The use of filter-bank representations are known to outperform the cepstral coefficients [178], and in the case of the PNCC we have found that the same principle applies as well. Removing the DCT allows us to have non-null cross-correlations between the components of the feature vectors and let DNNs take advantage of them [161, 168].

This modification increases our network input dimension (in particular 40 or 64 filters are customarily used) as we intended. Note that a high dimensional input representation is required in order to increase the number of convolutional layers.

Another modification needed is to substitute the power-law nonlinearity with the traditional logarithmic function since it performs better in conjunction with deep-learning back-ends. This is in line with [161, 178] where it was shown that Mel Filter Bank log-energiess (MelFBs) perform better than traditional Mel-Frequency Cepstral Coefficientss (MFCCs).

In summary, all the proposed feature sets are PNCC-based, where the non-linearity and DCT are removed. Four different sets are tested:

1. The classic log filter-bank representation based on MFCC, denoted as MelFB.
2. The filter-bank version of PNCC where the power-law non-linearity is replaced with the logarithmic non-linearity, denoted as Power Normalized Filter Banks (PNFB).
3. The PNFB with the masking modeling presented in Chapter 4, using the final Structuring Element (SE) and without Spectral Subtraction (SS). The Morphological Filtering (MF) is applied to the output of the PNFB, after the logarithmic non-linearity

as denoted in Figure 4.1 but without DCT. These features are denoted as Morphological Filtering Power Normalized Filter Banks (MF-PNFB).

4. Finally for testing the effect of the synchrony modeling presented in Chapter 5, we use the Modified Generalized Synchrony Detector with Power-Normalized Cepstral Coefficients Noise Reduction (MGSD-PNCC-NR) features with a logarithmic non-linearity and without the final DCT. Then the synchrony spectrum is filtered using the MF technique. These features are denoted as Modified Generalized Synchrony Detector with Morphological Filtering Power-Normalized Cepstral Coefficients Noise Reduction (MGSD-MF-PNCC-NR).

In all cases, Mean and Variance Normalization (MVN) is applied in a per utterance basis.

7.5 EXPERIMENTS

In this section we report on the effectiveness of ResNet and the proposed features in robust ASR using the Aurora-4 corpus [47] presented in Section 2.6.2. The Aurora 4 dataset allows us to train deep models with limited computational capabilities (a single Graphics Processing Unit (GPU) is available to perform the experiments) in a reasonable amount of time to implement and train multiples architectures.

The experiments were performed using the clean and multi-condition training sets and the standard test sets. The clean and multi-condition development sets were only used for parameter tuning and validation of the neural networks training process. The results presented in this section are averaged across all the tests sets.

Traditional triphone systems are used as a baseline and to obtain the alignments for training the neural networks using the Kaldi Speech Recognition Toolkit [30]. All the proposed deep learning architectures are built following a hybrid architecture (see section 2.4).

In summary, five acoustic modeling systems are evaluated: a traditional triphone Gaussian Mixture Model-Hidden Markov Model (GMM-HMM), a fully connected DNN, a state of the art CNN, a very deep CNN version and the proposed ResNet. All the systems are trained in clean and multi-condition scenarios with the four different feature extraction methods: MelFB, PNFB, MF-PNFB and MGSD-MF-PNCC-NR.

In the first four acoustic modeling architectures, the static acoustic parameters are composed of 40 filters, whereas in the ResNet case, the number of filters is set to 64.

The triphone GMM-HMM baseline system is trained employing the cepstral version of each four different feature extraction methods, Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT), using the Kaldi Aurora-4 recipe. The training recipe

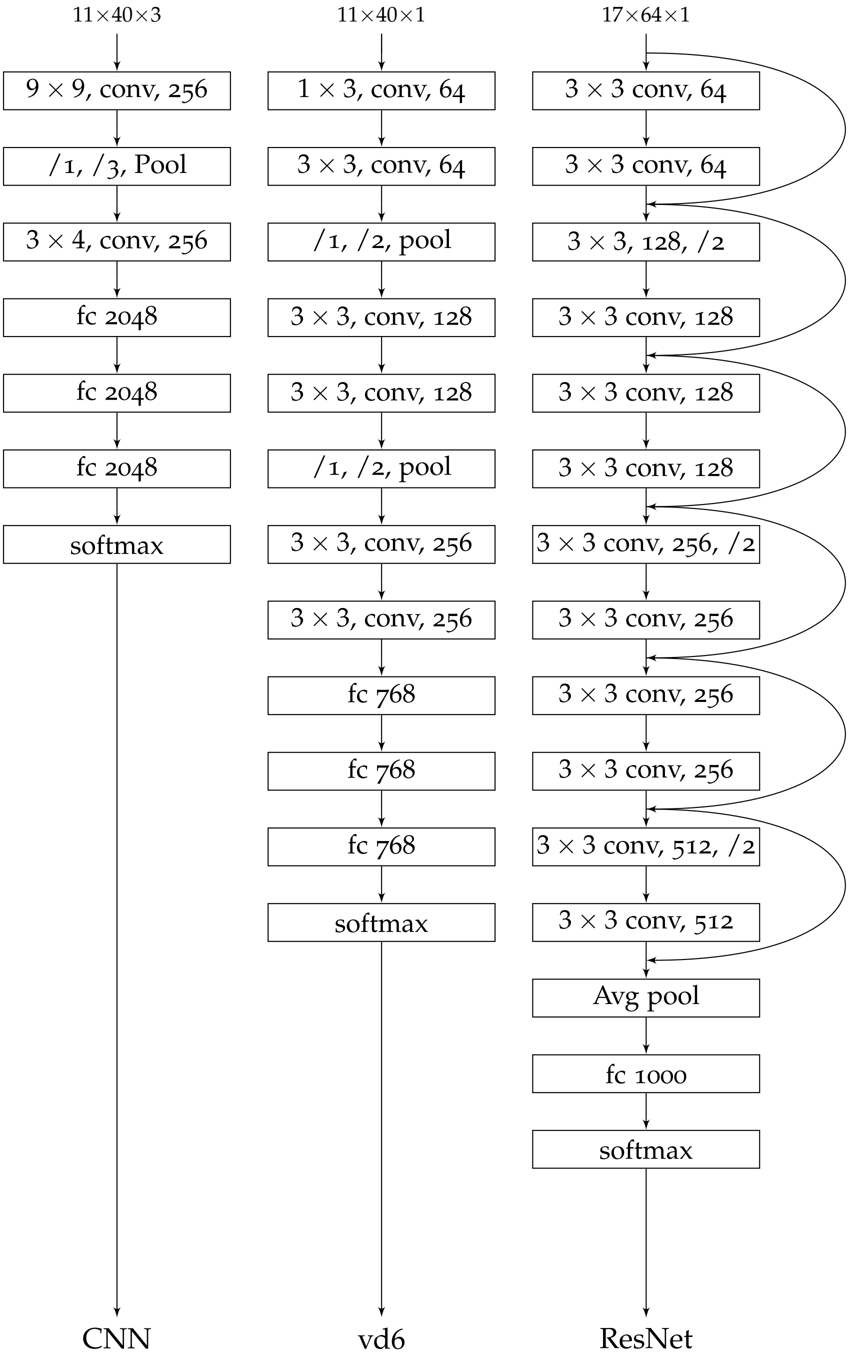


Figure 7.2: Block diagrams of the three CNN architectures. The input, convolution and pooling sizes are given in time × frequency scale. In the ResNet architecture, the stride is denoted as /2 and is applied in both dimensions.

starts building a monophone system and then employs the alignments obtained in this first stage to train an initial triphone system. A final triphone system with LDA and MLLT is subsequently retrained using the alignments of the later triphone system. DNN, CNN, ResNet models are trained using the alignments obtained by the final triphone system using each of the features tested. Clean training data is used to obtain the alignments in the mismatched case and the multi-condition data is used to obtain the alignments in the matched one.

The deep neural fully connected network baseline is composed of 5 hidden layers with 2048 units in each layer with ReLU activations functions and batch normalization in each layer. The input for this configuration consists of 40 features and the corresponding first and second order derivatives. In this case, an 11 frame context window is used.

The CNN-based architectures can be observed in Figure 7.2. The classic CNN proposed in [175, 177] is used as a CNN baseline consisting of two convolutional layers with 256 feature maps each one, with 9×9 and 3×3 filter sizes and a pooling layer in-between, followed by 3 fully connected layers with 2048 hidden units each. ReLU activation functions and BN are employed and the convolutions are performed with zero padding to mantain the feature maps dimensions. The input for this configuration is encoded as a $11 \times 40 \times 3$ where 3 feature maps are used for the features corresponding to the static, first and second order derivatives of the parameters respectively including an 11 frame temporal context window.

The so called very deep CNN is based on the vd6 network proposed by [163], where six 3×3 convolutional layers are staked together with non-overlapping pooling in the convolutional layers. After the first two and four layers, a 2 max-pooling operation is performed only in the frequency domain. In addition to the convolutional layers 3 fully connected ones are added. The vd6 only uses a 11×40 feature map as the first and second order derivatives are not considered.

The last configuration tested is the proposed ResNet architecture as described in Section 7.3. The input for this architecture is expanded to 17×64 to increase the number of residual layers, i.e. 64 features are used with a temporal context of 17 frames.

All the networks have a final softmax output layer whose output size is the number of senones of the final GMM-HMM system described above since we use a hybrid architecture.

The training pipeline used for the deep neural networks is almost the same for all the architectures: an Adam optimizer [129] with cross-entropy as a loss function with an initial learning rate of 0.001, Xavier initialization [142] for all the layers, early stopping with 3 retries of patience, where the learning rate is halved if the validation error is greater than the previous epoch. A maximum number of 20 epochs is allowed with no dropout and a batch size of 128 utterances for all the networks except for ResNet where 64 is used due computational limitations.

The neural networks are trained using Tensorflow [48]. The connection between Kaldi and Tensorflow can be found on [181]. Also the scripts used to train the networks can be found in [103].

Figure 7.3 shows the recognition results for each of the ASR systems in terms of the WER[%] averaged over all test sets for the clean training condition or mismatched condition. Figure 7.4 shows the results for multi-condition training or matched conditions. Both figures compare the four parameterizations considered: MelFB, PNFB, MF-PNFB and MGSD-MF-PNCC-NR, over the five acoustic modeling systems: GMM-HMM, fully connected DNN, CNN, very deep CNN and ResNet. Both figures show the 95% confidence intervals.

7.6 DISCUSSION

From the results shown in Figures 7.3 and 7.4, three main conclusions can be drawn.

First, we have analyzed the influence of ResNets on the ASR system performance. As can be observed, when using the conventional MelFB features, the ResNet architecture produces relative error reductions of 13.12% with respect to the plain DNN, 5.18% with respect to the CNN and 4.46% with respect to vd6 in clean training conditions. For the multi-condition training scenario and the same features, ResNet also attains the best recognition rate, achieving relative error reductions of 26.27%, 18.34% and 11.71% with respect to, respectively, DNN, CNN and vd6 systems. In all cases, these performance differences are statistically significant. Similar observations can be made with the PNFB features and the proposed auditory features MF-PNFB and MGSD-MF-PNCC-NR, in both, clean and multi-condition training. These results suggest that the proposed ResNet model, which was initially designed for computer vision tasks, is also suitable for speech recognition due to its remarkable generalization capabilities. In fact, ResNet outperforms the other acoustic models considered (GMM-HMM, DNN, CNN and vd6) in mismatched and matched conditions, showing its robustness against noise.

Second, the comparison of the different features, MelFB, PNFB, MF-PNFB and MGSD-MF-PNCC-NR, was investigated for clean training.

As expected, the PNFB, MF-PNFB and MGSD-MF-PNCC-NR features clearly outperforms the MelFB baseline for all the acoustic modelings considered. In case of the MGSD-MF-PNCC-NR it outperform only outperform the PNFB, MF-PNFB methods in the traditional GMM-HMM. It clearly obtain a decrease in performance over the MF-PNFB, and the results obtained are similar to the PNFB.

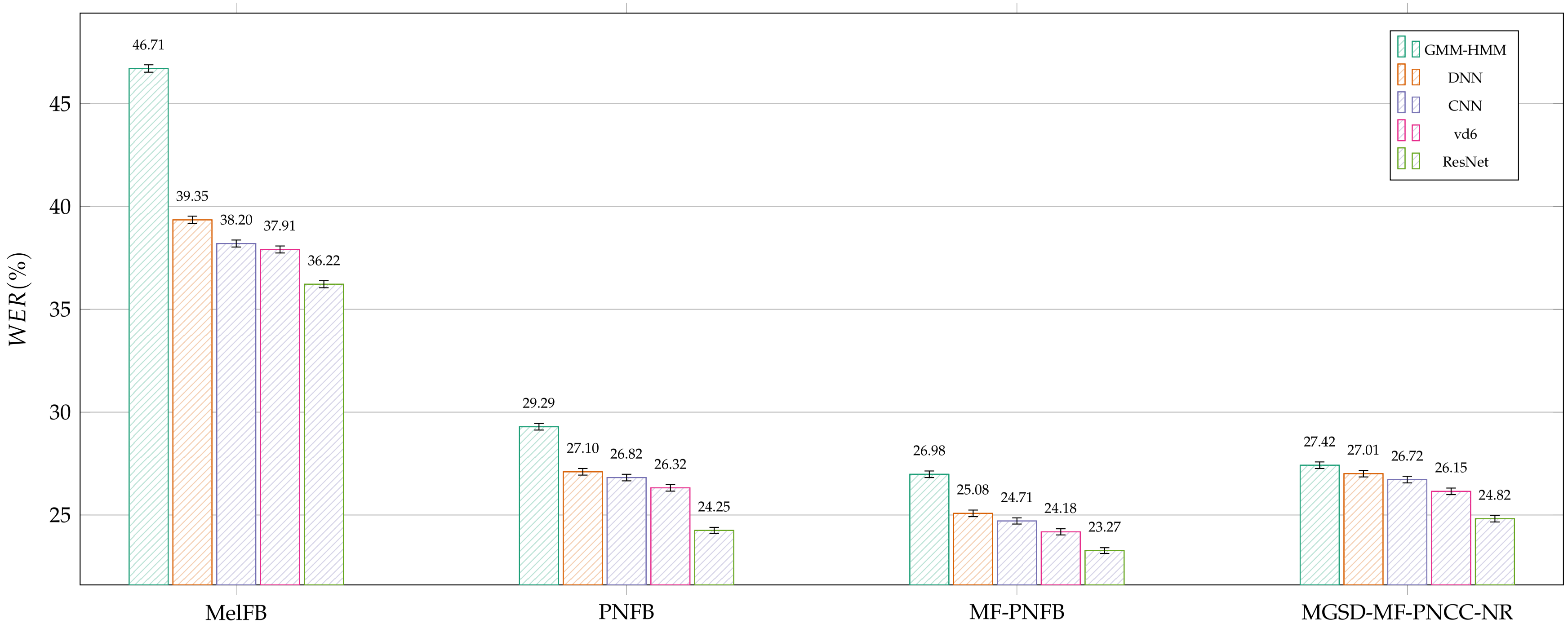


Figure 7.3: Recognition results in terms of WER(%) using the Aurora 4 dataset, averaged over all test sets in mismatched conditions, for all the architectures, and for the four types of features. Note that for the GMM-HMM baseline system, cepstral versions of the features are employed.

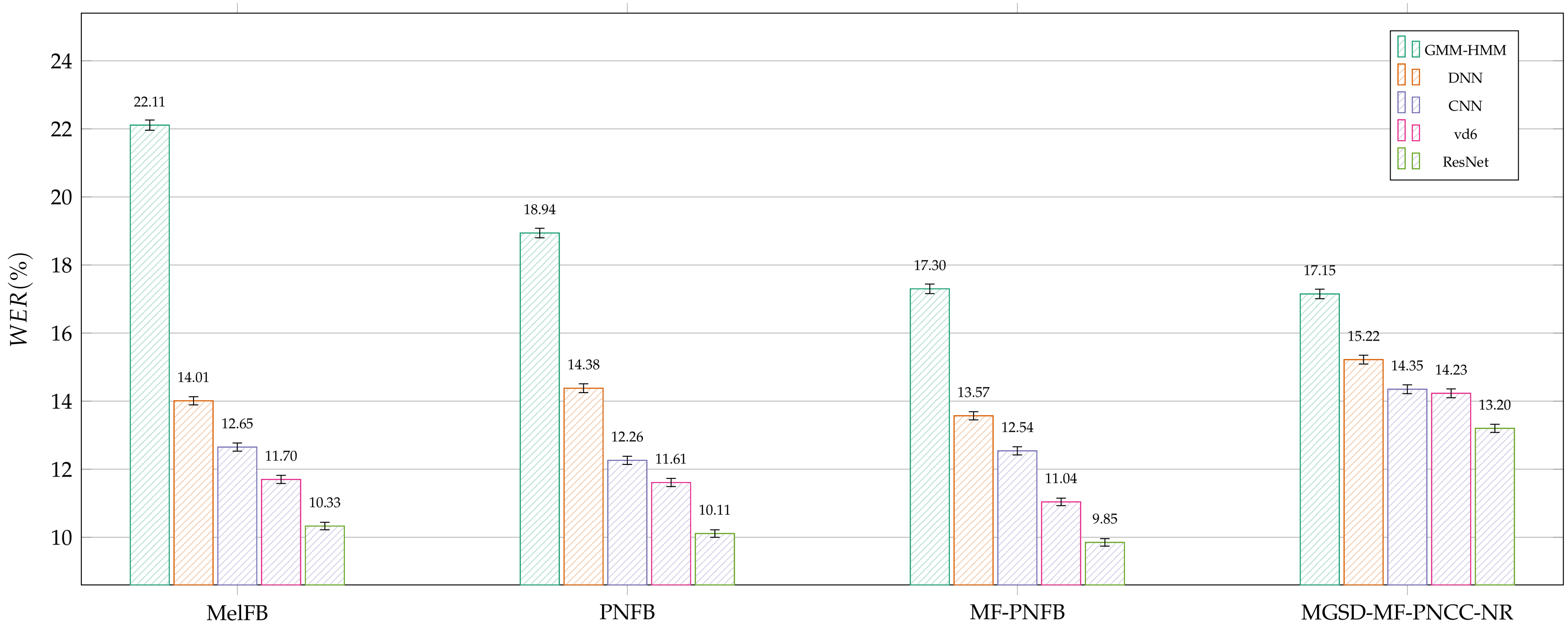


Figure 7.4: Recognition results in terms of WER(%) using the Aurora 4 dataset, averaged over all test sets in matched conditions, for all the architectures, and for the four types of features. Note that for the GMM-HMM baseline system, cepstral versions of the features are employed.

In particular, for the ResNet architecture the use of MF-PNFB with respect to MelFB obtains a relative error reduction of 36.22%, and 4.04% with respect the PNFB. Which is statistically significant in both cases.

It is worth noting that the result obtained by MelFB in combination to ResNet does not outperform the traditional GMM-HMM with MF-PNFB features. This observation indicates that the performance of ASR systems based on deep neural networks still suffer from important degradations when the mismatch between train and test data is high. Of course, this deterioration can be partially solved in some cases by the application of dataset augmentation techniques if some priors over the test data distribution are available. Nevertheless, when this solution is not feasible, the use of robust acoustic features (in particular, MF-PNFB) in deep neural networks architectures is helpful for reducing the error rate of the ASR system when mismatches between train and test data occur.

Third, the comparison of MelFB, PNFB, MF-PNFB and MGSD-MF-PNCC-NR was evaluated for the multi-condition training scenario.

Results show that for deep neural networks ASR systems, the use of our auditory motivated features produces small gains with respect to the conventional MelFB. For example, MF-PNFB achieves a relative error reduction of 4.64% with respect to MelFB and 2.57% with respect to PNFB for the ResNet architecture, although this performance difference is not statistically significant. As in the case of clean training the MGSD-MF-PNCC-NR only obtain advantage in the traditional GMM-HMM.

A reason for this behavior is that, in general, when the train and test data distributions are similar, the deep architectures can properly extract the more suitable features by themselves and, in consequence, the application of robust techniques on the feature extraction stage does not help significantly.

In particular, in the Aurora-4 the multi-condition train set has the same noises at different Signal-to-Noise Ratios (SNRs) than the test set, this allows us to conclude that the deep architectures can generalize under different signal noise scenarios when those particular noises are shown in the training stage. Nevertheless, it is worth mentioning that MF-PNFB do not damage the recognition rates, suggesting that its use is advisable to obtain a performance gain whenever it is plausible that the test data changes drastically from the train data as, for example, in real situations where channel and noise may vary over time.

To conclude, our best system (ResNet + MF-PNFB) attains a better relative error reduction than other state-of-the-art techniques for both, matched and mismatched cases in the Aurora-4 database. In comparison, for instance, we obtain better recognition rates than the features based on LNFB [179] in clean and multi-condition training, although the DNN-based ASR system in [179] is trained with alignments from the clean set in both scenarios. Also, our system outperforms in a significant amount the very deep convolutional networks (vd6 baseline) presented in [131] for both cases, clean and multi-condition training.

7.7 CONCLUSIONS

In this chapter, ResNets are employed for robust speech recognition in a hybrid ASR system showing a better performance than standard DNNs and state of the art CNNs in both matched and mismatched conditions using the Aurora-4 dataset. This behaviour is due to their well known convergence properties and generalization capabilities that allow a better modeling of speech variability. The other main contribution is the use of our adaptation of the robust input features of Chapters 4 and 5 in combination to the aforementioned deep neural network architectures. In particular, our modification of PNCCs with the masking modeling of Chapter 4, achieves significant improvements as compared with conventional features under mismatch and match conditions.

8 CONCLUSIONS AND FUTURE LINES OF RESEARCH

8.1 CONCLUSIONS

In this thesis we have proposed different methods to model the Human Auditory System (HAS) aiming at improving the recognition accuracy of Automatic Speech Recognition (ASR) systems in adverse conditions. Specifically, these methods increase the robustness of the ASR system by modeling the auditory masking and the synchrony effects. Both have been integrated in the well known Power Normalized Cepstral Coefficients (PNCC) feature extraction scheme complementing its strengths.

With respect to the acoustic modeling stage, we have proposed the use of a new architecture based on Convolutional Neural Networks (CNNs), namely Residual Network (ResNet), in a hybrid ASR system showing improved robustness. In addition, we have employed the proposed auditory motivated features together with this Deep Neural Network (DNN)-based back-end, demonstrating that the final system outperforms the baseline in mismatched conditions. Furthermore, DNNs and the ResNet in particular, can be understood as a way to model the last HAS stage, the auditory cortex.

Regarding the specific contributions of this Thesis to modeling the masking behavior, they can be summarized as follows:

- Despite ingrained intuitions that this imitation of auditory masking degrades the quality of the extracted features producing a blurring effect, we hypothesize that it is a sophisticated mechanism for selecting the most important parts of the spectrum from an intelligibility point of view, taking away irrelevant information and emphasizing the most robust parts of the spectrum.

- To model this effect image filtering techniques are employed, in particular a spectro-temporal representation based on the well known PNCCs is morphologically filtered with an Structuring Element (SE) specifically designed for this purpose.

- Empirical data derive form psychoacoustics experiments, in either temporal or frequency domains were interpolated to produce the three-dimensional SE for morphological filtering mentioned above, modeling both simultaneous and temporal masking.

The proposed methods provided the following results:

- The application of our masking model in conjunction with the PNCC representation produces a significant increase in recognition rates in Aurora 2, Aurora 4, Isolet and a noisy contaminated version of the Wall Street Journal datasets.

- Also the results show that our method improves the recognition rates in both hybrid and traditional Hidden Markov Model (HMM) based back-ends.

- The best results have been reached with a combination of PNCC, Spectral Subtraction (SS) and morphological processing.

- The results supports the theory that auditory masking is a human mechanism to improve the intelligibility.

The contributions to modeling of the synchrony effect of the auditory nerve are:

- The application of models of temporal patterns of auditory-nerve firings to enhance robustness of automatic speech recognition systems is presented.

- Contrary to most conventional feature extraction schemes (such as Mel-Frequency Cepstral Coefficients (MFCC) and Perceptually-based Linear Prediction (PLP)) based on short-time energies computed in each frequency band that discard the temporal patterns of auditory-nerve activity, our proposed features extract valuable information from those patterns.

- Different approaches to model the synchrony of auditory-nerve are proposed. In particular our feature extraction stage is based on a modified version of the Generalized Synchrony Detector (GSD) proposed by Seneff [66], and a modified version of the Average Localized Synchrony Rate (ALSR) proposed by Young and Sachs [71].

- A noise removal technique based on the noise suppression mechanisms included in the PNCC feature extraction procedure is proposed to complement the previous synchrony model.

A summary of the results achieved follows:

- The use of features based on auditory-nerve synchrony can indeed improve speech recognition accuracy in the presence of additive noise based on experiments using multiple standard speech databases.

- The recognition accuracy obtained using synchrony-based features is further increased if some form of noise removal is applied to the signal before the synchrony measure is estimated.

- The proposed noise removal technique based on PNCC is more effective towards this end than conventional spectral subtraction.

- Synchrony-based features based on Modified Generalized Synchrony Detector (MGSD) preceded by noise removal based on PNCC provides substantially better recognition accuracy than baseline PNCC features for speech that is degraded by white noise, interfering speakers, and reverberation. Improvements for speech in the presence of street noise and background music are more modest.

- These results suggest that synchronization in the HAS has a role in its remarkable robustness in addition to its well-known application to binaural localization.

- The addition of synchrony information and masking modeling to the PNCC completes some shortcomings of the PNCC related with modeling the HAS properties.

Finally, regarding the application of deep learning architectures the following contributions are reached:

- ResNets are employed for robust speech recognition in a hybrid ASR system over the Aurora 4 dataset.

- A modification of the original ResNet architecture is proposed to adapt the input dimensions to speech recognition tasks where the input size is smaller than the size used in the original version designed for computer vision tasks.

- The combination of our auditory motivated features with deep neural network architectures is addressed. In particular, the auditory masking and synchrony features integrated into the PNCC are tested.

Achieving the following results:

- The use of ResNets obtains better performance than standard DNNs and state of the art CNNs in both matched and mismatched conditions using the Aurora-4 dataset.

- The ResNets results profit from the well known convergence properties and generalization capabilities of ResNets allowing a robust modeling of speech.

- Regarding the auditory motivated features, our modification of the PNCC with masking modeling, achieves a significant improvement with respect to the conventional features when the mismatch between training and testing data is high, while maintaining the performance in matched scenarios.

8.2 FUTURE LINES OF RESEARCH

Several future lines of research have being identified. First, related to the masking modeling using morphological filtering, future work will focus on the introduction of the dependency of the masker strength. Also the designed morphological filter could be used to initialize the first layer in a CNN, by using different versions of our proposed SE.

Regarding the synchrony-based feature extraction, another future line of research would be to develop more efficient ways of combining the noise removal mechanisms provided by PNCC with the synchrony representation of GSD processing, as for now an intermediate transformation to the time domain is required. Also new alternatives to obtain a filter-bank representation of the synchrony feature extraction needs to be addressed, as for example, the use of other references to synchronize the filter outputs.

Finally, respect to the deep learning models explained, the assessment of the ResNet-based ASR system in larger datasets needs to be addressed. Also the combination of CNN with Recurrent Neural Networks (RNN) using our auditory features, to achieve higher recognition rates is worth exploring.

A

SPANISH INTRODUCTION AND CONCLUSIONS

A.1 INTRODUCCIÓN

A.1.1 *Motivación*

El habla es la forma natural de comunicación entre humanos. Desde los inicios de la humanidad, la comunicación ha evolucionado desde lenguaje de signos a un complejo lenguaje hablado, dada la necesidad de trasmitir cada vez mensajes más complejos. Actualmente, con la llegada de las tecnologías de la comunicación y la información, se ha producido un gran cambio en la manera en que nos relacionamos con las máquinas. Cada vez nos gusta más interactuar con los dispositivos electrónicos, como si de seres humanos se trataran.

Un papel importante en este proceso lo desempeñan los sistemas de reconocimiento de voz, que permiten sustituir las interfaces tradicionales no naturales, como son los teclados, ratones o pantallas, por la forma más frecuente de comunicación utilizada por el género humano: el habla.

El reconocimiento automático del habla cuenta con una larga historia de investigación, fracasos y éxitos. Su desarrollo a lo largo de los años, refleja el mismo proceso que los humanos emplearon para aprender a hablar y entender el habla. Desde el aprendizaje de sílabas, hasta el entendimiento de miles de palabras en oraciones complejas. Desde 1952, cuando los investigadores de los Laboratorios Bell diseñaron una máquina capaz de entender dígitos [7], el avance no ha parado.

Se han producido dos grandes puntos de inflexión a lo largo de la historia. El primero fue el cambio de enfoque, desde la comparación de plantillas a la modelización estadística. En particular, el uso de Hidden Markov Model (HMM) a principios de los años 80 [8, 9] que incrementaba el rendimiento de los reconocedores de habla y el número de palabras a reconocer. Estos avances permitieron desarrollar los primeros productos comerciales, capaces de reconocer el habla continua. Aun así, su rendimiento estaba todavía lejos del humano.

El segundo punto de inflexión fue la mejora sustancial del rendimiento logrado, utilizando técnicas de aprendizaje profundo [10]. En 2010, los sistemas de reconocimiento fueron la principal aplicación industrial del aprendizaje profundo [11], provocando que actualmente, la mayoría de los sistemas de reconocimiento de voz estén basados en estos algoritmos.

Como se puede observar, la tecnología de reconocimiento del habla presenta un desarrollo muy avanzado. En concreto el segundo hecho, la inclusion del aprendizaje profundo, ha dado la posibilidad de que las aplicaciones de reconocimiento del habla en tiempo real sean de uso habitual.

La mayoría de los *smartphones* y sistemas operativos cuentan con su propio motor de reconocimiento integrado. Sin olvidar también que resulta esencial en los sistemas de información y entretenimiento de los vehículos y otras aplicaciones industriales como *call-centers* o servicios de mensajería.

Existen numerosos retos en la utilización de reconocedores de voz en situaciones reales, donde estos sistemas se exponen a condiciones adversas, como son el ruido ambiental o la reverberación. Estos escenarios degradan de forma drástica su rendimiento, disminuyendo la posibilidad de que sean factibles en diversos escenarios como los vehículos aéreos no tripulados que son controlados de forma remota, para los cuales la fiabilidad es crítica.

Como se podrá ver en los siguientes capítulos, la robustez de los sistemas de reconocimento resulta ser un campo de estudio muy amplio, pero el rendimiento de estos en condiciones adversas, en particular cuando las condiciones con desconocidas, es todavía una pregunta abierta para la investigación, llegando a ser por tanto, uno de los mayores retos para las aplicaciones reales.

El sistema auditivo humano ha evolucionado durante miles de años para reconocer habla, de tal forma que los humanos cuentan con una extraordinaria capacidad para entender el habla en condiciones difíciles. Imitar las características más notables del sistema auditivo de una manera realista, puede ser una forma realista de abordar el problema.

A.1.2 *Marco socio-económico*

Las compañías tecnológicas más importantes del mundo están imponiendo la tecnología de reconocimiento de voz a sus clientes, en sus dispositivos y sistemas operativos móviles. Apple tiene Siri; Amazon ha diseñado Alexa; Google ha creado el Asistente de Google. Microsoft desarrolló Cortana y Facebook incluyó Oculus Voice como un asistente de reconocimiento del habla, en sus gafas de realidad virtual. El interés por llevar los sistemas de reconocimiento hacia otros campos nace de la necesidad de satisfacer la demanda de sus consumidores.

Siguiendo la línea de las empresas interesadas en las tecnologías del habla, los avances de investigación logrados en esta tesis en el área del reconocimiento robusto del habla se han desarrollado bajo el proyecto Situational Awareness Virtual Environment (SAVIER) de Airbus, donde nuevas interfaces hombre máquina se han desarrollado y probado en la futura estación de control en tierra, como medio de control y comandado de vehículos aéreos no tripulados.

A.1.3 *Objetivos*

El objetivo de esta tesis se centra en proponer soluciones al problema del reconocimiento de habla robusto; por ello, se han llevado a cabo dos líneas de investigación.

En la primera línea se han propuesto esquemas de extracción de características novedosos, basados en el modelado del comportamiento del sistema auditivo humano, modelando especialmente los fenómenos de enmascaramiento y sincronía. En la segunda, se propone mejorar las tasas de reconocimiento mediante el uso de técnicas de aprendizaje profundo, en conjunto con las características propuestas.

Los métodos propuestos tienen como principal objetivo, mejorar la precisión del sistema de reconocimiento cuando las condiciones de operación no son conocidas, aunque el caso contrario también ha sido abordado.

En concreto, nuestros principales propuestas son los siguientes:

- Simular el sistema auditivo humano con el objetivo de mejorar la tasa de reconocimiento en condiciones difíciles, principalmente en situaciones de alto ruido, proponiendo esquemas de extracción de características novedosos. Siguiendo esta dirección nuestros principales propuestas se detallan a continuación:
  - Modelar el comportamiento de enmascaramiento del sistema auditivo humano, usando técnicas del procesado de imagen sobre el espectro, en concreto, llevando a cabo el diseño de un filtro morfológico que captura este efecto.
  - Modelar el efecto de la sincronía que tiene lugar en el nervio auditivo.
  - La integración de ambos modelos en los conocidos Power Normalized Cepstral Coefficients (PNCC) [12].
- La aplicación de técnicas de aprendizaje profundo con el objetivo de hacer el sistema más robusto frente al ruido, en particular con el uso de redes neuronales convolucionales profundas, como pueden ser las redes residuales [13].
- Por último, la aplicación de las características propuestas en combinación con las redes neuronales profundas, con el objetivo principal de obtener mejoras significativas, cuando las condiciones de entrenamiento y test no coinciden.

A.1.4 *Estructura de la tesis*

El material presentado en esta tesis está organizado de la siguiente forma:

- El *Capítulo* 2 introduce los fundamentos del reconocimiento del habla, define el problema del reconocimiento de habla robusta y cómo puede solucionarse.
- El *Capítulo* 3 describe el sistema auditivo humano y cómo diferentes procesos de extracción de características han sido motivados por el mismo. Se revisan detalladamente las características motivadas auditivamente más comunes, dado que nuestras contribuciones están basadas en las mismas.
- El *Capítulo* 4 presenta nuestra contribución en el modelado del efecto de enmascaramiento que tiene lugar en el sistema auditivo y el diseño de características robustas.
- El *Capítulo* 5 presenta el efecto auditivo de la sincronía y cómo puede integrarse en la fase de extracción de características de un sistema de reconocimiento de habla.
- El *Capítulo* 6 introduce una descripción general de las técnicas de aprendizaje profundo y cómo se integran en el reconocimiento de habla moderno.
- El *Capítulo* 7 describe la aplicación de técnicas y arquitecturas novedosas, basadas en redes convolucionales profundas. También se evalúa la combinación de estos modelos con las características previamente presentadas.
- El *Capítulo* 8 presenta las principales conclusiones y futuras líneas de investigación.
- El *Apéndice* A contiene la introducción y conclusiones traducidas al castellano.

A.2 CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

A.2.1 *Conclusiones*

En esta tesis se han propuesto diferentes métodos que modelan el sistema auditivo humano, que aplicados a tareas de reconocimiento de habla proporcionan mejoras en la tasa de reconocimiento en condiciones adversas. En concreto, los métodos propuestos mejoran la robustez de los sistemas de reconocimiento, modelando el enmascaramiento auditivo y el efecto de sincronía. Ambos modelos han sido integrados en el esquema de extracción de características PNCC complementándolo.

Con respecto a las contribuciones obtenidas en la etapa del modelado acústico, se ha propuesto el uso de arquitecturas profundas basadas en redes convolucionales, en concreto, redes residuales (Residual Network (ResNet)), usando un reconocedor híbrido. Estos modelos se han utilizado junto con los esquemas de extracción de características propuestos, obteniendo mejoras significativas cuando las condiciones de test no son conocidas. El uso de estos modelos se puede interpretar como una forma de seguir modelando el sistema auditivo humano, en concreto la última etapa, el cortex auditivo.

Con respecto a las contribuciones de esta tesis al modelado del enmascaramiento, se pueden resumir en:

- Existe la intuición de que la imitación del comportamiento del enmascaramiento sonoro degrada las características obtenidas, dado que podría producir un efecto de distorsión del espectro. El modelo propuesto valida la teoría de que este comportamiento es un sofisticado mecanismo del sistema auditivo, que selecciona las partes más relevantes del espectro desde el punto de vista de la inteligibilidad, eliminando la información irrelevante y enfatizando las partes que son más robustas frente al ruido.

- Este efecto se ha modelado utilizando técnicas de filtrado morfológico sobre la representación espectral obtenida a través del los PNCCs. El elemento estructurante utilizado ha sido diseñado específicamente para este propósito.

- Se han interpolado datos empíricos derivados de experimentos psicoacústicos sobre los dominios del tiempo y la frecuencia, produciendo un elemento estructurante que modela el enmascaramiento sonoro simultáneo y temporal.

Los métodos propuestos han obtenido los siguientes resultados:

- La aplicación del modelo de enmascaramiento en conjunto con la representación espectral propuesta por los PNCC, produce mejoras significativas en las tasas de reconocimiento en las bases de datos Aurora 2, Aurora 4, Isolet y una versión contaminada de la base de datos Wall Street Journal.

- Los resultados muestran que el método propuesto mejora las tasas de reconocimiento en sistemas tradicionales e híbridos.

- El rendimiento mejor se ha obtenido con la combinación de PNCC, sustración espectral y el filtrado morfológico propuesto.

- Estos resultados soportan la teoría que afirma que el enmascaramiento sonoro es un mecanismo humano que mejora la inteligibilidad de la voz.

Las contribuciones resultantes del modelado del efecto de la sincronía se detallan a continuación:

- Se presenta la aplicación de modelos basados en los patrones temporales de los impulsos del nervio auditivo, para mejorar la robustez de los sistemas automáticos de reconocimiento del habla.

- A diferencia de la mayoría de los esquemas de extracción de características convencionales (como Mel-Frequency Cepstral Coefficients (MFCC) y PNCC), basados en la energía producida en cortos intervalos de tiempo para cada banda de frecuencia y descartando los patrones temporales de la actividad del nervio auditivo, las características propuestas extraen información valiosa de esos patrones.

- Se han propuesto diferentes aproximaciones para modelar el efecto de la sincronía del nervio auditivo. En particular, nuestra etapa de extracción de características se basa en una versión modificada de la respuesta del detector generalizado de sincronía (Generalized Synchrony Detector (GSD)) propuesto por Seneff y en una versión modificada de Average Localized Synchrony Rate (ALSR), propuesto por Young y Sachs.

- Otra contribución es el desarrollo de una técnica de reducción de ruido basada en los mecanismos propuestos en el proceso de extracción de características de los PNCCs y su integración en el citado modelo de sincronía.

Dando lugar a los siguientes resultados:

- El uso de características basadas en la sincronía del nervio auditivo mejora el rendimiento de los sistemas de reconocimiento, en presencia de ruido aditivo. Este resultado se ha demostrado en múltiples experimentos sobre diferentes bases de datos del estado del arte.

- La tasa de reconocimiento obtenida utilizando las características basadas en la sincronía es significativamente mejor, cuando se aplica en junto con alguna técnica de reducción de ruido.

- La técnica de eliminación de ruido propuesta basada en los PNCCs es más efectiva que otras técnicas convencionales, como puede ser la sustracción espectral.

- Las características basadas en la sincronía obtenidas a partir de Modified Generalized Synchrony Detector (MGSD), precedidas por la técnica de reducción frente al ruido basada en los PNCCs, obtienen mejoras sustanciales sobre los PNCCs originales. Estas mejoras son relevantes en condiciones de degradación por ruido blanco, por interlocutores que interfieren y reverberación. Para señales degradadas por ruido de calle y ruido musical, las mejoras resultan ser más modestas.

- Estos resultados sugieren que el efecto de la sincronía juega un rol importante en el sistema auditivo, respecto a la robustez que los humanos poseen frente a condiciones acústicas adversas, además de la ya conocida aplicación de localización binaural.
- La adición del modelado de la sincronía y del enmascaramiento sonoro a los PNCCs, los complementa en cuanto a lo que se refiere al modelado del sistema auditivo.

Finalmente, con respecto a la aplicación de arquitecturas profundas se alcanzaron las siguientes contribuciones:

- Las redes convolucionales residuales (ResNets) han sido integradas en un sistema de reconocimiento híbrido sobre la base de datos Aurora 4.
- Se ha propuesto una arquitectura adaptada a las dimensiones de entrada viables en un sistema de reconocimiento, en lugar de la arquitectura original diseñada para tareas de visión artificial.
- Otra contribución es la integración de las características auditivas propuestas en sistemas de reconocimiento híbridos compuestos de redes neuronales profundas. En concreto, se han integrado las características obtenidas de la combinación de los PNCCs, con un modelo del enmascaramiento sonoro basado en el filtrado morfológico y el modelo del efecto de la sincronía.

Dando lugar a los siguientes resultados:

- El uso de redes convolucionales residuales (ResNets) obtiene mejores resultados que las redes neuronales profundas y convolucionales propuestas en el estado del arte actual, sobre la base de datos Aurora 4. Este resultado es válido, tanto si las condiciones del conjunto de test son conocidas como si no lo son.
- La mejora obtenida por las redes convolucionales residuales (ResNets) se debe a sus conocida capacidad de convergencia y su buena generalización. Esto provoca que sean capaces de modelar mejor la variabilidad de la señal de voz en ambientes difíciles.
- En cuanto al uso de las características auditivas, la modificación de los PNCCs con el modelo del enmascaramiento sonoro, produce mejoras significativas con respecto a características convencionales cuando las condiciones del conjunto de entrenamiento y test son diferentes. El rendimiento se mantiene cuando las condiciones son conocidas.

A.2.2 *Líneas futuras de investigación*

A continuación se resumen posibles líneas futuras de investigación que se han identificado durante la realización de esta tesis. En primer lugar, en referencia al modelado del enmascaramiento, creemos que es necesario abordar la introducción de la dependencia de la intensidad de la señal enmascaradora en el filtrado morfológico. Además, diferentes versiones del filtro morfológico propuesto se podrían emplear para inicializar los filtros de la primera capa de una red convolucional profunda.

Con respecto a las características basadas en el efecto de la sincronía en el nervio auditivo, una línea futura de investigación sería llevar a cabo el desarrollo de una manera más eficaz de combinar los mecanismos de reducción de ruido de los PNCC con las características síncronas, dado que actualmente se require una transformación intermedia al dominio del tiempo. También es necesario proponer nuevas alternativas en la forma de medir la sincronía, en concreto proponiendo nuevas referencias con las que sincronizar la salida de los filtros.

Finalmente, en relación a los modelos de aprendizaje profundo abordados, se hace necesaria la evaluación de las redes convolucionales residuales en un sistema de reconocimiento de habla utilizando bases de datos más extensas. Otra posible línea sería el empleo de redes neuronales recurrentes utilizando las características propuestas.

BIBLIOGRAPHY

[1] F. de-la-Calle-Silos and R. M. Stern, "Synchrony-Based Feature Extraction for Robust Automatic Speech Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1158–1162, Aug. 2017.

[2] F. de-la-Calle-Silos, F. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "Morphologically Filtered Power - Normalized Cochleograms as Robust, Biologically Inspired Features for ASR," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 2070–2080, Nov. 2015.

[3] F. de-la-Calle-Silos, A. Gallardo-Antolín, and C. Peláez-Moreno, "Deep Maxout Networks Applied to Noise-Robust Speech Recognition," in *Advances in Speech and Language Technologies for Iberian Languages*, ser. Lecture Notes in Computer Science, vol. 8854, Springer International Publishing, 2014, pp. 109–118.

[4] F. de-la-Calle-Silos, F. J. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "ASR Feature Extraction with Morphologically - Filtered Power - Normalized Cochleograms," in *Proceedings of Interspeech (Annual Conference of the International Speech Communication Association)*, 2014, pp. 2430–2434.

[5] F. de-la-Calle-Silos, A. Gallardo-Antolín, and C. Peláez-Moreno, "An Analysis of Deep Neural Networks in Broad Phonetic Classes for Noisy Speech Recognition," in *Advances in Speech and Language Technologies for Iberian Languages: Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016, Proceedings*. Springer International Publishing, 2016, pp. 87–96.

[6] F. de-la-Calle-Silos, F. J. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "Preliminary experiments on the robustness of biologically motivated features for DNN-based ASR," in *Bioinspired Intelligence (IWOBI), 2015 4th International Work Conference on*, Jun. 2015, pp. 169–176.

[7] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic Recognition of Spoken Digits," *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637–642, 1952.

[8] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *The Journal of the Acoustical Society of America*, vol. 59, no. S1, S97–S97, 1976.

[9] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[10] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, 2012.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[12] C. Kim and R. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 4101–4104.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[14] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, ser. Kluwer international series in engineering and computer science: VLSI, computer architecture, and digital signal processing. Springer US, 1994.

[15] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[16] T. Virtanen, R. Singh, and B. Raj, *Techniques for Noise Robustness in Automatic Speech Recognition*, 1st. Wiley Publishing, 2012.

[17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[18] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSp Magazine*, 1986.

[19] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[20] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[21] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[22] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[23] R. Solera-Ureña, A. I. Garcia-Moral, C. Peláez-Moreno, M. Martínez-Ramón, and F. Díaz-de-María, "Real-time Robust Automatic Speech Recognition Using Compact Support Vector Machines," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1347–1361, 2012.

[24] P. Price, W. Fisher, J. Bernstein, and D. Pallett, *Resource Management RM2*, 1993.

[25] G. Hirsch, *Filtering and Noise Adding Tool*, http://dnt.kr.hsnr.de/download.html, 2005.

[26] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[27] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Norwell, MA, USA: Kluwer Academic Publishers, 1992.

[28] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An Overview of Noise-Robust Automatic Speech Recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 4, Apr. 2014.

[29] E. A. Habets, *Room Impulse Response Generator*, https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society, Dec. 2011.

[31] H. Hermansky, "Perceptual linear predictive (PLP) anlysis of speech," *J. Acoustic. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, 1990.

[32] H. Hermansky and N. Morgan, "RASTA processing of speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[33] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1315–1329, Jul. 2016.

[34] N. Morgan, "Deep and Wide: Multiple Layers in Automatic Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 7–13, Jan. 2012.

[35] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*, 1st. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

[36] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech and Signal Processing (ICASSP), 1979 IEEE International Conference on*, vol. 4, 1979, pp. 208–211.

[37] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[38] P. Scalart and J. Filho, "Speech enhancement based on a priori signal to noise estimation," in *Acoustics, Speech and Signal Processing (ICASSP), 1996 IEEE International Conference on*, vol. 2, 1996, pp. 629–632.

[39] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short time spectral amplitude estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 6, pp. 1109–1121, 1984.

[40] "Speech Processing Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression algorithms," European Telecommunications Standards Institute, Tech. Rep. ES 202 050 Rev. 1.1.5, 2007.

[41] K. Yu, M. Gales, and P. C. Woodland, "Unsupervised Adaptation With Discriminative Mapping Transforms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 714–723, May 2009.

[42] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[43] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, May 1996, 733–736 vol. 2.

[44] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 352–359, Sep. 1996.

[45] D. Pearce and H.-G. Hirsch, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ICSLP 2000, 6th International Conference on Spoken Language Processing*, 2000, pp. 16–19.

[46] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK book, ver 3.4*. Entropic Cambridge Research Laboratory, 2006.

[47] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," *Inst. for Signal and Information Process, Mississippi State University, Tech. Rep*, vol. 40, p. 94, 2002.

[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015.

[49] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91, Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.

[50] K. Vertanen, "Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments," Technical report). Cambridge, United Kingdom: Cavendish Laboratory, Tech. Rep., 2006.

[51] K. Bache and M. Lichman, *The ISOLET Spoken Letter Database*, last accessed: 2015-07-02.

[52] D. Gelbart, H. W., M. Holmberg, and N. Morgan, *Noisy ISOLET and ISOLET testbeds*.

[53] H. Bourlard and N. Morgan, "Hybrid HMM/ANN systems for speech recognition: Overview and new research directions," *Adaptive Process. of Sequences and Data Structures*, pp. 389–417, 1998.

[54] D. Johnson, *ICSI quicknet software package*, http://www.icsi.berkeley.edu/Speech/qn.html.

[55] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM*, https://catalog.ldc.upenn.edu/LDC93S1, 1993.

[56] Y. Miao, "Kaldi+PDNN: Building DNN-based ASR Systems with Kaldi and PDNN," *CoRR*, 2014.

[57] H. Fastl and E. Zwicker, *Psycho-acoustics: Facts and Models*, 3rd ed. Springer, 2007.

[58] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Am.*, vol. 68, p. 1523, 1980.

[59] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude of pitch," *J. Acoust. Soc. Am.*, vol. 8, pp. 185–190, 1937.

[60] B. Glasberg and B. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Res.*, vol. 47, no. 1-2, pp. 103–138, 1990.

[61] B. Moore and B. Glasberg, "A revised model of loudness perception applied to cochlear hearing loss," *Hearing Research*, vol. 188, no. 1-2, pp. 70–88, 2004.

[62] ——, "Suggested formula for calculating auditory-filter bandwidths and excitation patterns," *J. Acoust. Soc. Am.*, vol. 74, p. 750, 1983.

[63] R. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," *Auditory physiology and perception*, vol. 83, pp. 429–446, 1992.

[64] M. G. Heinz, X. Zhang, I. C. Bruce, and L. H. Carney, "Auditory nerve model for predicting performance limits of normal and impaired listeners," *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 91–96, 2001.

[65] H. Fletcher and W. A. Munson, "Loudness, Its Definition, Measurement and Calculation," *The Journal of the Acoustical Society of America*, vol. 5, no. 2, pp. 82–108, Oct. 1933.

[66] S. Seneff, "A Joint Synchrony/Mean-rate Model of Auditory Speech Processing," in *Readings in Speech Recognition*, A. Waibel and K.-F. Lee, Eds., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 101–111.

[67] O. Ghitza, "Auditory Nerve Representation As a Front-end for Speech Recognition in a Noisy Environment," *Comput. Speech Lang.*, vol. 1, no. 2, pp. 109–130, Dec. 1986.

[68] R. Lyon, "A computational model of filtering, detection, and compression in the cochlea," in *Acoustics, Speech and Signal Processing (ICASSP), 1982 IEEE International Conference on*, vol. 7, May 1982, pp. 1282–1285.

[69] ——, "A computational model of binaural localization and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 1983 IEEE International Conference on*, vol. 8, Apr. 1983, pp. 1148–1151.

[70] N. Y.-S. Kiang, T. Watanabe, W. C. Thomas, and L. F. Clark, *Discharge Patterns of Single Fibers in the Cat's Auditory Nerve*. MIT Press, 1966.

[71] E. D. Young and M. B. Sachs, "Representation of steady-state vowels in the temporal aspects of the discharge patterns of populations of auditory-nerve fibers," *J. Acoustic. Soc. Amer.*, vol. 66, pp. 1381–1403, 1979.

[72] J. B. Allen, "Cochlear modeling," *IEEE ASSP Magazine*, vol. 1, pp. 3–29, 1985.

[73] R. Meddis, "Simulation of mechanical to neural transduction in the auditory receptor," *The Journal of the Acoustical Society of America*, vol. 79, no. 3, pp. 702–711, 1986.

[74] X. Zhang, M. G. Heinz, I. C. Bruce, and L. H. Carney, "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*, vol. 109, no. 2, p. 648, 2001.

[75] C. Kim, Y. Chiu, and R. Stern, "Physiologically-motivated synchrony-based processing for robust automatic speech recognition.," in *Proceedings of Interspeech (7th International Conference on Speech Communication and Technology)*, 2006, pp. 1483–1486.

[76] H. Hermansky, B. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," in *Acoustics, Speech and Signal Processing (ICASSP), 1985 IEEE International Conference on*, vol. 10, Apr. 1985, pp. 509–512.

[77] D.-S. Kim, S.-Y. Lee, and R. Kil, "Auditory Proccessing of Speech Signals for Robust Speech Recognition in Real World Noisy Environments," *IEEE Trans. on Speech and Audio Processing*, vol. 7, pp. 55–59, 1999.

[78] A. M. A. Ali, J. V. der Spiegel, and P. Mueller, "Robust auditory-based speech processing using the average localized synchrony detection," *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 279–292, 1999.

[79] U. H. Yapanel and J. H. Hansen, "A new perceptually motivated MVDR-based acoustic front-end (PMVDR) for robust automatic speech recognition," *Speech Communication*, vol. 50, no. 2, pp. 142–152, 2008.

[80] F. Müller and A. Mertins, "Contextual invariant-integration features for improved speaker-independent speech recognition," *Speech Communication*, vol. 53, no. 6, pp. 830–841, 2011.

[81] N. Moritz, J. Anemuller, and B. Kollmeier, "Amplitude modulation spectrogram based features for robust speech recognition in noisy and reverberant environments," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 5492–5495.

[82] N. Moritz, J. Anemüller, and B. Kollmeier, "An Auditory Inspired Amplitude Modulation Filter Bank for Robust Feature Extraction in Automatic Speech Recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 1926–1937, Nov. 2015.

[83] A. Fazel and S. Chakrabartty, "Sparse Auditory Reproducing Kernel (SPARK) Features for Noise-Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1362–1371, May 2012.

[84] B. T. Meyer and B. Kollmeier, "Robustness of spectro-temporal features against intrinsic and extrinsic variations in automatic speech recognition," *Speech Commun.*, vol. 53, no. 5, pp. 753–767, 2011.

[85] B. T. Meyer, C. Spille, B. Kollmeier, and N. Morgan, "Hooking up spectro-temporal filters with auditory-inspired representations for robust automatic speech recognition," in *Proceedings of Interspeech (13th International Conference on Speech Communication and Technology)*, 2012.

[86] B. T. Meyer, S. V. Ravuri, M. R. Schdler, and N. Morgan, "Comparing Different Flavors of Spectro-Temporal Features for ASR," in *Proceedings of Interspeech (12th International Conference on Speech Communication and Technology)*, 2011, pp. 1269–1272.

[87] M. Heckmann, X. Domont, F. Joublin, and C. Goerick, "A hierarchical framework for spectro-temporal feature extraction," *Speech Communication*, vol. 53, no. 5, pp. 736–752, 2011.

[88] A. Hurmalainen and T. Virtanen, "Modelling spectro-temporal dynamics in factorisation-based noise-robust automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4113–4116.

[89] C. Martínez, J. Goddard, D. Milone, and H. Rufiner, "Bioinspired sparse spectro-temporal representation of speech for robust classification," *Computer Speech and Language*, vol. 26, no. 5, pp. 336–348, 2012.

[90] R. Stern and N. Morgan, "Hearing Is Believing: Biologically Inspired Methods for Robust Automatic Speech Recognition," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 34–43, Nov. 2012.

[91] K. Paliwal and B. T. Lilly, "Auditory masking based acoustic front-end for robust speech recognition," in *TENCON 97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE*, vol. 1, 1997, 165–168 vol.1.

[92] Y. Hu and P. Loizou, "Incorporating a psychoacoustical model in frequency domain speech enhancement," *Signal Processing Letters, IEEE*, vol. 11, no. 2, pp. 270–273, 2004.

[93] S. Haque, "Utilizing auditory masking in automatic speech recognition," in *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, 2010, pp. 1758–1764.

[94] J. Cadore, F. J. Valverde-Albacete, A. Gallardo-Antolín, and C. Peláez-Moreno, "Auditory-Inspired Morphological Processing of Speech Spectrograms: Applications in Automatic Speech Recognition and Speech Enhancement," *Cognitive Computation*, vol. 5, no. 4, pp. 426–441, 2013.

[95] J. Cadore, A. Gallardo-Antolín, and C. Peláez-Moreno, "Morphological Processing of Spectrograms for Speech Enhancement," in *Advances in Nonlinear Speech Processing*, Springer-Verlag, 2011, pp. 224–231.

[96] J. Hansen, "Morphological constrained feature enhancement with adaptive cepstral compensation (MCE-ACC) for speech recognition in noise and Lombard effect," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 598–614, 1994.

[97] G. V. Békésy, "On the Resonance Curve and the Decay Period at Various Points on the Cochlear Partition," *The Journal of the Acoustical Society of America*, vol. 21, no. 3, pp. 245–254, 1949.

[98] E. Zwicker and A. Jaroszewski, "Inverse frequency dependence of simultaneous tone-on-tone masking patterns at low levels," *The Journal of the Acoustical Society of America*, vol. 71, no. 6, pp. 1508–1512, 1982.

[99] W. Jesteadt, S. P. Bacon, and J. Lehman, "Forward masking as a function of frequency, masker level, and signal delay," *The Journal of the Acoustical Society of America*, vol. 71, no. 4, pp. 950–962, 1982.

[100] G. Matheron and J. Serra, "The birth of mathematical morphology," in *Proc. 6th Int. Symp. Mathematical Morphology*, Sydney, Australia, 2002, pp. 1–16.

[101] E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing*, ser. Tutorial Texts in Optical Engineering. SPIE press, 2003.

[102] J. Cadore, C. Peláez-Moreno, and A. Gallardo-Antolín, "Morphological Processing of a Dynamic Compressive Gammachirp Filterbank for Automatic Speech Recognition," in *IberSPEECH 2012*, 2012.

[103] F. de-la-Calle-Silos, *Personal web*, http://www.tsc.uc3m.es/~fsilos/.

[104] N. A. Weiss and M. J. Hasset, *Introductory Statistics*. Addison-Wesley, 1993.

[105] D. H. Johnson, "The relationship between spike rate and synchrony in responses of auditory-nerve fibers to single tones," *J. Acust. Soc. Amer.*, vol. 68, no. 4, pp. 1115–1122, Aug. 1980.

[106] J. E. Rose, J. E. Hind, D. J. Anderson, and J. F. Brugge, "Some effects of stimulus intensity on response of auditory nerve fibers in the squirrel monkey.," *Journal of Neurophysiology*, vol. 34, no. 4, pp. 685–699, 1971.

[107] A. Dreyer and B. Delgutte, "Phase locking of auditory-nerve fibers to the envelopes of high-frequency sounds: implications for sound localization," *J. Neurophysiol.*, vol. 96, no. 5, pp. 2327–2341, 2006.

[108] V. Mitra, H. Franco, and M. Graciarena, "Damped oscillator cepstral coefficients for robust speech recognition," in *Annual Conference of the International Speech Communication Association*, Lyon, France, Aug. 2013.

[109] V. Poblete, F. Espic, S. King, R. M. Stern, F. Huenupan, J. Fredes, and N. B. Yoma, "A perceptually-motivated low-complexity instantaneous linear channel normalization technique applied to speaker verification," *Computer Speech and Language*, vol. 31, pp. 1–27, Jan. 2015.

[110] M. Slaney, *Auditory Toolbox (V.2)*, http://www.slaney.org/malcolm/pubs.html, 1998.

[111] R. Schlüter, I. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007, pp. 649–652.

[112] C. Kim, K. Kumar, B. Raj, and R. M. Stern, "Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain," in *Proceedings of Interspeech (10 th Annual Conference of the International Speech Communication Association)*, 2009, pp. 2495–2498.

[113] J. B. Allen and L. R. Rabiner, "A unified theory of short-time spectrum analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, Nov. 1977.

[114] K.-F. Lee and R. Reddy, *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. Norwell, MA, USA: Kluwer Academic Publishers, 1988.

[115] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[116] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[117] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[118] A. Munoz, "Machine Learning and Optimization," 2014. [Online]. Available: https://www.cims.nyu.edu/~munoz/files/ml_optimization.pdf.

[119] T. M. Mitchell, *Machine learning*, 1997.

[120] J. Shlens, "A Tutorial on Principal Component Analysis," *CoRR*, vol. abs/1404.1100, 2014.

[121] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, 2012.

[122] J. Sjöberg and L. Ljung, "Overtraining, regularization and searching for a minimum, with application to neural networks," *International Journal of Control*, vol. 62, no. 6, pp. 1391–1407, 1995.

[123] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain," *Psychological Review*, pp. 65–386, 1958.

[124] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1," in, D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds., Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[125] M. D. Richard and R. P. Lippmann, "Neural Network Classifiers Estimate Bayesian a posteriori Probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[126] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," PhD thesis, Harvard University, Cambridge, MA, 1974.

[127] Y. Lecun, "Une procédure d'apprentissage pour réseau à seuil asymétrique," *Proceedings of Cognitiva 85, Paris*, pp. 599–604, 1985.

[128] T. S. Ferguson, “An Inconsistent Maximum Likelihood Estimate,” *Journal of the American Statistical Association*, vol. 77, no. 380, pp. 831–834, 1982.

[129] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[130] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” *CoRR*, vol. abs/1311.2901, 2013.

[131] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv technical report*, 2014.

[132] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9.

[133] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines.,” in, ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., vol. 7700, Springer, 2012.

[134] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion,” *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[135] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The Loss Surfaces of Multilayer Networks.,” in *AISTATS*, 2015.

[136] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[137] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.

[138] L. Breiman, “Bagging Predictors,” *Machine Learning*, vol. 24, no. 2, 1996.

[139] Y. Miao and F. Metze, “Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training.,” in *Annual Conference of the International Speech Communication Association*, ISCA, 2013, pp. 2237–2241.

[140] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout Networks,” *ArXiv e-prints*, 2013.

[141] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 448–456.

[142] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[143] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[144] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, p. 2012.

[145] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning.," *CoRR*, vol. abs/1603.07285, 2016.

[146] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, 2012.

[147] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," *CoRR*, vol. abs/1512.02595, 2015.

[148] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[149] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 369–376.

[150] G. Gosztolya, T. Grósz, and L. Tóth, "GMM-Free Flat Start Sequence-Discriminative DNN Training," in *Proceedings of Interspeech (17 th Annual Conference of the International Speech Communication Association)*, 2016, pp. 3409–3413.

[151] A. Senior, H. Sak, and K. Rao, "Flatstart-CTC: a new acoustic model training procedure for speech recognition," in *ICASSP*, 2016.

[152] H. Sak, A. W. Senior, K. Rao, and F. Beaufays, “Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition,” *CoRR*, vol. abs/1507.06947, 2015.

[153] K. Rao, F. Peng, H. Sak, and F. Beaufays, “Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4225–4229.

[154] G. Pundak and T. Sainath, “Lower Frame Rate Neural Network Acoustic Models,” in *Annual Conference of the International Speech Communication Association*, 2016.

[155] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, 2012.

[156] V. Mitra, H. Franco, R. Stern, J. V. Hout, L. Ferrer, M. Graciarena, W. Wang, D. Vergyri, A. Alwan, and J. H. Hansen, “Robust Features in Deep Learning-Based Speech Recognition,”

[157] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “Achieving Human Parity in Conversational Speech Recognition,” *CoRR*, vol. abs/1610.05256, 2016.

[158] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, “JHU ASpIRE system: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMS,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 539–546.

[159] L. Couvreur, F. P. D. Mons, and C. Couvreur, *Robust Automatic Speech Recognition In Reverberant Environments By Model Selection*, 2001.

[160] M. Karafiát, F. Grézl, L. Burget, I. Szöke, and J. Cernocký, “Three ways to adapt a CTS recognizer to unseen reverberated speech in BUT system for the ASpIRE challenge,” in *Proceedings of Interspeech (16 th Annual Conference of the International Speech Communication Association)*, 2015, pp. 2454–2458.

[161] M. L. Seltzer, D. Yu, and Y. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on,*, 2013.

[162] Y. Miao, F. Metze, and S. Rawat, “Deep Maxout Networks for Low-Resurce Speech Recognition,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, 2013.

[163] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 24, no. 12, pp. 2263–2276, Dec. 2016.

[164] L. Tóth, "Convolutional deep maxout networks for phone recognition," in *Proceedings of Interspeech (Annual Conference of the International Speech Communication Association)*, ISCA, 2014, pp. 1078–1082.

[165] V. Mitra, W. Wang, H. Franco, Y. Lei, C. Bartels, and M. Graciarena, "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[166] V. Mitra, W. Wang, and H. Franco, "Deep convolutional nets and robust features for reverberation-robust speech recognition," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec. 2014, pp. 548–553.

[167] O. Abdel-Hamid, A. r. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 4277–4280.

[168] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Retrieving Tract Variables From Acoustics: A Comparison of Different Machine Learning Strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1027–1045, Dec. 2010.

[169] S.-Y. Chang and N. Morgan, "Robust CNN-based Speech Recognition With Gabor Filter Kernels," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[170] M. Gales and P. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Language*, vol. 10, no. 4, pp. 249–264, 1996.

[171] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[172] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2013, pp. 55–59.

[173] L. Deng, D. Yu, and A. Acero, "Structured speech modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1492–1504, Sep. 2006.

[174] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," *CoRR*, vol. abs/1603.05027, 2016.

[175] T. N. Sainath, A. r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8614–8618.

[176] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very Deep Multilingual Convolutional Neural Networks for LVCSR," *CoRR*, vol. abs/1509.08967, 2015.

[177] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran, "Deep Convolutional Neural Networks for Large-scale Speech Tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015, Special Issue on "Deep Learning of Representations".

[178] T. Drugman, Y. Stylianou, L. Chen, X. Chen, and M. J. F. Gales, "Robust excitation-based features for Automatic Speech Recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4664–4668.

[179] J. Fredes, J. Novoa, S. King, R. M. Stern, and N. B. Yoma, "Locally Normalized Filter Banks Applied to Deep Neural-Network-Based Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 377–381, Apr. 2017.

[180] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 2010, pp. 807–814.

[181] V. Renkens, *Kaldi with TensorFlow Neural Net*, https://github.com/vrenkens/tfkaldi.

COLOPHON

This document was typeset in LaTeX using the typographical look-and-feel `classicthesis`. Most of the graphics are generated using `pgfplots` and `pgf/tikz`.